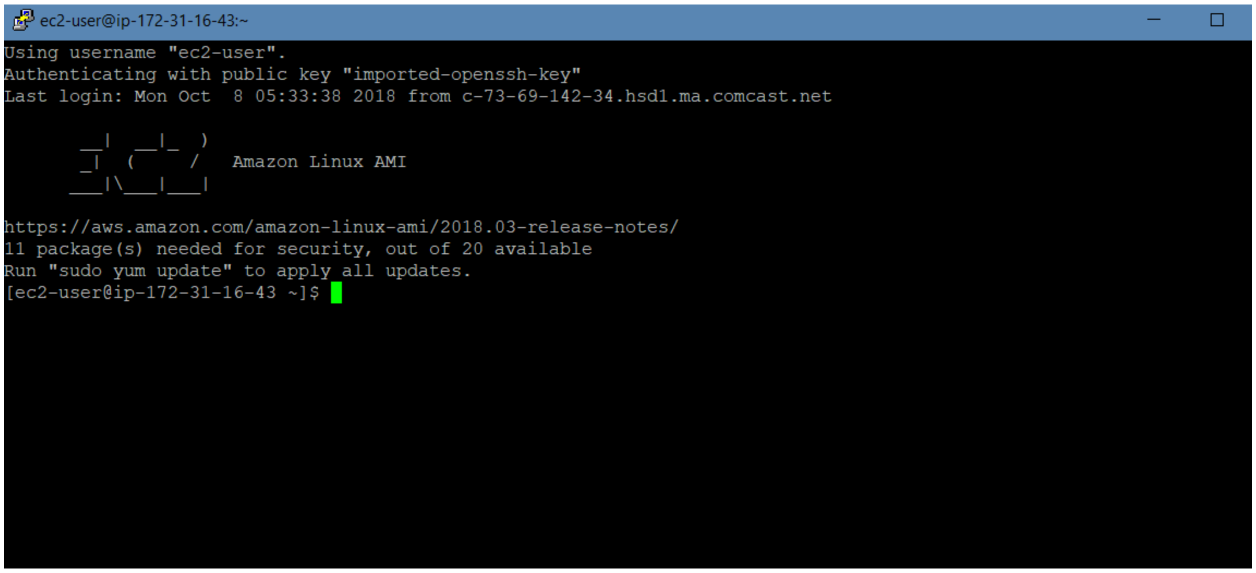
Installing Kubernetes on AWS EC2

Prerequisites:

– Ubuntu instance (You may use other linux instance as well)
– AWS-cli setup
– S3 bucket

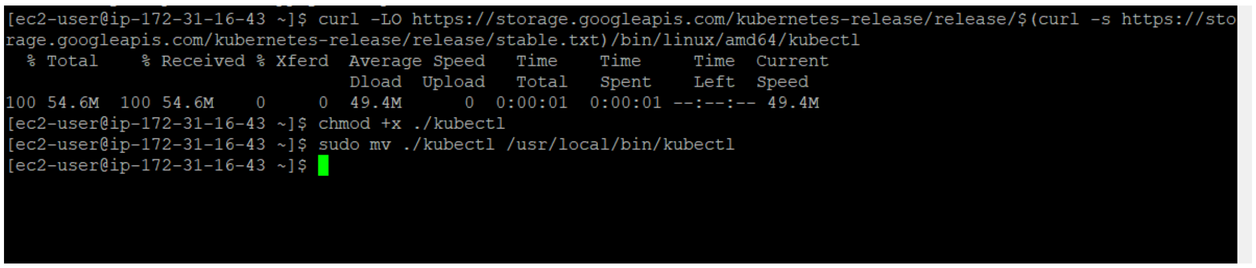


```
ec2-user@ip-172-31-16-43:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Mon Oct  8 05:33:38 2018 from c-73-69-142-34.hsd1.ma.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
11 package(s) needed for security, out of 20 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-16-43 ~]$
```

Install kubectl

On ubuntu instance, make sure you have AWS cli and KOPS setup. We shall also need **kubectl** (Kubernetes cli)
– Install Kubectl on Linux:

