Last updated on: 2024-01-22 11:47:53.

KVM: BIG-IP Virtual Edition User's Guide¶

Version notice:

This content applies to F5® BIG-IP® Virtual Edition (VE) 14.0.0 and later

Get help when working with F5 BIG-IP VE in Linux KVM.

Licensing BIG-IP VE¶

In order to use BIG-IP VE, you must have a license from F5.

To choose the license you need, see this webpage (https://f5.com/products/how-to-buy/simplified-licensing). You can get a trial license (https://www.f5.com/trial/big-ip-trial.php) if you need one.

To learn how to license a BIG-IP, consult the K7752 article (https://my.f5.com/manage/s/article/K7752).

Reusing licenses¶

In BIG-IP VE **version 12.1.3.3+**, and **13.1.0.2**+ ONLY, you can revoke the license from a virtual machine and re-use it on another virtual machine.

- From the Configuration utility, to revoke the license, go to **System -> License** and click **Revoke**.
- From tmsh, to revoke the license, run the command:

  ```
  tmsh revoke sys license
  ```

This functionality works for these BIG-IP VE versions ONLY.

Per-App licenses¶

When you want to use BIG-IP VE with one back-end application, choose a Per-App license.

Per-App licenses are currently available:

- As BYOL (Bring Your Own License)
- With throughput speeds of 25 or 200 Mbps
- In BIG-IP VE 13.1.0.2 and later

This license is available for Local Traffic Manager (LTM) only, or you can use it to run LTM and Application Security Manager (ASM) as a WAF (Web Application Firewall).

Per-App licenses are also used as part of BIG-IP Cloud Edition, to license BIG-IP VE instances when they are auto-scaled from within BIG-IQ.

Per-App licensing virtual server options¶

With a Per-App license, you can create one virtual IP address and three virtual servers.

The three virtual servers in this case would have the same destination IP address, but would use different ports.

For example, you can have the following virtual servers:

Name	Description	Destination Address/Service Port
VS1	Website traffic	10.10.10.10:443
VS2	Redirect for website traffic	10.10.10.10:80
VS3	Website instrumentation/health checking	10.10.10.10:8080

If you'd prefer, the virtual server can be a wildcard. For example, the Destination Address would be 0.0.0.0 instead of 10.10.10.10 in the previous example. You can specify the Service Port, or use *.

Name	Description	Destination Address/Service Port
VS1	Catch-all, log with iRule	0.0.0.0:*
VS2	Website traffic	0.0.0.0:443
VS3	Website instrumentation/health checking	0.0.0.0:8080

Licenses for BIG-IP add-on modules¶

Some add-on modules for BIG-IP will require an add-on license. For complete details, consult the **Activating add-on modules** procedure in the K7752 article (https://support.f5.com/csp/article/K7752#add).

Configure TSO support¶

The TCP Segmentation Offloading (TSO) feature, which includes support for large receive offload (LRO) and Jumbo Frames, is enabled by default.

Note: You must have the admin user role to enable or disable TSO support.

1. Use an SSH tool to access the BIG-IP VE command line.
2. Log in as `root`.
3. Type `tmsh` to open the tmsh utility.

   You will see `tmsh` at the prompt once the utility is opened.

4. To determine whether TSO support is currently enabled, use the command:

   ```
   show sys db tm.tcpsegmentationoffload
   ```

5. To enable support for TSO, use the command:

```
sys db tm.tcpsegmentationoffload enable
```

6. To disable support for TSO, use the command:

   ```
   sys db tm.tcpsegmentationoffload disable
   ```

Turn off LRO or GRO¶

In KVM, there are multiple ways to turn off LRO. However, the most reliable way is to connect to the virtual machine and use the `ethtool` utility.

See the KVM documentation for more details.

Increase performance with multiple queues¶

In KVM, you can increase throughput by enabling multiple queues on the host.

1. Shut down the guest.
2. Edit the guest’s XML by using a tool like virsh edit.
3. In each desired virtio <interface> section, add the following element:

   ```
   <driver name='vhost' queues=’#’/>
   ```

   where # equals the number of queues.

   Note: The default number of queues for a KVM guest is 1. The optimum number of queues varies by deployment and depends on throughput needs and total desired guest footprint.
4. Save the file.
5. Start the guest.

About NIC teaming¶

NIC teaming, also known as *port channels*, *bonding*, or in BIG-IP, *trunking*, is defined by the Link Aggregation standard IEEE 802.1AX-2008 (formerly IEEE 802.3ad). This standard offers both increased bandwidth and link failure redundancy at Layer 2.

In a virtual environment, the same MAC address must be assigned to all teamed/aggregated ports. Either the hypervisor or the guest (in kernel trusted mode) can do this assignment.

SR-IOV uses virtual functions to send packets from two or more separate physical ports to the guest. The hypervisor never sees these packets.

In this case, LACP active monitoring in the guest is not possible because the guest does not receive bridge control packets.

Configure NIC teaming¶

To increase bandwidth and provide link failure redundancy, enable NIC teaming.

