

HP Moonshot

Web Infrastructure in a Box user guide

[Introduction](#)

[Purpose](#)

[Assumptions](#)

[Node 1](#)

[CM \(Chassis Manager\)](#)

[Solution Components](#)

[Moonshot Chassis and M300 Cartridges \(Minimum 5\)](#)

[Ubuntu: The Moonshot OS](#)

[MAAS \(Metal as a Service\)](#)

[Juju](#)

[Equipment & Internet Access Requirements](#)

[Preparing the Environment](#)

[Hardware Setup](#)

[Connecting to the Chassis Manager](#)

[Powering on the first node](#)

[Connecting to the node's virtual serial port and logging into Ubuntu](#)

[Optional: Network Address Configuration](#)

[DNS verification](#)

[Proxy Setup](#)

[Installing and configuring MAAS and Juju](#)

[Installation complete](#)

[Accessing MAAS](#)

[Accessing the Juju GUI](#)

[Deploying the "Web-infrastructure-in-a-Box" bundle](#)

[Deploying the bundle](#)

[Appendix A](#)

[Appendix B](#)

[Juju command line](#)

[References](#)

Introduction

Web-infrastructure-in-a-Box is an effort by HP and Canonical to demonstrate the deployment of a dynamic web workload on HP moonshot hardware at an HP customer's site. After completing the set-up of the software tools required (MAAS/Juju), a user will be able to not only deploy the Web-infrastructure-in-a-Box solution, but also scale services up and down using Juju. The user will be able to destroy the workload and create new 3-tier dynamic web workloads by selecting their own components for the Juju charm store.

Purpose

The Purpose of this document is to guide you through the installation steps of MAAS/JUJU on HP moonshot server. Once MAAS/Juju are installed on Node 1 you will be ready to use Juju to deploy the Web-infrastructure-in-a-Box workload on an HP moonshot server.

Assumptions

This document assumes that the Ubuntu Server 14.04 LTS operating system is already installed and fully functional on Node 1. Node 1 is the first cartridge on the moonshot chassis available in slot 1. A separate set of instructions for downloading and installing the OS will be provided in case Node 1 is shipped without it.

The following is a brief overview of the different nodes and their respective network configurations. We will go through each of them in detail later in this document.

Node 1

- eth0 is "inward" facing, i.e. connected to the internal network. By default, the IP address for this node is 10.14.4.1. MAAS will run dhcp, dns, pxe, and squid on this network. Should you need to change this IP address, please refer to the 'network configuration' section below.
- eth1 is "outward" facing, i.e. it provides an uplink to the internet or any other external networks. Its IP address will be configured via dhcp, and be something like 10.15.225.10. Please contact your lab administrator for the correct IP address.

CM (Chassis Manager)

- Will be connected to the same "outward" facing network as eth1. It will have an IP address such as 16.100.118.140. Ask your lab administrator for the IP address of the CM of moonshot chassis.

Solution Components

Moonshot Chassis and M300 Cartridges (Minimum 5)

A new class of server for Cloud, Mobility, Security, and Big Data is transforming what the business expects from IT resulting in a “New Style of IT.” The HP Moonshot System is modular, allowing data center managers to purchase server cartridges optimized for particular tasks. With 3,600 cores and 14 TB of memory/rack, the Proliant M300 Server Cartridge is ideal for:

- **Cost-Effective, dynamic Web delivery**--Serving rich Web content, e-commerce, images, videos, application downloads and more over the internet at scale.
- **High Availability and performance content for CDN**--Serving content including Web objects, applications, streaming media and social networks.
- **Right-sized IT for Web Hosting**--Delivering just enough capacity to keep utilization rates high, at a low cost.



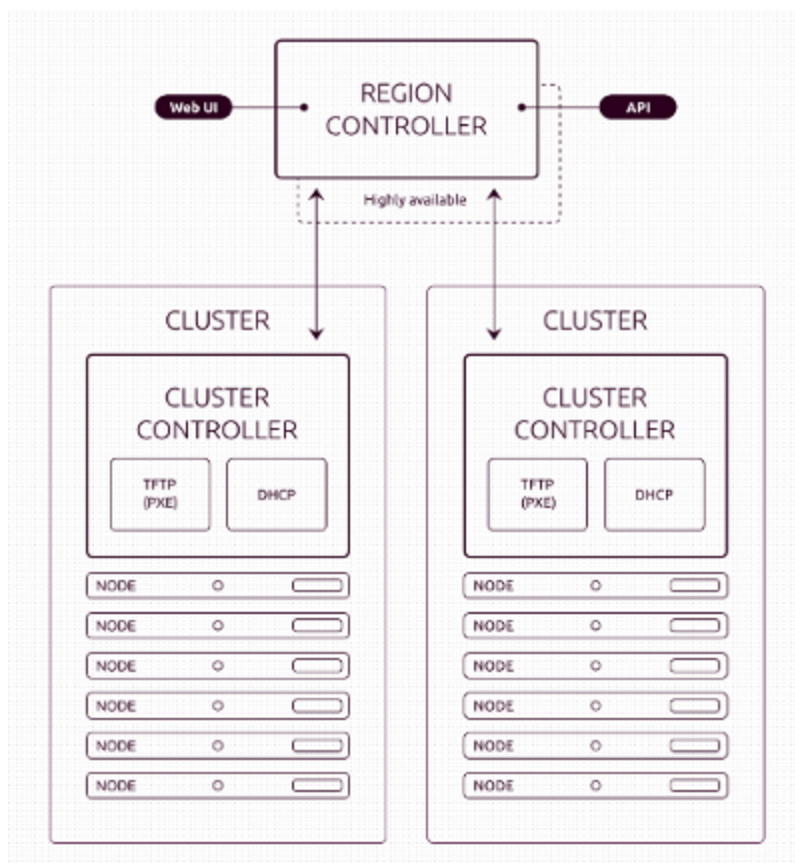
Ubuntu: The Moonshot OS

The leading platform for scale-out computing, Ubuntu Server helps you make the most of your Moonshot infrastructure. Ubuntu Server is the only operating system the boasts full support for every Moonshot server cartridge, including the M300 cartridge which is the target cartridge for the Web-infrastructure-in-a-Box solution.

Whether it's web hosting, big data processing or any other hyperscale application, Ubuntu runs on every Moonshot cartridge and supports every conceivable workload.

MAAS (Metal as a Service)

