How to setup your own OVA/KVM server

Foreword

First, let's keep in mind this is not a generic open source cloud/virtualization service. The OVA deployment service on a KVM server is unique to Calix, designed for efficient deployment of AXOS Sandbox OVA images built from the Calix CI process. The basic deployment principle for the KVM server is as follows:

1. **It should have sufficient CPU/MEM/Disk resrouces.**
   a. The smallest OVA instance would often require 2 CPU, 4G mem, and 8+GB disk.
   b. A large E9 sandbox OVA (with multiple shelves/slots) may require up to 10 CPU, 20G mem, and 60+GB disk.
   c. We don't recommend a KVM server with less than 8 CPU, 16G mem, and 100GB disk (SSD preferred)
2. **It must be accessible on the Calix.local network**
   a. Proper DNS configuration is required for bamboo download and OVA auto-license to work
   b. It will preferably have a static IP, but dhcp is acceptable
   c. 1G uplink connection is recommended (OVA images can be 1-5GB in size)
3. **It should not be a shared server for other mission critical services**
   a. A shared user/passwd is required for script automation
   b. The server is typically not managed and not trusted by IT
   c. No AXOS (or other product) source code should be on the server
4. **OVA instances on the server is firewalled from Calix network**
   a. Access from OVA to calix/internet is typically NATed
   b. OVA name or ip addrs are not directly reachable on the Calix network
   c. OVA test traffic does not leak to the Calix network
5. **The maintainer of the KVM server should have access to sandbox-caal git repo**
   a. to check out the deployment server script
   b. to get feature update and bug fixes

Installation Overview

Here are the installation components:

1. Ubuntu 18.04 LTS (HWE kernel preferred) HWE=Hardware Enhanced Kernel. Newer kernels are backported into LTS. 5.3 is the latest kernel. After the base install if not prompted during the installation, use apt-get update to get the new HWE kernel. Google for details.
   - Both kernel 4.15 and 5.3 has been tested.
   - LVM disk partition should be used during installation. Installer defaults to a single partition, and this is fine. Use entire disk. Subsequent installer scripts will take care of resizing in the event of new disk addition later.
   - Server BIOS settings recommended as follows:
     a. enable hyper-threading
     b. enable virtualization features (VT-d, SVM, VMX, IOMMU, x2apic, etc)
     c. switch to legacy bios (to work around UEFI issues, if any)
2. Sandbox-caal
   - GIT checkout (ssh://git@stash.calix.local:7999/tools/sandbox-caal.git) preferred.
   - Alternatively, it can be copied from another server (but will be hard to update feature/bug-fix later)
3. Local sandbox license server (optional)
   - Useful for a completely self contained OVA deployment
   - License server may require additional configuration
   - Should not be used outside of Calix network
4. Exa-ova and ecrack scripts (optional)
   - Useful for local ova deployment
   - scripts copied from various calix repos and requires some manual updates
   - Should not be used outside of Calix network
5. smx-scale deployment tools (optional)
   - Useful for ova batch deployment
   - Scripts copied from axos-auto repo

Installation details

Note:

- to expedite server-to-server install replication, a snapshot archive /home/calix/kvm_server.tgz is available.

- Copy/untar this file to ~calix/ the new server after completing the linux install.

Step-1: Install Ubuntu Server 18.04 LTS

- BIOS configuration for AMD server (GigaByte R282-Z92):
  1. switch from UEFI to Legacy boot (if Linux can't seem to detect any nvme drives)
  2. enable AMD CBS->Performance ->Accept->CCD/Core/Thread Enablement
  3. enable AMB CBS->Local APIC Mode->x2APIC
  4. enable AMD CBS→NBIO -> IOMMU
- BIOS configuration recommended for Intel Server:
  1. enable HT (hyperthreading)
  2. enable x2apic (or extended APIC)
  3. enable VT-x
  4. enable VT-d
  5. enable NUMA
- Install Ubuntu (HWE kernel if available)
  1. Enable dhcp networking
  2. Create user/passwd calix/calix123
  3. use all available disk partition using LVM (note: by default Ubuntu will only use 4GB of disk)
  4. enable ssh server
  5. HWE kernel (if not available via the install usb):
     ```
     sudo apt install --install-recommends --linux-generic-hwe-18.04
     reboot
     ```

