**AWS EKS and Kubernetes External DNS**

**Pre-Requisites:**

* To be Installed EKS Cluster with nodes
* Domain-Name

**Step1:**

* Create a Policy and Attached to Role which is created automatically while doing worker nodes stack at the time of EKS cluster
* Go to IAM 🡪 Policies 🡪 Create Policy

Click on Json and Replace with bellow data

----------------------------------------------------------------------------------------------------------------

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"route53:ListHostedZones",

"route53:ListResourceRecordSets"

],

"Resource": [

"\*"

]

},

{

"Effect": "Allow",

"Action": [

"route53:ChangeResourceRecordSets"

],

"Resource": [

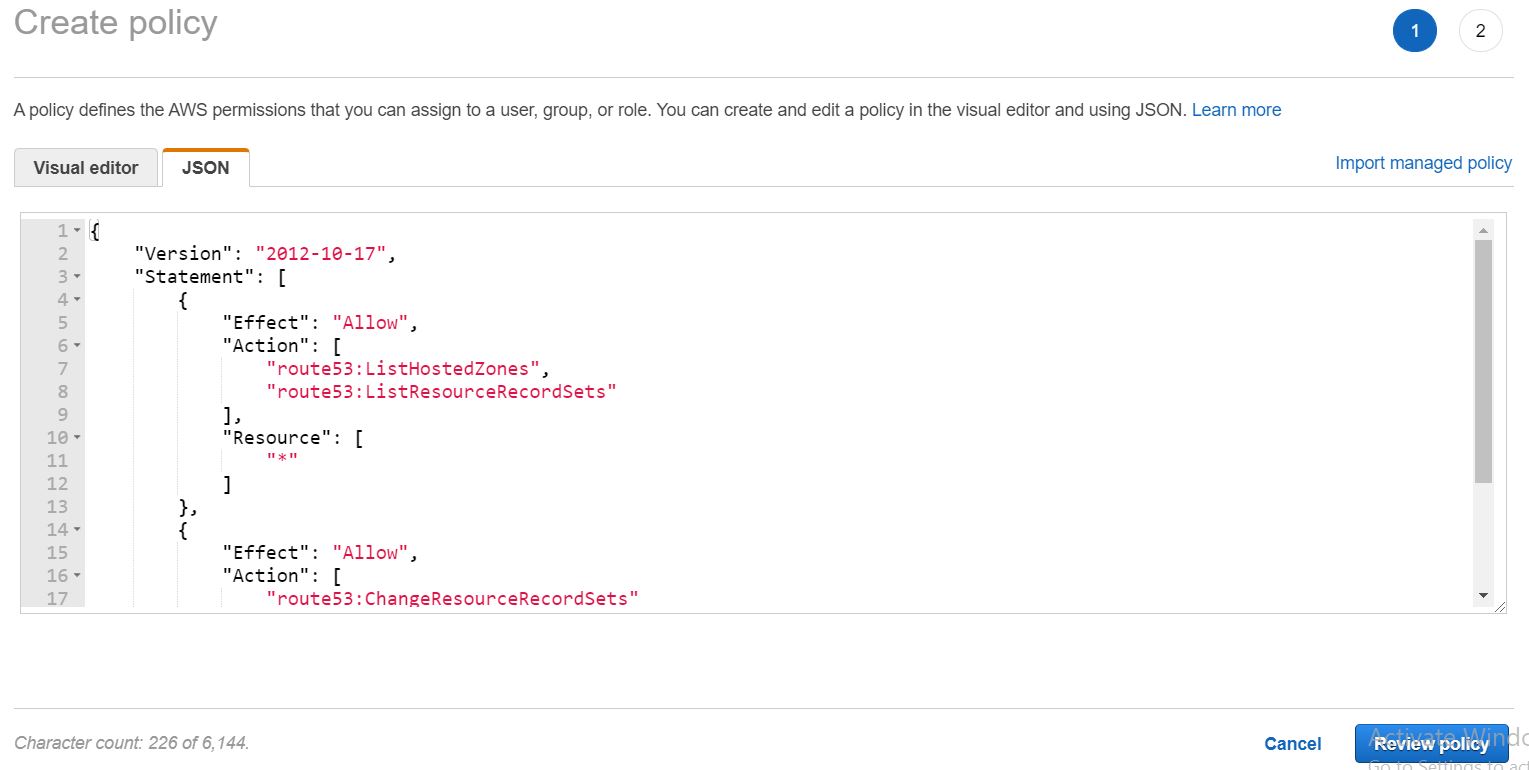
"\*"

]

