**Openshift Installation On IBM Cloud Infrastructure**

This document will help you to install Openshift on a VMWare based Infrastructure, It provides detailed steps you need to follow to access IBM Cloud, to provision the needed Infrastructure, and to install Openshift 3.11 on top of it.

This document is divided into three main sections:

1- How to access IBM Cloud using VPN.

2- How to provision ESXi Host. and how to use it to create the required VMs.

3- How to install Openshift 3.11

3.1- Preparing the hosts (CentOS7 VMs will be used here)

3.2- Installing Openshift using Ansible Playbook

3.3- Troubleshooting

**Section 01- IBM Cloud Access**

When you provision an ESXi host on top of IBM Cloud, In order to access it, you will need to setup a VPN.

You will need to establish a VPN connection to IBM Cloud private network via SSL, or IPSec.

**Pre-Req:**

Identify which Endpoint you will use, Which DC you will provision your BM on. [Here is a full list of end points](https://cloud.ibm.com/docs/iaas-vpn?topic=VPN-available-vpn-endpoints)

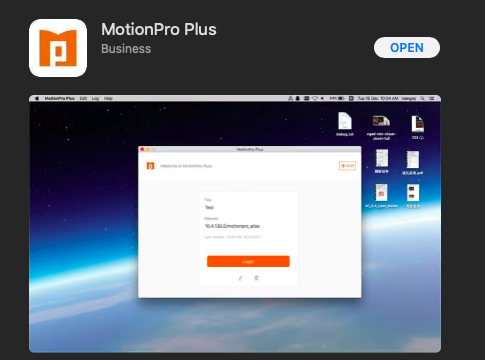
The environment I will use in here will be based on Frankfurt , which have endpoint vpn.fra02.softlayer.com

Based on your local machine, install your VPN Client

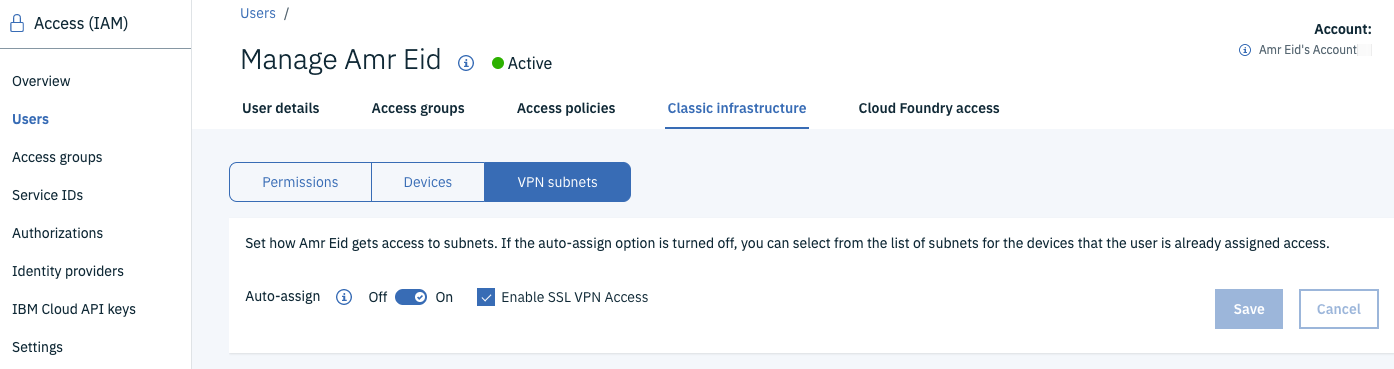
[For Windows Users](https://cloud.ibm.com/docs/infrastructure/iaas-vpn?topic=VPN-connect-ssl-vpn-windows7)

[For Mac Users](https://cloud.ibm.com/docs/infrastructure/iaas-vpn?topic=VPN-connect-ssl-vpn-mac-osx)

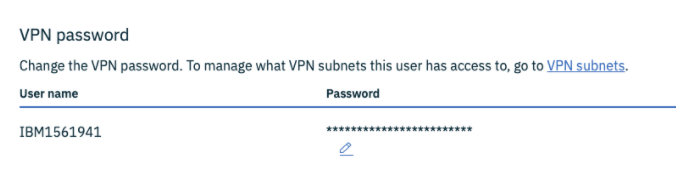
From a Mac machine, install MotionPro Plus using App Store



On IBM Cloud Portal, Enable SSL VPN Access for the user and manage its password

****

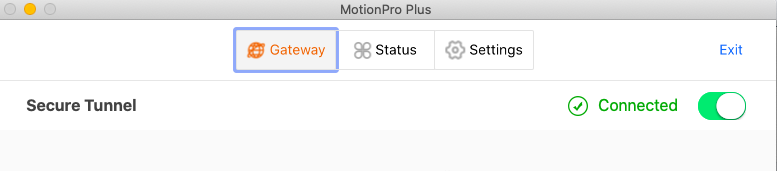
Preferably change the password. If not you need to capture the user/password from IAM/Users/User Page

****

Get the Client up and running

You will need to feed MotionPro with the endpoint, In our case “vpn.fra02.softlayer.com”

You will also need to provide it with the user/password after you hit login.

