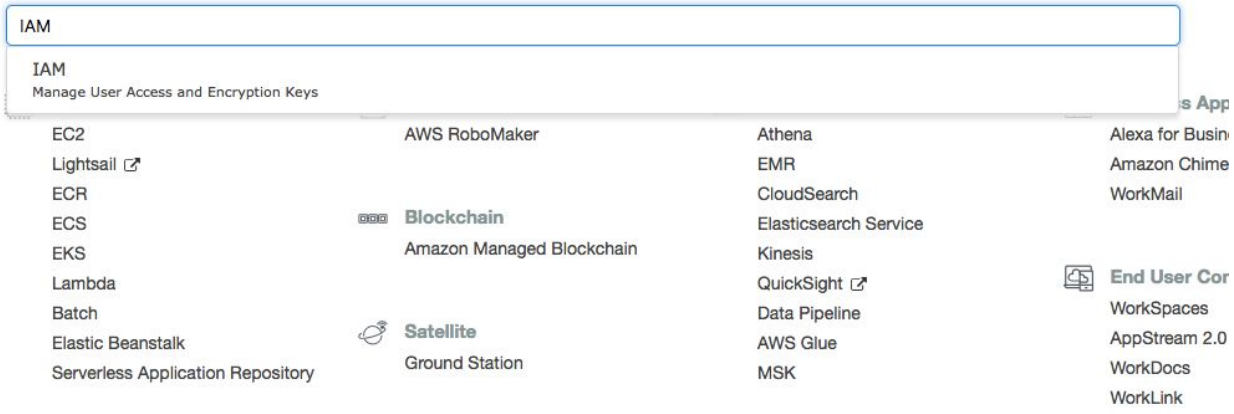
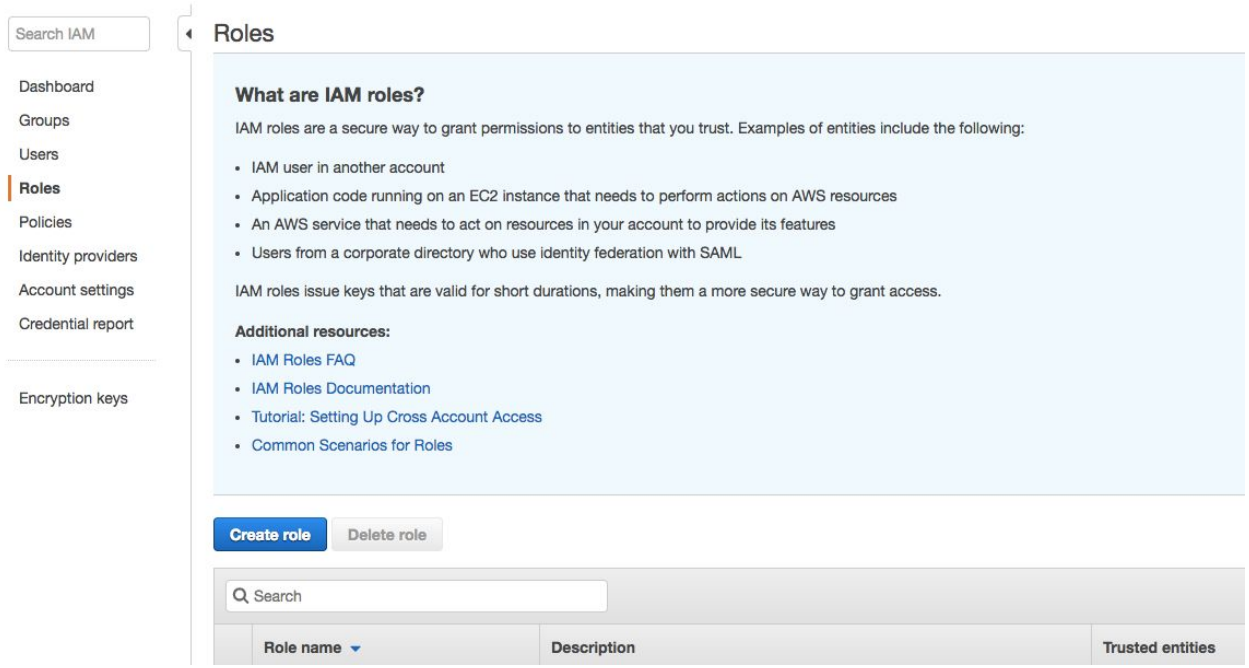
EKS setup

Step1: Create a IAM role to assign permission EKS to control resources on your behalf.

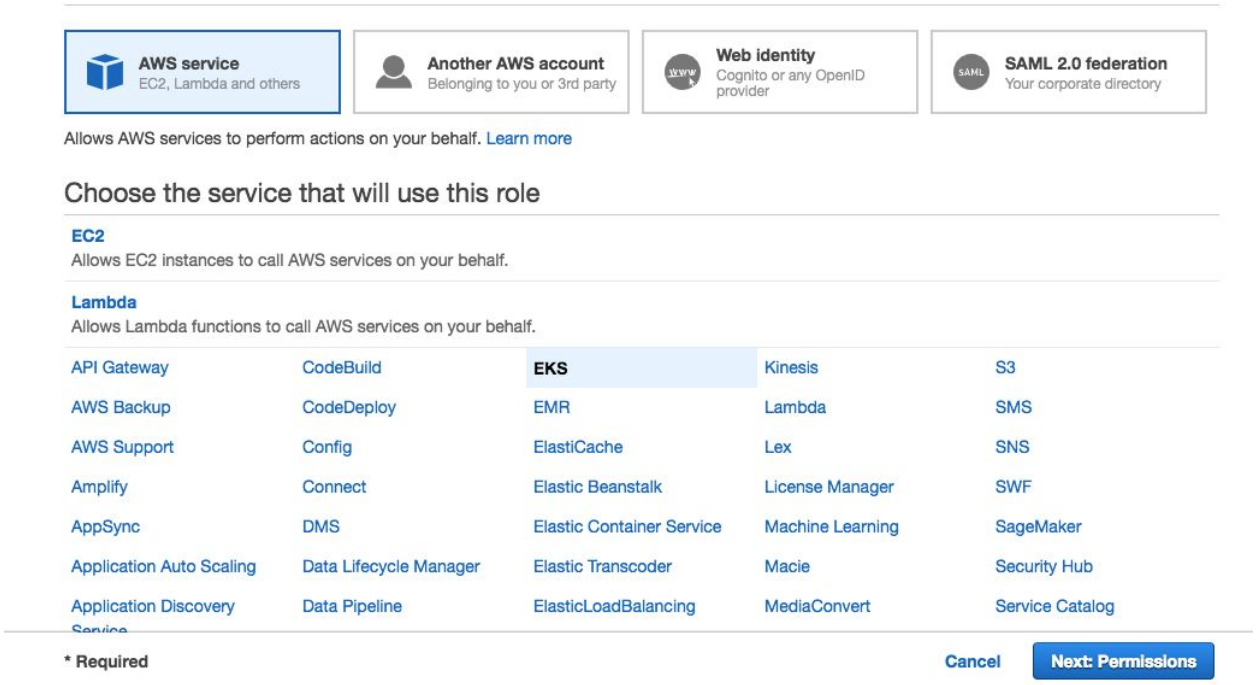
Goto aws aws services and search IAM.