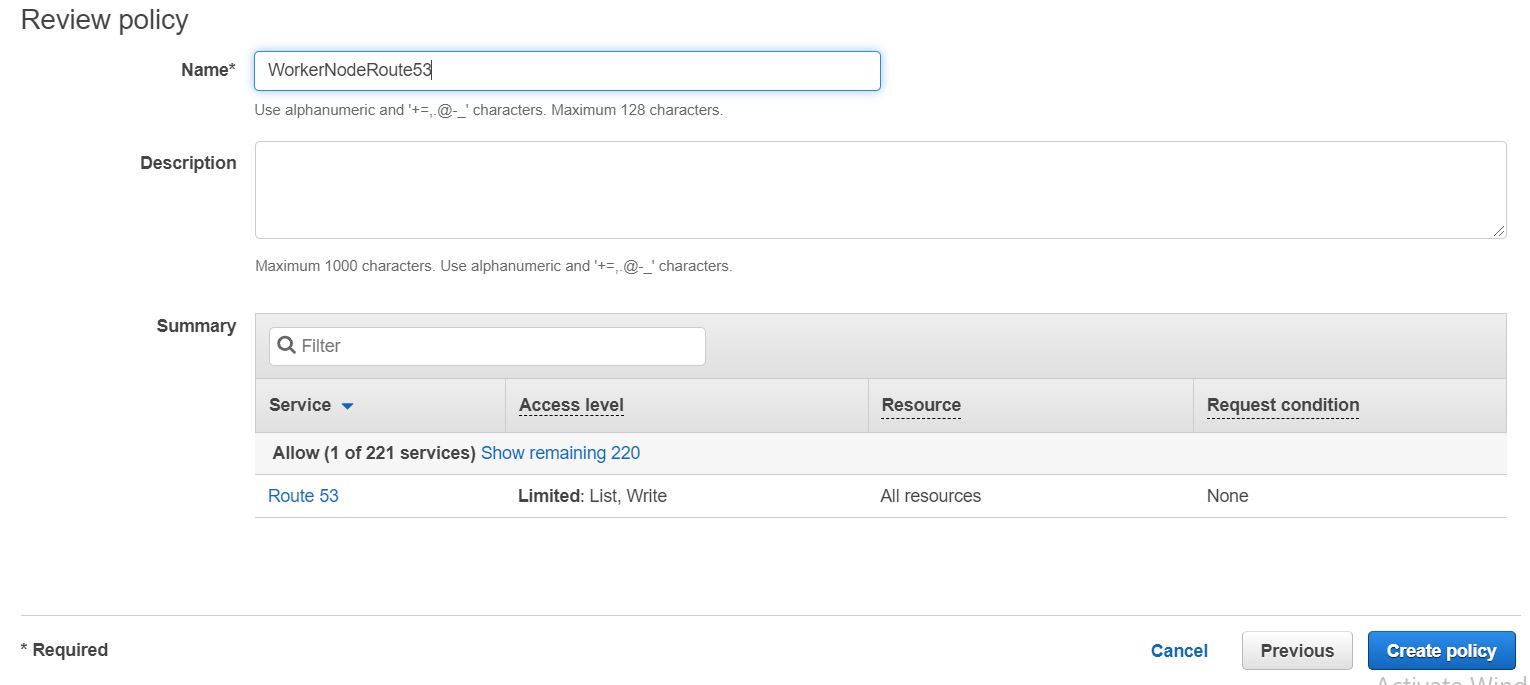
}

]