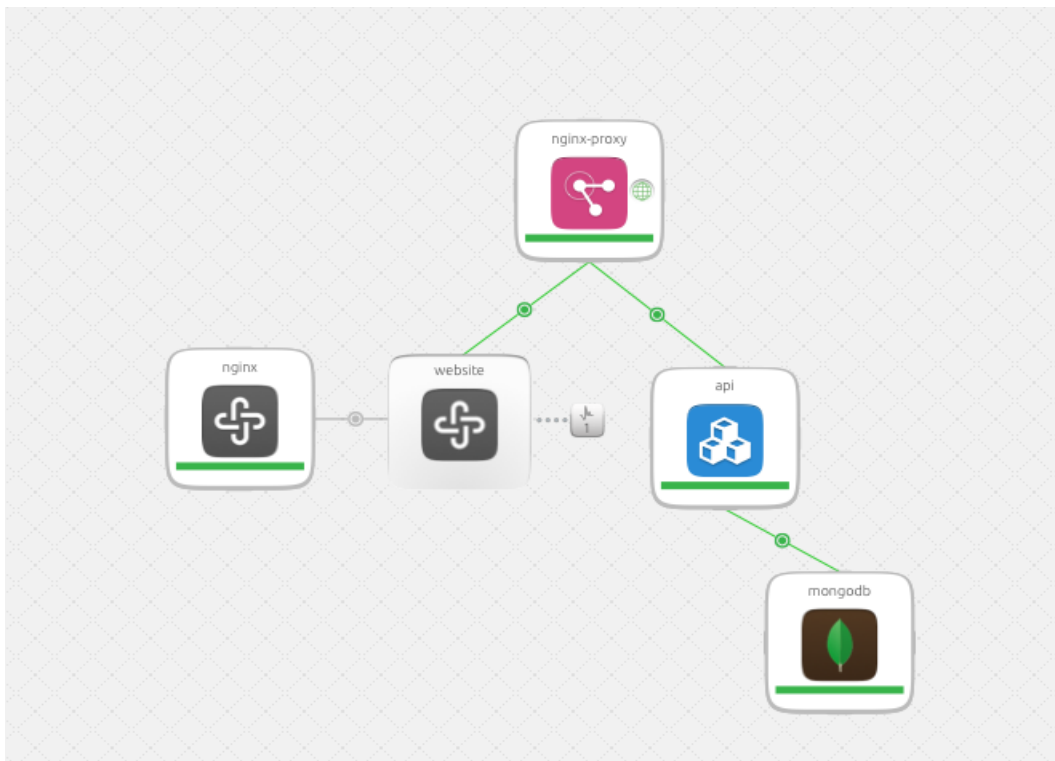
Metal as a Service (MAAS) brings the language of the cloud to physical servers. It makes it easy to set up the hardware on which to deploy any service that needs to scale up and down dynamically; a dynamic web infrastructure such as Web-infrastructure-in-a-Box being just one example. With a simple web interface, you can add, commission, update and recycle your servers (in this case Moonshot cartridges) at will. As your needs change, you can respond rapidly, by adding new nodes and dynamically re-deploying them between services. When the time comes, nodes can be retired for use outside the MAAS. The diagram below shows the MAAS architecture on Node 1.



MAAS works closely with the service orchestration tool [Juju](#) to make deploying services fast, reliable, repeatable and scalable.

Juju

Juju provides both a command-line interface and a powerful GUI to configure, manage, deploy, scale and maintain a workload on your Moonshot infrastructure. Juju charms make sure your services are deployed and configured in the most optimal way for your environment, and automates the arduous tasks of setting up data interfaces between services, so all you have to do is draw a line between two applications to make them communicate. You basically “draw” your desired architecture, and Juju implements it for you.



Equipment & Internet Access Requirements

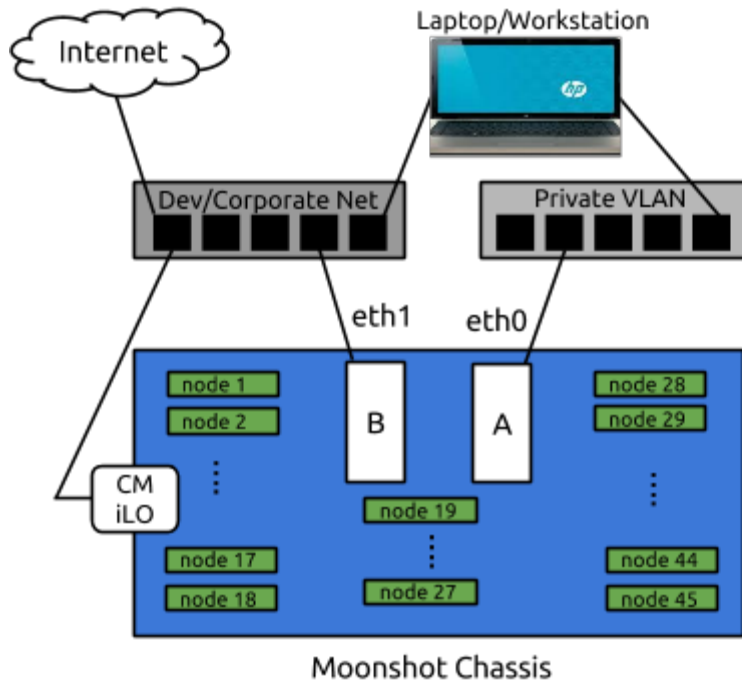
- Any model laptop with dual nic (either two wired network ports or one wired and one wireless if there is wireless access to at least one of the chassis networks).

- Ubuntu Desktop 14.04 LTS is preferable but any OS can be used. The screenshots you will see in this guide are based on Ubuntu.
- Ethernet cables as necessary to connect to the chassis and management controller networks
- Firefox web browser (get the most recent version to ensure full compatibility)
- Putty (for Windows machines)
- For HP moonshot chassis hardware requirements please consult www.hp.com/go/moonshot

Preparing the Environment

Hardware Setup

Connect your system as per the diagram below.



- CM: Chassis Manager
- A: Switch A in the moonshot chassis
- B: Switch B in the moonshot chassis
- eth0: uplink port on switch A
- eth1: uplink port on switch B

Your laptop needs to connect to the private vlan in order to access the Juju and MAAS GUIs after installation is done. It also needs to connect to the outside world, i.e. the internet, and that is done through the dev/corporate network. The connection to the Chassis Manager

(CM) also happens through the corporate/dev network, hence both your laptop and the CM should be connected to it.

Connecting to the Chassis Manager

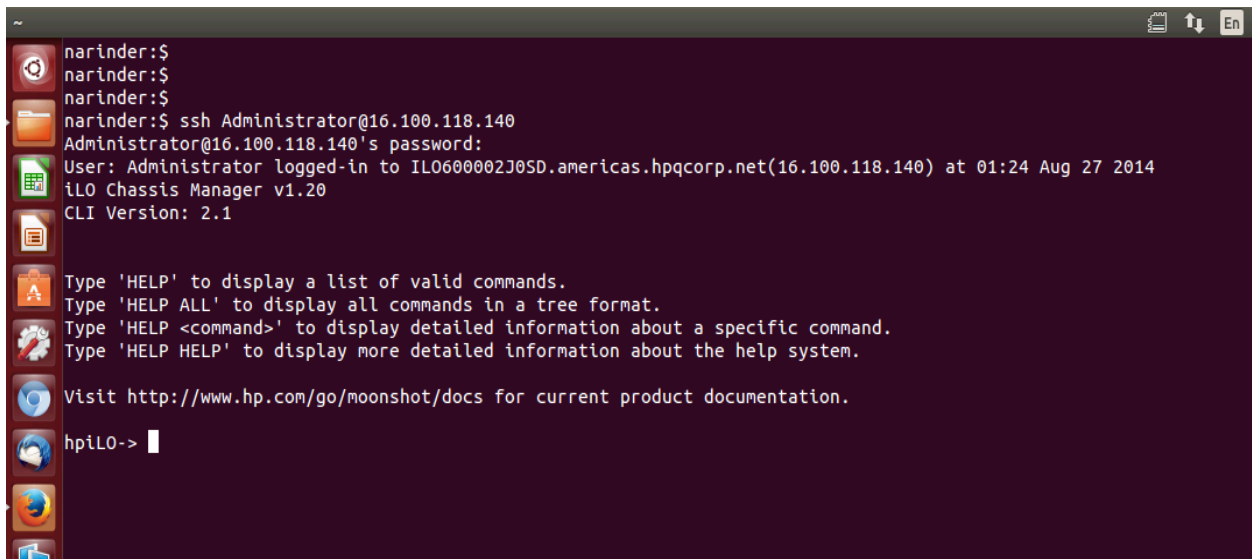
If you are using Ubuntu: open a terminal using Ctrl+Alt+t. You will see a blank terminal screen with a prompt like "user:\$". If you are using Windows you will need to install and run the Putty utility first before you can proceed.

For the purposes of this example, we will assume that the IP of the Chassis Manager is 16.100.118.140. Please ask your lab administrator for your IP address.

