BIG-IP® TMOS®: Implementations

Version 11.5.1

Table of Contents

Customizing the BIG-IP Dashboard

Overview: BIG-IP dashboard customization

The BIG-IP® dashboard displays system statistics in selectable graphs, gauges, and tables. In addition to the pre-defined views, you can create custom combinations of the dashboard windows, called *views*, and save them in groups, called *view sets*. You can combine windows from different BIG-IP modules in a single view, or use just the windows you want for a single module. Windows are available only for those modules that you have licensed and provisioned.

Note: *The view set name for all pre-defined views is* `standard`.

Customizing the BIG-IP dashboard

You can create custom dashboard displays using the windows for any modules that are available on the BIG-IP® system.

1. On the Main tab, expand **Overview**, and click **Dashboard**.
   A separate window opens for the BIG-IP dashboard.
2. On the Views control bar, click the Create custom view icon.
   A blank canvas opens in design mode. The Dashboard Windows Chooser displays the available windows, grouped by module. You can click a module to display the available windows.
3. From the Dashboard Windows Chooser, drag and drop the windows you want onto the canvas.

   After you drag a window to the canvas, you can resize it or change it to display the information you want by clicking a tab or filter.

   Note: *The windows are not active when in design mode, so the data does not update in real time.*
4. When you have placed the windows you want onto the canvas, click the Save icon on the Custom Views control bar.
   The Save View popup window opens.
5. Type a name for the view.
6. Type a new name for the view set, or select from the list.
7. Click **OK**.
   The new view is saved, and appears in the **Views** list.
8. Click the double-gear icon on the Custom Views control bar to return to active mode.
   The dashboard displays the custom view you just created, and updates the display with real-time data.

Creating an Active-Standby Configuration Using the Setup Utility

Overview: Creating a basic active-standby configuration

This implementation describes how to use the Setup utility to configure two new BIG-IP® devices that function as an active-standby pair. An *active-standby pair* is a pair of BIG-IP devices configured so that one device is actively processing traffic while the other device remains ready to take over if failover occurs. The two devices synchronize their configuration data and can fail over to one another in the event that one of the devices becomes unavailable.

Important: *The same version of BIG-IP system software must be running on all devices in the device group.*

First, you run the Setup utility on each device to configure base network components (that is, a management port, administrative passwords, and the default VLANs and their associated self IP addresses). Continue running it on each device to establish a trust relationship between the two devices, and create a Sync-Failover type of device group that contains two member devices.

After the Setup utility is run on both devices, each device contains the default traffic group that the BIG-IP system automatically created during setup. A *traffic group* represents a set of configuration objects (such as floating self IP addresses and virtual IP addresses) that process application traffic. This traffic group actively processes traffic on one of the two devices, making that device the active device. When failover occurs, the traffic group becomes active on (that is, floats to) the peer BIG-IP device.

By default, the traffic group contains the floating self IP addresses of the default VLANs. Whenever you create additional configuration objects such as self IP addresses, virtual IP addresses, and SNATs, the system automatically adds these objects to the default traffic group.

Example

In this configuration example, the device group is named `Device Group A`. This device group contains two BIG-IP devices, named `Device 1` and `Device 2`, and these two devices are peers of one another. The default traffic group, named `traffic-group-1`, resides on each device.

`Device 1` actively processes traffic because `traffic-group-1` is in an Active state on that device. `Device 2` remains idle until failover occurs because `traffic-group-1` is in a Standby state on that device.

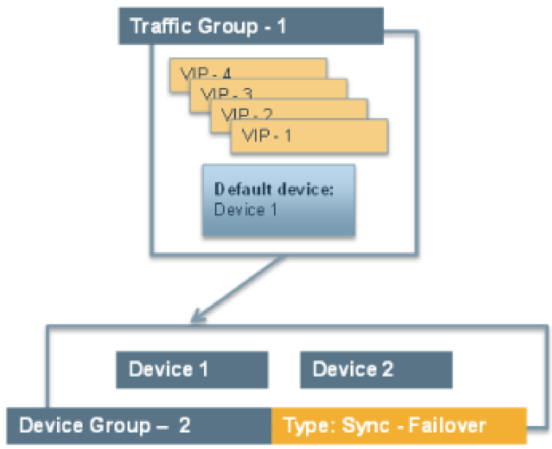


Figure 1: Example active-standby configuration

By implementing this configuration, you ensure that:

- Each device has base network components (such as self IPs and VLANs) configured.
- The two devices can synchronize their configuration to one another.
- Failover capability and connection mirroring are enabled on each device.

Task summary

The configuration process for a BIG-IP® system entails running the Setup utility on each of the two BIG-IP devices. When you run the Setup utility, you perform several tasks. Completing these tasks results in both BIG-IP devices being configured properly for an active-standby implementation.

***Important:** After using the Setup utility to create an active-standby configuration, you can re-enter the utility at any time to adjust the configuration. Simply click the F5 logo in the upper-left corner of the BIG-IP Configuration utility, and on the Welcome screen, click **Run the Setup Utility**. Then page through the utility to find the appropriate screens.*

Licensing and provisioning the BIG-IP system
Configuring a device certificate
Configuring the management port and administrative user accounts
Enabling ConfigSync and high availability
Configuring the internal network
Configuring the external network
Configuring the network for high availability
Configuring a ConfigSync address
Configuring failover and mirroring addresses
Discovering a peer device

Licensing and provisioning the BIG-IP system

Using the Setup utility, you can activate the BIG-IP® license and provision BIG-IP software.

1. From a workstation attached to the network on which you configured a primary cluster IP address for the management interface, log in to the BIG-IP system by typing the following URL, where <primary_cluster_management_IP_address> is the address you configured as the floating management IP address for the primary blade in the cluster: https://<primary_cluster_management_IP_address>
2. At the login prompt, type the default user name `admin`, and password `admin`, and click **Log in**. The Setup utility screen opens.
3. Click **Next**.
4. Click **Activate**. The License screen opens.
5. In the **Base Registration Key** field, paste the registration key.
6. Click **Next** and follow the process for licensing and provisioning the system.

   Note: *When you perform the licensing task so that you can run the F5 cloud ADC, you can accept the default provisioning values.*

7. Click **Next**. This displays the screen for configuring general properties and user administration settings.

The BIG-IP system license is now activated, and the relevant BIG-IP modules are provisioned.

Configuring a device certificate

Import or verify the certificate for the BIG-IP device.

Do one of the following:

- Click **Import**, import a certificate, click **Import**, and then click **Next**.
- Verify the displayed information for the certificate and click **Next**.

Configuring the management port and administrative user accounts

Configure the management port, time zone, and the administrative user names and passwords.

1. On the screen for configuring general properties, for the **Management Port Configuration** setting, select **Manual** and specify the IP address, network mask, and default gateway.
2. In the **Host Name** field, type a fully-qualified domain name (FQDN) for the system.

   The FQDN can consist of letters, numbers, and/or the characters underscore (_), dash (-), or period (.).
3. For the **Host IP Address** setting, retain the default value **Use Management Port IP Address**.
4. From the **Time Zone** list, select a time zone.

   The time zone you select typically reflects the location of the F5® system.
5. For the **Root Account** setting, type and confirm a password for the `root` account.

   The `root` account provides console access only.
6. For the **Admin Account** setting, type and confirm a password.

Typing a password for the `admin` account causes the system to terminate the login session. When this happens, log in to the F5 Configuration utility again, using the new password. The system returns to the appropriate screen in the Setup utility.

7. For the **SSH Access** setting, select or clear the check box.
8. From the **SSH IP Allow** list, retain the default value of ***All Addresses**, or specify a range.
9. Click **Next**.
10. In the Standard Network Configuration area of the screen, click **Next**.
    This displays the screen for enabling configuration synchronization and high availability.

Enabling ConfigSync and high availability

When you perform this task, you set up config sync and connection mirroring, and you can specify the failover method (network, serial, or both).

1. For the **Config Sync** setting, select the **Display configuration synchronization options** check box.
   This causes an additional ConfigSync screen to be displayed later.
2. For the **High Availability** setting, select the **Display failover and mirroring options** check box.
   This displays the **Failover Method** list and causes additional failover screens to be displayed later.
3. From the **Failover Method** list, select **Network and serial cable**.

   If you have a VIPRION® system, select **Network**.
4. Click **Next**.
   This displays the screen for configuring the default VLAN **internal**.

Configuring the internal network

Specify self IP addresses and settings for VLAN **internal**, which is the default VLAN for the internal network.

1. Specify the **Self IP** setting for the internal network:
   a) In the **Address** field, type a self IP address.
   b) In the **Netmask** field, type a network mask for the self IP address.
   c) For the **Port Lockdown** setting, retain the default value.
2. Specify the **Floating IP** setting:
   a) In the **Address** field, type a floating IP address.

      This address should be distinct from the address you type for the **Self IP** setting.

      Important: *If the BIG-IP device you are configuring is accessed using Amazon Web Services and the device needs to failover to a device group peer, use the second, Secondary Private IP address for the floating IP address.*

   b) For the **Port Lockdown** setting, retain the default value.
3. For the **VLAN Tag ID** setting, retain the default value, **auto**.

   This is the recommended value.
4. For the **VLAN Interfaces** setting, click the interface **1.2** and, using the Move button, move the interface number from the **Available** list to the **Untagged** list.

5. Click **Next**.
   This completes the configuration of the internal self IP addresses and VLAN, and displays the screen for configuring the default VLAN **external**.

Configuring the external network

Specify self IP addresses and settings for VLAN `external`, which is the default VLAN for the external network.

1. Specify the **Self IP** setting for the external network:
   a) In the **Address** field, type a self IP address.
   b) In the **Netmask** field, type a network mask for the self IP address.
   c) For the **Port Lockdown** setting, retain the default value.
2. In the **Default Gateway** field, type the IP address that you want to use as the default gateway to VLAN **external**.
3. Specify the **Floating IP** setting:
   a) In the **Address** field, type a floating IP address.
      This address should be distinct from the address you type for the **Self IP** setting.

      Important: *If the BIG-IP device you are configuring is accessed using Amazon Web Services and the device needs to failover to a device group peer, use the second, Secondary Private IP address for the floating IP address.*

   b) For the **Port Lockdown** setting, retain the default value.
4. For the **VLAN Tag ID** setting, retain the default value, **auto**.
   This is the recommended value.
5. For the **VLAN Interfaces** setting, click the interface **1.2** and, using the Move button, move the interface number from the **Available** list to the **Untagged** list.
6. Click **Next**.
   This completes the configuration of the external self IP addresses and VLAN, and displays the screen for configuring the default VLAN **HA**.

Configuring the network for high availability

To configure a network for high availability, specify self IP addresses and settings for VLAN `HA`, which is the VLAN that the system will use for failover and connection mirroring.

1. For the **High Availability VLAN** setting, retain the default value, **Create VLAN HA**.
2. Specify the **Self IP** setting for VLAN **HA**:
   a) In the **Address** field, type a self IP address.
   b) In the **Netmask** field, type a network mask for the self IP address.
3. For the **VLAN Tag ID** setting, retain the default value, **auto**.
   This is the recommended value.

4. For the **VLAN Interfaces** setting, click an interface number, and using the Move button, move the interface number from the **Available** list to the **Untagged** list.
5. Click **Next**.
   This configures the self IP address and VLAN that the system will use for high availability and displays the default IP address that the system will use for configuration synchronization.

Configuring a ConfigSync address

Use this task to specify the address that you want the system to use for configuration synchronization.

1. From the **Local Address** list, select a self IP address.

   Do not select a management IP address.
2. Click **Next**.
   This displays the screen for configuring unicast and multicast failover addresses.

Configuring failover and mirroring addresses

Follow these steps to specify the local unicast and mirroring addresses that you want the BIG-IP® system to use for high availability. During the final step of running the Setup utility, the system exchanges these addresses with its trusted peer. If you are configuring a VIPRION® system, configure a multicast failover address as well.

1. Locate the Failover Unicast Configuration area of the screen.
2. Under Local Address, confirm that there are entries for the self IP address that is assigned to the **HA** and **internal** and VLANs and for the local management IP address for this device. If these entries are not absent, click the **Add** button to add the missing entries to the list of Failover Unicast Addresses.
   a) For the **Address** setting, select the self IP address for the VLAN you need to add (either `HA` or `internal`).
   b) In the **Port** field, type a port number or retain the default port number, `1026`.
   c) Either click **Repeat** to add additional addresses, or click **Finished**.
3. Click **Next**.
4. From the **Primary Local Mirror Address** list, retain the default value, which is the self IP address for VLAN `HA`.
5. From the **Secondary Local Mirror Address** list, select the address for VLAN `internal`.
6. Click **Finished**.

Discovering a peer device

You can use the Setup utility to discover a peer device for the purpose of exchanging failover and mirroring information.

1. Under **Standard Pair Configuration**, click **Next**.
2. If this is the first device of the pair that you are setting up, then under **Configure Peer Device**, click **Finished**.

   To activate device discovery, you must first run the Setup utility on the peer device.

3. If this is the second device of the pair that you are setting up:
   a) Under **Discover Configured Peer Device**, click **Next**.
   b) Under **Remote Device Credentials**, specify the `Management IP address`, `Administrator Username`, and `Administrator Password`.
   c) Click **Retrieve Device Information**.

4. Click **Finished**.

After the second device has discovered the first device, the two devices have a trust relationship and constitute a two-member device group. Also, each device in the pair contains a default traffic group named `Traffic-Group-1`. By default, this traffic group contains the floating IP addresses that you defined for VLANs `internal`and `external`.

Implementation result

To summarize, you now have the following BIG-IP® configuration on each device of the pair:

- A management port, management route, and administrative passwords defined.
- A VLAN named `internal`, with one static and one floating IP address.
- A VLAN named `external`, with one static and one floating IP address.
- A VLAN named `HA` with a static IP address.
- Configuration synchronization, failover, and mirroring enabled.
- Failover methods of serial cable and network (or network-only, for a VIPRION® platform.
- A designation as an authority device, where trust was established with the peer device.
- A Sync-Failover type of device group with two members defined.
- A default traffic group that floats to the peer device to process application traffic when this device becomes unavailable. This traffic group contains two floating self IP addresses for VLANs `internal` and `external`.
- One end of an iSession™ connection for WAN traffic optimization.

On either device in the device group, you can create additional configuration objects, such as virtual IP addresses and SNATs. The system automatically adds these objects to `Traffic-Group-1`.

Creating an Active-Active Configuration Using the Setup Utility

Overview: Creating a basic active-active configuration

This implementation describes how to use the Setup utility to configure two new BIG-IP® devices that function as an active-active pair. An *active-active* pair is a pair of BIG-IP devices configured so that both devices are actively processing traffic and are ready to take over one another if failover occurs. The two devices synchronize their configuration data to one another.

Note: *Access Policy Manager (APM) is not supported in an Active-Active configuration. APM is supported in an Active-Standby configuration with two BIG-IP systems only.*

Important: *The same version of BIG-IP system software must be running on all devices in the device group.*

Using this implementation, you begin by running the Setup utility on each device to configure its base network components. Base network components include a management port, administrative passwords, and default VLANs and their associated self IP addresses. You also use Setup to configure configuration synchronization and high availability.

You then use the BIG-IP® Configuration utility to:

- Establish trust between the two devices
- Create a Sync-Failover type of device group that contains two member devices
- Create a second traffic group
- Create two iApp™ application services

In this configuration, both devices actively process application traffic, each for a different application. One device processes its application traffic using the configuration objects associated with the default floating traffic group, `traffic-group-1`. By default, this traffic group contains the floating self IP addresses of the default VLANs. The other device processes its application traffic using a second traffic group that you create.

If one of the devices becomes unavailable for any reason, the other device automatically begins processing traffic for the unavailable peer, while continuing to process the traffic for its own application.

This illustration shows an example of the device group that this implementation creates, named `Device Group A`. This device group contains two BIG-IP devices, `Device 1` and `Device 2`.

The configuration shows two traffic groups, `traffic-group-1` and `traffic-group-2`, each containing failover objects. For `traffic-group-1`, `Device 1` is the default device. For `traffic-group-2`, `Device 2` is the default device. If `Device 1` becomes unavailable, `traffic-group-1` floats to `Device 2`. If `Device 2` becomes unavailable, `traffic-group-2` floats to `Device 1`.

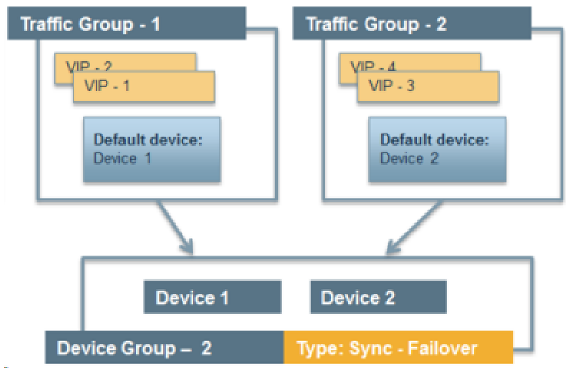


Figure 2: Device group with active-active configuration

By implementing this configuration, you ensure that:

- Each device has base network components configured.
- Any objects on a BIG-IP device that you configure for synchronization remain synchronized between the two devices.
- Failover capability and connection mirroring are enabled on each device.

Important: *For active-active configurations, you must enable network failover instead of hard-wired serial failover.*

Task summary

The BIG-IP® configuration process begins with running the Setup utility on each of the two BIG-IP devices. Once you have completed that task, you can log into either of the BIG-IP devices and perform all of the remaining tasks, on that device only. This results in both BIG-IP devices being configured properly for an active-active implementation.

Important: *After using the Setup utility to create a redundant system configuration, you can re-enter the utility at any time to adjust the configuration. Simply click the F5 logo in the upper-left corner of the BIG-IP Configuration utility, and on the Welcome screen, click* ***Run the Setup Utility****. Then page through the utility to find the appropriate screens.*

Licensing and provisioning the BIG-IP system
Configuring a device certificate
Configuring the management port and administrative user accounts
Enabling ConfigSync and high availability
Configuring the internal network
Configuring the external network
Configuring the network for high availability
Configuring a ConfigSync address
Configuring failover and mirroring addresses
Establishing device trust
Creating a Sync-Failover device group
Creating an iApp application for the local device
Creating a traffic group for a remote device
Creating an iApp application for a remote device

Forcing a traffic group to a standby state
Syncing the BIG-IP configuration to the device group

Licensing and provisioning the BIG-IP system

Using the Setup utility, you can activate the BIG-IP® license and provision BIG-IP software.

1. From a workstation attached to the network on which you configured a primary cluster IP address for the management interface, log in to the BIG-IP system by typing the following URL, where `<primary_cluster_management_IP_address>` is the address you configured as the floating management IP address for the primary blade in the cluster:
   `https://<primary_cluster_management_IP_address>`
2. At the login prompt, type the default user name `admin`, and password `admin`, and click **Log in**.
   The Setup utility screen opens.
3. Click **Next**.
4. Click **Activate**.
   The License screen opens.
5. In the **Base Registration Key** field, paste the registration key.
6. Click **Next** and follow the process for licensing and provisioning the system.

   Note: *When you perform the licensing task so that you can run the F5 cloud ADC, you can accept the default provisioning values.*

   ---

7. Click **Next**.
   This displays the screen for configuring general properties and user administration settings.

The BIG-IP system license is now activated, and the relevant BIG-IP modules are provisioned.

Configuring a device certificate

Import or verify the certificate for the BIG-IP device.

Do one of the following:

- Click **Import**, import a certificate, click **Import**, and then click **Next**.
- Verify the displayed information for the certificate and click **Next**.

Configuring the management port and administrative user accounts

Configure the management port, time zone, and the administrative user names and passwords.

1. On the screen for configuring general properties, for the **Management Port Configuration** setting, select **Manual** and specify the IP address, network mask, and default gateway.
2. In the **Host Name** field, type a fully-qualified domain name (FQDN) for the system.

   The FQDN can consist of letters, numbers, and/or the characters underscore (_), dash (-), or period (.).
3. For the **Host IP Address** setting, retain the default value **Use Management Port IP Address**.
4. From the **Time Zone** list, select a time zone.

The time zone you select typically reflects the location of the F5® system.

5. For the **Root Account** setting, type and confirm a password for the `root` account.

   The `root` account provides console access only.

6. For the **Admin Account** setting, type and confirm a password.

   Typing a password for the `admin` account causes the system to terminate the login session. When this happens, log in to the F5 Configuration utility again, using the new password. The system returns to the appropriate screen in the Setup utility.

7. For the **SSH Access** setting, select or clear the check box.
8. From the **SSH IP Allow** list, retain the default value of ***All Addresses**, or specify a range.
9. Click **Next**.
10. In the Standard Network Configuration area of the screen, click **Next**.
    This displays the screen for enabling configuration synchronization and high availability.

Enabling ConfigSync and high availability

When you perform this task, you set up config sync and connection mirroring, and you can specify the failover method (network, serial, or both).

1. For the **Config Sync** setting, select the **Display configuration synchronization options** check box.
   This causes an additional ConfigSync screen to be displayed later.
2. For the **High Availability** setting, select the **Display failover and mirroring options** check box.
   This displays the **Failover Method** list and causes additional failover screens to be displayed later.
3. From the **Failover Method** list, select **Network and serial cable**.

   If you have a VIPRION® system, select **Network**.

4. Click **Next**.
   This displays the screen for configuring the default VLAN **internal**.

Configuring the internal network

Specify self IP addresses and settings for VLAN **internal**, which is the default VLAN for the internal network.

1. Specify the **Self IP** setting for the internal network:
   a) In the **Address** field, type a self IP address.
   b) In the **Netmask** field, type a network mask for the self IP address.
   c) For the **Port Lockdown** setting, retain the default value.

2. Specify the **Floating IP** setting:
   a) In the **Address** field, type a floating IP address.

      This address should be distinct from the address you type for the **Self IP** setting.

      Important: *If the BIG-IP device you are configuring is accessed using Amazon Web Services and the device needs to failover to a device group peer, use the second, Secondary Private IP address for the floating IP address.*

      ---

   b) For the **Port Lockdown** setting, retain the default value.

3. For the **VLAN Tag ID** setting, retain the default value, **auto**.
   This is the recommended value.
4. For the **VLAN Interfaces** setting, click the interface **1.2** and, using the Move button, move the interface number from the **Available** list to the **Untagged** list.
5. Click **Next**.
   This completes the configuration of the internal self IP addresses and VLAN, and displays the screen for configuring the default VLAN **external**.

Configuring the external network

Specify self IP addresses and settings for VLAN `external`, which is the default VLAN for the external network.

1. Specify the **Self IP** setting for the external network:
   a) In the **Address** field, type a self IP address.
   b) In the **Netmask** field, type a network mask for the self IP address.
   c) For the **Port Lockdown** setting, retain the default value.
2. In the **Default Gateway** field, type the IP address that you want to use as the default gateway to VLAN **external**.
3. Specify the **Floating IP** setting:
   a) In the **Address** field, type a floating IP address.
   This address should be distinct from the address you type for the **Self IP** setting.

   Important: *If the BIG-IP device you are configuring is accessed using Amazon Web Services and the device needs to failover to a device group peer, use the second, Secondary Private IP address for the floating IP address.*

   b) For the **Port Lockdown** setting, retain the default value.
4. For the **VLAN Tag ID** setting, retain the default value, **auto**.
   This is the recommended value.
5. For the **VLAN Interfaces** setting, click the interface **1.2** and, using the Move button, move the interface number from the **Available** list to the **Untagged** list.
6. Click **Next**.
   This completes the configuration of the external self IP addresses and VLAN, and displays the screen for configuring the default VLAN **HA**.

Configuring the network for high availability

To configure a network for high availability, specify self IP addresses and settings for VLAN `HA`, which is the VLAN that the system will use for failover and connection mirroring.

1. For the **High Availability VLAN** setting, retain the default value, **Create VLAN HA**.
2. Specify the **Self IP** setting for VLAN **HA**:
   a) In the **Address** field, type a self IP address.
   b) In the **Netmask** field, type a network mask for the self IP address.

3. For the **VLAN Tag ID** setting, retain the default value, **auto**.

   This is the recommended value.
4. For the **VLAN Interfaces** setting, click an interface number, and using the Move button, move the interface number from the **Available** list to the **Untagged** list.
5. Click **Next**.
   This configures the self IP address and VLAN that the system will use for high availability and displays the default IP address that the system will use for configuration synchronization.

Configuring a ConfigSync address

Use this task to specify the address that you want the system to use for configuration synchronization.

1. From the **Local Address** list, select a self IP address.

   Do not select a management IP address.
2. Click **Next**.
   This displays the screen for configuring unicast and multicast failover addresses.

Configuring failover and mirroring addresses

Follow these task steps to specify the unicast IP addresses of the local device that you want the system to use for failover. Typically, you specify the self IP address for the local VLAN `HA`, as well as the IP address for the management port of the local device. If you are configuring a VIPRION® system, configure a multicast failover address as well.

Important: *When configuring failover and mirroring IP addresses, you select addresses of the local device only. Later, during the process of device discovery, the two devices in the device group discover each other's addresses.*

---

1. Locate the Failover Unicast Configuration area of the screen.
2. Under Local Address, confirm that there are entries for the self IP address that is assigned to the **HA** and **internal** and VLANs and for the local management IP address for this device. If these entries are not absent, click the **Add** button to add the missing entries to the list of Failover Unicast Addresses.
   a) For the **Address** setting, select the self IP address for the VLAN you need to add (either `HA` or `internal`).
   b) In the **Port** field, type a port number or retain the default port number, `1026`.
   c) Either click **Repeat** to add additional addresses, or click **Finished**.
3. Click **Next**.
4. From the **Primary Local Mirror Address** list, retain the default value, which is the self IP address for VLAN `HA`.
5. From the **Secondary Local Mirror Address** list, select the address for VLAN `internal`.
6. Click **Finished**.
   This causes you to leave the Setup utility.

Establishing device trust

Before you begin this task, verify that:

- Each BIG-IP® device that is to be part of the local trust domain has a device certificate installed on it.
- The local device is designated as a certificate signing authority.

You perform this task to establish trust among devices on one or more network segments. Devices that trust each other constitute the *local trust domain*. A device must be a member of the local trust domain prior to joining a device group.

By default, the BIG-IP software includes a local trust domain with one member, which is the local device. You can choose any one of the BIG-IP devices slated for a device group and log into that device to add other devices to the local trust domain. For example, devices `A`, `B`, and `C` each initially shows only itself as a member of the local trust domain. To configure the local trust domain to include all three devices, you can simply log into device `A` and add devices `B` and `C` to the local trust domain. Note that there is no need to repeat this process on devices `B` and `C`.

1. On the Main tab, click **Device Management** > **Device Trust**, and then either **Peer List** or **Subordinate List**.
2. Click **Add**.
3. Type a device IP address, administrator user name, and administrator password for the remote BIG-IP® device with which you want to establish trust. The IP address you specify depends on the type of BIG-IP device:
   - If the BIG-IP device is a non-VIPRION device, type the management IP address for the device.
   - If the BIG-IP device is a VIPRION device that is not licensed and provisioned for vCMP, type the primary cluster management IP address for the cluster.
   - If the BIG-IP device is a VIPRION device that is licensed and provisioned for vCMP, type the cluster management IP address for the guest.
   - If the BIG-IP device is an Amazon Web Services EC2 device, type one of the Private IP addresses created for this EC2 instance.
4. Click **Retrieve Device Information**.
5. Verify that the certificate of the remote device is correct.
6. Verify that the name of the remote device is correct.
7. Verify that the management IP address and name of the remote device are correct.
8. Click **Finished**.

The device you added is now a member of the local trust domain.

Repeat this task for each device that you want to add to the local trust domain.

Creating a Sync-Failover device group

This task establishes failover capability between two BIG-IP® devices. If the active device in a Sync-Failover device group becomes unavailable, the configuration objects fail over to another member of the device group and traffic processing is unaffected. You can perform this task on any authority device within the local trust domain.

1. On the Main tab, click **Device Management** > **Device Groups**.
2. On the Device Groups list screen, click **Create**.
   The New Device Group screen opens.

3. Type a name for the device group, select the device group type **Sync-Failover**, and type a description for the device group.
4. In the Configuration area of the screen, select a host name from the **Available** list for each BIG-IP device that you want to include in the device group, including the local device. Use the Move button to move the host name to the **Includes** list.

   The **Available** list shows any devices that are members of the device's local trust domain but not currently members of a Sync-Failover device group. A device can be a member of one Sync-Failover group only.
5. For the **Network Failover** setting, select or clear the check box:
   - Select the check box if you want device group members to handle failover communications by way of network connectivity. This choice is required for active-active configurations.
   - Clear the check box if you want device group members to handle failover communications by way of serial cable (hard-wired) connectivity.

   For active-active configurations, you must select network failover, as opposed to serial-cable (hard-wired) connectivity.
6. Click **Finished**.

You now have a Sync-Failover device group containing two BIG-IP devices as members.

Creating an iApp application for the local device

Use this procedure to create a set of related configuration objects on the system (that is, an application).

1. On the Main tab, click **iApp** > **Application Services**.
2. Click **Create**.
3. In the **Name** field, type the name for your application service.
4. From the **Template** list, select a template.
5. From the Template Selection list, select **Advanced**.
   This causes additional settings to appear.
6. For the `Configure Sync and/or Failover for this application?` setting, select **Yes**.
7. For the `Traffic Group` setting, ensure that the **Inherit traffic group from current partition / path** field and **traffic-group-1** are selected.
8. Configure remaining settings as needed.
9. At the bottom of the screen click **Finished** to save your changes.

You now have an iApp application service, which is associated with the traffic group assigned to the **root** folder, `traffic-group-1`.

Creating a traffic group for a remote device

Prerequisite: If you intend to specify a MAC masquerade address when creating a traffic group, you must first create the address, using an industry-standard method for creating a locally-administered MAC address.

Perform this procedure to create a traffic group to run on the remote BIG-IP ®device. You create this traffic group on the local device. Later, you move the traffic group to the remote device by forcing this traffic group on the local device to a standby state.

1. On the Main tab, click **Device Management** > **Traffic Groups**.

2. On the lower half of the screen, verify that the list shows the default floating traffic group (traffic-group-1) for the local device.
3. On the Traffic Group List screen, click **Create**.
4. Type the name `traffic-group-2` for the new traffic group.
5. Type a description of the new traffic group.
6. Click **Next**.
7. In the **MAC Masquerade Address** field, type a MAC masquerade address.

   When you specify a MAC masquerade address, you reduce the risk of dropped connections when failover occurs. This setting is optional.
8. Click **Next**.
9. Select or clear the check box for the **Auto Failback** option:
   - Select the check box to cause the traffic group, after failover, to fail over again to the first device in the traffic group's ordered list when that device (and only that device) is available.
   - Clear the check box to cause the traffic group, after failover, to remain active on its current device until failover occurs again.
10. Click **Next**.
11. Confirm that the displayed traffic group settings are correct.
12. Click **Finished**.

You now have a floating traffic group for which the default device is the peer device.

Creating an iApp application for a remote device

Use this procedure when you want to create an application to run on a remote device and associate it with the traffic group named `traffic-group-2` that you previously created.

1. On the Main tab, click **iApp** > **Application Services**.
2. Click **Create**.
3. From the **Template** list, select a template.
4. From the Template Selection list, select **Advanced**.
   This causes additional settings to appear.
5. In the **Name** field, type the name for your application service.
6. For the `Configure Sync and/or Failover for this application?` setting, select **Yes**.
7. For the `Traffic Group` setting, clear the **Inherit traffic group from current partition / path** field and from the list, select **traffic-group-2**.
8. Configure remaining settings as needed.
9. At the bottom of the screen click **Finished** to save your changes.

You now have an iApp application associated with `traffic-group-2`.

Forcing a traffic group to a standby state

You perform this task when you want the selected traffic group on the local device to fail over to another device (that is, switch to a `Standby` state). Users typically perform this task when no automated method is configured for a traffic group, such as auto-failback or an HA group. By forcing the traffic group into a

`Standby` state, the traffic group becomes active on another device in the device group. For device groups with more than two members, you can choose the specific device to which the traffic group fails over.

1. Log in to the device on which the traffic group is currently active.
2. On the Main tab, click **Device Management** > **Traffic Groups**.
3. In the Name column, locate the name of the traffic group that you want to run on the peer device.
4. Select the check box to the left of the traffic group name.
   If the check box is unavailable, the traffic group is not active on the device to which you are currently logged in. Perform this task on the device on which the traffic group is active.
5. Click **Force to Standby**.
   This displays target device options.
6. Choose one of these actions:
   - If the device group has two members only, click **Force to Standby**. This displays the list of traffic groups for the device group and causes the local device to appear in the Next Active Device column.
   - If the device group has more than two members, then from the **Target Device** list, select a value and click **Force to Standby**.

The selected traffic group is now in a standby state on the local device and active on another device in the device group.

Syncing the BIG-IP configuration to the device group

Before you sync the configuration, verify that the devices targeted for config sync are members of a device group and that device trust is established.

This task synchronizes the BIG-IP® configuration data from the local device to the devices in the device group. This synchronization ensures that devices in the device group operate properly. When synchronizing self IP addresses, the BIG-IP system synchronizes floating self IP addresses only.

Important: *You perform this task on either of the two devices, but not both.*

---

1. On the Main tab, click **Device Management** > **Overview**.
2. In the Device Groups area of the screen, in the Name column, select the name of the relevant device group.
   The screen expands to show a summary and details of the sync status of the selected device group, as well as a list of the individual devices within the device group.
3. In the Devices area of the screen, in the Sync Status column, select the device that shows a sync status of `Changes Pending`.
4. In the Sync Options area of the screen, select **Sync Device to Group**.
5. Click **Sync**.
   The BIG-IP system syncs the configuration data of the selected device in the Device area of the screen to the other members of the device group.

Except for non-floating self IP addresses, the entire set of BIG-IP configuration data is replicated on each device in the device group.

Implementation Results

To summarize, you now have the following BIG-IP® configuration on each device of the pair:

- A management port, management route, and administrative passwords defined
- A VLAN named `internal`, with one static and one floating IP address
- A VLAN named `external`, with one static and one floating IP address
- A VLAN named `HA` with a static IP address
- Configuration synchronization, failover, and mirroring enabled
- Failover methods of serial cable and network
- Local IP addresses defined for failover and connection mirroring
- A designation as an authority device, where trust is established with the peer device
- A Sync-Failover type of device group with two members
- The default traffic group named `traffic-group-1` with `Device 1` as the default device
- An iApp application associated with `traffic-group-1`
- A traffic group named `traffic-group-2` with `Device 2` as the default device
- An iApp application associated with `traffic-group-2`

Creating an Active-Standby Configuration using the Configuration Utility

Overview: Creating an active-standby DSC configuration

The most common TMOS® device service clustering (DSC™) implementation is an *active-standby* configuration, where a single traffic group is active on one of the devices in the device group and is in a standby state on a peer device. If failover occurs, the standby traffic group on the peer device becomes active and begins processing the application traffic.

To implement this DSC implementation, you can create a Sync-Failover device group. A Sync-Failover device group with two or more members and one traffic group provides configuration synchronization and device failover, and optionally, connection mirroring.

If the device with the active traffic group goes offline, the traffic group becomes active on a peer device, and application processing is handled by that device.

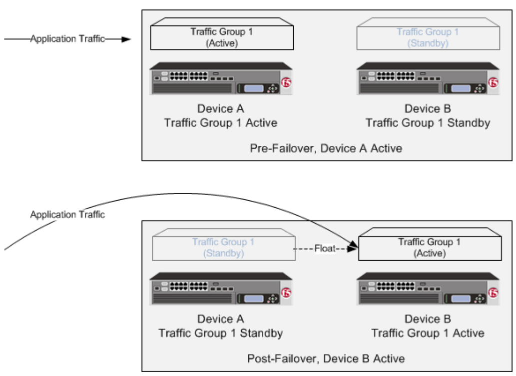


Figure 3: A two-member Sync-Failover device group for an active-standby configuration

About DSC configuration on a VIPRION system

The way you configure device service clustering (DSC™) (also known as redundancy) on a VIPRION® system varies depending on whether the system is provisioned to run the vCMP® feature.

For non-vCMP systems

For a device group that consists of VIPRION systems that are not licensed and provisioned for vCMP, each VIPRION cluster constitutes an individual device group member. The following table describes the IP addresses that you must specify when configuring redundancy.

Table 1: Required IP addresses for DSC configuration on a non-vCMP system

Feature	IP addresses required
Device trust	The primary floating management IP address for the VIPRION cluster.
ConfigSync	The unicast non-floating self IP address assigned to VLAN `internal`.

Feature	IP addresses required
Failover	• Recommended: The unicast non-floating self IP address that you assigned to an internal VLAN (preferably VLAN `HA`), as well as a multicast address. • Alternative: All unicast management IP addresses that correspond to the slots in the VIPRION cluster.
Connection mirroring	For the primary address, the non-floating self IP address that you assigned to VLAN `HA`. The secondary address is not required, but you can specify any non-floating self IP address for an internal VLAN..

For vCMP systems

On a vCMP system, the devices in a device group are virtual devices, known as *vCMP guests*. You configure device trust, config sync, failover, and mirroring to occur between equivalent vCMP guests in separate chassis.

For example, if you have a pair of VIPRION systems running vCMP, and each system has three vCMP guests, you can create a separate device group for each pair of equivalent guests. Table 4.2 shows an example.

Table 2: Sample device groups for two VIPRION systems with vCMP

Device groups for vCMP	Device group members
`Device-Group-A`	• `Guest1` on chassis1 • `Guest1` on chassis2
`Device-Group-B`	• `Guest2` on chassis1 • `Guest2` on chassis2
`Device-Group-C`	• `Guest3` on chassis1 • `Guest3` on chassis2

By isolating guests into separate device groups, you ensure that each guest synchronizes and fails over to its equivalent guest. The following table describes the IP addresses that you must specify when configuring redundancy:

Table 3: Required IP addresses for DSC configuration on a VIPRION system with vCMP

Feature	IP addresses required
Device trust	The cluster management IP address of the guest.
ConfigSync	The non-floating self IP address on the guest that is associated with VLAN `internal` on the host.
Failover	• Recommended: The unicast non-floating self IP address on the guest that is associated with an internal VLAN on the host (preferably VLAN `HA`), as well as a multicast address. • Alternative: The unicast management IP addresses for all slots configured for the guest.
Connection mirroring	For the primary address, the non-floating self IP address on the guest that is associated with VLAN `internal` on the host. The secondary address is not required, but you can specify any non-floating self IP address on the guest that is associated with an internal VLAN on the host.

DSC prerequisite worksheet

Before you set up device service clustering (DSC™), you must configure these BIG-IP® components on each device that you intend to include in the device group.

Table 4: DSC deployment worksheet

Configuration component	Considerations
Hardware, licensing, and provisioning	Devices in a device group must match with respect to product licensing and module provisioning. Heterogeneous hardware platforms within a device group are supported.
BIG-IP software version	Each device must be running BIG-IP version 11.x. This ensures successful configuration synchronization.
Management IP addresses	Each device must have a management IP address, a network mask, and a management route defined.
FQDN	Each device must have a fully-qualified domain name (FQDN) as its host name.
User name and password	Each device must have a user name and password defined on it that you will use when logging in to the BIG-IP Configuration utility.
`root` folder properties	The platform properties for the `root` folder must be set correctly (`Sync-Failover` and `traffic-group-1`).
VLANs	You must create these VLANs on each device, if you have not already done so: • A VLAN for the internal network, named `internal` • A VLAN for the external network, named `external` • A VLAN for failover communications, named `HA`
Self IP addresses	You must create these self IP addresses on each device, if you have not already done so: • Two self IP addresses (floating and non-floating) on the same subnet for VLAN `internal`. • Two self IP addresses (floating and non-floating) on the same subnet for VLAN `external`. • A non-floating self IP address on the internal subnet for VLAN `HA`. ***Note:*** *When you create floating self IP addresses, the BIG-IP system automatically adds them to the default floating traffic group,* `traffic-group-1`*. To add a self IP address to a different traffic group, you must modify the value of the self IP address* ***Traffic Group*** *property.* ***Important:*** *If the BIG-IP device you are configuring is accessed using Amazon Web Services, then the IP address you specify must be the floating IP address for high availability fast failover that you configured for the EC2 instance.*
Port lockdown	For self IP addresses that you create on each device, you should verify that the **Port Lockdown** setting is set to **Allow All**, **All Default**, or **Allow Custom**. Do not specify **None**.

Configuration component	Considerations
Application-related objects	You must create any virtual IP addresses and optionally, SNAT translation addresses, as part of the local traffic configuration. You must also configure any iApp™ application services if they are required for your application. When you create these addresses or services, the objects automatically become members of the default traffic group, `traffic-group-1`.
Time synchronization	The times set by the NTP service on all devices must be synchronized. This is a requirement for configuration synchronization to operate successfully.
Device certificates	Verify that each device includes an x509 device certificate. Devices with device certificates can authenticate and therefore trust one another, which is a prerequisite for device-to-device communication and data exchange.

Task summary

Use the tasks in this implementation to create a two-member device group, with one active traffic group, that syncs the BIG-IP® configuration to the peer device and provides failover capability if the peer device goes offline. Note that on a vCMP® system, the devices in a specific device group are vCMP guests, one per chassis.

Important: *When you use this implementation, F5 Networks recommends that you synchronize the BIG-IP configuration twice, once after you create the device group, and again after you specify the IP addresses for failover.*

Task list

Specifying an IP address for config sync
Specifying an IP address for connection mirroring
Specifying the HA capacity of a device
Establishing device trust
Creating a Sync-Failover device group
Syncing the BIG-IP configuration to the device group
Specifying IP addresses for failover communication
Syncing the BIG-IP configuration to the device group

Specifying an IP address for config sync

Before configuring the config sync address, verify that all devices in the device group are running the same version of BIG-IP® system software.

You perform this task to specify the IP address on the local device that other devices in the device group will use to synchronize their configuration objects to the local device.

Note: *You must perform this task locally on each device in the device group.*

1. Confirm that you are logged in to the actual device you want to configure.
2. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.

3. In the Name column, click the name of the device to which you are currently logged in.
4. From the Device Connectivity menu, choose ConfigSync.
5. For the **Local Address** setting, retain the displayed IP address or select another address from the list.

   F5 Networks recommends that you use the default value, which is the self IP address for VLAN `internal`. This address must be a non-floating self IP address and not a management IP address.

   Important: *If the BIG-IP device you are configuring is accessed using Amazon Web Services, then the internal self IP address that you specify must be the internal private IP addresses that you configured for this EC2 instance as the* ***Local Address****.*

   ---

6. Click **Update**.

After performing this task, the other devices in the device group can sync their configurations to the local device.

Specifying an IP address for connection mirroring

You can specify the local self IP address that you want other devices in a device group to use when mirroring their connections to this device. Connection mirroring ensures that in-process connections for an active traffic group are not dropped when failover occurs. You typically perform this task when you initially set up device service clustering (DSC®).

Note: *You must perform this task locally on each device in the device group.*

---

1. Confirm that you are logged in to the actual device you want to configure.
2. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
3. In the Name column, click the name of the device to which you are currently logged in.
4. From the Device Connectivity menu, choose Mirroring.
5. For the **Primary Local Mirror Address** setting, retain the displayed IP address or select another address from the list.

   The recommended IP address is the self IP address for either VLAN `HA` or VLAN `internal`.

   Important: *If the BIG-IP device you are configuring is accessed using Amazon Web Services, then the self IP address you specify must be one of the private IP addresses that you configured for this EC2 instance as the* ***Primary Local Mirror Address****.*

   ---

6. For the **Secondary Local Mirror Address** setting, retain the default value of **None**, or select an address from the list.

   This setting is optional. The system uses the selected IP address in the event that the primary mirroring address becomes unavailable.

7. Click **Update**.

In addition to specifying an IP address for mirroring, you must also enable connection mirroring on the relevant virtual servers on this device.

Specifying the HA capacity of a device

Before you perform this task, verify that this device is a member of a device group and that the device group contains three or more devices.

You perform this task when you have more than one type of hardware platform in a device group and you want to configure load-aware failover. *Load-aware failover* ensures that the BIG-IP® system can intelligently select the next-active device for each active traffic group in the device group when failover occurs. As part of configuring load-aware failover, you define an HA capacity to establish the amount of computing resource that the device provides relative to other devices in the device group.

Note: *If all devices in the device group are the same hardware platform, you can skip this task.*

---

1. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
2. In the Name column, click the name of the device for which you want to view properties.
   This displays a table of properties for the device.
3. In the **HA Capacity** field, type a relative numeric value.

   You need to configure this setting only when you have varying types of hardware platforms in a device group and you want to configure load-aware failover. The value you specify represents the relative capacity of the device to process application traffic compared to the other devices in the device group.

   Important: *If you configure this setting, you must configure the setting on every device in the device group.*

   If this device has half the capacity of a second device and a third of the capacity of a third device in the device group, you can specify a value of `100` for this device, `200` for the second device, and `300` for the third device.
   When choosing the next active device for a traffic group, the system considers the capacity that you specified for this device.
4. Click **Update**.

After you perform this task, the BIG-IP system uses the **HA Capacity** value to calculate the current utilization of the local device, to determine the next-active device for failover of other traffic groups in the device group.

Establishing device trust

Before you begin this task, verify that:

- Each BIG-IP® device that is to be part of the local trust domain has a device certificate installed on it.
- The local device is designated as a certificate signing authority.

You perform this task to establish trust among devices on one or more network segments. Devices that trust each other constitute the *local trust domain*. A device must be a member of the local trust domain prior to joining a device group.

By default, the BIG-IP software includes a local trust domain with one member, which is the local device. You can choose any one of the BIG-IP devices slated for a device group and log into that device to add other devices to the local trust domain. For example, devices `A`, `B`, and `C` each initially shows only itself as a member of the local trust domain. To configure the local trust domain to include all three devices, you can simply log into device `A` and add devices `B` and `C` to the local trust domain. Note that there is no need to repeat this process on devices `B` and `C`.

1. On the Main tab, click **Device Management** > **Device Trust**, and then either **Peer List** or **Subordinate List**.
2. Click **Add**.
3. Type a device IP address, administrator user name, and administrator password for the remote BIG-IP® device with which you want to establish trust. The IP address you specify depends on the type of BIG-IP device:
   - If the BIG-IP device is a non-VIPRION device, type the management IP address for the device.
   - If the BIG-IP device is a VIPRION device that is not licensed and provisioned for vCMP, type the primary cluster management IP address for the cluster.
   - If the BIG-IP device is a VIPRION device that is licensed and provisioned for vCMP, type the cluster management IP address for the guest.
   - If the BIG-IP device is an Amazon Web Services EC2 device, type one of the Private IP addresses created for this EC2 instance.
4. Click **Retrieve Device Information**.
5. Verify that the certificate of the remote device is correct.
6. Verify that the name of the remote device is correct.
7. Verify that the management IP address and name of the remote device are correct.
8. Click **Finished**.

The device you added is now a member of the local trust domain.

Repeat this task for each device that you want to add to the local trust domain.

Creating a Sync-Failover device group

This task establishes failover capability between two or more BIG-IP® devices. If an active device in a Sync-Failover device group becomes unavailable, the configuration objects fail over to another member of the device group and traffic processing is unaffected. You perform this task on any one of the authority devices within the local trust domain.