1. On the host, complete the following steps.

   a. Enable SR-IOV.

   b. Expose virtual functions (VFs). For example:

   ```
   echo 4 > /sys/class/net/<device name>/device/sriov_numvfs
   ```

   c. Create VLANs for the VFs and assign MAC addresses. For example:

   ```
   ip link set p2p1 vf <VF#> mac "00:01:02:03:00:21"
   ip link set p2p1 vf <VF#> vlan <VLAN ID>
   ```

   VF# is 0, 1, ... and choose an arbitrary MAC address.

   Important: Ensure the MAC address is the same for each teamed NIC.

   d. Add the VFs to the guest.

2. Then on BIG-IP VE, complete the following steps.

   a. Create the VLAN.

   b. Create the trunk. For example:

   ```
   tmsh create net <trunkname> interfaces add { 1.1 1.2 }
   ```

   c. Add the trunk to the VLAN. For example:

   ```
   tmsh create net vlan <vlanname> interfaces add { <trunkname> }
   ```

For more information on creating trunks in BIG-IP, see the trunking documentation on askf5.com (https://support.f5.com/csp/article/K1689).

Increase disk space for BIG-IP VE¶

You must increase disk space on most cloud and hypervisor platforms, if you want to provision multiple modules. For more information, refer to the following topics: Disk usage by the BIG-IP modules with application volumes (https://support.f5.com/csp/article/K17477), Getting Started with BIG-IP Virtual Edition (https://techdocs.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/bigip-ve-setup-vmware-vcloud-11-5-0/1.html), and Overview of BIG-IP VE image sizes (https://support.f5.com/csp/article/K14946). To increase the disk space for BIG-IP VE allowing for additional module provisioning, expand (https://my.f5.com/manage/s/article/K14952) the following VE directory:

- `/appdata`

1. Plan for your disk space requirements (../shared/app_reqmnts.html#modulesysreqs).

2. Extend your disk space for provisioning multiple modules (https://my.f5.com/manage/s/article/K14952).

Before proceeding with the following steps for expanding configuration disk space, use hypervisor tools to expand the disk size for the BIG-IP VE virtual machine and reboot. Then use the BIG-IP VE `tmsh` utility to increase the amount of disk space used by the four BIG-IP VE directories:

- `/config`
- `/shared`
- `/var`
- `/var/log`

Note

At the time of this release, decreasing the VE disk size is not supported.

For each of the previous directories you want to resize, complete the following steps.

1. Use an SSH tool to access the BIG-IP VE tmsh utility.
2. From the command line, log in as root.
3. List the current size of the directories on your disk so you can determine which ones need to be resized.

   ```
   tmsh show sys disk directory
   ```

4. Expand the size of the directories in which you need additional space.

   ```
   tmsh modify sys disk directory <directory name> new-size <new directory size in
   ```

   For example, use `tmsh modify sys disk directory /config new-size 3145740` to increase the size of /config directory to 3145740 1KB blocks (or roughly 3,221,237,760 bytes).

5. To confirm that the command you just submitted is properly scheduled, you can show the new list of directories again.

   ```
   tmsh show sys disk directory
   ```

6. If you change your mind about a submitted size change, you can revoke the size change.

   ```
   tmsh modify sys disk directory /config new-size 0
   ```

   In this example, the size of the `/config` directory is left as is, revoking any scheduled size changes.

After you submit this sequence of `tmsh` commands, the directory size changes will be scheduled to occur the next time the BIG-IP VE virtual machine (VM) is rebooted.

The next time the VM running BIG-IP VE reboots, the changes are applied.

See Also

- Azure: Change the size of a virtual machine (https://learn.microsoft.com/en-us/azure/virtual-machines/resize-vm?tabs=portal)
- AWS EC2: Modify an EBS volume using the console (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/requesting-ebs-volume-modifications.html)
- Google Cloud: Increase the size of a persistent disk (https://cloud.google.com/compute/docs/disks/resize-persistent-disk)
- Alibaba: Resize disks online for Linux and Windows instances (https://www.alibabacloud.com/help/en/elastic-compute-service/latest/resize-disks-online-for-linux-instances)
- Oracle Cloud: Resizing a Volume (https://docs.oracle.com/en-us/iaas/Content/Block/Tasks/resizingavolume.htm)
- IBM Cloud: Resizing a virtual server instance (https://cloud.ibm.com/docs/vpc?topic=vpc-resizing-an-instance&interface=ui)

Change the NIC used for BIG-IP VE management¶

By default, management traffic goes through the `eth0` NIC and data traffic goes through the other available NICs.

If you need to use `eth0` for data traffic, you can change the NIC that management traffic goes through.

1. Use SSH to connect to BIG-IP VE.
2. If you need to determine which NICs are available, stop TMM by typing `bigstart stop tmm`. Then type `ip addr` to view the list of available NICs.
3. Change the management NIC by typing `tmsh modify sys db provision.managementeth value eth1` where `eth1` is the NIC you want to use for management. You can use any available NIC.
4. Press Enter.
5. Reboot BIG-IP VE by typing `reboot` and then pressing Enter.

