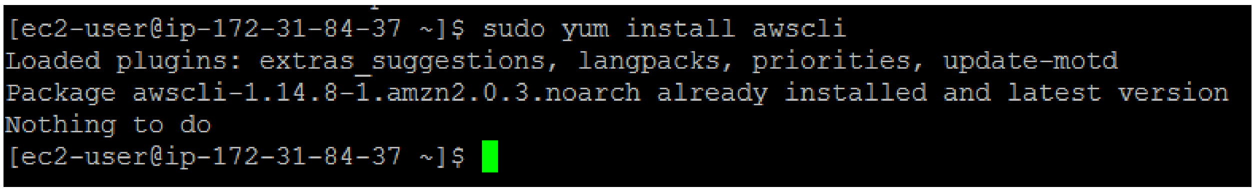
Assumption:

I am assuming that EC2 instance that will be used as Kubernetes master is already setup. And a connection could be established to it.

In this document, I am demonstrating how to setup kubernetes and form a cluster of AWS EC2 instances.

Install AWS CLI:

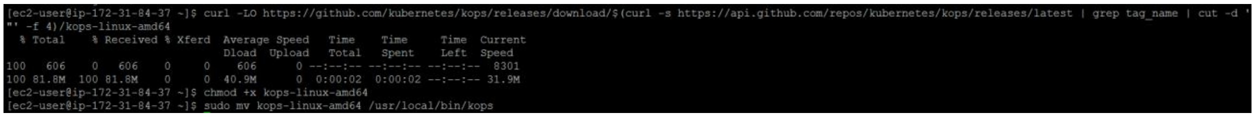
sudo yum install awscli



```
[ec2-user@ip-172-31-84-37 ~]$ sudo yum install awscli
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package awscli-1.14.8-1.amzn2.0.3.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-84-37 ~]$
```

Installing kops:

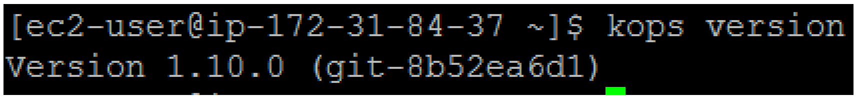
```
curl -LO
https://github.com/kubernetes/kops/releases/download/$(curl -s
https://api.github.com/repos/kubernetes/kops/releases/latest |
grep tag_name | cut -d '"' -f 4)/kops-linux-amd64
chmod +x kops-linux-amd64
sudo mv kops-linux-amd64 /usr/local/bin/kops
```



```
[ec2-user@ip-172-31-84-37 ~]$ curl -LO https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d '
"' -f 4)/kops-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   606    0   606    0     0    606      0 --:--:-- --:--:-- --:--:--  8301
100 81.8M  100 81.8M    0     0  40.9M      0  0:00:02  0:00:02 --:--:-- 31.9M
[ec2-user@ip-172-31-84-37 ~]$ chmod +x kops-linux-amd64
[ec2-user@ip-172-31-84-37 ~]$ sudo mv kops-linux-amd64 /usr/local/bin/kops
```

Verify kops installation:

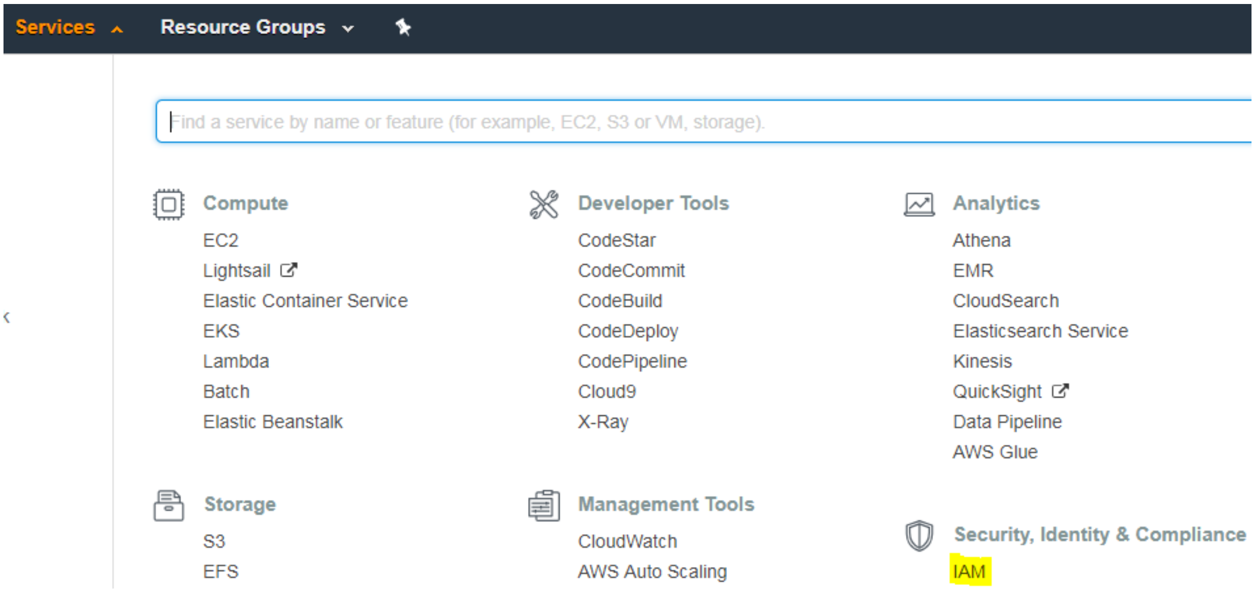
kops version



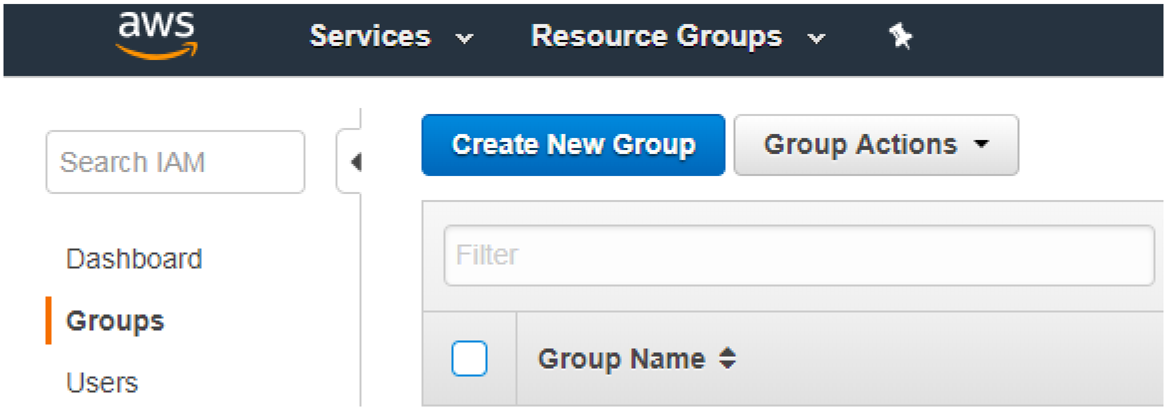
```
[ec2-user@ip-172-31-84-37 ~]$ kops version
Version 1.10.0 (git-8b52ea6d1)
```

Provide below security access to IAM user:

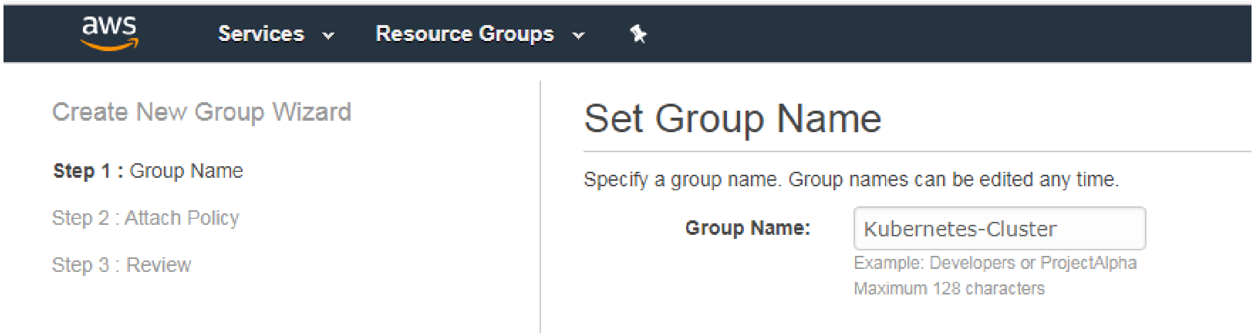
```
AmazonEC2FullAccess
AmazonRoute53FullAccess
AmazonS3FullAccess
AmazonVPCFullAccess
```



Go to Group → Create New Group



Give Group Name



Create New Group Wizard

Step 1 : Group Name

Step 2 : Attach Policy

Step 3 : Review

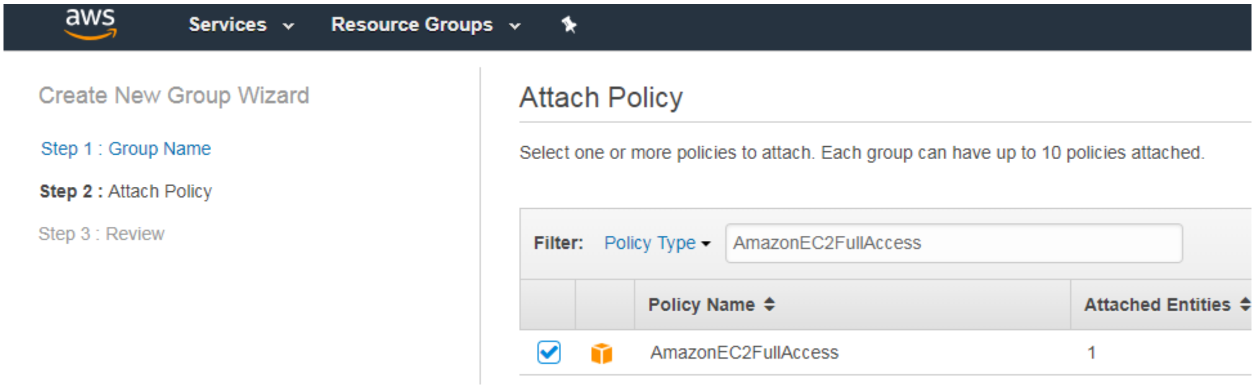
Set Group Name

Specify a group name. Group names can be edited any time.

Group Name: Kubernetes-Cluster

Example: Developers or ProjectAlpha
Maximum 128 characters

Search and select the above mentioned 4 security groups as shown below:



Create New Group Wizard

Step 1 : Group Name

Step 2 : Attach Policy

Step 3 : Review

Attach Policy

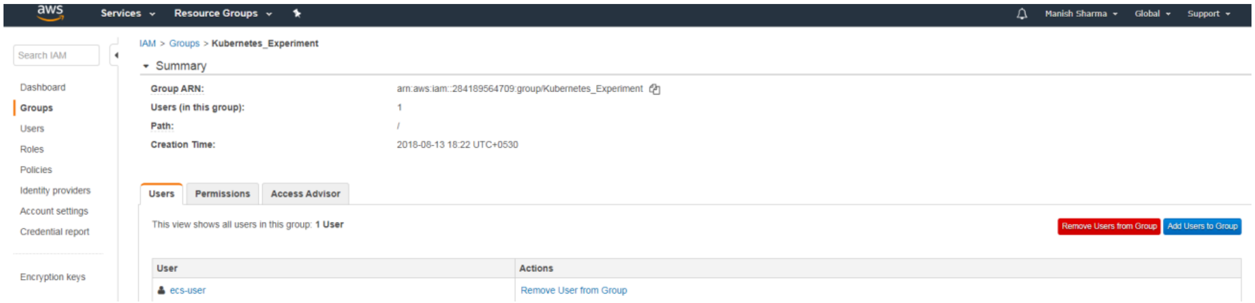
Select one or more policies to attach. Each group can have up to 10 policies attached.