The default username is 'Administrator' and the password is 'password'. In your terminal window, type the following command then press 'Enter'.

```
ssh Administrator@16.100.118.140
```

When asked for the password, type 'password' then press 'Enter'. You are now connected to the CM and should see the screen below:

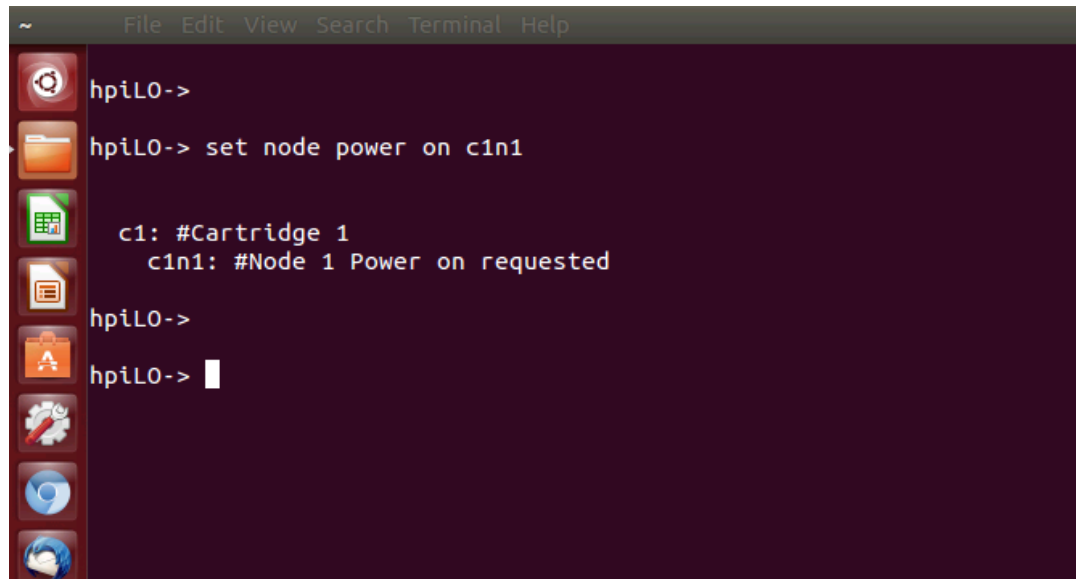
A screenshot of a terminal window. The prompt is 'narinder:\$'. The user enters 'ssh Administrator@16.100.118.140'. The terminal shows the connection process: 'Administrator@16.100.118.140's password:', 'User: Administrator logged-in to IL0600002J05D.americas.hpqcorp.net(16.100.118.140) at 01:24 Aug 27 2014', 'iLO Chassis Manager v1.20', and 'CLI Version: 2.1'. Below this, there is a help menu with instructions for 'HELP', 'HELP ALL', and 'HELP <command>'. At the bottom, there is a URL: 'Visit http://www.hp.com/go/moonshot/docs for current product documentation.' The prompt is now 'hpiLO->'.

Powering on the first node

This step will power on your first node (aka 'Node 1') in your Moonshot chassis. This is where MAAS and Juju will be installed. As mentioned above, please make sure this node is already running Ubuntu Server 14.04 LTS. If you're not sure, please consult your Moonshot systems administrator.

To power on Node 1, type the following command on the CM login screen at the hpiLO-> prompt:

```
set node power on c1n1
```

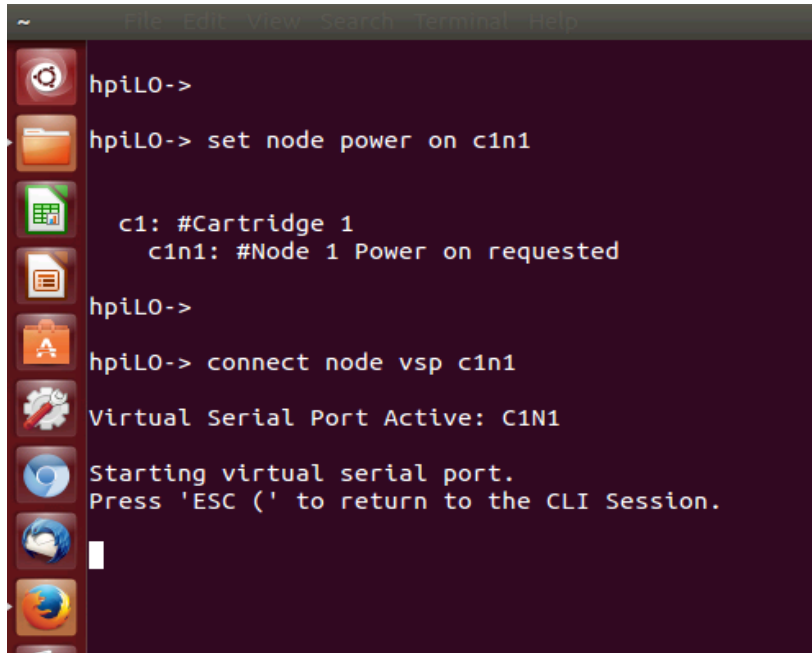


Connecting to the node's virtual serial port and logging into Ubuntu

This step will allow you to login to the server console on Node 1, so you can start running commands. 'c1n1' means cartridge 1 node 1. Type the following command into your terminal window as shown below:

```
connect node vsp c1n1
```

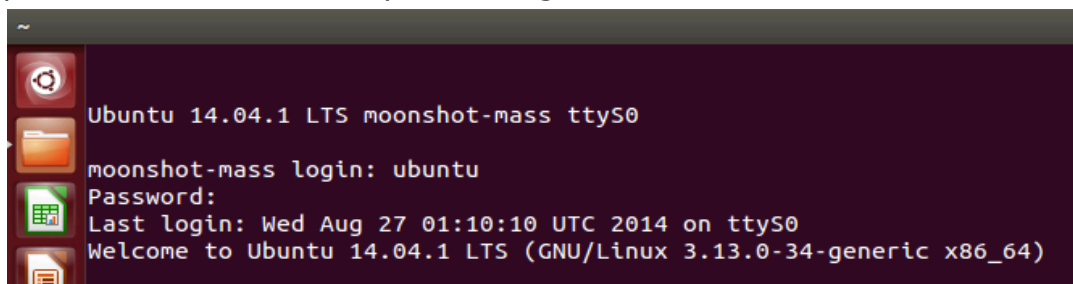
After a short wait, you will be presented with a login prompt.



```
hpiLO->
hpiLO-> set node power on c1n1
c1: #Cartridge 1
c1n1: #Node 1 Power on requested
hpiLO->
hpiLO-> connect node vsp c1n1
Virtual Serial Port Active: C1N1
Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.
```

You are now ready to login to the Ubuntu Server 14.04 OS to start the installation of MAAS and Juju. These tools will enable you to control your Moonshot chassis and deploy your workload.

At the login prompt "moonshot-mass login:", type 'ubuntu' and when prompted for a password, enter 'ubuntu', as per the image below:



```
Ubuntu 14.04.1 LTS moonshot-mass ttyS0
moonshot-mass login: ubuntu
Password:
Last login: Wed Aug 27 01:10:10 UTC 2014 on ttyS0
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-34-generic x86_64)
```

Optional: Custom Network Configuration

This section only applies if you wish to modify your network configuration from the preconfigured default. If you don't need to modify it or you are not sure what this means, please skip to the next section.