}



* Click on Review Policy



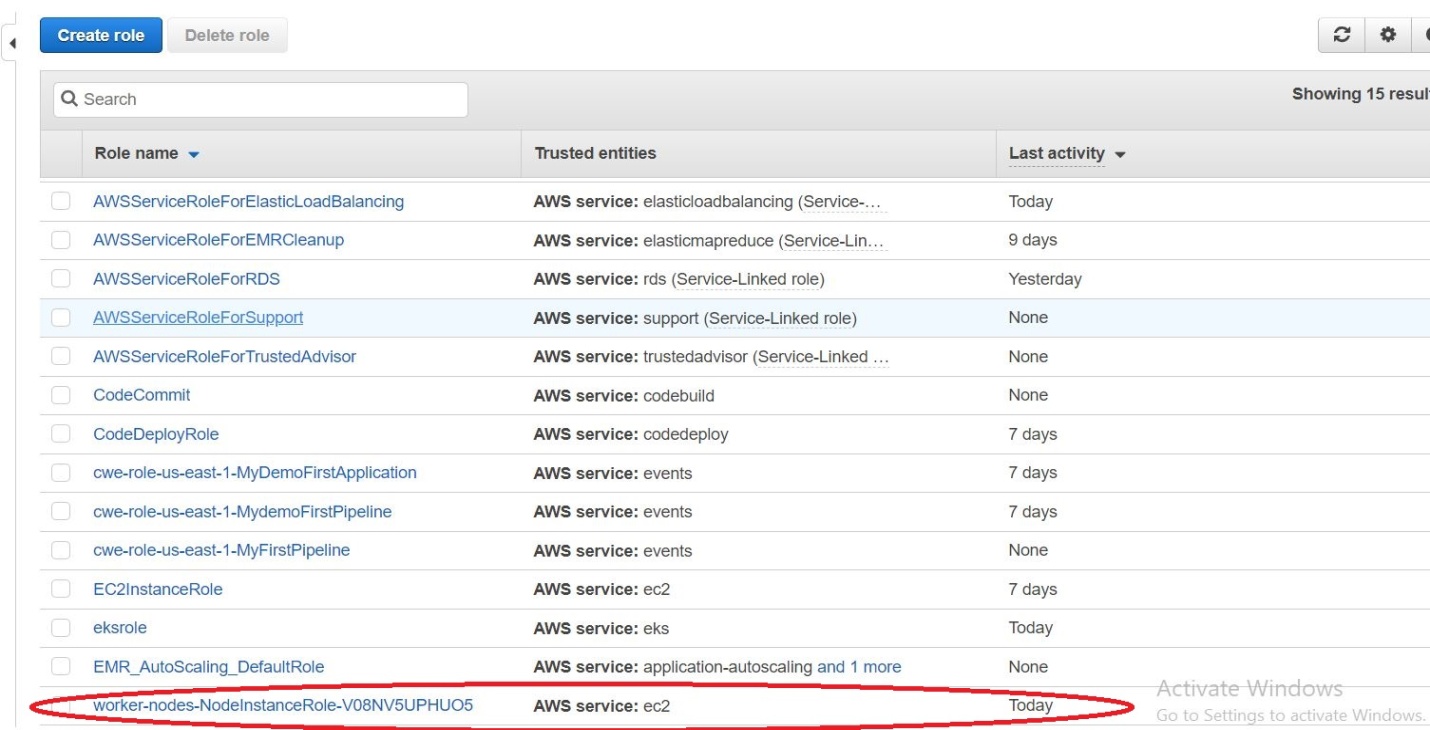
* Click on Create Policy

**Step-2:**

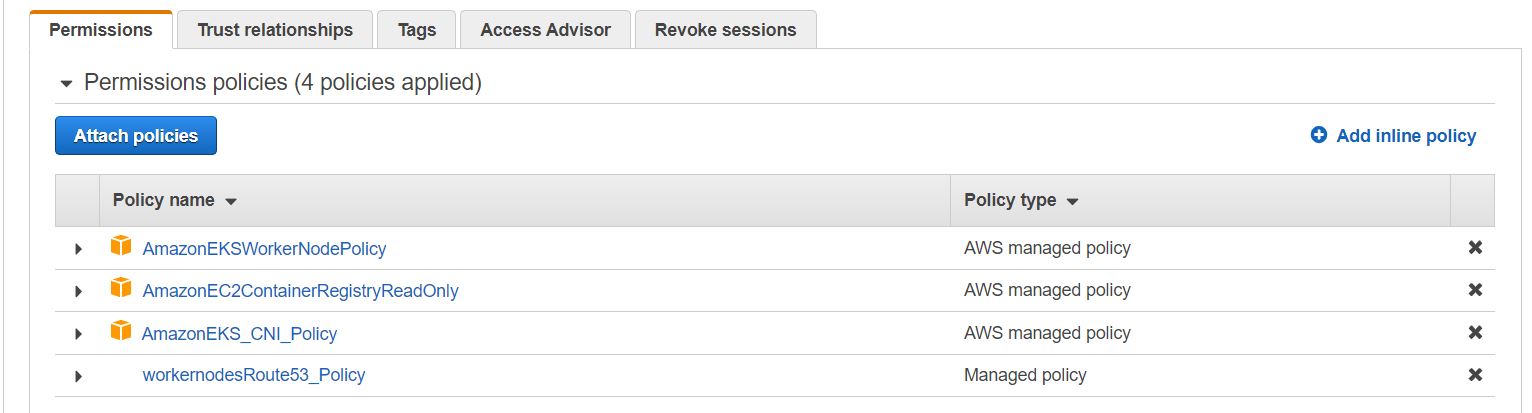
Goto Roles: IAM 🡪 Roles

Find Role which is created at the