****

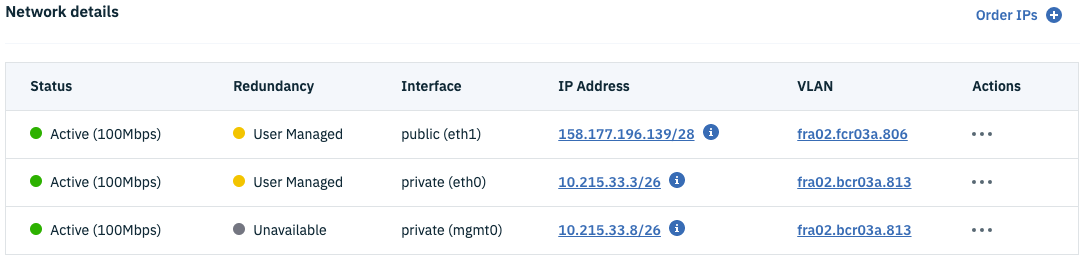
**Section 02- ESXi Host Provisioning**

Provisioning the Bare Metal usually takes from 8 to 24 hours, so plan ahead.

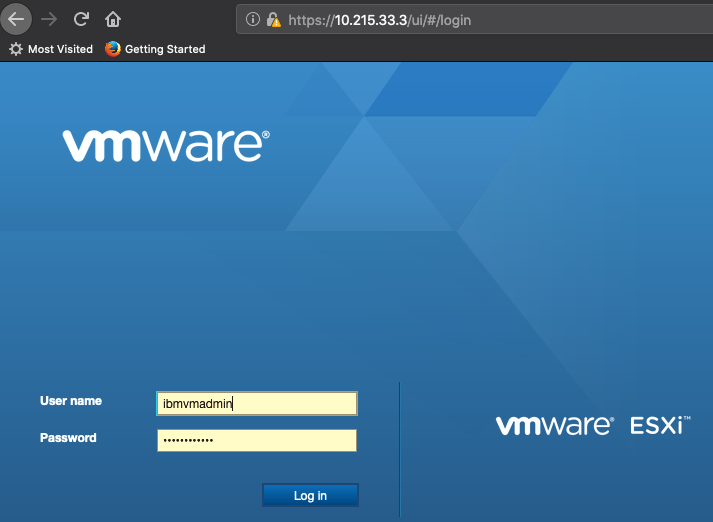
Form the portal, Pick the vmWare BM version you like.



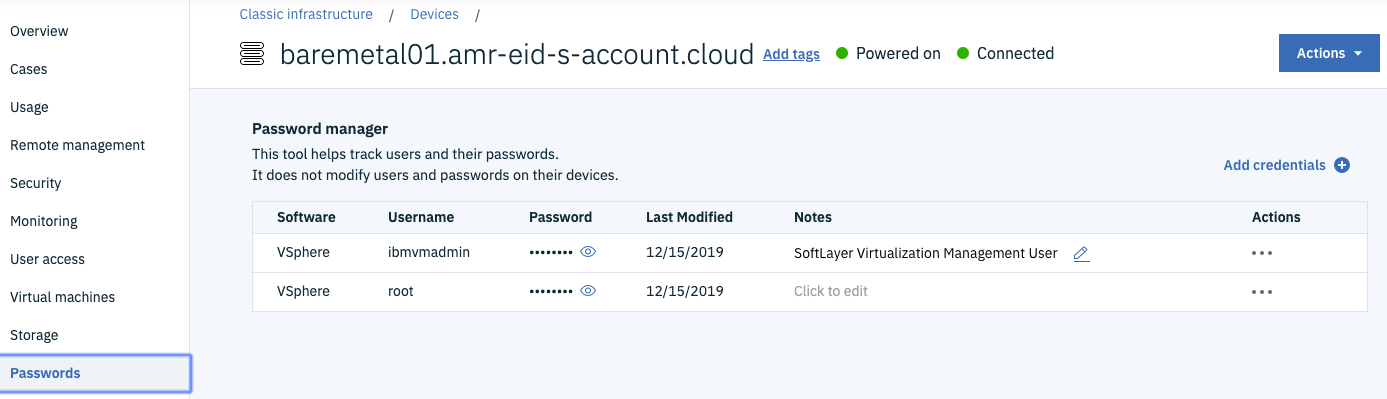
Check the Subnets / IP Addresses attached to the BM, Pick the private IP ( usually attached to eth0), the other private IP usually is used for the management and could be accessed through KVM.



From any browser type the private IP and hit enter, then accept the default certificates you may face. You should receive the login page like below.



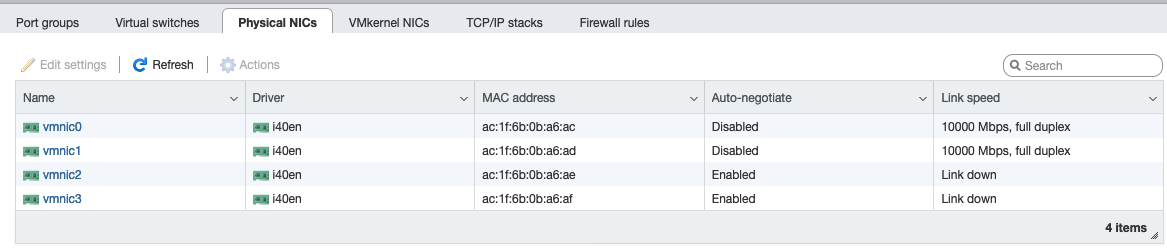
From your BM page on IBM Cloud portal, navigate to Passwords to get the vSphere username and password. (ibmvmadmin is the ESXi user).