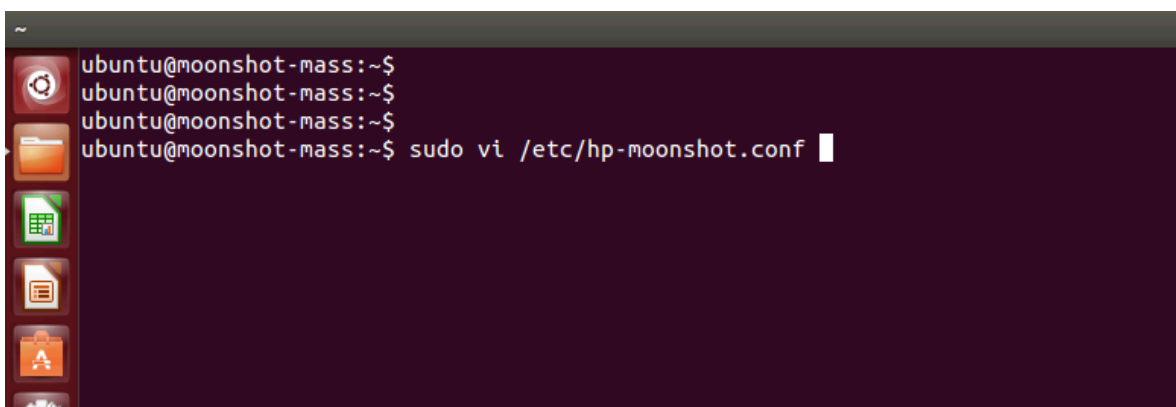
This section will guide you through modifying the network configuration for your private vlan. You will need to do this if you wish to run this infrastructure on a different network than the preconfigured 10.14.

To change the network configuration you will need to modify the `/etc/hp-moonshot.conf` file. The default value for the MAAS Network on your vlan is `10:14:0:0/16`. The network address (`/16`) cannot be changed, so all you can change are the first two values, i.e. `"10:14"`. This is the only modification you can make to this file.

You can use any editor to modify the file. The example below assumes you are using `vi` under Ubuntu. If you are using a different editor, please refer to that editor's documentation if needed.

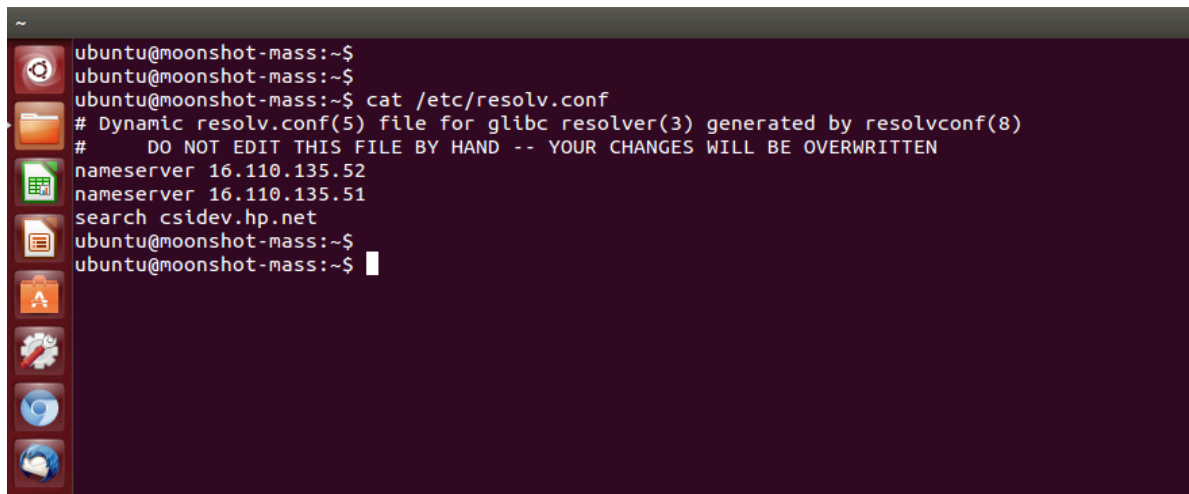
At the prompt, type the following command to start `vi` and open the file:

```
sudo vi /etc/hp-moonshot.conf
```



You will see a screen similar to the one below. Let's assume you want to change the value from `10:14` to `10:10`. Use the arrow keys to move your cursor to the number `'4'` on the first line. Press `"r"` then `"0"` to replace the number `'4'` with the number `'0'`.

contains the correct DNS server address. If the DNS server information is incorrect, MAAS will not function properly.

A terminal window screenshot showing the command 'cat /etc/resolv.conf' and its output. The output lists two nameservers (16.110.135.52 and 16.110.135.51) and a search domain (csidev.hp.net). The terminal prompt is 'ubuntu@moonshot-mass:~\$' and the window title is '~'.

```
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$ cat /etc/resolv.conf  
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)  
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN  
nameserver 16.110.135.52  
nameserver 16.110.135.51  
search csidev.hp.net  
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$
```

APT Proxy Setup

This section is only relevant if your dev/corporate network requires a proxy server to reach out to the external network. If a proxy is not required please skip this section.

Ubuntu downloads packages automatically via APT. If a proxy is required in your network, then configuring APT's proxy information will ensure proper functionality and security updates. To do this, run the below commands at the command-line prompt. Please make sure you replace the appropriate value of proxy server and port as per your environment. In case you do not have https/ftp proxy then no need to run the respective command.

```
echo 'Acquire::http::proxy "http://proxy.example.com:8080"' \  
    | sudo tee -a /etc/apt/apt.conf > /dev/null  
echo 'Acquire::https::proxy "http://proxy.example.com:8080"' \  
    | sudo tee -a /etc/apt/apt.conf > /dev/null  
echo 'Acquire::ftp::proxy "ftp://proxy.example.com:8080"' \  
    | sudo tee -a /etc/apt/apt.conf > /dev/null
```

You can confirm with `cat` the values written to the file:

```
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$ cat /etc/apt/apt.conf  
Acquire::http::proxy "http://proxy.compaq.com:8080/";  
  
Acquire::https::proxy "http://proxy.compaq.com:8080/";  
  
Acquire::ftp::proxy "ftp://proxy.compaq.com:8080/";  
  
Acquire::ftp::proxy "ftp://proxy.compaq.com:8080/";  
ubuntu@moonshot-mass:~$
```

Installing and configuring MAAS and Juju

Now you are ready to install and configure MAAS and Juju. At the shell prompt, run the following command:

```
sudo hp-moonshot-maas-install
```

It will take approximately 30 minutes to run, after which the bootstrap node will be ready and will display the IP address of the Juju GUI.

```
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$  
ubuntu@moonshot-mass:~$ sudo hp-moonshot-maas-install
```

Installation complete

Once installation is complete you should see a screen similar to the one below, displaying the name of the machine which contains the Juju GUI (in this case 'moonshot.maas' and its IP address (in this case 10.10.100.2). Please make a note of this machine name and IP address as you will need them later.

```
ubuntu@moonshot-maas: ~
Running apt-get update
Running apt-get upgrade
Installing package: git
Installing package: curl
Installing package: cpu-checker
Installing package: bridge-utils
Installing package: rsyslog-gnutls
Fetching tools: curl -sSfw 'tools from %{url_effective} downloaded: HTTP %{http_
code}; time %{time_total}s; size %{size_download} bytes; speed %{speed_download}
bytes/s' --retry 10 -o $bin/tools.tar.gz 'http://10.10.4.1/MAAS/api/1.0/files/
?key=b7979008-3093-11e4-acff-f0921cb486f5&op=get_by_key'
Bootstrapping Juju machine agent
Starting Juju machine agent (jujud-machine-0)
2014-08-30 22:28:08 INFO juju.cmd supercommand.go:329 command finished
+ juju set-constraints tags=
+ maas admin nodes accept-all
[]
+ juju deploy --to 0 cs:~hazmat/trusty/juju-gui-2 juju-gui
Added charm "cs:~hazmat/trusty/juju-gui-2" to the environment.
+ juju expose juju-gui
+ bzip branch lp:~hp-moonshot/charms/bundles/web-in-a-box/bundle /home/ubuntu/web
-in-a-box
You have not informed bzip of your Launchpad ID, and you must do this to
write to Launchpad or access private data. See "bzip help launchpad-login".
bzip: ERROR: Already a branch: "/home/ubuntu/web-in-a-box".
+ nslookup moonshotvm.maas
+ + grep Address:
cut -d -f2
+ juju_address=10.10.100.2
+ echo JUJU GUI details: name: moonshotvm.maas IP Address: 10.10.100.2
JUJU GUI details: name: moonshotvm.maas IP Address: 10.10.100.2
ubuntu@moonshot-maas:~$
```