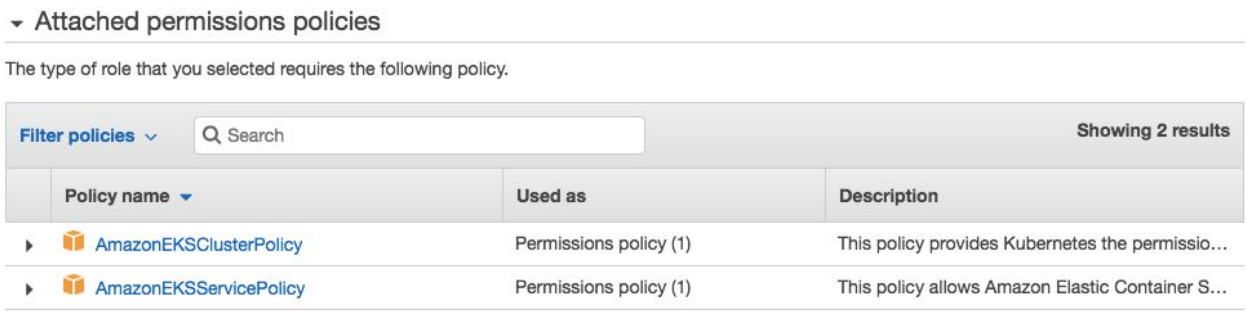
After enter in IAM roles in left side you will get **Roles** menu, click on this you will get windows like this.



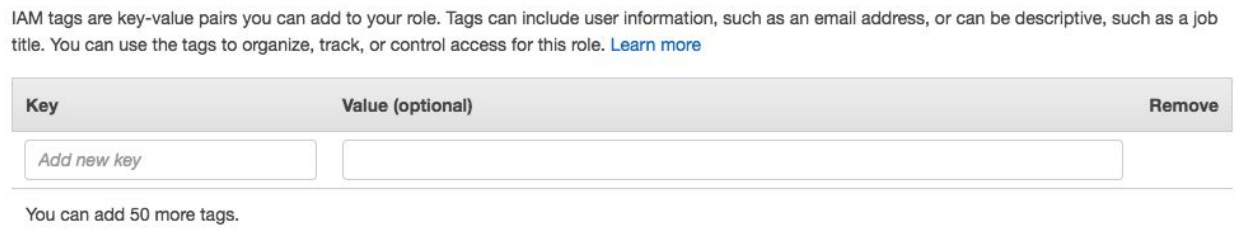
Then click on Create role then select **EKS** get a window like