Repeat this task for each Sync-Failover device group that you want to create for your network configuration.

1. On the Main tab, click **Device Management** > **Device Groups**.
2. On the Device Groups list screen, click **Create**.
   The New Device Group screen opens.
3. Type a name for the device group, select the device group type **Sync-Failover**, and type a description for the device group.
4. From the **Configuration** list, select **Advanced**.
5. In the Configuration area of the screen, select a host name from the **Available** list for each BIG-IP device that you want to include in the device group, including the local device. Use the Move button to move the host name to the **Includes** list.

   The **Available** list shows any devices that are members of the device's local trust domain but not currently members of a Sync-Failover device group. A device can be a member of one Sync-Failover group only.
6. For the **Network Failover** setting, select or clear the check box:
   - Select the check box if you want device group members to handle failover communications by way of network connectivity. This choice is required for active-active configurations.
   - Clear the check box if you want device group members to handle failover communications by way of serial cable (hard-wired) connectivity.

For active-active configurations, you must select network failover, as opposed to serial-cable (hard-wired) connectivity.

7. For the **Automatic Sync** setting, select or clear the check box:
   - Select the check box when you want the BIG-IP system to automatically sync the BIG-IP configuration data whenever a config sync operation is required. In this case, the BIG-IP system syncs the configuration data whenever the data changes on any device in the device group.
   - Clear the check box when you want to manually initiate each config sync operation. In this case, F5 networks recommends that you perform a config sync operation whenever configuration data changes on one of the devices in the device group.
8. For the **Full Sync** setting, select or clear the check box:
   - Select the check box when you want all sync operations to be full syncs. In this case, the BIG-IP system syncs the entire set of BIG-IP configuration data whenever a config sync operation is required.
   - Clear the check box when you want all sync operations to be incremental (the default setting). In this case, the BIG-IP system syncs only the changes that are more recent than those on the target device. When you select this option, the BIG-IP system compares the configuration data on each target device with the configuration data on the source device and then syncs the delta of each target-source pair.

   If you enable incremental synchronization, the BIG-IP system might occasionally perform a full sync for internal reasons. This is a rare occurrence and no user intervention is required.
9. In the **Maximum Incremental Sync Size (KB)** field, retain the default value of `1024`, or type a different value.

   This value specifies the total size of configuration changes that can reside in the incremental sync cache. If the total size of the configuration changes in the cache exceeds the specified value, the BIG-IP system performs a full sync whenever the next config sync operation occurs.
10. Click **Finished**.

You now have a Sync-Failover type of device group containing BIG-IP devices as members.

Syncing the BIG-IP configuration to the device group

Before you sync the configuration, verify that the devices targeted for config sync are members of a device group and that device trust is established.

This task synchronizes the BIG-IP® configuration data from the local device to the devices in the device group. This synchronization ensures that devices in the device group operate properly. When synchronizing self IP addresses, the BIG-IP system synchronizes floating self IP addresses only.

Important: *You perform this task on either of the two devices, but not both.*

---

1. On the Main tab, click **Device Management** > **Overview**.
2. In the Device Groups area of the screen, in the Name column, select the name of the relevant device group.
   The screen expands to show a summary and details of the sync status of the selected device group, as well as a list of the individual devices within the device group.
3. In the Devices area of the screen, in the Sync Status column, select the device that shows a sync status of `Changes Pending`.
4. In the Sync Options area of the screen, select **Sync Device to Group**.
5. Click **Sync**.

The BIG-IP system syncs the configuration data of the selected device in the Device area of the screen to the other members of the device group.

Except for non-floating self IP addresses, the entire set of BIG-IP configuration data is replicated on each device in the device group.

Specifying IP addresses for failover communication

You typically perform this task during initial Device Service Clustering (DSC®) configuration, to specify the local IP addresses that you want other devices in the device group to use for continuous health-assessment communication with the local device or guest. You must perform this task locally on each device in the device group.

Important: *If the system is running vCMP, you must log in to each guest to perform this task.*

Note: *The IP addresses that you specify must belong to route domain* `0`.

---

1. Confirm that you are logged in to the actual device you want to configure.
2. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
3. In the Name column, click the name of the device to which you are currently logged in.
4. From the Device Connectivity menu, choose Failover.
5. For the Failover Unicast Configuration settings, click **Add** for each IP address on this device that other devices in the device group can use to exchange failover messages with this device. The unicast IP addresses you specify depend on the type of device:

Platform	Action
Appliance without vCMP	Type a static self IP address associated with an internal VLAN (preferably VLAN `HA`) and the static management IP address currently assigned to the device.
Appliance with vCMP	Type a static self IP address associated with an internal VLAN (preferably VLAN `HA`) and the unique management IP address currently assigned to the guest.
VIPRION without vCMP®	Type a static self IP address associated with an internal VLAN (preferably VLAN `HA`), and, if you choose to specify unicast addresses only (and not a multicast address), you must also type the existing, static management IP addresses that you previously configured for all slots in the cluster. If you choose to specify one or more unicast addresses and a multicast address, then you do not need to specify the existing, per-slot static management IP addresses when configuring addresses for failover communication.
VIPRION with vCMP	Type a self IP address that is defined on the guest and associated with an internal VLAN on the host (preferably VLAN `HA`), and, if you choose to specify unicast failover addresses only (and not a a multicast address), you must also type the existing, virtual static management IP addresses that you previously configured for all slots in the guest's virtual cluster. If you choose to specify one or more unicast addresses and a multicast address, you do not need to specify the existing, per-slot static and virtual management IP addresses when configuring addresses for failover communication.

Important: *Failover addresses should always be static, not floating, IP addresses.*

---

6. To enable the use of a failover multicast address on a VIPRION® platform (recommended), then for the **Use Failover Multicast Address** setting, select the **Enabled** check box.
7. If you enabled **Use Failover Multicast Address**, either accept the default **Address** and **Port** values, or specify values appropriate for the device.

   If you revise the default **Address** and **Port** values, but then decide to revert to the default values, click **Reset Defaults**.
8. Click **Update**.

After you perform this task, other devices in the device group can send failover messages to the local device using the specified IP addresses.

Syncing the BIG-IP configuration to the device group

Before you sync the configuration, verify that the devices targeted for config sync are members of a device group and that device trust is established.

This task synchronizes the BIG-IP® configuration data from the local device to the devices in the device group. This synchronization ensures that devices in the device group operate properly. When synchronizing self IP addresses, the BIG-IP system synchronizes floating self IP addresses only.

Important: *You perform this task on either of the two devices, but not both.*

1. On the Main tab, click **Device Management** > **Overview**.
2. In the Device Groups area of the screen, in the Name column, select the name of the relevant device group.
   The screen expands to show a summary and details of the sync status of the selected device group, as well as a list of the individual devices within the device group.
3. In the Devices area of the screen, in the Sync Status column, select the device that shows a sync status of `Changes Pending`.
4. In the Sync Options area of the screen, select **Sync Device to Group**.
5. Click **Sync**.
   The BIG-IP system syncs the configuration data of the selected device in the Device area of the screen to the other members of the device group.

Except for non-floating self IP addresses, the entire set of BIG-IP configuration data is replicated on each device in the device group.

Implementation result

You now have a Sync-Failover device group set up with an active-standby DSC™ configuration. This configuration uses the default floating traffic group (named `traffic-group-1`), which contains the application-specific floating self IP and virtual IP addresses, and is initially configured to be active on one of the two devices. If the device with the active traffic group goes offline, the traffic group becomes active on the other device in the group, and application processing continues.

Creating an Active-Active Configuration using the Configuration Utility

Overview: Creating an active-active DSC configuration

A common TMOS® device service clustering (DSC™) implementation is an active-standby configuration, where a single traffic group is active on one of the devices in the device group, and is in a standby state on a peer device. Alternatively however, you can create a second traffic group and activate that traffic group on a peer device. In this *active-active* configuration, the devices each process traffic for a different application simultaneously. If one of the devices in the device group goes offline, the traffic group that was active on that device fails over to a peer device. The result is that two traffic groups can become active on one device.

To implement this DSC implementation, you create a Sync-Failover device group. A Sync-Failover device group with two or more members provides configuration synchronization and device failover, and optionally, connection mirroring.

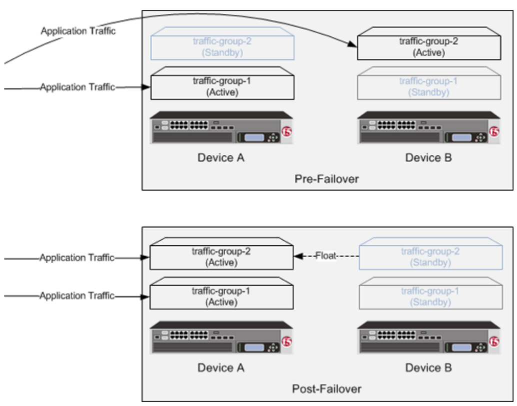


Figure 4: A two-member Sync-Failover group for an active-active configuration

About DSC configuration on a VIPRION system

The way you configure device service clustering (DSC™) (also known as redundancy) on a VIPRION® system varies depending on whether the system is provisioned to run the vCMP® feature.

For non-vCMP systems

For a device group that consists of VIPRION systems that are not licensed and provisioned for vCMP, each VIPRION cluster constitutes an individual device group member. The following table describes the IP addresses that you must specify when configuring redundancy.

Table 5: Required IP addresses for DSC configuration on a non-vCMP system

Feature	IP addresses required
Device trust	The primary floating management IP address for the VIPRION cluster.
ConfigSync	The unicast non-floating self IP address assigned to VLAN `internal`.

Feature	IP addresses required
Failover	• Recommended: The unicast non-floating self IP address that you assigned to an internal VLAN (preferably VLAN `HA`), as well as a multicast address. • Alternative: All unicast management IP addresses that correspond to the slots in the VIPRION cluster.
Connection mirroring	For the primary address, the non-floating self IP address that you assigned to VLAN `HA`. The secondary address is not required, but you can specify any non-floating self IP address for an internal VLAN..

For vCMP systems

On a vCMP system, the devices in a device group are virtual devices, known as *vCMP guests*. You configure device trust, config sync, failover, and mirroring to occur between equivalent vCMP guests in separate chassis.

For example, if you have a pair of VIPRION systems running vCMP, and each system has three vCMP guests, you can create a separate device group for each pair of equivalent guests. Table 4.2 shows an example.

Table 6: Sample device groups for two VIPRION systems with vCMP

Device groups for vCMP	Device group members
`Device-Group-A`	• `Guest1` on chassis1 • `Guest1` on chassis2
`Device-Group-B`	• `Guest2` on chassis1 • `Guest2` on chassis2
`Device-Group-C`	• `Guest3` on chassis1 • `Guest3` on chassis2

By isolating guests into separate device groups, you ensure that each guest synchronizes and fails over to its equivalent guest. The following table describes the IP addresses that you must specify when configuring redundancy:

Table 7: Required IP addresses for DSC configuration on a VIPRION system with vCMP

Feature	IP addresses required
Device trust	The cluster management IP address of the guest.
ConfigSync	The non-floating self IP address on the guest that is associated with VLAN `internal` on the host.
Failover	• Recommended: The unicast non-floating self IP address on the guest that is associated with an internal VLAN on the host (preferably VLAN `HA`), as well as a multicast address. • Alternative: The unicast management IP addresses for all slots configured for the guest.
Connection mirroring	For the primary address, the non-floating self IP address on the guest that is associated with VLAN `internal` on the host. The secondary address is not required, but you can specify any non-floating self IP address on the guest that is associated with an internal VLAN on the host.

DSC prerequisite worksheet

Before you set up device service clustering (DSC™), you must configure these BIG-IP® components on each device that you intend to include in the device group.

Table 8: DSC deployment worksheet

Configuration component	Considerations
Hardware, licensing, and provisioning	Devices in a device group must match with respect to product licensing and module provisioning. Heterogeneous hardware platforms within a device group are supported.
BIG-IP software version	Each device must be running BIG-IP version 11.x. This ensures successful configuration synchronization.
Management IP addresses	Each device must have a management IP address, a network mask, and a management route defined.
FQDN	Each device must have a fully-qualified domain name (FQDN) as its host name.
User name and password	Each device must have a user name and password defined on it that you will use when logging in to the BIG-IP Configuration utility.
`root` folder properties	The platform properties for the `root` folder must be set correctly (`Sync-Failover` and `traffic-group-1`).
VLANs	You must create these VLANs on each device, if you have not already done so: • A VLAN for the internal network, named `internal` • A VLAN for the external network, named `external` • A VLAN for failover communications, named `HA`
Self IP addresses	You must create these self IP addresses on each device, if you have not already done so: • Two self IP addresses (floating and non-floating) on the same subnet for VLAN `internal`. • Two self IP addresses (floating and non-floating) on the same subnet for VLAN `external`. • A non-floating self IP address on the internal subnet for VLAN `HA`. ***Note:*** *When you create floating self IP addresses, the BIG-IP system automatically adds them to the default floating traffic group,* `traffic-group-1`*. To add a self IP address to a different traffic group, you must modify the value of the self IP address* ***Traffic Group*** *property.* ***Important:*** *If the BIG-IP device you are configuring is accessed using Amazon Web Services, then the IP address you specify must be the floating IP address for high availability fast failover that you configured for the EC2 instance.*
Port lockdown	For self IP addresses that you create on each device, you should verify that the **Port Lockdown** setting is set to **Allow All**, **All Default**, or **Allow Custom**. Do not specify **None**.

Configuration component	Considerations
Application-related objects	You must create any virtual IP addresses and optionally, SNAT translation addresses, as part of the local traffic configuration. You must also configure any iApp™ application services if they are required for your application. When you create these addresses or services, the objects automatically become members of the default traffic group, `traffic-group-1`.
Time synchronization	The times set by the NTP service on all devices must be synchronized. This is a requirement for configuration synchronization to operate successfully.
Device certificates	Verify that each device includes an x509 device certificate. Devices with device certificates can authenticate and therefore trust one another, which is a prerequisite for device-to-device communication and data exchange.

Configurations using Sync-Failover device groups

This illustration shows two separate Sync-Failover device groups. In the first device group, only **LTM1** processes application traffic, and the two BIG-IP devices are configured to provide active-standby high availability. This means that **LTM1** and **LTM2** synchronize their configurations, and the failover objects on **LTM1** float to **LTM2** if **LTM1** becomes unavailable.

In the second device group, both **LTM1** and **LTM2** process application traffic, and the BIG-IP devices are configured to provide active-active high availability. This means that **LTM1** and **LTM2** synchronize their configurations, the failover objects on **LTM1** float to **LTM2** if **LTM1** becomes unavailable, and the failover objects on **LTM2** float to **LTM1** if **LTM2** becomes unavailable.

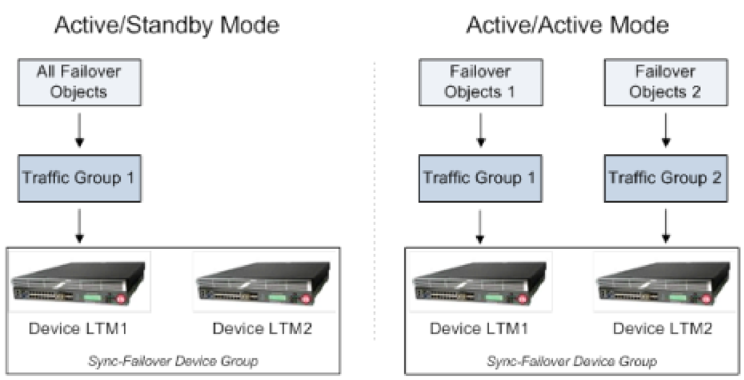


Figure 5: Comparison of Active-Standby and Active-Active device groups

Task summary

Use the tasks in this implementation to create a two-member device group, with two active traffic groups, that syncs the BIG-IP® configuration to the peer device and provides failover capability if the peer device goes offline. Note that on a vCMP® system, the devices in a specific device group are vCMP guests, one per chassis.

Important: *When you use this implementation, F5 Networks recommends that you synchronize the BIG-IP configuration twice, once after you create the device group, and again after you specify the IP addresses for failover.*

Task list

Specifying an IP address for config sync

Before configuring the config sync address, verify that all devices in the device group are running the same version of BIG-IP® system software.

You perform this task to specify the IP address on the local device that other devices in the device group will use to synchronize their configuration objects to the local device.

Note: *You must perform this task locally on each device in the device group.*

---

1. Confirm that you are logged in to the actual device you want to configure.
2. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
3. In the Name column, click the name of the device to which you are currently logged in.
4. From the Device Connectivity menu, choose ConfigSync.
5. For the **Local Address** setting, retain the displayed IP address or select another address from the list.

   F5 Networks recommends that you use the default value, which is the self IP address for VLAN `internal`. This address must be a non-floating self IP address and not a management IP address.

   Important: *If the BIG-IP device you are configuring is accessed using Amazon Web Services, then the internal self IP address that you specify must be the internal private IP addresses that you configured for this EC2 instance as the* ***Local Address****.*

   ---

6. Click **Update**.

After performing this task, the other devices in the device group can sync their configurations to the local device.

Specifying an IP address for connection mirroring

You can specify the local self IP address that you want other devices in a device group to use when mirroring their connections to this device. Connection mirroring ensures that in-process connections for an active traffic group are not dropped when failover occurs. You typically perform this task when you initially set up device service clustering (DSC®).

***Note:** You must perform this task locally on each device in the device group.*

1. Confirm that you are logged in to the actual device you want to configure.
2. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
3. In the Name column, click the name of the device to which you are currently logged in.
4. From the Device Connectivity menu, choose Mirroring.
5. For the **Primary Local Mirror Address** setting, retain the displayed IP address or select another address from the list.

   The recommended IP address is the self IP address for either VLAN `HA` or VLAN `internal`.

   ***Important:** If the BIG-IP device you are configuring is accessed using Amazon Web Services, then the self IP address you specify must be one of the private IP addresses that you configured for this EC2 instance as the **Primary Local Mirror Address**.*

6. For the **Secondary Local Mirror Address** setting, retain the default value of **None**, or select an address from the list.

   This setting is optional. The system uses the selected IP address in the event that the primary mirroring address becomes unavailable.

7. Click **Update**.

In addition to specifying an IP address for mirroring, you must also enable connection mirroring on the relevant virtual servers on this device.

Specifying the HA capacity of a device

Before you perform this task, verify that this device is a member of a device group and that the device group contains three or more devices.

You perform this task when you have more than one type of hardware platform in a device group and you want to configure load-aware failover. *Load-aware failover* ensures that the BIG-IP® system can intelligently select the next-active device for each active traffic group in the device group when failover occurs. As part of configuring load-aware failover, you define an HA capacity to establish the amount of computing resource that the device provides relative to other devices in the device group.

***Note:** If all devices in the device group are the same hardware platform, you can skip this task.*

1. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
2. In the Name column, click the name of the device for which you want to view properties.
   This displays a table of properties for the device.
3. In the **HA Capacity** field, type a relative numeric value.

   You need to configure this setting only when you have varying types of hardware platforms in a device group and you want to configure load-aware failover. The value you specify represents the relative capacity of the device to process application traffic compared to the other devices in the device group.

   ***Important:** If you configure this setting, you must configure the setting on every device in the device group.*

If this device has half the capacity of a second device and a third of the capacity of a third device in the device group, you can specify a value of `100` for this device, `200` for the second device, and `300` for the third device.
When choosing the next active device for a traffic group, the system considers the capacity that you specified for this device.

4. Click **Update**.

After you perform this task, the BIG-IP system uses the **HA Capacity** value to calculate the current utilization of the local device, to determine the next-active device for failover of other traffic groups in the device group.

Establishing device trust

Before you begin this task, verify that:

- Each BIG-IP® device that is to be part of the local trust domain has a device certificate installed on it.
- The local device is designated as a certificate signing authority.

You perform this task to establish trust among devices on one or more network segments. Devices that trust each other constitute the *local trust domain*. A device must be a member of the local trust domain prior to joining a device group.

By default, the BIG-IP software includes a local trust domain with one member, which is the local device. You can choose any one of the BIG-IP devices slated for a device group and log into that device to add other devices to the local trust domain. For example, devices `A`, `B`, and `C` each initially shows only itself as a member of the local trust domain. To configure the local trust domain to include all three devices, you can simply log into device `A` and add devices `B` and `C` to the local trust domain. Note that there is no need to repeat this process on devices `B` and `C`.

1. On the Main tab, click **Device Management** > **Device Trust**, and then either **Peer List** or **Subordinate List**.
2. Click **Add**.
3. Type a device IP address, administrator user name, and administrator password for the remote BIG-IP® device with which you want to establish trust. The IP address you specify depends on the type of BIG-IP device:
   - If the BIG-IP device is a non-VIPRION device, type the management IP address for the device.
   - If the BIG-IP device is a VIPRION device that is not licensed and provisioned for vCMP, type the primary cluster management IP address for the cluster.
   - If the BIG-IP device is a VIPRION device that is licensed and provisioned for vCMP, type the cluster management IP address for the guest.
   - If the BIG-IP device is an Amazon Web Services EC2 device, type one of the Private IP addresses created for this EC2 instance.
4. Click **Retrieve Device Information**.
5. Verify that the certificate of the remote device is correct.
6. Verify that the name of the remote device is correct.
7. Verify that the management IP address and name of the remote device are correct.
8. Click **Finished**.

The device you added is now a member of the local trust domain.

Repeat this task for each device that you want to add to the local trust domain.

Creating a Sync-Failover device group

This task establishes failover capability between two or more BIG-IP devices that you intend to run in an active-active configuration. If an active device in a Sync-Failover device group becomes unavailable, the configuration objects fail over to another member of the device group and traffic processing is unaffected. You perform this task on any one of the authority devices within the local trust domain.

Repeat this task for each Sync-Failover device group that you want to create for your network configuration.

1. On the Main tab, click **Device Management** > **Device Groups**.
2. On the Device Groups list screen, click **Create**.
   The New Device Group screen opens.
3. Type a name for the device group, select the device group type **Sync-Failover**, and type a description for the device group.
4. In the Configuration area of the screen, select a host name from the **Available** list for each BIG-IP device that you want to include in the device group, including the local device. Use the Move button to move the host name to the **Includes** list.

   The **Available** list shows any devices that are members of the device's local trust domain but not currently members of a Sync-Failover device group. A device can be a member of one Sync-Failover group only.
5. For the **Network Failover** setting, verify that network failover is enabled.

   Network failover must be enabled for active-active configurations (that is, device groups that will contain two or more active traffic groups).
6. Click **Finished**.

You now have a Sync-Failover type of device group containing BIG-IP devices as members. This device group is configured for environments that require the use of two or more active traffic groups to process application traffic.

Syncing the BIG-IP configuration to the device group

Before you sync the configuration, verify that the devices targeted for config sync are members of a device group and that device trust is established.

This task synchronizes the BIG-IP® configuration data from the local device to the devices in the device group. This synchronization ensures that devices in the device group operate properly. When synchronizing self IP addresses, the BIG-IP system synchronizes floating self IP addresses only.

Important: *You perform this task on either of the two devices, but not both.*

---

1. On the Main tab, click **Device Management** > **Overview**.
2. In the Device Groups area of the screen, in the Name column, select the name of the relevant device group.
   The screen expands to show a summary and details of the sync status of the selected device group, as well as a list of the individual devices within the device group.
3. In the Devices area of the screen, in the Sync Status column, select the device that shows a sync status of `Changes Pending`.
4. In the Sync Options area of the screen, select **Sync Device to Group**.
5. Click **Sync**.
   The BIG-IP system syncs the configuration data of the selected device in the Device area of the screen to the other members of the device group.

Except for non-floating self IP addresses, the entire set of BIG-IP configuration data is replicated on each device in the device group.

Specifying IP addresses for failover communication

You typically perform this task during initial Device Service Clustering (DSC®) configuration, to specify the local IP addresses that you want other devices in the device group to use for continuous health-assessment communication with the local device or guest. You must perform this task locally on each device in the device group.

Important: *If the system is running vCMP, you must log in to each guest to perform this task.*

Note: *The IP addresses that you specify must belong to route domain* `0`.

1. Confirm that you are logged in to the actual device you want to configure.
2. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
3. In the Name column, click the name of the device to which you are currently logged in.
4. From the Device Connectivity menu, choose Failover.
5. For the Failover Unicast Configuration settings, click **Add** for each IP address on this device that other devices in the device group can use to exchange failover messages with this device. The unicast IP addresses you specify depend on the type of device:

Platform	Action
Appliance without vCMP	Type a static self IP address associated with an internal VLAN (preferably VLAN `HA`) and the static management IP address currently assigned to the device.
Appliance with vCMP	Type a static self IP address associated with an internal VLAN (preferably VLAN `HA`) and the unique management IP address currently assigned to the guest.
VIPRION without vCMP®	Type a static self IP address associated with an internal VLAN (preferably VLAN `HA`), and, if you choose to specify unicast addresses only (and not a multicast address), you must also type the existing, static management IP addresses that you previously configured for all slots in the cluster. If you choose to specify one or more unicast addresses and a multicast address, then you do not need to specify the existing, per-slot static management IP addresses when configuring addresses for failover communication.
VIPRION with vCMP	Type a self IP address that is defined on the guest and associated with an internal VLAN on the host (preferably VLAN `HA`), and, if you choose to specify unicast failover addresses only (and not a a multicast address), you must also type the existing, virtual static management IP addresses that you previously configured for all slots in the guest's virtual cluster. If you choose to specify one or more unicast addresses and a multicast address, you do not need to specify the existing, per-slot static and virtual management IP addresses when configuring addresses for failover communication.

   Important: *Failover addresses should always be static, not floating, IP addresses.*

6. To enable the use of a failover multicast address on a VIPRION® platform (recommended), then for the **Use Failover Multicast Address** setting, select the **Enabled** check box.
7. If you enabled **Use Failover Multicast Address**, either accept the default **Address** and **Port** values, or specify values appropriate for the device.

If you revise the default **Address** and **Port** values, but then decide to revert to the default values, click **Reset Defaults**.

8. Click **Update**.

After you perform this task, other devices in the device group can send failover messages to the local device using the specified IP addresses.

Creating a second traffic group for the device group

This task creates a second active floating traffic group to process application traffic. The default floating traffic group (traffic-group-1) processes application traffic for the local device.

Note: *For this implementation, name this traffic group **traffic-group-2**.*

---

1. On the Main tab, click **Device Management** > **Traffic Groups**.
2. On the Traffic Group List screen, click **Create**.
3. Type the name `traffic-group-2` for the new traffic group.
4. In the **HA Load Factor** field, specify a value that represents the application load for this traffic group relative to other active traffic groups on the local device.

   Important: *If you configure this setting, you must configure the setting on every traffic group in the device group.*

   ---

5. In the **MAC Masquerade Address** field, type a MAC masquerade address.

   When you specify a MAC masquerade address, you reduce the risk of dropped connections when failover occurs. This setting is optional.

6. Select or clear the check box for the **Auto Failback** option:
   - Select the check box to cause the traffic group, after failover, to fail over again to the first device in the traffic group's ordered list when that device (and only that device) is available.
   - Clear the check box to cause the traffic group, after failover, to remain active on its current device until failover occurs again.

7. For the **Failover Order** setting, in the **Available** box, select a device name and using the Move button, move the device name to the **Enabled** box. Repeat for each device that you want to include in the ordered list.

   This setting is optional. Only devices that are members of the relevant Sync-Failover device group are available for inclusion in the ordered list. If you have enabled the auto-failback feature on the traffic group, ensure that the first device in the ordered list is the device to which you want this traffic group to fail back to when that first device becomes available.

   If auto-failback is enabled and the first device in the **Failover Order** list is unavailable, no auto-failback occurs and the traffic group continues to run on the current device. Also, if none of the devices in the **Failover Order** list is currently available when failover occurs, the BIG-IP system ignores the **Failover Order** setting and performs load-aware failover instead, using the **HA Load Factor** setting.

8. Click **Finished**.

You now have a second floating traffic group on the local device (in addition to the default floating traffic group) so that once the traffic group is activated on the remote devices, devices in the device group can process traffic for different applications.

Assigning traffic-group-2 to a floating virtual IP address

This task assigns a floating traffic group to a virtual IP address on a device.

1. On the Main tab, click **Local Traffic** > **Virtual Servers** > **Virtual Address List**.
   The Virtual Address List screen opens.
2. In the Name column, click the virtual address that you want to assign to the traffic group.
   This displays the properties of that virtual address.
3. From the **Traffic Group** list, select **traffic-group-2 (floating)**.
4. Click **Update**.

The device's floating virtual IP address is now a member of your second traffic group. The virtual IP address can now fail over to other devices in the device group.

Assigning traffic-group-2 to a floating self IP address

This task assigns your floating self IP address to traffic-group-2.

1. On the Main tab, click **Network** > **Self IPs**.
2. In the Name column, click the floating self IP address assigned to VLAN `internal`.
   This displays the properties of that self IP address.
3. From the **Traffic Group** list, select **traffic-group-2 (floating)**.
4. Click **Update**.

The device's floating self IP address is now a member of your second traffic group. The self IP address can now fail over to other devices in the traffic group.

Syncing the BIG-IP configuration to the device group

Before you sync the configuration, verify that the devices targeted for config sync are members of a device group and that device trust is established.

This task synchronizes the BIG-IP® configuration data from the local device to the devices in the device group. This synchronization ensures that devices in the device group operate properly. When synchronizing self IP addresses, the BIG-IP system synchronizes floating self IP addresses only.

Important: *You perform this task on either of the two devices, but not both.*

---

1. On the Main tab, click **Device Management** > **Overview**.
2. In the Device Groups area of the screen, in the Name column, select the name of the relevant device group.
   The screen expands to show a summary and details of the sync status of the selected device group, as well as a list of the individual devices within the device group.
3. In the Devices area of the screen, in the Sync Status column, select the device that shows a sync status of `Changes Pending`.
4. In the Sync Options area of the screen, select **Sync Device to Group**.
5. Click **Sync**.
   The BIG-IP system syncs the configuration data of the selected device in the Device area of the screen to the other members of the device group.

Except for non-floating self IP addresses, the entire set of BIG-IP configuration data is replicated on each device in the device group.

Forcing a traffic group to a standby state

You perform this task when you want the selected traffic group on the local device to fail over to another device (that is, switch to a `Standby` state). Users typically perform this task when no automated method is configured for a traffic group, such as auto-failback or an HA group. By forcing the traffic group into a `Standby` state, the traffic group becomes active on another device in the device group. For device groups with more than two members, you can choose the specific device to which the traffic group fails over.

1. Log in to the device on which the traffic group is currently active.
2. On the Main tab, click **Device Management** > **Traffic Groups**.
3. In the Name column, locate the name of the traffic group that you want to run on the peer device.
4. Select the check box to the left of the traffic group name.

   If the check box is unavailable, the traffic group is not active on the device to which you are currently logged in. Perform this task on the device on which the traffic group is active.
5. Click **Force to Standby**.
   This displays target device options.
6. Choose one of these actions:
   - If the device group has two members only, click **Force to Standby**. This displays the list of traffic groups for the device group and causes the local device to appear in the Next Active Device column.
   - If the device group has more than two members, then from the **Target Device** list, select a value and click **Force to Standby**.

The selected traffic group is now in a standby state on the local device and active on another device in the device group.

Implementation result

You now have a Sync-Failover device group set up with an active-active DSC™ configuration. In this configuration, each device has a different active traffic group running on it. That is, the active traffic group on one device is the default traffic group (named `traffic-group-1`), while the active traffic group on the peer device is a traffic group that you create. Each traffic group contains the floating self IP and virtual IP addresses specific to the relevant application.

If one device goes offline, the traffic group that was active on that device becomes active on the other device in the group, and processing for both applications continues on one device.

Configuring Load-aware Failover

Overview: Implementing load-aware failover

Load-aware failover is a BIG-IP® feature designed for use in a Sync-Failover device group. Configuring *load-aware failover* ensures that the traffic load on all devices in a device group is as equivalent as possible, factoring in any differences in device capacity and the amount of application traffic that traffic groups process on a device.

For example, suppose you have a heterogeneous three-member device group in which one device (`Bigip_C`) has twice the hardware capacity of the other two devices (`Bigip_A` and `Bigip_B`).

If the device group has four active traffic groups that each process the same amount of application traffic, then the load on all devices is equivalent when devices `Bigip_A` and `Bigip_B` each contain one active traffic group, while device `Bigip_C` contains two active traffic groups.

The figure shows a Sync-Failover device group where application traffic is directed to the device with the most capacity relative to the other device group members.

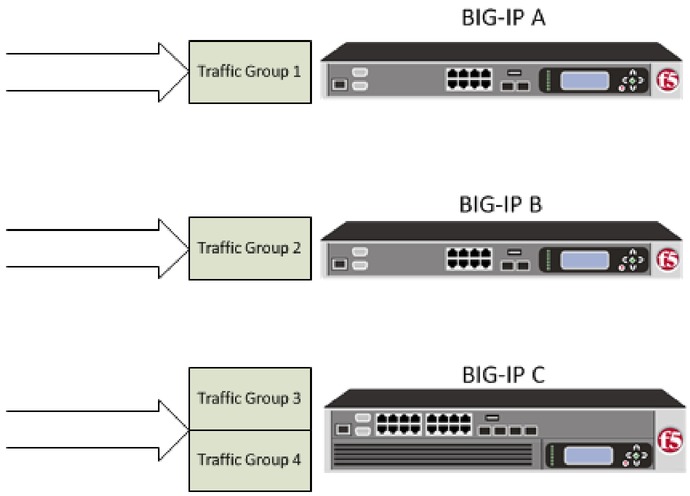


Figure 6: Sync-Failover device group with traffic groups of equal load:

The BIG-IP system implements load-aware failover by calculating a numeric, current utilization score for each device, based on numeric values that you specify for each device and traffic group relative to the other devices and traffic groups in the device group. The system then uses this current utilization score to determine which device is the best device in the group to become the next-active device when failover occurs for a traffic group.

The overall result is that the traffic load on each device is as equivalent as possible in a relative way, that is, factoring in individual device capacity and application traffic load per traffic group.

Task List

About device utilization calculation

The BIG-IP® system on each device performs a calculation to determine the device's current level of utilization. This utilization level indicates the ability for the device to be the next-active device in the event that an active traffic group on another device must fail over within a heterogeneous device group.

The calculation that the BIG-IP performs to determine the current utilization of a device is based on these factors:

Device capacity
A local device capacity relative to other device group members.

Active local traffic groups
The number of active traffic groups on the local device.

Active remote traffic groups
The number of remote active traffic groups for which the local device is the next-active device.

A multiplying load factor for each active traffic group
A multiplier value for each traffic group. The system uses this value to weight each active traffic group's traffic load compared to the traffic load of each of the other active traffic groups in the device group.

The BIG-IP system uses all of these factors to perform a calculation to determine, at any particular moment, a score for each device that represents the current utilization of that device. This utilization score indicates whether the BIG-IP system should, in its attempt to equalize traffic load on all devices, designate the device as a next-active device for an active traffic group on another device in the device group.

The calculation that the BIG-IP performs for each device is:

```
   (The sum of local active traffic group loads + The sum of remote active
traffic group loads) / device capacity
```

Task summary

To implement load-aware failover you specify a value representing the relative traffic load for a traffic group and, optionally, a value representing the relative capacity of the BIG-IP® device.

Task list

Specifying the HA capacity of a device

Before you perform this task, verify that this device is a member of a device group and that the device group contains three or more devices.

You perform this task when you have more than one type of hardware platform in a device group and you want to configure load-aware failover. *Load-aware failover* ensures that the BIG-IP® system can intelligently select the next-active device for each active traffic group in the device group when failover occurs. As part of configuring load-aware failover, you define an HA capacity to establish the amount of computing resource that the device provides relative to other devices in the device group.

Note: *If all devices in the device group are the same hardware platform, you can skip this task.*

1. On the Main tab, click **Device Management** > **Devices**.
   This displays a list of device objects discovered by the local device.
2. In the Name column, click the name of the device for which you want to view properties.
   This displays a table of properties for the device.
3. In the **HA Capacity** field, type a relative numeric value.

   You need to configure this setting only when you have varying types of hardware platforms in a device group and you want to configure load-aware failover. The value you specify represents the relative capacity of the device to process application traffic compared to the other devices in the device group.

   Important: *If you configure this setting, you must configure the setting on every device in the device group.*

   If this device has half the capacity of a second device and a third of the capacity of a third device in the device group, you can specify a value of `100` for this device, `200` for the second device, and `300` for the third device.
   When choosing the next active device for a traffic group, the system considers the capacity that you specified for this device.
4. Click **Update**.

After you perform this task, the BIG-IP system uses the **HA Capacity** value to calculate the current utilization of the local device, to determine the next-active device for failover of other traffic groups in the device group.

Specifying an HA load factor for a traffic group

You perform this task when you want to specify the relative application load for an existing traffic group, for the purpose of configuring load-aware failover. *Load-aware failover* ensures that the BIG-IP® system can intelligently select the next-active device for each active traffic group in the device group when failover occurs. When you configure load-aware failover, you define an application traffic load (known as an *HA load factor*) for a traffic group to establish the amount of computing resource that an active traffic group uses relative to other active traffic groups.

1. On the Main tab, click **Device Management** > **Traffic Groups**.
2. In the Name column, click the name of a traffic group.

   This displays the properties of the traffic group.
3. From the **Failover Methods** list, select **Load Aware**.
   This displays the **HA Load Factor** setting.
4. In the **HA Load Factor** field, specify a value that represents the application load for this traffic group relative to other active traffic groups on the local device.

   Important: *If you configure this setting, you must configure the setting on every traffic group in the device group.*
5. Click **Update**.

After performing this task, the BIG-IP system uses the **HA Load Factor** value as a factor in calculating the current utilization of the local device, to determine whether this device should be the next-active device for failover of other traffic groups in the device group.

Implementation Results

For this implementation example, the load-aware configuration now consists of both a user-specified relative high availability (HA) hardware capacity for each device and a relative load factor for each active traffic group.

Using the example in the overview, devices `Bigip_A` and `Bigip_B` are the same hardware platform and therefore have the same HA capacity, while `Bigip_C` has twice the HA capacity of the other two devices. Also, devices `Bigip_A` and `Bigip_B` currently have one active traffic group each, while `Bigip_C` has two active traffic groups. All three traffic groups process the same amount of application traffic.

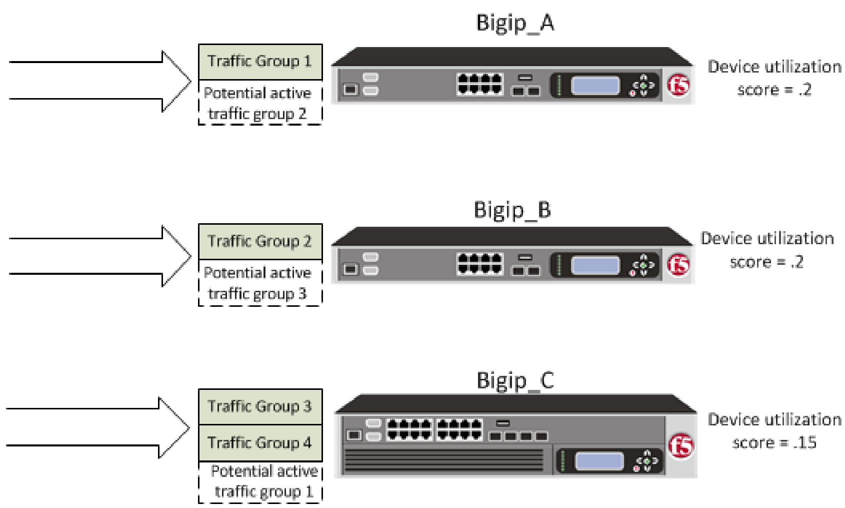


Figure 7: Device utilization scores based on device capacity and traffic group load

The device utilization score that the BIG-IP® system calculates in this implementation is the sum of all traffic load values on a device divided by the device capacity.

Table 9: Calculating the utilization score for Bigip_A

HA capacity	Active traffic group	HA load factor	Potential active traffic group	HA load factor	Device utilization score
`10`	`Traffic-group-1`	1	`Traffic-group-2`	1	2/10 = .2

Table 10: Calculating the utilization score for Bigip_B

HA capacity	Active traffic group	HA load factor	Potential active traffic group	HA load factor	Device utilization score
`10`	`Traffic-group-2`	1	`Traffic-group-3`	1	2/10=.2

Table 11: Calculating the utilization score for Bigip_C

HA capacity	Active traffic group	HA load factor	Potential active traffic group	HA load factor	Device utilization score
20	Traffic-group-3 and Traffic-group-4	1 and 1	Traffic-group-1	1	3/20=.15

This example shows the results of the calculations that the BIG-IP system performs for each device in the device group. The example shows that although device `Bigip_C` currently has the two active traffic groups, the device has the most available resource due to having the lowest utilization score of .15. In this case, `Bigip_C` is most likely the next-active device for the other two devices in the device group.

Managing Traffic with Bandwidth Controllers

Overview: Bandwidth control management

Fine-grained bandwidth control is essential to service providers, large enterprises, and remote access services (RAS) solutions. Bandwidth controllers on the BIG-IP® system can scale easily, work well in a distributed environment, and are easy to configure for various networks. Depending on the type of policy you configure, you can use bandwidth controllers to apply specified rate enforcement to traffic flows or mark traffic that exceeds limits.

Bandwidth control policies can be static or dynamic. Through the user interface (browser or `tmsh` command-line utility), when you apply a bandwidth control policy to a virtual server, packet filter, or route domain, you can apply only one policy at a time, and that is a static policy. Using iRules®, you can combine static and dynamic bandwidth control policies up to eight policies on a connection, but only one of the eight policies can be a dynamic policy. A packet is transmitted only when all the attached policies allow it. The system as a whole supports a maximum of 1024 policies.

Bandwidth controllers vs. rate shaping

Bandwidth controller is the updated version of rate shaping on the BIG-IP® system. These features are mutually exclusive. You can configure and use either rate shaping or bandwidth controllers, but not both. Bandwidth controllers include distributed control, subscriber fairness, and support for a maximum rate of 320 Gbps. Rate shaping is hierarchical and supports minimum bandwidth (committed information rate), priority, and flow fairness.

About static bandwidth control policies

A *static* bandwidth control policy controls the aggregate rate for a group of applications or a network path. It enforces the total amount of bandwidth that can be used, specified as the maximum rate of the resource you are managing. The rate can be the total bandwidth of the BIG-IP® device, or it might be a group of traffic flows.

Task summary for creating a static bandwidth control policy

This procedure includes the steps for assigning a static bandwidth control policy to traffic, using a virtual server. Alternatively, you can assign a static bandwidth control policy to a packet filter or a route domain.

Task list

Creating a static bandwidth control policy

Adding a static bandwidth control policy to a virtual server

Creating a static bandwidth control policy

You can create a static bandwidth control policy to limit the bandwidth that traffic uses on the BIG-IP® system.

1. On the Main tab, click **Acceleration** > **Bandwidth Controllers**.
2. Click **Create**.
3. In the **Name** field, type a name for the bandwidth control policy.
4. In the **Maximum Rate** field, type a number and select the unit of measure to indicate the total throughput allowed for the resource you are managing.

   The number must be in the range from `1 Mbps` to `320 Gbps`. This value is the amount of bandwidth available to all the connections going through this static policy.
5. Click **Finished**.

For the bandwidth control policy to take effect, you must apply the policy to traffic, using a virtual server, packet filter, or route domain.

Adding a static bandwidth control policy to a virtual server

Adding a static bandwidth control policy to a virtual server is one way to apply the policy to traffic. Alternatively, you can add the bandwidth control policy to a packet filter or a route domain.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. From the **Configuration** list, select **Advanced**.
4. From the **Bandwidth Controller** list, select a bandwidth control policy.

   Only static bandwidth control policies are available in this list.
5. Click **Update** to save the changes.

The BIG-IP® system now applies rate enforcement to the traffic intercepted by this virtual server, according to the static bandwidth policy you selected. A static bandwidth policy associated with a virtual server applies only to client-initiated flows, and not to bandwidth for traffic flowing toward the client.

About dynamic bandwidth control policies

You can create dynamic bandwidth control policies to restrict bandwidth usage per subscriber or group of subscribers, per application, per network egress link, or any combination of these. A *dynamic* bandwidth control policy provides fairness on traffic flows, according to configurable parameters, within an upper bandwidth limit. The BIG-IP® system activates the dynamic bandwidth control policy for each user only when the user participates. When you create a dynamic bandwidth control policy, it acts as a policy in waiting, until the system detects egress traffic that matches the traffic you want to control and creates an instance of the policy. At that moment, the system applies the bandwidth control policy limits, as specified. No bandwidth control occurs until the system detects traffic and creates an instance of the policy. With this feature, an Internet service provider (ISP) can create and revise a single policy that can apply to millions of users.

The BIG-IP system can enforce multiple levels of bandwidth limits through the dynamic policy. For example, a user could be limited by the maximum rate, a per user rate, and a per category rate (such as for an application), all from the same dynamic policy. When the total of the maximum user rate for all the instances exceeds the maximum rate specified in the dynamic policy, the BIG-IP system maintains fairness among all users and spreads the limitation equally among users belonging to a dynamic policy.

You can also configure a dynamic bandwidth control policy to mark packets that exceed the maximum per-user rate for a specified session. The WAN router should handle the marked packets. The BIG-IP system passes packets that conform to the maximum per-user rate without marking them. You configure marking by using the **IP Type of Service** or **Link Quality of Service** setting. For example, a common use of QoS marking is for Voice over IP (VoIP) traffic. VoIP is usually assigned to the Expedited Forwarding (EF) class by using the DSCP value of 46, thus prioritized according to importance and sensitivity to loss/latency.

Alternatives for identifying users and applying dynamic bandwidth control policies to traffic are using iRules®, Policy Enforcement Manager™, or Access Policy Manager®.

Task summary for creating a dynamic bandwidth control policy

Before you create a dynamic bandwidth control policy, F5 recommends that you select the **Source Address** for the **CMP Hash** setting on the VLAN properties screen for the VLAN that carries the traffic you want to manage. The BIG-IP® system uses source and destination hashes to control the way incoming traffic is distributed among the instances of the Traffic Management Microkernel (TMM) service. Subscriber-based bandwidth control depends on having a unique one-to-one relationship between bandwidth control policy and subscriber. Subscribers are commonly identified using a unique IP address, and, therefore, load distribution among the instances of TMM service must use the source IP address as the key.

This screen snippet highlights the proper setting.

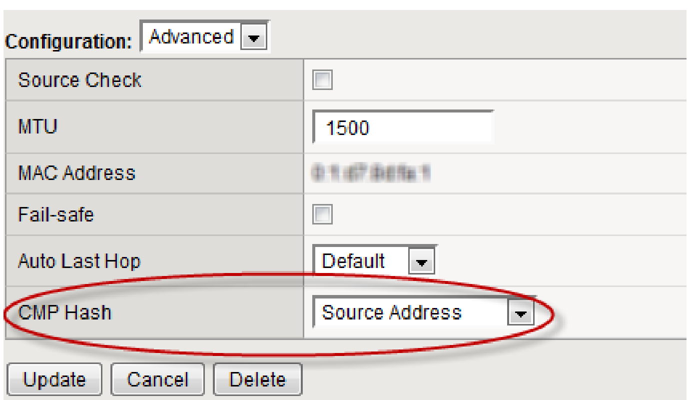


Figure 8: CMP Hash setting for dynamic bandwidth control

This procedure describes the steps for attaching a dynamic bandwidth control policy to a traffic flow, and then applying the policy to traffic, using a virtual server. For information about using Policy Enforcement Manager™ to implement the policy, refer to the F5 documentation for Policy Enforcement Manager.