time of Worker node stack creation

Here I am showing Role which is created at the time of my worker node stack



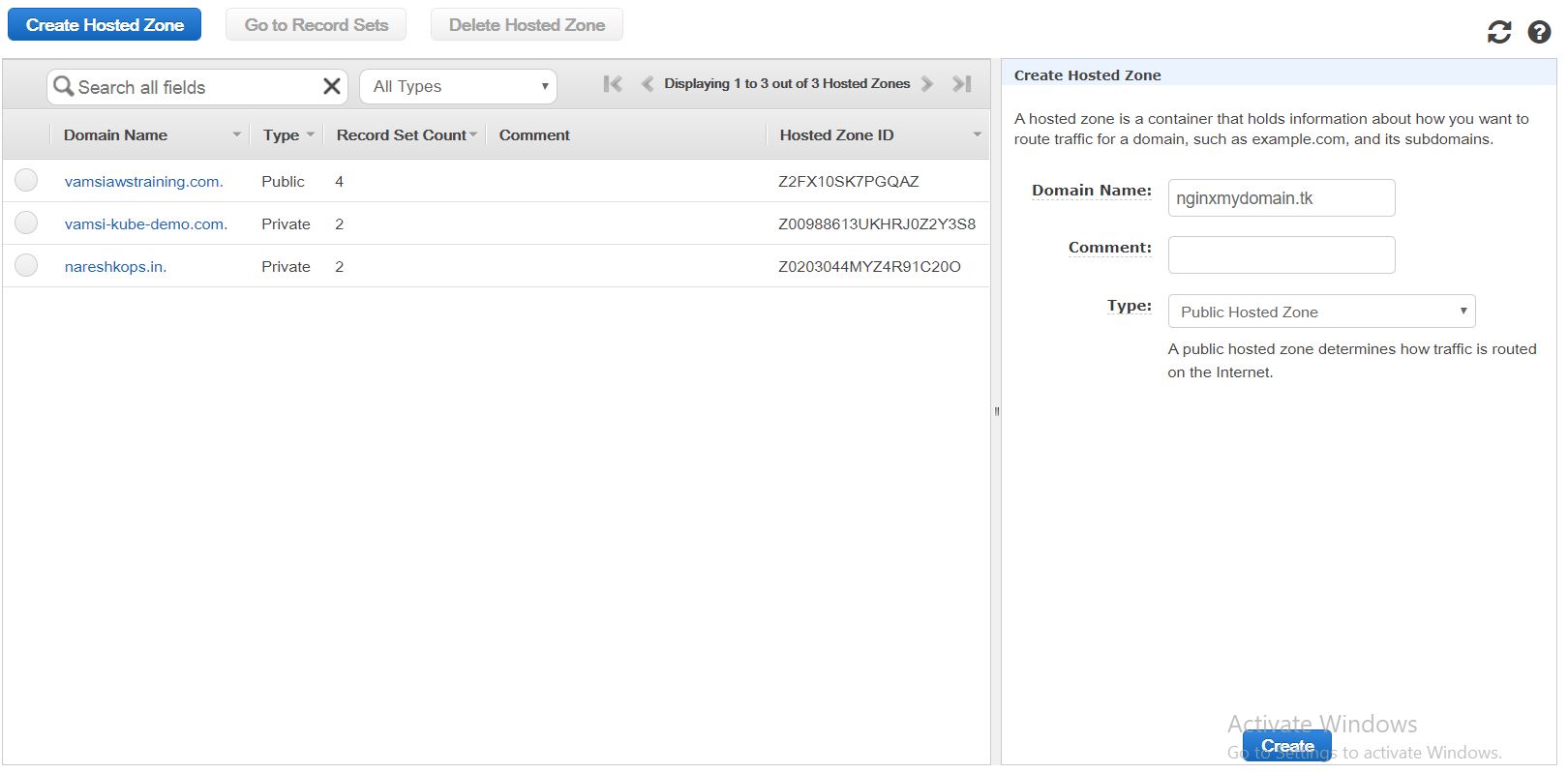
* Click on the Role and attach policy which is created above