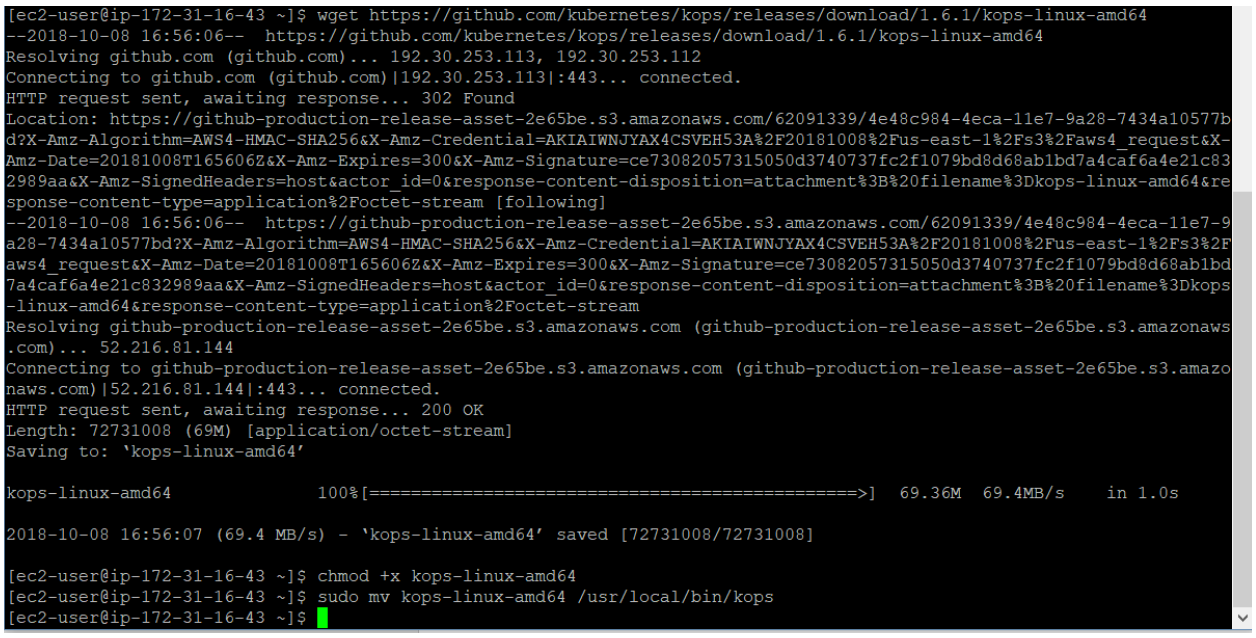
```
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin/kubectl
```



```
[ec2-user@ip-172-31-16-43 ~]$ curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://sto
rage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 54.6M  100 54.6M    0     0  49.4M      0  0:00:01  0:00:01 --:--:-- 49.4M
[ec2-user@ip-172-31-16-43 ~]$ chmod +x ./kubectl
[ec2-user@ip-172-31-16-43 ~]$ sudo mv ./kubectl /usr/local/bin/kubectl
[ec2-user@ip-172-31-16-43 ~]$ 
```

Now, let’s install kops on ubuntu box:

wget https://github.com/kubernetes/kops/releases/download/1.6.1/kops-linux-amd64
chmod +x kops-linux-amd64
sudo mv kops-linux-amd64 /usr/local/bin/kops



```
[ec2-user@ip-172-31-16-43 ~]$ wget https://github.com/kubernetes/kops/releases/download/1.6.1/kops-linux-amd64
--2018-10-08 16:56:06--  https://github.com/kubernetes/kops/releases/download/1.6.1/kops-linux-amd64
Resolving github.com (github.com)... 192.30.253.113, 192.30.253.112
Connecting to github.com (github.com)|192.30.253.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/62091339/4e48c984-4eca-11e7-9a28-7434a10577b
d?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20181008%2Fus-east-1%2Fs3%2Faws4_request&X-
Amz-Date=20181008T165606Z&X-Amz-Expires=300&X-Amz-Signature=ce73082057315050d3740737fc2f1079bd8d68ab1bd7a4caf6a4e21c83
2989aa&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dkops-linux-amd64&re
sponse-content-type=application%2Foctet-stream [following]
--2018-10-08 16:56:06--  https://github-production-release-asset-2e65be.s3.amazonaws.com/62091339/4e48c984-4eca-11e7-9
a28-7434a10577bd?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20181008%2Fus-east-1%2Fs3%2F
aws4_request&X-Amz-Date=20181008T165606Z&X-Amz-Expires=300&X-Amz-Signature=ce73082057315050d3740737fc2f1079bd8d68ab1bd
7a4caf6a4e21c832989aa&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dkops
-linux-amd64&response-content-type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws
.com)... 52.216.81.144
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazo
naws.com)|52.216.81.144|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72731008 (69M) [application/octet-stream]
Saving to: ‘kops-linux-amd64’

kops-linux-amd64             100%[=============================================>]  69.36M  69.4MB/s    in 1.0s

2018-10-08 16:56:07 (69.4 MB/s) - ‘kops-linux-amd64’ saved [72731008/72731008]

[ec2-user@ip-172-31-16-43 ~]$ chmod +x kops-linux-amd64
[ec2-user@ip-172-31-16-43 ~]$ sudo mv kops-linux-amd64 /usr/local/bin/kops
[ec2-user@ip-172-31-16-43 ~]$ 
```

Create Route53 domain for the cluster

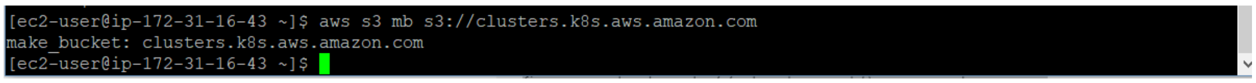
kubernetes make use of DNS for discovery within the cluster so that you can reach out kubernetes-API-server from clients.