Task list

Creating a dynamic bandwidth control policy
Creating an iRule for a dynamic bandwidth control policy
Adding a dynamic bandwidth control policy to a virtual server

Creating a dynamic bandwidth control policy

You can create a dynamic bandwidth control policy to shape the traffic to which you apply the policy.

1. On the Main tab, click **Acceleration** > **Bandwidth Controllers**.
2. Click **Create**.
3. In the **Name** field, type a name for the bandwidth control policy.
4. In the **Maximum Rate** field, type a number and select the unit of measure to indicate the total throughput allowed for all the instances created for this dynamic policy.

   The number must be in the range from `1 Mbps` to `320 Gbps`.
5. From the **Dynamic** list, select **Enabled**.
   The screen displays additional settings.
6. In the **Maximum Rate Per User** field, type a number and select the unit of measure to indicate the most bandwidth that each user or session associated with the bandwidth control policy can use.

   The number must be in the range from `1 Mbps` to `2 Gbps`.
7. From the **IP Type of Service** list, select **Specify** and type a number between `0` and `63` to assign a Type of Service (ToS) level to packets that exceed the maximum per-user rate.

   If you do not want to set a ToS level, maintain the default setting, **Pass Through**.
8. From the **Link Quality of Service** list, select **Specify** and type a number between `0` and `7` to assign a Quality of Service (QoS) level to packets that exceed the maximum per-user rate.

   If you do not want to set a QoS level, maintain the default setting, **Pass Through**.
9. In the **Categories** field, add up to eight categories of traffic that this bandwidth control policy manages.

   All the categories share the specified bandwidth, in accordance with the rate specified for each category.

   Note: *Use the* ***Categories*** *setting only if you have not set values for the* ***IP Type of Service*** *or the* ***Link Quality of Service*** *setting.*

   a) In the **Category Name** field, type a descriptive name for the category.
   b) In the **Max Category Rate** field, type a value to indicate the most bandwidth that this category of traffic can use, and select the unit of measure from the list, or select **%** and type a percentage from `1` to `100`.

      If you specify a rate, the number must be in the range from `500 Kbps` to the rate specified for the **Maximum Rate Per User** setting. A percentage indicates that this category can use up to the specified percentage of the maximum per-user rate. These values are upper limits (not minimum or guaranteed), so the sum can exceed the value you specified for the **Maximum Rate Per User** setting.
   c) Click **Add** to add the category to the **Categories** list.
   d) Repeat these steps to add more categories, up to a maximum of eight.

10. Click **Finished**.

For the dynamic bandwidth control policy to take effect, you must attach the policy to a traffic flow, and then apply the policy to traffic, using a virtual server, Policy Enforcement Manager™, or Access Policy Manager®.

Creating an iRule for a dynamic bandwidth control policy

To implement a dynamic bandwidth control policy, you can use iRules® to attach the policy to a user.

***Note:** For complete and detailed information iRules syntax, see the F5 Networks DevCentral web site (`http://devcentral.f5.com`).*

1. On the Main tab, click **Local Traffic** > **iRules**.
   The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
   The New iRule screen opens.
3. In the **Name** field, type a unique name between 1 and 31 characters for the iRule.
4. In the **Definition** field, type the syntax for the iRule using Tool Command Language (Tcl) syntax.
   For example, to apply the dynamic bandwidth policy `dynamic_bwc_policy200` to a user session, type the following iRule, where `set mycookie` defines a user session. A *session* is a combination of client IP address and port.

```
when CLIENT_ACCEPTED {
    set mycookie [IP::remote_addr]:[TCP::remote_port]
    BWC::policy attach dynamic_bwc_policy200 $mycookie

}
```

5. Click **Finished**.
   The new iRule appears in the list of iRules on the system.

You have now identified the user for a dynamic bandwidth control policy.

You must then apply the iRule to the virtual server that intercepts the traffic you want to manage.

Adding a dynamic bandwidth control policy to a virtual server

After you attach a dynamic bandwidth control policy to a user, using iRules®, you must apply the policy to traffic by adding the iRule to a virtual server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. On the menu bar, click **Resources**.
4. In the iRules area, click **Manage**.
5. From the **Available** list, select the name of the iRule that you want to assign, and using the Move button, move the name to the **Enabled** list.
6. Click **Finished**.

The BIG-IP® system now manages bandwidth for the traffic intercepted by this virtual server, according to the dynamic bandwidth policy specified in the assigned iRule.

Example of a dynamic bandwidth control policy

This screen is an example of a dynamic bandwidth control policy that might be created by an Internet service provider (ISP) to manage individual mobile subscribers.

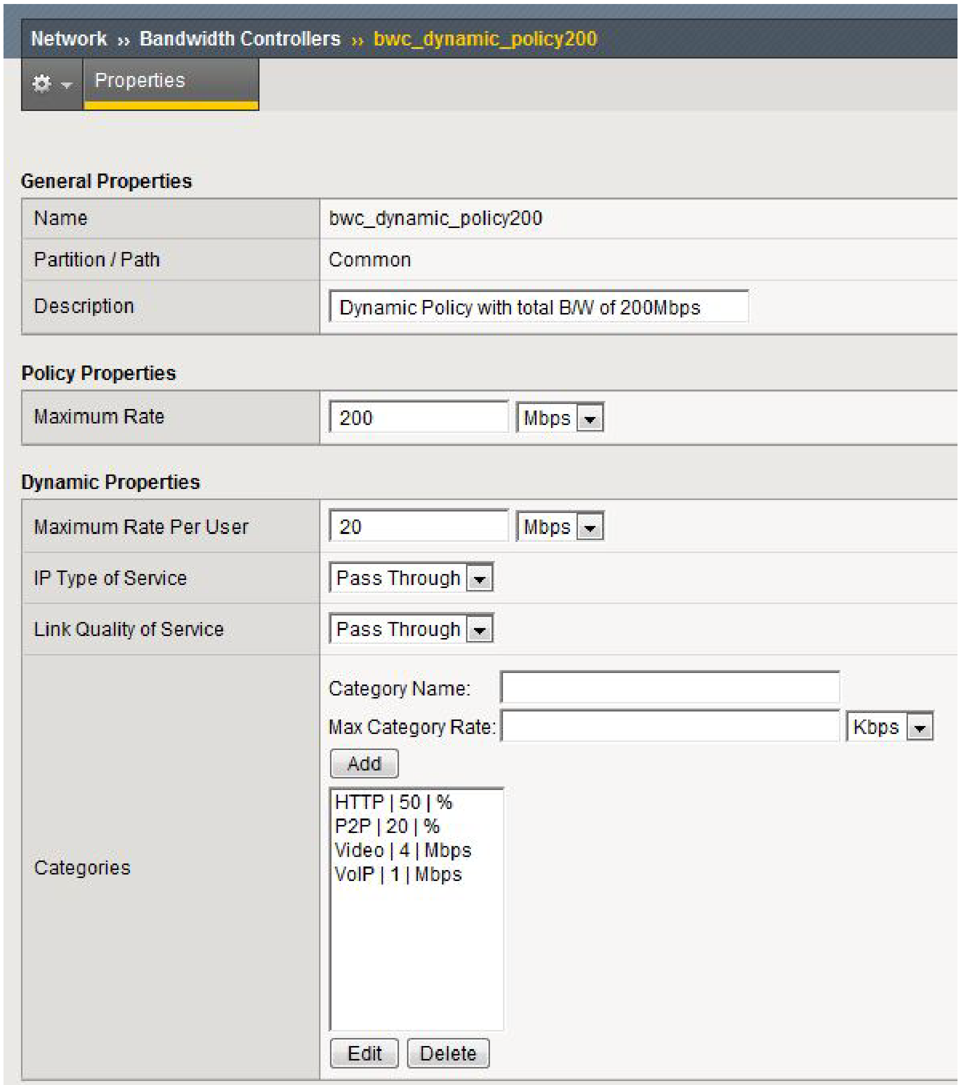


Figure 9: Example of completed dynamic bandwidth control policy screen

In the example, the ISP sets the maximum bandwidth at 200 Mbps. Of that bandwidth, a maximum of 20 Mbps is allocated to each user. Of that allocation, application traffic is apportioned, as follows.

- 50% applies to browser traffic (HTTP)
- 20% applies to P2P
- 4 Mbps applies to video
- 1 Mpbs applies to Voice over IP (VoIP)

To activate this policy, the ISP needs to create an iRule to attach the policy to a user session, and then apply the policy to a virtual server.

The bandwidth controller is only an enforcer. For a dynamic bandwidth control policy, you also need iRules®, Policy Enforcement Manager™, or Access Policy Manager® to identify a flow and map it to a category.

Configuring Network Virtualization Segments

Overview: Configuring network virtualization tunnels

Large data centers and cloud service providers are benefiting from large scale network virtualization. Network Virtualization provides connectivity in cloud environments by overlaying Layer 2 segments over a Layer 3 infrastructure. The overlay network can be dynamically extended with multiple virtualized networks without affecting the Layer 3 infrastructure. This number of virtualized networks is typically much larger than the number of VLANS the infrastructure can support.

You can configure a BIG-IP® system to function as a gateway in a virtualized network, bridging the data center virtualized networks with the physical network (L2 gateway), or performing routing and higher L4-L7 functionality among virtual networks of different types (L3 gateway). Connecting these networks allows for expansion, and provides a mechanism to streamline the transition of data centers into a virtualized model, while maintaining connectivity.

This illustration shows the BIG-IP system as a network virtualization gateway.

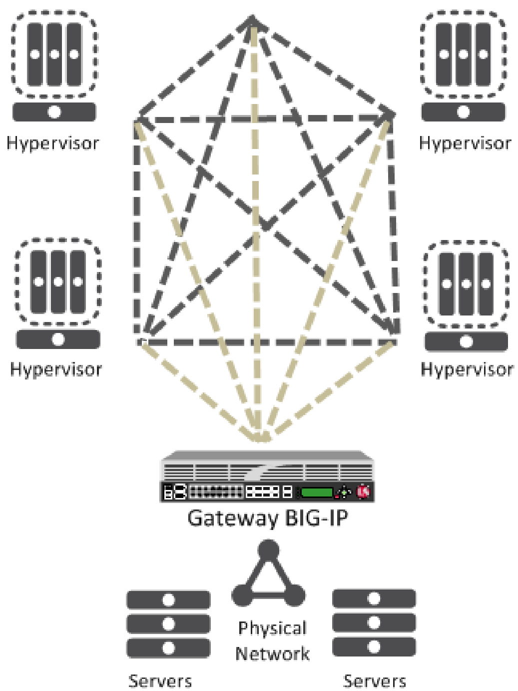


Figure 10: The BIG-IP system as a network virtualization gateway

In a virtualized network, the BIG-IP system needs to learn about other virtualization tunnel endpoints. Each hypervisor has a tunnel endpoint. The hypervisor needs to locate the virtual machines it manages, by maintaining a form of the L2 location records, typically, IP addresses and MAC addresses, virtual network identifiers, and virtual tunnel endpoints.

About network virtualization tunnels on the BIG-IP system

When you configure a BIG-IP® system as a network virtualization gateway, the system represents the connection as a tunnel, which provides a Layer 2 interface on the virtual network. You can use the tunnel interface in both Layer 2 and Layer 3 configurations. After you create the network virtualization tunnels, you can use the tunnels like you use VLANs on a BIG-IP system, such as for routing, assigning self IP addresses, and associating with virtual servers.

Creating a network virtualization tunnel

Creating a network virtualization tunnel on a BIG-IP® system provides an L2 gateway to connect the physical underlay network with a virtual overlay network.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select the tunnel profile you created for network virtualization.

   This selection must be a profile based on either the `gre` or `vxlan` parent profile, depending on your virtualized network environment.
4. In the **Local Address** field, type the self IP address of the VLAN through which the remote hypervisor is reachable.
5. For the **Remote Address** list, retain the default selection, **Any**.
6. In the **Key** field, type the VNI (Virtual Network Identifier) to use for the VXLAN tunnel.
7. Click **Finished**.

This tunnel is now available to use in virtualized network routing configurations, depending on how you configure your network.

Virtualized network terminology

These terms are associated with virtualized networks.

forwarding database (FDB)
The *FDB* is the database that contains mappings between the MAC address of each virtual machine and the IP address of the hypervisor machine on which it resides.

L2 gateway
The Layer 2 gateway performs the bridge functionality between VLAN and virtual segments in a virtualized network.

L3 gateway
The Layer 3 gateway performs routing and higher L4-L7 functionality among virtualized network segments of different types.

overlay network
The *overlay network* is a virtual network of VMs built on top of a stable L2-L3 structure. The view from one VM to another is as if they were on the same switch, but, in fact, they could be far afield.

tunnel endpoint

A *tunnel endpoint* originates or terminates a tunnel. In a virtualized network environment, the tunnel IP addresses are part of the L2 underlay network. The same local IP address can be used for multiple tunnels.

underlay network

The *underlay network* is the L2 or L3 routed physical network, a mesh of tunnels.

virtualized network

A *virtualized network* is when you create a virtual L2 or L3 topology on top of a stable physical L2 or L3 network. Connectivity in the virtual topology is provided by tunneling Ethernet frames in IP over the physical network.

VNI

The Virtual Network Identifier (VNI) is also called the VXLAN segment ID. The system uses the VNI to identify the appropriate tunnel.

VSID

The Virtual Subnet Identifier (VSID) is a 24-bit identifier used in an NVGRE environment that represents a virtual L2 broadcast domain, enabling routes to be configured between virtual subnets.

VTEP

The *VXLAN Tunnel Endpoint (VTEP)* originates or terminates a VXLAN tunnel. The same local IP address can be used for multiple tunnels.

VXLAN

Virtual eXtended LAN (VXLAN) is a network virtualization scheme that overlays Layer 2 over Layer 3. VLXAN uses Layer 3 multicast to support the transmission of multicast and broadcast traffic in the virtual network, while decoupling the virtualized network from the physical infrastructure.

VXLAN gateway

A *VXLAN gateway* bridges traffic between VXLAN and non-VXLAN environments. The BIG-IP® system uses a VXLAN gateway to bridge a traditional VLAN and a VXLAN network, by becoming a network virtualization endpoint.

VXLAN header

In addition to the UDP header, encapsulated packets include a *VXLAN header*, which carries a 24-bit VNI to uniquely identify Layer 2 segments within the overlay.

VXLAN segment

A *VXLAN segment* is a Layer 2 overlay network over which VMs communicate. Only VMs within the same VXLAN segment can communicate with each other.

Centralized vs. decentralized models of network virtualization

Using the BIG-IP® system as a network virtualization gateway, you can set up virtualized network segments using either a centralized or decentralized model.

Centralized model

In a centralized model, a network orchestrator or controller manages the virtualized network segments. The orchestrator has full view of VTEPs, L2, and L3 information in the overlay, and is responsible for pushing this information to hypervisors and gateways. Microsoft Hyper-V and VMware NSX environments use this model.

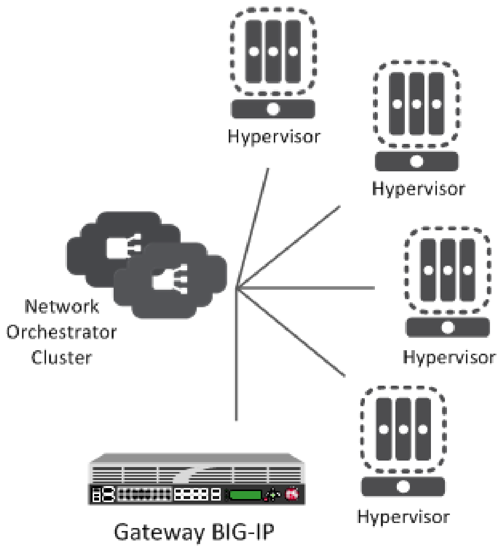


Figure 11: Centralized model of network virtualization

Decentralized model

A decentralized model of network virtualization does not require a network orchestrator or controller. In this model, the router learns the tunnel endpoint and MAC address locations by flooding broadcast, multicast, and unknown destination frames over IP multicast. VMware vSphere 5.1 environments use this model.

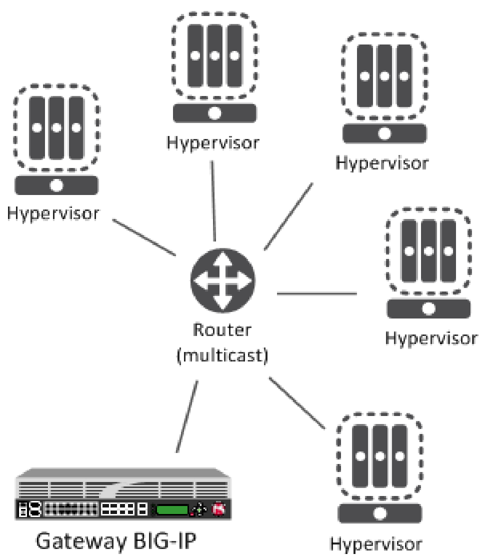


Figure 12: Decentralized model of network virtualization

About network virtualization tunnel types

The BIG-IP® system supports multiple network virtualization tunnel types. You can even combine virtualized network segments based on different tunnel types. This table offers a quick comparison of the tunnel types.

VXLAN (Multicast)	VXLAN (Unicast)	NVGRE	Transparent Ethernet Bridging
Decentralized	Centralized	Centralized	Centralized

VXLAN (Multicast)	VXLAN (Unicast)	NVGRE	Transparent Ethernet Bridging
VMware vSphere 5.1	VMware NSX	Microsoft SCVMM/Hyper-V	OpenStack
VXLAN UDP Encapsulation	VXLAN UDP Encapsulation	GRE-based Encapsulation	GRE-based Encapsulation
24-bit ID	24-bit ID	24-bit ID	32-bit ID
Endpoints discovered dynamically	Endpoints statically configured	Endpoints statically configured	Endpoints statically configured
Floods unknown and broadcast frames using IP multicast.	Can flood using unicast replication.	Does not flood (completely static).	Floods using unicast replication.

About statically configured network virtualization tunnels

For the centralized model, you can use VXLAN (Unicast), NVGRE, or Transparent Ethernet Bridging, depending on the cloud environment. Using an agent or plug-in, or the `tmsh` command-line utility, you can statically configure the FDB and ARP forwarding table entries. Using the `tmsh` command-line utility or browser interface, you can create the network virtualization tunnels, which are managed by the network controller.

Considerations for statically configured network virtualization tunnels

As you configure a BIG-IP® system to be an L2 or L3 gateway for statically configured network virtualization tunnels, keep these considerations in mind.

- The BIG-IP system must be licensed for SDN Services.
- If you have over 2000 connections, set the **Management (MGMT)** setting on the Resource Provisioning screen is to **Large** (**System** > **Resource Provisioning**).

Examples for manually populating L2 location records

Using the `tmsh` command-line utility, you can add static FDB records and ARP entries for each virtual tunnel endpoint.

- Add static FDB (forwarding database) entries to associate MAC addresses with specified tunnel endpoints. For example, the following command creates an FDB entry that associates the MAC address `00:01:02:03:04:05` with the tunnel endpoint `10.1.1.1` of the tunnel `vxlan0`.

```
# tmsh modify net fdb tunnel vxlan0 records add {
    00:01:02:03:04:05 { endpoint 10.1.1.1 } }
```

- Delete a MAC address from an FDB entry.

```
# tmsh modify net fdb tunnel vxlan0 records del { 00:01:02:03:04:05 }
```

- Add an IP address to a MAC address in the ARP table.

```
# tmsh modify net arp 10.3.3.1 { ip-address 10.3.3.1 mac-address
00:01:02:03:04:05 }
}
```

Using the iControl/REST API, you can program a network controller to build and maintain network virtualization tunnels. This example adds an entry to the FDB table that associates the MAC address `00:01:02:03:04:05` with the tunnel endpoint `10.1.1.2` of the tunnel `vxlan0-tunnel`.

```
$ curl -u admin:f5site02 -H "Content-Type:=application/json" -k -X PUT
'https://172.30.69.69/mgmt/tm/net/fdb/tunnel/~Common~vxlan0-tunnel' -d
'{"kind":"tm:net:fdb:tunnel:tunnelstate","name":"vxlan0-tunnel","partition":"Common",
"fullPath":"/Common/vxlan0-tunnel","generation":1,
"selfLink":"https://localhost/mgmt/tm/net/fdb/tunnel/~Common~vxlan0-tunnel?
ver=11.5.0","records":[{"name":"00:01:02:03:04:05",
"endpoint":"10.1.1.2"}]}'  |python -m json.tool
{
    "fullPath": "/Common/vxlan0-tunnel",
    "generation": 1,
    "kind": "tm:net:fdb:tunnel:tunnelstate",
    "name": "vxlan0-tunnel",
    "partition": "Common",
    "records": [
        {
            "endpoint": "10.1.1.2",
            "name": "00:01:02:03:04:05"
        }
    ],
    "selfLink":
"https://localhost/mgmt/tm/net/fdb/tunnel/~Common~vxlan0-tunnel?ver=11.5.0"
}
```

Sample NVGRE configuration using tmsh

This listing example illustrates the steps for creating a routing configuration that includes an NVGRE tunnel on the BIG-IP® system. F5 Networks provides an API for you to configure the F5 SCVMM Gateway Provider plug-in to build and manage NVGRE tunnels.

```
create net vlan wan {
    interfaces add { 1.1 }
    mtu 1550
}
create net self 10.1.1.1/24 {
    address 10.1.1.1/24
    vlan wan
}
create net tunnels gre nvgre {
    encapsulation nvgre
}
create net tunnels tunnel nvgre5000 {
    local-address 10.1.1.1
    remote-address any
    profile nvgre
    key 5000
}
create net vlan legacy5000 {
    interfaces add { 2.1 }
}
```

```
create net route-domain 5000 {
    id 5000
    vlans add { nvgre5000 legacy5000 }
}
create net self 10.3.3.1%5000/24 {
    address 10.3.3.1%5000/24
    vlan nvgre5000
}
create net self 10.4.4.1%5000/24 {
    address 10.4.4.1%5000/24
    vlan legacy5000
}
create net route 10.5.5.0%5000/24 {
    network 10.5.5.0%5000/24
    gw 10.3.3.2%5000
}
create net route 10.6.6.0%5000/24 {
    network 10.6.6.0%5000/24
    gw 10.3.3.3%5000
}
modify net fdb tunnel nvgre5000 {
    records add {
        00:FF:0A:03:03:02 { endpoint 10.1.2.1 }
        00:FF:0A:03:03:03 { endpoint 10.1.3.1 }
    }
}
create net arp 10.3.3.2%5000 {
    mac-address 00:FF:0A:03:03:02
}
create net arp 10.3.3.3%5000 {
    mac-address 00:FF:0A:03:03:03
}
```

Sample VXLAN unicast configuration using tmsh

This example listing illustrates the steps for creating a routing configuration that includes a VXLAN tunnel on the BIG-IP® system. This configuration adds the tunnel to a route domain. You can use the iControl/REST API to configure a network controller to build and manage VXLAN (unicast) tunnels.

```
create net vlan wan {
    interfaces add { 1.1 }
    mtu 1550
}
create net self 10.1.1./24 {
    address 10.1.1.1/24
    vlan wan
}
create net tunnels vxlan vxlan-static {
    flooding-type none
}
create net tunnels tunnel vxlan5000 {
    local-address 10.1.1.1
    remote-address any
    profile vxlan-static
    key 5000
}
create net vlan legacy5000 {
    interfaces add { 2.1 }
}
create net self 10.3.3.1%5000/24 {
    address 10.3.3.1%5000/24
    vlan vxlan5000
}
create net self 10.4.4.1%5000/24 {
```

```
    address 10.4.4.1%5000/24
    vlan legacy5000
}
create net route 10.5.5.0%5000/24 {
    network 10.5.5.0%5000/24
    gw 10.3.3.2%5000
}
create net route 10.6.6.0%5000/24 {
    network 10.6.6.0%5000/24
    gw 10.3.3.3%5000
}
modify net fdb tunnel vxlan5000 {
    records add {
        00:FF:0A:03:03:02 { endpoint 10.1.2.1 }
        00:FF:0A:03:03:03 { endpoint 10.1.3.1 }
    }
}
create net arp 10.3.3.2%5000 {
    mac-address 00:FF:0A:03:03:02
}
create net arp 10.3.3.3%5000 {
    mac-address 00:FF:0A:03:03:03
}
}
```

Sample command for virtual server to listen on a VXLAN tunnel

An alternative for including a network virtualization tunnel in a routing configuration is to create a virtual server that listens for the tunnel traffic, such as in the following example.

```
# tmsh create ltm virtual http_virtual destination 10.3.3.15%5000:http
ip-protocol tcp vlans add { vxlan5000 }
```

The code in this example creates a virtual server `http_virtual` that listens for traffic destined for the IP address `10.3.3.15`on the tunnel named `vxlan5000`.

Commands for viewing tunnel statistics

You can use the `tmsh` command-line utility to view tunnel statistics, listing either all the tunnels on the BIG-IP® system or statistics about a particular tunnel.

View per-tunnel statistics:

```
# tmsh show net tunnels tunnel
```

View static and dynamic FDB entries:

```
# tmsh show net fdb tunnel
```

About VXLAN multicast configuration

In a VMware vSphere 5.1 environment, you can configure VXLAN without knowing all the remote tunnel endpoints. The BIG-IP® system uses multicast flooding to learn unknown and broadcast frames. VXLAN can extend the virtual network across a set of hypervisors, providing L2 connectivity among the hosted virtual machines (VMs). Each hypervisor represents a VXLAN tunnel endpoint (VTEP). In this environment, you can configure a BIG-IP system as an L2 VXLAN gateway device to terminate the VXLAN tunnel and forward traffic to and from a physical network.

About bridging VLAN and VXLAN networks

You can configure Virtual eXtended LAN (VXLAN) on a BIG-IP® system to enable a physical VLAN to communicate with virtual machines (VMs) in a virtual network.

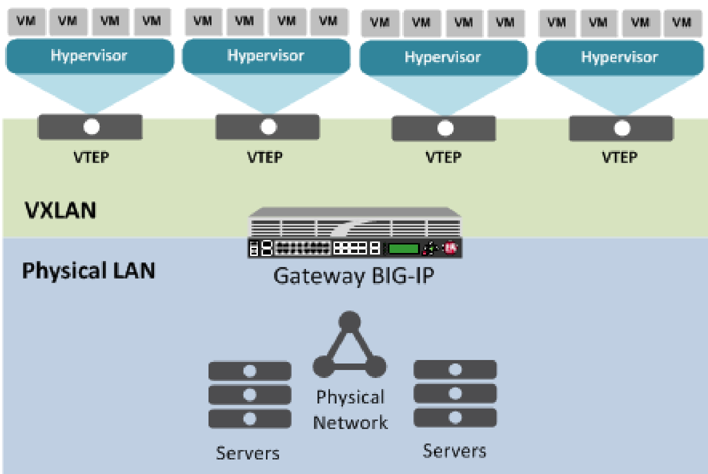


Figure 13: The VXLAN gateway

When you configure a BIG-IP system as an L2 VXLAN gateway, the BIG-IP system joins the configured multicast group, and can forward both unicast and multicast or broadcast frames on the virtual network. The BIG-IP system learns about MAC address and VTEP associations dynamically, thus avoiding unnecessary transmission of multicast traffic.

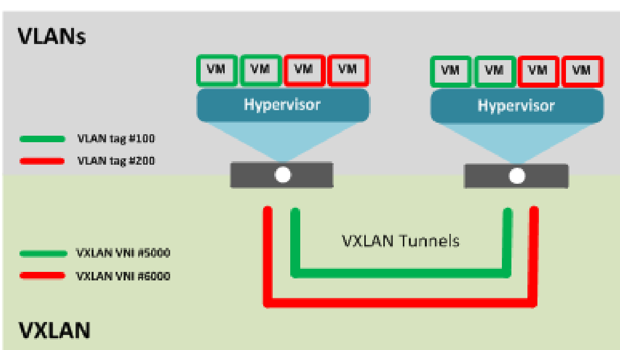


Figure 14: Multiple VXLAN tunnels

Considerations for configuring VXLAN multicast

As you configure VXLAN on a BIG-IP® system, keep these considerations in mind.

- If you configure the BIG-IP device as a bridge between physical VLANs and a VXLAN tunnel, the number of virtualized network segments in the overlay is limited to the maximum number of physical VLANs (4094). This limitation does not apply to Layer 3 configurations.
- You need to configure a separate tunnel for each VNI. The tunnels can have the same local and remote endpoint addresses.
- For the Layer 2 network, you must ensure a loop-free topology.
- Do not modify the configuration of a VXLAN tunnel after it is created. Instead, delete the existing tunnel and create a new one.

Task summary

Before you configure VXLAN, ensure that these conditions are met:

- The BIG-IP® system must be licensed for SDN Services.
- Network connectivity exists between the BIG-IP system and the hypervisors.
- If you have over 2000 connections, the **Management (MGMT)** setting on the Resource Provisioning screen is set to **Large** (**System** > **Resource Provisioning**).

Task list

Creating a VXLAN multicast tunnel

Creating a VXLAN multicast tunnel on a BIG-IP® system provides an L2 VXLAN gateway to connect the physical network with a virtualized network.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select **vxlan**.

   This setting tells the system which tunnel profile to use. The system-supplied VXLAN profile specifies port `4789`. To change the port number, you can create a new VXLAN profile, which then appears in this list.
4. In the **Local Address** field, type the self IP address of the VLAN through which the remote hypervisor is reachable.
5. In the **Remote Address** field, type the multicast group address associated with the VXLAN segment.
6. In the **Key** field, type the VNI (Virtual Network Identifier) to use for the VXLAN tunnel.
7. Click **Finished**.

Creating a bridge between VXLAN and non-VXLAN networks

Before you begin this task, verify that a VXLAN multicast tunnel exists on the BIG-IP® system.

You can create a VLAN group to bridge the traffic between a VXLAN overlay network (Layer 3) and a non-VXLAN (Layer 2) network.

1. On the Main tab, click **Network** > **VLANs** > **VLAN Groups**.
   The VLAN Groups list screen opens.
2. Click **Create**.
   The New VLAN Group screen opens.
3. In then **Name** field, type a unique name for the VLAN group.
4. For the **VLANs** setting, select the VLAN that connects to the non-VXLAN Layer-2 network and the VXLAN tunnel you created, and using the Move button, move your selections from the **Available** list to the **Members** list.
5. Click **Finished**.

Web Hosting Multiple Customers Using an External Switch

Overview: Web hosting multiple customers using an external switch

You can use the BIG-IP® system to provide hosting services, including application delivery, for multiple customers.

To host multiple web customers, you can incorporate an external switch into the configurations. In this illustration, the BIG-IP system has an interface (5.1) assigned to three VLANs on a network. The three VLANs are **vlanA**, **vlanB**, and **vlanB**. Interface **5.1** processes traffic for all three VLANs. Note that each VLAN contains two servers, and serves a specific customer.

Tip: *An alternate way to implement web hosting for multiple customers is to use the route domains feature.*

Illustration for hosting multiple customers using an external switch

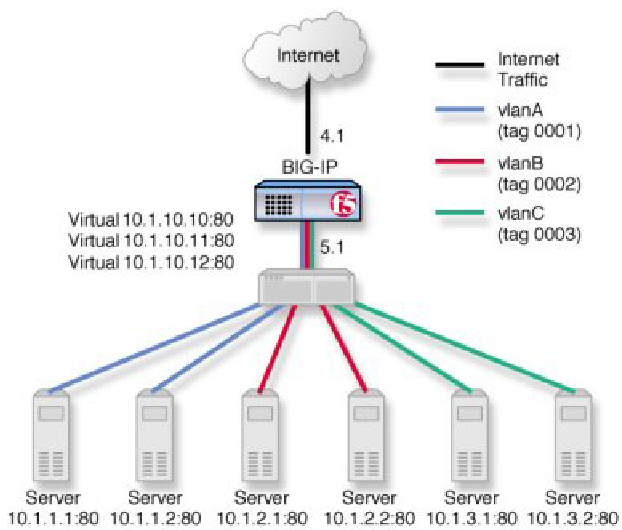


Figure 15: Hosting multiple customers using an external switch

Task summary for hosting multiple customers

Perform these tasks to host multiple customers using an external switch.

Task list

Creating a VLAN with a tagged interface

Creating a load balancing pool
Creating a virtual server for HTTP traffic

Creating a VLAN with a tagged interface

When you create a VLAN with tagged interfaces, each of the specified interfaces can process traffic destined for that VLAN.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type a unique name for the VLAN.
4. In the **Tag** field, type a numeric tag, from 1-4094, for the VLAN, or leave the field blank if you want the BIG-IP system to automatically assign a VLAN tag.

   The VLAN tag identifies the traffic from hosts in the associated VLAN.
5. For the **Interfaces** setting, click an interface number or trunk name from the **Available** list, and use the Move button to add the selected interface or trunk to the **Tagged** list. Repeat this step as necessary.

   You can use the same interface for other VLANs later, if you always assign the interface as a tagged interface.
6. If you want the system to verify that the return route to an initial packet is the same VLAN from which the packet originated, select the **Source Check** check box.
7. In the **MTU** field, retain the default number of bytes (**1500**).
8. From the **Configuration** list, select **Advanced**.
9. If you want to base redundant-system failover on VLAN-related events, select the **Fail-safe** box.
10. From the **Auto Last Hop** list, select a value.
11. From the **CMP Hash** list, select a value.
12. To enable the **DAG Round Robin** setting, select the check box.
13. Click **Finished**.
    The screen refreshes, and displays the new VLAN from the list.

The new VLAN appears in the VLAN list.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

---

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type a port number in the **Service Port** field, or select a service name from the list.
   c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
   d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for HTTP traffic

This task creates a destination IP address for application traffic. As part of this task, you must assign the relevant pool to the virtual server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type `80`, or select **HTTP** from the list.
6. From the **HTTP Profile** list, select **http**.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. Click **Finished**.

You now have a virtual server to use as a destination address for application traffic.

Web Hosting Multiple Customers Using Untagged Interfaces

Overview: Web hosting multiple customers using untagged interfaces

One way to implement web hosting for multiple customers is to use multiple interfaces on the BIG-IP® system to directly host traffic for multiple customers, without the need for an external switch. With this scenario, you must configure the VLANs with untagged instead of tagged interfaces. As shown in the following illustration, two BIG-IP system interfaces are assigned to each VLAN. For example, interfaces **1.1** and **1.2** are assigned to VLAN **vlanA**. Each interface is assigned to a VLAN as an untagged interface.

Tip: *An alternate way to implement web hosting for multiple customers is to use the route domains feature.*

Illustration for hosting multiple customers using untagged interfaces

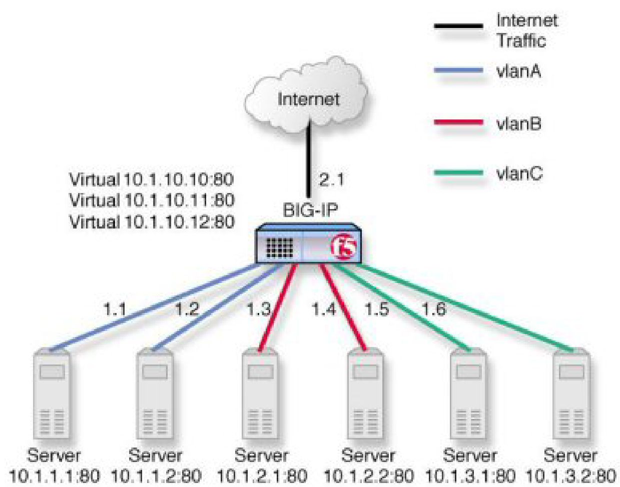


Figure 16: Hosting multiple customers using untagged interfaces

Task summary for hosting multiple customers

Perform these tasks to host multiple customers using tagged interfaces on VLANs.

Task list

Creating a VLAN with an untagged interface
Creating a load balancing pool
Creating a virtual server for HTTP traffic

Creating a VLAN with an untagged interface

You can create a VLAN that uses untagged interfaces.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type a unique name for the VLAN.
4. In the **Tag** field, type a numeric tag, from 1-4094, for the VLAN, or leave the field blank if you want the BIG-IP system to automatically assign a VLAN tag.

   The VLAN tag identifies the traffic from hosts in the associated VLAN.
5. For the **Interfaces** setting, from the **Available** list, click an interface number or trunk name and add the selected interface or trunk to the **Untagged** list. Repeat this step as necessary.
6. Click **Finished**.
   The screen refreshes, and displays the new VLAN from the list.

The interfaces that you specified in this task process traffic for this VLAN only.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

   Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type a port number in the **Service Port** field, or select a service name from the list.
   c) To specify a priority group, type a priority number in the **Priority Group Activation** field.

d) Click **Add**.

8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for HTTP traffic

This task creates a destination IP address for application traffic. As part of this task, you must assign the relevant pool to the virtual server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type `80`, or select **HTTP** from the list.
6. From the **HTTP Profile** list, select **http**.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. Click **Finished**.

You now have a virtual server to use as a destination address for application traffic.

Web Hosting Multiple Customers Using Route Domains

Overview: Use of route domains to host multiple web customers on the BIG-IP system

Using the *route domains* feature of the BIG-IP® system, you can provide hosting service for multiple customers by isolating each type of application traffic within a defined address space on the network. This enhances security and dedicates BIG-IP resources to each application.

Using route domains, you can also use duplicate IP addresses on the network, provided that each of the duplicate addresses resides in a separate route domain and is isolated on the network through a separate VLAN. For example, if you are processing traffic for two different customers, you can create two separate route domains. The same node address (such as `10.0.10.1`) can reside in each route domain, in the same pool or in different pools, and you can assign a different monitor to each of the two corresponding pool members.

A good example of the use of traffic isolation on a network is an ISP that services multiple customers, where each customer deploys a different application. The first illustration shows two route domain objects on a BIG-IP system, where each route domain corresponds to a separate customer, and thus, resides in its own partition. Within each partition, the ISP created the network objects and local traffic objects required for that customer's application (AppA or AppB).

The sample configuration results in the BIG-IP system segmenting traffic for two different applications into two separate route domains. The routes for each application's traffic cannot cross route domain boundaries because cross-routing restrictions are enabled on the BIG-IP system by default. The second illustration shows the resulting route isolation for AppA and AppB application traffic.

Illustration of sample BIG-IP configuration using route domains

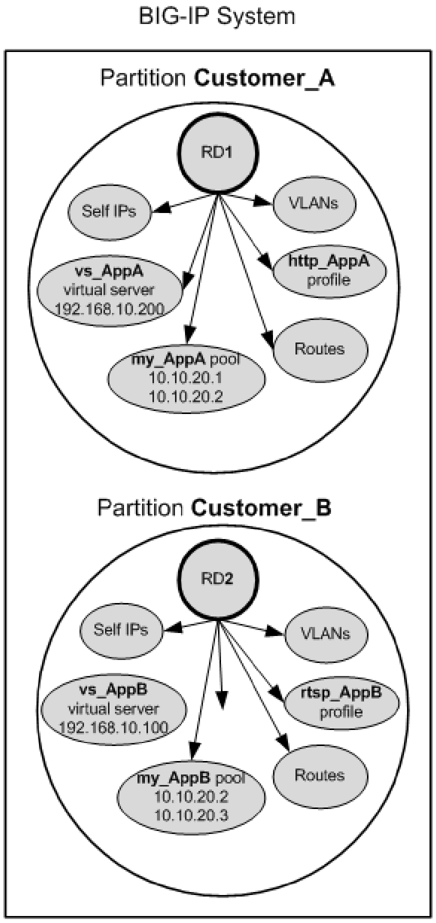


Figure 17: Sample BIG-IP configuration using route domains

Illustration of resulting route domain configuration

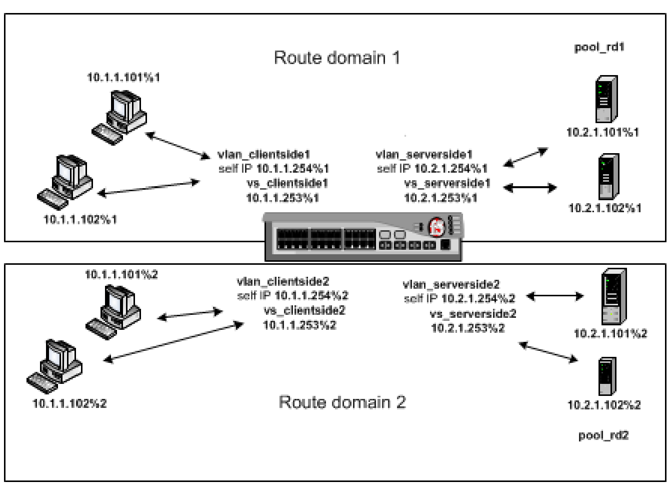


Figure 18: Resulting route domain configuration

Task summary

Perform these tasks to host multiple web customers using route domains.

Task list

Creating an administrative partition
Creating a VLAN with a tagged interface
Creating a self IP address for a default route domain in an administrative partition
Creating a route domain on the BIG-IP system
Creating a load balancing pool
Creating a virtual server
Configuring route advertisement for a virtual address
Adding routes that specify VLAN internal as the resource

Creating an administrative partition

An administrative partition creates an access control boundary for users and applications.

1. On the Main tab, expand **System** and click **Users**.
   The Users List screen opens.
2. On the menu bar, click **Partition List**.
3. Click **Create**.
   The New Partition screen opens.
4. In the **Name** field, type a unique name for the partition.
   An example of a partition name is `app1_partition`.
5. Type a description of the partition in the **Description** field.

   This field is optional.
6. For the **Device Group** setting, choose an action:

Action	Result
Retain the default value.	Choose this option if you want the folder corresponding to this partition to inherit the value of the device group attribute from folder `root`.
Clear the check box and select the name of a device group.	Choose this option if you do not want the folder corresponding to this partition to inherit the value of the device group attribute from folder `root`.

7. For the **Traffic Group** setting, choose an action:

Action	Result
Retain the default value.	Choose this option if you want the folder corresponding to this partition to inherit the value of the traffic group attribute from folder `root`.
Clear the check box and select the name of a traffic group.	Choose this option if you do not want the folder corresponding to this partition to inherit the value of the traffic group attribute from folder `root`.

8. Click **Finished**.

The new partition appears in the partition list.

Creating a VLAN with a tagged interface

When you create a VLAN with tagged interfaces, each of the specified interfaces can process traffic destined for that VLAN.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type a unique name for the VLAN.
4. In the **Tag** field, type a numeric tag, from 1-4094, for the VLAN, or leave the field blank if you want the BIG-IP system to automatically assign a VLAN tag.

   The VLAN tag identifies the traffic from hosts in the associated VLAN.
5. For the **Interfaces** setting, click an interface number or trunk name from the **Available** list, and use the Move button to add the selected interface or trunk to the **Tagged** list. Repeat this step as necessary.

   You can use the same interface for other VLANs later, if you always assign the interface as a tagged interface.
6. If you want the system to verify that the return route to an initial packet is the same VLAN from which the packet originated, select the **Source Check** check box.
7. In the **MTU** field, retain the default number of bytes (**1500**).
8. From the **Configuration** list, select **Advanced**.
9. If you want to base redundant-system failover on VLAN-related events, select the **Fail-safe** box.
10. From the **Auto Last Hop** list, select a value.
11. From the **CMP Hash** list, select a value.
12. To enable the **DAG Round Robin** setting, select the check box.
13. Click **Finished**.
    The screen refreshes, and displays the new VLAN from the list.

The new VLAN appears in the VLAN list.

Creating a self IP address for a default route domain in an administrative partition

Before creating a self IP address, ensure that you have created an internal VLAN and an external VLAN on the BIG-IP system.

Using this procedure, you must create two self IP addresses on the BIG-IP system. One self IP address is associated with the internal VLAN, and the other is associated with the external VLAN. Self IP addresses enable the BIG-IP system and other devices on the network to route application traffic through the associated VLAN.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **IP Address** field, type an IP address.

This IP address should represent the address space of the VLAN that you specify with the **VLAN** setting. Because the route domain that you previously created is the default route domain for the administrative partition, you do not need to append the route domain ID to this IP address.

The system accepts IP addresses in both the IPv4 and IPv6 formats.

4. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
5. From the **VLAN/Tunnel** list, select the VLAN to associate with this self IP address.
   - On the internal network, select the internal or high availability VLAN that is associated with an internal interface or trunk.
   - On the external network, select the external VLAN that is associated with an external interface or trunk.
6. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

The BIG-IP system has a self IP address that is associated with the internal or external network.

Creating a route domain on the BIG-IP system

Before you create a route domain:

- Ensure that an external and an internal VLAN exist on the BIG-IP® system.
- If you intend to assign a static bandwidth controller policy to the route domain, you must first create the policy. You can do this using the BIG-IP Configuration utility.
- Verify that you have set the current partition on the system to the partition in which you want the route domain to reside.

You can create a route domain on BIG-IP system to segment (isolate) traffic on your network. Route domains are useful for multi-tenant configurations.

1. On the Main tab, click **Network** > **Route Domains**.
   The Route Domain List screen opens.
2. Click **Create**.
   The New Route Domain screen opens.
3. In the **Name** field, type a name for the route domain.

   This name must be unique within the administrative partition in which the route domain resides.
4. In the **ID** field, type an ID number for the route domain.

   This ID must be unique on the BIG-IP system; that is, no other route domain on the system can have this ID.
5. In the **Description** field, type a description of the route domain.
   For example: `This route domain applies to traffic for application MyApp.`
6. For the **Strict Isolation** setting, select the **Enabled** check box to restrict traffic in this route domain from crossing into another route domain.
7. For the **Parent Name** setting, retain the default value.
8. For the **VLANs** setting, from the **Available** list, select a VLAN name and move it to the **Members** list.

   Select the VLAN that processes the application traffic relevant to this route domain.

   Configuring this setting ensures that the BIG-IP system immediately associates any self IP addresses pertaining to the selected VLANs with this route domain.

9. For the **Dynamic Routing Protocols** setting, from the **Available** list, select one or more protocol names and move them to the **Enabled** list.

   You can enable any number of listed protocols for this route domain. This setting is optional.
10. From the **Bandwidth Controller** list, select a static bandwidth control policy to enforce a throughput limit on traffic for this route domain.
11. From the **Partition Default Route Domain** list, select either **Another route domain (0) is the Partition Default Route Domain** or **Make this route domain the Partition Default Route Domain**.

    This setting does not appear if the current administrative partition is partition `Common`.

    When you configure this setting, either route domain `0` or this route domain becomes the default route domain for the current administrative partition.
12. Click **Finished**.
    The system displays a list of route domains on the BIG-IP system.

