Unique Requirements for IOS XE Service Containers

- YAML (derived from LibVirt XML) header file(s) within the OVA
  - Outlines the resource requirements for the application so the system knows what to do with it.
  - Memory, storage, CPU shares, CDROM ISO, etc.
- Properly formatted disk image
  - Supported formats are qcow2, raw and raw with Cisco capacity XML tag
- IDE virtio driver within the VM kernel for disk access
- Optional TTY0 and TTY1 specification for console/aux connection

Mandatory Service Container OVA Contents

- YAML Descriptor File Defining:
  - Number of VCPUs and Share of CPU cycles
  - Memory
  - Disks including size and source image if applicable
  - Virtual NICs
  - Console/Aux connectivity

- Disk Image – One or more disk image files.
  - ISO: Supported for read-only file systems like a CDROM.
  - RAW: Supported for read-write file systems.
  - QCOW2: Supported for read-write with compression. Longer initial install time but much smaller disk images as a result of compression. Generally the recommended format for standard disk images.
- Manifest File – Simple text file with the SHA1 hash for all files in the OVA.
- Version File – Simple text file with application version number.

Example YAML File

```
manifest-version: 1.0

info:
  name: kvm_prof_2
  description: "KVM Montavista Test Distro
  version: 1.0
  author-name: Cisco Systems, Inc.
  author-link: "http://www.cisco.com"

app:
  # Indicate app type (vm, paas, lxc etc.,)
  apptype: vm

  resources:
   cpu: 6
   memory: 262144
   vcpu: 1
```

```
disk:
  - target dev: hdc
    file: montavista.iso
  - target dev: sda
    file: kvm_storage_4000MB.img
    upgrade-model: ha-sync

 interfaces:
  - target-dev: net1
    alias: net1
  - target-dev: net2
    type: management

 serial:
  - serial
  - console

# Specify runtime and startup
startup:
  runtime: kvm
  boot-dev: cdrom
```

App Info & Definition

Memory/CPU Reservation

Disk(s) Definition

Ethernet Interfaces

Serial Devices

Boot Details

Example libvirt.xml File

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0' id='1'>
<name>ubuntuserver</name>
<uuid>cdc7b1e3-4a61-8452-98cd-2932f8d781da</uuid>

<memory>262144</memory>
<currentMemory>262144</currentMemory>

<vcpu>1</vcpu>

<os>
  <type arch='x86_64' machine='pc-0.12'>hvm</type>
  <bootdev='hd'/>
</os>

<features>
  <acpi/>
  <pae/>
</features>
<clock offset='localtime'/>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>destroy</on_crash>

<devices>
  <emulator>/usr/bin/qemu-kvm</emulator>

<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2'/>
  <source file='UbuntuServer.qcow2'/>
  <target dev='hdb' bus='virtio'/>
  <alias name='virtio-0-0-4'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</disk>
<controller type='ide' index='0'>
  <alias name='ide0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
</controller>
```

```
<interface type='network'>
  <mac address='52:54:00:89:c4:96'/>
  <source network='default'/>
  <target dev='net1'/>
  <model type='virtio'/>
  <alias name='net1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>

<serial type='tcp'>
  <source mode='bind' host='' service='4444'/>
  <target port='0'/>
  <protocol type='telnet'/>
  <alias name='serial0'/>
</serial>
<serial type='tcp'>
  <source mode='bind' host='' service='4445'/>
  <target port='1'/>
  <protocol type='telnet'/>
  <alias name='serial1'/>
</serial>
<serial type='unix'>
  <source mode='bind' path='syslog'/>
  <target port='2'/>
  <alias name='serial2'/>
</serial>
<serial type='unix'>
  <source mode='bind' path='logger'/>
  <target port='3'/>
  <alias name='serial3'/>
</serial>

</devices>
<qemu:commandline>
<qemu:arg value='-cpu'/>
  <qemu:arg value='host'/>
  <qemu:arg value='-device'/>
  <qemu:arg value='usb-tablet'/>
</qemu:commandline>

</domain>
```

Same VM Definition as Previous Slide

Potential Security Holes

Useful Open Source Tools for Developers

virt-manager – GUI Linux tool for creating and managing VMs.

qemu-img – Useful tool for converting disk images

Example: qemu-img convert -p -c -f raw -O qcow2 <raw.img> <qcow2.img>

openssl – Generates manifest file.

Example: openssl sha1 *.qcow2 *.ver *.yaml > vm.mf

tar – An OVA is nothing more than a tar file with a fancy name.

Example: tar -cvf VM.ova vm.qcow2 platform.xml 4300.xml 4400.xml vm.mf

create_ova.sh – Cisco script to help build an ova in one step.