Filter: Policy Type ▾ AmazonEC2FullAccess

		Policy Name	Attached Entities
☑		AmazonEC2FullAccess	1

Finally Review and submit

Add user to the newly formed group



IAM > Groups > Kubernetes_Experiment

Summary

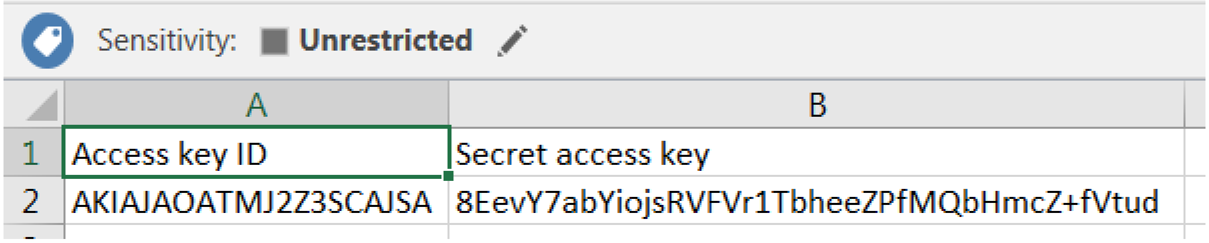
Group ARN:	arn:aws:iam::284189564709:group/Kubernetes_Experiment
Users (in this group):	1
Path:	/
Creation Time:	2018-08-13 18:22 UTC+0530

Users | Permissions | Access Advisor

This view shows all users in this group: **1 User**

User	Actions
ecs-user	Remove User from Group

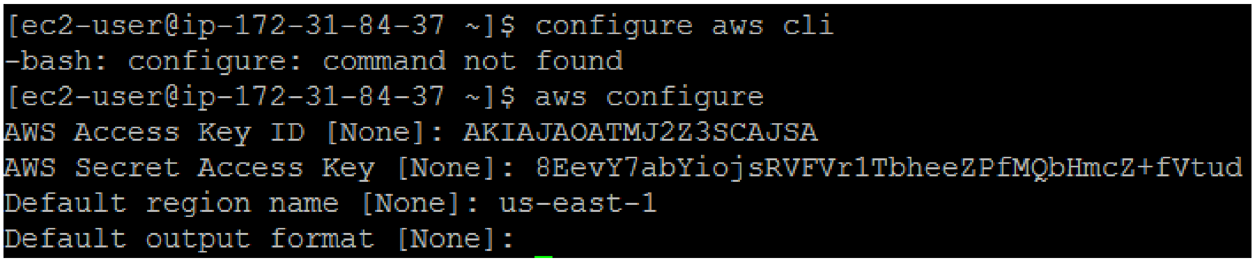
Once user is added, an option to download authorization key will appear



	A	B
1	Access key ID	Secret access key
2	AKIAJAOATMJ2Z3SCAJSA	8EevY7abYiojsRVFVr1TbheeZPfMQbHmcZ+fVtud

Configure the AWS CLI by providing the Access

aws configure



```
[ec2-user@ip-172-31-84-37 ~]$ configure aws cli
-bash: configure: command not found
[ec2-user@ip-172-31-84-37 ~]$ aws configure
AWS Access Key ID [None]: AKIAJAOATMJ2Z3SCAJSA
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
```

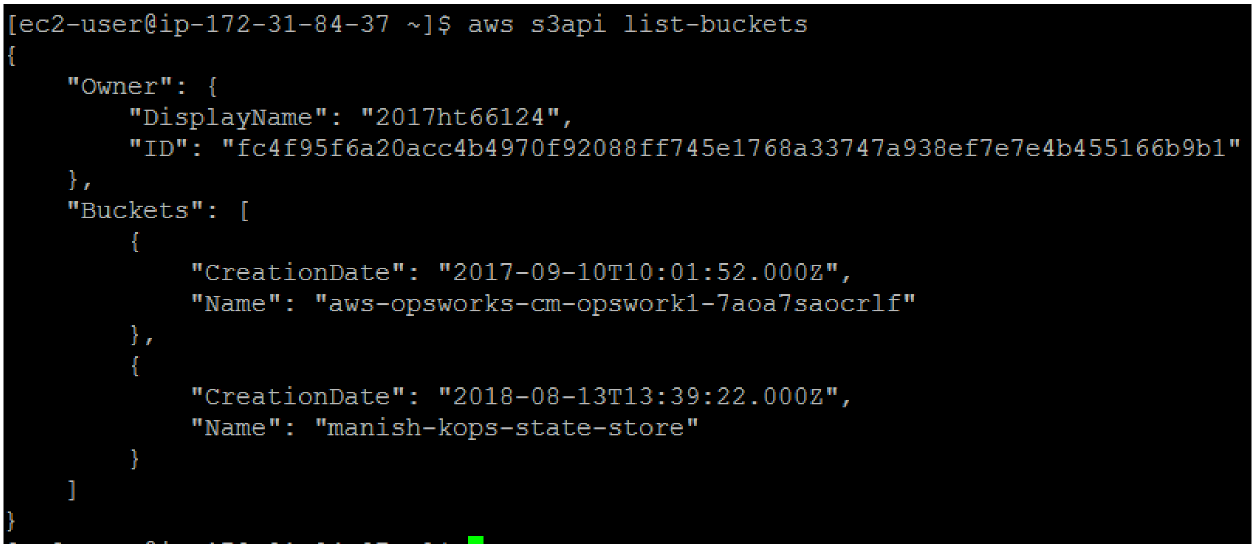
Create S3 bucket

```
bucket_name=manish-kops-state-store
```

```
aws s3api create-bucket \
--bucket ${bucket_name} \
--region us-east-1
```