You now have another route domain on the BIG-IP system.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

   Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type a port number in the **Service Port** field, or select a service name from the list.
   c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
   d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server

A virtual server represents a destination IP address for application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. In the Resources area of the screen, from the **Default Pool** list, select a pool name.

The web customer now has a destination IP address on the BIG-IP system for application traffic.

Configuring route advertisement for a virtual address

Before configuring route advertisement on a virtual address, verify that you have enabled one or more dynamic routing protocols on the route domain pertaining to this virtual address. Also verify that you have configured the relevant dynamic routing protocols for route redistribution.

Perform this task to advertise a route for this virtual address to other routers on your network.

Important: *This task pertains only to configurations for which you have enabled dynamic routing protocols on the relevant route domain. If you have not enabled dynamic routing protocols on the relevant route domain, you can skip this task.*

---

1. On the Main tab, click **Local Traffic** > **Virtual Servers** > **Virtual Address List**.
   The Virtual Address List screen opens.
2. In the Name column, click the virtual address for which you want to advertise a route.
   This displays the properties of that virtual address.
3. Verify that the **ARP** field is selected.
4. From the **Advertise Route** list, choose one of these options:

Option	Description
When any virtual server is available	Specifies that the system advertises a route for this virtual IP address whenever any virtual server associated with this virtual IP address is available.
When all virtual servers(s) are available	Specifies that the system advertises a route for this virtual IP address whenever all virtual servers associated with this virtual IP address is available.
Always	Specifies that the system always advertises a route for this virtual IP address.

5. For the **Route Advertisement** setting, select the box.

   This makes it possible for the BIG-IP system to advertise this virtual IP address when you have enabled any dynamic routing protocols.

6. Click **Update**.
7. Repeat this task for each virtual address for which you want to advertise a route.

The BIG-IP system advertises a route for this virtual address to other routers when one or more dynamic routing protocols are enabled and are configured for route redistribution.

Adding routes that specify VLAN internal as the resource

Ensure that you set the current administrative partition to the partition in which you want a specific customer's configuration to reside.

You must add a route for each destination IP address pertaining to the route domain. A destination address in this case is typically a node address for a pool member.

1. On the Main tab, click **Network** > **Routes**.
2. Click **Add**.
   The New Route screen opens.
3. In the **Name** field, type a unique user name.

   This name can be any combination of alphanumeric characters, including an IP address.
4. In the **Destination** field, type either the destination IP address for the route, or IP address `0.0.0.0` for the default route.

   This address can represent either a host or a network. Also, if you are using the route domains and the relevant route domain is the partition default route domain, you do not need to append a route domain ID to this address.
5. In the **Netmask** field, type the network mask for the destination IP address.
6. From the **Resource** list, select **Use VLAN/Tunnel**.

   A VLAN represents the VLAN through which the packets flow to reach the specified destination.
7. From the **VLAN** list, select **Internal**.
8. At the bottom of the screen, click **Finished**.

The BIG-IP system now includes routes to the nodes in the load balancing pool for a specific route domain.

Implementing the Link Layer Discovery Protocol

Overview: Implementing Link Layer Discovery Protocol

The BIG-IP® system supports Link Layer Discovery Protocol (LLDP). LLDP is a Layer 2 industry-standard protocol (IEEE 802.1AB) that gives a network device such as the BIG-IP system the ability to advertise its identity and capabilities to multi-vendor neighbor devices on a network. The protocol also enables a network device to receive information from neighbor devices.

LLDP transmits device information in the form of LLDP messages known as LLDP Packet Data Units (LLDPDUs).

In general, this protocol:

- Advertises connectivity and management information about the local BIG-IP device to neighbor devices on the same IEEE 802 LAN.
- Receives network management information from neighbor devices on the same IEEE 802 LAN.
- Operates with all IEEE 802 access protocols and network media.

Using the BIG-IP Configuration utility or `tmsh`, you can use this particular implementation to configure BIG-IP system interfaces to transmit LLDPDUs to neighbor devices. More specifically, you can:

- Specify the exact content of LLDPDUs that a BIG-IP system interface transmits to a neighbor device. You specify this content by configuring the **LLDP Attributes** setting on each individual interface.
- Globally specify the frequencies of various message transmittal properties, and specify the number of neighbors from which interfaces can receive messages. These properties apply to all interfaces on the BIG-IP system.

The following illustration shows a BIG-IP system that transmits LLDP messages to three neighbor devices: another BIG-IP system, an external switch, and an external router. Note that LLDP is enabled on all of the devices.

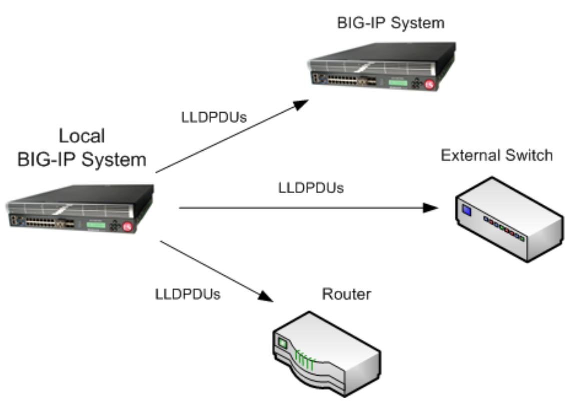


Figure 19: The BIG-IP system and LLDP transmittal

Task summary

Perform these tasks to implement Link Layer Discovery Protocol (LLDP) on selected BIG-IP system interfaces.

Task list

Configuring global LLDP properties

You can configure a set of general LLDP properties that apply to all interfaces on the BIG-IP system. These settings pertain to LLDP message transmission frequencies and the maximum number of neighbors to which each interface can send LLDP messages.

Note: *Although you use this procedure to globally enable the LLDP feature on the BIG-IP system, you can also disable LLDP for any individual interface. You do this by configuring the specific properties of that interface.*

---

1. On the Main tab, click **Network** > **Interfaces** > **LLDP** > **General**.
   This displays the general LLDP properties that you can configure on the system.
2. From the **LLDP** list, select **Enabled**.
3. For the remainder of the settings, retain or change the default values.
4. Click **Update**.

This task activates support for the LLDP protocol on the BIG-IP system, and configures the system to transmit LLDPDUs according to the specified frequencies.

Configuring LLDP settings for an individual interface

You can use this procedure to configure the settings for an individual interface on the BIG-IP system.

1. On the Main tab, click **Network** > **Interfaces** > **Interface List**.
   The Interface List screen displays the list of interfaces on the system.
2. In the Name column, click an interface number.
   This displays the properties of the interface.
3. For the **State** setting, verify that the interface is set to **Enabled**.
4. For the **LLDP** setting, verify that the property is set to **Transmit Only**.
5. For the **LLDP Attributes** setting, verify that the list of attributes in the **Send** field includes all Time Length Values (TLVs) that you want the BIG-IP system interface to send to neighbor devices.
6. Click **Update**.

After you perform this task, the interface is configured to send the specified LLDP information to neighbor devices.

Implementation result

This implementation results in this LLDP configuration:

- Support for the LLDP protocol is enabled on the BIG-IP system.
- For all BIG-IP system interfaces, the BIG-IP system attempts to transmit LLDPDUs to neighbor devices every 30 seconds, with a minimum delay between transmissions of 2 seconds.
- The maximum number of neighbors to which each BIG-IP system interface can send LLDPDUs is 10.
- Every BIG-IP system interface can send LLDPDUs to its neighbors.
- No BIG-IP system interface can receive LLDPDUs from its neighbors.

In addition, the content of the LLDPDUs that each BIG-IP system interface sends to its neighbors contains this information:

- Chassis ID
- Port ID
- Time-to-Live value
- Port description
- System name
- System description
- System capabilities
- Port VLAN ID
- Port and protocol VLAN ID
- VLAN name
- Protocol identity
- MAC/PHY config status
- Link aggregation
- Max frame size
- Product model

Configuring an EtherIP Tunnel

Overview: Preserving BIG-IP connections during live virtual machine migration

In some network configurations, the BIG-IP® system is configured to send application traffic to destination servers that are implemented as VMware® virtual machines (VMs). These VMs can undergo live migration, using VMware vMotion™ and an iSession™ tunnel, across a wide area network (WAN) to a host in another data center.

To preserve any existing connections between the BIG-IP system and a virtual machine while the virtual machine migrates to another data center, you can create an EtherIP tunnel.

An *EtherIP tunnel* is an object that you create on each of two BIG-IP systems that sit on either side of a WAN. The EtherIP tunnel uses the industry-standard EtherIP protocol to tunnel Ethernet and IEEE 802.3 media access control (MAC) frames across an IP network. The two EtherIP tunnel objects together form a tunnel that logically connects two data centers. When the application traffic that flows between one of the BIG-IP systems and the VM is routed through the EtherIP tunnel, connections are preserved during and after the VM migration.

After you have configured the BIG-IP system to preserve connections to migrating VMs, you can create a Virtual Location monitor for the pool. A *Virtual Location* monitor ensures that the BIG-IP system sends connections to a local pool member rather than a remote pool one, when some of the pool members have migrated to a remote data center.

Tip: *The BIG-IP system that is located on each end of an EtherIP tunnel can be part of a redundant system configuration. Make sure that both units of any redundant system configuration reside on the same side of the tunnel.*

Illustration of EtherIP tunneling in a vMotion environment

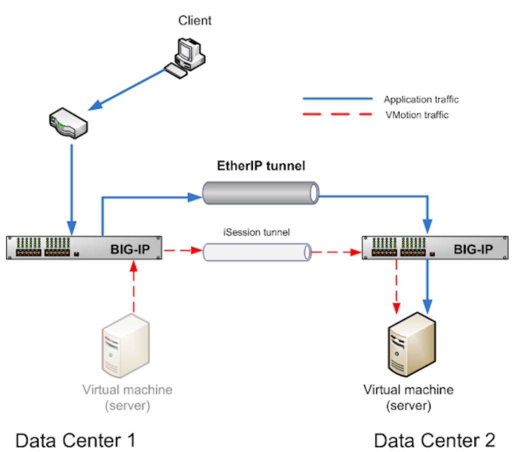


Figure 20: EtherIP tunneling in a vMotion environment

Task summary

Implement an EtherIP tunneling configuration to prevent the BIG-IP® system from dropping existing connections to migrating virtual machines in a vMotion environment. To set up this configuration, you must verify a few prerequisite tasks, as well as create some configuration objects on the BIG-IP system.

Important: *Perform these tasks on the BIG-IP system in both the local data center and the remote data center.*

Before you begin configuring EtherIP tunneling, verify that these BIG-IP objects and module exist on the BIG-IP system:

An iSession profile
This profile creates an iSession tunnel to optimize the live migration of virtual machine servers from one data center to another.

A load balancing pool
This pool represents a collection of virtual machines on a host server in the data center.

A standard TCP or UDP virtual server
This virtual server load balances application traffic and optimizes vMotion traffic. This virtual server must reference the iSession profile and the load balancing pool.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP Global Traffic Manager™
This module directs traffic to the correct BIG-IP® Local Traffic Manager™ virtual server.

Task List

Creating a VLAN
Creating an EtherIP profile
Creating an EtherIP tunnel object
Creating a VLAN group
Creating a self IP address for a VLAN
Creating a self IP for a VLAN group
Creating a Virtual Location monitor
Syncing the BIG-IP configuration to the device group

Creating a VLAN

VLANs represent a collection of hosts that can share network resources, regardless of their physical location on the network. You create a VLAN to associate physical interfaces with that VLAN.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type a unique name for the VLAN.

4. In the **Tag** field, type a numeric tag, from 1-4094, for the VLAN, or leave the field blank if you want the BIG-IP system to automatically assign a VLAN tag.

   The VLAN tag identifies the traffic from hosts in the associated VLAN.
5. For the **Interfaces** setting, from the **Available** list, click an interface number or trunk name and add the selected interface or trunk to the **Untagged** list. Repeat this step as necessary.
6. If you want the system to verify that the return route to an initial packet is the same VLAN from which the packet originated, select the **Source Check** check box.
7. In the **MTU** field, retain the default number of bytes (**1500**).
8. From the **Configuration** list, select **Advanced**.
9. If you want to base redundant-system failover on VLAN-related events, select the **Fail-safe** box.
10. From the **Auto Last Hop** list, select a value.
11. From the **CMP Hash** list, select a value.
12. To enable the **DAG Round Robin** setting, select the check box.
13. Click **Finished**.
    The screen refreshes, and displays the new VLAN from the list.

Creating an EtherIP profile

An EtherIP profile is a required component of an EtherIP tunnel in a vMotion™ environment. An *EtherIP profile* manages application traffic that traverses an EtherIP tunnel, for the purpose of preserving connections when a virtual machine is migrating to another data center. You must perform this task using the Traffic Management shell (tmsh), a command-line utility.

1. On the BIG-IP®system, start a console session.
2. Type a user name and password, and press Enter.
3. At the command prompt, type `tmsh`, and press Enter.
   This opens the Traffic Management shell (tmsh).
4. At the **tmsh** prompt, type `net tunnel`, and press Enter.
5. Type `create etherip` *`etherip_profile_name`*, and press Enter.
   This command creates an EtherIP profile, assigning all of the default values.
6. Type `save / sys config`, and press Enter.
7. To close the Traffic Management shell (tmsh), type `quit`, and press Enter.

You now have an EtherIP profile that you can specify when you create an EtherIP tunnel object.

Creating an EtherIP tunnel object

Before you perform this task, you must know the self IP address of the instance of the VLAN that exists, or will exist, on the BIG-IP® system in the other data center.

The purpose of an EtherIP tunnel that contains an EtherIP type of profile is to enable the BIG-IP system to preserve any current connections to a server that is migrating to another data center by way of vMotion™. You must perform this task using the Traffic Management shell (tmsh), a command-line utility.

1. On the BIG-IP system, start a console session.
2. Type a user name and password, and press Enter.
3. At the command prompt, type `tmsh` and press Enter.

This opens the Traffic Management shell (tmsh).

4. Type `net tunnels`, and press Enter.
5. Type the following command, and then press Enter:

   Note that the self IP addresses that you specify are those that you create for the VLAN on both the local and the remote BIG-IP system.

   ```
   create tunnel tunnel_name profile etherip local-address local_self_ip_address
   remote-address remote_self_ip_address
   ```
6. Type `save / sys config`, and press Enter.
7. To close the Traffic Management shell (tmsh), type `quit`, and press Enter.

The BIG-IP system configuration now includes a tunnel object.

Creating a VLAN group

VLAN groups consolidate Layer 2 traffic from two or more separate VLANs.

1. On the Main tab, click **Network** > **VLANs** > **VLAN Groups**.
   The VLAN Groups list screen opens.
2. Click **Create**.
   The New VLAN Group screen opens.
3. In the General Properties area, in the **VLAN Group** field, type a unique name for the VLAN group.
4. For the **VLANs** setting, move the VLANs that you want to include in the group from the **Available** list to the **Members** list.
5. From the **Transparency Mode** list, select a transparency mode, or retain the default setting, `Transparent`.

   The transparency mode determines the level of exposure of remote MAC addresses within the VLAN group traffic.

Mode	Purpose
Transparent	The MAC addresses of remote systems are exposed in Layer 2 traffic forwarding.
Translucent	Similar to `Transparent` mode, except the locally-unique bit is set in the MAC addresses of remote systems.
Opaque	The system uses proxy ARP with Layer 3 forwarding, so the MAC addresses of remote systems are not exposed.

6. Select the **Bridge All Traffic** check box if you want the VLAN group to forward all frames, including non-IP traffic.

   The default setting is disabled (not selected).
7. Leave the **Bridge in Standby** check box selected if you want the VLAN group to forward frames even when the system is the standby unit of a redundant system.
8. Click **Finished**.

Creating a self IP address for a VLAN

Before you create a self IP address, ensure that you have created at least one VLAN or VLAN group.

A self IP address enables the BIG-IP® system and other devices on the network to route application traffic through the associated VLAN or VLAN group.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type an IPv4 or IPv6 address.

   This IP address should represent the address space of the VLAN that you specify with the **VLAN/Tunnel** setting.
5. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
6. From the **VLAN/Tunnel** list, select the VLAN to associate with this self IP address.

   - On the internal network, select the internal or high availability VLAN that is associated with an internal interface or trunk.
   - On the external network, select the external VLAN that is associated with an external interface or trunk.
7. From the **Port Lockdown** list, select **Allow Default**.
8. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

The BIG-IP system can send and receive traffic through the specified VLAN or VLAN group.

Creating a self IP for a VLAN group

Before you create a self IP address, ensure that you have created at least one VLAN or VLAN group.

After you have created the VLAN group, create a self IP address for the VLAN group. The self IP address for the VLAN group provides a route for packets destined for the network. With the BIG-IP® system, the path to an IP network is a VLAN. However, with the VLAN group feature used in this procedure, the path to the IP network `10.0.0.0` is actually through more than one VLAN. As IP routers are designed to have only one physical route to a network, a routing conflict can occur. With a self IP address on the BIG-IP system, you can resolve the routing conflict by associating a self IP address with the VLAN group.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **IP Address** field, type an IPv4 or IPv6 address.

   This IP address should represent the address space of the VLAN group that you specify with the **VLAN/Tunnel** setting.
4. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
5. From the **VLAN/Tunnel** list, select the VLAN group with which to associate this self IP address.
6. From the **Port Lockdown** list, select **Allow Default**.
7. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

The BIG-IP system can send and receive traffic through the specified VLAN or VLAN group.

Creating a Virtual Location monitor

When the BIG-IP® system is directing application traffic to pool members that are implemented as virtual machines, you should configure a Virtual Location type of monitor on the BIG-IP system. A *Virtual Location* monitor determines if a pool member is local to the data center or remote, and assigns a priority group to the pool member accordingly. The monitor assigns remote pool members a lower priority than local members, thus ensuring that the BIG-IP directs application requests to local pool members whenever possible.

1. On the Main tab, click **Local Traffic** > **Monitors**.
   The Monitor List screen opens.
2. Click **Create**.
   The New Monitor screen opens.
3. Type `my_virtual_location_monitor` in the **Name** field.
4. From the **Type** list, select **Virtual Location**.
5. From the **Configuration** list, select **Advanced**.
6. Retain the default value (in seconds) of `5` in the **Interval** field.
7. Retain the default value of `Disabled` in the **Up Interval** list.
8. Retain the default value (in seconds) of `0` in the **Time Until Up** field.
9. Retain the default value (in seconds) of `16` in the **Timeout** field.
10. Type the name of the pool that you created prior to configuring EtherIP tunneling in the **Pool Name** field.
11. Click **Finished**.

After configuring the Virtual Location monitor, the BIG-IP system assigns each member of the designated pool a priority group value to ensure that incoming connections are directed to a local pool member whenever possible.

F5 Networks recommends that you verify that BIG-IP® Global Traffic Manager™ (GTM™) has automatically assigned a BIG-IP type of monitor to BIG-IP® Local Traffic Manager™ (LTM®). A BIG-IP type of monitor can use the priority group assigned to each pool member to retrieve a `gtm_score` value.

Syncing the BIG-IP configuration to the device group

Before you sync the configuration, verify that the devices targeted for config sync are members of a device group and that device trust is established.

This task synchronizes the BIG-IP® configuration data from the local device to the devices in the device group. This synchronization ensures that devices in the device group operate properly. When synchronizing self IP addresses, the BIG-IP system synchronizes floating self IP addresses only.

Important: *You perform this task on either of the two devices, but not both.*

---

1. On the Main tab, click **Device Management** > **Overview**.
2. In the Device Groups area of the screen, in the Name column, select the name of the relevant device group.
   The screen expands to show a summary and details of the sync status of the selected device group, as well as a list of the individual devices within the device group.

3. In the Devices area of the screen, in the Sync Status column, select the device that shows a sync status of `Changes Pending`.
4. In the Sync Options area of the screen, select **Sync Device to Group**.
5. Click **Sync**.
   The BIG-IP system syncs the configuration data of the selected device in the Device area of the screen to the other members of the device group.

Except for non-floating self IP addresses, the entire set of BIG-IP configuration data is replicated on each device in the device group.

Implementation result

After you configure EtherIP tunneling on the BIG-IP system, you must perform the same configuration procedure on the BIG-IP system in the remote data center to fully establish the EtherIP tunnel.

After the tunnel is established, the BIG-IP system preserves any open connections to migrating (or migrated) virtual machine servers.

Creating IP Tunnels

About IP tunnels

Using F5® tunneling technologies, you can set up tunneling from devices on different Layer 2 networks, or scale multi-site data centers over Layer 3 pathways. When you know the IP address of the devices at both ends of the tunnel, you can create a point-to-point encapsulation tunnel between a BIG-IP® system and another device. When multiple devices feed into a BIG-IP system, you can create a tunnel by specifying only the IP address on the BIG-IP device.

The BIG-IP system provides the following tunneling types, available using the browser-based Configuration utility or the Traffic Management shell (`tmsh`) command-line utility, and iControl®.

- EtherIP
- FEC
- GRE
- IPIP
  - DS-Lite
  - IPv4IPv4
  - IPv4IPv6
  - IPv6IPv4
  - IPv6IPv6
- PPP
- VXLAN
- WCCPGRE

For information about deploying some of these tunneling types, consult additional F5 Networks documentation, including CGNAT (DS-Lite), acceleration (FEC), and TMOS (VXLAN). Licensing restrictions apply.

About point-to-point tunnels

Point-to-point IP encapsulation tunnels carry traffic through a routed network between known devices. For example, you can create a GRE tunnel to connect a BIG-IP® system to a remotely located pool member.

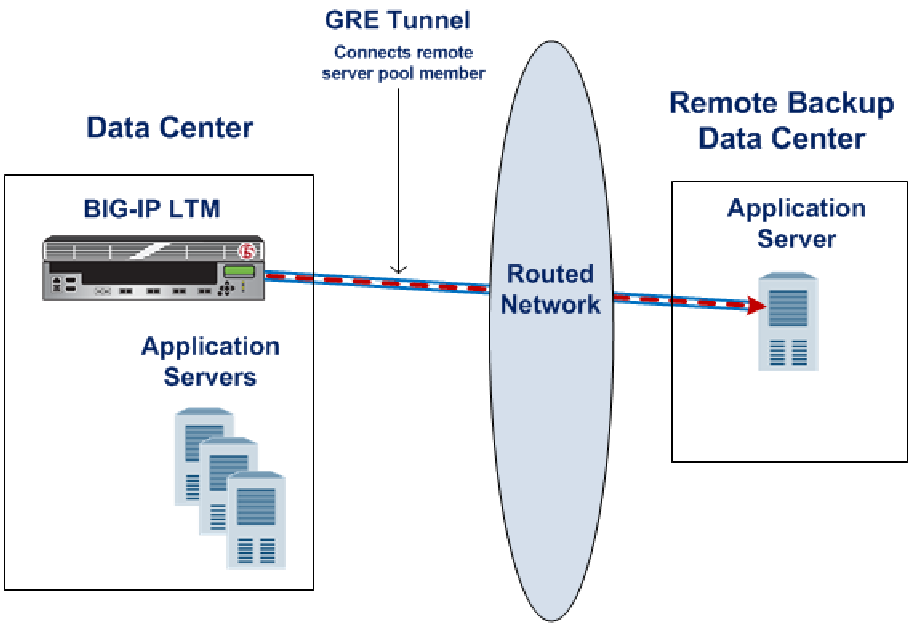


Figure 21: Illustration of a point-to-point GRE tunnel

Task summary

Creating a point-to-point IP tunnel
Assigning a self IP address to an IP tunnel endpoint
Routing traffic through an IP tunnel interface

Creating a point-to-point IP tunnel

To create a point-to-point tunnel, you specify the encapsulation protocol and the IP addresses of the devices at both ends of the tunnel.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select the type that corresponds to the encapsulation protocol you want to use.

   The selection **ipip** is the same as **ip4ip4**, but **ipip** is compatible with configurations from an earlier release.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, select **Specify**, and type the IP address of the device at the other end of the tunnel.
6. Click **Finished**.

After you complete this task, traffic is encapsulated using the protocol you specified between the BIG-IP system and the remote device you specified.

The BIG-IP®system requires that tunnels used as routes have a self IP address associated with the tunnel itself, distinct from the self IP address configured as a tunnel endpoint. After configuring a self IP address, you can configure routes that use the tunnel as a resource.

Assigning a self IP address to an IP tunnel endpoint

Ensure that you have created an IP tunnel before starting this task.

Self IP addresses can enable the BIG-IP® system, and other devices on the network, to route application traffic through the associated tunnel, similar to routing through VLANs and VLAN groups.

Note: *If the other side of the tunnel needs to be reachable, make sure the self IP addresses that you assign to both sides of the tunnel are in the same subnet.*

---

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type the IP address of the tunnel.

   The system accepts IPv4 and IPv6 addresses.

   Note: *This is not the same as the IP address of the tunnel local endpoint.*

   ---

5. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
6. From the **VLAN/Tunnel** list, select the tunnel with which to associate this self IP address.
7. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

Assigning a self IP to a tunnel ensures that the tunnel appears as a resource for routing traffic.

To direct traffic through the tunnel, add a route for which you specify the tunnel as the resource.

Routing traffic through an IP tunnel interface

Before starting this task, ensure that you have created an IP tunnel, and have assigned a self IP address to the tunnel.

You can route traffic through a tunnel interface, much like you use a VLAN or VLAN group.

1. On the Main tab, click **Network** > **Routes**.
2. Click **Add**.
   The New Route screen opens.
3. In the **Name** field, type a unique user name.

   This name can be any combination of alphanumeric characters, including an IP address.
4. In the **Destination** field, type the destination IP address for the route.
5. In the **Netmask** field, type the network mask for the destination IP address.
6. From the **Resource** list, select **Use VLAN/Tunnel**.
7. From the **VLAN/Tunnel** list, select a tunnel name.
8. At the bottom of the screen, click **Finished**.

The system now routes traffic destined for the IP address you specified through the tunnel you selected.

Example of a point-to-point IP tunnel configuration

This illustration is an example of a point-to-point IP tunnel configuration showing IP addresses. Note that the tunnel local endpoint address is different from the tunnel self IP address.

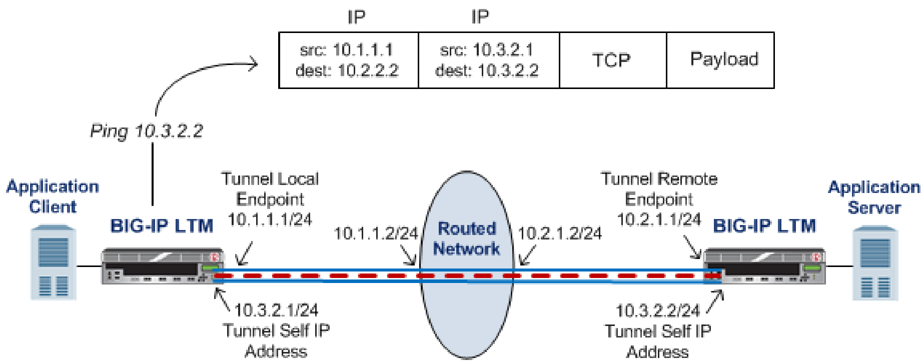


Figure 22: Illustration of a point-to-point IP tunnel configuration

About tunnels between the BIG-IP system and other devices

In a network that has multiple devices connected to a BIG-IP® system, you can create an IPIP or GRE encapsulation tunnel between the BIG-IP system and the remote devices without having to specify a remote (or source) IP address for every device. The use cases include situations where the source IP address is unknown or difficult to discover.

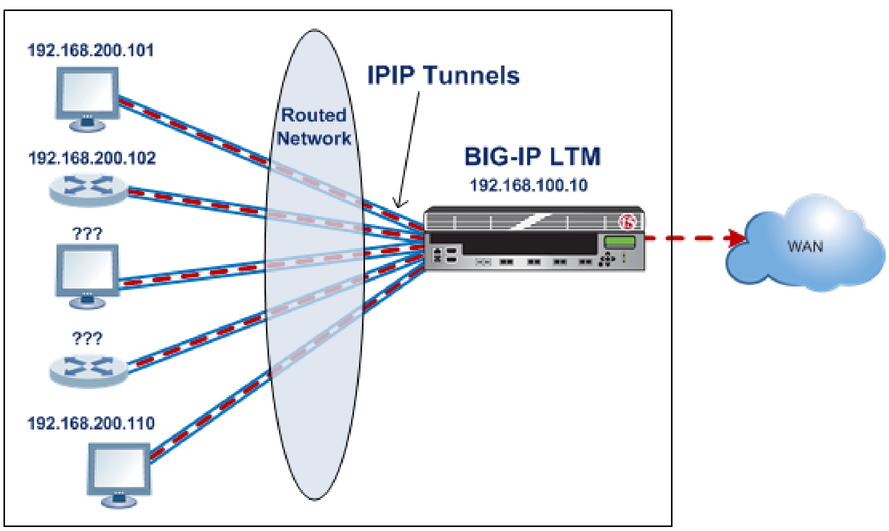


Figure 23: Illustration of an IPIP tunnel between a BIG-IP system and multiple unspecified devices

Creating an encapsulation tunnel between a BIG-IP device and multiple devices

You can create a tunnel between a BIG-IP® system and multiple remote devices without having to specify a remote (or source) IP address for every device.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select the type that corresponds to the encapsulation protocol you want to use.

   The selection **ipip** is the same as **ip4ip4**, but **ipip** is compatible with configurations from an earlier release.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, retain the default selection, **Any**.

   This entry means that you do not have to specify the IP address of the remote end of the tunnel, which allows multiple devices to use the same tunnel.
6. Click **Finished**.

When the BIG-IP system receives an encapsulated packet, the system decapsulates the packet, regardless of the source address, and re-injects it into the IP stack, thus allowing the inner IP address to be associated with a virtual server.

If you are configuring routes that use the tunnel as a resource, you must also assign a self IP address to the tunnel itself, which is different from the tunnel local endpoint IP address.

About transparent tunnels

You can create transparent tunnels when you want to inspect and/or manipulate encapsulated traffic that is flowing through a BIG-IP® system. The BIG-IP system terminates the tunnel, while presenting the illusion that the traffic flows through the device unchanged. In this case, the BIG-IP device appears as if it were an intermediate router that simply routes IP traffic through the device.

The transparent tunnel feature enables redirection of traffic based on policies. For example, service providers can redirect traffic with transparent tunnels to apply classification and bandwidth management policies using Policy Enforcement Manager™. To handle payload inspection and manipulation, you can create a policy in the form of a virtual server that accepts encapsulated packets. In the absence of a policy, the tunnel simply traverses the BIG-IP device.

Transparent tunnels are available for IPIP and GRE encapsulation types, with only one level of encapsulation.

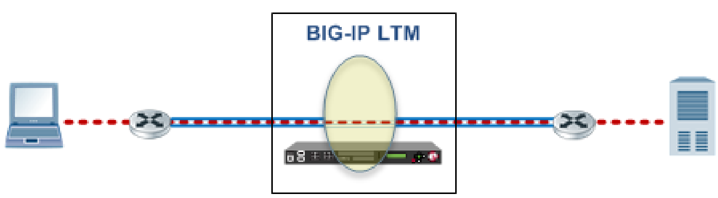


Figure 24: Illustration of a transparent tunnel

When the BIG-IP system receives an encapsulated packet from a transparent tunnel, the system decapsulates the packet, and re-injects it into the IP stack, where a virtual server can pick up the packet to apply a policy or rule. After applying the policy or rule, the BIG-IP can re-encapsulate the packet and route it, as if the packet had transited the BIG-IP unperturbed.

Creating a transparent tunnel

You can create transparent tunnels to inspect and modify tunneled traffic flowing through a BIG-IP® system.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select **ipip** or **gre**.
   The **ipip** selection can also be one of the IPIP variations: **ip4ip4**, **ip4ip6**, **ip6ip4**, or **ip6ip6**.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, retain the default selection, **Any**.
   This entry means that you do not have to specify the IP address of the remote end of the tunnel, which allows multiple devices to use the same tunnel.
6. Select the **Transparent** check box.
7. Click **Finished**.

Traffic flowing through the transparent tunnel you created is available for inspection and modification, before continuing to its destination.

After you create a transparent tunnel, additional configuration is required to process the traffic, such as creating a virtual server to intercept the traffic, and using Policy Enforcement Manager™ to apply classification and bandwidth management policies.

Configuring IPsec in Tunnel Mode between Two BIG-IP Systems

Overview: Configuring IPsec between two BIG-IP systems

You can configure an IPsec tunnel when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from one BIG-IP ®system to another. By following this procedure, you can configure an IKE peer to negotiate Phase 1 Internet Security Association and Key Management Protocol (ISAKMP) security associations for the secure channel between two systems. You can also configure a custom traffic selector and a custom IPsec policy that use this secure channel to generate IPsec Tunnel mode (Phase 2) security associations (SAs).

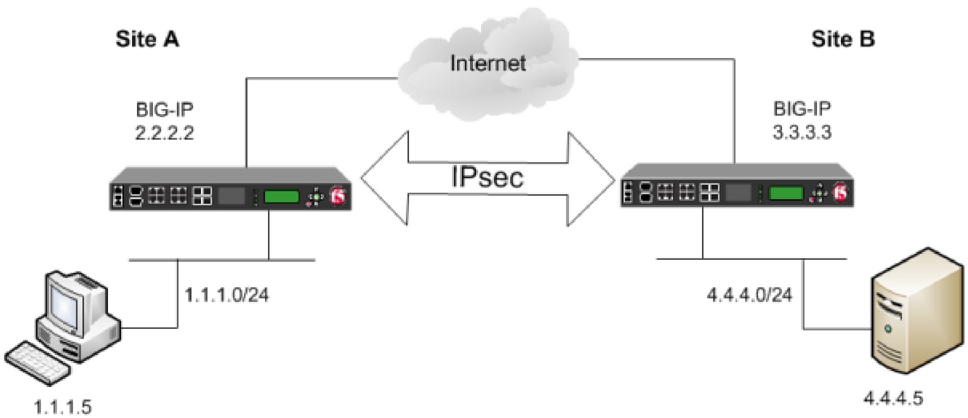


Figure 25: Example of an IPsec deployment

About negotiation of security associations

The way to dynamically negotiate security associations is to configure the Internet Key Exchange (IKE) protocol, which is included in the IPsec protocol suite. When you configure the *IKE protocol*, two IPsec tunnel endpoints (IKE peers) open a secure channel using an ISAKMP security association (ISAKMP-SA) to initially negotiate the exchange of peer-to-peer authentication data. This exchange is known as *Phase 1 negotiation*.

After Phase 1 is complete and the secure channel is established, *Phase 2 negotiation* begins, in which the IKE peers dynamically negotiate the authentication and encryption algorithms to use to secure the payload. Without IKE, the system cannot dynamically negotiate these security algorithms.

About IPsec Tunnel mode

Tunnel mode causes the IPsec protocol to encrypt the entire packet (the payload plus the IP header). This encrypted packet is then included as the payload in another outer packet with a new header. Traffic sent in this mode is more secure than traffic sent in Transport mode, because the original IP header is encrypted along with the original payload.

About BIG-IP components of the IPsec protocol suite

The IPsec protocol suite on the BIG-IP® system consists of these configuration components:

IKE peers
An *IKE peer* is a configuration object of the IPsec protocol suite that represents a BIG-IP system on each side of the IPsec tunnel. IKE peers allow two systems to authenticate each other (known as IKE Phase 1). The BIG-IP system includes the default IKE peer, named `anonymous`.

IPsec policies
An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession). For Tunnel mode, the policy also specifies the endpoints for the tunnel, and for IKE Phase 2 negotiation, the policy specifies the security parameters to be used in that negotiation. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets. The BIG-IP system includes two default IPsec policies, named `default-ipsec-policy` and `default-ipsec-policy-isession`. A common configuration includes a bidirectional policy on each BIG-IP system.

Traffic selectors
A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers. A common configuration includes a bidirectional traffic selector on each BIG-IP system.

About IP Payload Compression Protocol (IPComp)

IP Payload Compression Protocol (IPComp) is a protocol that reduces the size of IP payloads by compressing IP datagrams before fragmenting or encrypting the traffic. IPComp is typically used to improve encryption and decryption performance, thus increasing bandwidth utilization. Using an IPsec ESP tunnel can result in packet fragmentation, because the protocol adds a significant number of bytes to a packet. The additional bytes can push the packet over the maximum size allowed on the outbound link. Using compression is one way to mitigate fragmentation. IPComp is an option when you create a custom IPsec policy.

Task summary

You can configure the IPsec and IKE protocols to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec and IKE, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™
This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address
Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP connectivity

Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use ping to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address `0.0.0.0`.
   c) In the **Mask** field, type the netmask `0.0.0.0`.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IKE peer

The IKE peer object identifies to the system you are configuring the other BIG-IP system with which it communicates during Phase 1 negotiations. The IKE peer object also specifies the specific algorithms and credentials to be used for Phase 1 negotiation.

Important: *You must perform this task on both BIG-IP systems.*

---

1. On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
2. Click the **Create** button.
   The New IKE Peer screen opens.
3. In the **Name** field, type a unique name for the IKE peer.
4. In the **Description** field, type a brief description of the IKE peer.
5. In the **Remote Address** field, type the IP address of the BIG-IP system that is remote to the system you are configuring.
6. For the **State** setting, retain the default value, **Enabled**.

7. For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.
8. In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **RSA Signature** or **Preshared Key**.
   - If you select **RSA Signature** (default), the **Certificate**, **Key**, and **Verify Certificate** settings are available. If you have your own certificate file, key file, and certificate authority (CA), F5 recommends, for security purposes, that you specify these files in the appropriate fields. To reveal all these fields, select the **Verify Certificate** check box. If you retain the default settings, leave the check box cleared.

     Important: *If you select the check box, you must provide a certificate file, key, and certificate authority.*

   - If you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

   Note: *The key you type must be the same at both ends of the tunnel.*

9. For the Common Settings area, retain all default values.
10. Click **Finished**.
    The screen refreshes and displays the new IKE peer in the list.
11. Repeat this task on the BIG-IP system in the remote location.

You now have an IKE peer defined for establishing a secure channel.

Creating a custom IPsec policy

You create a custom IPsec policy when you want to use a policy other than the default IPsec policy (`default-ipsec-policy` or `default-ipsec-policy-isession`). A typical reason for creating a custom IPsec policy is to configure IPsec to operate in Tunnel rather than Transport mode. Another reason is to add payload compression before encryption.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. In the **Description** field, type a brief description of the policy.
5. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
6. From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.
7. In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
   This table shows sample tunnel local addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Local Address
BIG-IP A	`2.2.2.2`
BIG-IP B	`3.3.3.3`

8. In the **Tunnel Remote Address** field, type the IP address that is remote to the system you are configuring.

This address must match the **Remote Address** setting for the relevant IKE peer.

This table shows sample tunnel remote addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Remote Address
BIG-IP A	`3.3.3.3`
BIG-IP B	`2.2.2.2`

9. For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
10. For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
11. For the **Perfect Forward Secrecy** setting, select the option appropriate for your deployment.
12. For the **IPComp** setting, do one of the following:
    - Retain the default value **None**, if you do not want to enable packet-level compression before encryption.
    - Select **DEFLATE** to enable packet-level compression before encryption.
13. For the **Lifetime** setting, retain the default value, **1440**.

    This is the length of time (in minutes) before the current security association expires.
14. Click **Finished**.
    The screen refreshes and displays the new IPsec policy in the list.
15. Repeat this task on the BIG-IP system in the remote location.

Creating a bidirectional IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. For the **Order** setting, retain the default value (**First**).

   This setting specifies the order in which the traffic selector appears on the Traffic Selector List screen.
6. From the **Configuration** list, select **Advanced**.
7. For the **Source IP Address** setting, click **Host** or **Network**, and in the **Address** field, type an IP address.

   This IP address should be the host or network address from which the application traffic originates.

   This table shows sample source IP addresses for BIG-IP A and BIG-IP B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
BIG-IP B	`4.4.4.0/24`

8. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
9. For the **Destination IP Address** setting, click **Host**, and in the **Address** field, type an IP address.

   This IP address should be the final host or network address to which the application traffic is destined.

   This table shows sample destination IP addresses for BIG-IP A and BIG-IP B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
BIG-IP B	`1.1.1.0/24`

10. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
11. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.
12. From the **Direction** list, select **Both**.
13. From the **Action** list, select **Protect**.
    The **IPsec Policy Name** setting appears.
14. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
15. Click **Finished**.
    The screen refreshes and displays the new IPsec traffic selector in the list.
16. Repeat this task on the BIG-IP system in the remote location.

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

***Note:** Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   `tmsh modify net ipsec ike-daemon ikedaemon log-level info`

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. Check the IKE Phase 1 negotiation status by typing this command at the prompt.

   `racoonctl -l show-sa isakmp`

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies               ST S  V E Created             Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

   This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. Check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source          Destination          Status         Side
10.100.20.3     165.160.15.20        sa established [R]
```

   This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent

Column	Displayed
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
---------------------------------------------------------------------------

  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                      Create:  12/13/2012:10:39
```

8. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
```

```
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

9. Check the IPsec stats by typing this command at the prompt.

   `tmsh show net ipsec-stat`

   If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id          Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0          TRANSPORT           0         0            0          0
0          TRANSPORT           0         0            0          0
0             TUNNEL           0         0            0          0
0             TUNNEL           0         0            0          0
1             TUNNEL      353.9K    252.4M        24.9K       1.8M
2             TUNNEL      117.9K     41.0M       163.3K      12.4M
```

10. If the SAs are established, but traffic is not passing, type this command at the prompt.

    `tmsh delete net ipsec ipsec-sa`

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.
11. If traffic is still not passing, type this command at the prompt.

    `racoonctl flush-sa isakmp`

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.
12. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established
10.100.20.3[500]-165.160.15.20[500] spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
165.160.15.20[0]->10.100.20.3[0] spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
10.100.20.3[0]->165.160.15.20[0] spi=4573766(0x45ca46
```

13. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

    `tmsh modify net ipsec ike-daemon ikedaemon log-level debug2`

    Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

    Note: *Using this command flushes existing SAs.*

14. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level info
```

Note: *Using this command flushes existing SAs.*

Implementation result

You now have an IPsec tunnel for securing traffic that traverses the WAN, from one BIG-IP® system to another.

Configuring IPsec in Transport Mode between Two BIG-IP Systems

Overview: Configuring IPsec in Transport mode between two BIG-IP systems

You can configure IPsec when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from one BIG-IP® system to another. By following this procedure, you can configure an IKE peer to negotiate Phase 1 Internet Security Association and Key Management Protocol (ISAKMP) security associations for the secure channel between two systems. You can also configure a custom traffic selector and a custom IPsec policy that use this secure channel to generate IPsec Transport mode (Phase 2) security associations (SAs).

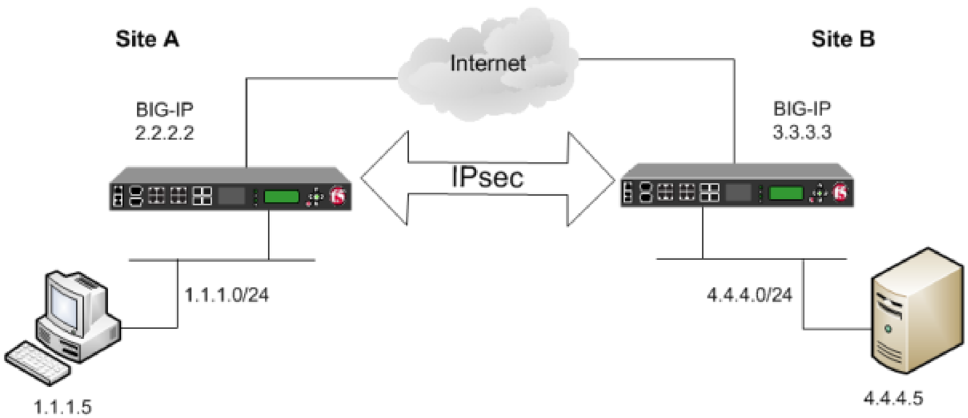


Figure 26: Example of an IPsec deployment

About negotiation of security associations

The way to dynamically negotiate security associations is to configure the Internet Key Exchange (IKE) protocol, which is included in the IPsec protocol suite. When you configure the *IKE protocol*, two IPsec tunnel endpoints (IKE peers) open a secure channel using an ISAKMP security association (ISAKMP-SA) to initially negotiate the exchange of peer-to-peer authentication data. This exchange is known as *Phase 1 negotiation*.

After Phase 1 is complete and the secure channel is established, *Phase 2 negotiation* begins, in which the IKE peers dynamically negotiate the authentication and encryption algorithms to use to secure the payload. Without IKE, the system cannot dynamically negotiate these security algorithms.

About IPsec Transport mode

Transport mode causes the IPsec protocol to encrypt only the payload of an IP packet. The protocol then encloses the encrypted payload in a normal IP packet. Traffic sent in Transport mode is less secure than traffic sent in Tunnel mode, because the IP header in each packet is not encrypted.

About BIG-IP components of the IPsec protocol suite

The IPsec protocol suite on the BIG-IP® system consists of these configuration components:

IKE peers
An *IKE peer* is a configuration object of the IPsec protocol suite that represents a BIG-IP system on each side of the IPsec tunnel. IKE peers allow two systems to authenticate each other (known as IKE Phase 1). The BIG-IP system includes the default IKE peer, named `anonymous`.

IPsec policies
An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession). For Tunnel mode, the policy also specifies the endpoints for the tunnel, and for IKE Phase 2 negotiation, the policy specifies the security parameters to be used in that negotiation. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets. The BIG-IP system includes two default IPsec policies, named `default-ipsec-policy` and `default-ipsec-policy-isession`. A common configuration includes a bidirectional policy on each BIG-IP system.

Traffic selectors
A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers. A common configuration includes a bidirectional traffic selector on each BIG-IP system.

About IP Payload Compression Protocol (IPComp)

IP Payload Compression Protocol (IPComp) is a protocol that reduces the size of IP payloads by compressing IP datagrams before fragmenting or encrypting the traffic. IPComp is typically used to improve encryption and decryption performance, thus increasing bandwidth utilization. Using an IPsec ESP tunnel can result in packet fragmentation, because the protocol adds a significant number of bytes to a packet. The additional bytes can push the packet over the maximum size allowed on the outbound link. Using compression is one way to mitigate fragmentation. IPComp is an option when you create a custom IPsec policy.

Task summary

With this task, you can configure the IPsec and IKE protocols to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec and IKE, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™
This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address
Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP connectivity

Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use ping to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address `0.0.0.0`.
   c) In the **Mask** field, type the netmask `0.0.0.0`.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IKE peer

The IKE peer object identifies to the system you are configuring the other BIG-IP system with which it communicates during Phase 1 negotiations. The IKE peer object also specifies the specific algorithms and credentials to be used for Phase 1 negotiation.

Important: *You must perform this task on both BIG-IP systems.*

---

1. On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
2. Click the **Create** button.
   The New IKE Peer screen opens.
3. In the **Name** field, type a unique name for the IKE peer.
4. In the **Description** field, type a brief description of the IKE peer.
5. In the **Remote Address** field, type the IP address of the BIG-IP system that is remote to the system you are configuring.
6. For the **State** setting, retain the default value, **Enabled**.

7. For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.
8. In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **RSA Signature** or **Preshared Key**.
   - If you select **RSA Signature** (default), the **Certificate**, **Key**, and **Verify Certificate** settings are available. If you have your own certificate file, key file, and certificate authority (CA), F5 recommends, for security purposes, that you specify these files in the appropriate fields. To reveal all these fields, select the **Verify Certificate** check box. If you retain the default settings, leave the check box cleared.

     Important: *If you select the check box, you must provide a certificate file, key, and certificate authority.*

   - If you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

   Note: *The key you type must be the same at both ends of the tunnel.*

9. For the Common Settings area, retain all default values.
10. Click **Finished**.
    The screen refreshes and displays the new IKE peer in the list.
11. Repeat this task on the BIG-IP system in the remote location.

You now have an IKE peer defined for establishing a secure channel.

Creating a bidirectional IPsec policy

You create a custom IPsec policy when you want to use a policy other than the default IPsec policy (`default-ipsec-policy` or `default-ipsec-policy-isession`). A typical reason for creating a custom IPsec policy is to configure IPsec to operate in Tunnel rather than Transport mode. Another reason is to add payload compression before encryption.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. In the **Description** field, type a brief description of the policy.
5. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
6. From the **Mode** list, select **Transport**.
7. For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
8. For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
9. For the **Perfect Forward Secrecy** setting, select the option appropriate for your deployment.
10. For the **IPComp** setting, do one of the following:
    - Retain the default value **None**, if you do not want to enable packet-level compression before encryption.
    - Select **DEFLATE** to enable packet-level compression before encryption.

11. For the **Lifetime** setting, retain the default value, **1440**.

    This is the length of time (in minutes) before the current security association expires.
12. Click **Finished**.
    The screen refreshes and displays the new IPsec policy in the list.
13. Repeat this task on the BIG-IP system in the remote location.

Creating a bidirectional IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. For the **Order** setting, retain the default value (**First**).

   This setting specifies the order in which the traffic selector appears on the Traffic Selector List screen.
6. From the **Configuration** list, select **Advanced**.
7. For the **Source IP Address** setting, click **Host** or **Network**, and in the **Address** field, type an IP address.

   This IP address should be the host or network address from which the application traffic originates.

   This table shows sample source IP addresses for BIG-IP A and BIG-IP B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
BIG-IP B	`4.4.4.0/24`

8. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
9. For the **Destination IP Address** setting, click **Host**, and in the **Address** field, type an IP address.

   This IP address should be the final host or network address to which the application traffic is destined.

   This table shows sample destination IP addresses for BIG-IP A and BIG-IP B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
BIG-IP B	`1.1.1.0/24`

10. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
11. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.
12. From the **Direction** list, select **Both**.

13. From the **Action** list, select **Protect**.
    The **IPsec Policy Name** setting appears.
14. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
15. Click **Finished**.
    The screen refreshes and displays the new IPsec traffic selector in the list.
16. Repeat this task on the BIG-IP system in the remote location.

Verifying IPsec connectivity for Transport mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

Note: *Only data traffic triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.
   `tmsh modify net ipsec ike-daemon ikedaemon log-level info`
   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the **Destination IP Address** in the traffic selector.
4. Check the IKE Phase 1 negotiation status by typing this command at the prompt.
   `racoonctl -l show-sa isakmp`
   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies                 ST S  V E Created             Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator

Column	Displayed	Description
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. Check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source           Destination          Status         Side
10.100.20.3      165.160.15.20        sa established [R]
```

   This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3  ->  165.160.15.20  SPI(0x164208ae)  out  esp  (tmm: 0)
165.160.15.20  ->  10.100.20.3  SPI(0xfa2ca7a8)  in   esp  (tmm: 0)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20
-------------------------------------------------------------------------------------------

  tmm: 0
  Direction: out;  SPI: 0x164208ae(373426350);  Policy ID: 0x87e9(34793)
  Protocol: esp;  Mode: transport;  State: mature
  Authenticated Encryption : aes-gcm128
  Current Usage: 196 bytes
  Hard lifetime: 51 seconds; unlimited bytes
  Soft lifetime: 39 seconds; unlimited bytes
  Replay window size: 64
  Last use: 01/24/2014:14:03
 Create:  01/24/2014:14:03

165.160.15.20 -> 10.100.20.3
-------------------------------------------------------------------------------------------

  tmm: 0
  Direction: in;  SPI: 0xfa2ca7a8(4197230504);  Policy ID: 0x87e8(34792)
  Protocol: esp;  Mode: transport;  State: mature
  Authenticated Encryption : aes-gcm128
  Current Usage: 264 bytes
  Hard lifetime: 51 seconds; unlimited bytes
  Soft lifetime: 39 seconds; unlimited bytes
  Replay window size: 64
  Last use: 01/24/2014:14:03
 Create:  01/24/2014:14:03
```

8. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that are associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3  ->  165.160.15.20  SPI(0x164208ae)  out  esp  (tmm: 0)
165.160.15.20  ->  10.100.20.3  SPI(0xfa2ca7a8)  in   esp  (tmm: 0)
```

9. Check the IPsec stats by typing this command at the prompt.

```
tmsh show net ipsec-stat
```

If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT      353.9K    252.4M        24.9K       1.8M
0           TRANSPORT      117.9K     41.0M       163.3K      12.4M
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL           0         0            0          0
2              TUNNEL           0         0            0          0
```

10. If the SAs are established, but traffic is not passing, type this command at the prompt.

    ```
    tmsh delete net ipsec ipsec-sa
    ```

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.
11. If traffic is still not passing, type this command at the prompt.

    ```
    racoonctl flush-sa isakmp
    ```

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.
12. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established
10.100.20.3[500]-165.160.15.20[500] spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Transport
165.160.15.20[0]->10.100.20.3[0] spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Transport
10.100.20.3[0]->165.160.15.20[0] spi=4573766(0x45ca46
```

13. For troubleshooting, increase the debug level by typing this command at the prompt.

    ```
    tmsh modify net ipsec ike-daemon ikedaemon log-level debug2
    ```

    Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

    Note: *Using this command flushes existing SAs.*
14. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

    ```
    tmsh modify net ipsec ike-daemon ikedaemon log-level info
    ```

    Note: *Using this command flushes existing SAs.*

Implementation result

You now have a secure IPsec channel for securing traffic that traverses the WAN, from one BIG-IP® system to another.

