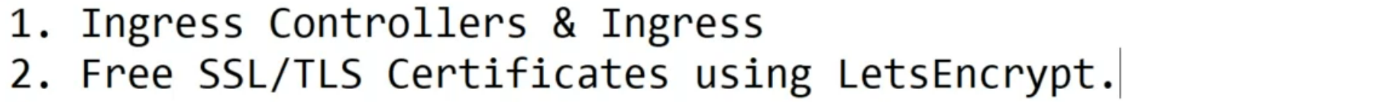
**7.IngressControllers-LetsEncrypt**

--- in this session, we will discuss about below things.

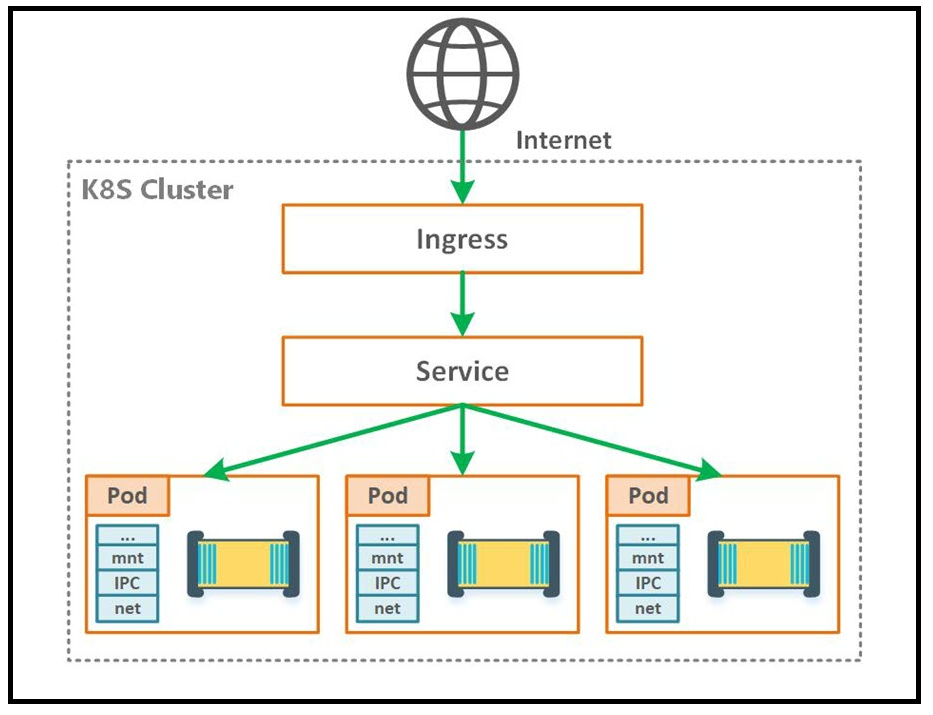


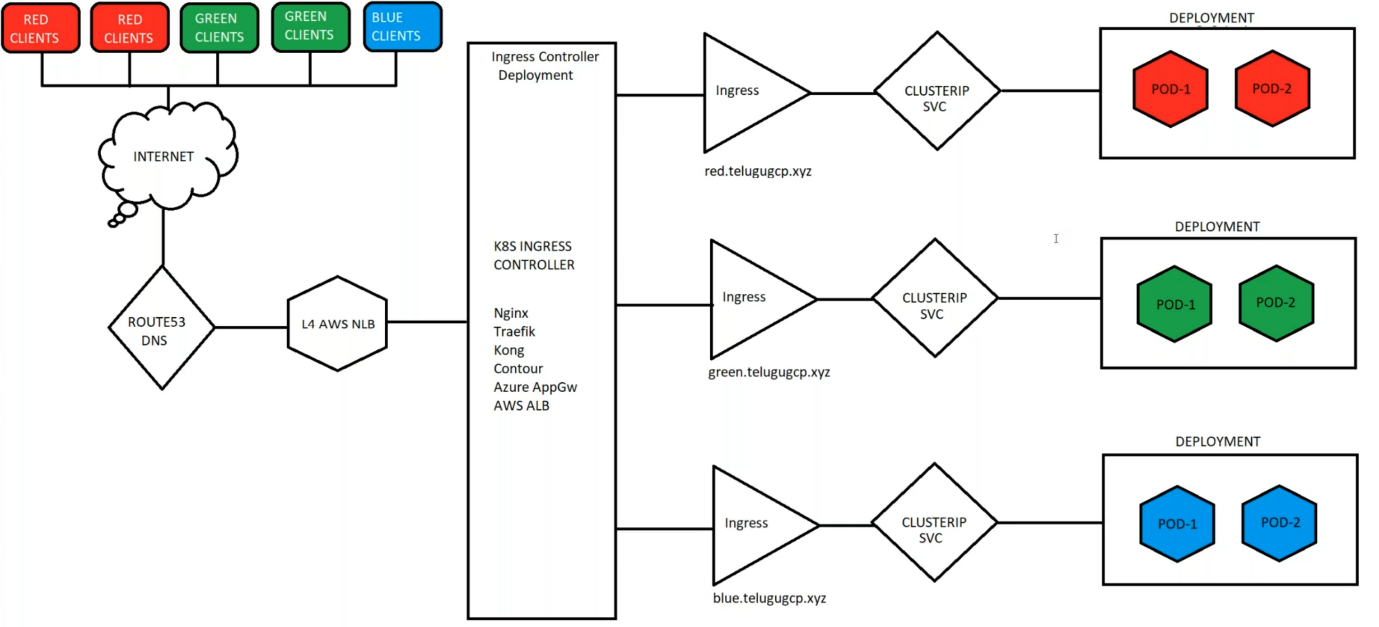
--- if you want to use load balancer then you will face some problems, you cannot use more than 50 load balancers. If you use load balancer for each service then it will cost you more money. To overcome this problem, we will use ingress controller’s concept.

--- **what is the difference between ingress controllers and ingress…?**

--- the traffic come outside to inside is called ingress, this process controlled by ingress controller.

--- based host name and ssl certificate, to which host the traffic must be sent to decide by ingress.





--- **scenario** – we have red, green and blue deployments. Imagine theses are 3 different deployments and we are exposing the deployments using clusterIP service. After this we will deploy one ingress controller, there are different types of ingress controllers

They are 1. Nginx

2. Traefik

3. kong

4. contour

5. Azure AppGw

6. AWS ALB

--- Here, the traffic from internet reaches to route53 and from there it reaches to l4 network load balancer. Based on host name, the traffic will be routed to required nodeport service.

--- **note** – this way, you can use single load balancer for multiple applications.

--- **note** – ingress controller is just like deployment and inside of this deployment, there will be multiple pods running. Ingress is just like NodePort service.

**Pre-requisites**

--- I will create red, green, sreek8s and blue deployments.

--- I will expose these deployments using clusterip service.

--- I will create ingress controller using deployment, it will create ingress-nginx name space and ingress pods in the name space, the deployment will also create network load balancer in the aws. You don’t have to create it separately.

**Red deployment**

# Create red-deploy deployment using kubectl command

--- kubectl create deploy red-deploy --image=sreeharshav/rollingupdate:v1 --replicas 3

# Expose deployment using clusterip service

--- kubectl expose deploy red-deploy --port=8000 --target-port=80

**Green deployment**

# Create green-deploy deployment using kubectl command

--- kubectl create deploy green-deploy --image=sreeharshav/rollingupdate:v5 --replicas 3

# Expose deployment using clusterip service

--- kubectl expose deploy green-deploy --port=8000 --target-port=80

**Blue deployment**

# Create green-deploy deployment using kubectl command

--- kubectl create deploy blue-deploy --image=sreeharshav/testcontainer:v1 --replicas 3