**Configuring ESXi**

In the ESXi portal , You need to know

* Physical NICs on the BM are reflected on the ESXi physical NICs view.
* You should have validate that you do have an up and running NICs like the below screenshot.

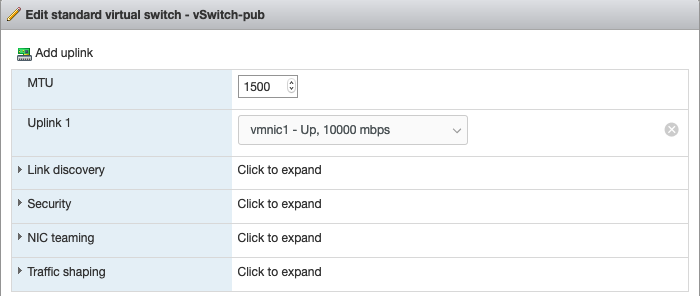


Here, There are two NICs down and the other two are up and running.

From the mapping you can easily detect that NIC0 is the private NIC and NIC1 is the Public.

In order to proceed , you just need to do the below actions

* Create Port Groups ( one for the private and one for the public.)
* link each vSwitch with its related Physical NIC.



Below screenshot shows how the public looks, attached to vmnic1 which is the public interface.

Apply the same for the private and make sure it is connected to vmnic0

VMKernel NICs will have one for private connection, you will need to create one more.



Run esxcfg-route -a 10.0.0.0/8 [your servers private gateway ip]

[root@baremetal01:~] esxcfg-route -a 10.0.0.0/8 10.194.251.65

Run esxcfg-route [your servers public gateway] to ensure that the public gateway is the default

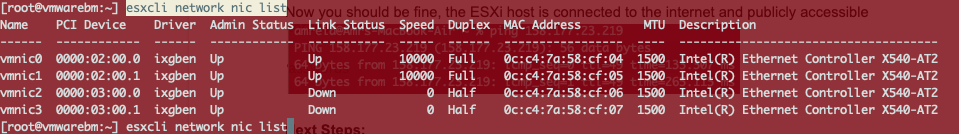
[root@baremetal01:~] esxcfg-route 158.177.23.209

VMkernel default gateway set to 158.177.23.209

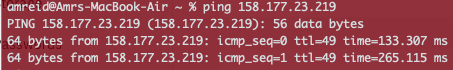
**Troubleshooting:**

esxcli network nic list

Make sure you have the right physical NICs , if your machine is not ordered dual, one public and one private are up. So typically you should find 4 entries, 2 of them are up. Like below

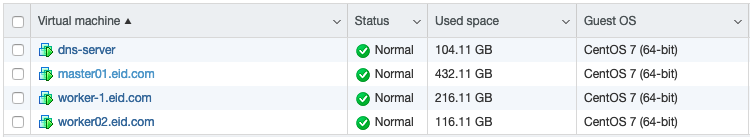


Now you got all set in the ESXi host, the ESXi host is connected to the internet and publicly accessible



**Provisioning the environment**

The list appears in the screenshot below is what we need to achieve, I will explain how to create the first VM only.

****

As a **prerequisite**,, we need to get a CentOS image to have it on the datastore.

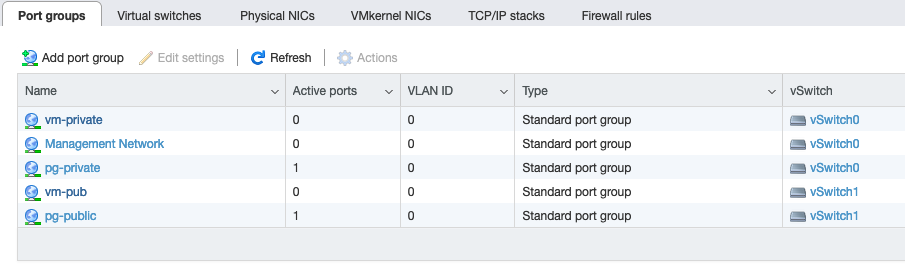
cd vmfs/volumes/datastore1/iso/

wget <http://mirror.fra10.de.leaseweb.net/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1908.iso>