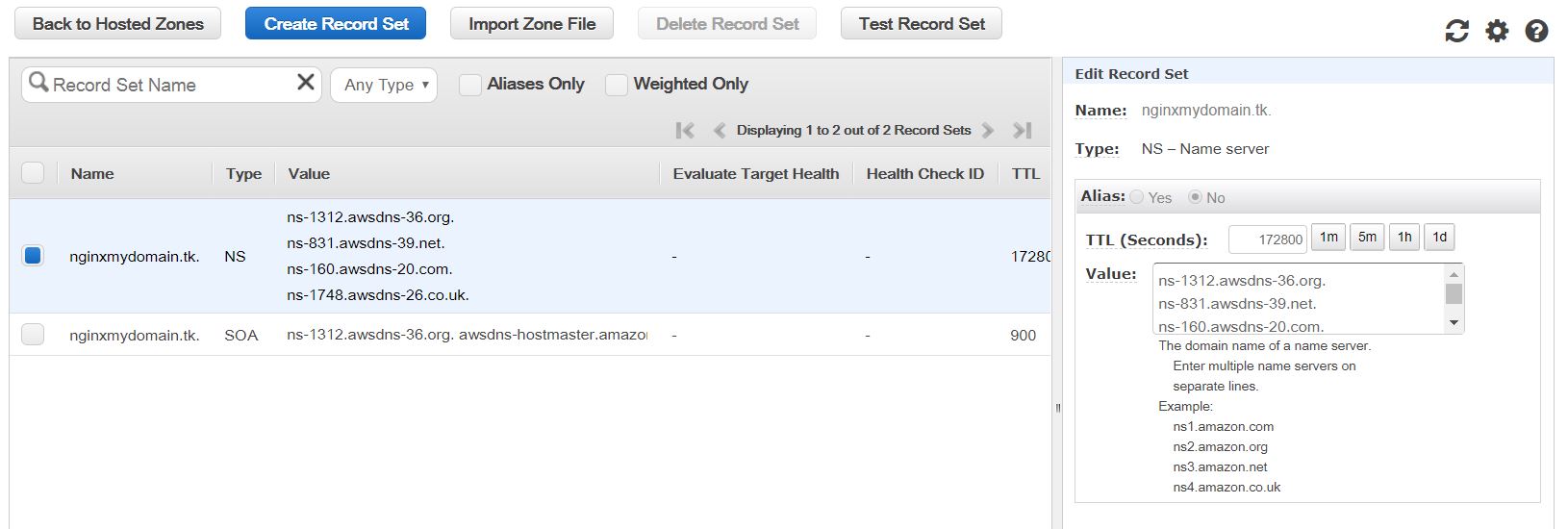


**Step3:**

Goto Route53 and create Hosted Zone

Click on Create Hosted Zone with your Domain name



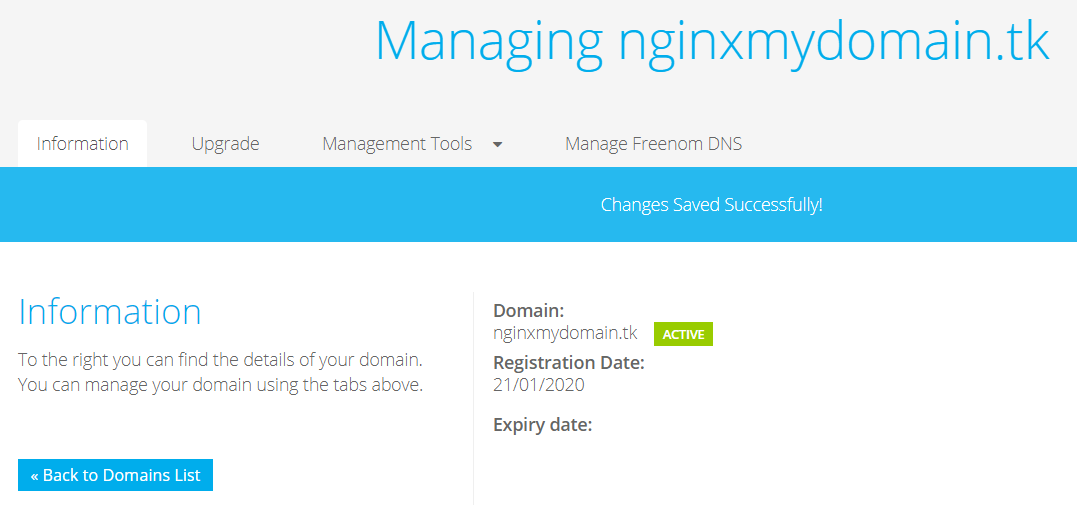


Once you created Hosted Zone we get Name servers (Shown in above picture)