Accessing MAAS

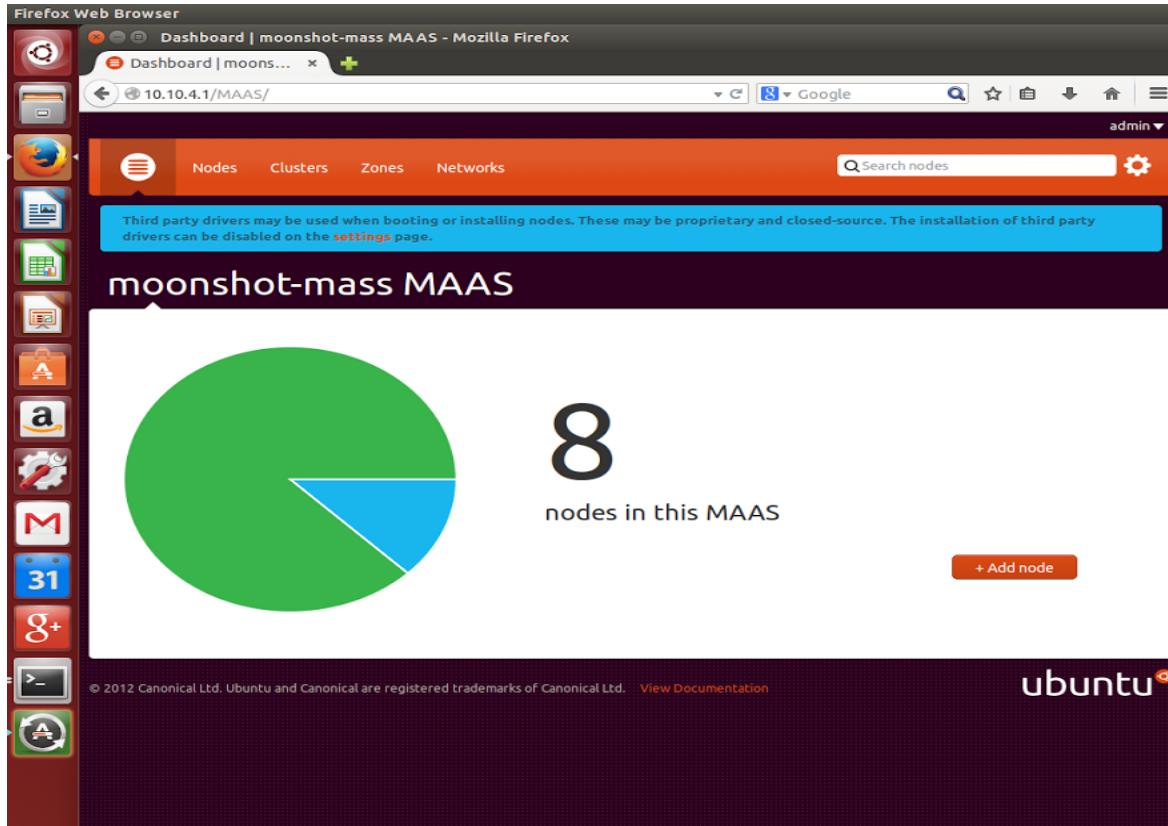
Open another window or tab in your browser, and enter the following into the address bar:

<http://10.10.4.1>

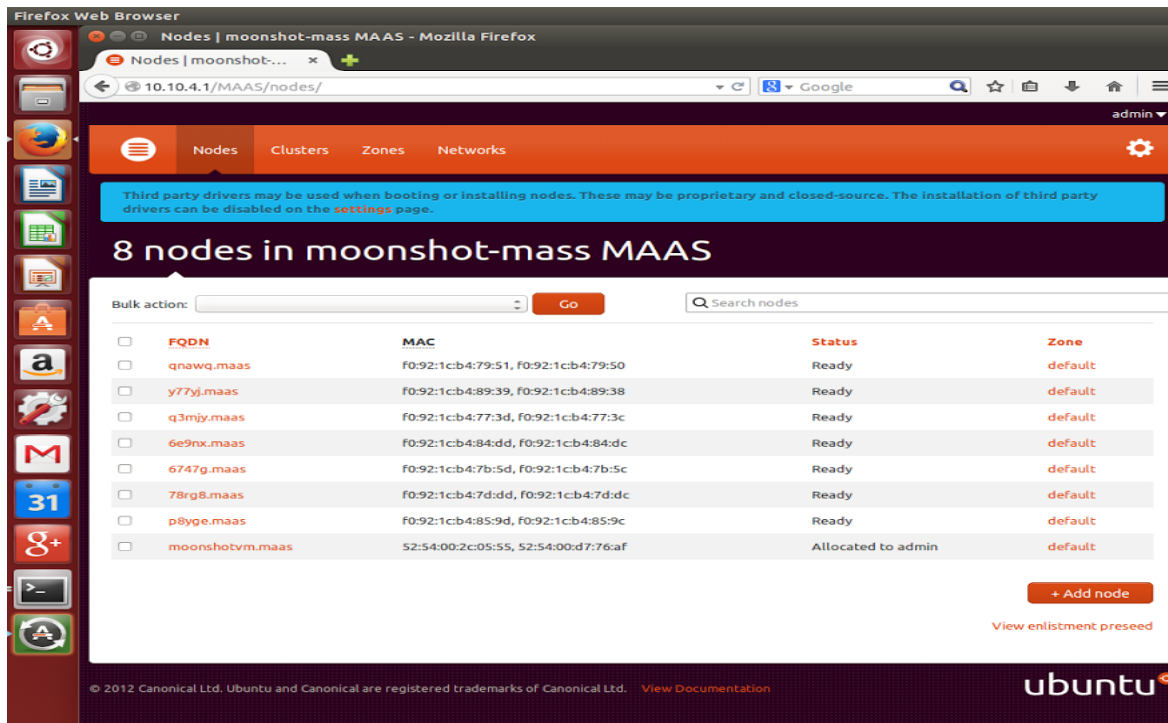
In the login box, enter username: 'admin' and password: 'admin'.