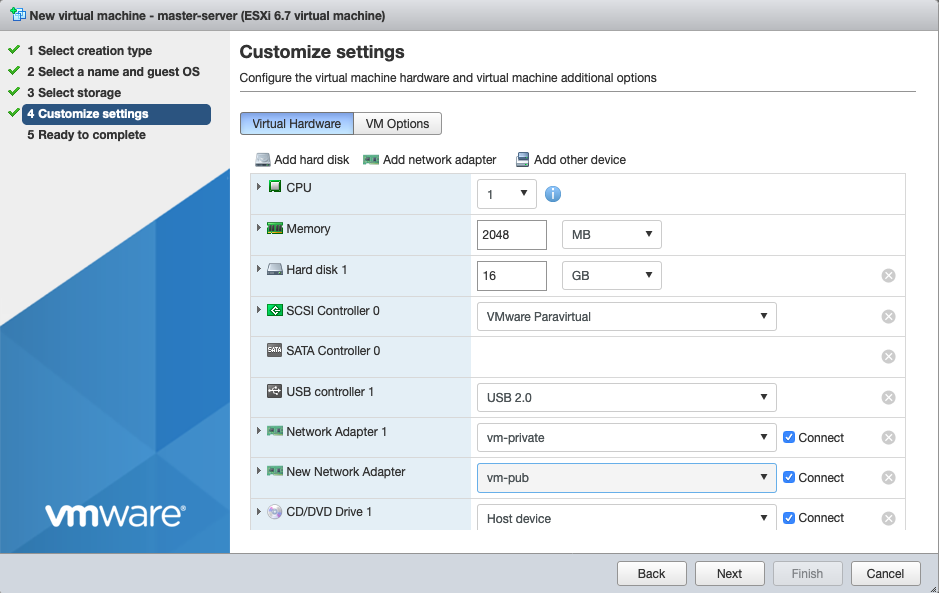
**Secondly**: Let’s us create a VM

First we need to define if the VM will have private / public network adapters or both.

So, we need to add the related port-groups like below



Then , go ahead and create the VM using the wizard, default setting till you reach “Customize Settings”, Add Network Adapter , In the network adapter pick the related port-group for each adapter.



Following the same wizard, From the above step, From CD/DVD pick the proper iso for your OS.

Complete the steps and power the VM on.

You can either configure the network/DNS before starting the real installation using CentOS setup wizard. For the Network, you will only be asked to provide an IP/Netmask and Gateway.

Or

After installing the OS, you can manually configure the network by accessing the VM from ESXi host, and using the CLi you can configure the network with a tool like nmtui

The below is to validate your network settings

* You can try to ping a domain like [www.ibm.com](http://www.google.com)
* If it worked, then you got all set. If not, look to DNS setup
* For now, you an use any public DNS like 8.8.8.8
* Try cat /etc/resolv.conf , if it is blank, you need to set up a DNS resolver. I would suggest editing the resolv.conf with nameserver 8.8.8.8
* Try ping [www.ibm.com](http://www.google.com) or yum update again.

One final step, as you will have to install many VMs, it is good to change the hostname for every VM to be descriptive, for CentOS, you can do it from ESXi installation wizard and you can do it later through

hostnamectl set-hostname "eidns.amreid.com"

exec bash

**Section 03- Installing Openshift**

**03-01 Host Preparation -** DNS Node Setup

**DNSMASQ: H**ow to install and configure

**Install**

yum install dnsmasq bind-utils

**Configure**

Edit those two files, and add the below line to both of them

/etc/resolv.conf and /etc/resolv.dnsmasq

nameserver 127.0.0.1

You will need to edit those two files 1- /etc/dnsmasq.conf and 2-/etc/hosts as follows

cp /etc/dnsmasq.conf /etc/dnsmasq.conf.orig

vi /etc/dnsmasq.conf

At the end of the file you can just add

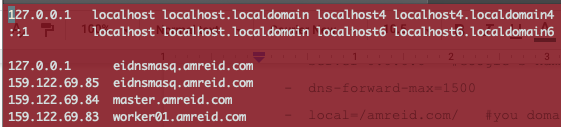
server=8.8.8.8 #Google’s nameserver

local=/amreid.com/ #you domain servers to be looked locally

#Log-queries # In case you need to enable logs

#log-facility=/var/log/dnsmasq.log

Now, edit vi /etc/hosts and add your hosts like below

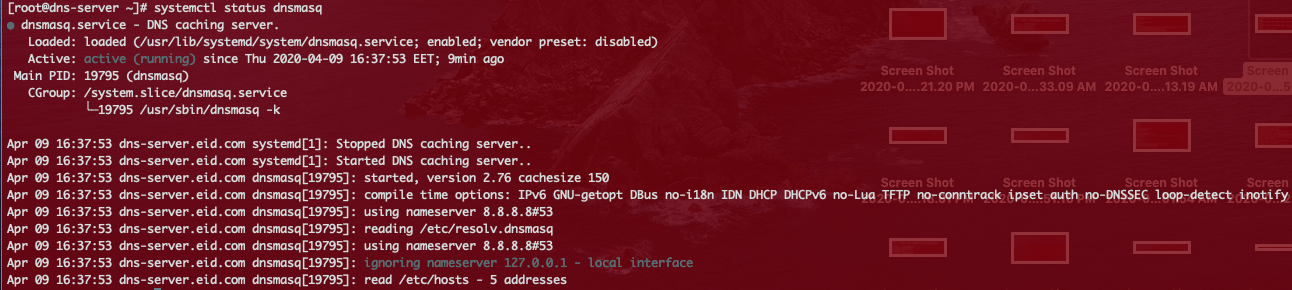


Then, start and enable the dns service.

systemctl start dnsmasq

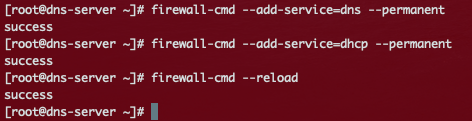
systemctl enable dnsmasq

systemctl status dnsmasq

You should expect results like below saying that it is up and running.

**Firewall**

For now only, to keep the installation easier, let’s either stop the firewall if you don’t need it, Or open DNS and DHCP services in the firewall configuration, to allow requests from hosts on your LAN to pass to the dnsmasq server.