List your buckets

```
aws s3api list-buckets
```



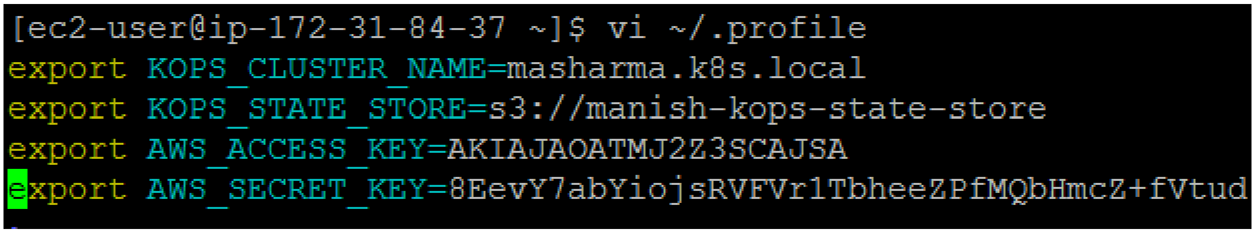
```
[ec2-user@ip-172-31-84-37 ~]$ aws s3api list-buckets
{
    "Owner": {
        "DisplayName": "2017ht66124",
        "ID": "fc4f95f6a20acc4b4970f92088ff745e1768a33747a938ef7e7e4b455166b9b1"
    },
    "Buckets": [
        {
            "CreationDate": "2017-09-10T10:01:52.000Z",
            "Name": "aws-opsworks-cm-opswork1-7aoa7saocrlf"
        },
        {
            "CreationDate": "2018-08-13T13:39:22.000Z",
            "Name": "manish-kops-state-store"
        }
    ]
}
```

Enable versioning on your s3 bucket

```
aws s3api put-bucket-versioning --bucket ${bucket_name} --
versioning-configuration Status=Enabled
```

Now Create your cluster using steps 9-12 in the below document:

Exporting cluster, volume and keys in ~/.profile file



```
[ec2-user@ip-172-31-84-37 ~]$ vi ~/.profile
export KOPS_CLUSTER_NAME=masharma.k8s.local
export KOPS_STATE_STORE=s3://manish-kops-state-store
export AWS_ACCESS_KEY=AKIAJAOATMJ2Z3SCAJSA
export AWS_SECRET_KEY=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
```

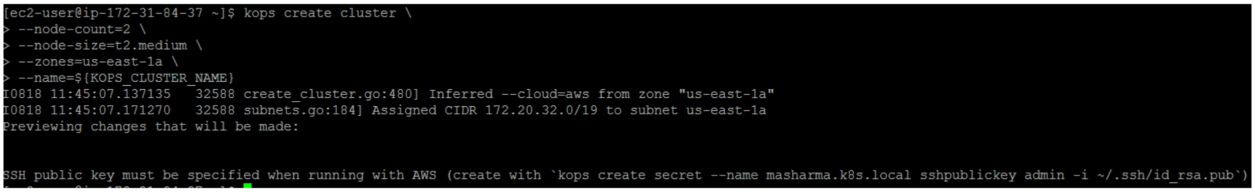
Source this profile file:

source ~/.profile

Creating cluster

```
kops create cluster \
--node-count=2 \
--node-size=t2.medium \
--zones=us-east-1a \
--name=${KOPS_CLUSTER_NAME}
```

Unsuccessful:

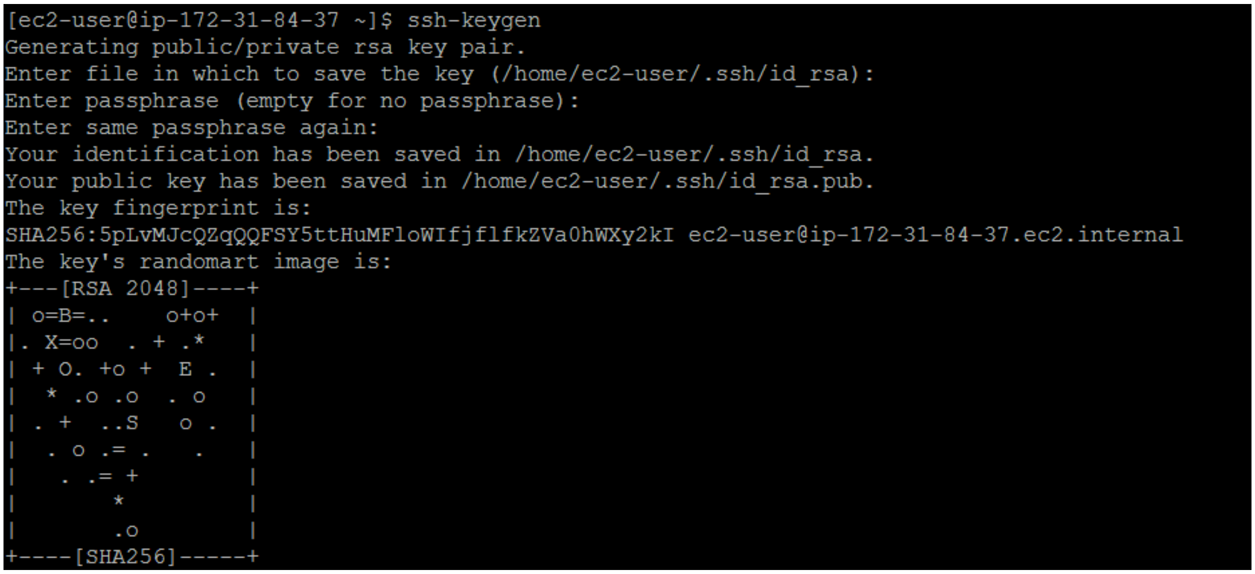


```
[ec2-user@ip-172-31-84-37 ~]$ kops create cluster \
> --node-count=2 \
> --node-size=t2.medium \
> --zones=us-east-1a \
> --name=${KOPS_CLUSTER_NAME}
I0818 11:45:07.137135   32588 create_cluster.go:480] Inferred --cloud=aws from zone "us-east-1a"
I0818 11:45:07.171270   32588 subnets.go:184] Assigned CIDR 172.20.32.0/19 to subnet us-east-1a
Previewing changes that will be made:


SSH public key must be specified when running with AWS (create with `kops create secret --name masharma.k8s.local sshpublickey admin -i ~/.ssh/id_rsa.pub`)
```

Above error is due to ssh key not being setup in this host.