Configuring IPsec in Interface Mode between Two BIG-IP Systems

Overview: Configuring IPsec in Interface mode between two BIG-IP systems

You can configure an IPsec tunnel when you want to secure traffic that traverses a wide area network (WAN), from one BIG-IP ®system to another. By following this procedure, you can create an IPsec tunnel interface that can be used as any other BIG-IP VLAN. When you configure an IPsec tunnel interface, the IKE tunnel mode security associations occur automatically as part of the tunnel negotiation. For the IPsec tunnel interface, only the IPsec Encapsulating Security Protocol (ESP) is supported for the tunnel interface, and IPComp is not available.

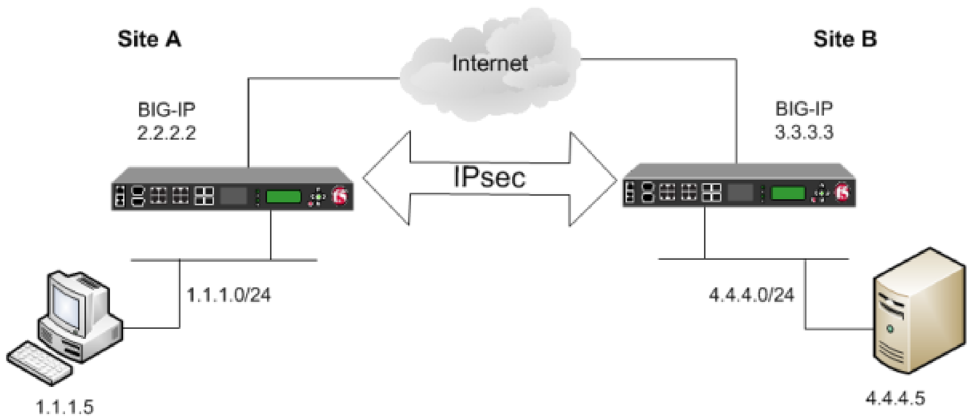


Figure 27: Example of an IPsec deployment

Task summary

Before you begin configuring IPsec, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™
This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address
Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP connectivity
Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use `ping` to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address `0.0.0.0`.
   c) In the **Mask** field, type the netmask `0.0.0.0`.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating a custom IPsec policy for Interface mode

You can create a custom IPsec policy to specify the Interface mode, which allows you to use the IPsec tunnel as a network interface object.

Important: *You must perform this task on the BIG-IP® systems at both sides of the tunnel.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
5. From the **Mode** list, select **Interface**.
6. In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
   This table shows sample tunnel local addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Local Address
BIG-IP A	`2.2.2.2`

System Name	Tunnel Local Address
BIG-IP B	3.3.3.3

7. In the **Tunnel Remote Address** field, type the IP address that is remote to the system you are configuring.
   This address must match the **Remote Address** setting for the relevant IKE peer.
   This table shows sample tunnel remote addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Remote Address
BIG-IP A	3.3.3.3
BIG-IP B	2.2.2.2

8. Click **Finished**.
   The screen refreshes and displays the new IPsec policy in the list.
9. Repeat this task on the BIG-IP system in the remote location.

Creating an IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses you specify and the custom IPsec policy you assign.

Important: *You must perform this task on the BIG-IP® systems on both sides of the WAN.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. For the **Source IP Address** setting, specify where the application traffic originates, either:
   - Click **Host** and type an IP address.
   - Click **Network**, and in the **Address** field, type an IP address.

   This table shows sample source IP addresses for BIG-IP A and BIG-IP B.

System Name	Source IP Address
BIG-IP A	1.1.1.0/24
BIG-IP B	4.4.4.0/24

5. For the **Destination IP Address** setting, specify where the application traffic is going, either:
   - Click **Host** and type an IP address.
   - Click **Network**, and in the **Address** field, type an IP address.

   This table shows sample destination IP addresses for BIG-IP A and BIG-IP B.

System Name	Destination IP Address
BIG-IP A	4.4.4.0/24
BIG-IP B	1.1.1.0/24

6. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
7. Click **Finished**.
   The screen refreshes and displays the new IPsec traffic selector in the list.
8. Repeat this task on the BIG-IP system in the remote location.

Specifying an IPsec tunnel interface traffic selector

You can create an IPsec tunnel profile to filter traffic according to the traffic selector you specify.

1. On the Main tab, click **Network** > **Tunnels** > **Profiles** > **IPsec** > **Create**.
   The New IPsec Profile screen opens.
2. In the **Name** field, type a unique name for the profile.
3. From the **Parent Profile** list, select **ipsec**.
4. Select the **Custom** check box.
5. From the **Traffic Selector** list, select the traffic selector you created.
6. Click **Finished**.

To use this IPsec profile to filter traffic, you must apply it to an IPsec tunnel.

Creating an IPsec interface tunnel

You can create an IPsec interface tunnel to apply an IPsec profile you have created to specify the traffic selector to filter the traffic.

1. On the Main tab, click **Network** > **Tunnels** > **Tunnel List** > **Create**.
   The New Tunnel screen opens.
2. In the **Name** field, type a unique name for the tunnel.
3. From the **Encapsulation Type** list, select **IPsec**.
4. In the **Local Address** field, type the IP address of the BIG-IP system.
5. From the **Remote Address** list, select **Specify**, and type the IP address of the BIG-IP device at the other end of the tunnel.
6. Click **Finished**.

After you create an IPsec tunnel interface, you can use it just like any other tunnel interface, such as assigning it a self IP address, associating it with route domains, and adding it to virtual servers.

Assigning a self IP address to an IP tunnel endpoint

Ensure that you have created an IP tunnel before starting this task.

Self IP addresses can enable the BIG-IP® system, and other devices on the network, to route application traffic through the associated tunnel, similar to routing through VLANs and VLAN groups.

Note: *If the other side of the tunnel needs to be reachable, make sure the self IP addresses that you assign to both sides of the tunnel are in the same subnet.*

---

1. On the Main tab, click **Network** > **Self IPs**.

2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a unique name for the self IP address.
4. In the **IP Address** field, type the IP address of the tunnel.

   The system accepts IPv4 and IPv6 addresses.

   Note: *This is not the same as the IP address of the tunnel local endpoint.*

   ---

5. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
6. From the **VLAN/Tunnel** list, select the tunnel with which to associate this self IP address.
7. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

Assigning a self IP to a tunnel ensures that the tunnel appears as a resource for routing traffic.

To direct traffic through the tunnel, add a route for which you specify the tunnel as the resource.

Configuring IPsec between a BIG-IP System and a Third-Party Device

Overview: Configuring IPsec between a BIG-IP system and a third-party device

You can configure an IPsec tunnel when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from a BIG-IP® system to third-party device. By following this process, you can configure an IKE peer to negotiate Phase 1 Internet Security Association and Key Management Protocol (ISAKMP) security associations for the secure channel between two systems. You can also configure a custom traffic selector and a custom IPsec policy that use this secure channel to generate IPsec Tunnel mode (Phase 2) security associations (SAs).

This implementation describes the tasks for setting up the IPsec tunnel on the BIG-IP system. You must also configure the third-party device at the other end of the tunnel. For those instructions, refer to the manufacturer's documentation for your device.

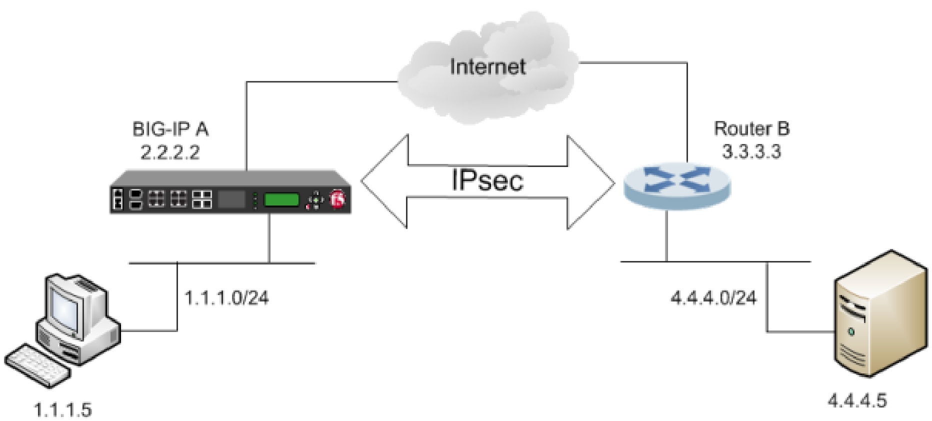


Figure 28: Example of an IPsec tunnel between a BIG-IP system and a third-party device

About negotiation of security associations

The way to dynamically negotiate security associations is to configure the Internet Key Exchange (IKE) protocol, which is included in the IPsec protocol suite. When you configure the *IKE protocol*, two IPsec tunnel endpoints (IKE peers) open a secure channel using an ISAKMP security association (ISAKMP-SA) to initially negotiate the exchange of peer-to-peer authentication data. This exchange is known as *Phase 1 negotiation*.

After Phase 1 is complete and the secure channel is established, *Phase 2 negotiation* begins, in which the IKE peers dynamically negotiate the authentication and encryption algorithms to use to secure the payload. Without IKE, the system cannot dynamically negotiate these security algorithms.

About IPsec Tunnel mode

Tunnel mode causes the IPsec protocol to encrypt the entire packet (the payload plus the IP header). This encrypted packet is then included as the payload in another outer packet with a new header. Traffic sent in this mode is more secure than traffic sent in Transport mode, because the original IP header is encrypted along with the original payload.

About BIG-IP components of the IPsec protocol suite

The IPsec protocol suite on the BIG-IP® system consists of these configuration components:

IKE peers
An *IKE peer* is a configuration object of the IPsec protocol suite that represents a BIG-IP system on each side of the IPsec tunnel. IKE peers allow two systems to authenticate each other (known as IKE Phase 1). The BIG-IP system includes the default IKE peer, named `anonymous`.

IPsec policies
An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession). For Tunnel mode, the policy also specifies the endpoints for the tunnel, and for IKE Phase 2 negotiation, the policy specifies the security parameters to be used in that negotiation. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets. The BIG-IP system includes two default IPsec policies, named `default-ipsec-policy` and `default-ipsec-policy-isession`. A common configuration includes a bidirectional policy on each BIG-IP system.

Traffic selectors
A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers. A common configuration includes a bidirectional traffic selector on each BIG-IP system.

Task summary

You can configure the IPsec and IKE protocols to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec and IKE, verify that this module, system objects, and connectivity exist on the BIG-IP® system:

BIG-IP Local Traffic Manager™
This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address
The BIG-IP system must have at least one self IP address, to be used in specifying the end of the IPsec tunnel.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP connectivity
Verify the connectivity between the client or server and its BIG-IP device, and between the BIG-IP device and its gateway. For example, you can use ping to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address `0.0.0.0`.
   c) In the **Mask** field, type the netmask `0.0.0.0`.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IKE peer

The IKE peer object identifies to the system you are configuring the other device with which it communicates during Phase 1 negotiations. The IKE peer object also specifies the specific algorithms and credentials to be used for Phase 1 negotiation.

Important: *You must also configure the device at the other end of the IPsec tunnel.*

---

1. On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
2. Click the **Create** button.
   The New IKE Peer screen opens.
3. In the **Name** field, type a unique name for the IKE peer.
4. In the **Description** field, type a brief description of the IKE peer.
5. In the **Remote Address** field, type the IP address of the device that is remote to the system you are configuring.

   This address must match the value of the **Tunnel Remote Address** setting in the relevant IPsec policy.
6. For the **State** setting, retain the default value, **Enabled**.
7. For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.

Important: *The values you select must match the IKE Phase 1 settings on the remote device.*

Setting	Options
Authentication Algorithm	**MD5** **SHA-1** (default) **SHA-256** **SHA-384** **SHA-512**
Encryption Algorithm	**DES** **3 DES** (default) **BLOWFISH** **CAST128** **AES** **CAMELLIA**
Perfect Forward Secrecy	**MODP768** **MODP1024** (default) **MODP1536** **MODP2048** **MODP3072** **MODP4096** **MODP6144** **MODP8192**
Lifetime	Length of time, in minutes, before the IKE security association expires.

8. In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **RSA Signature** or **Preshared Key**.
   - If you select **RSA Signature** (default), the **Certificate**, **Key**, and **Verify Certificate** settings are available. If you have your own certificate file, key file, and certificate authority (CA), F5 recommends, for security purposes, that you specify these files in the appropriate fields. To reveal all these fields, select the **Verify Certificate** check box. If you retain the default settings, leave the check box cleared.

     Important: *If you select the check box, you must provide a certificate file, key, and certificate authority.*
   - If you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

   Note: *The key you type must be the same at both ends of the tunnel.*
9. For the Common Settings area, retain all default values.
10. Click **Finished**.
    The screen refreshes and displays the new IKE peer in the list.

You now have an IKE peer defined for establishing a secure channel.

Creating a custom IPsec policy

You create a custom IPsec policy when you want to use a policy other than the default IPsec policy (`default-ipsec-policy` or `default-ipsec-policy-isession`). A typical reason for creating a custom IPsec policy is to configure IPsec to operate in Tunnel rather than Transport mode.

Important: *You must also configure the device at the other end of the IPsec tunnel.*

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. In the **Description** field, type a brief description of the policy.
5. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
6. From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.
7. In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
   For example, the tunnel local IP address for BIG-IP A is `2.2.2.2`.
8. In the **Tunnel Remote Address** field, type the IP address that is remote to the system you are configuring.

   This address must match the **Remote Address** setting for the relevant IKE peer.

   For example, the tunnel remote IP address configured on BIG-IP A is the IP address of Router B, which is `3.3.3.3`.
9. For the IKE Phase 2 area, retain the default values, or select the options that are appropriate for your deployment.

   Important: *The values you select must match the IKE Phase 2 settings on the remote device.*

Setting	Options
Authentication Algorithm	**SHA-1** **AES-GCM128** (default) **AES-GCM192** **AES-GCM256** **AES-GMAC128** **AES-GMAC192** **AES-GMAC256**
Encryption Algorithm	**AES-GCM128** (default)
Perfect Forward Secrecy	**MODP768** **MODP1024** (default) **MODP1536** **MODP2048** **MODP3072** **MODP4096** **MODP6144** **MODP8192**
Lifetime	Length of time, in minutes, before the IKE security association expires.

10. Click **Finished**.

The screen refreshes and displays the new IPsec policy in the list.

Creating a bidirectional IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must also configure the device at the other end of the IPsec tunnel.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. For the **Order** setting, retain the default value (**First**).

   This setting specifies the order in which the traffic selector appears on the Traffic Selector List screen.
6. From the **Configuration** list, select **Advanced**.
7. For the **Source IP Address** setting, click **Host** or **Network**, and in the **Address** field, type an IP address.

   This IP address should be the host or network address from which the application traffic originates.

   This table shows sample source IP addresses for BIG-IP A and Router B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
Router B	`4.4.4.0/24`

8. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
9. For the **Destination IP Address** setting, click **Host**, and in the **Address** field, type an IP address.

   This IP address should be the final host or network address to which the application traffic is destined.

   This table shows sample destination IP addresses for BIG-IP A and Router B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
Router B	`1.1.1.0/24`

10. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
11. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.
12. From the **Direction** list, select **Both**.
13. From the **Action** list, select **Protect**.
    The **IPsec Policy Name** setting appears.
14. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
15. Click **Finished**.

The screen refreshes and displays the new IPsec traffic selector in the list.

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

***Note:** Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.
   `tmsh modify net ipsec ike-daemon ikedaemon log-level info`
   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. Check the IKE Phase 1 negotiation status by typing this command at the prompt.
   `racoonctl -l show-sa isakmp`
   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies              ST S  V E Created            Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

   This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive

Column	Displayed	Description
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. Check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source            Destination          Status         Side
10.100.20.3       165.160.15.20        sa established [R]
```

   This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
---------------------------------------------------------------------------

  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                      Create:  12/13/2012:10:39
```

8. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

9. Check the IPsec stats by typing this command at the prompt.

   `tmsh show net ipsec-stat`

   If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

10. If the SAs are established, but traffic is not passing, type this command at the prompt.

    `tmsh delete net ipsec ipsec-sa`

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.

11. If traffic is still not passing, type this command at the prompt.

    `racoonctl flush-sa isakmp`

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.

12. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established
10.100.20.3[500]-165.160.15.20[500] spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
165.160.15.20[0]->10.100.20.3[0] spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
10.100.20.3[0]->165.160.15.20[0] spi=4573766(0x45ca46
```

13. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

    `tmsh modify net ipsec ike-daemon ikedaemon log-level debug2`

    Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

    Note: *Using this command flushes existing SAs.*

14. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

    `tmsh modify net ipsec ike-daemon ikedaemon log-level info`

    Note: *Using this command flushes existing SAs.*

Implementation result

You now have an IPsec tunnel for securing traffic that traverses the WAN, from one BIG-IP® system to a third-party device.

Configuring IPsec Using Manually Keyed Security Associations

Overview: Configuring IPsec using manual security associations

You can configure an IPsec tunnel when you want to use a protocol other than SSL to secure traffic that traverses a wide area network (WAN), from one BIG-IP ®system to another. Typically, you would use the Internet Key Exchange (IKE) protocol to negotiate the secure channel between the two systems. If you choose not to use IKE, you must create manual security associations for IPsec security. A *manual security association* statically defines the specific attribute values that IPsec should use for the authentication and encryption of data flowing through the tunnel.

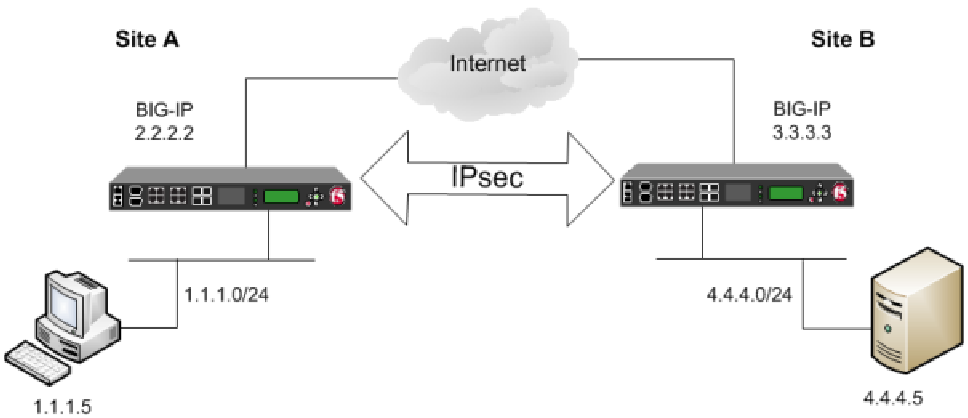


Figure 29: Illustration of an IPsec deployment

The implementation of the IPsec protocol suite with a manual security association consists of these components:

IPsec policy

An *IPsec policy* is a set of information that defines the specific IPsec protocol to use (ESP or AH), and the mode (Transport, Tunnel, or iSession). For Tunnel mode, the policy also specifies the endpoints for the tunnel. The way that you configure the IPsec policy determines the way that the BIG-IP system manipulates the IP headers in the packets. The BIG-IP system includes two default IPsec policies, named `default-ipsec-policy` and `default-ipsec-policy-isession`. A common configuration includes a bidirectional policy on each BIG-IP system.

Manual security association

A *manual security association* is set of information that the IPsec protocol uses to authenticate and encrypt application traffic.

Note: *When you create a manual security association instead of using IKE, the peer systems do not negotiate these attributes. Peers can communicate only when they share the same configured attributes.*

Traffic selector
A *traffic selector* is a packet filter that defines what traffic should be handled by a IPsec policy. You define the traffic by source and destination IP addresses and port numbers. A common configuration includes a bidirectional traffic selector on each BIG-IP system.

About IPsec Tunnel mode

Tunnel mode causes the IPsec protocol to encrypt the entire packet (the payload plus the IP header). This encrypted packet is then included as the payload in another outer packet with a new header. Traffic sent in this mode is more secure than traffic sent in Transport mode, because the original IP header is encrypted along with the original payload.

Task summary

You can configure an IPsec tunnel to secure traffic that traverses a wide area network (WAN), such as from one data center to another.

Before you begin configuring IPsec, verify that these modules, system objects, and connectivity exist on the BIG-IP® systems in both the local and remote locations:

BIG-IP Local Traffic Manager™
This module directs traffic securely and efficiently to the appropriate destination on a network.

Self IP address
Each BIG-IP system must have at least one self IP address, to be used in specifying the ends of the IPsec tunnel.

The default VLANs
These VLANs are named `external` and `internal`.

BIG-IP connectivity
Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. For example, you can use ping to test this connectivity.

Task list

Creating a forwarding virtual server for IPsec
Creating a manual IPsec security association
Creating a custom IPsec policy
Creating a bidirectional IPsec traffic selector
Verifying IPsec connectivity for Tunnel mode

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.

3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address `0.0.0.0`.
   c) In the **Mask** field, type the netmask `0.0.0.0`.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating a manual IPsec security association

Before starting this task, determine the source and destination IP addresses for the BIG-IP® systems in your network that will direct the application traffic.

You create a manual security association to specify the security attributes for a given IPsec communication session. These attributes include the specific source and destination IP addresses of the communicating devices, the authentication algorithm, and the encryption algorithm that the IPsec protocol should use.

Important: *You must perform this task on both BIG-IP systems.*

1. On the Main tab, click **Network** > **IPsec** > **Manual Security Associations**.
2. Click the **Create** button.
   The New Security Association screen opens.
3. In the **Name** field, type a unique name for the security association.
4. In the **Description** field, type a brief description of the security setting.
5. In the **SPI** field, type a unique number for the security parameter index.
   This number must be an integer between 256 and 4294967296.
6. In the **Source Address** field, type the source IP address.
7. In the **Destination Address** field, type the destination IP address.
8. In the **Authentication Key** field, type a key value.
   This value can by any double-quoted character string up to a maximum of 128 characters
9. From the **Encryption Algorithm** list, select the algorithm appropriate to your deployment.
10. In the **Encryption Key** field, type a key value.
    This value can by any double-quoted character string up to a maximum of 128 characters
11. Click **Finished**.
    The screen refreshes and displays the new IPsec security association in the list.
12. Repeat this task on the BIG-IP system in the remote location.

Creating a custom IPsec policy

You create a custom IPsec policy when you want to use a policy other than the default IPsec policy (`default-ipsec-policy` or `default-ipsec-policy-isession`). A typical reason for creating a custom IPsec policy is to configure IPsec to operate in Tunnel rather than Transport mode. Another reason is to add payload compression before encryption.

Important: *You must perform this task on both BIG-IP® systems.*

---

1. On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
2. Click the **Create** button.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. In the **Description** field, type a brief description of the policy.
5. For the **IPsec Protocol** setting, retain the default selection, **ESP**.
6. From the **Mode** list, select **Tunnel**.
   The screen refreshes to show additional related settings.
7. In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
   This table shows sample tunnel local addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Local Address
BIG-IP A	`2.2.2.2`
BIG-IP B	`3.3.3.3`

8. In the **Tunnel Remote Address** field, type the IP address that is remote to the system you are configuring.

   This address must match the **Remote Address** setting for the relevant IKE peer.

   This table shows sample tunnel remote addresses for BIG-IP A and BIG-IP B.

System Name	Tunnel Remote Address
BIG-IP A	`3.3.3.3`
BIG-IP B	`2.2.2.2`

9. For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
10. For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
11. For the **Perfect Forward Secrecy** setting, select the option appropriate for your deployment.
12. For the **IPComp** setting, do one of the following:
    - Retain the default value **None**, if you do not want to enable packet-level compression before encryption.
    - Select **DEFLATE** to enable packet-level compression before encryption.
13. For the **Lifetime** setting, retain the default value, **1440**.

    This is the length of time (in minutes) before the current security association expires.
14. Click **Finished**.
    The screen refreshes and displays the new IPsec policy in the list.
15. Repeat this task on the BIG-IP system in the remote location.

Creating a bidirectional IPsec traffic selector

The traffic selector you create filters traffic based on the IP addresses and port numbers that you specify, as well as the custom IPsec policy you assign.

Important: *You must perform this task on both BIG-IP® systems.*

1. On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
2. Click **Create**.
   The New Traffic Selector screen opens.
3. In the **Name** field, type a unique name for the traffic selector.
4. In the **Description** field, type a brief description of the traffic selector.
5. For the **Order** setting, retain the default value (**First**).

   This setting specifies the order in which the traffic selector appears on the Traffic Selector List screen.
6. From the **Configuration** list, select **Advanced**.
7. For the **Source IP Address** setting, click **Host** or **Network**, and in the **Address** field, type an IP address.

   This IP address should be the host or network address from which the application traffic originates.

   This table shows sample source IP addresses for BIG-IP A and BIG-IP B.

System Name	Source IP Address
BIG-IP A	`1.1.1.0/24`
BIG-IP B	`4.4.4.0/24`

8. From the **Source Port** list, select the source port for which you want to filter traffic, or retain the default value ***All Ports**.
9. For the **Destination IP Address** setting, click **Host**, and in the **Address** field, type an IP address.

   This IP address should be the final host or network address to which the application traffic is destined.

   This table shows sample destination IP addresses for BIG-IP A and BIG-IP B.

System Name	Destination IP Address
BIG-IP A	`4.4.4.0/24`
BIG-IP B	`1.1.1.0/24`

10. From the **Destination Port** list, select the destination port for which you want to filter traffic, or retain the default value *** All Ports**.
11. From the **Protocol** list, select the protocol for which you want to filter traffic.

    You can select *** All Protocols**, **TCP**, **UDP**, **ICMP**, or **Other**. If you select **Other**, you must type a protocol name.
12. From the **Direction** list, select **Both**.
13. From the **Action** list, select **Protect**.
    The **IPsec Policy Name** setting appears.
14. From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you created.
15. Click **Finished**.
    The screen refreshes and displays the new IPsec traffic selector in the list.
16. Repeat this task on the BIG-IP system in the remote location.

Verifying IPsec connectivity for Tunnel mode

After you have manually configured security associations for an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

Note: *Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Send data traffic to the destination IP address specified in the traffic selector.
3. Check the IPsec stats by typing this command at the prompt.

   `tmsh show net ipsec-stat`

   If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

4. To verify the establishment of manually configured security associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

5. To display the details of the manually configured security associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

   For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
---------------------------------------------------------------------------

  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
```

```
 Protocol: esp;  Mode: tunnel;  State: mature
 Authenticated Encryption : aes-gmac128
 Current Usage: 307488 bytes
 Hard lifetime: 94 seconds; unlimited bytes
 Soft lifetime: 34 seconds; unlimited bytes
 Replay window size: 64
Last use: 12/13/2012:10:42                      Create:  12/13/2012:10:39
```

Task summary
Creating a bidirectional IPsec traffic selector
Task summary

Setting Up IPsec To Use NAT Traversal on Both Sides of the WAN

Overview: Setting up IPsec to use NAT traversal on both sides of the WAN

When you are using IPsec to secure WAN traffic, you can set up an IPsec tunnel with NAT traversal (NAT-T) to get around a firewall or other NAT device. This implementation describes how to set up the IPsec tunnel when you have a NAT device on both sides of the tunnel.

The following illustration shows a network configuration with a firewall on both sides of the WAN.

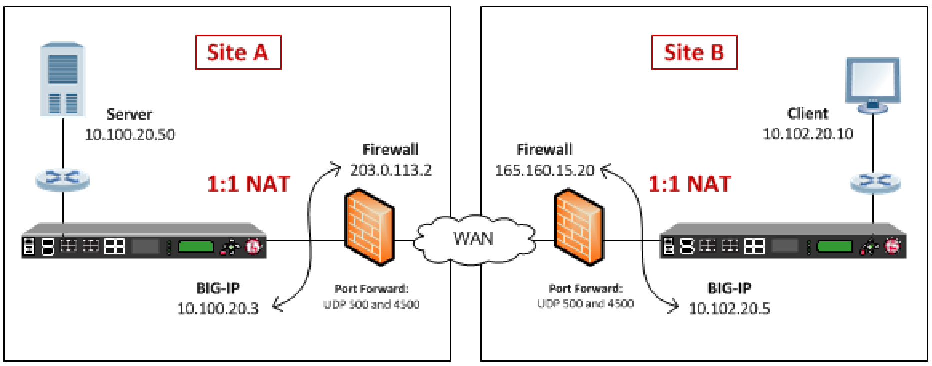


Figure 30: Example of an IPsec deployment with NAT-T on both sides of the WAN

Before you begin IPsec configuration

Before you configure IPsec on a BIG-IP® device, make sure that you have completed the following general prerequisites.

- You must have an existing routed IP network between the two locations where the BIG-IP devices will be installed.
- The BIG-IP hardware is installed with an initial network configuration applied.
- Application Acceleration Manager™ is provisioned at the level Nominal or Dedicated.
- The management IP address is configured on the BIG-IP system.
- If you are using NAT traversal, forward UDP ports 500 and 4500 to the BIG-IP system behind each firewall.
- Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. You can use ping to test connectivity.

Task summary

When you are configuring an IPsec tunnel, you must repeat the configuration tasks on the BIG-IP systems on both sides of the WAN.

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address `0.0.0.0`.
   c) In the **Mask** field, type the netmask `0.0.0.0`.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IPsec tunnel with NAT-T on both sides

You can create an IPsec tunnel to securely transport application traffic across the WAN. You must configure the IPsec tunnel on the BIG-IP systems on both sides of the WAN.

When you create an IKE peer for NAT traversal (NAT-T), the key configuration detail is that the **Remote Address** setting is the public IP address of the firewall or other NAT device (not the IP address of the remote BIG-IP system). Also, you must turn on NAT traversal. You can customize the remaining settings to conform to your network.

Important: *For the IKE peer negotiations to be successful, the IKE Phase 1 and IKE Phase 2 settings must be the same on the BIG-IP systems at both ends of the IPsec tunnel.*

---

1. Create an IKE peer that specifies the other end of the IPsec tunnel.
   a) On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
   b) Click the **Create** button.
   c) In the **Name** field, type a unique name for the IKE peer.

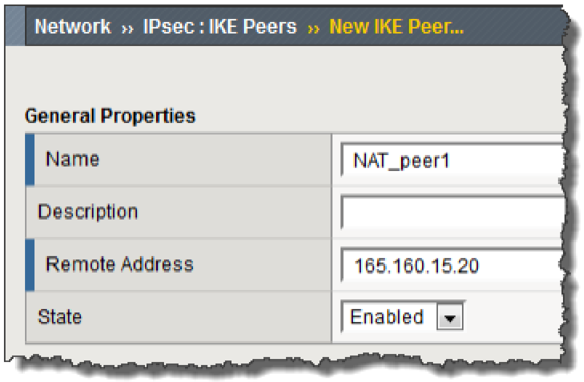
d) In the **Remote Address** field, type the public IP address of the firewall or other NAT device that is between the WAN and the remote BIG-IP system.

This address is the IP address of the remote peer, and must match the value of the **Tunnel Remote Address** setting in the relevant IPsec policy.

For example, the peer remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Remote (Peer) Address
Site A	`165.160.15.20`
Site B	`203.0.113.2`

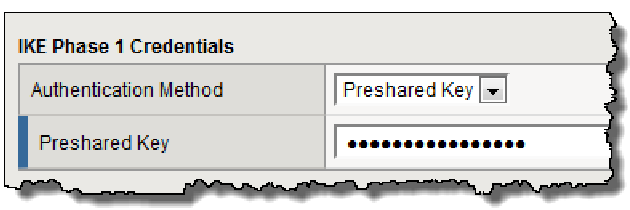
This screen snippet shows the peer **Remote Address** setting at Site A.



e) For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.

f) In the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **Preshared Key** or **RSA Signature**, and specify additional information in the fields that appear.

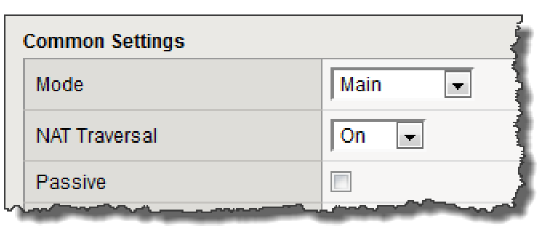
For example, if you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.



***Note:** The key you type must be the same at both ends of the tunnel.*

---

g) From the **NAT Traversal** list, select **On**.



h) Click **Finished**.

2. Create a custom IPsec policy that uses Tunnel mode and has the same remote IP address as the IKE peer.

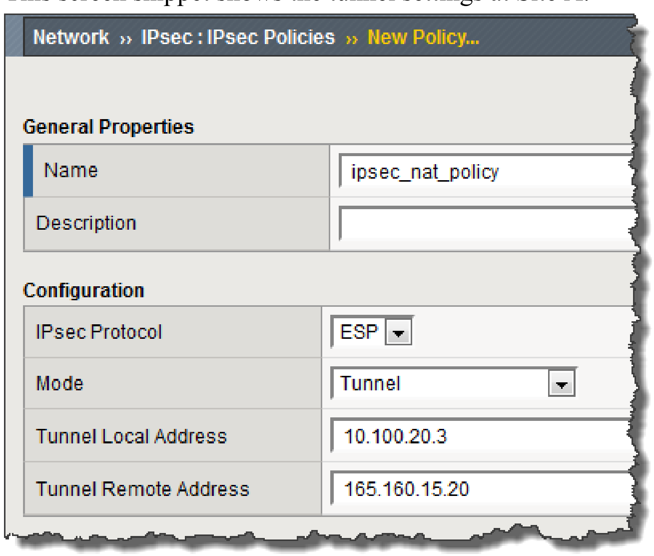
a) On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
b) Click the **Create** button.
c) In the **Name** field, type a unique name for the policy.
d) For the **IPsec Protocol** setting, retain the default selection, **ESP**.
e) From the **Mode** list, select **Tunnel**.
The screen refreshes to show additional related settings.
f) In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
For example, the tunnel local addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Local Address
Site A	`10.100.20.3`
Site B	`10.102.20.5`

g) In the **Tunnel Remote Address** field, type the public IP address of the firewall or other NAT device that is between the WAN and the remote BIG-IP system.
This address must match the value of the **Remote Address** setting for the relevant IKE peer.
For example, the tunnel remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Remote Address
Site A	`165.160.15.20`
Site B	`203.0.113.2`

This screen snippet shows the tunnel settings at Site A.



h) For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
i) For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
j) For the **Perfect Forward Secrecy** setting, retain the default value, or select the option appropriate for your deployment.
k) Click **Finished**.

3. Create a bidirectional traffic selector that uses the custom IPsec policy you created.

The traffic selector filters the application traffic based on the source and destination IP addresses you specify.

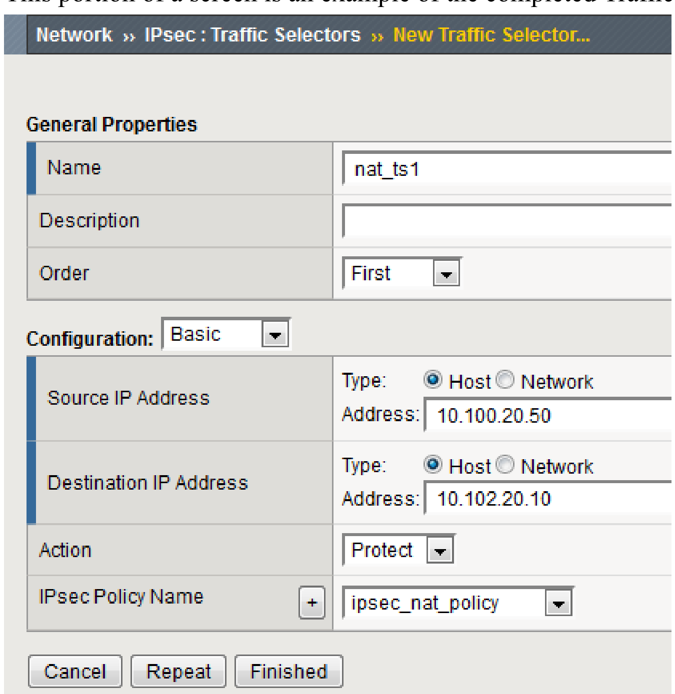
a) On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
b) Click **Create**.
c) In the **Name** field, type a unique name for the traffic selector.
d) For the **Order** setting, retain the default value (**First**).
e) For the **Source IP Address** setting, in the **Address** field, type the IP address from which the application traffic originates.
For example, the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Source IP Address
Site A	`10.100.20.50`
Site B	`10.102.20.10`

f) In the **Destination IP Address** setting **Address** field, type the final IP address for which the application traffic is destined.
For example, the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Destination IP Address
Site A	`10.102.20.10`
Site B	`10.100.20.50`

g) For the **Action** setting, retain the default value, **Protect**.
h) From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you just created.
This portion of a screen is an example of the completed Traffic Selector screen at Site A.



i) Click **Finished**.

You have now created an IPsec tunnel through which traffic travels in both directions across the WAN through firewalls on both sides.

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

***Note:** Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   `tmsh modify net ipsec ike-daemon ikedaemon log-level info`

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. Check the IKE Phase 1 negotiation status by typing this command at the prompt.

   `racoonctl -l show-sa isakmp`

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination       Cookies                ST S  V E Created             Phase2
165.160.15.20.500 98993e6 . . . 22c87f1  9 I 10 M 2012-06-27 16:51:19     1
```

   This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. Check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source            Destination           Status          Side
10.100.20.3       165.160.15.20         sa established [R]
```

   This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa`

   For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

   `tmsh show net ipsec ipsec-sa all-properties`

For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
---------------------------------------------------------------------------

  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                      Create:  12/13/2012:10:39
```

8. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

   `tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

   You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

   The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

9. Check the IPsec stats by typing this command at the prompt.

   `tmsh show net ipsec-stat`

   If traffic is passing through the IPsec tunnel, the stats will increment.

```
-------------------------------------------------------------------
Net::Ipsec
Cmd Id          Mode  Packets In  Bytes In  Packets Out  Bytes Out
-------------------------------------------------------------------
0          TRANSPORT           0         0            0          0
0          TRANSPORT           0         0            0          0
0             TUNNEL           0         0            0          0
0             TUNNEL           0         0            0          0
1             TUNNEL      353.9K    252.4M        24.9K       1.8M
2             TUNNEL      117.9K     41.0M       163.3K      12.4M
```

10. If the SAs are established, but traffic is not passing, type this command at the prompt.

    `tmsh delete net ipsec ipsec-sa`

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.

11. If traffic is still not passing, type this command at the prompt.

    `racoonctl flush-sa isakmp`

This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.

12. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established
10.100.20.3[500]-165.160.15.20[500] spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
165.160.15.20[0]->10.100.20.3[0] spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
10.100.20.3[0]->165.160.15.20[0] spi=4573766(0x45ca46
```

13. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

    `tmsh modify net ipsec ike-daemon ikedaemon log-level debug2`

    Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

    Note: *Using this command flushes existing SAs.*

14. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

    `tmsh modify net ipsec ike-daemon ikedaemon log-level info`

    Note: *Using this command flushes existing SAs.*

Setting Up IPsec To Use NAT Traversal on One Side of the WAN

Overview: Setting up IPsec to use NAT traversal on one side of the WAN

When you are using IPsec to secure WAN traffic, you can set up an IPsec tunnel with NAT traversal (NAT-T) to get around a firewall or other NAT device. This implementation describes how to set up the IPsec tunnel when you have a NAT device on one side of the tunnel.

The following illustration shows a network configuration with a firewall on one side of the WAN.

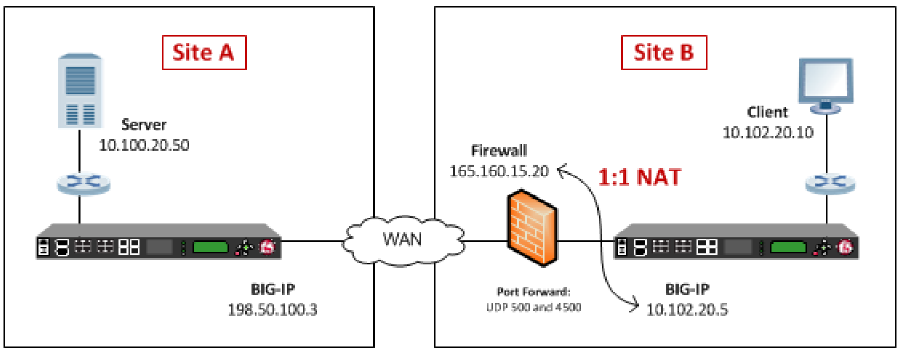


Figure 31: Example of an IPsec deployment with NAT-T on one side of the WAN

Before you begin IPsec configuration

Before you configure IPsec on a BIG-IP® device, make sure that you have completed the following general prerequisites.

- You must have an existing routed IP network between the two locations where the BIG-IP devices will be installed.
- The BIG-IP hardware is installed with an initial network configuration applied.
- Application Acceleration Manager™ is provisioned at the level Nominal or Dedicated.
- The management IP address is configured on the BIG-IP system.
- If you are using NAT traversal, forward UDP ports 500 and 4500 to the BIG-IP system behind each firewall.
- Verify the connectivity between the client or server and its BIG-IP device, and between each BIG-IP device and its gateway. You can use ping to test connectivity.