Create a hosted zone on Route53, say, ***k8s.appychip.vpc***. The API server endpoint will then be ***api.k8s.appychip.vpc***

Create a S3 bucket

Now, create a S3 bucket to store the configuration for the cluster. Make sure the instance have right role to access S3 and Route53:

```
$ aws s3 mb s3://clusters.k8s.appychip.vpc
```



```
[ec2-user@ip-172-31-16-43 ~]$ aws s3 mb s3://clusters.k8s.aws.amazon.com
make_bucket: clusters.k8s.aws.amazon.com
[ec2-user@ip-172-31-16-43 ~]$
```

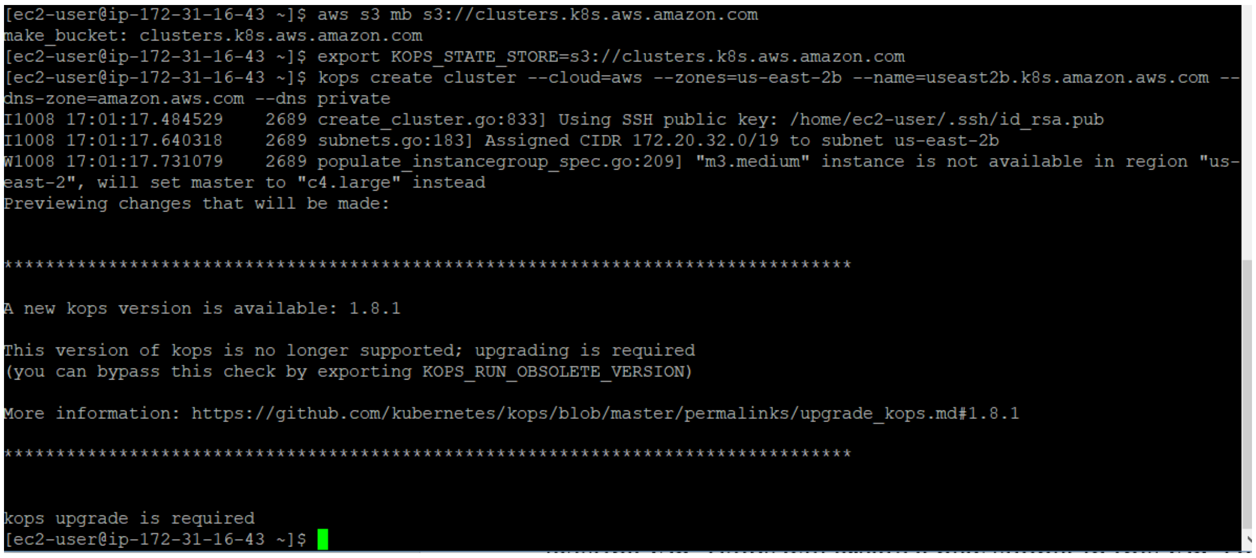
Expose environment variable:

```
$ export KOPS_STATE_STORE=s3://clusters.k8s.appychip.vpc
```

Create Kubernetes Cluster

Now comes the interesting part to create the cluster. You can reuse existing VPC (kops will create a new subnet in this VPC) by providing the **vpc-id**option. The following command will give you details what all things are going to happen:

```
$ kops create cluster --cloud=aws --zones=us-east-1d --
name=useast1.k8s.appychip.vpc --dns-zone=appychip.vpc --dns private
```