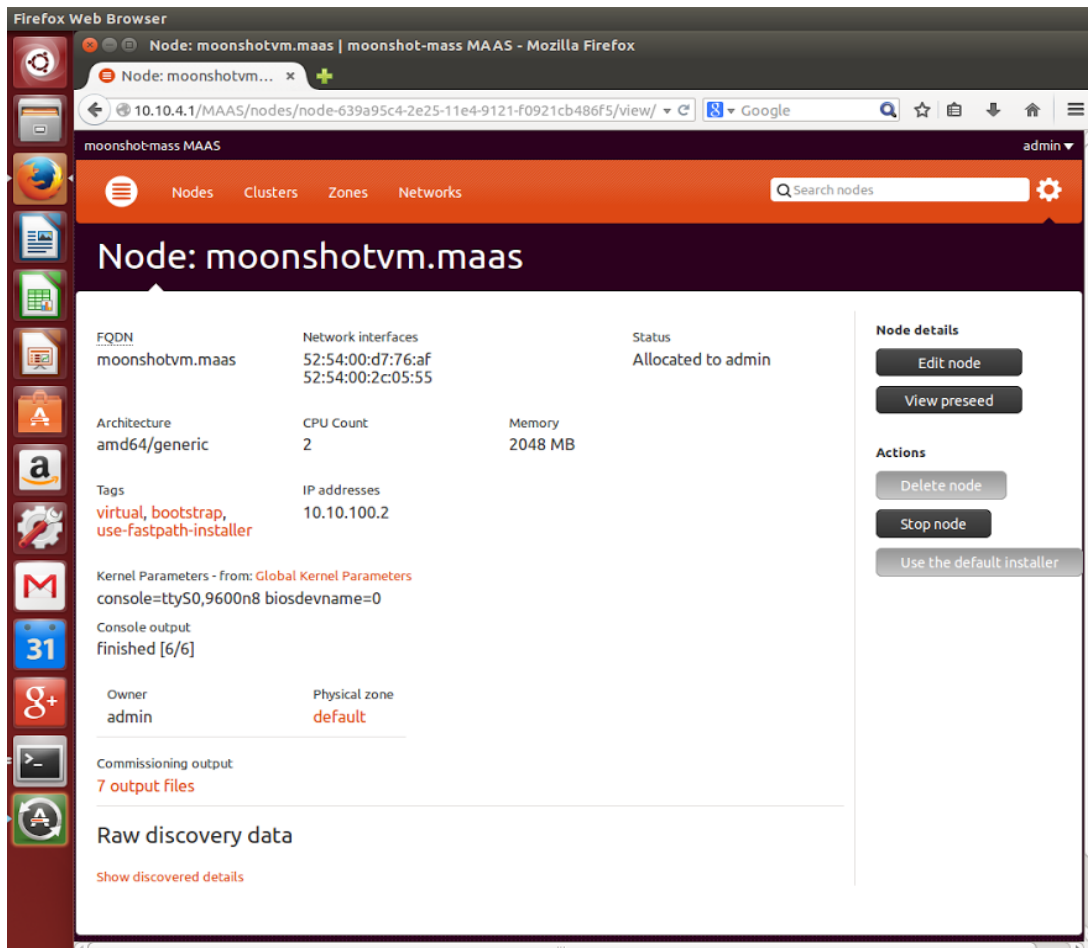
You can now see the MAAS UI. In this particular scenario, the Moonshot chassis has 9 cartridges and first node is taken by MAAS. This leaves 8 nodes available to deploy workloads on.



You can explore the information MAAS gives you by clicking on the pie chart. This will take you to a list of all machines in your MAAS and their respective information, as seen below:



If you click on the last machine in the list 'moonshotvm.maas', this will take you to another screen with more in-depth information about that particular machine, as seen below:



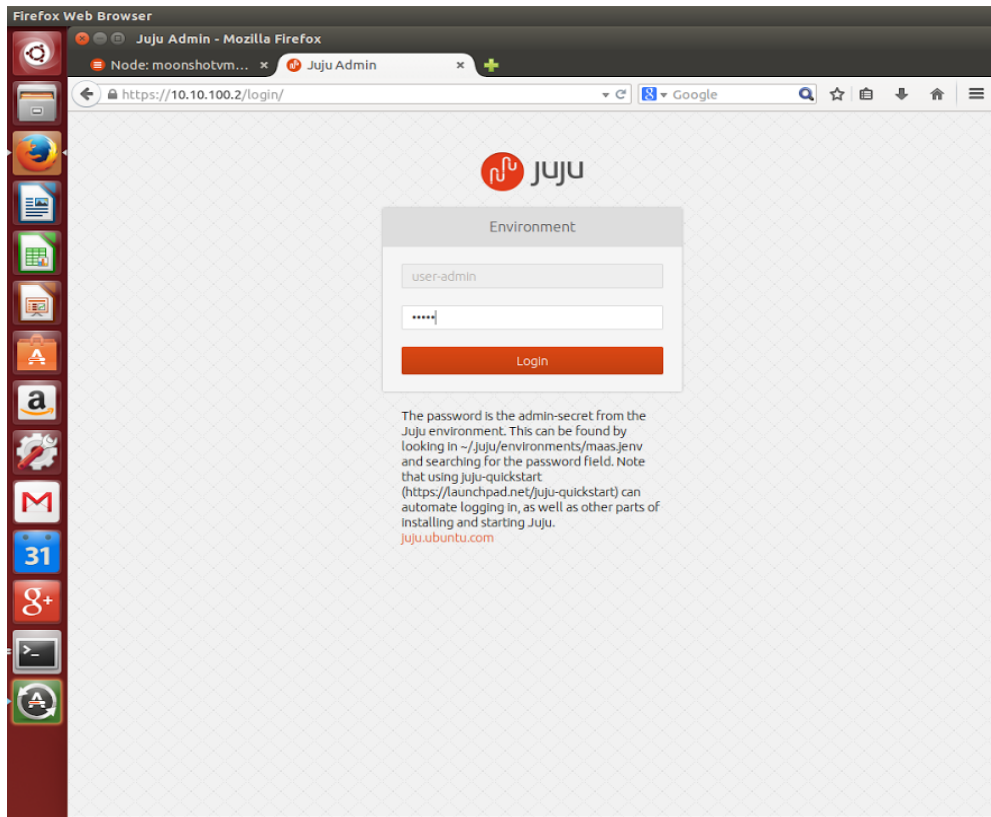
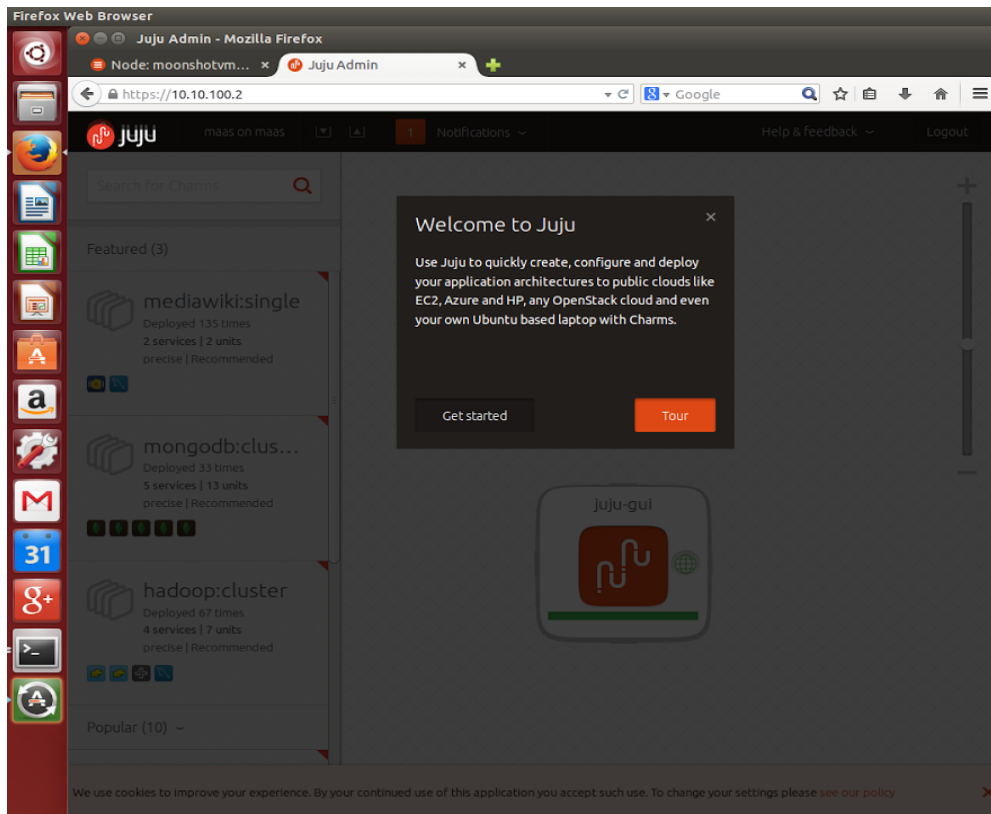
Don't worry about understanding what any of this means. This is just a prerequisite for deploying your workloads later on.

Accessing the Juju GUI

Recall from the previous step that you have noted down the machine name and IP address where the Juju GUI is running. You will need those now.

Point your web browser to IP address of the Juju GUI machine (in this case: 10.10.100.2

Login into the Juju GUI with username: 'admin' and password: 'admin'.