Task summary

When you are configuring an IPsec tunnel, you must repeat the configuration tasks on the BIG-IP systems on both sides of the WAN.

Creating a forwarding virtual server for IPsec

For IPsec, you create a forwarding virtual server to intercept IP traffic and direct it over the tunnel.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Type** list, select **Forwarding (IP)**.
5. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address `0.0.0.0`.
   c) In the **Mask** field, type the netmask `0.0.0.0`.
6. From the **Service Port** list, select ***All Ports**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN and Tunnel Traffic** list, retain the default selection, **All VLANs and Tunnels**.
9. Click **Finished**.

Creating an IPsec tunnel with NAT-T on one side

You can create an IPsec tunnel to securely transport application traffic across the WAN. You must configure an IPsec tunnel on the BIG-IP systems on both sides of the WAN.

When you create an IKE peer for NAT traversal (NAT-T), the key configuration detail is that the **Remote Address** setting is the public IP address of the firewall or other NAT device (not the IP address of the remote BIG-IP system). Also, you must turn on NAT traversal for that peer. You can customize the remaining settings to conform to your network.

Important: *For the IKE peer negotiations to be successful, the IKE Phase 1 and IKE Phase 2 settings must be the same on the BIG-IP systems at both ends of the IPsec tunnel.*

1. Create an IKE peer that specifies the other end of the IPsec tunnel.
   a) On the Main tab, click **Network** > **IPsec** > **IKE Peers**.
   b) Click the **Create** button.
   c) In the **Name** field, type a unique name for the IKE peer.

d) In the **Remote Address** field, type the IP address of the remote peer.

   If the remote BIG-IP system is behind a firewall or other NAT device, type the public IP address of that device.

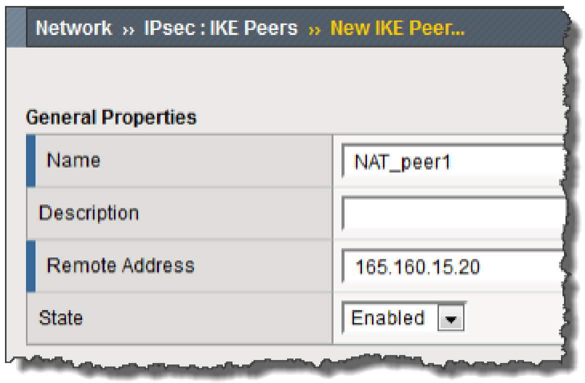
   If the remote BIG-IP system is reachable directly, type the IP address of the BIG-IP system.

   Note: *This address must match the value of the* ***Tunnel Remote Address*** *of the remote site setting in the relevant IPsec policy.*

   For example, Site A uses the WAN IP address of the Site B firewall. The peer remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Remote (Peer) Address
Site A	`165.160.15.20`
Site B	`198.50.100.3`

   This screen snippet shows the peer **Remote Address** setting at Site A.

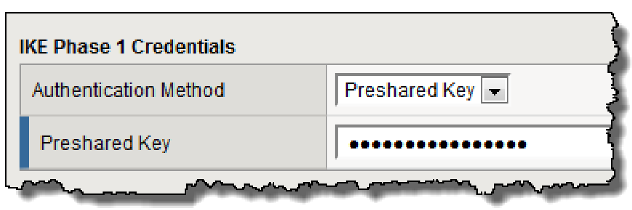


e) For the IKE Phase 1 Algorithms area, retain the default values, or select the options that are appropriate for your deployment.

f) For the IKE Phase 1 Credentials area, for the **Authentication Method** setting, select either **Preshared Key** or **RSA Signature**, and specify additional information in the fields that appear.

   For example, if you select **Preshared Key**, type the key in the **Preshared Key** field that becomes available.

   In this example, **Preshared Key** is selected.



   Note: *The key you type must be the same at both ends of the tunnel.*

g) From the **NAT Traversal** list, select **On** for Site A's IKE peer.

   Note: *Use this setting only for the IKE peer (remote BIG-IP system) that is behind a NAT device. On the Site B BIG-IP system, for the IKE peer, retain the default setting,* ***Off****.*

h) Click **Finished**.

2. Create a custom IPsec policy that uses Tunnel mode and has the same remote IP address as the IKE peer.
   a) On the Main tab, click **Network** > **IPsec** > **IPsec Policies**.
   b) Click the **Create** button.
   c) In the **Name** field, type a unique name for the policy.
   d) For the **IPsec Protocol** setting, retain the default selection, **ESP**.
   e) From the **Mode** list, select **Tunnel**.
      The screen refreshes to show additional related settings.
   f) In the **Tunnel Local Address** field, type the local IP address of the system you are configuring.
      For example, the tunnel local addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Local Address
Site A	`198.50.100.3`
Site B	`10.102.20.5`

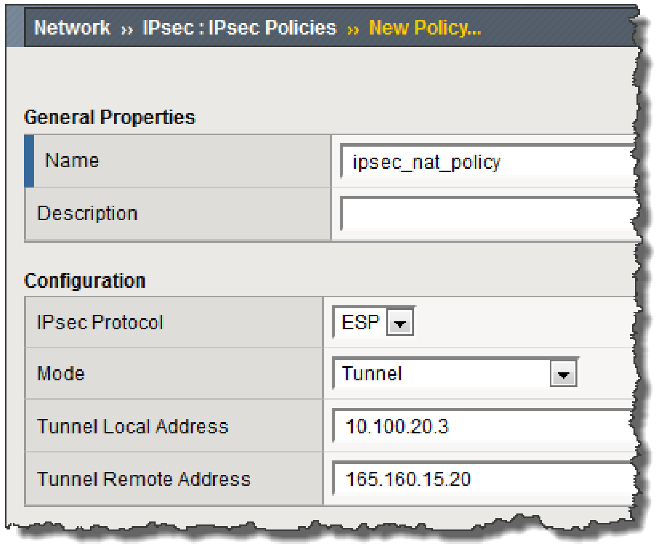
   g) In the **Tunnel Remote Address** field, type the IP address of the remote peer.
      If the remote BIG-IP system is behind a firewall or other NAT device, type the public IP address of that device.
      If the remote BIG-IP system is reachable directly, type the IP address of the BIG-IP system.

***Note:** This address must match the value of the **Remote Address** setting in the relevant IKE peer.*

For example, the tunnel remote addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Tunnel Remote Address
Site A	`165.160.15.20`
Site B	`198.50.100.3`

This screen snippet shows the tunnel settings at Site A.



h) For the **Authentication Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
i) For the **Encryption Algorithm** setting, retain the default value, or select the algorithm appropriate for your deployment.
j) For the **Perfect Forward Secrecy** setting, retain the default value, or select the option appropriate for your deployment.
k) Click **Finished**.

3. Create a bidirectional traffic selector that uses the custom IPsec policy you created.

   The traffic selector filters the application traffic based on the source and destination IP addresses you specify.

   a) On the Main tab, click **Network** > **IPsec** > **Traffic Selectors**.
   b) Click **Create**.
   c) In the **Name** field, type a unique name for the traffic selector.
   d) For the **Order** setting, retain the default value (**First**).
   e) For the **Source IP Address** setting, in the **Address** field, type the IP address from which the application traffic originates.
   In the illustration the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

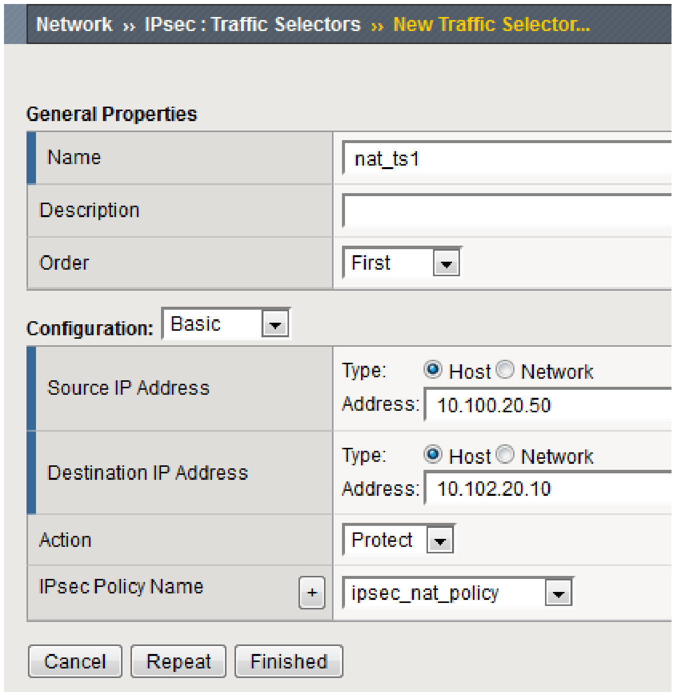
Location	Source IP Address
Site A	`10.100.20.50`
Site B	`10.102.20.10`

   f) For the **Destination IP Address** setting, in the **Address** field, type the final IP address for which the application traffic is destined.
   In the illustration, the source IP addresses for the BIG-IP systems in Site A and Site B are as follows.

Location	Destination IP Address
Site A	`10.102.20.10`
Site B	`10.100.20.50`

   g) For the **Action** setting, retain the default value, **Protect**.
   h) From the **IPsec Policy Name** list, select the name of the custom IPsec policy that you just created.

This screen snippet is an example of the completed Traffic Selector screen at Site A.



i) Click **Finished**.

You have now created an IPsec tunnel through which traffic travels in both directions across the WAN, and through a firewall on one side.

Task summary

Verifying IPsec connectivity for Tunnel mode

After you have configured an IPsec tunnel and before you configure additional functionality, you can verify that the tunnel is passing traffic.

Note: *Only data traffic matching the traffic selector triggers the establishment of the tunnel.*

.

1. Access the `tmsh` command-line utility.
2. Before sending traffic, type this command at the prompt.

   `tmsh modify net ipsec ike-daemon ikedaemon log-level info`

   This command increases the logging level to display the `INFO` messages that you want to view.
3. Send data traffic to the destination IP address specified in the traffic selector.
4. Check the IKE Phase 1 negotiation status by typing this command at the prompt.

   `racoonctl -l show-sa isakmp`

   This example shows a result of the command. `Destination` is the tunnel remote IP address.

```
Destination        Cookies                 ST  S   V E Created              Phase2
165.160.15.20.500  98993e6 . . . 22c87f1   9  I  10 M 2012-06-27 16:51:19      1
```

This table shows the legend for interpreting the result.

Column	Displayed	Description
ST (Tunnel Status)	1	Start Phase 1 negotiation
	2	msg 1 received
	3	msg 1 sent
	4	msg 2 received
	5	msg 2 sent
	6	msg 3 received
	7	msg 3 sent
	8	msg 4 received
	9	isakmp tunnel established
	10	isakmp tunnel expired
S	I	Initiator
	R	Responder
V (Version Number)	10	ISAKMP version 1.0
E (Exchange Mode)	M	Main (Identity Protection)
	A	Aggressive
Phase2	*<n>*	Number of Phase 2 tunnels negotiated with this IKE peer

5. Check the IKE Phase 2 negotiation status by typing this command at the prompt.

   `racoonctl -ll show-sa internal`

   This example shows a result of this command. `Source` is the tunnel local IP address. `Destination` is the tunnel remote IP address.

```
Source           Destination          Status         Side
10.100.20.3      165.160.15.20        sa established [R]
```

This table shows the legend for interpreting the result.

Column	Displayed
Side	I (Initiator)
	R (Responder)
Status	init
	start
	acquire

Column	Displayed
	getspi sent
	getspi done
	1st msg sent
	1st msg recvd
	commit bit
	sa added
	sa established
	sa expired

6. To verify the establishment of dynamic negotiated Security Associations (SAs), type this command at the prompt.

`tmsh show net ipsec ipsec-sa`

For each tunnel, the output displays IP addresses for two IPsec SAs, one for each direction, as shown in the example.

```
IPsec::SecurityAssociations
10.100.20.3 -> 165.160.15.20 SPI(0x7b438626) in esp (tmm: 6)
165.160.15.20 -> 10.100.20.3 SPI(0x5e52a1db) out esp (tmm: 5)
```

7. To display the details of the dynamic negotiated Security Associations (SAs), type this command at the prompt.

`tmsh show net ipsec ipsec-sa all-properties`

For each tunnel, the output displays the details for the IPsec SAs, as shown in the example.

```
IPsec::SecurityAssociations
165.160.15.20 -> 10.100.20.3
---------------------------------------------------------------------------

  tmm: 2
  Direction: out;  SPI: 0x6be3ff01(1810104065);  ReqID: 0x9b0a(39690)
  Protocol: esp;  Mode: tunnel;  State: mature
  Authenticated Encryption : aes-gmac128
  Current Usage: 307488 bytes
  Hard lifetime: 94 seconds; unlimited bytes
  Soft lifetime: 34 seconds; unlimited bytes
  Replay window size: 64
  Last use: 12/13/2012:10:42                     Create:  12/13/2012:10:39
```

8. To filter the Security Associations (SAs) by traffic selector, type this command at the prompt.

`tmsh show net ipsec ipsec-sa traffic-selector ts_codec`

You can also filter by other parameters, such as SPI (`spi`), source address (`src_addr`), or destination address (`dst_addr`)

The output displays the IPsec SAs that area associated with the traffic selector specified, as shown in the example.

```
IPsec::SecurityAssociations
10.100.115.12  ->  10.100.15.132  SPI(0x2211c0a9)  in  esp  (tmm: 0)
10.100.15.132  ->  10.100.115.12  SPI(0x932e0c44)  out  esp  (tmm: 2)
```

9. Check the IPsec stats by typing this command at the prompt.

   `tmsh show net ipsec-stat`

   If traffic is passing through the IPsec tunnel, the stats will increment.

```
--------------------------------------------------------------------
Net::Ipsec
Cmd Id           Mode  Packets In  Bytes In  Packets Out  Bytes Out
--------------------------------------------------------------------
0           TRANSPORT           0         0            0          0
0           TRANSPORT           0         0            0          0
0              TUNNEL           0         0            0          0
0              TUNNEL           0         0            0          0
1              TUNNEL      353.9K    252.4M        24.9K       1.8M
2              TUNNEL      117.9K     41.0M       163.3K      12.4M
```

10. If the SAs are established, but traffic is not passing, type this command at the prompt.

    `tmsh delete net ipsec ipsec-sa`

    This action deletes the IPsec tunnels. Sending new traffic triggers SA negotiation and establishment.
11. If traffic is still not passing, type this command at the prompt.

    `racoonctl flush-sa isakmp`

    This action brings down the control channel. Sending new traffic triggers SA negotiation and establishment.
12. View the `/var/log/racoon.log` to verify that the IPsec tunnel is up.

    These lines are examples of the messages you are looking for.

```
2012-06-29 16:45:13: INFO: ISAKMP-SA established
10.100.20.3[500]-165.160.15.20[500] spi:3840191bd045fa51:673828cf6adc5c61
2012-06-29 16:45:14: INFO: initiate new phase 2 negotiation:
10.100.20.3[500]<=>165.160.15.20[500]
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
165.160.15.20[0]->10.100.20.3[0] spi=2403416622(0x8f413a2e)
2012-06-29 16:45:14: INFO: IPsec-SA established: ESP/Tunnel
10.100.20.3[0]->165.160.15.20[0] spi=4573766(0x45ca46
```

13. For protocol-level troubleshooting, you can increase the debug level by typing this command at the prompt.

    `tmsh modify net ipsec ike-daemon ikedaemon log-level debug2`

    Important: *Use this command only for debugging. It creates a large log file, and can slow the tunnel negotiation.*

Note: *Using this command flushes existing SAs.*

14. After you view the results, return the debug level to normal to avoid excessive logging by typing this command at the prompt.

```
tmsh modify net ipsec ike-daemon ikedaemon log-level info
```

Note: *Using this command flushes existing SAs.*

Configuring Remote High-Speed Logging

Overview: Configuring high-speed remote logging of BIG-IP system processes

You can configure the BIG-IP® system to log information about BIG-IP system processes and send the log messages to remote high-speed log servers. You can filter the data that the system logs based on alert-level and source.

When configuring remote high-speed logging of BIG-IP system processes, it is helpful to understand the objects you need to create and why, as described here:

Object to create in implementation	Reason
Pool of remote log servers	Create a pool of remote log servers to which the BIG-IP system can send log messages.
Destination (unformatted)	Create a log destination of Remote High-Speed Log type that specifies a pool of remote log servers.
Destination (formatted)	If your remote log servers are the ArcSight, Splunk, IPFIX, or Remote Syslog type, create an additional log destination to format the logs in the required format and forward the logs to a remote high-speed log destination.
Publisher	Create a log publisher to send logs to a set of specified log destinations.
Filter	Create a log filter to define the messages to be included in the BIG-IP system logs and associate a log publisher with the filter.

This illustration shows the association of the configuration objects for remote high-speed logging of BIG-IP system processes.

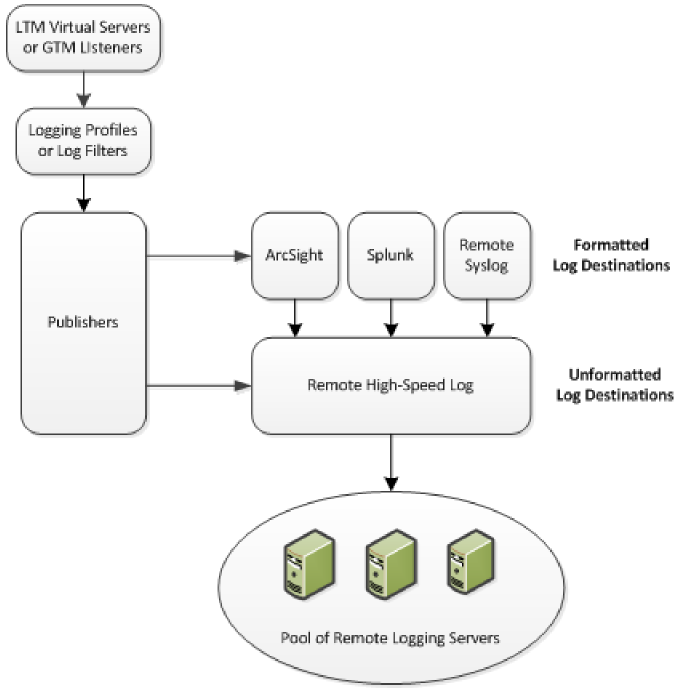


Figure 32: Association of remote high-speed logging configuration objects

Task summary

Perform these tasks to configure BIG-IP® system logging.

Note: *Enabling remote high-speed logging impacts BIG-IP system performance.*

Creating a pool of remote logging servers
Creating a remote high-speed log destination
Creating a formatted remote high-speed log destination
Creating a publisher
Creating a logging filter
Disabling system logging
Troubleshooting logs that contain unexpected messages

Creating a pool of remote logging servers

Before creating a pool of log servers, gather the IP addresses of the servers that you want to include in the pool. Ensure that the remote log servers are configured to listen to and receive log messages from the BIG-IP® system.

Create a pool of remote log servers to which the BIG-IP system can send log messages.

1. On the Main tab, click **DNS** > **Delivery** > **Load Balancing** > **Pools** or **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Using the **New Members** setting, add the IP address for each remote logging server that you want to include in the pool:

a) Type an IP address in the **Address** field, or select a node address from the **Node List**.
b) Type a service number in the **Service Port** field, or select a service name from the list.

Note: *Typical remote logging servers require port* `514`.

c) Click **Add**.

5. Click **Finished**.

Creating a remote high-speed log destination

Before creating a remote high-speed log destination, ensure that at least one pool of remote log servers exists on the BIG-IP® system.

Create a log destination of the **Remote High-Speed Log** type to specify that log messages are sent to a pool of remote log servers.

1. On the Main tab, click **System** > **Logs** > **Configuration** > **Log Destinations**.
   The Log Destinations screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique, identifiable name for this destination.
4. From the **Type** list, select **Remote High-Speed Log**.

   Important: *If you use log servers such as Remote Syslog, Splunk, or ArcSight, which require data be sent to the servers in a specific format, you must create an additional log destination of the required type, and associate it with a log destination of the* ***Remote High-Speed Log*** *type. With this configuration, the BIG-IP system can send data to the servers in the required format.*

   The BIG-IP system is configured to send an unformatted string of text to the log servers.
5. From the **Pool Name** list, select the pool of remote log servers to which you want the BIG-IP system to send log messages.
6. From the **Protocol** list, select the protocol used by the high-speed logging pool members.
7. Click **Finished**.

Creating a formatted remote high-speed log destination

Ensure that at least one remote high-speed log destination exists on the BIG-IP® system.

Create a formatted logging destination to specify that log messages are sent to a pool of remote log servers, such as Remote Syslog, Splunk, or ArcSight servers.

1. On the Main tab, click **System** > **Logs** > **Configuration** > **Log Destinations**.
   The Log Destinations screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique, identifiable name for this destination.
4. From the **Type** list, select a formatted logging destination, such as **IPFIX**, **Remote Syslog**, **Splunk**, or **ArcSight**.

Important: *ArcSight formatting is only available for logs coming from Advanced Firewall Manager (AFM), Application Security Manager (ASM™), and the Secure Web Gateway component of Access Policy Manager® (APM®). IPFIX is not available for Secure Web Gateway.*

The BIG-IP system is configured to send a formatted string of text to the log servers.

5. If you selected **Remote Syslog**, from the **Syslog Format** list, select a format for the logs, and then from the **High-Speed Log Destination** list, select the destination that points to a pool of remote Syslog servers to which you want the BIG-IP system to send log messages.
6. If you selected **Splunk** or **IPFIX**, from the **Forward To** list, select the destination that points to a pool of high-speed log servers to which you want the BIG-IP system to send log messages.
7. Click **Finished**.

Creating a publisher

Ensure that at least one destination associated with a pool of remote log servers exists on the BIG-IP® system.

Create a publisher to specify where the BIG-IP system sends log messages for specific resources.

1. On the Main tab, click **System** > **Logs** > **Configuration** > **Log Publishers**.
   The Log Publishers screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique, identifiable name for this publisher.
4. For the **Destinations** setting, select a destination from the **Available** list, and click **<<** to move the destination to the **Selected** list.

   Note: *If you are using a formatted destination, select the destination that matches your log servers, such as Remote Syslog, Splunk, or ArcSight.*

5. Click **Finished**.

Creating a logging filter

Ensure that at least one log publisher is configured on the BIG-IP® system.

Create a custom log filter to specify the system log messages that you want to publish to a particular log.

1. On the Main tab, click **System** > **Logs** > **Configuration** > **Log Filters**.
   The Log Filters screen opens.
2. In the **Name** field, type a unique, identifiable name for this filter.
3. From the **Severity** list, select the level of alerts that you want the system to use for this filter.

   Note: *The severity level that you select includes all of the severity levels that display above your selection in the list. For example, if you select **Emergency**, the system publishes only emergency messages to the log. If you select **Critical**, the system publishes critical, alert, and emergency-level messages in the log.*

4. From the **Source** list, select the system processes from which messages will be sent to the log.

5. In the **Message ID** field, type the first eight hex-digits of the specific message ID that you want the system to include in the log. Use this field when you want a log to contain only each instance of one specific log message.

   Note: *BIG-IP system log messages contain message ID strings in the format:* `xxxxxxxx:x:`*. For example, in this log message:* `Oct 31 11:06:27 olgavmmgmt notice mcpd[5641]: 01070410:5: Removed subscription with subscriber id lind` *, the message ID string is:* `01070410:5:`*. You enter only the first eight hex-digits:* `01070410`*.*

   ---

6. From the **Log Publisher** list, select the publisher that includes the destinations to which you want to send log messages.
7. Click **Finished**.

Disabling system logging

When you no longer want the BIG-IP® system to log information about its internal systems, you can delete the log filter that you created. For example, when mitigating a DoS attack, if you created a log filter that includes only one specific message in the log, you can delete that log filter once you handle the attack.

1. On the Main tab, click **System** > **Logs** > **Configuration** > **Log Filters**.
   The Log Filters screen opens.
2. Select the check box next to the name of the log filter that you want to delete. Click **Delete**, and then click **Delete** again.

Troubleshooting logs that contain unexpected messages

If you configured a filter to send all instances of a specific message ID to your remote logging servers and this message ID is still displaying in the local log in the BIG-IP system, you can disable legacy log message processing in order to display instances of this message ID only on the remote logging servers.

Important: *When you create a filter that disables legacy log message processing, the legacy logs are completely disabled. Therefore, you must also create a filter for every source from which you want log messages to be sent to the pool of remote log servers.*

---

1. On the Main tab, click **System** > **Logs** > **Configuration** > **Log Filters**.
   The Log Filters screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique, identifiable name for this filter.
4. From the **Severity** list, select **Debug**.
5. From the **Source** list, select **All**.
6. From the **Log Publisher** list, select **None**.
7. Click **Finished**.

Setting Up Secure Remote Logging

Introduction to secure logging configuration

The BIG-IP® system can securely log messages using Transport Layer Security (TLS) encryption to a secure syslog server that resides on a shared, external network. This implementation describes a sample configuration consisting of two BIG-IP systems, in a Device Service Clustering (DSC®) Sync-Only or Sync-Failover device group, that encrypt log messages using a local virtual server before sending the messages on to the remote secure syslog server.

In the example, the BIG-IP systems (`bigip1.syslog.secure.com` and `bigip2.syslog.secure.com`) and the secure syslog server (`server.syslog.secure.com`) mutually authenticate each other using X.509 certificates and keys on their TLS connections. This certificate validation requires a dedicated certificate for each BIG-IP system's logging interface (the self IP address on the logging VLAN for that BIG-IP system) and a certificate for the secure syslog server. In this sample configuration, all three certificates are signed by the same Certificate Authority (CA) and each have the same CA certificate bundle installed, to be used for X.509 certificate validation. The configuration is based on the assumption that you have configured an external Domain Name System (DNS) server with forward and reverse DNS entries for the names and IP addresses used in the X.509 certificate authentication.

In most configurations, the shared, external network should be deployed as a dedicated VLAN connecting only the BIG-IP systems and secure syslog server, due to the potential for high-bandwidth logging from the High Speed Logging (HSL) subsystem.

Note: *Some BIG-IP software versions do not include the HSL subsystem. If the BIG-IP systems in your device group do not include HSL, you can still configure secure logging to a remote syslog server. In this case, as long as you can configure the local syslog service to direct messages to the local log encrypting virtual server, the secure logging configuration supports the encrypting of messages from the local syslog service.*

Sample secure logging configuration

This illustration shows an example of the entire secure logging configuration. The logging traffic proceeds from top to bottom in the illustration.

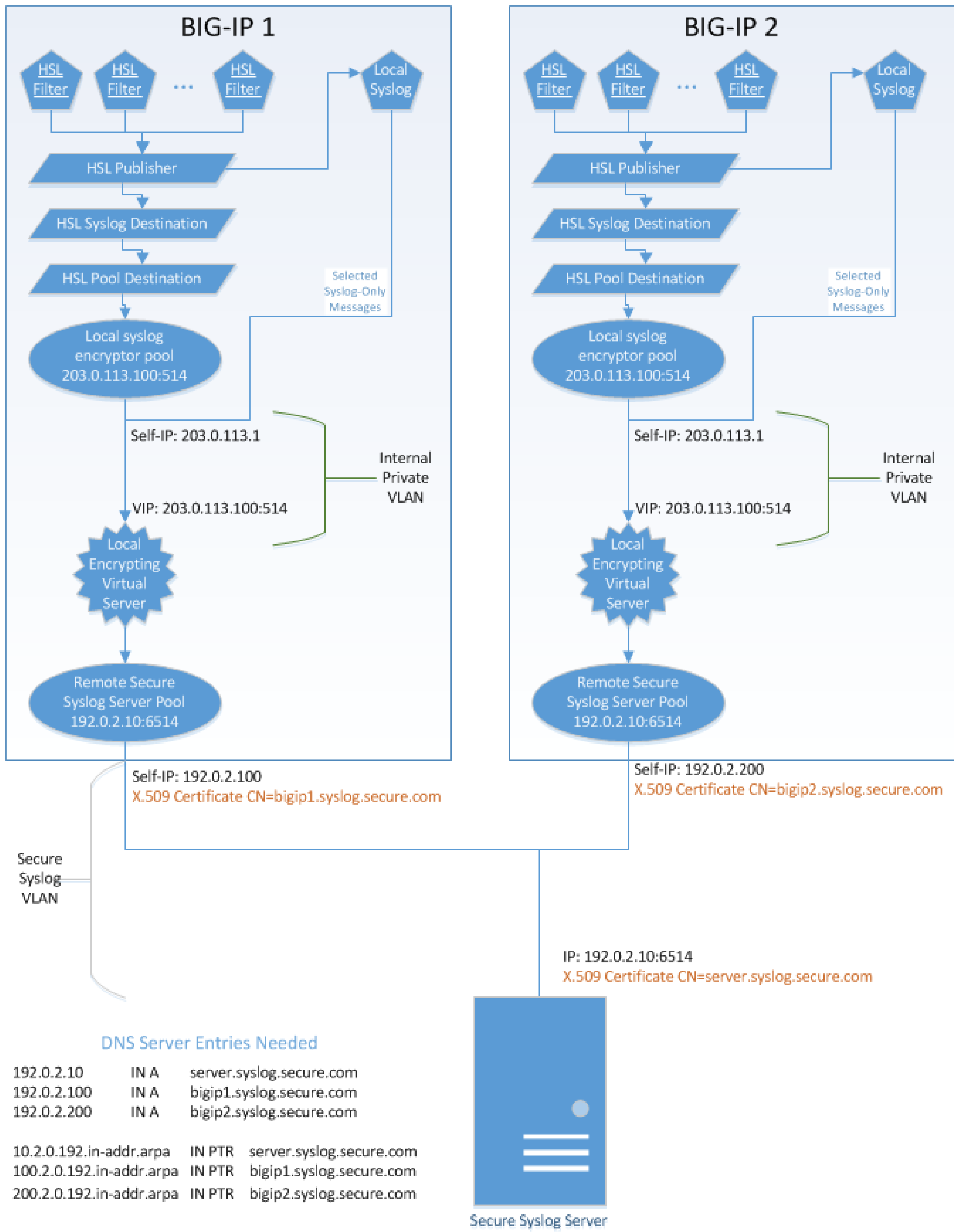


Figure 33: High-level secure logging configuration

In the example:

- Each BIG-IP® system has one or more HSL filters directing certain kinds of log messages to an HSL destination. The HSL destination forwards the messages to both the local syslog server (for local log retention, in case the external syslog server is unreachable), and an HSL syslog destination, whose purpose is to add the timestamp and other information expected by RFC5424-compliant syslog servers. The HSL syslog destination then sends the decorated log messages to an HSL pool destination, which

directs them to the local syslog encryptor pool containing the IP address of a local encrypting virtual server.
- The two BIG-IP systems include identically-configured local syslog encrypting virtual servers. The virtual servers are configured using a non-floating IP address on a private VLAN that is internal to each BIG-IP system, with no external interfaces attached. This VLAN exists solely to provide a private communications link between the local syslog encryptor pool, the local syslog server, and the local encrypting virtual server. For messages that are not currently processed by the HSL subsystem, the local syslog server uses this VLAN to send selected messages directly to the local encrypting virtual server, to be encrypted and sent on to the remote secure syslog server.
- The local encrypting virtual server is configured with a Server SSL profile for the purpose of sending the BIG-IP system's client certificate to the server for X.509 validation, as well as for validating the server's X.509 certificate using a locally-installed CA certificate bundle. Once authenticated and connected to the server listed in the remote secure syslog server pool, the local syslog encrypting virtual server sends the outbound encrypted syslog messages to the remote syslog server. The outbound TCP sessions are retained for subsequent syslog messages until the TCP timeout on the virtual server expires; then the next syslog message initiates a new TCP session.

The result is that when the high speed logging subsystem or the standard syslog service of either BIG-IP system sends TCP syslog traffic, the messages are forwarded to the remote syslog server over an authenticated and encrypted, secure channel.

Important: *In this implementation, you must configure the objects shown in the illustration by starting with those at the bottom and then proceeding toward the top. This ensures that configuration objects are available when needed to configure other objects.*

Prerequisite tasks

Before configuring secure logging, you must perform these tasks on the BIG-IP® systems in the configuration.

Table 12: Prerequisite tasks

Task	Description
Create a device group.	The Device Service Clustering (DSC®) device group must contain the BIG-IP® systems as members. You perform this task on only one device in the device group.
Enable Automatic Sync on the device group.	Enabling automatic sync for the device group ensures that every change you make to a BIG-IP system is internally propagated to all device group members. In most cases, this eliminates the need to manually sync configuration changes to the peer device. You perform this task on only one device in the device group, and the change is propagated to the other device.
Assign fully-qualified domain names (FQDNs).	Each BIG-IP system in the device group, and the remote, secure syslog server, must have a unique fully-qualified domain name (FQDN). In our example, these FQDNs are: `bigip1.syslog.secure.com`, `bigip2.syslog.secure.com`, and `server.syslog.secure.com`.
Specify the DNS name server.	You must specify an external Domain Name System (DNS) server with forward and reverse DNS entries for the names and IP addresses used in the X.509 certificate authentication. Once configured, the DNS server resolves the FQDN used in the X.509 certificate for each device's secure logging configuration to the IP address on the logging VLAN for that device. You must perform this task on each BIG-IP device in the device group.

About X.509 certificates for secure logging

One of the required elements of the secure logging configuration is the mutual validation of the X.509 certificate for each device in the configuration (that is, each BIG-IP® device, as well as the secure logging server). Each device must have a valid X.509 certificate and key assigned, where the `Common Name` attribute of the certificate resolves to the Fully Qualified Domain Name (FQDN) of that device's IP address on the shared secure logging VLAN. For the certificate on each of the two BIG-IP systems, this IP address is a self IP address. For the certificate of the secure, remote syslog server, this IP address is the IP address of that server.

For either BIG-IP system to successfully validate the certificate of the other device, all X.509 certificates must be signed by a parent certificate authority (CA) whose certificate chain is included in the certificate bundle referenced in the SSL profile of each of the BIG-IP encrypting virtual servers. The CA's certificate chain must also be included in the certificate bundle of the secure syslog server's configuration.

Task summary

You must perform several tasks to create a BIG-IP® system configuration that performs secure logging to a remote syslog server. Each of the tasks in this document is based on the sample configuration shown in Figure 1.

Note: *When entering* `tmsh` *commands, enter the commands as a single command line; the examples shown include newlines for readability only. Also, ensure that you perform the tasks in the order presented.*

Task List

Importing an X.509 certificate, key, and CA bundle

To ensure that secure logging operates successfully, you must import the required certificate, key, and CA bundle to the local BIG-IP® device.

Important: *Perform this task on each device in the device group.*

---

1. On the Main tab, click **System** > **Device Certificates**.
   The Device Certificate screen opens.
2. Click **Import**.
3. From the **Import Type** list, select **Certificate and Key**.
4. For the **Certificate Source** setting, select **Upload File** and browse to select the certificate signed by the CA server.
5. For the **Key Source** setting, select **Upload File** and browse to select the device key file.
6. Click **Import**.

Creating a pool containing the syslog server

On either of the BIG-IP® systems in the device group, use the Traffic Management Shell (`tmsh`) to create a pool containing the IP address and TCP port number of the logging network interface on the remote syslog server.

1. At the `tmsh` prompt, create a pool containing a remote syslog server. For example:

   ```
   create ltm pool pool_remote_secure_syslog {
      members replace-all-with { 192.0.2.10:6514 { address 192.0.2.10 } }
      monitor tcp_half_open
   }
   ```

   In this example, `192.0.2.10:6514` represents the IP address of the remote syslog server.
2. Save the configuration by typing `save /sys config`.

Configuring system BIG-IP 1

Before you perform this task, verify that you have created a one-member pool containing the remote syslog server.

The main goal of this task is to create a virtual server and associated objects on one of the two BIG-IP® systems (in the example, a system named `bigip1.syslog.secure.com`) that encrypts server-side traffic destined for the remote syslog server. This encrypting virtual server is on an internal, private VLAN and is associated with a non-floating virtual address, using the local BIG-IP system's key and certificate. You also use this task to create a shared, external VLAN and an associated self IP address. This is the VLAN with which the remote syslog server is associated.

The encrypting virtual server that you create has the same destination address and port as the encrypting virtual server that you create on the peer system (in the example, `bigip2.syslog.secure.com`). Also, the virtual server targets the same pool as the peer system (the pool containing the remote syslog server).

Note: *Perform all steps in this task at the* `tmsh` *prompt.*

1. Create an SSL Server profile to encrypt traffic destined for the syslog server pool. For example:

   ```
   create ltm profile server-ssl profile_serverssl_syslog-1 {
      ca-file F5secureLoggingCA_bundle.crt
      cert b3-1.logging.f5cc.com.crt
      defaults-from serverssl
      key b3-1.logging.f5cc.com.key
      peer-cert-mode require
   }
   ```

   In this example, `profile_serverssl_syslog-1` represents the name of the Server SSL profile.

   Important: *The certificate bundle that you specify must include the certificate chain of the certificate authority.*
2. Create a VLAN on the private, internal network, with no interfaces assigned. For example: `create net vlan vlan_securelog`.

3. Create a self IP address in the traffic group `traffic-group-local-only` and associate it with VLAN `vlan_securelog`. For example: `create net self 203.0.113.1/24 vlan vlan_securelog` .

   Important: *The IP address that you specify must be a non-routable address and must be identical on all BIG-IP systems in the configuration.*

4. Create a non-floating virtual address on the private, internal network. For example:

```
create ltm virtual-address 203.0.113.100
   traffic-group traffic-group-local-only
   auto-delete false
```

   Important: *You must use `tmsh` to create the virtual address, and you must create the virtual address prior to creating the associated virtual server. Also, the IP address you specify must be the same virtual address that you specify on the peer BIG-IP system.*

5. Create a virtual server network for the virtual address, assigning the pool, SSL Server profile, and private VLAN. For example:

```
create ltm virtual vs_secure_syslog_target-1 {
   destination 203.0.113.100:514
   ip-protocol tcp
   pool pool_remote_secure_syslog
   profiles replace-all-with { profile_serverssl_syslog-1 tcp }
   vlans replace-all-with { vlan_securelog }
   vlans-enabled
```

   Important: *In this example, `vs_secure_syslog_target-1` represents the name of the virtual server, and the destination IP address is `203.0.113.100:514`. The destination IP address and port that you specify must be the same destination IP address and port that you specify on the peer BIG-IP system.*

6. Create a VLAN on the shared, external network with all appropriate BIG-IP interfaces assigned. For example: `create net vlan vlan_logging { tag 4089 interfaces add { 1.1 {tagged} } }.`
7. Create a self IP address in the traffic group `traffic-group-local-only` and associate it with VLAN `vlan_logging`. For example: `create net self 192.0.2.100 vlan vlan_logging.`

After you perform this task, system `bigip1.syslog.secure.com` contains a virtual server that references a Server SSL profile, a private, internal VLAN, and the pool containing the remote syslog server. The virtual server destination IP address and port match those of the virtual server on system `bigip2.syslog.secure.com`. System `bigip1.syslog.secure.com` also contains a shared, external VLAN with an associated self IP address.

Configuring system BIG-IP 2

Before you perform this task, verify that you have created a one-member pool containing the remote syslog server.

The main goal of this task is to create a virtual server and associated objects on one of the two BIG-IP® systems (in the example, a system named `bigip2.syslog.secure.com`) that encrypts server-side traffic destined for the remote syslog server. This encrypting virtual server is on an internal, private VLAN and is associated with a non-floating virtual address, using the local BIG-IP system's key and certificate. You also use this task to create a shared, external VLAN and an associated self IP address. This is the VLAN with which the remote syslog server is associated.

The encrypting virtual server has the same destination address and port as the encrypting virtual server that you create on the peer system (in the example, `bigip1.syslog.secure.com`). Also, the virtual server targets the same pool as the peer system (the pool containing the remote syslog server).

Note: *Perform all steps in this task at the* `tmsh` *prompt.*

---

1. Create an SSL Server profile to encrypt traffic destined for the syslog server pool. For example:

```
create ltm profile server-ssl profile_serverssl_syslog-2 {
   ca-file F5secureLoggingCA_bundle.crt
   cert b3-2.logging.f5cc.com.crt
   defaults-from serverssl
   key b3-2.logging.f5cc.com.key
   peer-cert-mode require
}
```

   In this example, `profile_serverssl_syslog-2` represents the name of the Server SSL profile.

   Important: *The certificate bundle that you specify must include the certificate chain of the certificate authority.*

2. Create a VLAN on the private, internal network, with no interfaces assigned. For example: `create net vlan vlan_securelog`.
3. Create a self IP address in the traffic group `traffic-group-local-only` and associate it with the VLAN. For example: `create net self 203.0.113.1/24 vlan vlan_securelog`.

   Important: *The IP address that you specify must be a non-routable address and must be identical on all BIG-IP systems in the configuration.*

4. Create a non-floating virtual address on the private, internal network. For example:

```
create ltm virtual-address 203.0.113.100
   traffic-group traffic-group-local-only
   auto-delete false
```

   Important: *You must use* `tmsh` *to create the virtual address, and you must create the virtual address prior to creating the associated virtual server. Also, the IP address you specify must be the same virtual address that you specify on the peer BIG-IP system.*

5. Create a virtual server for the virtual address, assigning the pool, SSL Server profile, and private VLAN. For example:

```
create ltm virtual vs_secure_syslog_target-2 {
   destination 203.0.113.100:514
   ip-protocol tcp
   pool pool_remote_secure_syslog
   profiles replace-all-with { profile_serverssl_syslog-2 tcp }
```

```
vlans replace-all-with { vlan_securelog }
vlans-enabled
```

In this example, `vs_secure_syslog_target-2` represents the name of the virtual server, and the destination IP address is `203.0.113.100:514`. The destination IP address and port that you specify must be the same destination IP address and port that you specify on the peer BIG-IP system.

6. Create a VLAN on the shared, external network with all appropriate BIG-IP interfaces assigned. For example: `create net vlan vlan_logging { tag 4089 interfaces add { 1.1 {tagged} } }`.
7. Create a self IP address in the traffic group `traffic-group-local-only` and associate it with VLAN `vlan_logging`. For example: `create net self 192.0.2.200 vlan vlan_logging`.

After you perform this task, system `bigip2.syslog.secure.com` contains a virtual server that references a Server SSL profile, a private, internal VLAN, and the pool containing the remote syslog server. The virtual server destination IP address and port match those of the virtual server on system `bigip1.syslog.secure.com`. System `bigip2.syslog.secure.com` also contains a shared, external VLAN with an associated self IP address.

Modifying the local syslog server

Because some of the older audit log messages do not use the high-speed logging (HSL) system, you must modify the BIG-IP® system's local syslog server to send audit data to one of the encrypting virtual servers.

Note: *You can perform this task on either one of the BIG-IP systems in the device group.*

At the `tmsh` prompt, modify the syslog server to create a destination that targets the IP address and port number of the local encrypting virtual server. For example:

```
modify sys syslog {
   include "
      destination d_to_secure_syslog { tcp( 203.0.113.100 port(514)); };
      log { source(s_syslog_pipe); filter(f_audit);
destination(d_to_secure_syslog); };
      log { source(s_syslog_pipe); filter(f_authpriv);
destination(d_to_secure_syslog); };
      log { source(s_syslog_pipe); filter(f_apm);
destination(d_to_secure_syslog); };
      log { source(s_syslog_pipe); filter(f_sso);
destination(d_to_secure_syslog); };
   "
}
```

In this example, `d_to_secure_syslog` represents the name of the HSL destination, which targets the local syslog destination, which targets the local encrypting virtual server's destination IP address and port `203.0.113.100:514`.

Creating a pool for the local encrypting virtual server

For the High-Speed Logging (HSL) system, you must create a pool containing the IP address and TCP port of the encrypting virtual servers. This pool becomes the target pool for the HSL pool destination.

***Note:** You can perform this task on either one of the BIG-IP® systems in the device group.*

---

1. At the `tmsh` prompt, create a pool with the address and port of the encrypting virtual servers as the pool member. For example:

```
create ltm pool pool_syslog_encryptor {
   members replace-all-with {
      203.0.113.100:514 { address 203.0.113.100 }
   }
   monitor tcp_half_open
}
```

   In this example, `pool_syslog_encryptor` represents the name of the pool that contains pool member `203.0.113.100:514`.
2. Save the configuration by typing `save /sys config`.

Creating an HSL destination targeting the encrypting pool

You must create a `remote high-speed log` destination that targets the local encrypting syslog pool. This pool contains a single pool member, which is the destination IP address and port of the encrypting virtual server on each BIG-IP® system.

***Note:** You can perform this task on either one of the BIG-IP systems in the device group.*

At the `tmsh` prompt, create a `remote high-speed log` destination. For example:

```
create sys log-config destination remote-high-speed-log hsldest_to_encryptor
 {
  pool-name pool_syslog_encryptor
}
```

In this example, a `remote high-speed log` destination named `hsldest_to_encryptor` targets the local encrypting syslog pool named `pool_syslog_encryptor`.

Creating an RFC 5424 (syslog) HSL destination

To ensure that the syslog timestamp and other identifying information is included with each log message, you must create a formatted `remote-syslog` destination that targets the `remote high-speed log` destination.

***Note:** You can perform this task on either one of the BIG-IP® systems in the device group.*

At the `tmsh` prompt, create a `remote-syslog` destination.

```
create sys log-config destination remote-syslog hsldest_syslog {
  format rfc5424
```

```
  remote-high-speed-log hsldest_to_encryptor
}
```

In this example, a formatted `remote-syslog` destination named `hsldest_syslog` targets the `remote high-speed log` destination named `hsldest_to_encryptor`.

Creating an HSL publisher

You must create a high-speed logging (HSL) publisher, which sends the selected audit logging messages to both the local syslog server (for local logging) and the formatted `remote-syslog` destination.

Note: *You can perform this task on either one of the BIG-IP® systems in the device group.*

At the `tmsh` prompt, create the HSL publisher. For example::

```
create sys log-config publisher hslpub_secure_remote_syslog {
  destinations replace-all-with {
    hsldest_syslog
    local-syslog
  }
}
```

In this example, a publisher named `hslpub_secure_remote_syslog` targets the local syslog server named `local-syslog`, as well as the formatted `remote-syslog` destination named `hsldest_syslog`.

Creating HSL filters for log messages

You must create high-speed-logging (HSL) filters to select log messages and send the messages through the chain to the secure remote syslog server. Types of filters you can create are packet, SSL, `tamd`, and `tmsh`.

Note: *You can perform this task on either one of the BIG-IP® systems in the device group.*

---

1. At the `tmsh` prompt, create a packet filter. For example:

```
Create sys log-config filter hslfilter_packet_filter {
   publisher hslpub_secure_remote_syslog
   source packet_filter
}
```

2. Create an SSL filter. For example:

```
create sys log-config filter hslfilter_ssl {
  publisher hslpub_secure_remote_syslog
  source ssl
}
```

3. Create a `tamd` filter. For example:

```
create sys log-config filter hslfilter_tamd {
  publisher hslpub_secure_remote_syslog
  source tamd
}
```

4. Create a `tmsh`filter. For example:

```
create sys log-config filter hslfilter_tmsh {
  publisher hslpub_secure_remote_syslog
  source tmsh
}
```

Configuring APM logging (APM systems only)

If you are testing a system on which you have provisioned BIG-IP® Access Policy Manager® (APM®), (also known as ADC-AP), you must enable APM syslog logging and create additional high-speed logging (HSL) filters.