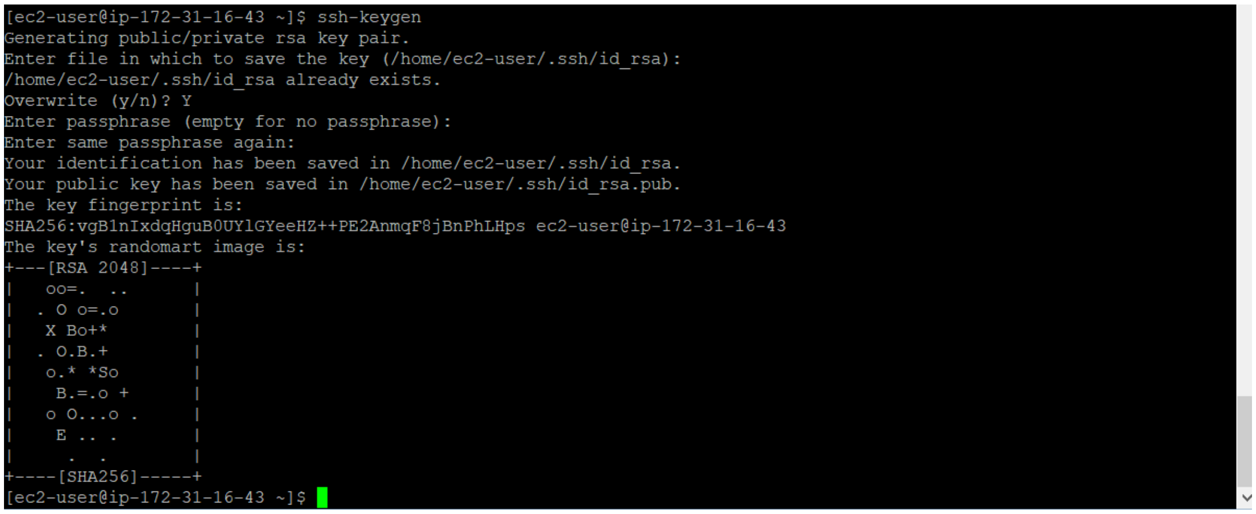
```
[ec2-user@ip-172-31-16-43 ~]$ aws s3 mb s3://clusters.k8s.aws.amazon.com
make_bucket: clusters.k8s.aws.amazon.com
[ec2-user@ip-172-31-16-43 ~]$ export KOPS_STATE_STORE=s3://clusters.k8s.aws.amazon.com
[ec2-user@ip-172-31-16-43 ~]$ kops create cluster --cloud=aws --zones=us-east-2b --name=useast2b.k8s.amazon.aws.com --
dns-zone=amazon.aws.com --dns private
I1008 17:01:17.484529    2689 create_cluster.go:833] Using SSH public key: /home/ec2-user/.ssh/id_rsa.pub
I1008 17:01:17.640318    2689 subnets.go:183] Assigned CIDR 172.20.32.0/19 to subnet us-east-2b
W1008 17:01:17.731079    2689 populate_instancegroup_spec.go:209] "m3.medium" instance is not available in region "us-
east-2", will set master to "c4.large" instead
Previewing changes that will be made:


********************************************************************************

A new kops version is available: 1.8.1

This version of kops is no longer supported; upgrading is required
(you can bypass this check by exporting KOPS_RUN_OBSOLETE_VERSION)

More information: https://github.com/kubernetes/kops/blob/master/permalinks/upgrade_kops.md#1.8.1

********************************************************************************


kops upgrade is required
[ec2-user@ip-172-31-16-43 ~]$
```

UPGRADE KOPS

wget -O kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d '"' -f 4)/kops-linux-amd64

NOTE: Make sure you have ssh keys already generated otherwise it will throw an error.

Generating SSH key

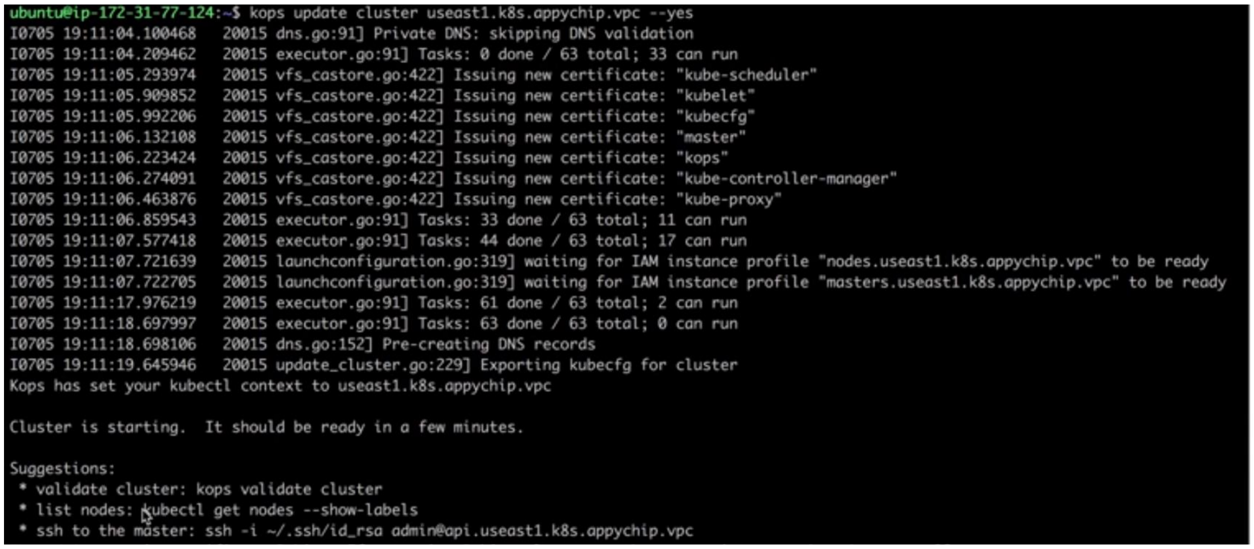


```
[ec2-user@ip-172-31-16-43 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ec2-user/.ssh/id_rsa):
/home/ec2-user/.ssh/id_rsa already exists.
Overwrite (y/n)? Y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/id_rsa.
Your public key has been saved in /home/ec2-user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:vgB1nIxdqHguB0UYlGYeeHZ++PE2AnmqF8jBnPhLHps ec2-user@ip-172-31-16-43
The key's randomart image is:
+---[RSA 2048]----+
|   oo=.  ..      |
|  . O o=.o       |
|   X Bo+*        |
|  . O.B.+        |
|   o.* *So       |
|    B.=.o +      |
|   o O...o .     |
|    E .. .       |
|     .  .        |
+----[SHA256]-----+
[ec2-user@ip-172-31-16-43 ~]$
```