Step-2: Check out sandbox-caal git repo

- `login to the new server with user/passwd: calix/calix123`
- `git clone ssh://git@stash.calix.local:7999/tools/sandbox-caal.git`
- `edit ~/sandbox-caal/caal_ova/config/SBox-KVM.conf, verify the following:`

  ```
  [server]
  server_port = 20000
  image_expire_days = 30
  instance_expire_days = 999
  image_dir = /home/calix/ova_downloads
  instance_dir = /home/calix/ova_instance
  instance_pool = kvm
  ```

Note: since the server will likely not have permission to access to stash.calix.local, the following workaround can be used:

- checkout sandbox-caal repo in a IT managed calix build VM (e.g. as <user_name> on <buildhost_name>)
- login as user calix on the KVM server:

  git clone git+ssh://<user_name>@<buildhost_name>/<path_to_sandbox-caal_repo_dir>

- this will proxy git access via a checked out repo on your own build VM.

Step-3: Run host install script

- `~/sandbox-caal/caal_ova/kvm/host_install.sh`
- `reboot`
- `verify server is running`

-
  ```
  $ ~/sandbox-caal/caal_ova/caal_ova_client.py -s localhost
  server uptime: 0 days 5:28:55
  server config:
      server_port = 20000
      socat_port = 40000
      image_expire_days = 30
      instance_expire_days = 999
      image_dir = /home/calix/ova_downloads
      instance_pool = kvm
      provisioned_instances = 900

  hypervisor stats:
  Total VMs: 0
  Total CPUs: 128
   06:13:51 up  5:30,  2 users,  load average: 1.19, 1.32, 1.34
  ```

Step-4: Install local sandbox license server (optional)

Note:

-- Local license server can be used if the machines doesn't have a Calix network connection.

-- This area is still under development, so instructions below will be updated.

- `Copy ~/sandbox-caal/licmake dir from a working kvm server`
- `cd ~/sandbox-caal/licmake; ./install`

Step-5: Install Exa-ova and ecrack scripts (optional)

Note:

-- these scripts are not used by OVA server, and only needed at the client side to launch OVA instances

-- having these scripts on the OVA server allows it to launch/access OVA locally

-- this area is still under development, so instructions below will be updated.

- `Copy ~/bin from a working kvm server`
- `echo 'PATH=~/bin:$PATH' > ~/.bashrc`

Step-6: Install SMX scale deployment tools (optional)

Note:

-- these scripts are used by SMX team to perform large scale batch OVA deployment

-- this area is still under development, so instructions below will be updated.

- `Copy ~/smx-scale from a working kvm server, or`
- `Copy get it from axos-auto repo: (http://stash.calix.local/projects/SQA/repos/axos-auto/browse/tools/smx-scale`

Server Maintenance

For now, maintenance is still manual. Below are some key points:

- `cd ~/sandbox-caal; git pull  ## To update the server scripts (not in effect until script restart)`
- `cd ~/sandbox-caal/caal_ova; sudo ./kvm_ova_server.sh restart  ## To restart the server script`
- `cat /var/tmp/SBox-KVM.log  ## To see log file`

Important notes:

1. Server script restart is required to effect changes in config/SBox-KVM.conf
2. After server script daemon restart, all existing OVAs will need to be re-created (due to libvert network re-config)
3. If a physical server is rebooted, all previous running OVA will be in 'shut off' state, and need to be rebooted manually
4. OVA serial console is available via ***virsh console kvm-ova-<n>***
5. ~~In order to see the boot console, you must use ***virt-manager --connect qemu:///system --show-domain-console kvm-ova-<n>***~~
6. As of 20.4, each OVA image will have its serial console logged to a file under <$ova_instance>/kvm-ova-<nnn>.disk/console.log
7. There is a secondary serial console available for interactive use, and can be accessed via 'virsh console kvm-ova-<nnn> -devname serial1'