Note: *You can perform this task on either one of the BIG-IP systems in the device group.*

---

1. At the `tmsh` prompt, enable syslog logging for BIG-IP® Access Policy Manager® (APM®): `modify sys db log.access.syslog value enable`
2. Create an APM filter. For example:

```
create sys log-config filter remote_apm_filter {
  level info
  publisher hslpub_secure_remote_syslog
  source accesscontrol
}
```

3. Create an access control filter. For example:

```
create sys log-config filter remote_acl_filter {
  level info
  publisher hslpub_secure_remote_syslog
  source apmacl
}
```

4. Create a filter for single sign-on. For example:

```
create sys log-config filter remote_sso_filter {
  level info
  publisher hslpub_secure_remote_syslog
  source sso
}
```

Saving the secure logging configuration

After performing all tasks to configure secure logging on the BIG-IP® system, you must save the full secure logging configuration.

At the `tmsh` prompt, save the configuration by typing `save /sys config`.

Using Link Aggregation with Tagged VLANs for a One-network Topology

Overview: Configuring link aggregation using tagged VLANs on one network

You can use the BIG-IP® system in an aggregated two-interface load balancing topology. *Link aggregation* is the process of combining multiple links so that the links function as a single link with higher bandwidth. Aggregating multiple interfaces into a trunk to create a link has the following advantages:

- Link aggregation increases the bandwidth of the individual network interface cards (NICs) in an additive manner.
- If one link goes down, the other link can handle the traffic by itself.

Link aggregation occurs when you create a trunk. A *trunk* is a combination of two or more interfaces and cables configured as one link.

The examples in this implementation show a trunk that includes two tagged interfaces aggregated together. A *tagged interface* is an interface that is configured to process traffic for multiple VLANs. A VLAN tag identifies the specific VLAN and enables traffic to pass through that specific VLAN. To cause traffic for multiple VLANs to be passed through a single trunk, you must assign the same trunk to each VLAN.

In the example, we create a trunk (**trunk1**) that includes two interfaces, **1.1** and **1.2**, and then assign **trunk1** as a tagged interface to both VLAN **external** and VLAN **internal**. Both VLANs (**external** and **internal**) reside on the same network, and are combined to form a VLAN group.

With this configuration, inbound and outbound traffic passing between the BIG-IP system and the vendor switch can use either interface. For example, traffic destined for VLAN **externa**l can pass through either interface, **1.1** or **1.2**.

Illustration of link aggregation for a one-network topology

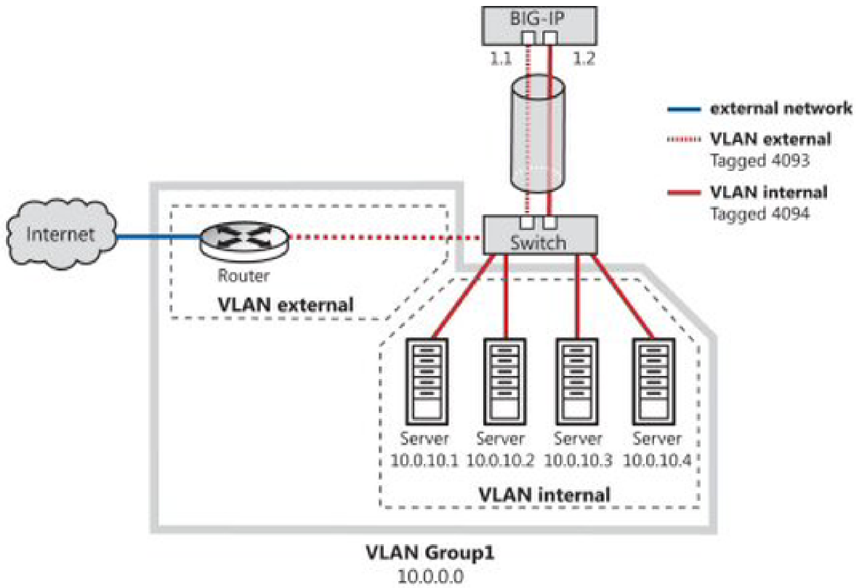


Figure 34: Link aggregation for a one-network topology

Task summary

Perform the following tasks to configure two interfaces (tagged VLANs) to function as a single link with higher bandwidth. In this implementation, you combine the two tagged VLANs into one VLAN group, where the two VLANs are on the same IP network.

Task list

Creating a trunk
Adding a tagged interface to a VLAN
Creating a load balancing pool
Creating a virtual server with source address affinity persistence
Removing the self IP addresses from the default VLANs
Creating a VLAN group
Creating a self IP for a VLAN group

Creating a trunk

You create a trunk on the BIG-IP® system so that the system can then aggregate the links to enhance bandwidth and ensure link availability.

1. On the Main tab, click **Network** > **Trunks**.
   The Trunk List screen opens.
2. Click **Create**.

3. Name the trunk.
4. For the **Interfaces** setting, in the **Available** field, select an interface, and using the Move button, move the interface to the **Members** field. Repeat this action for each interface that you want to include in the trunk.

   Trunk members must be untagged interfaces and cannot belong to another trunk. Therefore, only untagged interfaces that do not belong to another trunk appear in the **Available** list.
5. Select the **LACP** check box.
6. From the **Link Selection Policy** list, retain the default value, **Auto**.
7. From the **Frame Distribution Hash** list, select the default value, **Source/Destination IP address port**.

   Important: *On certain F5 platforms, packets can incorrectly egress on the same BIG-IP trunk member that the external switch ingressed the packets on. You can prevent this by configuring the external switch to use the same algorithm for its frame distribution hash value as you configure on the BIG-IP trunk. For example, if you configure the BIG-IP trunk to base the frame distribution hash value on both source and destination IP addresses, then you must configure the external switch to do the same.*
8. Click **Finished**.

After you create a trunk, the BIG-IP system aggregates the links to enhance bandwidth and prevent interruption in service.

Adding a tagged interface to a VLAN

After you aggregate the links, you assign the trunk to the VLAN as a tagged interface.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Perform the following for the **external** and **internal** VLANs:
   a) Click the VLAN name.
   b) For the **Interfaces** setting, in the **Available** field, select a trunk, and using the **Move** button, move the trunk that you created to the **Tagged** field.
   c) Click **Update**.

The trunk is assigned to the **external** and **internal** VLAN as a tagged interface.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.

4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

   Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.

6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.

7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type a port number in the **Service Port** field, or select a service name from the list.
   c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
   d) Click **Add**.

8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server with source address affinity persistence

A virtual server represents a destination IP address for application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.

5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. Locate the relevant profile type for the traffic being managed, and either retain the default value or select a custom profile name.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. For the **Default Persistence Profile** setting, select **source_addr**.

   This implements simple persistence, using the default source address affinity profile.

A client system now has a destination IP address on the BIG-IP system.

Removing the self IP addresses from the default VLANs

Remove the self IP addresses from the individual VLANs. After you create the VLAN group, you will create another self IP address for the VLAN group for routing purposes. The individual VLANs no longer need their own self IP addresses.

1. On the Main tab, click **Network** > **Self IPs**.
2. Select the check box for each IP address and VLAN that you want to delete.
3. Click **Delete**.
4. Click **Delete**.

The self IP address is removed from the Self IP list.

Creating a VLAN group

Create a VLAN group that includes the internal and external VLANs. Packets received by a VLAN in the VLAN group are copied onto the other VLAN. This allows traffic to pass through the BIG-IP® system on the same IP network.

1. On the Main tab, click **Network** > **VLANs** > **VLAN Groups**.
   The VLAN Groups list screen opens.
2. Click **Create**.
   The New VLAN Group screen opens.
3. In the **Name** field, type the name `myvlangroup`.
4. For the **VLANs** setting, use the Move button to move the `internal` and `external` VLAN names from the **Available** field to the **Members** field.
5. Click **Finished**.

Creating a self IP for a VLAN group

Before you create a self IP address, ensure that you have created at least one VLAN or VLAN group.

After you have created the VLAN group, create a self IP address for the VLAN group. The self IP address for the VLAN group provides a route for packets destined for the network. With the BIG-IP® system, the path to an IP network is a VLAN. However, with the VLAN group feature used in this procedure, the path to the IP network `10.0.0.0` is actually through more than one VLAN. As IP routers are designed to have only one physical route to a network, a routing conflict can occur. With a self IP address on the BIG-IP system, you can resolve the routing conflict by associating a self IP address with the VLAN group.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **IP Address** field, type an IPv4 or IPv6 address.

   This IP address should represent the address space of the VLAN group that you specify with the **VLAN/Tunnel** setting.
4. In the **Netmask** field, type the full network mask for the specified IP address.

For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.

5. From the **VLAN/Tunnel** list, select the VLAN group with which to associate this self IP address.
6. From the **Port Lockdown** list, select **Allow Default**.
7. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

The BIG-IP system can send and receive traffic through the specified VLAN or VLAN group.

Using Link Aggregation with Tagged VLANs for a Two-network Topology

Overview: Configuring link aggregation of two interfaces using tagged VLANs on two networks

You can use the BIG-IP® system in an aggregated two-interface load balancing topology. *Link aggregation* is the process of combining multiple links so that the links function as a single link with higher bandwidth. Aggregating multiple interfaces into a trunk to create a link has the following advantages:

- Link aggregation increases the bandwidth of the individual network interface cards (NICs) in an additive manner.
- If one link goes down, the other link can handle the traffic by itself.

Link aggregation occurs when you create a trunk. A *trunk* is a combination of two or more interfaces and cables configured as one link.

The examples in this implementation show a trunk that includes two tagged interfaces aggregated together. A *tagged interface* is an interface that is configured to process traffic for multiple VLANs. A VLAN tag identifies the specific VLAN and allows traffic to be passed through that specific VLAN. To cause traffic for multiple VLANs to be passed through a single trunk, you must assign the same trunk to each VLAN.

In the examples, we create a trunk (**trunk1**) that includes two interfaces, **1.1** and **1.2**, and then assign **trunk1** as a tagged interface to both VLAN **external** and VLAN **internal**. One network is connected to VLAN **external**, and a separate network is connected to VLAN **internal**. Consequently, inbound and outbound traffic passing between the BIG-IP system and the vendor switch can use either interface. For example, traffic destined for VLAN **external**l can pass through either interface, **1.1** or **1.2**.

Illustration of link aggregation for a two-network topology

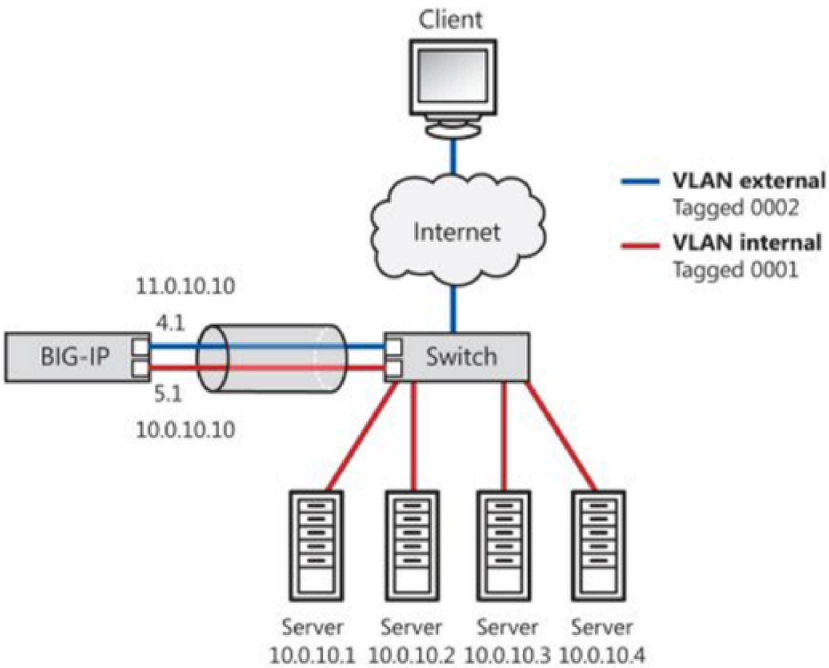


Figure 35: Link aggregation for a two-network topology

Task summary

Perform the following tasks to configure two interfaces (tagged VLANs) to function as a single link with higher bandwidth. In this implementation, each tagged VLAN is on a separate network.

Task list

Creating a trunk
Adding a tagged interface to a VLAN
Creating a load balancing pool
Creating a virtual server with source address affinity persistence

Creating a trunk

You create a trunk on the BIG-IP® system so that the system can then aggregate the links to enhance bandwidth and ensure link availability.

1. On the Main tab, click **Network** > **Trunks**.
   The Trunk List screen opens.
2. Click **Create**.
3. Name the trunk.

4. For the **Interfaces** setting, in the **Available** field, select an interface, and using the Move button, move the interface to the **Members** field. Repeat this action for each interface that you want to include in the trunk.

   Trunk members must be untagged interfaces and cannot belong to another trunk. Therefore, only untagged interfaces that do not belong to another trunk appear in the **Available** list.
5. Select the **LACP** check box.
6. From the **Link Selection Policy** list, retain the default value, **Auto**.
7. From the **Frame Distribution Hash** list, select the default value, **Source/Destination IP address port**.

   Important: *On certain F5 platforms, packets can incorrectly egress on the same BIG-IP trunk member that the external switch ingressed the packets on. You can prevent this by configuring the external switch to use the same algorithm for its frame distribution hash value as you configure on the BIG-IP trunk. For example, if you configure the BIG-IP trunk to base the frame distribution hash value on both source and destination IP addresses, then you must configure the external switch to do the same.*

   ---
8. Click **Finished**.

After you create a trunk, the BIG-IP system aggregates the links to enhance bandwidth and prevent interruption in service.

Adding a tagged interface to a VLAN

After you aggregate the links, you assign the trunk to the VLAN as a tagged interface.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Perform the following for the **external** and **internal** VLANs:
   a) Click the VLAN name.
   b) For the **Interfaces** setting, in the **Available** field, select a trunk, and using the **Move** button, move the trunk that you created to the **Tagged** field.
   c) Click **Update**.

The trunk is assigned to the **external** and **internal** VLAN as a tagged interface.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

---

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type a port number in the **Service Port** field, or select a service name from the list.
   c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
   d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server with source address affinity persistence

A virtual server represents a destination IP address for application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. Locate the relevant profile type for the traffic being managed, and either retain the default value or select a custom profile name.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. For the **Default Persistence Profile** setting, select **source_addr**.

   This implements simple persistence, using the default source address affinity profile.

A client system now has a destination IP address on the BIG-IP system.

Configuring Packet Filtering

Overview: Setting up packet filtering

Packet filters enhance network security by specifying whether a BIG-IP® system interface should accept or reject certain packets based on criteria that you specify. Packet filters enforce an access policy on incoming traffic. They apply to incoming traffic only.

You implement packet filtering by creating packet filter rules. The primary purpose of a packet filter rule is to define the criteria that you want the BIG-IP system to use when filtering packets. Examples of criteria that you can specify in a packet filter rule are:

- The source IP address of a packet
- The destination IP address of a packet
- The destination port of a packet

You specify the criteria for applying packet filter rules within an expression. When creating a packet filter rule, you can instruct the Configuration utility to build an expression for you, in which case you need only choose the criteria from predefined lists, or you can write your own expression text, using the syntax of the `tcpdump` utility.

Note: *Packet filter rules are unrelated to iRules®.*

You can also configure global packet filtering that applies to all packet filter rules that you create.

Task summary

By setting up some basic IP routing and configuring packet filtering, specific hosts on the internal VLAN can connect to the internal VLAN's self IP address. These hosts can also use common Internet services such as HTTP, HTTPS, DNS, FTP, and SSH. Traffic from all other hosts in the internal VLAN is rejected.

Task list

Enabling SNAT automap for internal and external VLANs

You can configure SNAT automapping on the BIG-IP system for internal and external VLANs.

1. On the Main tab, click **Local Traffic** > **Address Translation**.
   The **SNAT List** screen displays a list of existing SNATs.

2. Click **Create**.
3. Name the new SNAT.
4. From the **Translation** list, select **Automap**.
5. For the **VLAN / Tunnel List** setting, in the **Available** field, select **external** and **external**, and using the **Move** button, move the VLANs to the **Selected** field.
6. Click **Finished**.

SNAT automapping on the BIG-IP system is configured for internal and external VLANs.

Creating a default gateway pool

Create a default gateway pool for the system to use to forward traffic.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **gateway_icmp** monitor, and click **<<** to move the monitor to the **Active** list.
5. Using the **New Members** setting, add each router that you want to include in the default gateway pool:
   a) Type the IP address of a router in the **Address** field.
   b) Type an asterisk ***** in the **Service Port** field, or select ***All Services** from the list.
   c) Click **Add**.
6. Click **Finished**.

Creating a forwarding virtual server

A virtual server represents a destination IP address for application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the Destination setting:
   a) For **Type**, select **Network**.
   b) In the **Address** field, type the IP address **0.0.0.0**.
   c) In the **Mask** field, type the netmask **0.0.0.0**.
5. From the **Service Port** list, select ***All Ports**.
6. In the Configuration area of the screen, from the **Type** list, select **Forwarding (IP)**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN/Tunnel Traffic** list, select **Enabled On**.
9. For the **VLAN List** setting, from the **Available** box, select **internal**, and click the Move button to move the VLAN name to the **Selected** box.

10.
11. In the Resources area of the screen, locate the **Default Pool** setting and select the pool you created previously.
12. Click **Finished**.

You now have a destination IP address on the BIG-IP system for application traffic.

Enabling packet filtering on the BIG-IP system

Before creating a packet filtering rule, you must enable packet filtering.

1. On the Main tab, click **Network** > **Packet Filters**.
   The Packet Filters screen opens.
2. From the **Packet Filtering** list, select **Enabled**.
3. From the **Unhandled Packet Action** list, select **Accept**.
4. Click **Update**.

Packet filtering is enabled.

Creating a packet filter rule

When implementing packet filtering, you need to create a packet filter rule.

1. On the Main tab, click **Network** > **Packet Filters**.
   The Packet Filters screen opens.
2. Click **Rules**.
3. Click **Create**.
4. Name the rule.
5. From the **Order** list, select **First**.
6. From the **Action** list, select **Reject**.
7. From the **Apply to VLAN** list, select **internal**.
8. From the **Logging** list, select **Enabled**.
9. From the **Filter Expression Method** list, select **Enter Expression Text**.
10. In the **Filter Expression** field, type an expression.
    For example: `not dst port 80 and not dst port 443 and not dst port 53 and no dst port 22 and not dst port 20 and not dst port 21 and not dst host <internal self IP address>`

    Note: *Replace <internal self IP address> with the actual self IP address of VLAN internal.*

    ---

11. Click **Finished**.

The packet filter rule is available.

Referencing an External File from within an iRule

Overview: Referencing an external file from an iRule

Using the BIG-IP® Configuration utility or **tmsh**, you can import a file or URL from another system to the BIG-IP system, with content that you want an iRule to return to a client, based on some iRule event. Possible uses for this feature are:

- To send a web page other than the page that the client requested. For example, you might want the system to send a maintenance page instead of the requested page.
- To send an image.
- To use a file as a template and modify the file in the iRule before sending the file.
- To download policy information from an external server and merge that data with a locally-stored policy.

The file that an iRule accesses is known as an *iFile*, and can be any type of file, such as a binary file or a text file. These files are read-only files.

This example shows an iRule that references an iFile named `ifileURL`, in partition `Common`:

```
ltm rule ifile_rule {
   when HTTP_RESPONSE {
   # return a list of iFiles in all partitions
   set listifiles [ifile listall]
   log local0. "list of ifiles: $listifiles"

   # return the attributes of an iFile specified
   array set array_attributes [ifile attributes "/Common/ifileURL"]
   foreach {array attr} [array get array_attributes ] {
   log local0. "$array : $attr"
   }

   # serve an iFile when http status is 404.
   set file [ifile get "/Common/ifileURL"]
   log local0. "file: $file"
   if { [HTTP::status] equals "404" } {
     HTTP::respond 200 ifile "/Common/ifileURL"

  }
}
}
```

iRule commands for iFiles

This list shows the commands available for referencing an iFile within an iRule. All of these commands return a string, except for the command `[ifile attributes IFILENAME]`, which returns an array.

Available iRule commands for referencing an iFile

```
[ifile get IFILENAME]
```

```
[ifile listall]
[ifile attributes IFILENAME]
[ifile size IFILENAME]
[ifile last_updated_by IFILENAME]
[ifile last_update_time IFILENAME]
[ifile revision IFILENAME]
[ifile checksum IFILENAME]
[ifile attributes IFILENAME]
```

Task summary

You can import an existing file to the BIG-IP® system, create an iFile that is based on the imported file, and then write an iRule that returns the content of that file to a client system, based on an iRule event.

Task list

Importing a file to the BIG-IP system
Creating an iFile
Writing an iRule that references an iFile

Importing a file to the BIG-IP system

As a prerequisite, the file you want to import must reside on the BIG-IP® system you specify.

You can import a file from another system onto the BIG-IP system, as the first step in writing an iRule that references that file.

1. On the Main tab, click **System** > **File Management** > **iFile List** > **Import**.
2. For the **File Name** setting, click **Browse**.
   The system opens a browse window so you can locate the file that you want to import to the BIG-IP system.
3. Browse for the file and click **Open**.
   The name of the file you select appears in the **File Name** setting.
4. In the **Name** field, type a new name for the file, such as `1k.html`.
   The new file name appears in the list of imported files.
5. Click **Import**.

The result of this task is that the file you selected now resides on the BIG-IP system.

Creating an iFile

As a prerequisite, ensure that the current administrative partition is set to the partition in which you want the iFile to reside.

You perform this task to create an iFile that you can then reference in an iRule.

1. On the Main tab, click **Local Traffic** > **iRules** > **iFile List**.
2. Click **Create**.
3. In the **Name** field, type a new name for the iFile, such as `ifileURL`.

4. From the **File Name** list, select the name of the imported file object, such as `1k.html`.
5. Click **Finished**.
   The new iFile appears in the list of iFiles.

The result of this task is that you now have a file that an iRule can reference.

Writing an iRule that references an iFile

You perform this task to create an iRule that references an iFile.

***Note:** If the iFile resides in partition `/Common`, then specifying the partition when referencing the iFile is optional. If the iFile resides in a partition other than `/Common`, such as `/Partition_A`, you must include the partition name in the iFile path name within the iRule.*

---

1. On the Main tab, click **Local Traffic** > **iRules**.
   The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
   The New iRule screen opens.
3. In the **Name** field, type a name between 1 and 31 characters, such as `my_iRule`.
4. In the **Definition** field, type the syntax for the iRule using Tool Command Language (Tcl) syntax.

   For complete and detailed information iRules syntax, see the F5 Networks DevCentral web site (`http://devcentral.f5.com`).
5. Click **Finished**.
   The new iRule appears in the list of iRules on the system.

Implementation result

You now have an iRule that accesses a file on the BIG-IP®system, based on a particular iRule event.

Configuring Remote User Authentication and Authorization

Overview: Remote authentication and authorization of BIG-IP user accounts

The BIG-IP® system includes a comprehensive solution for managing BIG-IP administrative accounts on your network. With this solution, you can:

Use a remote server to store BIG-IP system user accounts.
The BIG-IP system includes support for using a remote authentication server to store BIG-IP system user accounts. After creating BIG-IP system accounts on the remote server (using the server vendor's instructions), you can configure the BIG-IP system to use remote user authentication and authorization (access control) for that server type.

Assign group-based access.
The BIG-IP system includes an optional feature known as *remote role groups*. With the *remote role groups* feature, you can use existing group definitions on the remote server to define the access control properties for users in a group. This feature not only provides more granularity in assigning user privileges, but also removes any need to duplicate remote user accounts on the BIG-IP system for the purpose of assigning those privileges.

Propagate a set of authorization data to multiple BIG-IP systems.
The BIG-IP system includes a tool for propagating BIG-IP system configuration data to multiple BIG-IP devices on the network. This tool is known as the Single Configuration File (SCF) feature.

Task summary

You can configure the BIG-IP® system to authorize user accounts that are stored on a remote authentication server.

Important: *If you configure access control settings for group-based accounts (using the remote role groups feature), the BIG-IP system always applies those settings, rather than the default access control settings, to group-based accounts.*

The BIG-IP® system supports several types of authentication servers for storing BIG-IP system administrative user accounts. The actual procedure you use to specify the type of remote server differs, depending on the server type.

Task list

Specifying LDAP or Active Directory server information

Before you begin:

- Verify that the BIG-IP® system user accounts have been created on the remote authentication server.
- Verify that the appropriate user groups, if any, are defined on the remote authentication server.
- If you want to verify the certificate of the authentication server, import one or more SSL certificates.

You can configure the BIG-IP system to use an LDAP or Microsoft® Windows® Active Directory ®server for authenticating BIG-IP system user accounts, that is, traffic that passes through the management interface (MGMT).

Important: *The values you specify in this procedure for the **Role**, **Partition Access**, and **Terminal Access** settings do not apply to group-based authorization. These values represent the default values that the BIG-IP system applies to any user account that is not part of a remote role group. Also, for the `Other External Users` user account, you can modify the **Role**, **Partition Access**, and **Terminal Access** settings only when your current partition on the BIG-IP system is set to `Common`. If you attempt to modify these settings when your current partition is other than `Common`, the system displays an error message.*

---

1. On the Main tab, click **System** > **Users** > **Authentication**.
2. On the menu bar, click **Authentication**.
3. Click **Change**.
4. From the **User Directory** list, select **Remote - LDAP** or **Remote - Active Directory**.
5. In the **Host** field, type the IP address of the remote server.

   The route domain to which this address pertains must be route domain `0`.
6. For the **Port** setting, retain the default port number (`389`) or type a new port number.

   This number represents the port number that the BIG-IP system uses to access the remote server.
7. In the **Remote Directory Tree** field, type the file location (tree) of the user authentication database on the LDAP or Active Directory server.

   At minimum, you must specify a domain component (that is, `dc=[value]`).
8. For the **Scope** setting, retain the default value (`Sub`) or select a new value.

   This setting specifies the level of the remote server database that the BIG-IP system should search for user authentication.
9. For the **Bind** setting, specify a user ID login for the remote server:
   a) In the **DN** field, type the distinguished name for the remote user ID.
   b) In the **Password** field, type the password for the remote user ID.
   c) In the **Confirm** field, re-type the password that you typed in the **Password** field.
10. To enable SSL-based authentication, from the **SSL** list select **Enabled** and, if necessary, configure these settings:
    a) From the **SSL CA Certificate** list, select the name of a chain certificate, that is, the third-party CA or self-signed certificate that normally resides on the remote authentication server.
    b) From the **SSL Client Key** list, select the name of the client SSL key.

       Use this setting only when the remote server requires that the client present a certificate.
    c) From the **SSL Client Certificate** list, select the name of the client SSL certificate.

       Use this setting only if the remote server requires that the client present a certificate.

11. From the **Role** list, select the user role that you want the BIG-IP system to assign by default to all BIG-IP system user accounts authenticated on the remote server.
12. From the **Partition Access** list, select the default administrative partition that all remotely-authenticated BIG-IP system user accounts can access.
13. From the **Terminal Access** list, select either of these as the default terminal access option for remotely-authenticated user accounts:

Option	Description
Disabled	Choose this option when you do not want the remotely-stored user accounts to have terminal access to the BIG-IP system.
tmsh	Choose this option when you want the remotely-stored user accounts to have only `tmsh` access to the BIG-IP system.

14. Click **Finished**.

You can now authenticate administrative traffic for user accounts that are stored on a remote LDAP or Active Directory server. If you have no need to configure group-based user authorization, your configuration tasks are complete.

Specifying client certificate LDAP server information

Verify that the required user accounts for the BIG-IP® system exist on the remote authentication server.

For authenticating BIG-IP system user accounts (that is, traffic that passes through the management interface [MGMT]), you can configure the BIG-IP system to authenticate certificates issued by a certificate authority's Online Certificate Status Protocol (OCSP) responder.

Important: *The values you specify in this procedure for the **Role**, **Partition Access**, and **Terminal Access** settings do not apply to group-based authorization. These values represent the default values or locally configured user accounts (which override the default role) that the BIG-IP system applies to any user account that is not part of a remote role group.*

---

1. On the Main tab, click **System** > **File Management** > **Apache Certificate List** > **Import**, browse for the certificate file to import, type a name, and click **Import**.
   The certificate will be added to the Apache Certificate list.
2. On the Main tab, click **System** > **Users** > **Authentication**.
3. On the menu bar, click **Authentication**.
4. Click **Change**.
5. From the **User Directory** list, select **Remote - ClientCert LDAP**.
6. In the **Host** field, type the IP address of the remote server.

   The route domain to which this address pertains must be route domain `0`.
7. For the **Port** setting, retain the default port number (`389`) or type a new port number.

   This number represents the port number that the BIG-IP system uses to access the remote server.
8. In the **Remote Directory Tree** field, type the file location (tree) of the user authentication database on the client certificate server.

   At minimum, you must specify a domain component (that is, `dc=[value]`).
9. For the **Scope** setting, retain the default value (`Sub`) or select a new value.

   This setting specifies the level of the remote server database that the BIG-IP system should search for user authentication.

10. For the **Bind** setting, specify a user ID login for the remote server:
    a) In the **DN** field, type the distinguished name for the remote user ID.
    b) In the **Password** field, type the password for the remote user ID.
    c) In the **Confirm** field, re-type the password that you typed in the **Password** field.

11. To enable SSL-based authentication, from the **SSL** list select **Enabled** and, if necessary, configure these settings:
    a) From the **SSL CA Certificate** list, select the name of a chain certificate; that is, the third-party CA or self-signed certificate that normally resides on the remote authentication server.
    b) From the **SSL Client Key** list, select the name of the client SSL key.

       Use this setting only when the remote server requires that the client present a certificate.
    c) From the **SSL Client Certificate** list, select the name of the client SSL certificate.

       Use this setting only if the remote server requires that the client present a certificate.

12. In the **CA Certificate** field, type the absolute folder path of `apache-ssl-cert fileobject` for the CA signing authority.

    The absolute folder path is `/Common/<folder path>/<certificate name>`. To determine the absolute folder path of the `apache-ssl-cert fileobject`, click **System** > **File Management** > **Apache Certificate List** and note the target certificate's partition and path.

    Important: *Apache certificates can only be stored within* `/Common`*.*

    ---

13. In the **Login Name** field, type an LDAP search prefix that will contain the distinguished name (DN) from the user certificate, such as `CN`.

    This specifies the LDAP attribute to be used as a login name. The default is disabled.
14. In the **Login LDAP Attribute** field, type the account name for the LDAP server.

    The value for this option is normally the user ID. However, if the server is a Microsoft® Windows® Active Directory®server, the value must be the account name `sAMAccountName` (case-sensitive). The default value is none.
15. In the **Login Filter** field, type the LDAP attribute that contains the short name of the user.

    This specifies the filter to be applied on the common name (CN) of the client certificate and usually this is the user ID or `sAMAccountName`. The filter is a regular expression used to extract required information from the CN of the client certificate that is matched against the LDAP search results. The default is disabled.
16. For the **Depth** setting, retain the default value (`10`) or type a new value for verification depth.
17. From the **Role** list, select the user role that you want the BIG-IP system to assign by default to all BIG-IP system user accounts authenticated on the remote server.
18. From the **Partition Access** list, select the default administrative partition that all remotely-authenticated BIG-IP system user accounts can access.
19. From the **Terminal Access** list, select either of these as the default terminal access option for remotely-authenticated user accounts:

Option	Description
Disabled	Choose this option when you do not want the remotely-stored user accounts to have terminal access to the BIG-IP system.
tmsh	Choose this option when you want the remotely-stored user accounts to have only `tmsh` access to the BIG-IP system.

20. Click **Finished**.

You can now authenticate administrative traffic for user accounts that are stored on a remote client certificate server. If you have no need to configure group-based user authorization, your configuration tasks are complete.

Specifying RADIUS server information

Before you begin:

- Verify that the BIG-IP® system user accounts have been created on the remote authentication server.
- Verify that the appropriate user groups, if any, are defined on the remote authentication server.

You can configure the BIG-IP system to use a RADIUS server for authenticating BIG-IP system user accounts, that is, traffic that passes through the management interface (MGMT).

Important: *The values you specify in this procedure for the **Role**, **Partition Access**, and **Terminal Access** settings do not apply to group-based authorization. These values represent the default values that the BIG-IP system applies to any user account that is not part of a role group that is defined on the remote authentication server. Also, for the* `Other External Users` *user account, you can modify the **Role**, **Partition Access**, and **Terminal Access** settings only when your current partition on the BIG-IP system is set to* `Common`*. If you attempt to modify these settings when your current partition is other than* `Common`*, the system displays an error message.*

---

1. On the Main tab, click **System** > **Users** > **Authentication**.
2. On the menu bar, click **Authentication**.
3. Click **Change**.
4. From the **User Directory** list, select **Remote - RADIUS**.
5. For the **Primary** setting:
   a) In the **Host** field, type the name of the primary RADIUS server.
      The route domain with which this host is associated must be route domain `0`.
   b) In the **Secret** field, type the password for access to the primary RADIUS server.
   c) In the **Confirm** field, re-type the RADIUS secret.
6. If you set the **Server Configuration** setting to **Primary and Secondary**, then for the **Secondary** setting:
   a) In the **Host** field, type the name of the secondary RADIUS server.
      The route domain with which this host is associated must be route domain `0`.
   b) In the **Secret** field, type the password for access to the secondary RADIUS server.
   c) In the **Confirm** field, re-type the RADIUS secret.
7. From the **Role** list, select the user role that you want the BIG-IP system to assign by default to all BIG-IP system user accounts authenticated on the remote server.
8. From the **Partition Access** list, select the default administrative partition that all remotely-authenticated BIG-IP system user accounts can access.
9. From the **Terminal Access** list, select either of these as the default terminal access option for remotely-authenticated user accounts:

Option	Description
Disabled	Choose this option when you do not want the remotely-stored user accounts to have terminal access to the BIG-IP system.
tmsh	Choose this option when you want the remotely-stored user accounts to have only `tmsh` access to the BIG-IP system.

10. Click **Finished**.

You can now authenticate administrative traffic for BIG-IP system user accounts that are stored on a remote RADIUS server. If you have no need to configure group-based user authorization, your configuration tasks are complete.

Specifying TACACS+ server information

Before you begin:

- Verify that the BIG-IP® system user accounts have been created on the remote authentication server.
- Verify that the appropriate user groups, if any, are defined on the remote authentication server.

You can configure the BIG-IP system to use a TACACS+ server for authenticating BIG-IP system user accounts, that is, traffic that passes through the management interface (MGMT).

Important: *The values you specify in this procedure for the **Role**, **Partition Access**, and **Terminal Access** settings do not apply to group-based authorization. These values represent the default values that the BIG-IP system applies to any user account that is not part of a remote role group. Also, for the* `Other External Users` *user account, you can modify the **Role**, **Partition Access**, and **Terminal Access** settings only when your current partition on the BIG-IP system is set to* `Common`*. If you attempt to modify these settings when your current partition is other than* `Common`*, the system displays an error message.*

---

1. On the Main tab, click **System** > **Users** > **Authentication**.
2. On the menu bar, click **Authentication**.
3. Click **Change**.
4. From the **User Directory** list, select **Remote - TACACS+**.
5. For the **Servers** setting, type an IP address for the remote TACACS+ server.

   The route domain to which this address pertains must be route domain `0`.
6. Click **Add**.
   The IP address for the remote TACACS+ server appears in the **Servers** list.
7. In the **Secret** field, type the password for access to the TACACS+ server.

   Warning: *Do not include the symbol* `#` *in the secret. Doing so causes authentication of local user accounts (such as* `root` *and* `admin`*) to fail.*

   ---
8. In the **Confirm Secret** field, re-type the TACACS+ secret.
9. From the **Encryption** list, select an encryption option:

Option	Description
Enabled	Specifies that the system encrypts the TACACS+ packets.
Disabled	Specifies that the system sends unencrypted TACACS+ packets.

10. In the **Service Name** field, type the name of the service that the user is requesting to be authenticated to use (usually `ppp`).

    Specifying the service causes the TACACS+ server to behave differently for different types of authentication requests. Examples of service names that you can specify are: `ppp`, `slip`, `arap`, `shell`, `tty-daemon`, `connection`, `system`, and `firewall`.
11. In the **Protocol Name** field, type the name of the protocol associated with the value specified in the **Service Name** field.

This value is usually `ip`. Examples of protocol names that you can specify are: `ip`, `lcp`, `ipx`, `atalk`, `vines`, `lat`, `xremote`, `tn3270`, `telnet`, `rlogin`, `pad`, `vpdn`, `ftp`, `http`, `deccp`, `osicp`, and `unknown`.

12. From the **Role** list, select the user role that you want the BIG-IP system to assign by default to all BIG-IP system user accounts authenticated on the remote server.
13. From the **Partition Access** list, select the default administrative partition that all remotely-authenticated BIG-IP system user accounts can access.
14. From the **Terminal Access** list, select either of these as the default terminal access option for remotely-authenticated user accounts:

Option	Description
Disabled	Choose this option when you do not want the remotely-stored user accounts to have terminal access to the BIG-IP system.
tmsh	Choose this option when you want the remotely-stored user accounts to have only `tmsh` access to the BIG-IP system.

15. Click **Finished**.

You can now authenticate administrative traffic for BIG-IP system user accounts that are stored on a remote TACACS+ server. If you have no need to configure group-based user authorization, your configuration tasks are complete.

Configuring access control for remote role-based user groups

On the BIG-IP® system, you can configure access control properties (permissions) for existing user groups that are defined on a remote authentication server. For example, if the configuration of a remote LDAP authentication server includes the attribute string `memberOF=cn=BigIPOperatorsGroup,cn=users,dc=dev,dc=net`, you can assign a specific set of access control properties to all user accounts in the group `BigIPOperatorsGroup`.

1. On the Main tab, click **System** > **Users**.
2. On the menu bar, click **Remote Role Groups**.
3. Click **Create**.
4. In the **Group Name** field, type the group name that is defined on the remote authentication server. An example of a group name is **BigIPOperatorsGroup**.
5. In the **Line Order** field, type a number.

   An example of a line order is **1**.
6. In the **Attribute String** field, type an attribute.

   An example of an attribute string is `memberOF=cn=BigIPOperatorsGroup,cn=users,dc=dev,dc=net`.

   The BIG-IP system attempts to match this attribute with an attribute on the remote authentication server. On finding a match, the BIG-IP system applies the access control settings defined here to the users in that group. If a match is not found, the system applies the default access control settings to all remotely-stored user accounts (excluding any user account for which access control settings are individually configured).
7. From the **Remote Access** list, select a value.

Option	Description
Enabled	Choose this value if you want to enable remote console access for the defined user group.

Option	Description
Disabled	Choose this value if you want to disable remote console access for the defined user group.

8. From the **Assigned Role** list, select a user role for the remote user group.
9. From the **Partition Access** list, select an administrative partition value.

Option	Description
All	Choose this value to give users in the defined group access to their authorized objects in all partitions on the BIG-IP system.
`partition_name`	Choose a specific partition name to give users in the defined group access to that partition only.
Common	Choose this value to give users in the defined group access to partition **Common** only.

10. From the **Terminal Access** list, select the type of command-line access you want to grant users in the group, if any.
11. Click **Finished**.

The user group that you specified now has the assigned role, partition access, and terminal access properties assigned to it.

Saving access control settings to a file

You can save the running configuration of the system, including all settings for remote user authentication and authorization, in a flat, text file with a specified name and the extension `.scf`.

1. On the BIG-IP® system, access a command-line prompt.
2. At the prompt, open the Traffic Management Shell by typing the command `tmsh`.
3. Type `sys save` *`filename`*.
   `sys save myConfiguration053107` creates the file `myConfiguration053107.scf` in the `var/local/scf` directory.
   `sys save /config/myConfiguration` creates the file `myConfiguration.scf` in the `/config` directory.

You can now import this file onto other BIG-IP devices on the network.

Importing BIG-IP configuration data onto other BIG-IP systems

You can use the `tmsh sys load` command to import a single configuration file (SCF), including access control data, onto other BIG-IP® devices on the network.

***Note:** This task is optional.*

---

1. On the BIG-IP system on which you created the SCF, access a command-line prompt.
2. Copy the SCF that you previously created to a location on your network that you can access from the system that you want to configure.

3. Edit the SCF to reflect the management routing and special passwords of the BIG-IP system that you want to configure:
   a) Open the SCF in an editor.
   b) Where necessary, change the values of the management IP address, network mask, management default route, self IP addresses, virtual server IP addresses, routes, default routes, and host name fields to the values for the new system.
   c) If necessary, change the passwords for the `root` and `admin` accounts using the command `user` *`name`* `password none newpassword` *`password`*.

   Important: *When configuring a unit that is part of a redundant system configuration and that is using the SCF from the peer unit, do not modify the* `root` *and* `admin` *accounts. These accounts must be identical on both units of the redundant system.*

   ---

   d) Save the edited SCF.

4. On the BIG-IP system that you want to configure, open the Traffic Management Shell by typing the command `tmsh`.
5. Type `sys load` *`scf_filename`*.
   `sys load myConfiguration053107.scf` saves a backup of the running configuration in the `/var/local/scf` directory, and then resets the running configuration with the configuration contained in the SCF you are loading.

Configuring Administrative Partitions to Control User Access

Overview: Administrative partitions for user access control

The BIG-IP® system includes a powerful authorization feature known as administrative partitions. Using the *administrative partitions* feature, you ensure that BIG-IP system grants administrative users exactly the right type and amount of access to BIG-IP system resources. As a result, you can tailor user access to resources to exactly fit the needs of your organization.

Task summary

There are two main tasks for controlling user access to BIG-IP® system objects.

Task list

Creating an administrative partition
Configuring user access to a partition

Creating an administrative partition

An administrative partition creates an access control boundary for users and applications.

1. On the Main tab, expand **System** and click **Users**.
   The Users List screen opens.
2. On the menu bar, click **Partition List**.
3. Click **Create**.
   The New Partition screen opens.
4. In the **Name** field, type a unique name for the partition.
   An example of a partition name is `app1_partition`.
5. Type a description of the partition in the **Description** field.
   This field is optional.
6. For the **Device Group** setting, choose an action:

Action	Result
Retain the default value.	Choose this option if you want the folder corresponding to this partition to inherit the value of the device group attribute from folder `root`.
Clear the check box and select the name of a device group.	Choose this option if you do not want the folder corresponding to this partition to inherit the value of the device group attribute from folder `root`.

7. For the **Traffic Group** setting, choose an action:

Action	Result
Retain the default value.	Choose this option if you want the folder corresponding to this partition to inherit the value of the traffic group attribute from folder `root`.
Clear the check box and select the name of a traffic group.	Choose this option if you do not want the folder corresponding to this partition to inherit the value of the traffic group attribute from folder `root`.

8. Click **Finished**.

The new partition appears in the partition list.

Configuring user access to a partition

You can configure user access to a partition either when you first create the user account or when you modify the user account properties. This procedure shows how to configure partition access to an existing user account.

1. On the Main tab, click **System** > **Users**.
2. In the User Name column, click the user account name.
3. To grant an access level other than **No Access**, use the **Role** list to select a user role.
4. From the **Partition Access** list, select a partition name.

   You can select a single partition name, or **All**.

5. Click **Update**.

Working with Single Configuration Files

Overview: Working with single configuration files

A *single configuration file (SCF)* is a flat, text file that contains a series of `tmsh` commands, and the attributes and values of those commands, that reflect the configuration of the BIG-IP® system. Specifically, the SCF contains the local traffic management and TMOS® configuration of the BIG-IP system. This figure shows a small part of a sample SCF.

```
  vlan external {
    tag 4093
    interfaces 1.3
}
vlan internal {
    tag 4094
    interfaces 1.10
}
pool dev_https3 {
    members {
        10.60.10.105:https{}
        10.60.10.106:https{}
    }
}
```

The single configuration file feature allows you to save the configuration of a BIG-IP system in a text file. You can then use the text file to easily replicate the configuration across multiple BIG-IP systems. This not only saves you time, but also allows you to create a consistent, secure, comprehensive local traffic management environment on your network.

tmsh commands for single configuration files

You use the `tmsh` utility to perform the basic management of a single configuration file (SCF). This table contains an overview of the commands to accomplish this.

tmsh command	Usage
`save sys config file [filename]`	Use this command to save a copy of the currently running configuration to an SCF. It is important to note that saving a configuration to an SCF does not affect the running or stored configuration of the BIG-IP® system on which you run the command.
`load sys config file [filename]`	Use this command to replace or restore an SCF with a saved configuration. When you use this command, the system saves any previously running configuration to the directory `/var/local/scf/`, by default.

tmsh command	Usage
`load sys config default`	Use this command to restore the factory default settings of the configuration file, while retaining the management IP address and the administrator user name and password.

Task summary

You can perform three main tasks with respect to single configuration files.

Task list

Creating and saving an SCF

Use this procedure to create and save a single configuration file.

1. Access the `tmsh` utility.
2. Use the following syntax: `save sys config file [filename]`

   If you include the `.scf` extension in the file name, the system does not add an additional file extension.

   If you create an SCF file twice (on two different occasions), you can compare the contents of the two files.

This procedure causes the `tmsh` utility to gather all of the commands (and their attributes and values) that compose the running configuration. Once gathered, the system saves the configuration to a flat file with the name you specify and the extension of `.scf`. By default, the system stores this file in the `/var/local/scf` directory, but you can specify a different path if you prefer.

Loading an SCF onto a target BIG-IP system

The primary benefit of the SCF feature is that it gives you the ability to create a configuration on one BIG-IP® system that you can load onto other BIG-IP systems (hereafter referred to as the target BIG-IP system), rather than having to recreate the configuration multiple times.

After you have created and saved the SCF using the `save sys config file [filename]` command, you can modify any data unique to the specific target BIG-IP system, then load the configuration on that system.

Note: *To successfully load a configuration you have replicated, ensure that no line of the configuration is longer than 4096 characters. If there are more than 4096 characters in a single line, the system reverts to the previous running configuration.*

---

1. On the target BIG-IP system, load the saved SCF file by typing the following command: `tmsh load sys config file [filename]`

The `tmsh` utility first saves the system's stored configuration in a backup file (named `/var/local/scf/backup.scf`), and then uses the configuration stored in the SCF that you are loading.

2. Use a text editor to open the SCF and edit any data that is unique to the target BIG-IP system, such as the management IP address.
3. Save the SCF to the target BIG-IP system by typing the following command: `sys save config file [filename]`

   If a backup SCF already exists, the `tmsh` utility appends a number to the name of the existing backup file, and then creates a new backup file. Thus:

   - The first time the system backs up the running configuration during a load operation, the system names the backup file `/var/local/scf/backup.scf`.
   - The next time the system backs up the running configuration, the system renames the file from `/var/local/scf/backup.scf` to `/var/local/scf/backup-1.scf` and creates a new file named `/var/local/scf/backup.scf`.
   - If you run the `load` command a third time, the system renames the file from `/var/local/scf/backup-1.scf` to `/var/local/scf/backup-2.scf`, renames the file `/var/local/scf/backup.scf` to `/var/local/scf/backup-1.scf`, and once again creates a new file named `/var/local/scf/backup.scf`.

