KVM deployments

Cisco IoT FND can be deployed on commercial hypervisors (such as VMware ESXi) and on Kernel-based Virtual Machines (KVM) using supported Linux distributions.

- Cisco IoT FND manages device lifecycles, configurations, firmware, telemetry, and alarms across distributed field networks.
- KVM provides an open, Linux-native virtualization layer, enabling flexible resource allocation and performance for production or lab deployments.

Feature name	Release	Description
KVM deployments	Cisco IoT FND Release 5.1.0	Deploy Cisco IoT FND on KVMs using supported Linux distributions.

Verify QCOW2 signature

This task guides you to run the signature verification program to verify QCOW2 signature.

Before you begin

- Python 2.7.x
- OpenSSL
- Tar package
- Verification scripts running on your premises need internet connectivity to reach Cisco to download root and sub-CA certs.

Here are the steps to verify the QCOW2 signature:

Procedure

Step 1 Unzip the downloaded release file `CISCO-IOTFND-VPI-K9-5.1.0-155-SHA512-QCOW2.zip`

You should obtain the following two files:

- `qcow2-5.1.0.155.tar.xz`
- `qcow2-5.1.0.155.tar.xz.signature`

Step 2 Run the signature verification script `python2 cisco_x509_verify_release.py -e FND_RPM_SIGN-CCO_RELEASE.pem -s qcow2-5.1.0.155.tar.xz.signature -i qcow2-5.1.0.155.tar.xz -v dgst -sha512` to verify the integrity of the tar.xz file.

Example:

```
python2 cisco_x509_verify_release.py -e {PUBLIC_KEY_PEM} -s {DATA_FILE}.signature -i {DATA_FILE} -v
 dgst -sha512

Retrieving CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...
Successfully retrieved and verified crcam2.cer.
Retrieving SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...
Successfully retrieved and verified innerspace.cer.
Successfully verified root, subca and end-entity certificate chain.
Successfully fetched a public key from FND_RPM_SIGN-CCO_RELEASE.pem.
Successfully verified the signature of qcow2-5.1.0.155.tar.xz using FND_RPM_SIGN-CCO_RELEASE.pem
```

You've successfully verified the QCOW2 signature.

What to do next

Deploy Cisco IoT FND on a KVM.

Deploy Cisco IoT FND on a KVM

This task guides you to deploy Cisco IoT FND on a KVM using Linux distributions.

Before you begin

- Log in to the IP address of KVM RHEL server running 9.6 using a web browser with your credentials (username and password).
- Ensure to install the `cockpit-machines` package.
- Ensure your distribution includes a Postgres DB with PSK mode enabled by default, which establishes tunnels using pre-shared keys rather than certificates, since PSK mode does not support certificate-based tunnels.
- Ensure that the system requirements for the various device scales. For more information see, System Requirements in Cisco IoT FND 5.1 Release.

Here are the instructions to deploy Cisco IoT FND on a KVM:

Procedure

Step 1 Create virtual machines using the import a disk image option on RHEL 9 web console. For more information see, Creating Virtual Machines.

Step 2 Manage virtual CPUs using the RHEL 9 web console. For more information see, Managing virtual CPUs.

Step 3 View and edit the VM interface information using the RHEL 9 web console. For more information see, View and edit virtual network information.

Step 4 From the **Interface type** drop-down list, select **Direct attachment**. Select the interface on which you configured the KVM IP from the **Source** drop-down list. Leave the **Model** and **MAC address** fields with the default values.

Step 5 (Optional) Add an additional network interface based on your requirement.

Step 6 Start the VM. For more information see, Starting VM using the web console.

Step 7 View the serial console of the VM and log in to the RHEL server using your default credentials.

```
User Name: root

Default Password: C!sco123
```

Note
If you do not wish to enable root login for SSH (by keeping `PermitRootLogin no`), check if the `fnduser` account is available. You can use the `fnduser` account for SSH access instead of root. Ensure that fnduser has appropriate permissions (such as sudo access) to perform necessary tasks.

```
User Name: fnduser
Default Password: C!sco123
```

a) Ensure to reset the default password.

Step 8 Configure IPv4 address using `nmcli` command and verify the rechability. For more information see, Configure network interfaces using nmcli command.

Step 9 Enable SSH and edit the `/etc/ssh/ssh_config` file. Locate the parameter `PermitRootLogin` in the serial console to change **no** to **yes**.

If you prefer not to enable root login `PermitRootLogin no`, you can use the `fnduser` account for SSH access, provided it is enabled and has the required permissions.

Restart the SSH service using the `systemctl restart sshd.service` command.

a) Restart the SSH service using the `systemctl restart sshd.service` command.

You've successfully installed Cisco IoT FND on KVM.

What to do next

Create and Edit a swap file on Linux.

Create and activate a swap file

You can create a swap file to create a temporary storage space on a solid-state drive or hard disk when the system runs low on memory.

Before you begin

Ensure to have enough disk space.

Procedure

Step 1 Create and enable a swap file. For more information see, Creating and activating a swap file.

Step 2 Verify if the swap file is active or not using the `swapon --show` command.

- Do we really need swap on modern systems? For more information see, Do we really need swap on modern systems?
- In scale setups, swap space helps systems handle memory demands that exceed physical RAM by temporarily offloading inactive memory pages to disk. This approach supports the stability and scalability of applications and services.
- You've created and activated a swap file.