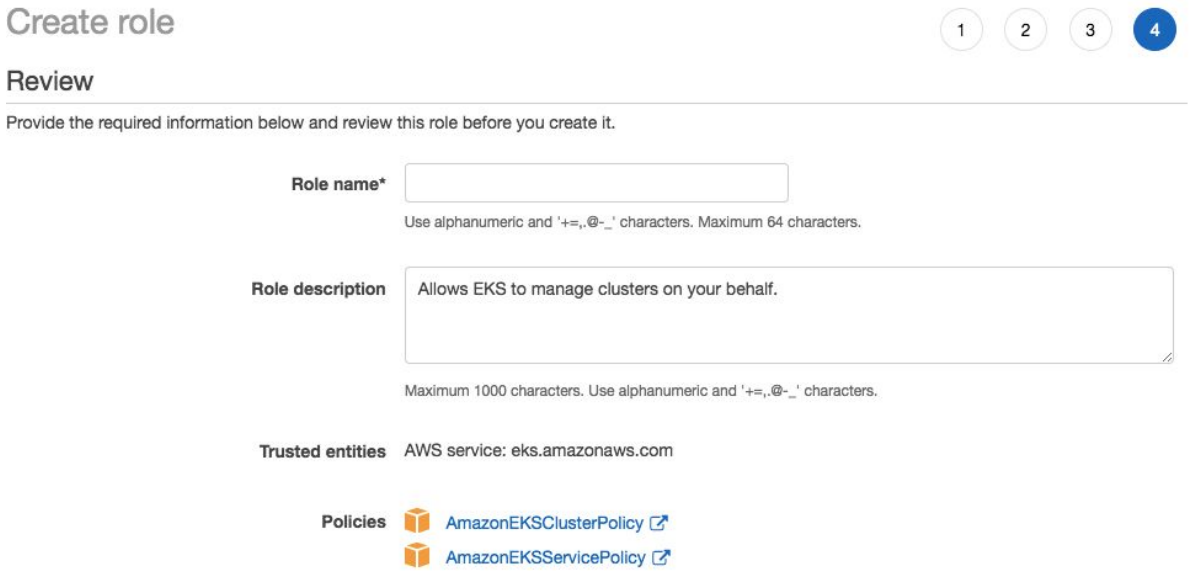
Click on **Next**: Permission get like



Then click on **Next:Tags** at rigth bottom will get like



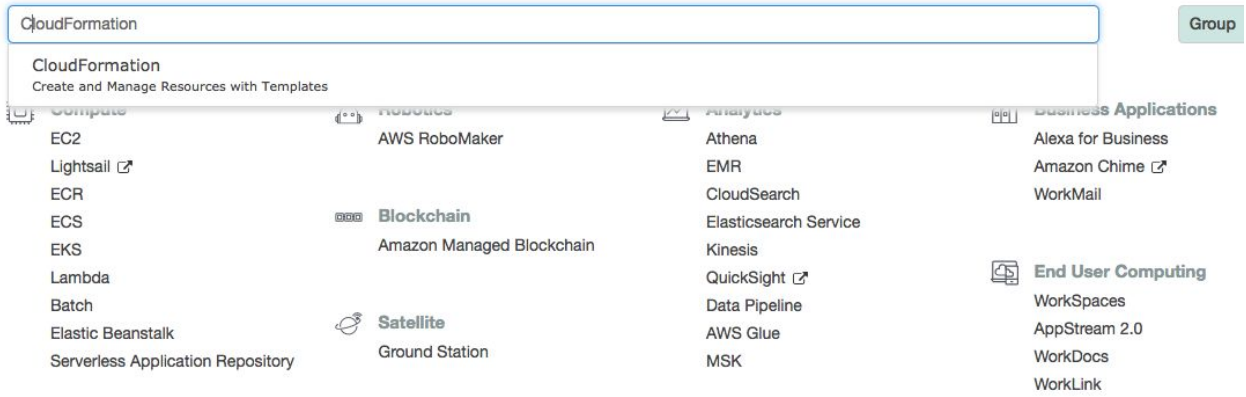
Put any key and value you want, by the way its optional you may leave this.
Now click on right bottom **Next: Review**



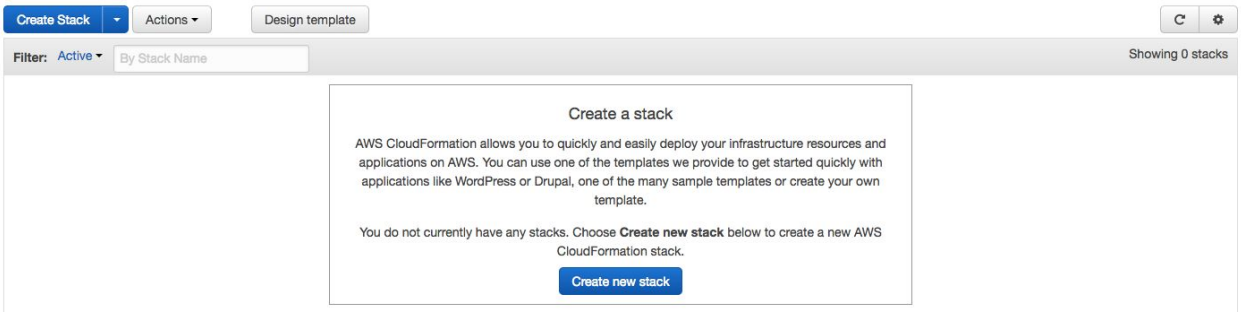
Put the name as you wish, i put **eks-role** here. Now click on **Create role** .
You IAM role created now.

Step2: Now create VPC and security groups using cloud formation templates.

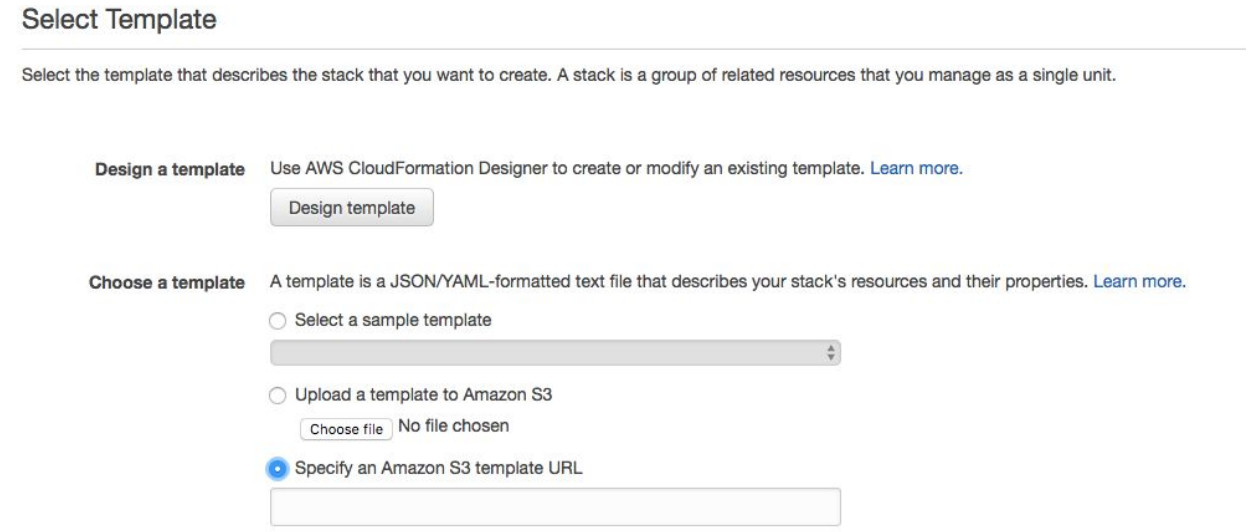
Goto aws services and search cloud formation



