

Experimental Cloud Using Commodity Hardware

*A Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of*

Bachelor of Technology

by

Kaushal Kishore
(111601008)

under the guidance of

Dr. Sandeep Chandran



INDIAN INSTITUTE
OF TECHNOLOGY
PALAKKAD

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

*This is to certify that the work contained in this thesis entitled “**Experimental Cloud Using Commodity Hardware**” is a bonafide work of **Kaushal Kishore (Roll No. 111601008)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Palakkad under my supervision and that it has not been submitted elsewhere for a degree.*

Dr. Sandeep Chandran

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Palakkad

Contents

1	Introduction	1
1.1	Cloud	1
1.2	Cloud Services	2
1.2.1	SaaS	2
1.2.2	PaaS	2
1.2.3	IaaS	2
1.3	Experimental Cloud using Commodity Hardware	2
1.4	Organization of The Report	3
2	MaaS : Metal as a Service	5
2.1	Bare metal cloud	5
2.2	IaaS vs. MaaS	6
2.3	Canonical's MAAS	6
3	MAAS in VENV - I	7
3.1	Software Requirements	7
3.2	Hardware Requirements	7
4	MAAS in VENV - II	9
4.1	Creating an Isolated Network	9
4.2	Creating a maas controller	10

Experimental Cloud using Commodity Hardware	Contents
4.3 Installing ubuntu server image	11
4.4 Setup MAAS	12
4.5 SSH & DHCP configuration for MAAS	14
5 MAAS in VENV - III	17
6 Conclusion and Future Work	19
References	21

Chapter 1

Introduction

1.1 Cloud

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet.

Cloud services refer to any IT services that are provisioned and accessed from a cloud computing provider. This is a broad term that incorporates all delivery and service models of cloud computing and related solutions. Cloud services are delivered over the internet and accessible globally from the internet. There are three basic types of cloud services:

- Software as a Service (SaaS)
- Platform as a service (PaaS)
- Infrastructure as a service (IaaS)

1.2 Cloud Services

1.2.1 SaaS

SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the internet. Examples include G Suite – formerly Google Apps, Microsoft Office 365, Salesforce and Workday.

1.2.2 PaaS

PaaS refers to the delivery of operating systems and associated services over the internet without downloads or installation. The approach lets customers create and deploy applications without having to invest in the underlying infrastructure. Examples include Amazon Web Services' Elastic Beanstalk, Microsoft Azure – which refers to its PaaS offering as Cloud Services – and Salesforce's App Cloud.

1.2.3 IaaS

IaaS involves outsourcing the equipment used to support operations, including storage, hardware, servers and networking components, all of which are made accessible over a network. Examples include Amazon Web Services, IBM Bluemix and Microsoft Azure.

1.3 Experimental Cloud using Commodity Hardware

The objective of this project is to create an experimental cloud by repurposing commodity hardware. The cloud we create would be made available to students as virtual desktops which may be used to host web services which can vary from simple static page to complex web applications.

1.4 Organization of The Report

This chapter provides an overview of cloud computing and cloud services. In the next chapter we will introduce MaaS(Metal as a Service), which is a relatively new approach for cloud based service. In chapter 3, we will discuss some of the tools that we need to be familiar with to break the ice. In chapter 4 and 5, we will discuss the approach by which we can create a MaaS based cloud environment. And finally in chapter 6, we conclude with some future works.

Chapter 2

MaaS : Metal as a Service

IaaS customers are given access to servers which can be dedicated or, more often, virtual and free to install the OS and applications of their choice. The customer doesn't host or manage the underlying infrastructure but is able to use the resources as they wish.

As with all 'as a Service' computing models, customers benefit from access to the resources they need without having to invest in expensive hardware upfront, instead they pay monthly and only for what they use.

2.1 Bare metal cloud

Bare metal cloud is an environment in which physical, dedicated servers can be provisioned to customers with cloud-like ease and speed. Bare metal cloud customers are given access to the entire processing power of individual servers, as well as any storage, networking or other services they require.

Within a bare metal infrastructure there is **no multi-tenanting** (sharing of machines) and the servers provisioned are not virtual ones created on top of any hypervisor.

Customers of bare metal cloud are free to use their dedicated servers in any way they want, including running any OS and applications as well as installing hypervisors to create their own virtual machines if they want. And bare metal cloud is provided as a service.

2.2 IaaS vs. MaaS

Is there any difference between IaaS and MaaS?

This depends on your view point. Many define IaaS as the provision of virtual resources only. Some include dedicated servers in their definition. In our view, bare metal cloud is the true IaaS whereas virtualised versions are really a form of Platform as a Service (PaaS).