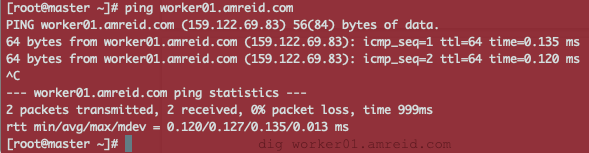


If you like to disable it you can use the below commands

systemctl status firewalld  
service firewalld stop  
systemctl disable firewalld

**Validation**

From worker or Master node, you can validate also by testing if you can reach to each other like



From the DNS server , from each other host, you can check if it can access the others.

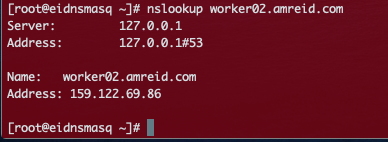
dig worker01.amreid.com

OR # nslookup worker01.amreid.com

Preferably nslookup and make sure to read the results carefully, If it shows this last line message, then it is not going to work properly.



The right execution should looks like



**03-02 Installing Openshift -3.11**

**Host Preparation: Stage 1**

Make sure that all the prerequisites are working fine. DNS is up and running , Masters and Worker nodes can see each other on top of the DNS.

**Actions:**

ssh-keygen on master node

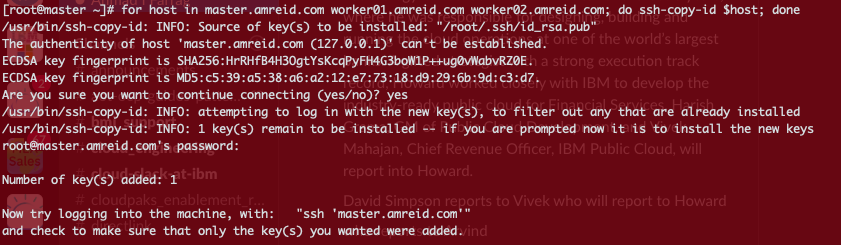
distribute the key all over the nodes. (for host in xxxxxxx; do ssh-copy-id $host done)

On the Master Node:

ssh-keygen

Then

for host in master.amreid.com worker01.amreid.com worker02.amreid.com; do ssh-copy-id $host; done



As a sanity check, you can validate the above step has been successfully by trying to ssh to each one of the worker nodes while being on the master node, you should login without any need to provide a password or a key



**Host Preparation: Stage 2**

**Actions:** Install base packages

On the master node: Install all the below packages

yum install wget git net-tools bind-utils yum-utils iptables-services bridge-utils bash-completion kexec-tools sos psacct

yum install epel-release

yum install ansible pyOpenSSL

cd ~

git clone <https://github.com/openshift/openshift-ansible>

cd openshift-ansible

git checkout release-3.11

**Host Preparation: Stage 3**

**Actions:** Install docker

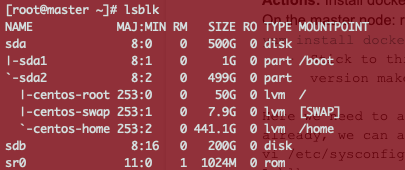
On the master node: run the below commands

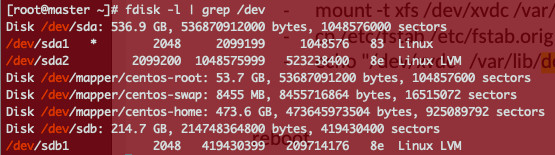
yum install docker-1.13.1

This exact version is advised to be used with openshift3.11.

Here we need to add a dedicated LVM Volume Group , so if not added already, we can add a separate disk to the master VM then

Lsblk or fdisk -l





To get the name of the disk, here it is /dev/sdb

* Edit the file vi /etc/sysconfig/docker-storage-setup

Add

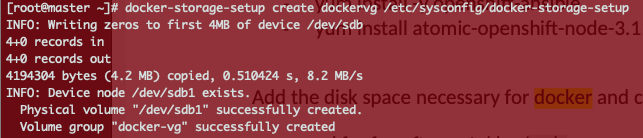
STORAGE\_DRIVER=overlay2

DEVS=/dev/sdb

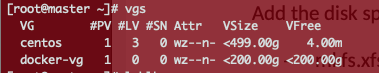
VG=docker-vg

Run the file using the below command

* docker-storage-setup create dockervg /etc/sysconfig/docker-storage-setup



In order to verify this phase, just run vgs to see the created Volume Groups. (20G is enough for the dedicated VG)



**Openshift Installation: Using Ansible Installer**

**Actions:**

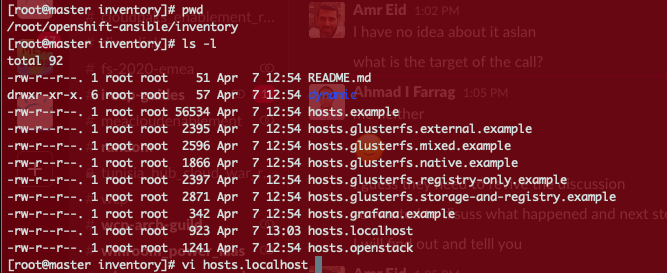
* Installing Openshift using Ansible Playbook
* The initial configuration is in Ansible Inventory file /etc/ansible/hosts (on the Master Node).