Click on CloudFormation

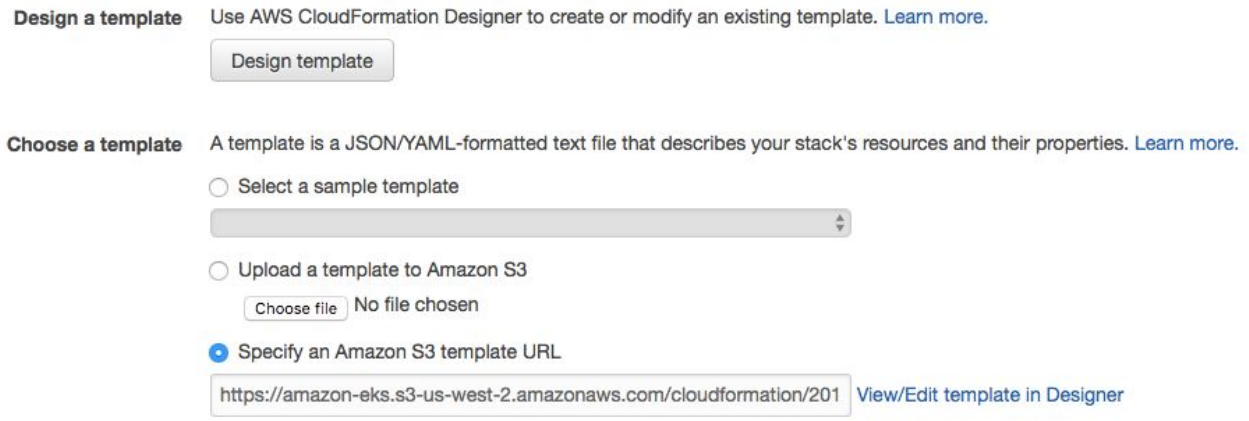


Now click on Create Stack

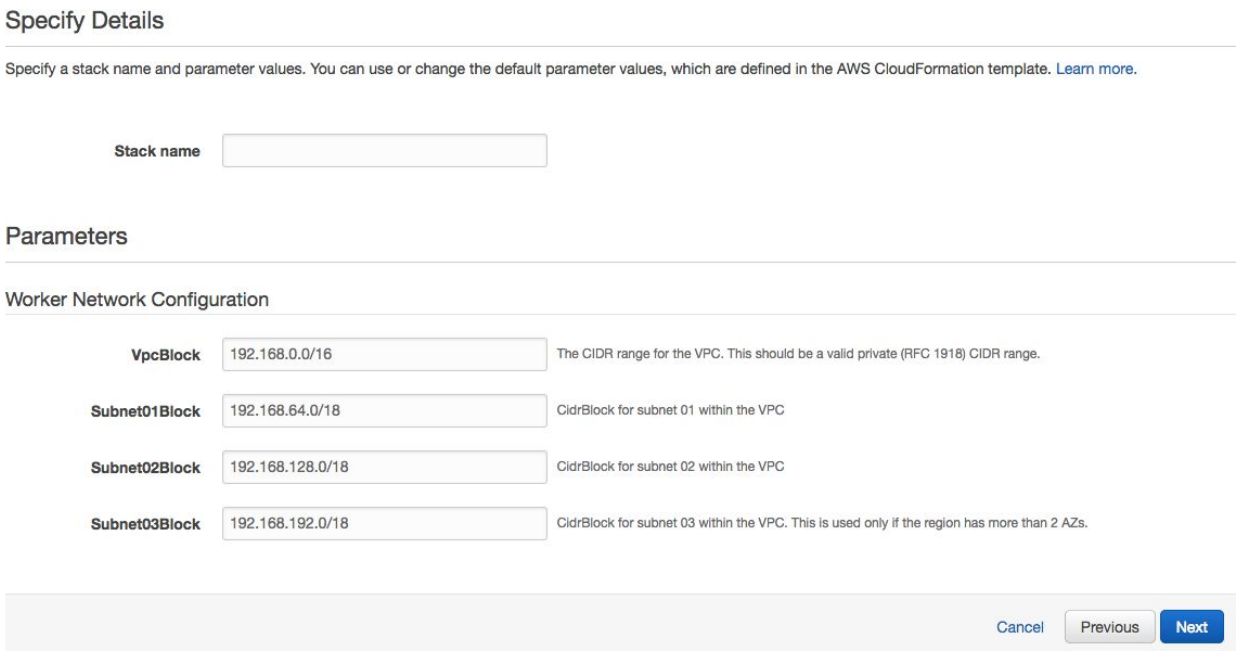


Now select the last option(Specify an Amazon S3 template) and copy below link

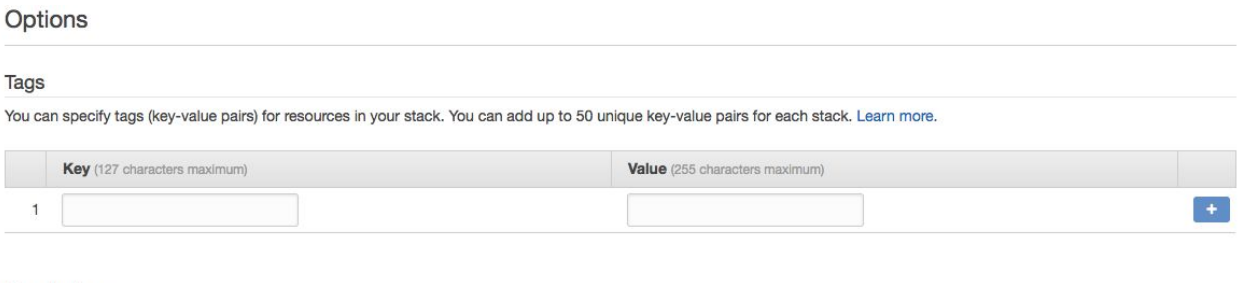
https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-02-11/amazon-eks-vpc-sample.yaml



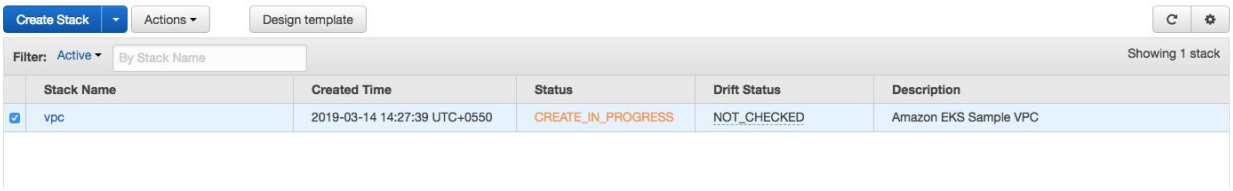
Now click on **Next**



Now put the stack name here i have put vpc for me.



