

# An Openstack Integration Method with Linux Containers

A short paper of a new approach for creating Openstack-based test environment without nested KVM

Akos Leiter

Component Development Unit  
Ericsson Hungary  
Budapest, Hungary  
akos.leiter@ericsson.com

Robert Fekete

Component Development Unit  
Ericsson Hungary  
Budapest, Hungary  
robert.a.fekete@ericsson.com

**Abstract**—Today’s telecommunication networks change their traditional hardware-based approach to a software-driven cloud environment. Meanwhile corporate software test solutions have been being adopted for fulfilling the new requirements, mainly with the help of virtualization. Openstack, as a cloud orchestration framework, requires at least three specific elements to operate at an acceptable level. Thus for testing purpose if we want to use a single computer, three Kernel-based Virtual Machine (KVM) need to be placed in a single computer. Furthermore it creates a nested virtualization scenario as inside a virtualized Compute Node, there are going to be additional virtual machines. In this paper, we would like to present a solution for replacing the Openstack KVM nodes with Linux containers (LXC) in order to avoid nested KVMs.

**Keywords**—Linux containers, Kernel-based Virtual Machine, Openstack, Cloud testing, Software testing

## I. INTRODUCTION

Virtualization is a key driver for today’s telecommunication research and development. There are many cloud infrastructure which have cloud solution for virtualized telecom equipments as well, e.g.: VMware NSX [3], Openstack [1]. So before releasing a software, it is worth being tested in these special environments so as to avoid integration and execution problems. However maintaining all the cloud environments where a software could run, could cost high amount of human work, time and money, not mentioning the large number of required hardware. That’s why many simplified, “mini” cloud have been created for development and testing purposes, i.e.: DevStack [4] for Openstack. DevStack can run on a simple laptop where a Kernel-based Virtual Machine (KVM) [5] has been started, and all the required Openstack services take place for ensuring proper developing. But inside this KVM, there may be other DevStack-instantiated KVMs for testing, developing or examining cloud features. The proposed new approach replaces the KVM layer with Linux container [6] in order to save resources and simplify the test architecture.

The aim of this paper is to present a new approach for Openstack-based testing and give advice for being able to

recreate the proposed test environment for others’ own purposes.

In section II, there is a short introduction of virtualization technologies (KVM and LXC). Section III presents a basic Openstack architecture. Section IV deals with Openstack-based testing in general. Section V describes the replacement of KVM with LXC, and the obstacles we have experienced. Some initial measurement results are shown in Section VI and in Section VII, draws the conclusion and describes the way forward.

## II. BACKGROUND

### A. Kernel-based Virtual Machines

KVM is a virtualization solution for Linux-based operating systems. `kvm.ko` provides the basic virtualization infrastructure as a loadable kernel module for Intel (`kvm-intel.ko`) and AMD (`kvm-amd.ko`) processors as well. It enables to run unmodified Linux and Windows systems and also deals with virtualizing network cards, disks, graphic adapters etc.

### B. Linux containers

The aim of Linux containers [6] is to be as close as possible to normal Linux installation without the need of having a separate kernel. It could be considered as being between chroot and a standard virtual machine. On the other hand, Linux container are a couple of namespaces managed together.

## III. BASIC OPENSTACK INTRODUCTION AND ARCHITECTURE

Openstack is an open source Infrastructure-as-a-Service provider framework relying on Linux environment. There are three different, designated nodes for a minimum working scenario [2].

### A. Controller node

A collection of management services, responsible for Identity Service (Keystone), Image Service (Glance) and control part of Compute (Nova) and Network (Neutron). It also

deals with SQL service, message queues and Network Time Protocol (NTP) [11].

### B. Compute node

The hypervisors reside on Compute Nodes. Openstack supports a large scale of hypervisors, such as Libvirt [14], Hyper-V [15], VMware [17] and Xen [16]. From design point of view, the number of Compute nodes can be very huge as their resources are offered mainly during a basic instantiation. Security groups are also provided by Nova.

### C. Network node

Neutron is responsible for creating the required (overlay) network topologies by tenants. Network Address Translation (NAT) [7] and Dynamic Host Configuration Protocol (DHCP) [8] are also placed here.

## IV. TESTING WITH OPENSTACK

A normal way of software testing in Openstack environment can be very hardware-intensive as at least three nodes need to exist (mentioned in Section III). Of course, this approach extrapolates many other problems such as financial and storage (Figure 1). (In a test setup, multiplication of Openstack node can be eliminated).