Using an SCF to restore a BIG-IP system configuration

You can use an SCF to restore a BIG-IP® system configuration. The BIG-IP system ships with a default SCF. Depending on whether you want to restore the factory default configuration or load a specific configuration, perform this step.

1. From a console window, access the `tmsh` prompt.
2. Choose one of these options.

Option	Description
Restore a system to the factory default configuration	Type the command `tmsh load sys config default`. This command retains the management IP and the assigned **root** and administrator passwords. When you use this command, the system first saves the running configuration in the `backup.scf` file and then resets the local traffic management and the operating system configuration to the factory default settings by loading the SCF, `/defaults/defaults.scf`.
Restore a system with values defined in the specified SCF	Type the command `tmsh load sys config file [filename]`. When you use this command, the system first saves the running configuration in the `backup.scf` file, and then resets the running configuration to the values contained in the specified SCF. You must run the `save sys config partitions all` command to save the running configuration in the stored configuration files.

Forcing Renewal of the BIG-IP System Management Port DHCP Lease

About the BIG-IP system management port and DHCP

The BIG-IP® system uses the management port for administrative traffic. The system does not forward application traffic, such as traffic slated for load balancing, through this interface.

By default, DHCP is disabled for the BIG-IP system management port on physical devices, and enabled for the BIG-IP system management port on virtual editions. When enabled, DHCP uses UDP ports `67` and `68`. On the first boot, the BIG-IP system contacts the DHCP server and obtains a lease for an IP address and a default route for the management port.

Forcing a DHCP lease renewal for the BIG-IP system management port

Ensure that DHCP is enabled on the BIG-IP® system.

Force the renewal of the DHCP lease for an IP address and a default route for the BIG-IP system management port.

1. Log on to the command-line interface of the BIG-IP system.
2. At the BASH prompt, type: `bigstart restart dhclient`
   The `dhclient` on the BIG-IP system restarts, contacts the DHCP server, and acquires a new lease for the BIG-IP system management port.

The DHCP lease renews automatically at the previously configured interval.

Configuring a One-Arm Deployment Using WCCPv2

Overview: Configuring a one-arm deployment using WCCPv2

In certain cases, it is not advantageous or even possible to deploy the BIG-IP® system inline. For example, in the case of a collapsed backbone where the WAN router and the LAN switch are in one physical device, you might not be able to deploy the BIG-IP system inline.

If you choose not to deploy the BIG-IP system inline, you can use a one-arm deployment. In a *one-arm deployment*, the BIG-IP system has a single (hence, one-arm) connection to the WAN router or LAN switch. The WAN router (or switch) redirects all relevant traffic to the BIG-IP system. In this configuration, the WAN router typically uses Web Cache Communication Protocol version 2 (WCCPv2) to redirect traffic to the BIG-IP system.

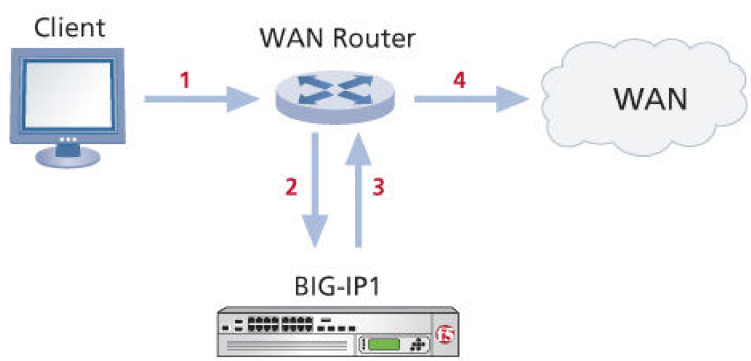


Figure 36: Network topology for a one-arm connection

The traffic flow sequence in this illustration is as follows:

1. The client initiates a session.
2. A WAN router redirects traffic to the BIG-IP system.
3. The BIG-IP1 processes traffic and sends it back to the WAN router.
4. The WAN router forwards traffic across the WAN.

About WCCPv2 redirection on the BIG-IP system

TMOS® includes support for Web Cache Communication Protocol version 2 (WCCPv2). *WCCPv2* is a content-routing protocol developed by Cisco® Systems. It provides a mechanism to redirect traffic flows in real time. The primary purpose of the interaction between WCCPv2-enabled routers and a BIG-IP® system is to establish and maintain the transparent redirection of selected types of traffic flowing through those routers.

To use WCCPv2, you must enable WCCPv2 on one or more routers connected to the BIG-IP® system, and configure a service group on the BIG-IP system that includes the router information. The BIG-IP system then receives all the network traffic from each router in the associated service group, and determines both the traffic to optimize and the traffic to which to apply a service.

In configuring WCCPv2 on a network, you define a *service group* on the BIG-IP system, which is a collection of WCCPv2 services configured on the BIG-IP system. A WCCPv2 *service* in this context is a set of redirection criteria and processing instructions that the BIG-IP system applies to any traffic that a router in the service group redirects to the BIG-IP system. Each service matches a service identifier on the router.

The following illustration shows a one-arm configuration on one side of the WAN and an inline (bridge) configuration on the other side.

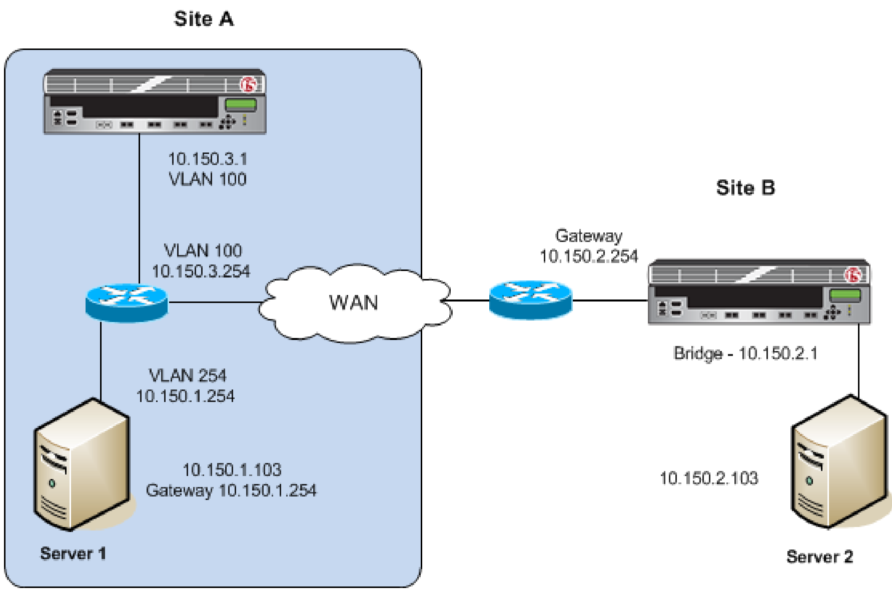


Figure 37: Example of a one-arm configuration

Before you begin configuring an iSession connection

Before you configure an iSession™ connection on the BIG-IP® system, make sure that you have completed the following general prerequisites.

- You must have an existing routed IP network between the two locations where the BIG-IP devices will be installed.
- One BIG-IP system is located on each side of the WAN network you are using.
- The BIG-IP hardware is installed with an initial network configuration applied.
- F5® recommends that both units be running the same BIG-IP software version.
- The Application Acceleration Manager™ license is enabled.
- Application Acceleration Manager (AAM) is provisioned at the level **Nominal**.
- The management IP address is configured on the BIG-IP system.
- You must have administrative access to both the Web management and SSH command line interfaces on the BIG-IP system.
- If there are firewalls, you must have TCP port `443` open in both directions. Optionally, you can allow TCP port `22` for SSH access to the command line interface for configuration verification, but not for actual BIG-IP iSession traffic. After you configure the BIG-IP system, you can perform this verification from the Configuration utility (**Acceleration** > **Symmetric Optimization** > **Diagnostics**).

Task summary

To use WCCPv2 for traffic redirection, you configure a service group on the BIG-IP® system that includes at least one service. You also configure this service on the WCCPv2-enabled router connected to the BIG-IP system.

For optimization, you also need to configure the BIG-IP system on the other side of the WAN to complete the connection. The BIG-IP system on the other side of the WAN can be set up in either a one-arm or inline configuration.

***Note:** The example described in this implementation applies to the Cisco 3750 and Cat 6500 routers.*

Prerequisites

Before you begin configuring WCCPv2 for traffic redirection, ensure that you have performed the following actions on the other devices in your network.

- The interface and associated VLAN have been configured on the router or switch. For instructions, refer to the Cisco documentation for your device.
- IP addresses have been assigned on the Cisco router or switch interface. Note the router identification address, which you will use when configuring WCCPv2 on the BIG-IP system.

Task list

Creating a VLAN for a one-arm deployment
Creating a self IP address for a one-arm deployment
Defining a route
Configuring WCCPv2
Verifying connectivity
Verifying WCCPv2 configuration for one-arm deployment
Creating an iSession connection
Validating iSession configuration in a one-arm deployment
Configuring the Cisco router for a one-arm deployment using WCCPv2
Viewing pertinent configuration details from the command line

Creating a VLAN for a one-arm deployment

For a one-arm deployment, you create only one VLAN on the BIG-IP® system, because the system has only a single connection to the WAN router or switch.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type `wan`.
4. In the **Tag** field, type a numeric tag, from `1` to `4094` for the VLAN, depending on your network configuration.
5. For the **Interfaces** setting, click an interface number in the **Available** list, and move the selected interface to the **Untagged** or **Tagged** list, depending on your network configuration.
6. Click **Finished**.

The screen refreshes, and displays the new VLAN from the list.

Creating a self IP address for a one-arm deployment

A VLAN must be configured before you create a self IP address.

This self IP address is the local endpoint for the iSession™ connection.

1. On the Main tab, click **Network** > **Self IPs**.
2. Click **Create**.
   The New Self IP screen opens.
3. In the **Name** field, type a descriptive name for the self IP address, for example `onearm`.
4. In the **IP Address** field, type an IP address that is not in use and resides on the `wan` VLAN you created. In the example shown, this is `10.150.3.1`.
5. In the **Netmask** field, type the full network mask for the specified IP address.

   For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
6. From the **VLAN/Tunnel** list, select **wan**.
7. From the **Port Lockdown** list, select **Allow None**.

   This selection avoids potential conflicts (for management and other control functions) with other TCP applications. However, to access any of the services typically available on a self IP address, select **Allow Custom**, so that you can open the ports that those services need.
8. In the **Traffic Group** field, clear the check box, and select **traffic-group-local-only (non-floating)** from the drop-down menu.
9. Click **Finished**.
   The screen refreshes, and displays the new self IP address.

The self IP address is assigned to the external (WAN) VLAN.

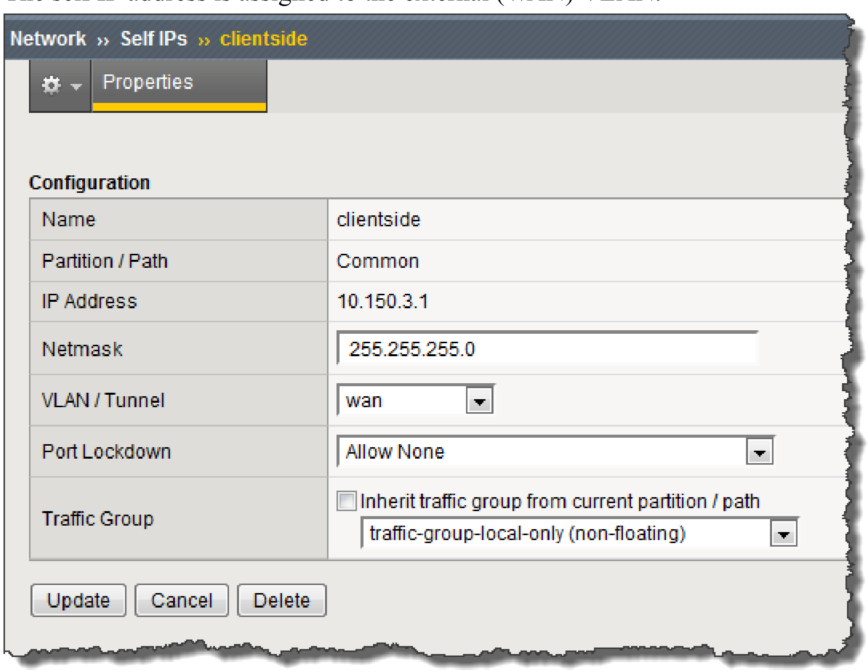


Figure 38: Example of the Properties screen for the self IP address you created

Use this self IP address on the WAN Optimization Quick Start screen for the **WAN Self IP Address**, which is the local endpoint for the iSession connection.

Defining a route

You must define a route on the local BIG-IP® system for sending traffic to its destination. In the example shown, the route defined uses the default gateway to send traffic to the router.

1. On the Main tab, click **Network** > **Routes**.
2. Click **Add**.
   The New Route screen opens.
3. In the **Name** field, type `default-gateway`.
4. In the **Destination** field, type the IP address `0.0.0.0`.
   An IP address of `0.0.0.0` in this field indicates that the destination is a default route.
5. In the **Netmask** field, type `0.0.0.0`, the network mask for the default route.
6. From the **Resource** list, select **Use Gateway**.
   The gateway represents a next-hop or last-hop address in the route.
7. For the **Gateway Address** setting, select **IP Address** and type an IP address. In the example shown, this is `10.150.3.254`.

Configuring WCCPv2

To configure traffic redirection using WCCPv2 for a one-arm deployment, follow these steps on the BIG-IP® system. This implementation specifies the Layer 2 (L2) method of traffic forwarding and mask assignment as the load-balancing method for a WCCPv2 service.

__Note:__ The values you select for __Redirection Method__, __Return Method__, and __Traffic Assign__ are automatically selected by the Cisco router or switch, provided that the Cisco device supports these settings.

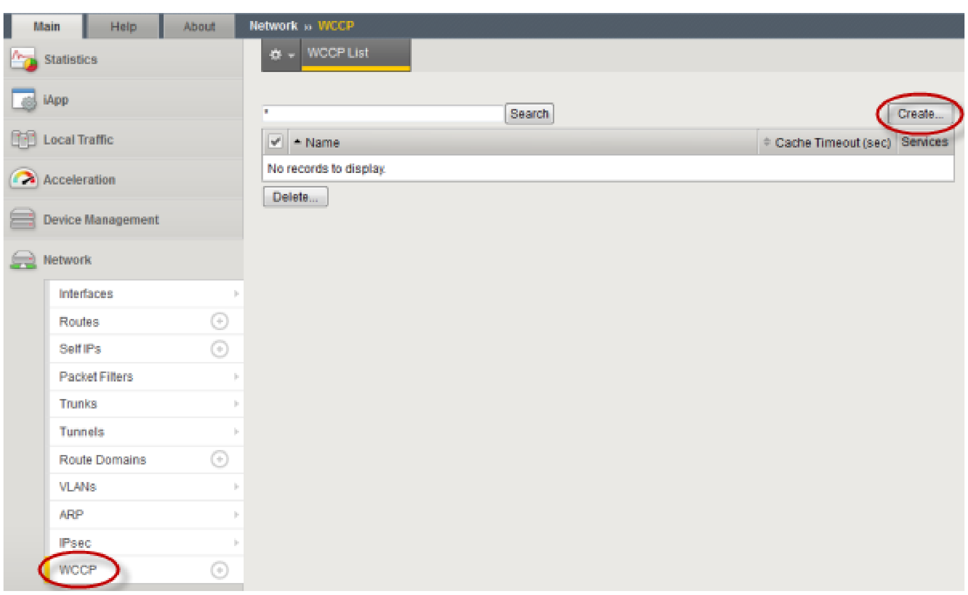


Figure 39: Example showing browser interface for configuring WCCP

1. On the Main tab of the BIG-IP® system user interface, click **Network** > **WCCP**.
2. Click the **Create** button.
   The New WCCP List screen opens.
3. In the **Service Group** field, type a name for the service group, for example, `service-wccp`.
4. In the **Service** field, type a service group identifier, which is a number between `51` and `255`.

   This number must match the service ID you configure on the Cisco router. In the illustration shown, this number is `75`.
5. From the **Port Type** list, select **Destination**.

   If you specify a port in the **Port List** , this setting specifies the port on which the server listens for incoming traffic that has been redirected by WCCP. For best results, select **Destination**, even if you do not specify a port.
6. From the **Redirection Method** list, select **L2**.

   This setting specifies the method the router uses to redirect traffic to the BIG-IP system. Typically, L2 has a faster throughput rate than GRE, but GRE traffic has the advantage that it can be forwarded by a Layer-3 router. This example uses **L2**.

   Note: *The router or switch uses the same redirection method, if supported.*
7. From the **Return Method** list, select **L2**.

   This setting specifies the method the BIG-IP system uses to return pass-through traffic to the router. Typically, L2 has a faster throughput rate than GRE, but GRE traffic has the advantage that it can be forwarded by a Layer-3 router. This example uses **L2**.

   Note: *The router or switch uses the same return method, if supported.*
8. From the **Traffic Assign** list, select **Mask**.

   This setting specifies whether load balancing is achieved by a hash algorithm or a mask. This example uses a mask.

   Note: *The router or switch uses the same setting, if supported.*
9. In the **Routers** field, type the IP address of the Cisco router, and click **Add**.

   In the illustration shown, this is `10.150.3.254`.

   Important: *Do not use a secondary IP address for the Cisco router or switch.*
10. In the **Port List** field, select an application, or leave it blank to indicate all ports.
11. For the **Router Identifier** setting, type the Router Identifier IP address of the router.

    If you do not know the Router Identifier IP address, consult the Cisco documentation that applies to the router or switch you are using.
12. In the **Client ID** field, type the IP address of the VLAN that connects to the Cisco router.

    In the illustration shown, this is `10.150.3.1`.
13. Click **Finished**.

The BIG-IP is configured for WCCPv2 traffic redirection in a one-arm deployment. The completed screen looks similar to the following example.

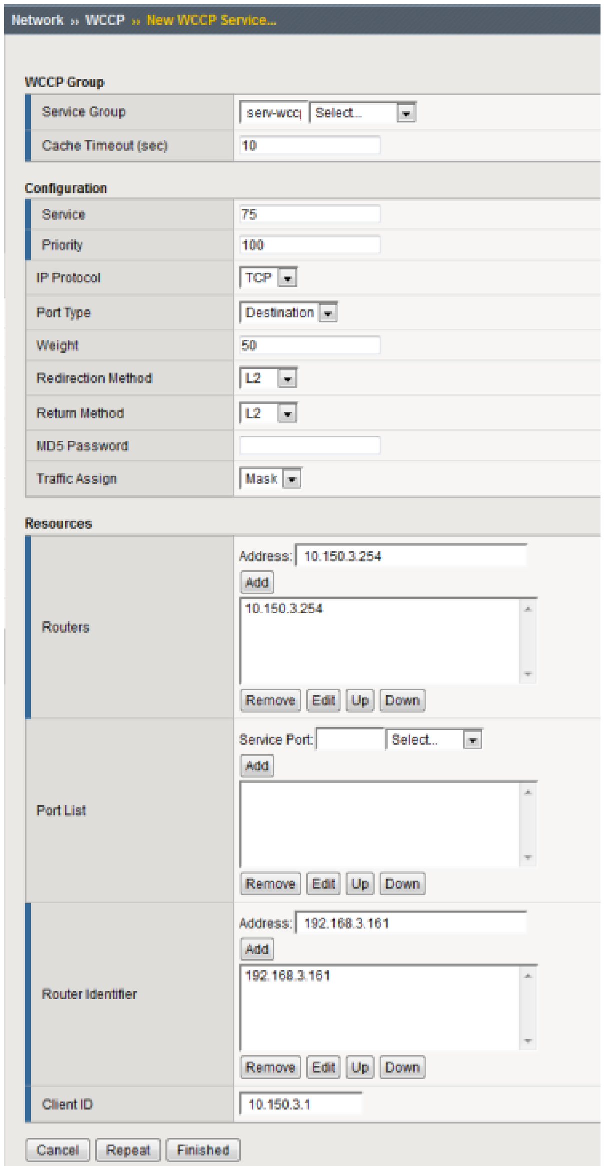


Figure 40: Example of completed configuration screen

Verifying connectivity

Important: *Use this task as a checkpoint before proceeding with the one-arm setup.*

You can verify connectivity from the command-line interface.

1. Ping the router interface using the command-line access to the BIG-IP® system.
2. Use TCPdump on TCP traffic between the servers at both sites to verify that TCP packets are redirected when you initiate TCP traffic.
3. Review the log `/var/log/wccpd.log` and look for the `SESSION up` message.

The following example is an excerpt from the log of a one-arm configuration.

```
Aug  2 17:26:18 clientside3600 notice router_ip 10.150.3.254
Aug  2 17:26:18 clientside3600 notice ports: 0,0,0,0,0,0,0,0,
Aug  2 17:26:18 clientside3600 notice tunnel_remote_addr: 192.31.3.161
Aug  2 17:26:18 clientside3600 notice
Aug  2 17:26:18 clientside3600 notice wccpd-1[1db1:f73f46d0]
WccpMcpInterface.cpp:113 :
Aug  2 17:26:18 clientside3600 notice wccpd-1[1db1:f73f46d0] WccpApp.cpp:208
 : Failover status active 0
Aug  2 17:26:18 clientside3600 notice wccpd-1[1db1:f73f46d0] WccpApp.cpp:208
 : Failover status active 1
Aug  2 17:26:18 clientside3600 notice wccpd-1[1db1:f73f46d0]
ServiceGroup.cpp:194 : Sending Wccp Capabilities Service group 75, Forwarding
 Type: L2, Return Type: L2, Assignment Type: MASK
Aug  2 17:26:18 clientside3600 notice wccpd-1[1db1:f73f46d0]
ServiceGroup.cpp:468 : Final Wccp Capabilities Service group 75, Redirection:
 L2, Return: L2, Traffic Assign: MASK
Aug  2 17:26:18 clientside3600 notice wccpd-1[1db1:f73f46d0]
ServiceGroup.cpp:615 : SESSION up
```

Verifying WCCPv2 configuration for one-arm deployment

You can use the command line interface to verify the WCCPv2 configuration on the BIG-IP® system.

1. Log on to the command-line interface using the root account.
2. At the command prompt, type `tmsh list net wccp`, and verify the WCCP values you configured.
   A listing similar to the following appears.

```
            net wccp server-wccp
              services
                75
                   port-type dest
                   redirection-method l2
                   return-method l2
                   routers { 10.150.1.254 }
                   traffic-assign mask
                   tunnel-local-address 10.150.3.1
                   tunnel-remote-addresses { 10.150.2.1 }
```

Creating an iSession connection

You cannot view the Quick Start screen until you have defined at least one VLAN and at least one self IP on a configured BIG-IP® system that is provisioned for symmetric optimization.

Use the Quick Start screen to set up symmetric optimization for a one-arm deployment.

1. Log in to the BIG-IP system that you want to configure.
   The default login value for both user name and password is `admin`.

2. On the Main tab, click **Acceleration** > **Quick Start** > **Symmetric Properties**.
3. In the **WAN Self IP Address** field, type the local endpoint IP address.

   In the example shown, this is `10.150.3.1`.
4. Verify that the **Discovery** setting is set to **Enabled**.

   If you disable the **Discovery** setting, or discovery fails, you must manually configure any remote endpoints and advertised routes.
5. In the **Select VLANs** field, select the `wan` VLAN for both the **LAN VLANs** and **WAN VLANs** settings.

   You select only one VLAN, because the system has only a single connection to the WAN router or switch.
6. Click **Apply**.

This example shows a completed Quick Start screen.

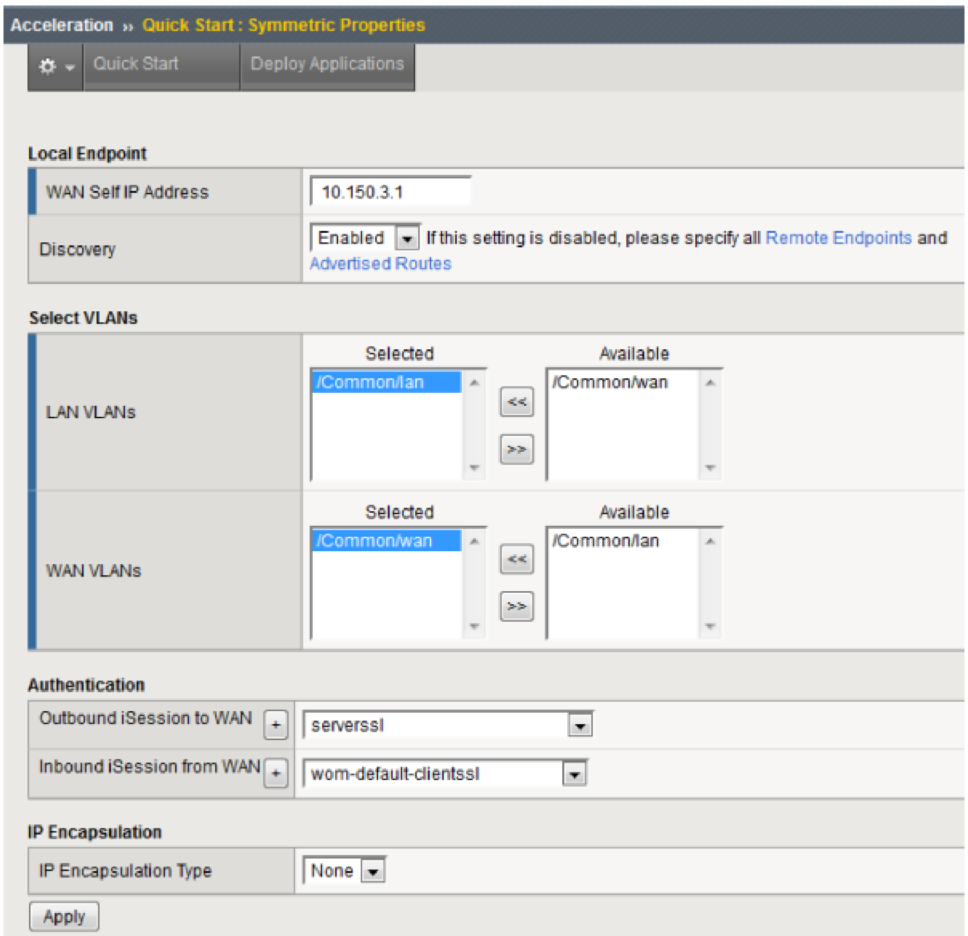


Figure 41: Example of completed Quick Start screen

After you configure the iSession™ endpoints, use an iApp template to select the application traffic for optimization. Click **Acceleration** > **Quick Start** > **Deploy Applications**. Click **Create**, from the **Template** list select **f5.replication**, and follow the online instructions.

Validating iSession configuration in a one-arm deployment

At this point,you have finished configuring BIG-IP® systems at opposite sides of the WAN, and the systems have discovered their remote iSession™ endpoints.

Important: *Use this task as a checkpoint to allow for troubleshooting before you complete the setup.*

You can validate the configuration using the browser and command-line interfaces.

1. Run diagnostics to verify the configuration.
   a) On the Main tab, click **Acceleration** > **Symmetric Optimization** > **Diagnostics**.
   b) Next to **Diagnose WOM Configuration**, click **Run**.
   c) Correct any configuration errors as indicated on the screen.

2. Transfer data between the servers at the two sites, and verify that the transfer was successful.
3. Using the command-line interface, enter `tmsh show wom remote-endpoint all`, and verify the remote endpoint IP address and the `STATE: Ready` message.
   The following listing is an example of the results for this command.

```
------------------------------------------------------------
Remote endpoint: 10.150.2.1                    □-----------
------------------------------------------------------------
Status
    HOSTNAME: server_bridge3600.example.net
    MGMT ADDR: 192.X.X.X  VERSION: 11.4.0
    UUID: 195f:74a0:d242:eab6:57fe:c3a:c1d2:6e22
    enabled                      STATE: ready □-----------
    BEHIND NAT: no
    CONFIG STATUS: none
    DEDUP CACHE: 43.5G
    REFRESH count: 0             REFRESH timestamp: 12/31/12 16:00:00
    ALLOW ROUTING: enabled

------------------------------------------------------------
    Endpoint Isession Statistic: _tunnel_data_10.150.2.1
------------------------------------------------------------
Connections                         Current  Maximum   Total
    Connections OUT IDLE:                 0        0       0
    Connections OUT ACTIVE:               1        1       1
    Connections IN ACTIVE:                0        0       0
Direction                            Action      Raw     Opt
    Out (to WAN) bits        Deduplication      880    1.2K
    Out (to WAN) bits          Compression     1.2K    1.2K
Direction                            Action      Opt     Raw
    In (from WAN) bits       Decompression   273.9M  273.8M
    In (from WAN) bits       Deduplication   272.6M  272.5M
```

4. Using the browser interface, view the green status indicator on the Remote Endpoints screen.
5. On the Main tab, click **WAN Optimization** > **Dashboard**, and view the traffic optimization data.

Configuring the Cisco router for a one-arm deployment using WCCPv2

To configure traffic redirection using Web Cache Communication Protocol version 2 (WCCPv2) for a one-arm deployment, follow these steps on the Cisco router.

1. Configure the service ID that you configured on the BIG-IP® device.
   a) Enable WCCP globally.
   b) In Command mode, configure the service ID; for example, `75`.

      In the example shown, the command line might look like the following.

```
(config)#ip wccp 75
```

2. Using the router interface that is connected to the client from which you want to redirect traffic, associate the VLAN with the service ID you configured.

   In the example shown, the command-line interface might look like the following.

```
                 (config)#interface vlan 254
                 (config)#ip wccp 75 redirect in
```

The following listing is an example of the information displayed for a Cisco router configured to redirect traffic to the BIG-IP system using WCCPv2.

```
Clientside_Top_switch#sh run
Building configuration...
Current configuration : 4848 bytes
version 12.2
no service pad
hostname Clientside_Top_switch
!
no aaa new-model
switch 1 provision ws-c3750g-48ts
system mtu routing 1500
vtp mode transparent
ip subnet-zero
ip routing
ip wccp 75
!
interface GigabitEthernet1/0/4
 switchport access vlan 200
 switchport mode access
!
interface GigabitEthernet1/0/5
 switchport access vlan 100
 switchport mode access
!
interface GigabitEthernet1/0/6
!
interface GigabitEthernet1/0/7
 switchport access vlan 254
 switchport mode access
!
interface Vlan1
 ip address 192.31.3.161 255.255.255.0
!
interface Vlan100
 ip address 10.15.3.254 255.255.255.0
!
interface Vlan200
 ip address 10.15.2.254 255.255.255.0
!
interface Vlan254
 ip address 10.15.1.254 255.255.255.0
```

```
 ip wccp 75 redirect in
!
```

Viewing pertinent configuration details from the command line

You can view details of the BIG-IP® iSession™ configuration from the command line.

1. Log on to the command-line interface of the BIG-IP system using the root account.
2. At the command prompt, type `tmsh`.
3. At the command prompt, type `list all-properties`.
   The following listing is an example of the pertinent information displayed for a one-arm configuration.

```
ltm profile tcp wom-tcp-lan-optimized {
    abc enabled
    ack-on-push enabled
    app-service none
    close-wait-timeout 5
    cmetrics-cache disabled
    congestion-control high-speed
    defaults-from tcp-lan-optimized
    deferred-accept disabled
    delay-window-control disabled
    delayed-acks disabled
    description none
    dsack disabled
    ecn disabled
    fin-wait-timeout 5
    idle-timeout 600
    init-cwnd 0
    init-rwnd 0
    ip-tos-to-client 0
    keep-alive-interval 1800
    limited-transmit enabled
    link-qos-to-client 0
    max-retrans 8
    md5-signature disabled
    md5-signature-passphrase none
    nagle enabled
    partition Common
    pkt-loss-ignore-burst 0
    pkt-loss-ignore-rate 0
    proxy-buffer-high 1228800
    proxy-buffer-low 98304
    proxy-mss disabled
    proxy-options disabled
    receive-window-size 65535
    reset-on-timeout enabled
    rfc1323 enabled
    selective-acks enabled
    selective-nack disabled
    send-buffer-size 65535
    slow-start disabled
    syn-max-retrans 3
    syn-rto-base 0
    tcp-options none
    time-wait-recycle enabled
    time-wait-timeout 2000
    verified-accept disabled
    zero-window-timeout 20000
}
```

```
ltm profile tcp wom-tcp-wan-optimized {
    abc enabled
    ack-on-push disabled
    app-service none
    close-wait-timeout 5
    cmetrics-cache enabled
    congestion-control high-speed
    defaults-from tcp-wan-optimized
    deferred-accept disabled
    delay-window-control disabled
    delayed-acks disabled
    description none
    dsack disabled
    ecn disabled
    fin-wait-timeout 5
    idle-timeout 600
    init-cwnd 0
    init-rwnd 0
    ip-tos-to-client 0
    keep-alive-interval 1800
    limited-transmit enabled
    link-qos-to-client 0
    max-retrans 8
    md5-signature disabled
    md5-signature-passphrase none
    nagle enabled
    partition Common
    pkt-loss-ignore-burst 8
    pkt-loss-ignore-rate 10000
    proxy-buffer-high 196608
    proxy-buffer-low 131072
    proxy-mss disabled
    proxy-options disabled
    receive-window-size 2048000
    reset-on-timeout enabled
    rfc1323 enabled
    selective-acks enabled
    selective-nack enabled
    send-buffer-size 2048000
    slow-start disabled
    syn-max-retrans 3
    syn-rto-base 0
    tcp-options none
    time-wait-recycle enabled
    time-wait-timeout 2000
    verified-accept disabled
    zero-window-timeout 300000
}
ltm virtual isession-virtual {
    app-service none
    auth none
    auto-lasthop default
    clone-pools none
    cmp-enabled yes
    connection-limit 0
    description none
    destination 10.150.3.1:any
    enabled
    fallback-persistence none
    gtm-score 0
    http-class none
    ip-protocol tcp
    last-hop-pool none
    mask 255.255.255.255
    mirror disabled
    nat64 disabled
    partition Common
    persist none
    pool none
```

```
    profiles {
        isession {
            context clientside
        }
        wom-default-clientssl {
            context clientside
        }
        wom-tcp-lan-optimized {
            context serverside
        }
        wom-tcp-wan-optimized {
            context clientside
        }
    }
    rate-class none
    rules none
    snat none
    source-port preserve
    traffic-classes none
    translate-address enabled
    translate-port disabled
    vlans none
    vlans-disabled
}
net interface 1.1 {
    app-service none
    description none
    enabled
    flow-control tx-rx
    force-gigabit-fiber disabled
    mac-address 0:1:d7:79:9a:84
    media none
    media-active 1000T-FD
    media-fixed auto
    media-max 1000T-FD
    media-sfp auto
    mtu 1500
    prefer-port sfp
    stp enabled
    stp-auto-edge-port enabled
    stp-edge-port true
    stp-link-type auto
    vendor none
}
net route def {
    description none
    gw 10.150.3.254
    mtu 0
    network default
    partition Common
}
net self "clientside Self" {
    address 10.150.3.1/24
    allow-service none
    app-service none
    description none
    floating disabled
    inherited-traffic-group false
    partition Common
    traffic-group traffic-group-local-only
    unit 0
    vlan wan
}
net vlan wan {
    app-service none
    auto-lasthop default
    description none
    failsafe disabled
    failsafe-action failover-restart-tm
```

```
    failsafe-timeout 90
    interfaces {
        1.1 {
            app-service none
            untagged
        }
    }
    learning enable-forward
    mtu 1500
    partition Common
    source-checking disabled
    tag 4094
}
sys datastor {
    cache-size 1066
    description none
    disk enabled
    high-water-mark 90
    low-water-mark 80
    store-size 97152
}
sys disk application-volume datastor {
    logical-disk HD1
    owner datastor
    preservability discardable
    resizeable false
    size 97152
    volume-set-visibility-restraint none
}
sys management-route default {
    app-service none
    description none
    gateway 192.31.3.129
    mtu 1500
    network default
}
sys provision wom {
    app-service none
    cpu-ratio 0
    description none
    disk-ratio 0
    level nominal
    memory-ratio 0
}
sys provision woml {
    app-service none
    cpu-ratio 0
    description none
    disk-ratio 0
    level none
    memory-ratio 0
}
wom deduplication {
    description none
    dictionary-size 256
    disk-cache-size 97152
    enabled
    max-endpoint-count 1
}
wom endpoint-discovery {
    auto-save enabled
    description none
    discoverable enabled
    discovered-endpoint enabled
    icmp-max-requests 1024
    icmp-min-backoff 5
    icmp-num-retries 10
    max-endpoint-count 0
    mode enable-all
```

```
}
wom local-endpoint {
    addresses { 10.150.3.1 }
    allow-nat enabled
    description none
    endpoint enabled
    ip-encap-mtu 0
    ip-encap-profile { /Common/default-ipsec-policy-isession }
    ip-encap-type ipsec
    no-route passthru
    server-ssl serverssl
    snat none
    tunnel-port https
}
wom profile isession isession-http {
    adaptive-compression enabled
    app-service none
    compression enabled
    compression-codecs { deflate lzo bzip2 }
    data-encryption disabled
    deduplication enabled
    defaults-from isession
    deflate-compression-level 1
    description none
    mode enabled
    partition Common
    port-transparency enabled
    reuse-connection enabled
    target-virtual virtual-match-all
}
wom remote-endpoint 10.150.2.1 {
    address 10.150.2.1
    allow-routing enabled
    app-service none
    description none
    endpoint enabled
    ip-encap-mtu 0
    ip-encap-profile none
    ip-encap-type default
    origin manually-saved
    server-ssl none
    snat default
    tunnel-encrypt enabled
    tunnel-port https
}
wom server-discovery {
    auto-save enabled
    description none
    filter-mode exclude
    idle-time-limit 0
    ip-ttl-limit 5
    max-server-count 50
    min-idle-time 0
    min-prefix-length-ipv4 32
    min-prefix-length-ipv6 128
    mode enabled
    rtt-threshold 10
    subnet-filter none
    time-unit days
}
```

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system is configured in a one-arm deployment. For symmetric optimization, you must also configure the other side of the WAN. The other BIG-IP deployment can be in bridge, routed, or one-arm mode.

Legal Notices and Acknowledgments

Legal Notices

Publication Date

This document was published on November 28, 2017.

Publication Number

MAN-0379-08

Copyright

Trademarks

Patents

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Gabriel Forté.

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes software with glib library utility functions, which is protected under the GNU Public License.

This product includes software with grub2 bootloader functions, which is protected under the GNU Public License.

This product includes software with the Intel Gigabit Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes software with the Intel 10 Gigabit PCI Express Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes software under license from Qosmos (www.qosmos.com).

This product includes software developed by Andrew Tridgell, which is protected under the GNU Public License, copyright ©1992-2000.

This product includes software developed by Jeremy Allison, which is protected under the GNU Public License, copyright ©1998.

This product includes software developed by Guenther Deschner, which is protected under the GNU Public License, copyright ©2008.

This product includes software developed by www.samba.org, which is protected under the GNU Public License, copyright ©2007.

This product includes software from Allan Jardine, distributed under the MIT License.

This product includes software from Trent Richardson, distributed under the MIT License.

This product includes vmbus drivers distributed by Microsoft Corporation.

This product includes software from Cavium.

This product includes software from Webroot, Inc.

This product includes software from Maxmind, Inc.

This product includes software from OpenVision Technologies, Inc. Copyright ©1993-1996, OpenVision Technologies, Inc. All Rights Reserved.

This product includes software developed by Matt Johnson, distributed under the MIT License. Copyright ©2012.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,
2. add special version identification to distinguish your version in addition to the base release version number,
3. provide your name and address as the primary contact for the support of your modified version, and
4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product includes software developed by Brian Gladman, Worcester, UK Copyright ©1998-2010. All rights reserved. The redistribution and use of this software (with or without changes) is allowed without the payment of fees or royalties provided that:

- source code distributions include the above copyright notice, this list of conditions and the following disclaimer;
- binary distributions include the above copyright notice, this list of conditions and the following disclaimer in their documentation.

This software is provided 'as is' with no explicit or implied warranties in respect of its operation, including, but not limited to, correctness and fitness for purpose.

This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory. Copyright ©1990-1994 Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
4. Neither the name of the University nor of the Laboratory may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY SONY CSL AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SONY CSL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software incorporates JFreeChart, ©2000-2007 by Object Refinery Limited and Contributors, which is protected under the GNU Lesser General Public License (LGPL).

This product contains software developed by the Mojarra project. Source code for the Mojarra software may be obtained at https://javaserverfaces.dev.java.net/.

This product includes software developed by McAfee®.

This product includes software developed by Ian Gulliver ©2006, which is protected under the GNU General Public License, as published by the Free Software Foundation.

This product contains software developed by the RE2 Authors. Copyright ©2009 The RE2 Authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes the Zend Engine, freely available at http://www.zend.com.

This product includes software developed by Digital Envoy, Inc.

This product contains software developed by NuSphere Corporation, which is protected under the GNU Lesser General Public License.

This product contains software developed by Erik Arvidsson and Emil A Eklund.

This product contains software developed by Aditus Consulting.

This product contains software developed by Dynarch.com, which is protected under the GNU Lesser General Public License, version 2.1 or later.

This product contains software developed by InfoSoft Global (P) Limited.

This product includes software written by Steffen Beyer and licensed under the Perl Artistic License and the GPL.

This product includes software written by Makamaka Hannyaharamitu ©2007-2008.

Rsync was written by Andrew Tridgell and Paul Mackerras, and is available under the GNU Public License.

This product includes Malloc library software developed by Mark Moraes. (©1988, 1989, 1993, University of Toronto).

This product includes open SSH software developed by Tatu Ylonen (ylo@cs.hut.fi), Espoo, Finland (©1995).

This product includes open SSH software developed by Niels Provos (©1999).

This product includes SSH software developed by Mindbright Technology AB, Stockholm, Sweden, www.mindbright.se, info@mindbright.se (©1998-1999).

This product includes free SSL software developed by Object Oriented Concepts, Inc., St. John's, NF, Canada, (©2000).

This product includes software developed by Object Oriented Concepts, Inc., Billerica, MA, USA (©2000).

This product includes free software developed by ImageMagick Studio LLC (©1999-2011).

This product includes software developed by Bob Withers.

This product includes software developed by Jean-Loup Gaily and Mark Adler.

This product includes software developed by Markus FXJ Oberhumer.

This product includes software developed by Guillaume Fihon.

This product includes QPDF software, developed by Jay Berkenbilt, copyright ©2005-2010, and distributed under version 2 of the OSI Artistic License (http://www.opensource.org/licenses/artistic-license-2.0.php).

This product includes JZlib software, Copyright © 2000-2011 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This product includes Apache Lucene software, distributed by the Apache Software Foundation under the Apache License, version 2.0.

This product includes Apache MINA software, distributed by the Apache Software Foundation under the Apache License, version 2.0.

This product includes OData4J software, distributed under the Apache License version 2.0.

This product includes software developed by the Visigoth Software Society (http://www.visigoths.org/).

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes software developed by Addy Osmani, and distributed under the MIT license. Copyright © 2012 Addy Osmani.

This product includes software developed by Charles Davison, and distributed under the MIT license. Copyright © 2013 Charles Davison.

This product includes software developed by The Dojo Foundation, and distributed under the MIT license. Copyright © 2010-2011, The Dojo Foundation.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

This product includes software developed by Douglas Crockford, douglas@crockford.com.

This product includes ec2-tools software, copyright © 2008, Amazon Web Services, and licensed under the Amazon Software License. A copy of the License is located at http://aws.amazon.com/asl/ .

This product includes the ixgbevf Intel Gigabit Linux driver, Copyright © 1999 - 2012 Intel Corporation, and distributed under the GPLv2 license, as published by the Free Software Foundation.

This product includes Apache Ant software, distributed by the Apache Software Foundation under the Apache License, version 2.0.

This product includes libwebp software. Copyright © 2010, Google Inc. All rights reserved.

This product includes isc-dhcp software. Copyright © 2004-2013 by Internet Systems Consortium, Inc. ("ISC"); Copyright © 1995-2003 by Internet Software Consortium.

This product includes jQuery Sparklines software, developed by Gareth Watts, and distributed under the new BSD license.

This product includes jsdifflib software, developed by Chas Emerick, and distributed under the BSD license.

This product includes winston software, copyright © 2010, by Charlie Robbins.

This product includes Q software developed by Kristopher Michael Kowal, and distributed under the MIT license. Copyright © 2009-2013 Kristopher Michael Kowal.

This product includes SlickGrid software developed by Michael Liebman, and distributed under the MIT license.

This product includes JCraft Jsch software developed by Atsuhiko Yamanaka, copyright © 2002-2012 Atsuhiko Yamanaka, JCraft, Inc. All rights reserved.

This product includes DP_DateExtensions software developed by Jim Davis, Copyright © 1996-2004, The Depressed Press of Boston (depressedpres.com). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the DEPRESSED PRESS OF BOSTON (DEPRESSEDPRESS.COM) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

All code not authored by the Depressed Press is attributed (where possible) to its rightful owners/authors, used with permission and should be assumed to be under copyright restrictions as well.

This product includes Boost libraries, which are distributed under the Boost license (http://www.boost.org/LICENSE_1_0.txt).

This product includes Angular software developed by Google, Inc., http://angulargs.org, copyright © 2010-2012 Google, Inc., and distributed under the MIT license.

This product includes node.js software, copyright © Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

c. be redistributable at no charge.

Open Source Software includes, without limitation, software licensed or distributed under any of the following licenses or distribution models, or licenses or distribution models substantially similar to any of the following:

a. GNU's General Public License (GPL) or Lesser/Library GPL (LGPL),
b. the Artistic License (e.g., PERL),
c. the Mozilla Public License,
d. the Netscape Public License,
e. the Sun Community Source License (SCSL),
f. vi) the Sun Industry Source License (SISL),
g. vii) the Apache Software license, and
h. viii) the Common Public License (CPL).

3. OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to materials referenced therein, at any time and without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right or license under Intel patents, copyrights, trademarks, or other intellectual property rights.
4. Entire Agreement. This Agreement contains the complete and exclusive statement of the agreement between You and Intel and supersedes all proposals, oral or written, and all other communications relating to the subject matter of this Agreement. Only a written instrument duly executed by authorized representatives of Intel and You may modify this Agreement.
5. LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety (90) days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.
6. EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND, INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for any errors, the accuracy or completeness of any information, text, graphics, links or other materials contained within the Software.
7. LIMITATION OF LIABILITY. IN NO EVENT WILL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.
8. TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if You violate its terms. Upon termination, You will immediately destroy the Software or return all copies of the Software to Intel.
9. APPLICABLE LAWS. Claims arising under this Agreement will be governed by the laws of Delaware, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

10. GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or their successors. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95054.

Index

M

N

O

P

Q

R

S

T

U