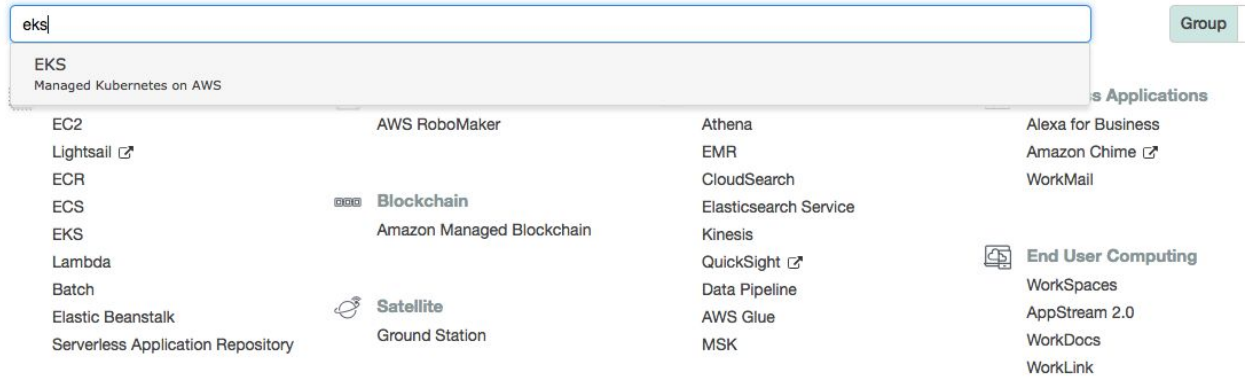
Put key and value here its optional, now click right bottom **Next** get Review window then click on **Create** Button. It will look like



It will take few minutes to complete.