Setup ssh key and try again.

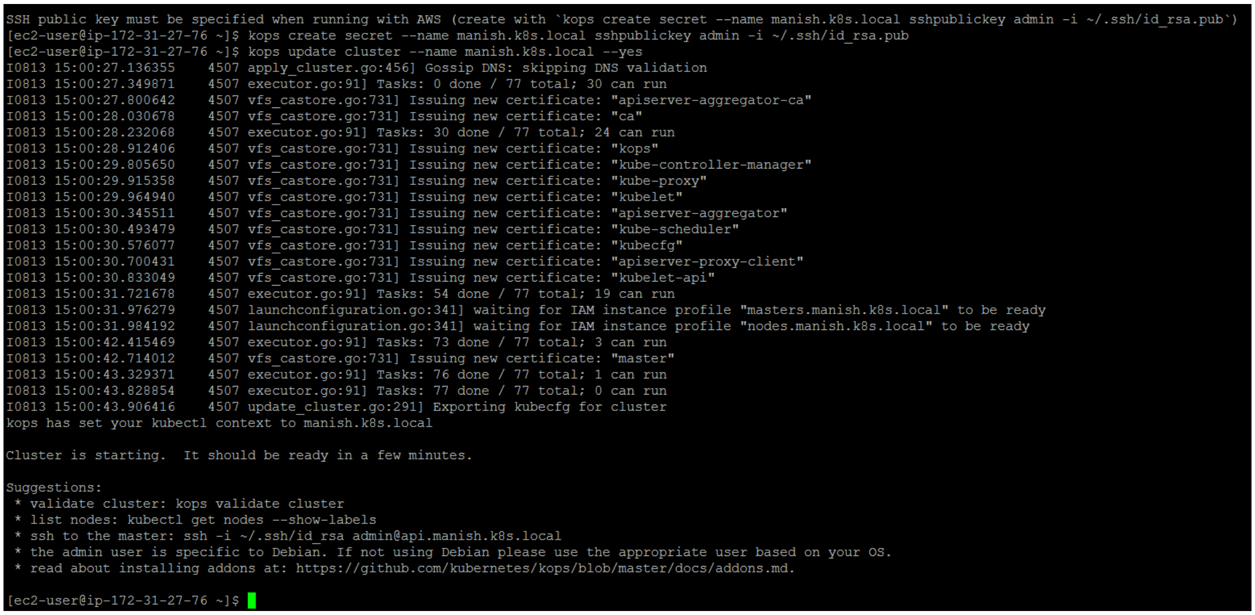
ssh-keygen



```
[ec2-user@ip-172-31-84-37 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ec2-user/.ssh/id_rsa):
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:
SHA256:5pLvMJcQZqQQFSY5ttHuMFloWIfjflfkZVa0hWXy2kI ec2-user@ip-172-31-84-37.ec2.internal
The key's randomart image is:
+---[RSA 2048]----+
| o=B=..    o+o+  |
|. X=oo  . + .*   |
| + O. +o +  E .  |
|  * .o .o  . o   |
| . +  ..S   o .  |
|  . o .= .   .   |
|   . .= +        |
|       *         |
|       .o        |
+----[SHA256]-----+
```