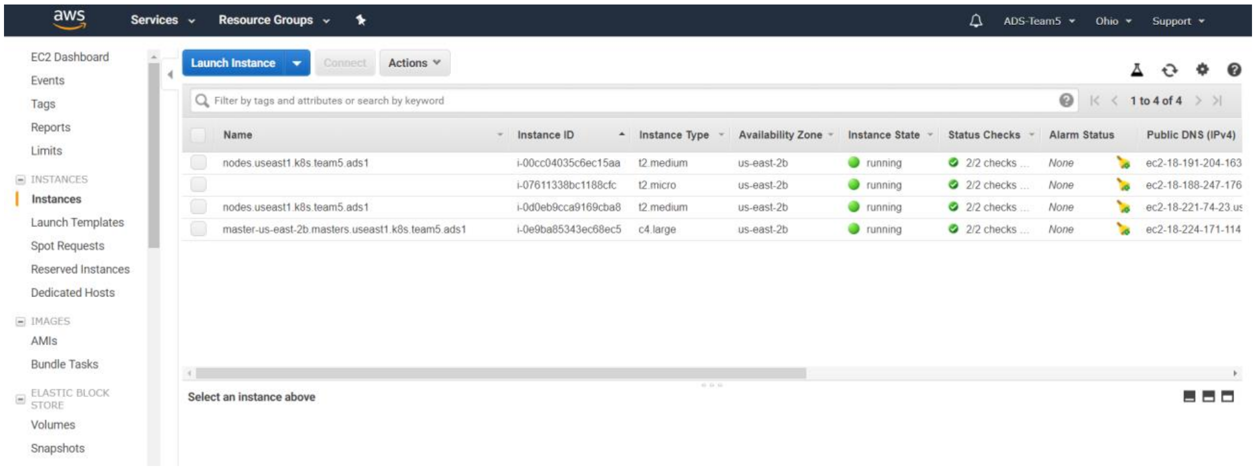
Now to actually create cluster run the following command:

kops update cluster useast1.k8s.appychip.vpc --yes

Now after the update cluster command has been executed the clusters will be created in AWS EC2