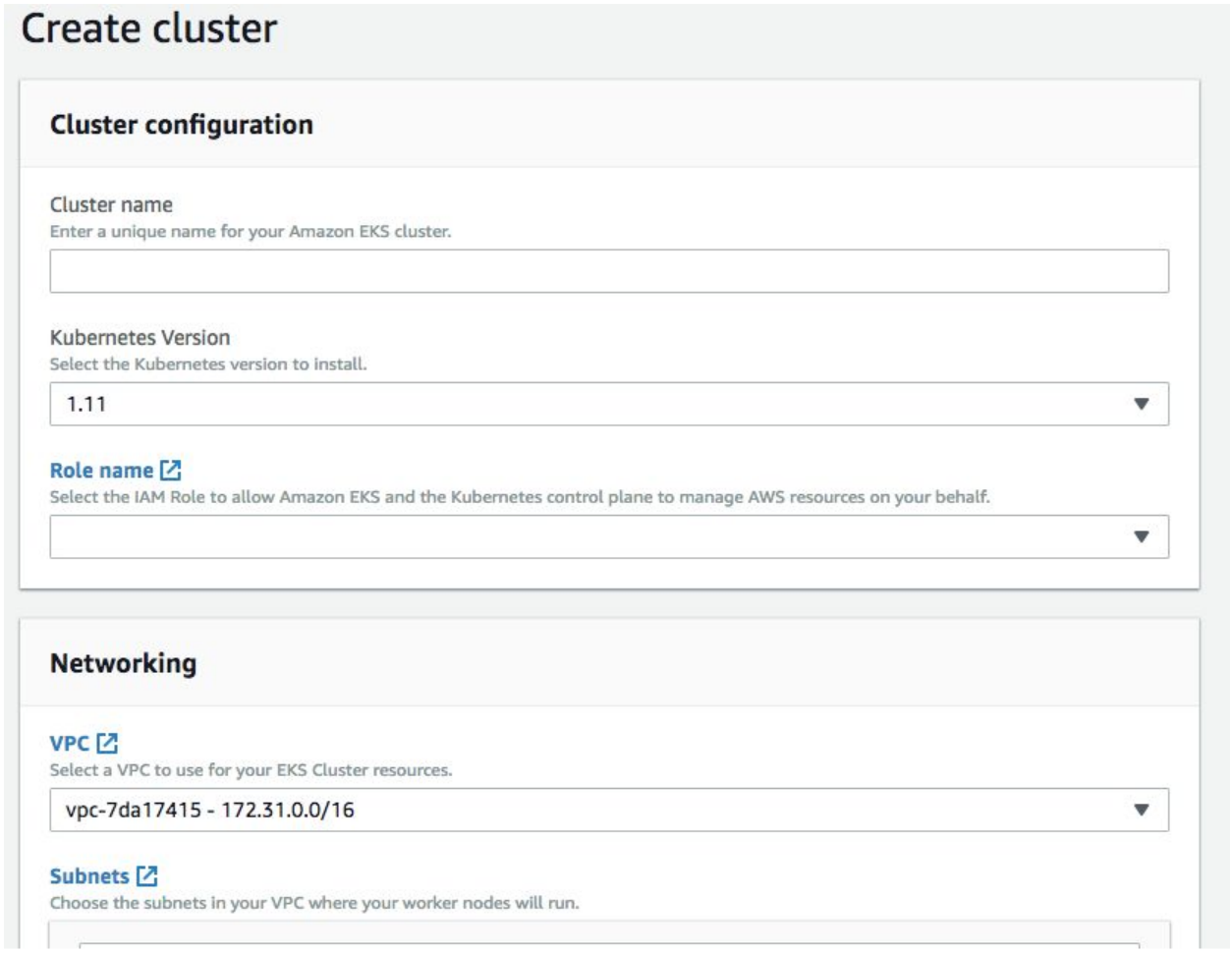
Step3: Create EKS cluster

Goto aws services and search **eks**



Inside EKS gets like

Click on Next step,

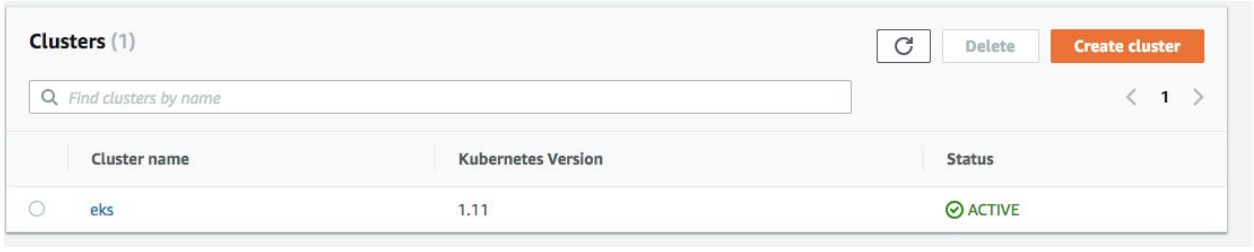


Put the following fields

1. Cluster name
2. Role name(we have created earlier)
3. VPC(we have created earlier)
4. Security Group(not default)

Then click on bottom right Create

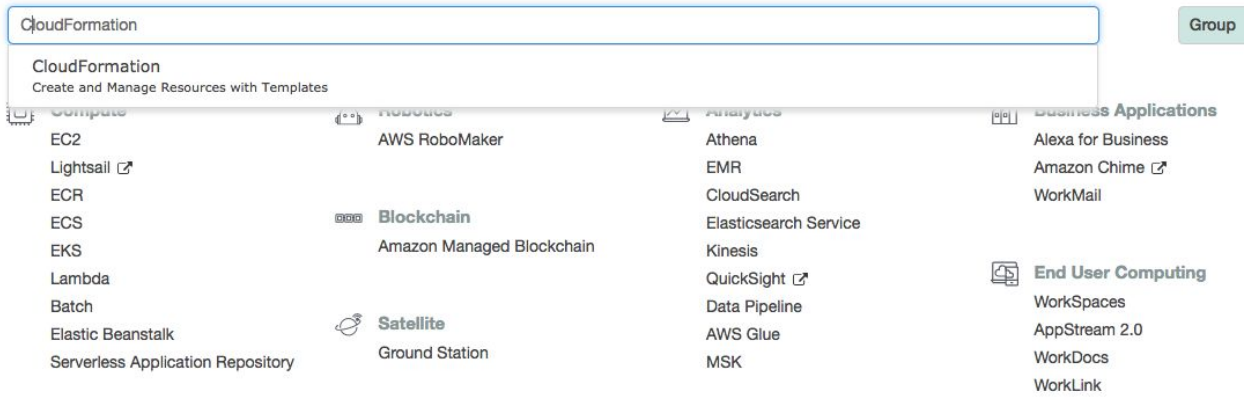
It will take some time, after creation it will like



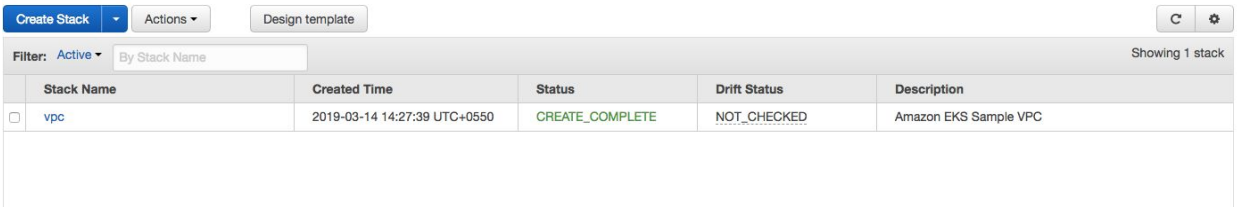
Here eks is the cluster name.

Step4: Now create worker nodes using cloud formation templates.

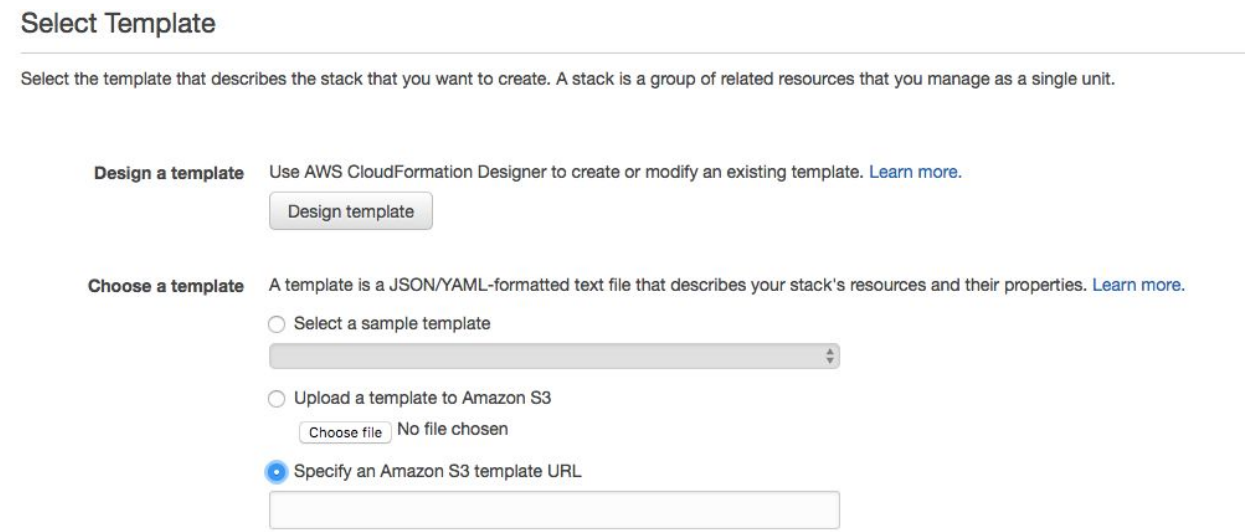
Goto aws services and search cloud formation



Inside cloudformation you will get like.

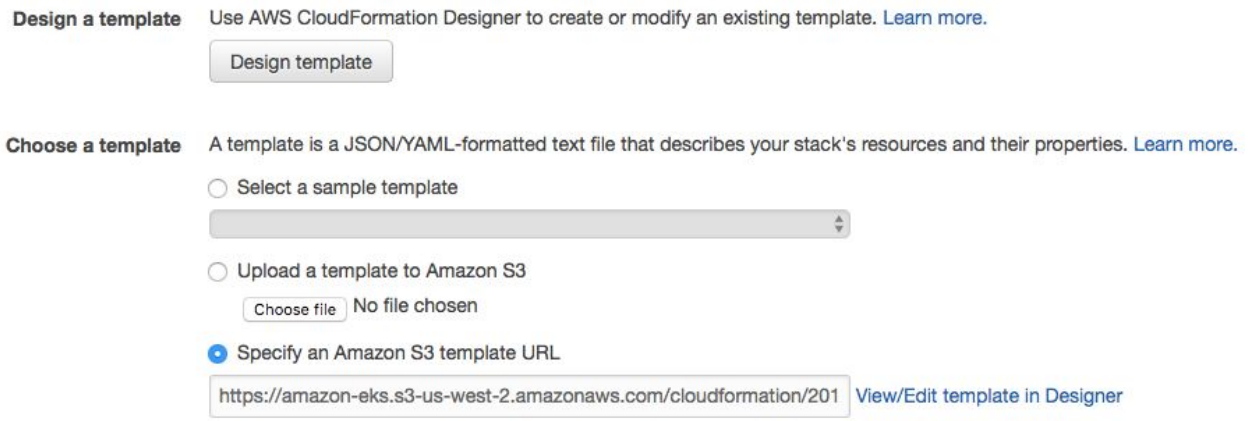


This vpc is what we have created earlier now have to create worker nodes.
Click on Create Stack