**Step 01**

cd ~/openshift-ansible

cp hosts.localhost /etc/ansible/hosts

vi /etc/ansible/hosts



**Step 02**

Edit your ansible hosts file to have your nodes under the proper sections and make sure that you’ve added group names like below.

openshift\_deployment\_type=origin

openshift\_portal\_net=172.30.0.0/16

openshift\_disable\_check=disk\_availability,memory\_availability

[masters]

master01.eid.com

[etcd]

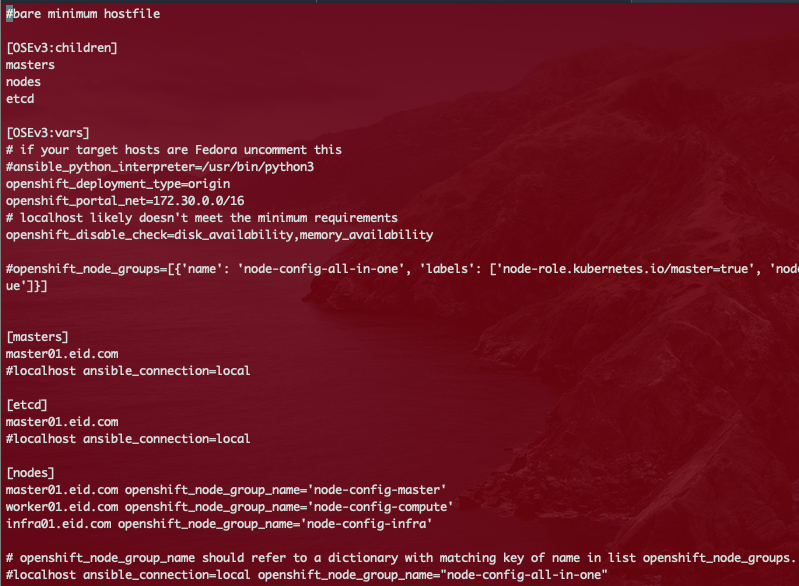
master01.eid.com

[nodes]

master01.eid.com openshift\_node\_group\_name='node-config-master'

worker01.eid.com openshift\_node\_group\_name='node-config-compute'

infra01.eid.com openshift\_node\_group\_name='node-config-infra'

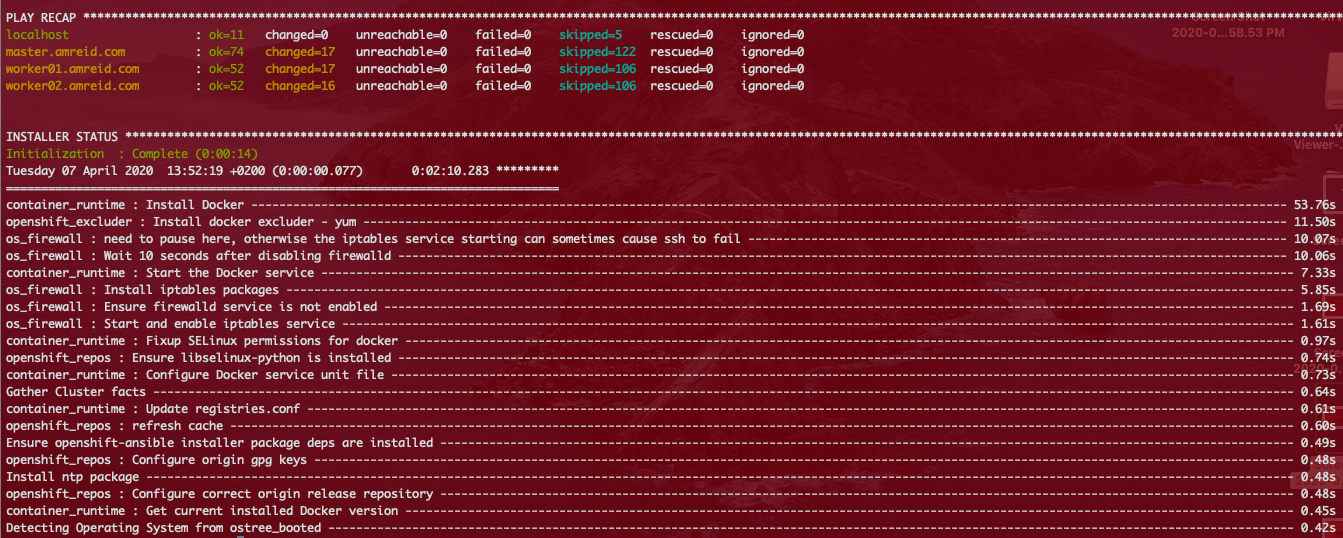


**Step 03**

Make sure that you are on the openshift-ansible root folder then run the prerequisites.yml script

ansible-playbook playbooks/prerequisites.yml

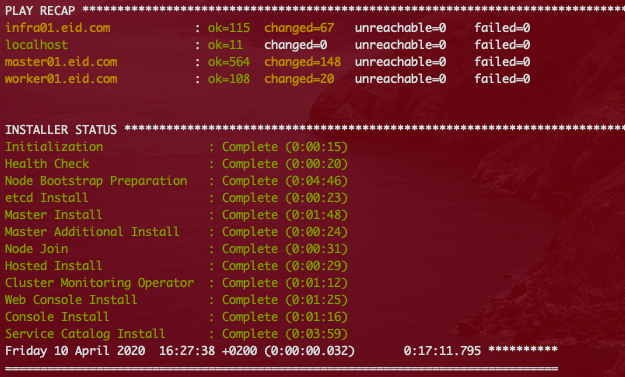
This prerequisites script is testing the environment before the installation. You should receive green results like below.