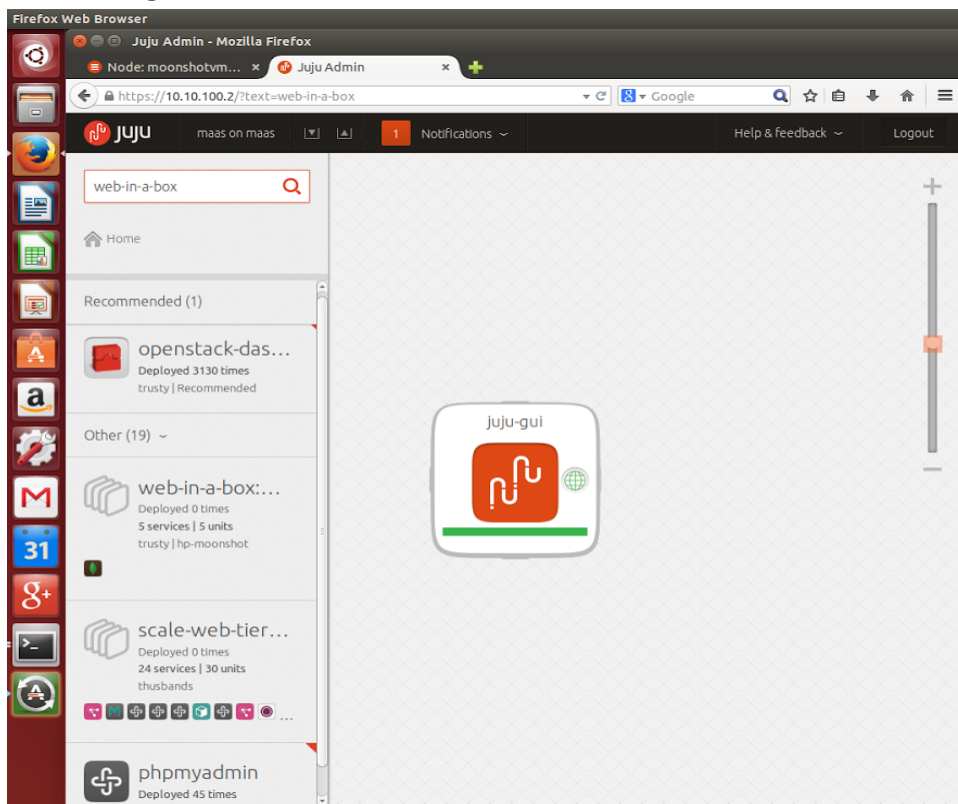
You are now ready to start deploying your workloads!

Deploying the “Web-infrastructure-in-a-Box” bundle

A Juju ‘bundle’ is a ready-made collection of charms, each representing a service you want to run in your environment. The Juju GUI makes it incredibly easy to ‘import’ an existing bundle and just deploy it via drag and drop. This is the equivalent of hundreds of shell commands and lots of tedious configuration steps (with lots of room for errors!). Juju takes the pain and risk out of this process and makes it quick, easy, and enjoyable, as you will see below.

First, you need to locate the bundle you wish to deploy. Our bundle is called ‘Web-infrastructure-in-a-Box’. The search box at the top left of the GUI allows you to locate any bundle by simply searching for it.

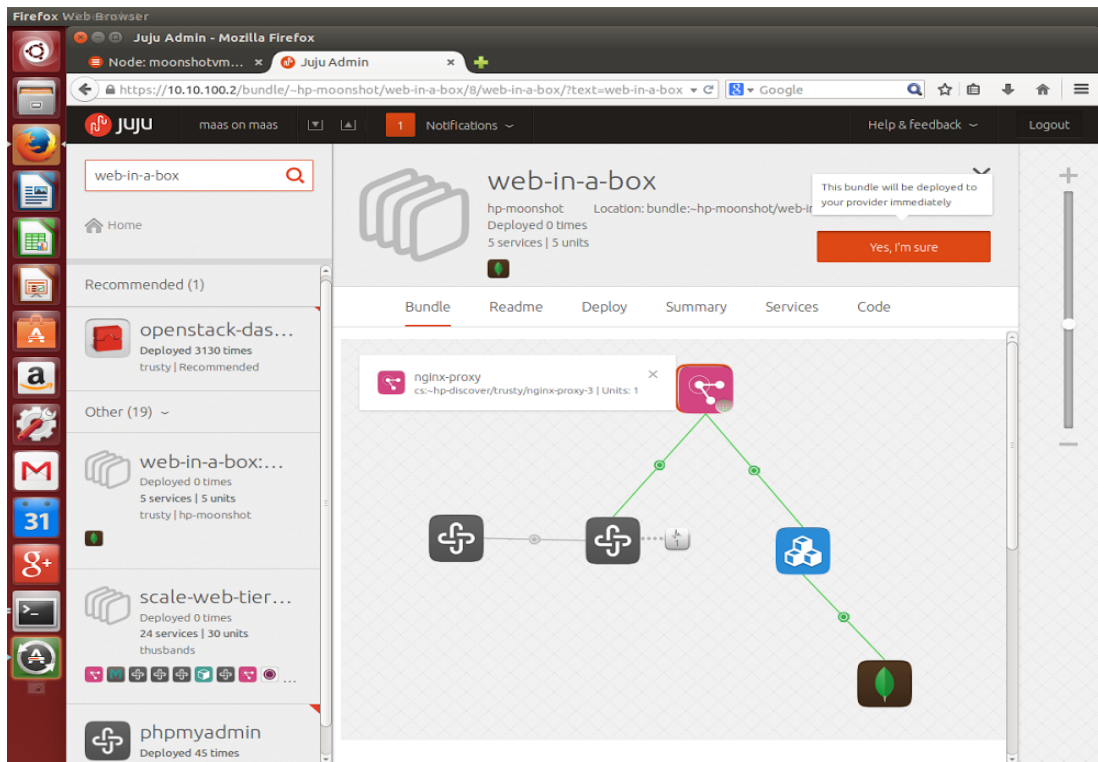
Type the following into the search box: Web-infrastructure-in-a-Box .