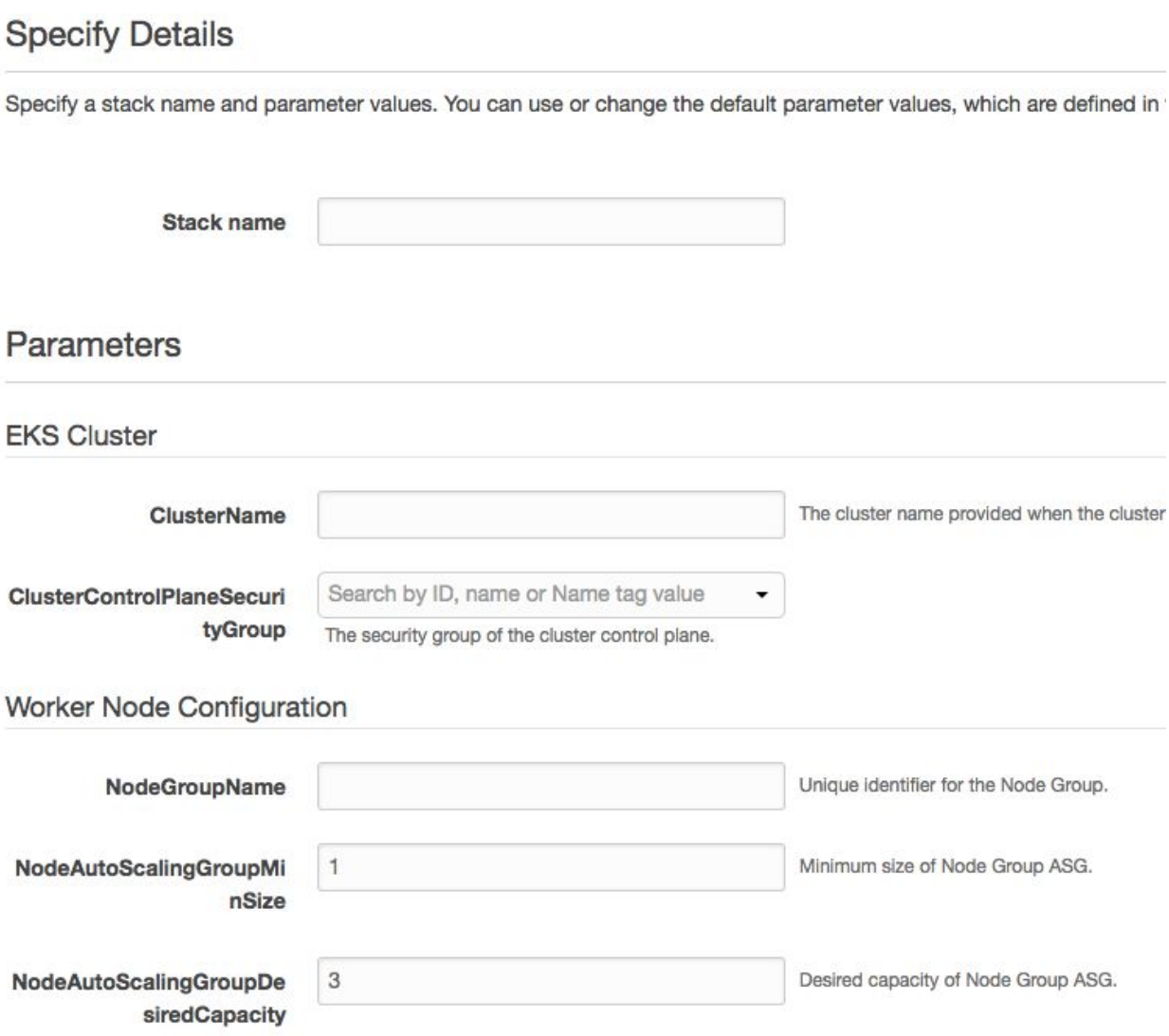


Now select the last option(Specify an Amazon S3 template) and copy below link
https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-02-11/amazon-eks-nodegroup.yaml

after put this link it will like



Now click on Next Button , see

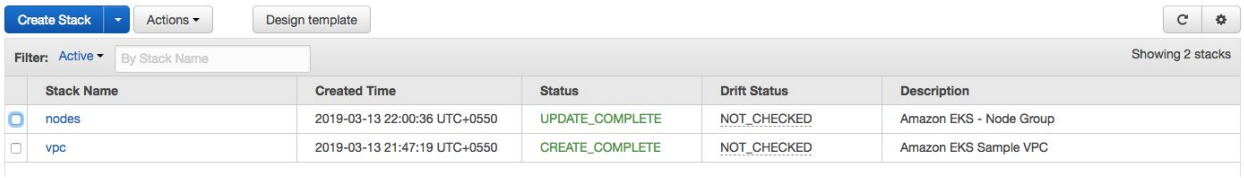


Put the following values in the following fields.

1. Stack name
2. ClusterName (cluster name created earlier)
3. NodeGroupName
4. NodeAutoScalingGroupMinSize(1)
5. NodeAutoScalingGroupDesiredCapacity(2)
6. NodeAutoScalingGroupMaxSize(3)
7. NodeImageId(`ami-0eeeef929db40543c`)
8. KeyName
9. VpcId(created earlier)
10. Subnets

After fulfill the above details click on **next** button.

Put the key & value its opetional then click Next button. Then it will show you all resources that will create.
Now click on right bottom **Create** button.
Its creating your resources, it will take some time.