**Step 04**

ansible-playbook playbooks/deploy\_cluster.ym

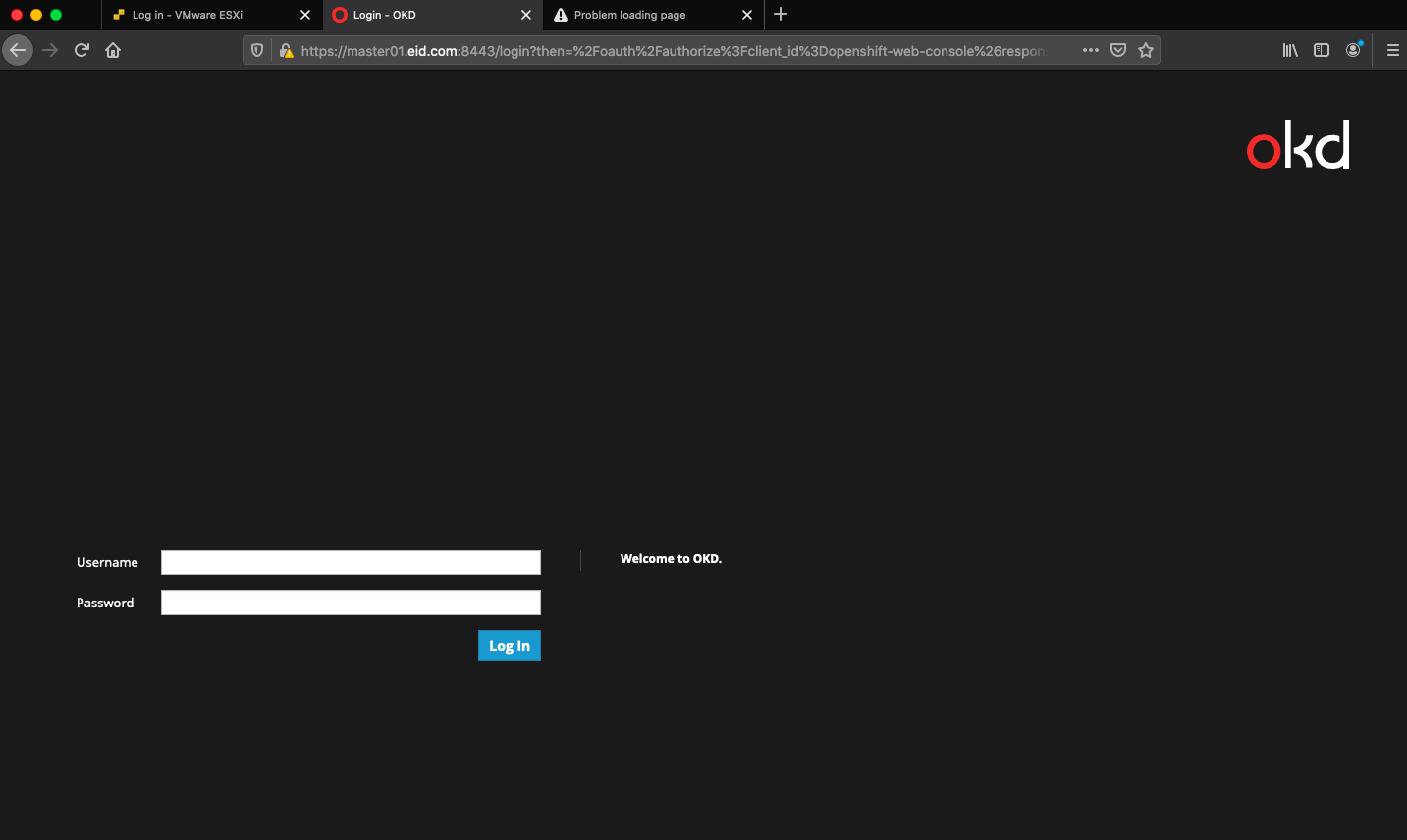
This is going to take time (~15min), then you should receive green results similar to the previous step.



**Step 05** - Validating and Testing your open shift environment

With the previous good news, you can add your master node IP to your hosts file and then using https open your browser to access openshift console, note that openshift 3.11 is using port 8433.