Then retry creating the cluster

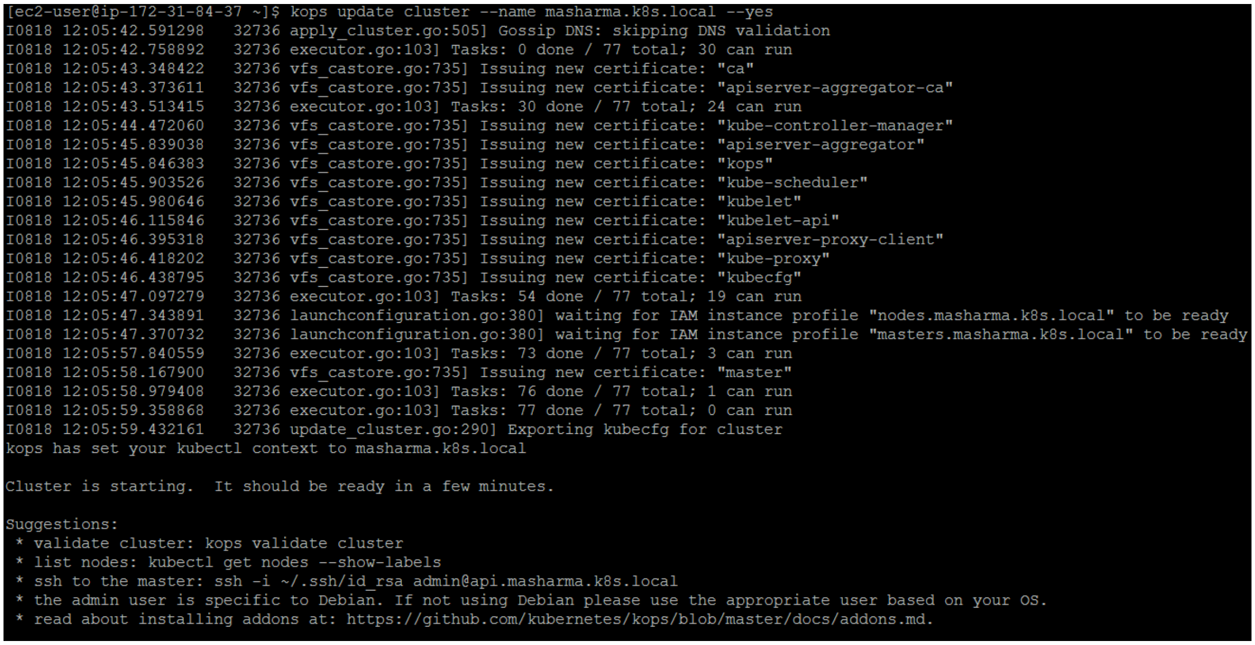
kops create secret --name masharma.k8s.local sshpublickey admin -i ~/.ssh/id_rsa.pub



```
SSH public key must be specified when running with AWS (create with `kops create secret --name manish.k8s.local sshpublickey admin -i ~/.ssh/id_rsa.pub`)
[ec2-user@ip-172-31-27-76 ~]$ kops create secret --name manish.k8s.local sshpublickey admin -i ~/.ssh/id_rsa.pub
[ec2-user@ip-172-31-27-76 ~]$ kops update cluster --name manish.k8s.local --yes
I0813 15:00:27.136355    4507 apply_cluster.go:456] Gossip DNS: skipping DNS validation
I0813 15:00:27.349871    4507 executor.go:91] Tasks: 0 done / 77 total; 30 can run
I0813 15:00:27.800642    4507 vfs_castore.go:731] Issuing new certificate: "apiserver-aggregator-ca"
I0813 15:00:28.030678    4507 vfs_castore.go:731] Issuing new certificate: "ca"
I0813 15:00:28.232068    4507 executor.go:91] Tasks: 30 done / 77 total; 24 can run
I0813 15:00:28.912406    4507 vfs_castore.go:731] Issuing new certificate: "kops"
I0813 15:00:29.805650    4507 vfs_castore.go:731] Issuing new certificate: "kube-controller-manager"
I0813 15:00:29.915358    4507 vfs_castore.go:731] Issuing new certificate: "kube-proxy"
I0813 15:00:29.964940    4507 vfs_castore.go:731] Issuing new certificate: "kubelet"
I0813 15:00:30.345511    4507 vfs_castore.go:731] Issuing new certificate: "apiserver-aggregator"
I0813 15:00:30.493479    4507 vfs_castore.go:731] Issuing new certificate: "kube-scheduler"
I0813 15:00:30.576077    4507 vfs_castore.go:731] Issuing new certificate: "kubecfg"
I0813 15:00:30.700431    4507 vfs_castore.go:731] Issuing new certificate: "apiserver-proxy-client"
I0813 15:00:30.833049    4507 vfs_castore.go:731] Issuing new certificate: "kubelet-api"
I0813 15:00:31.721678    4507 executor.go:91] Tasks: 54 done / 77 total; 19 can run
I0813 15:00:31.976279    4507 launchconfiguration.go:341] waiting for IAM instance profile "masters.manish.k8s.local" to be ready
I0813 15:00:31.984192    4507 launchconfiguration.go:341] waiting for IAM instance profile "nodes.manish.k8s.local" to be ready
I0813 15:00:42.415469    4507 executor.go:91] Tasks: 73 done / 77 total; 3 can run
I0813 15:00:42.714012    4507 vfs_castore.go:731] Issuing new certificate: "master"
I0813 15:00:43.329371    4507 executor.go:91] Tasks: 76 done / 77 total; 1 can run
I0813 15:00:43.828854    4507 executor.go:91] Tasks: 77 done / 77 total; 0 can run
I0813 15:00:43.906416    4507 update_cluster.go:291] Exporting kubecfg for cluster
kops has set your kubectl context to manish.k8s.local

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.manish.k8s.local
 * the admin user is specific to Debian. If not using Debian please use the appropriate user based on your OS.
 * read about installing addons at: https://github.com/kubernetes/kops/blob/master/docs/addons.md.

[ec2-user@ip-172-31-27-76 ~]$ 
```

In case cluster is already existing (as in my case), you need to run update command with –yes

kops update cluster --name masharma.k8s.local –yes

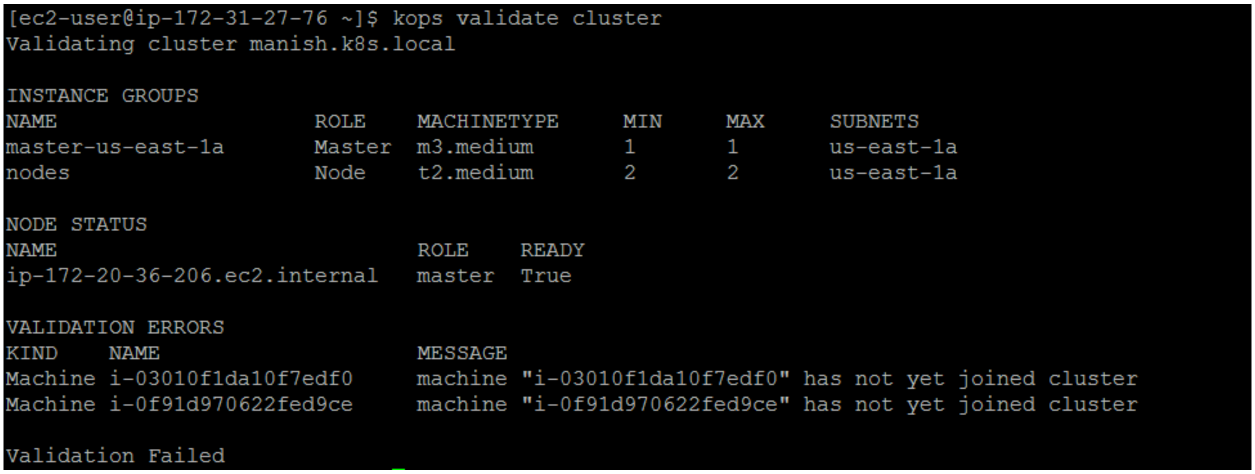


```
[ec2-user@ip-172-31-84-37 ~]$ kops update cluster --name masharma.k8s.local --yes
I0818 12:05:42.591298   32736 apply_cluster.go:505] Gossip DNS: skipping DNS validation
I0818 12:05:42.758892   32736 executor.go:103] Tasks: 0 done / 77 total; 30 can run
I0818 12:05:43.348422   32736 vfs_castore.go:735] Issuing new certificate: "ca"
I0818 12:05:43.373611   32736 vfs_castore.go:735] Issuing new certificate: "apiserver-aggregator-ca"
I0818 12:05:43.513415   32736 executor.go:103] Tasks: 30 done / 77 total; 24 can run
I0818 12:05:44.472060   32736 vfs_castore.go:735] Issuing new certificate: "kube-controller-manager"
I0818 12:05:45.839038   32736 vfs_castore.go:735] Issuing new certificate: "apiserver-aggregator"
I0818 12:05:45.846383   32736 vfs_castore.go:735] Issuing new certificate: "kops"
I0818 12:05:45.903526   32736 vfs_castore.go:735] Issuing new certificate: "kube-scheduler"
I0818 12:05:45.980646   32736 vfs_castore.go:735] Issuing new certificate: "kubelet"
I0818 12:05:46.115846   32736 vfs_castore.go:735] Issuing new certificate: "kubelet-api"
I0818 12:05:46.395318   32736 vfs_castore.go:735] Issuing new certificate: "apiserver-proxy-client"
I0818 12:05:46.418202   32736 vfs_castore.go:735] Issuing new certificate: "kube-proxy"
I0818 12:05:46.438795   32736 vfs_castore.go:735] Issuing new certificate: "kubecfg"
I0818 12:05:47.097279   32736 executor.go:103] Tasks: 54 done / 77 total; 19 can run
I0818 12:05:47.343891   32736 launchconfiguration.go:380] waiting for IAM instance profile "nodes.masharma.k8s.local" to be ready
I0818 12:05:47.370732   32736 launchconfiguration.go:380] waiting for IAM instance profile "masters.masharma.k8s.local" to be ready
I0818 12:05:57.840559   32736 executor.go:103] Tasks: 73 done / 77 total; 3 can run
I0818 12:05:58.167900   32736 vfs_castore.go:735] Issuing new certificate: "master"
I0818 12:05:58.979408   32736 executor.go:103] Tasks: 76 done / 77 total; 1 can run
I0818 12:05:59.358868   32736 executor.go:103] Tasks: 77 done / 77 total; 0 can run
I0818 12:05:59.432161   32736 update_cluster.go:290] Exporting kubecfg for cluster
kops has set your kubectl context to masharma.k8s.local

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.masharma.k8s.local
 * the admin user is specific to Debian. If not using Debian please use the appropriate user based on your OS.
 * read about installing addons at: https://github.com/kubernetes/kops/blob/master/docs/addons.md.
```