In all scenarios you gain access to a server on which you can install and run your chosen OS and applications. In this sense, IaaS and bare metal cloud are the same.

On a virtual IaaS however, you have no knowledge of or control over the actual infrastructure on which your services are built. The provider has control of these and your services are abstracted from them.

With bare metal cloud on the other hand, you are provisioned full dedicated servers, with no virtualisation or sharing. It's up to you how you use these and, in the case of installing a hypervisor, how many virtual machines you run on each.

With bare metal you get control of the full stack, from the tin right up to the user interface, and can optimise utilisation and performance to a granular level, something you simply cannot do in a virtualised environment.

2.3 Canonical's MAAS

<https://maas.io/>

Metal-as-a-Service (MAAS) is a provisioning construct created by Canonical, developers of the Ubuntu Linux-based operating system. MAAS is designed to help facilitate and automate the deployment and dynamic provisioning of hyperscale computing environments such as big data workloads and cloud services.

Chapter 3

MAAS in VENV - I

In this chapter we will discuss some of the tools that we will be using in the next chapter to create a MAAS based virtual cloud environment.

3.1 Software Requirements

- libvirt
- QEMU
- OpenSSH
- Ubuntu 18.04 Server ISO

3.2 Hardware Requirements

It is recommended to have atleast 16 GB RAM and 100 GB free space in the host system for deploying virtual cloud environment.

Chapter 4

MAAS in VENV - II

In this chapter we will discuss the steps required to create a basic MAAS based virtual cloud environment. Remember that the problem statement is to repurpose the commodity hardware to create an experimental cloud. At present we don't have access to those hardwares hence we are conducting our experiments in a virtualized environment.

4.1 Creating an Isolated Network

MAAS uses DHCP and DNS with PXE boot to enlist the nodes, so naturally you could have a conflict if you deploy this on a network with an existing DHCP server. Hence create a new (virtual) network with the following given steps from the QEMU interface:

1. Network name - maasisotest
2. Disable DHCP configuration
3. Choose IPv4 address which is not in use by any other connected network. I prefer to use 10.17.17.0/24.
4. Select forwarding to physical network option and set destination to any physical device and mode to NAT

The network configuration for 'maasisotest' should look similar to Fig. 4.1.

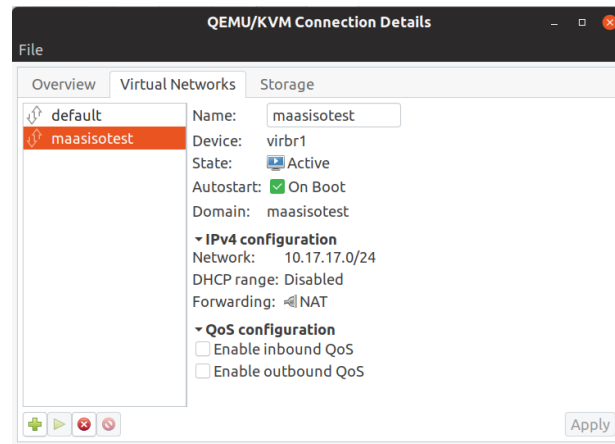


Fig. 4.1 Isolated network configuration

This will essentially create a router at the edge of this virtual network.

4.2 Creating a maas controller

Configurations for creating a new virtual machine with ubuntu 18.04 server installation. This machine will act as controller for the maas nodes.

- Memory - 1536 MB
- CPU Cores - 3
- Storage - 20 GB qcow2
- Name - maas-controller
- Network - maasisotest
- NIC Interface
 - Network source - maasisotest
 - Device model - virtio
- Disk bus - virtio
- Remove unnecessary virtual hardware from the list for eg. sound



Fig. 4.2 maas-controller configuration

The basic configuration for maas-controller should look similar to Fig. 4.2.

Once the configuration is done we can proceed to installation.

4.3 Installing ubuntu server image

During the installation it is expected that DHCP acquisition will fail, since there is no DHCP server on that network. Hence, we need to configure this manually with a static IP address.



Fig. 4.3 IPv4 configuration

Kindly refer to the Fig. 4.3 for the IPv4 configuration.

Once the installation is finished, the system should reboot.