These Name Servers to be added where we purchased Domain

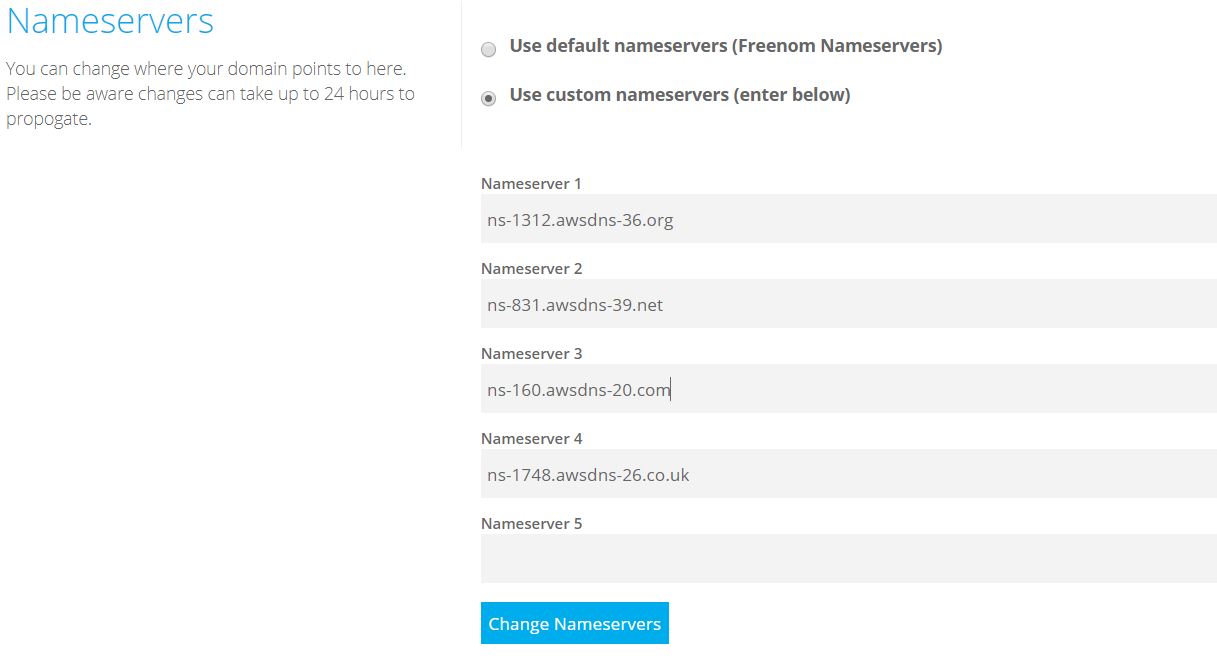
Goto Freenom Account: Open our Domain

Here I get Domain in Freenom account



Click on Manage Tools 🡪 Name Servers

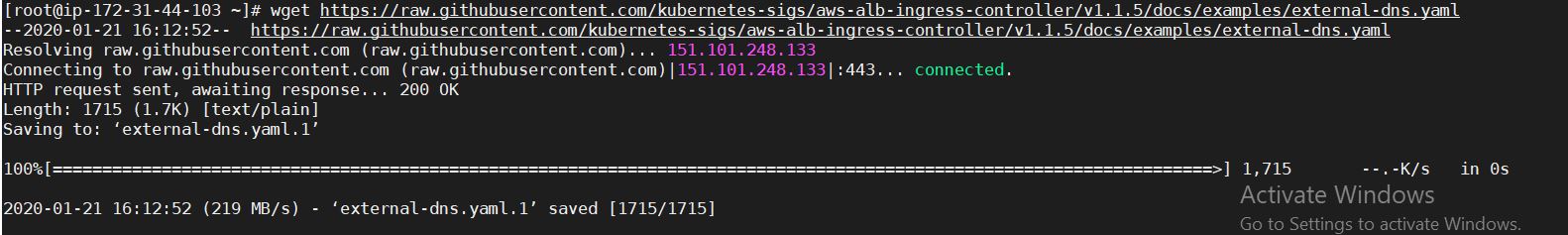
And paste Nameservers Here and click on Change Name Servers



**Step4:**

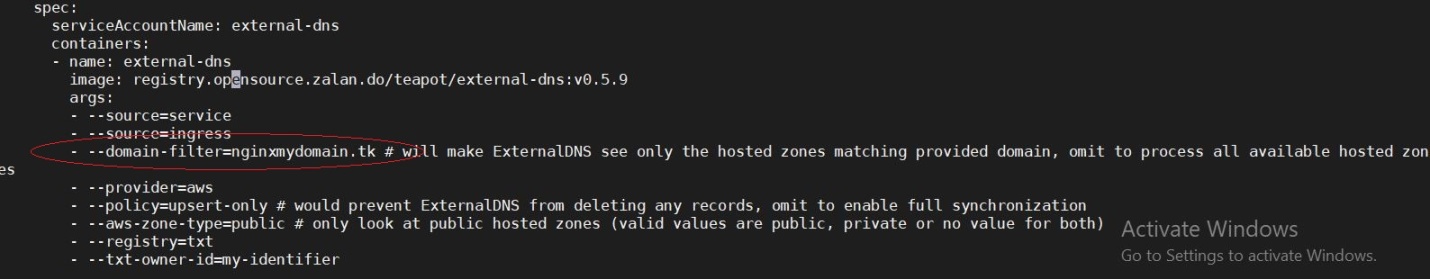
Now Login to Kubernates Master and Download sample external-dns manifest

wget <https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-ingress-controller/v1.1.5/docs/examples/external-dns.yaml>



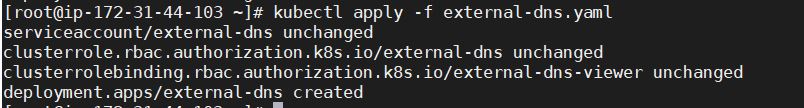
* Open external-dns.yaml file and Replace --domain-filter=<our Domain-Name>