When BIG-IP VE is running again, you can use `eth0` for data.

Note

If the subnet associated with the management NIC lacks DHCP, you must assign a new IP address by using the BIG-IP Configuration utility tool.

About single NIC and multi-NIC configurations¶

A typical BIG-IP VE configuration can include four NICs: one for management, one for internal, one for external, and one for high availability.

However, if you want to create a VM for a quick test, you can create a configuration with just one NIC. In this case, BIG-IP VE creates basic networking objects for you.

When BIG-IP VE first boots, it determines the number of active NICs. If BIG-IP VE detects one NIC, then:

- Networking objects (vNIC 1.0, a VLAN named Internal, and an associated self IP address) are created automatically for you.
- The port for the Configuration utility is moved from 443 to 8443.

Note

If there is no DHCP server in your environment and no IP address automatically assigned, then the networking objects will not be created and the port will not be moved. As an example, do the following, which uses the same IP address 192.168.80.53/24 for management and self IP:

1. Disable DHCP and enable setting a static address, `tmsh modify sys global-settings mgmt-dhcp disabled`. See this routes topic (https://clouddocs.f5.com/cloud/public/v1/shared/routes.html) for more information.
2. Disable single NIC auto-config, `tmsh modify sys db provision.1nicautoconfig value disable`. See this KVM topic for BIG-IP VE 13.1.X (https://techdocs.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/bigip-ve-setup-linux-kvm-13-1-0/4.html) for more information.
3. Ensure management route will persist, `tmsh modify sys db provision.1nic value forced_enable`.
4. Move management port, `tmsh modify sys httpd ssl-port 8443`. See this K31003634 (https://my.f5.com/manage/s/article/K31003634) article for more information.
5. Add TCP port to the default port lockdown protocols and services, `tmsh modify net self-allow defaults add { tcp:8443 }`.
6. Configure static management IP address, `tmsh create sys management-ip 192.168.80.53/24 description 'provisioned by tmos_static_mgmt'`
7. Create and attach internal VLAN to interface 1.0, `tmsh create net vlan internal { interfaces replace-all-with { 1.0 { } } tag 4094 mtu 1450 }`. Be aware that this configuration my already exist and can produce the following error: “The requested VLAN (/Common/internal) already exists in partition Common.”
8. Create self IP, assign the same IP as the management IP, and assign internal VLAN to default port lockdown policy, `tmsh create net self self_1nic { address 192.168.80.53/24 allow-service default vlan internal }`.

9. Create management route gateway, `tmsh create sys management-route default gateway 192.168.80.1`.

10. Define the TMM default route, `tmsh create net route default network default gw 192.168.80.1`.

11. Save the configuration, `tmsh save sys config base`.

- High availability (failover) is not supported, but config sync is supported.
- VLANs must have untagged interface.

If BIG-IP VE detects multiple NICs, then you create the networking objects manually:

- The port for the Configuration utility remains 443.
- You can change the number of NICs after first boot and move from single to multi-NIC and vice versa.
- VLANs can have tagged interfaces.

About routes in a single NIC configuration¶

If you want to configure a static route that relies on a gateway in the same subnet as the self IP address, you must first disable the setting that enforces single NIC setup:

```
modify sys db provision.1nicautoconfig value disable
```

Confirm that the value is correct by typing:

```
list sys db provision.1nicautoconfig
```

The return value should be `disable`.

If you do not change this value, any time you reboot BIG-IP VE, the manually-configured static route will cause validation errors during `load sys config`.

Change from single NIC to multi-NIC¶

When you initially boot BIG-IP VE, if it recognized only one NIC, then some network configuration was done automatically. If you want to use more than one NIC, complete the following steps.

1. Use an SSH tool to connect to BIG-IP VE.

2. Return to the default configuration.

   ```
   tmsh load sys config default
   ```

3. Save the changes.

   ```
   tmsh save sys config
   ```

4. Set a database variable so that the number of NICs will be recognized on reboot.

```
setdb provision.1nic enable
```

5. Reboot BIG-IP VE.

```
reboot
```

6. When the BIG-IP VE instance is up and running, multiple NICs will be recognized and you can begin configuring BIG-IP VE.

Change from multi-NIC to single NIC¶

If you have a BIG-IP VE configuration with multiple NICs, you can simplify the configuration to have only one NIC. When you boot the BIG-IP VE and only one NIC is recognized, some networking objects are created automatically for you.

1. Use an SSH tool to connect to BIG-IP VE.
2. Return to the default configuration of BIG-IP VE.

```
tmsh load sys config default
```

3. Save the changes.

```
tmsh save sys config
```

4. Set a database variable so that the number of NICs will be recognized on reboot.

```
setdb provision.1nic forced_enable
```

5. Reboot BIG-IP VE.

```
reboot
```

When the BIG-IP VE instance is up and running, it will have a single NIC and related network objects will be created.

See Also

- Update BIG-IP VE (../shared/update.html)
- Linux KVM: BIG-IP VE Setup Guide (kvm_setup.html)