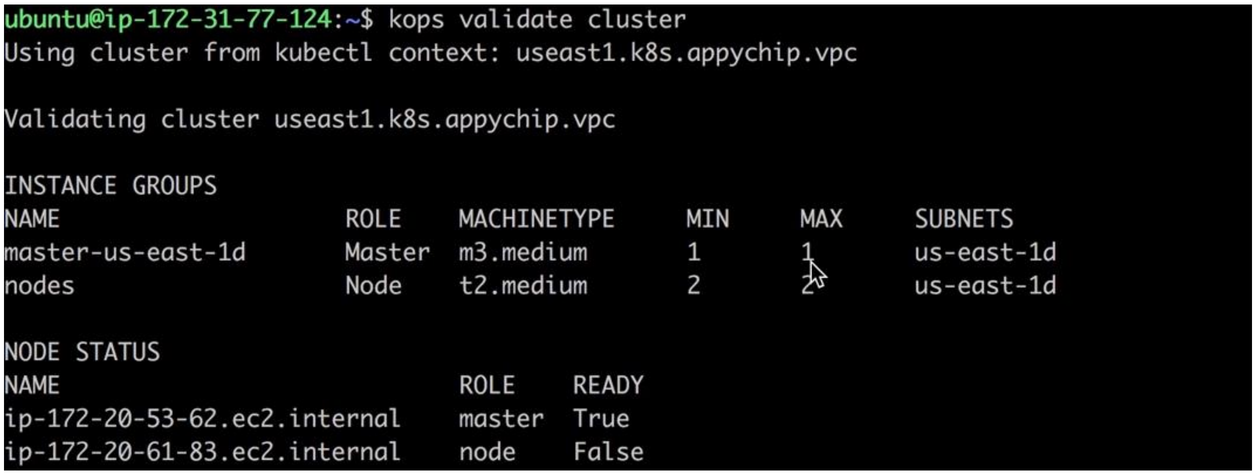


```
ubuntu@ip-172-31-77-124:~$ kops update cluster useast1.k8s.appychip.vpc --yes
I0705 19:11:04.100468   20015 dns.go:91] Private DNS: skipping DNS validation
I0705 19:11:04.209462   20015 executor.go:91] Tasks: 0 done / 63 total; 33 can run
I0705 19:11:05.293974   20015 vfs_castore.go:422] Issuing new certificate: "kube-scheduler"
I0705 19:11:05.909852   20015 vfs_castore.go:422] Issuing new certificate: "kubelet"
I0705 19:11:05.992206   20015 vfs_castore.go:422] Issuing new certificate: "kubecfg"
I0705 19:11:06.132108   20015 vfs_castore.go:422] Issuing new certificate: "master"
I0705 19:11:06.223424   20015 vfs_castore.go:422] Issuing new certificate: "kops"
I0705 19:11:06.274091   20015 vfs_castore.go:422] Issuing new certificate: "kube-controller-manager"
I0705 19:11:06.463876   20015 vfs_castore.go:422] Issuing new certificate: "kube-proxy"
I0705 19:11:06.859543   20015 executor.go:91] Tasks: 33 done / 63 total; 11 can run
I0705 19:11:07.577418   20015 executor.go:91] Tasks: 44 done / 63 total; 17 can run
I0705 19:11:07.721639   20015 launchconfiguration.go:319] waiting for IAM instance profile "nodes.useast1.k8s.appychip.vpc" to be ready
I0705 19:11:07.722705   20015 launchconfiguration.go:319] waiting for IAM instance profile "masters.useast1.k8s.appychip.vpc" to be ready
I0705 19:11:17.976219   20015 executor.go:91] Tasks: 61 done / 63 total; 2 can run
I0705 19:11:18.697997   20015 executor.go:91] Tasks: 63 done / 63 total; 0 can run
I0705 19:11:18.698106   20015 dns.go:152] Pre-creating DNS records
I0705 19:11:19.645946   20015 update_cluster.go:229] Exporting kubecfg for cluster
Kops has set your kubectl context to useast1.k8s.appychip.vpc

Cluster is starting.  It should be ready in a few minutes.

Suggestions:
 * validate cluster: kops validate cluster
 * list nodes: kubectl get nodes --show-labels
 * ssh to the master: ssh -i ~/.ssh/id_rsa admin@api.useast1.k8s.appychip.vpc
```



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
nodes.useast1.k8s.team5.ads1	i-00cc04035c6ec15aa	t2.medium	us-east-2b	running	2/2 checks ...	None	ec2-18-191-204-163
	i-07611338bc1188cfc	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-18-188-247-176
nodes.useast1.k8s.team5.ads1	i-0d0eb9cca9169cba8	t2.medium	us-east-2b	running	2/2 checks ...	None	ec2-18-221-74-23.us
master-us-east-2b.masters.useast1.k8s.team5.ads1	i-0e9ba85343ec68ec5	c4.large	us-east-2b	running	2/2 checks ...	None	ec2-18-224-171-114