Validating clusters

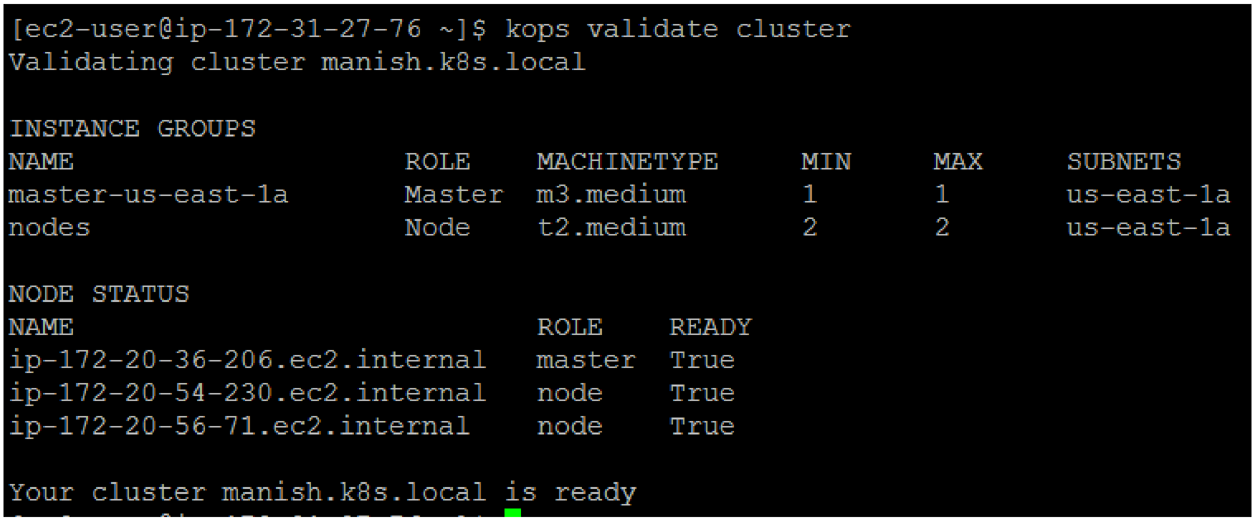
kops validate cluster

Unsuccessful:



```
[ec2-user@ip-172-31-27-76 ~]$ kops validate cluster
Validating cluster manish.k8s.local

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

NODE STATUS
NAME                            ROLE    READY
ip-172-20-36-206.ec2.internal   master  True

VALIDATION ERRORS
KIND    NAME                    MESSAGE
Machine i-03010f1da10f7edf0     machine "i-03010f1da10f7edf0" has not yet joined cluster
Machine i-0f91d970622fed9ce     machine "i-0f91d970622fed9ce" has not yet joined cluster

Validation Failed
```

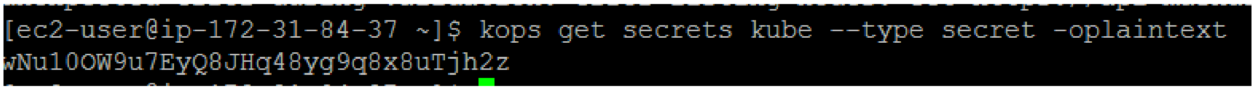
Successful:



```
[ec2-user@ip-172-31-27-76 ~]$ kops validate cluster
Validating cluster manish.k8s.local

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

NODE STATUS
NAME                            ROLE    READY
ip-172-20-36-206.ec2.internal   master  True
ip-172-20-54-230.ec2.internal   node    True
ip-172-20-56-71.ec2.internal    node    True

Your cluster manish.k8s.local is ready
```

Find Admin user password

```
kops get secrets kube --type secret -oplaintext
```



```
[ec2-user@ip-172-31-84-37 ~]$ kops get secrets kube --type secret -oplaintext
wNu10OW9u7EyQ8JHq48yg9q8x8uTjh2z
```