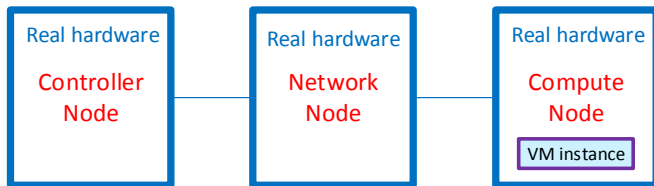


Figure 1. Basic Openstack test environment illustration

At this point, virtualization can be a solution for reducing the number of required hardware. In this scenario, virtual machines can act as an Openstack designated node (depicted in Figure 2.).

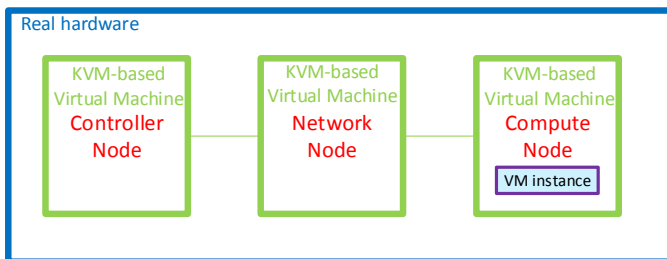


Figure 2. KVM-based virtualized Openstack test environment illustration

Each node has its own Linux kernel, including the underlying operating system, altogether four. But the amount of required hardware has decreased to one blade. Test virtual machines are placed in Compute Node KVM, thus a so called Nested KVM scenario has been created.

Openstack also allows to couple all the specific nodes to one virtual machines in order to get a development tool

(DevStack). But deep examining of this architecture is out of scope of this paper.

## V. PROPOSED TEST ARCHITECTURE

Nested KVM solution may be very slow because of the four running kernels. But Linux containers avoid running additional kernels while keeping the process separation feature. The proposed test architecture deals with replacing virtual machines with Linux containers (Figure 3).

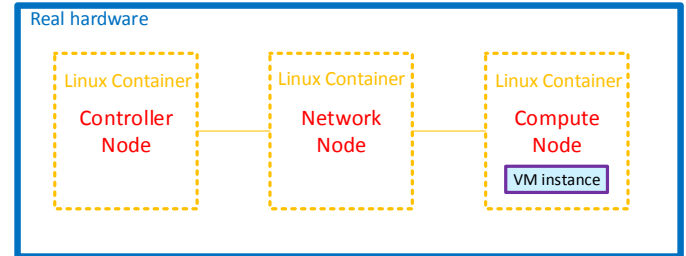


Figure 3. LXC-based virtualized Openstack test environment illustration

### A. Building Blocks for reproduction

The proposed test architecture has been verified on Ubuntu 14.04 and Ubuntu 15.04 as well. First of all, make sure all the required interfaces are up and running between the LXCs. It is quite important not to have any bridge created by brctl [12] on the host OS; ovs-vsctl [13] is recommended. Further general suggestion is to turn off Apparmor [10] for all the containers (mainly Network node) in order to ensure the creation of namespaces. As Neutron using network namespace for running a router, Apparmor might reject the setting of a nested network namespace. (LXC, itself, has already been given a network namespace and it would fork another one for a router as well). On the Compute Node, there are many devices and parameters which need to be set:

- Create Network Block Devices (NBD) if needed with the following instruction:

```
mknod /dev/nbd0 b 43 0
mknod /dev/nbd1 b 43 16
...
mknod /dev/nbd15 b 43 240
chmod 660 /dev/nbd*
chgrp disk /dev/nbd*
```

- Create the following devices:

```
mknod /dev/net/tun c 10 200
mknod /dev/kvm c 10 232
chgrp kvm /dev/kvm
chmod 660 /dev/kvm
```

- On Ubuntu 14.04 the below line might need to be uncommented in `/etc/init/nova-compute.conf` (it is not possible to call kernel modules in LXC and causes error during startup):

```
modprobe nbd
```

- Change the following lines in these configuration files:

```
/etc/libvirt/qemu.conf:
security_driver="none"
```

```
/etc/nova/nova.conf: libvirt_driver=qemu
```

### B. vCPU pinning

It is worth considering to improve the performance of the system to pin virtual CPUs (vCPU) to physical CPUs (pCPU). With this approach, the overhead of vCPU-pCPU binding can be avoided and more resource remain for running instances. From Openstack Kilo release it is possible to use this KVM feature on Openstack Compute Node. See Ref [9].

LXCs use the same CPU set as host OS (that is why using more compute nodes are not recommended in proposed setup as they share and see the same compute resources opposite of nested KVM scenario). Thus on the host OS it is worth separating CPU cores for the kernel and make sure the Nova configuration file of Compute Node Linux Container does not use those CPU cores for CPU pinning.