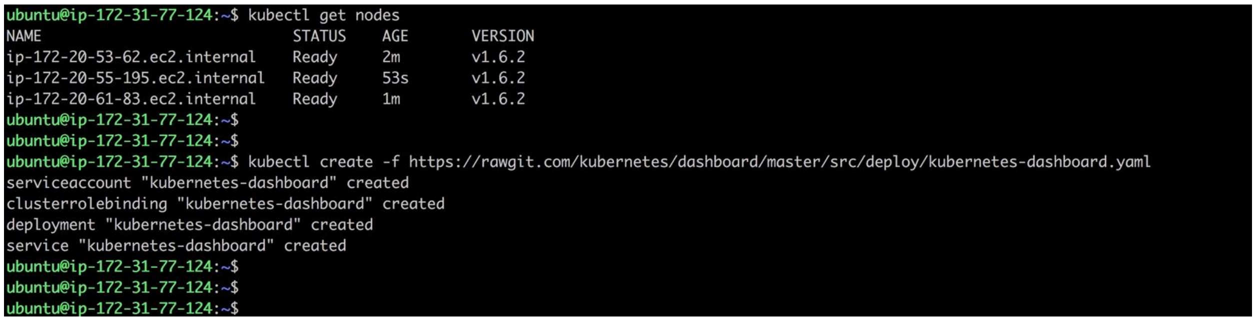
Select an instance above



```
ubuntu@ip-172-31-77-124:~$ kops validate cluster
Using cluster from kubectl context: useast1.k8s.appychip.vpc

Validating cluster useast1.k8s.appychip.vpc

INSTANCE GROUPS
NAME                    ROLE    MACHINETYPE     MIN     MAX     SUBNETS
master-us-east-1d       Master  m3.medium       1       1       us-east-1d
nodes                   Node    t2.medium       2       2       us-east-1d

NODE STATUS
NAME                            ROLE    READY
ip-172-20-53-62.ec2.internal    master  True
ip-172-20-61-83.ec2.internal    node    False
```

To get nodes

Run—kubectl get nodes



```
ubuntu@ip-172-31-77-124:~$ kubectl get nodes
NAME                            STATUS    AGE       VERSION
ip-172-20-53-62.ec2.internal    Ready     2m        v1.6.2
ip-172-20-55-195.ec2.internal   Ready     53s       v1.6.2
ip-172-20-61-83.ec2.internal    Ready     1m        v1.6.2
ubuntu@ip-172-31-77-124:~$
ubuntu@ip-172-31-77-124:~$
ubuntu@ip-172-31-77-124:~$ kubectl create -f https://rawgit.com/kubernetes/dashboard/master/src/deploy/kubernetes-dashboard.yaml
serviceaccount "kubernetes-dashboard" created
clusterrolebinding "kubernetes-dashboard" created
deployment "kubernetes-dashboard" created
service "kubernetes-dashboard" created
ubuntu@ip-172-31-77-124:~$
ubuntu@ip-172-31-77-124:~$
ubuntu@ip-172-31-77-124:~$
```

Deploying Nginx Container

Let’s deploy a simple service made up of some nginx containers:

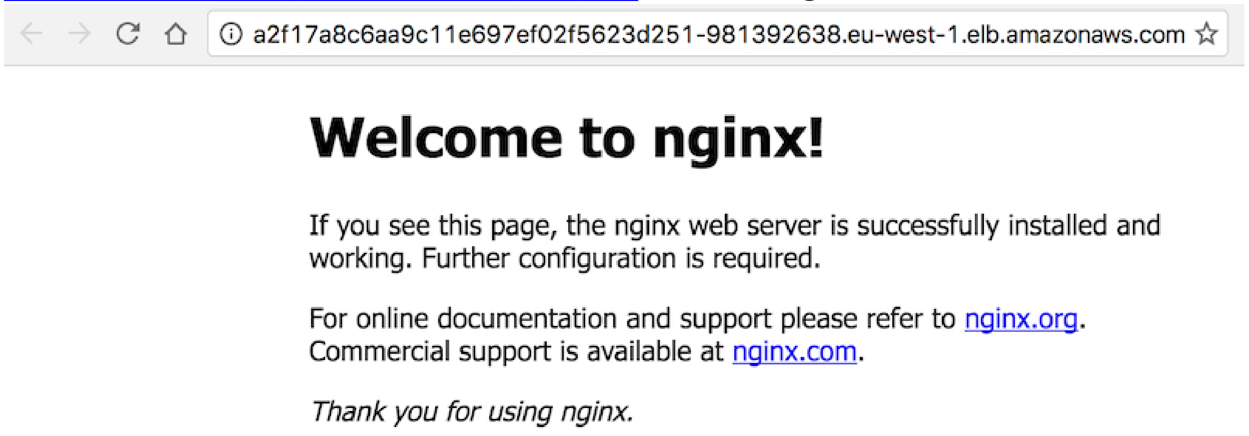
Create an nginx deployment:

```
$ kubectl run sample-nginx --image=nginx --replicas=2 --port=80
$ kubectl get pods
NAME                    READY    STATUS    RESTARTS   AGE
sample-nginx-379829228-xb9y3   1/1      Running   0        10s
sample-nginx-379829228-yhd25   1/1      Running   0        10s
$ kubectl get deployments
NAME      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
sample-nginx   2       2       2          2         29s
```