vi external-dns.yaml



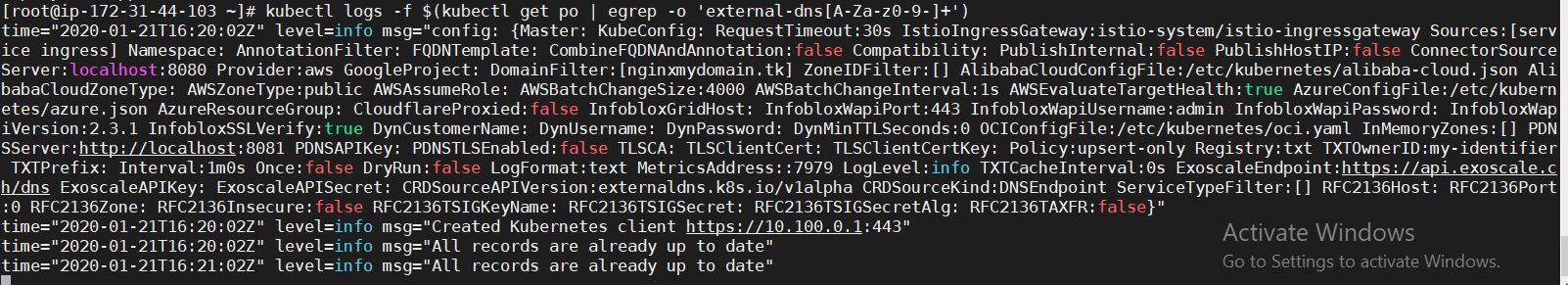
* Now Deploy external-dns

kubectl apply -f external-dns.yaml



* Verify it deployed successfully or not

kubectl logs -f $(kubectl get po | egrep -o 'external-dns[A-Za-z0-9-]+')



**Step5:**

Here I am deploying nginx to check whether records will be add or not for our Hosted Zone

Create nginx.yaml file with below data

vi nginx.yaml

apiVersion: v1

kind: Service

metadata:

name: nginx

annotations:

external-dns.alpha.kubernetes.io/hostname: nginx.nginxmydomain.tk.

spec:

type: LoadBalancer

ports:

- port: 80

name: http

targetPort: 80

selector:

app: nginx

---

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: nginx

spec:

template:

metadata:

labels:

app: nginx

spec:

containers:

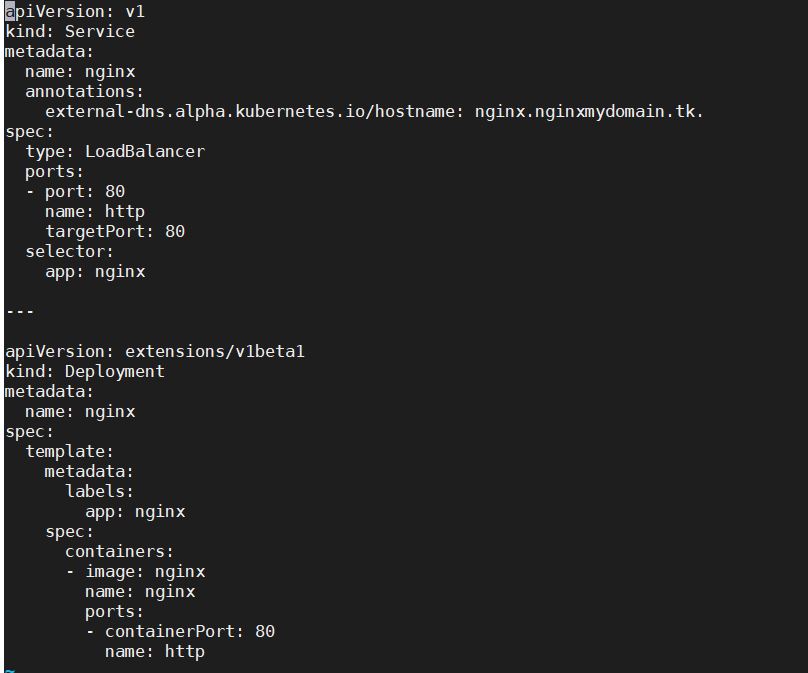
- image: nginx

name: nginx

ports:

- containerPort: 80

name: http



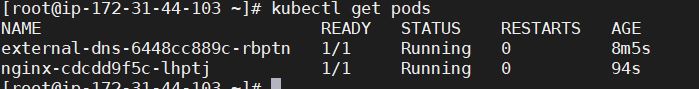
* Now Deploy nginx.yaml

kubectl apply -f nginx.yaml

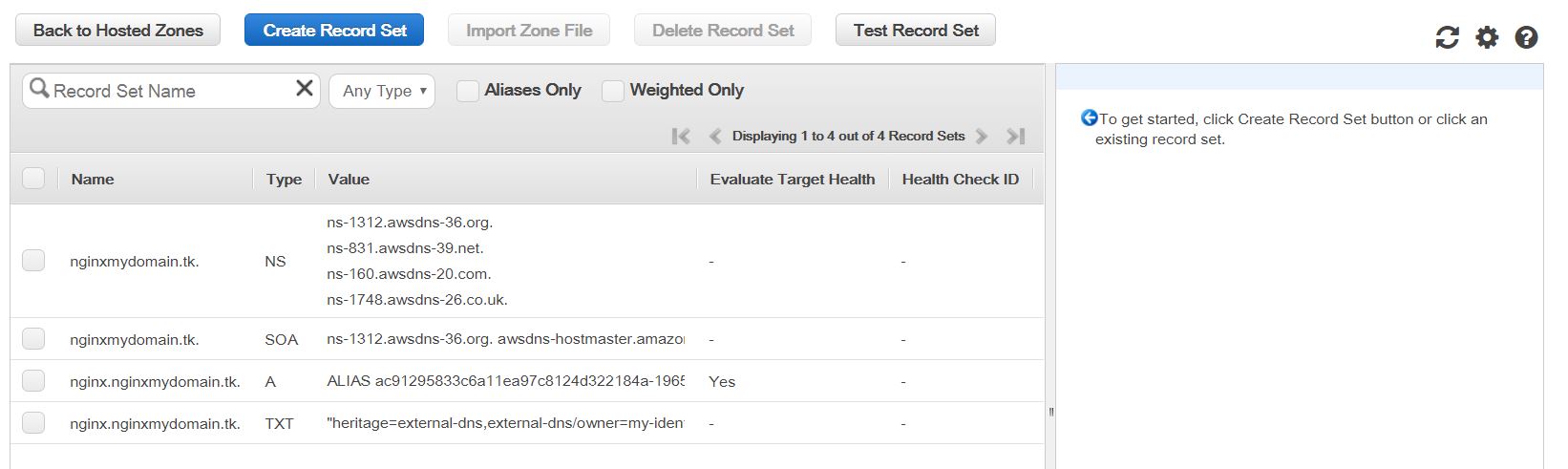
D:\Naresh\snipping\Capture.JPG

* Check pods are Running or not

kubectl get pods

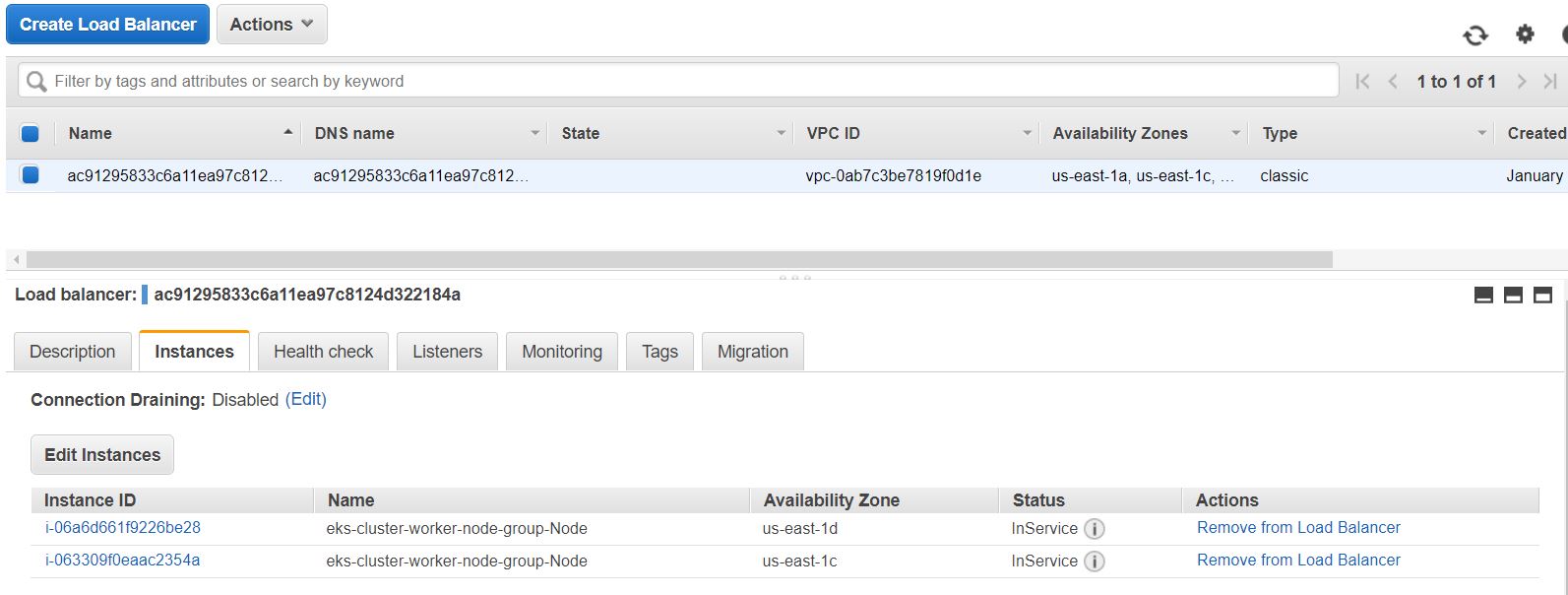


* Goto Route53 and check Record sets are added to our Hosted Zone or not



Here 2 record sets are added

* Goto LoadBalancer Tab and check Instances comes into InService or not



* Now check our Output With our Domain Name:

<http://nginx.nginxmydomain.tk/>