Example:

*grub.cfg* of host OS:

```
linux /vmlinuz-3.13.0-32-generic root=UUID=XXX ro
isolcpus=5,6,7,8
```

a line in nova.conf of Compute Node LXC:

```
vcpu_pin_set=4,5,6,7,8
```

## VI. MEASUREMENT RESULTS

The above mentioned possible improvements with LXCs have been verified with a simple performance test.

### Test case description:

The building time of the same Linux kernel with the same kernel configuration has been measured in the following environments:

- a native KVM on Ubuntu 15.04
- an LXC-based KVM on Ubuntu 15.04
- Nested KVM setup on Ubuntu 15.04

There was no vCPU pinning used. All the “tenant” instances had two vCPUs and 2048 MB RAM.

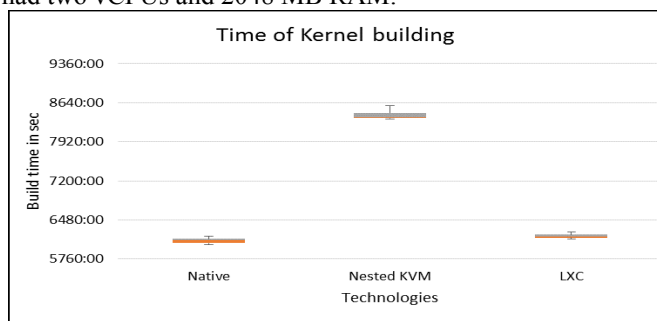


Diagram 1. Kernel build time results

During the test setup, default kernel configuration was used and the output of “time make” command served the building time.

From the initial measurement results (Diagram 1) the time of kernel building takes about 40% more in the case of nested KVM than LXC based. The main advantage comes from the decreased number of running kernels because only one is required. In this context, Linux container is a well-fitted process separation tool. So tenant instances almost run on the same level to not having LXC.

## VII. CONCLUSION

The above mentioned measurement results show that how much resource can be saved by using the LXC-based approach instead of nested KVMs. Those who are lack of hardware resources can apply this new way of Openstack-based test environment. It is simple enough to use it on a single laptop or also capable to run it on a one-blade system with general hardware. The functionality of LXC-based environment is the same, but the cost is lower compared to nested KVMs. It is also possible to run function and system level test as well. The proposed test environment can also support relatively complex networking.

Further results are expected with comparing all of the executed measurements with CPU pinning scenarios as well.

## VIII. REFERENCES

- [1] Openstack : <https://www.openstack.org/> ; Last visited: 2015.10.25
- [2] Openstack Kilo Installation Guide: [http://docs.openstack.org/kilo/install-guide/install/apt/content/ch\\_overview.html](http://docs.openstack.org/kilo/install-guide/install/apt/content/ch_overview.html) ; Last visited: 2015.10.25
- [3] VMware NSX: <http://www.vmware.com/hu/products/nsx> ;Last visited: 2015.10.25
- [4] Devstack: <http://docs.openstack.org/developer/devstack/> Last visited: 2015.10.25
- [5] Kernel Based Virtual Mashine: [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page) Last visited: 2015.10.25
- [6] Linux Containers: <https://linuxcontainers.org/> ; Last visited: 2015.10.25
- [7] P. Srisuresh, M. Holdrege (Lucent Technologies): RFC 2663, IP Network Address Translator (NAT) Terminology and Considerations, August 1999
- [8] S. Alexander (Silicon Graphics, Inc.), R. Droms (Bucknell University):RFC 2132,DHCP Options and BOOTP Vendor Extensions, March 1997
- [9] vCPU guide: <http://redhatstackblog.redhat.com/2015/05/05/cpu-pinning-and-numa-topology-awareness-in-openstack-compute/> ; Last visited: 2015.10.25
- [10] Apparmor: <https://wiki.ubuntu.com/AppArmor> ; Last visited: 2015.10.25
- [11] D. Mills, U. Delaware, J. Martin, Ed. (ISC), J. Burbank, W. Kasch (JHU/APL): RFC 5905, Network Time Protocol Version 4: Protocol and Algorithms Specification
- [12] brctl man page: [http://linuxcommand.org/man\\_pages/brctl8.html](http://linuxcommand.org/man_pages/brctl8.html) ;Last visited 2015.10.25
- [13] ovs-vsctl man page: <http://openvswitch.org/support/dist-docs/ovs-vsctl.8.txt> ; Last visited 2015.10.25
- [14] Libvirt: <http://libvirt.org/> ; Last visited 2015.10.25
- [15] Hyper-V: <http://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx> ; Last visited: 2015.10.25
- [16] Xen: <http://www.xenproject.org/> ; Last visited: 2015.10.25
- [17] VMware: <http://www.vmware.com/> ; Last visited 2015.10.25