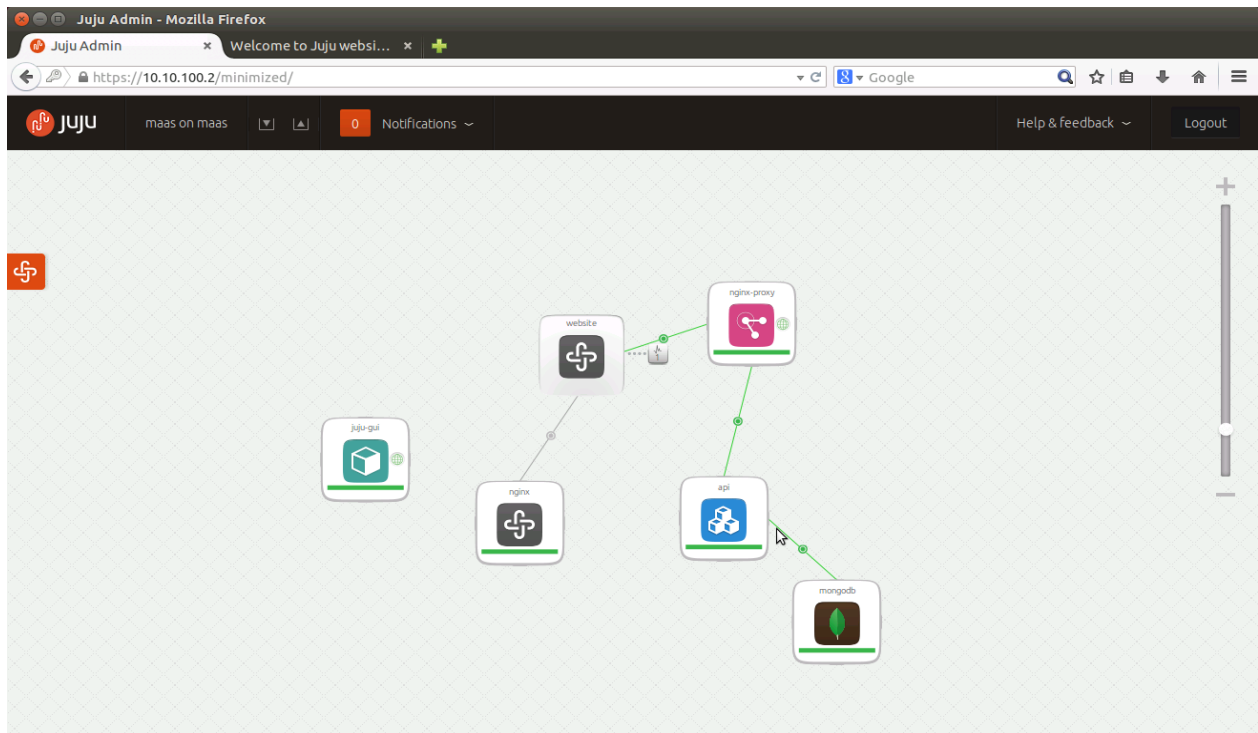
Deploying the bundle

You will see the Web-infrastructure-in-a-Box bundle in the left bar on your screen. Now you need to deploy it onto your canvas. There are a number of ways to do this. Click on it and you will see a larger pop-up box with more information about the bundle, as shown below. Click on the orange ‘deploy this bundle’ button at the top right of the box to start the deployment process.

Click 'yes, I'm sure' to confirm your choice. This will deploy the bundle to your canvas.



As an alternative, you can just drag and drop the bundle from the left bar to your canvas. This will immediately deploy the bundle without asking you for confirmation.



Appendix A

If you are interested in learning what's happening behind the scenes, install script will automated the entire installation and configuration process for you. It will run through the following steps:

- apt-get update (update your Ubuntu packages to make sure you have the latest release and versions)
- apt-get upgrade -y
- apt-get install hp-moonshot (install the Moonshot-specific packages)
 - Run preinstall script and install the remaining packages which include MAAS and JUJU packages as well.
 - bzip
 - charm-tools
 - git
 - iptables-persistent
 - libvirt-bin
 - openssh-server
 - qemu-kvm

- run-one
- squid3
- vim
- virtinst
- virt-manager
- ipmitool
- hp-moonshot-maas-install
- juju-core,
- juju-deployer,
- maas,
- maas-cluster-controller,
- nmap,
- remmina
- Run sudo hp-moonshot-maas-init as post install
 - configure and test network
 - symlink cached images
 - creates a maas user (admin/admin)
 - generate ssh key for ubuntu user, loads it into maas
 - setup MAAS environment.
 - Create and add virtual machine for bootstrap node named moonshotvm.maas into MAAS
 - enlist and have all physical machine in the moonshot chassis ready for deployment except first node
 - Setup JUJU environment for user ubuntu on MAAS server itself
- bootstrap juju node for virtual machine
- install and configure juju-gui
- Display the juju-gui node details

Appendix B

Juju command line

For more advanced users who prefer to use the command line, you can also run all Juju commands from your shell prompt, including importing and deploying bundles. Run the following set of commands from your terminal to download the bundle and deploy it

```
$ juju status juju-gui
environment: maas
```

```
machines:
"0":
  agent-state: started
  agent-version: 1.20.5.1
  dns-name: moonshotvm.maas
  instance-id: /MAAS/api/1.0/nodes/node-c5bd1986-2ff4-11e4-acff-f0921cb486f5/
  series: trusty
  state-server-member-status: has-vote
services:
juju-gui:
  charm: cs:~hazmat/trusty/juju-gui-2
  exposed: true
  units:
juju-gui/0:
  agent-state: started
  agent-version: 1.20.5.1
  machine: "0"
  open-ports:
  - 80/tcp
  - 443/tcp
  public-address: moonshotvm.maas
ubuntu@moonshot-mass:~$
```

```
$. /etc/environment
$ juju-deployer -e maas -d web-infrastructure-in-a-box -c
web-infrastructure-in-a-box/bundles.yaml
```

After around 10-12 minutes, your bundle will be deployed and ready to use.

```
$ juju status
environment: maas
machines:
"0":
  agent-state: started
  agent-version: 1.20.5.1
  dns-name: moonshotvm.maas
  instance-id: /MAAS/api/1.0/nodes/node-c5bd1986-2ff4-11e4-acff-f0921cb486f5/
  series: trusty
  state-server-member-status: has-vote
"1":
  agent-state: started
  agent-version: 1.20.5.1
  dns-name: acwqj.maas
```

```
instance-id: /MAAS/api/1.0/nodes/node-f3a21cf6-2ff5-11e4-acff-f0921cb486f5/  
series: trusty  
"2":  
agent-state: started  
agent-version: 1.20.5.1  
dns-name: bdj4t.maas  
instance-id: /MAAS/api/1.0/nodes/node-f72e2fd6-2ff5-11e4-8e0d-f0921cb486f5/  
series: trusty  
"3":  
agent-state: started  
agent-version: 1.20.5.1  
dns-name: mbf6y.maas  
instance-id: /MAAS/api/1.0/nodes/node-f51329e0-2ff5-11e4-acff-f0921cb486f5/  
series: trusty  
"4":  
agent-state: started  
agent-version: 1.20.5.1  
dns-name: fg6wq.maas  
instance-id: /MAAS/api/1.0/nodes/node-f72f9268-2ff5-11e4-acff-f0921cb486f5/  
series: trusty  
services:  
api:  
charm: cs:~hp-discover/trusty/node-app-1  
exposed: false  
relations:  
mongodb:  
- mongodb  
website:  
- nginx-proxy  
units:  
api/0:  
agent-state: started  
agent-version: 1.20.5.1  
machine: "1"  
open-ports:  
- 8000/tcp  
public-address: acwqj.maas  
juju-gui:  
charm: cs:~hazmat/trusty/juju-gui-2  
exposed: true  
units:  
juju-gui/0:  
agent-state: started
```



```
agent-version: 1.20.5.1
machine: "0"
open-ports:
- 80/tcp
- 443/tcp
public-address: moonshotvm.maas
mongodb:
charm: cs:trusty/mongodb-3
exposed: false
relations:
database:
- api
replica-set:
- mongodb
units:
mongodb/0:
agent-state: started
agent-version: 1.20.5.1
machine: "2"
open-ports:
- 27017/tcp
- 27019/tcp
- 27021/tcp
- 28017/tcp
public-address: bdj4t.maas
nginx:
charm: cs:~hp-discover/trusty/nginx-4
exposed: false
relations:
web-engine:
- website
units:
nginx/0:
agent-state: started
agent-version: 1.20.5.1
machine: "3"
public-address: mbf6y.maas
subordinates:
website/0:
upgrading-from: cs:~hp-discover/trusty/website-1
agent-state: started
agent-version: 1.20.5.1
public-address: mbf6y.maas
```

```
nginx-proxy:
  charm: cs:~hp-discover/trusty/nginx-proxy-3
  exposed: true
  relations:
  website:
  - api
  - website
  units:
  nginx-proxy/0:
  agent-state: started
  agent-version: 1.20.5.1
  machine: "4"
  open-ports:
  - 80/tcp
  public-address: fg6wq.maas
  website:
  charm: cs:~hp-discover/trusty/website-1
  exposed: false
  relations:
  nginx-engine:
  - nginx
  website:
  - nginx-proxy
  subordinate-to:
  - nginx
ubuntu@moonshot-mass:~$
```

References

For more information or documentation on Ubuntu, MAAS, Juju, etc. please visit:

www.ubuntu.com

juju.ubuntu.com

maas.ubuntu.com