Here I got: wNu10OW9u7EyQ8JHq48yg9q8x8uTjh2z

Reference for above steps:

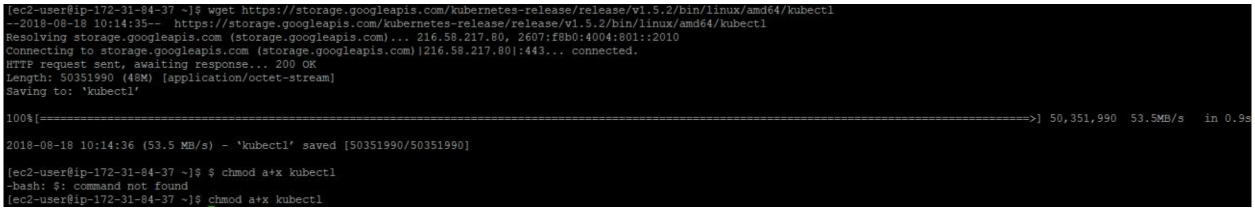
https://medium.com/containermind/how-to-create-a-kubernetes-cluster-on-aws-in-few-minutes-89dda10354f4

Installing kubectl:

```
wget https://storage.googleapis.com/kubernetes-release/release/v1.5.2/bin/linux/amd64/kubectl
```

```
chmod a+x kubectl
```

```
sudo mv kubectl /usr/local/bin/kubectl
```



```
[ec2-user@ip-172-31-84-37 ~]$ wget https://storage.googleapis.com/kubernetes-release/release/v1.5.2/bin/linux/amd64/kubectl
--2018-08-18 10:14:35--  https://storage.googleapis.com/kubernetes-release/release/v1.5.2/bin/linux/amd64/kubectl
Resolving storage.googleapis.com (storage.googleapis.com)... 216.58.217.80, 2607:f8b0:4004:801::2010
Connecting to storage.googleapis.com (storage.googleapis.com)|216.58.217.80|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 50351990 (48M) [application/octet-stream]
Saving to: ‘kubectl’

100%[==========================================================================================================================>] 50,351,990  53.5MB/s   in 0.9s

2018-08-18 10:14:36 (53.5 MB/s) - ‘kubectl’ saved [50351990/50351990]

[ec2-user@ip-172-31-84-37 ~]$ $ chmod a+x kubectl
-bash: $: command not found
[ec2-user@ip-172-31-84-37 ~]$ chmod a+x kubectl
```

Reference for above steps:

https://medium.com/@ikod/setting-up-kubernetes-cluster-in-aws-with-private-topology-7552374c7d7a

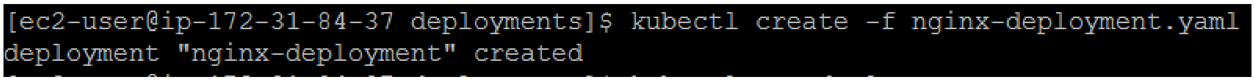
Creating Deployment:

Follow this document:

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

After creating nginx-deployment.yaml, **create the deployment:**

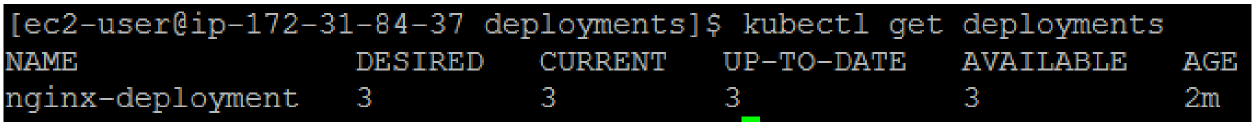
kubectl create -f nginx-deployment.yaml



```
[ec2-user@ip-172-31-84-37 deployments]$ kubectl create -f nginx-deployment.yaml
deployment "nginx-deployment" created
```

Check the status

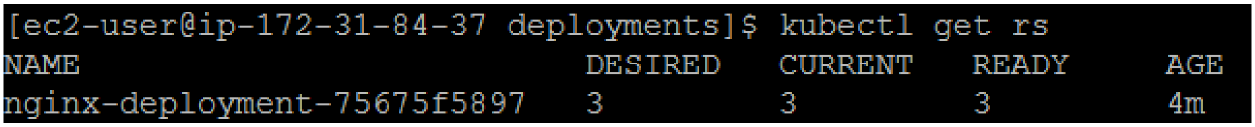
`kubectl get deployments`



```
[ec2-user@ip-172-31-84-37 deployments]$ kubectl get deployments
NAME               DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3         3         3            3           2m
```

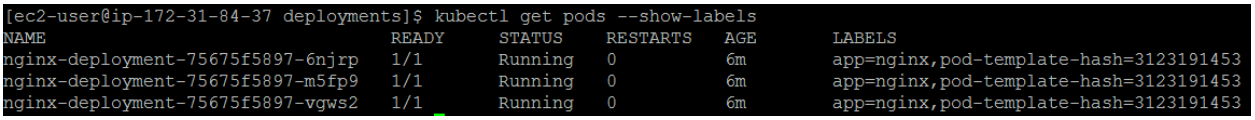
Check Replica Set

kubectl get rs



```
[ec2-user@ip-172-31-84-37 deployments]$ kubectl get rs
NAME                          DESIRED   CURRENT   READY     AGE
nginx-deployment-75675f5897   3         3         3         4m
```

To see the labels automatically generated for each pod, run **`kubectl get pods --show-labels`**



```
[ec2-user@ip-172-31-84-37 deployments]$ kubectl get pods --show-labels
NAME                                READY     STATUS    RESTARTS   AGE       LABELS
nginx-deployment-75675f5897-6njrp   1/1       Running   0          6m        app=nginx,pod-template-hash=3123191453
nginx-deployment-75675f5897-m5fp9   1/1       Running   0          6m        app=nginx,pod-template-hash=3123191453
nginx-deployment-75675f5897-vgws2   1/1       Running   0          6m        app=nginx,pod-template-hash=3123191453
```