<https://master01.eid.com:8443>

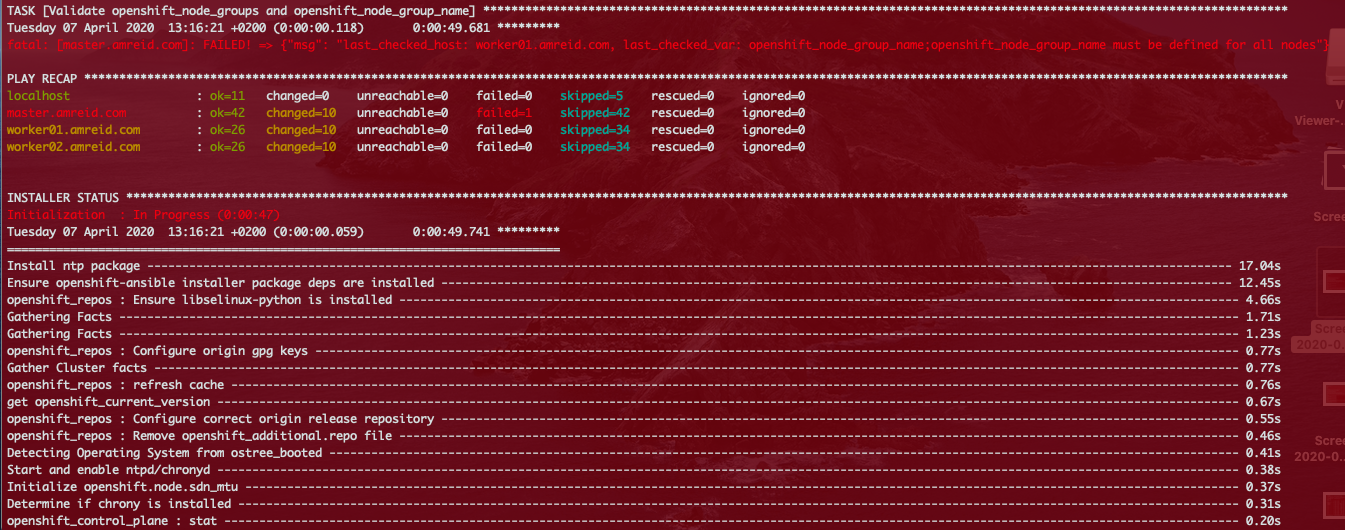
****

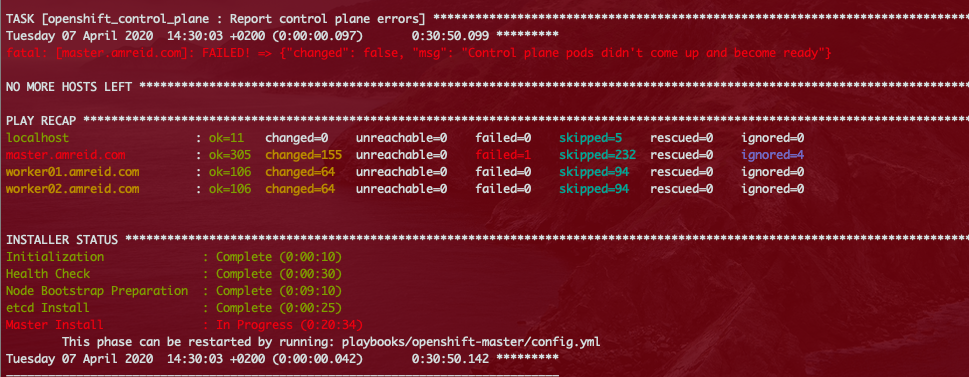
You can try log in using developer / any password

****

**Troubleshooting**

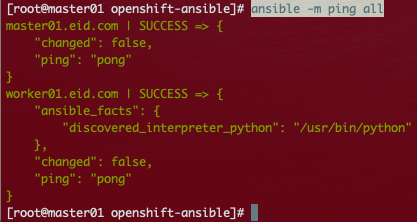
This section is trying to capture some of the errors which may happen while trying to install specially for your first time.





In order to fix these kinds of errors, Here are some notes you can keep in mind.

* Try using -vvv to run the scripts in verbose mode if you want to have more descriptive errors.
* Any connection related errors (e.x SSH connection to any node), most probably it will be related to the host preparation section in this document so, give Dnsmasq setup your main focus. I’ve listed it above in detail.
* A lot of ansible script issues will be related to missing some detail in Ansible hosts file. So review the hosts file carefully and make sure that the group name is setted for all the nodes.
* Before initiating the deploy script, make sure you can access all the nodes from Master Node using Ansible like below.
* ansible -m ping all
* The results should be looks like



* You may need to install Ansible stable version well tested with Openshift 3.1 recommended version is: 2.7.1

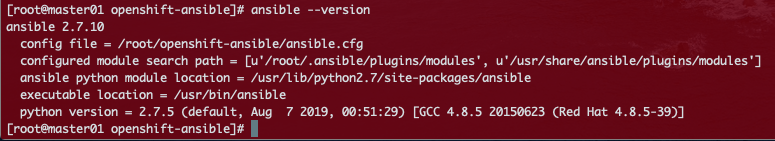
How to install Ansible 2.7.1

curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

python get-pip.py

pip install ansible==2.7.10

ansible --version



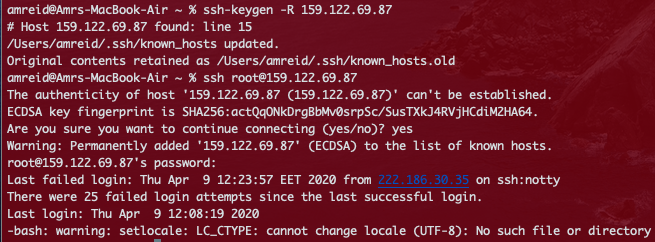
* You may face some issues while logging into the dns server like below

**@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@**

**@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @**

**@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@**

How to fix it, simply run the below command.

ssh-keygen -R ****