4.4 Setup MAAS

Use the following commands to install some utility packages and maas related packages on maas-controller. Then verify that you are able to connect through ssh from the host.

```
$ sudo apt-get install ssh iptraf htop wget lynx dnsutils
```

```
$ sudo apt-get install maas maas-dhcp maas-dns
```

```
kaushal@maas-controller:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:a2:ef:0c brd ff:ff:ff:ff:ff:ff
    inet 10.17.17.200/24 brd 10.17.17.255 scope global enp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fea2:ef0c/64 scope link
        valid_lft forever preferred_lft forever
kaushal@maas-controller:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.17.17.1     0.0.0.0         UG    0     0        0 enp1s0
10.17.17.0      0.0.0.0       255.255.255.0   U     0     0        0 enp1s0
kaushal@maas-controller:~$ cat /etc/resolv.conf | tail
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.
nameserver 127.0.0.53
options edns0
```

Fig. 4.4 verify internet connectivity

Verify the network connectivity by using the following commands:

```
$ ip addr
```

```
$ route -n
```

```
$ cat /etc/resolv.conf — tail
```

```
$ dig canonical.com
```

```
$ ping google.com
```

The results should look similar to Fig. 4.4.

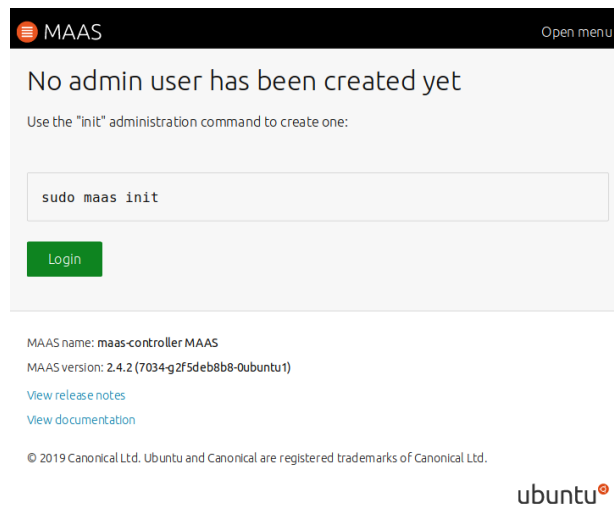


Fig. 4.5 MAAS initial page

Verify that the following link works: `http://10.17.17.200:5240/`. It will open a MAAS interface which looks something similar to Fig. 4.5. Create a super-user by using the following command:

```
$ sudo maas init
```

After setting up the maas superuser open the maas webui and login. You will land up on the page similar to Fig. 4-6.

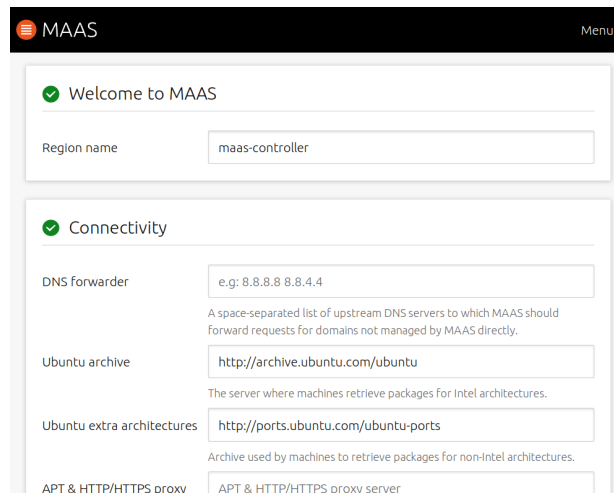


Fig. 4.6 MAAS intro page

4.5 SSH & DHCP configuration for MAAS

Change the default shell of the 'maas' user as shown in the Fig. 4.7.

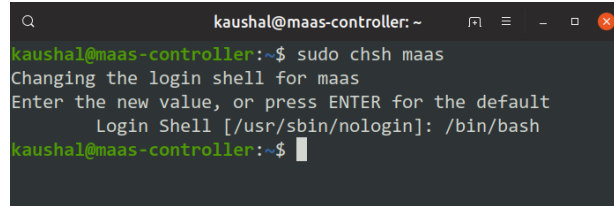


Fig. 4.7 Default shell for maas user

Login to the maas user and generate an ssh key and copy the public key to the host for passwordless ssh authentication. That will enable us to open a ssh session at that virtual machine host without being prompted for a password. Verify that you are able to open an ssh session to the host system from the controller without being asked for the password.

```
$ sudo su - maas
$ ssh-keygen -t rsa
$ ssh-copy-id -i ~/.ssh/id_rsa.pub kaushal@10.128.0.132
```

Verify that maas will be able to control the virtual machines power at the host level. To do that we need to verify that the virsh command is working.

```
virsh -c qemu+ssh://kaushal@10.128.0.132/system list --all
```

It should list all the virtual machines at the host.

The next thing we need to do is to import the ssh key of maas controller into maas, so that when maas provisions a new node and brings up an image on that node it can inject our ssh key allowing us to remotely access those managed nodes without that we really have no way aside from backdoor and breaking in to access those nodes that we have created.