Now all task has finished on aws

Now setup your local machine,

1. Aws cli
2. Kubectl (https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html)
3. Aws authenticators (https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html)

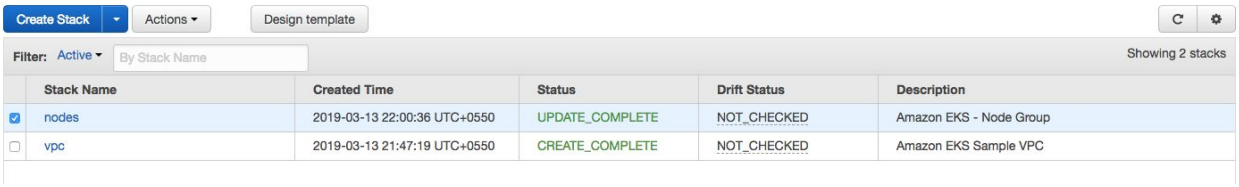
Now download the following

curl -O
https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-02-11/aws-auth-cm.yaml

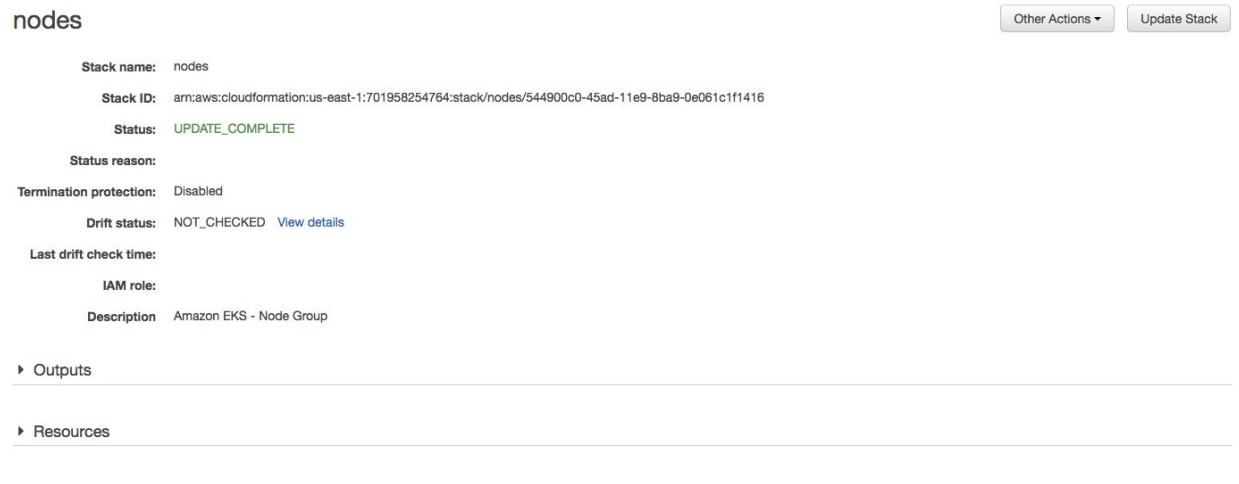
And have to edit this, its see like

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    - rolearn: <ARN of instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
~
~
```

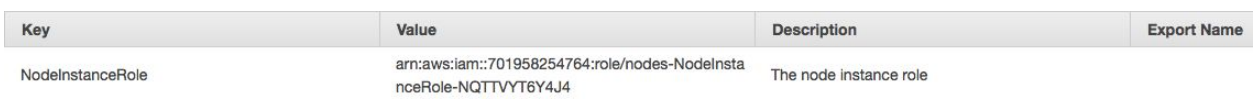
We have to replace value of - rolearn:



Click on node



Now click on output



Copy the value and past it in aws-auth-cm.yaml file, we have edit earlier.

This file will look like.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    - rolearn: arn:aws:iam::23858473249:role/nodes-NodeInstanceRole-dhsgtbbsw
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
```

Save this file.

Run the following commands

$ aws eks --region us-east-1 update-kubeconfig --name eks
$ kubectl apply -f aws-auth-cm.yaml

Now all things are setup

Now get the nodes using

$ kubectl get nodes

```
ubuntu@ip-172-31-26-223:~/msrcosmos$ kubectl get nodes
NAME                                        STATUS    ROLES     AGE   VERSION
ip-192-168-118-23.ec2.internal    Ready     <none>    17h   v1.11.5
ip-192-168-188-194.ec2.internal   Ready     <none>    17h   v1.11.5
ubuntu@ip-172-31-26-223:~/msrcosmos$
```

Step5:

1. **Tomcat deployment**: for tomcat deployment we have two file **tomcat-deployment.yaml** & **tomcat-service.yaml** i have shared these files on github also,

tomcat-deployment.yaml is kubernete object of type deployment, by this file we will launch an application of tomcat.

tomcat-service.yaml is kubernete object type of service define the logical setup of pods of tomcat application In cluster.

Run the following commands for Tomcat deployment
$ kubectl create -f tomcat-deployment.yaml
$ kubectl create -f tomcat-service.yaml

2. CouchDB deployent: for couchdb deployment we have two file **couchdb-deployment.yaml** & **couchdb-service.yaml** i have shared these files on github also,

couchdb-deployment.yaml is kubernete object of type deployment, by this file we will launch a couchdb application.

couchdb-service.yaml is kubernete object type of service define the logical setup of pods of couchdb application In cluster.

Run the following commands for CouchDB deployment

$ kubectl create -f couchdb-deployment.yaml
$ kubectl create -f couchdb-service.yaml

Now get services using

$ kubectl get services -o wide

Its looks like

```
service/tomcat-pod created
ubuntu@ip-172-31-26-223:~/msrcosmos$ kubectl get services -o wide
NAME          TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)          AGE   SELECTOR
couchdb-pod   LoadBalancer   10.100.187.107   a1f1e2ff4462811e996b60282e867f49-181338381.us-east-1.elb.amazonaws.com     5984:31757/TCP   4h    run=couchdb-pod
kubernetes    ClusterIP      10.100.0.1       <none>                                                                     443/TCP          19h   <none>
tomcat-pod    LoadBalancer   10.100.67.3      a4c437b30464c11e996b60282e867f49-2072243459.us-east-1.elb.amazonaws.com    80:30916/TCP     1m    run=tomcat-pod
ubuntu@ip-172-31-26-223:~/msrcosmos$
```

Copy the external-IP link for Tomcat like this with, its running on port 80

a4c437b30464c11e996b60282e867f49-2072243459.us-east-1.elb.amazonaws.com

Now you will see Tomcat application

Now external-IP link for couchdb like this with port 5984

a1f1e2ff4462811e996b60282e867f49-181338381.us-east-1.elb.amazonaws.com
:5984/_utils

Put the following credentials
User: msrcosmos
Pass: msrcosmos