Expose the deployment as service. This will create an ELB in front of those 2 containers and allow us to publicly access them:

```
$ kubectl expose deployment sample-nginx --port=80 --type=LoadBalancer
$ kubectl get services -o wide
NAME        CLUSTER-IP     EXTERNAL-IP                                   PORT(S)
AGE      SELECTOR
kubernetes   100.64.0.1     <none>                                       443/TCP
25m      <none>
sample-nginx     100.70.129.69   adca6650a60e611e7a66612ae64874d4-
175711331.us-east-1.elb.amazonaws.com/   80/TCP    19m      run=sample-nginx
```

There is an ELB running on http://adca6650a60e611e7a66612ae64874d4-175711331.us-east-1.elb.amazonaws.com with our nginx containers behind it:


a2f17a8c6aa9c11e697ef02f5623d251-981392638.eu-west-1.elb.amazonaws.com

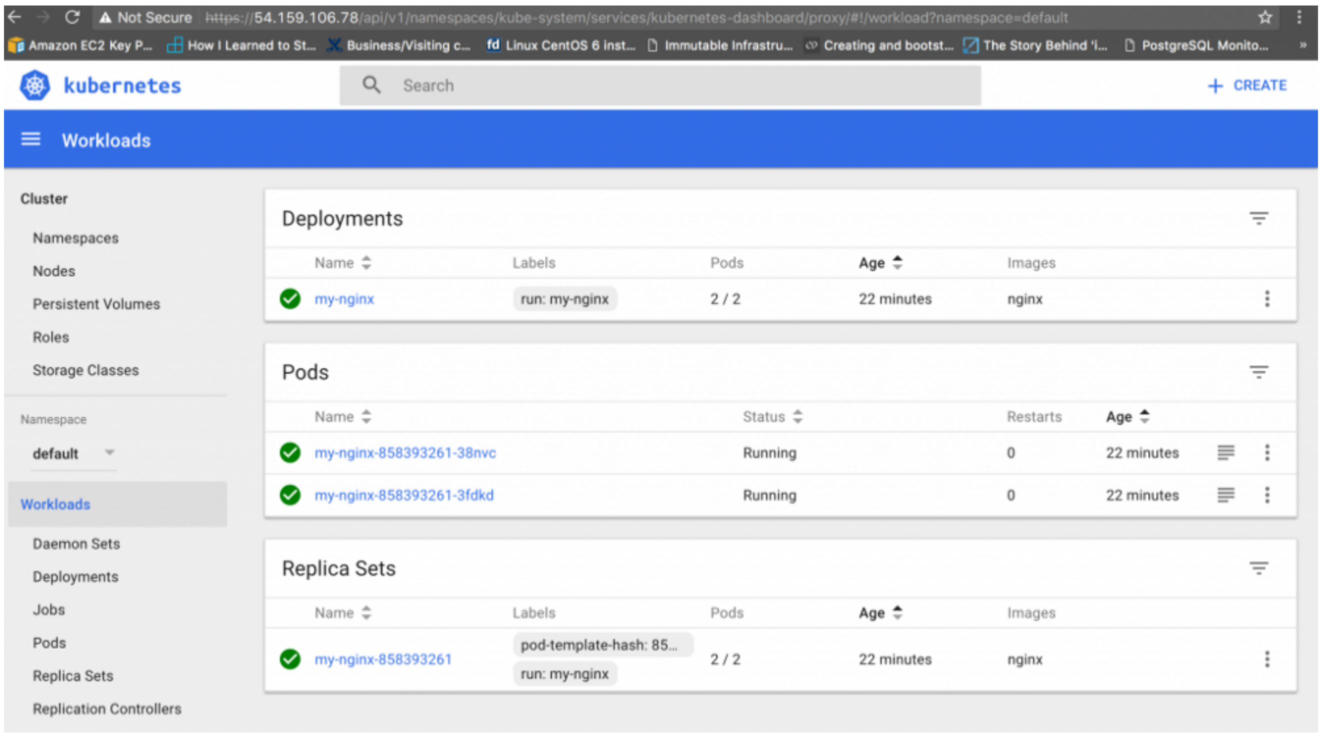
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

You can also view the UI by accessing master node. Hit master node's IP/Domain in browser, it will ask for credentials. Run command ***kubectl config view*** to see the credentials.



Deployments

Name	Labels	Pods	Age	Images
my-nginx	run: my-nginx	2 / 2	22 minutes	nginx

Pods

Name	Status	Restarts	Age
my-nginx-858393261-38nvc	Running	0	22 minutes
my-nginx-858393261-3fdkd	Running	0	22 minutes

Replica Sets

Name	Labels	Pods	Age	Images
my-nginx-858393261	pod-template-hash: 85... run: my-nginx	2 / 2	22 minutes	nginx

To delete the cluster and remove all AWS resources with, run the following command:

```
$ kops delete cluster --name=useast1.k8s.appychip.vpc –yes
```

That's All is required to setup kubernetes on AWS EC2.