To do this first create a ssh key by following the above mentioned steps.

Go to the following link: <http://10.17.17.200:5240/MAAS/#/intro/user> and upload your ssh key.

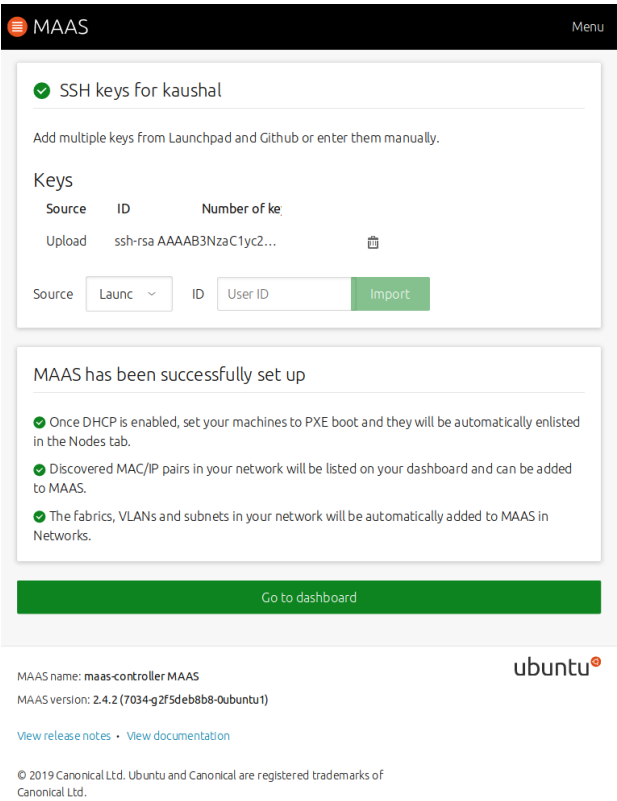


Fig. 4.8 Uploading the ssh key of maas controller

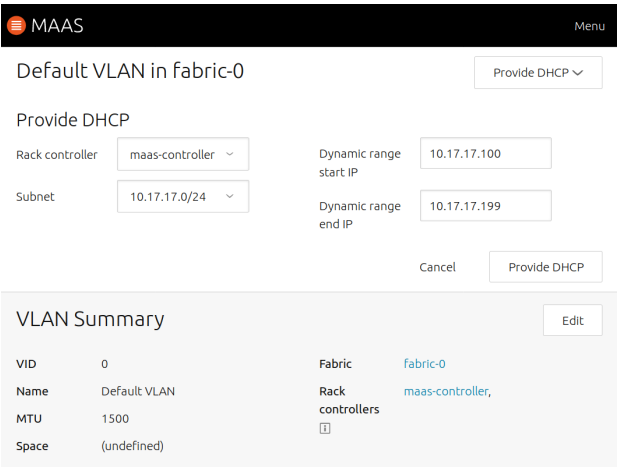


Fig. 4.9 Provide DHCP

Next we need to configure the DHCP configuration for the maas. To enable MAAS-managed DHCP, under the ‘Subnets’ page select the desired fabric and then VLAN and then under the ‘Take action’ button select ‘Provide DHCP’. Refer Fig. 4.9 for DHCP settings.

Finally go to you desired maas controller and verify the status. You should see a lot of green ticks and no red ticks, similar to Fig. 4.10.

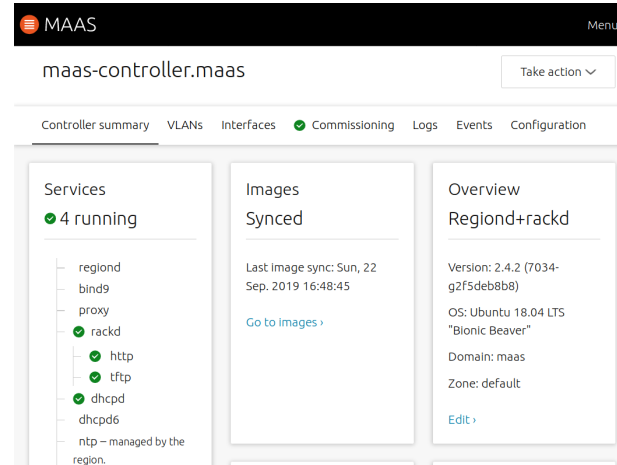


Fig. 4.10 Provide DHCP

With this our basic configuration of the maas controller is done and in the next chapter we will create some nodes.

Chapter 5

MAAS in VENV - III

Chapter 6

Conclusion and Future Work

write results of your thesis and future work.

References