# Expose deployment using clusterip service

--- kubectl expose deploy blue-deploy --port=8000 --target-port=80

**Sreek8s deployment**

**# Create sreek8s-deploy deployment using kubectl command**

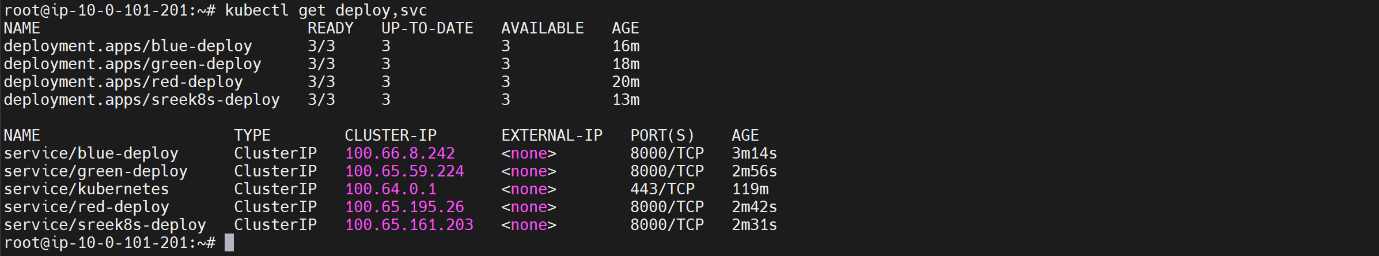
--- kubectl create deploy sreek8s-deploy --image=sreeharshav/testcontainer:v1 --replicas 3

# Expose deployment using clusterip service

--- kubectl expose deploy sreek8s-deploy --port=8000 --target-port=80

**# List deployment and service**.

--- kubectl get deploy,svc



**Ingress creating for sreek8s**

#Multiple services in the namespace can use the same port type but when we expose using NodePort

#they will have a random node port.

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

  name: sreek8s-deploy-ingress

spec:

  ingressClassName: nginx

  tls:

  - hosts:

    - www.stacksimplify.club

    secretName: sreeharshak8s

  rules:

  - host: www.stacksimplify.club

    http:

      paths:

      - path: /

        pathType: Prefix

        backend:

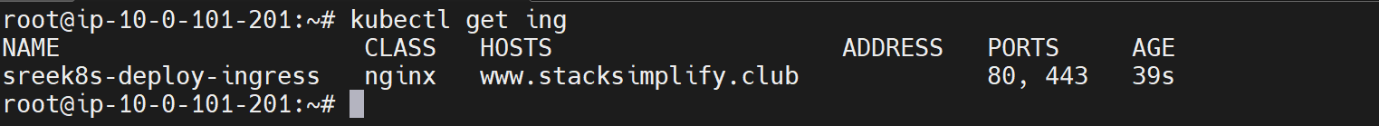
          service:

            name: sreek8s-deploy

            port:

              number: 8000

--- **kubectl get ing**



--- **note** - now go to the route53 and create record for www.stacksimplify.club

**Ingress controller deployment**

For Batch 08 I have used ingress v1.0.5

Kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.0.5/deploy/static/provider/aws/deploy.yaml

Use the ingress file red-blue-green-ingress.yml

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-0.32.0/deploy/static/provider/aws/deploy.yaml>

https://github.com/kubernetes/ingress-nginx

Kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.44.0/deploy/static/provider/aws/deploy.yaml>

NGinx Ingress Controller 1.0.0, got issue as no address when we run kubectl get ing. This issue is solved by using the annotation for ingress as kubernetes.io/ingress.class: "nginx"

Need to check the notes. Deploy 0.44.0 using above URL.

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.0.0/deploy/static/provider/aws/deploy.yaml

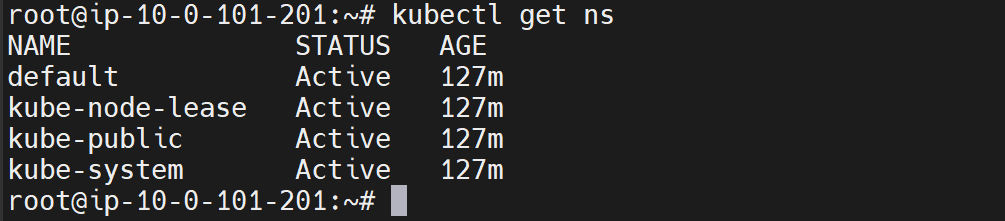
https://github.com/kubernetes/ingress-nginx

https://medium.com/@madeeshafernando/deploying-multiple-ingress-controllers-in-a-kubernetes-cluster-dc6c7700a795

**verify the ingress is installed or not**

**# List the name spaces**

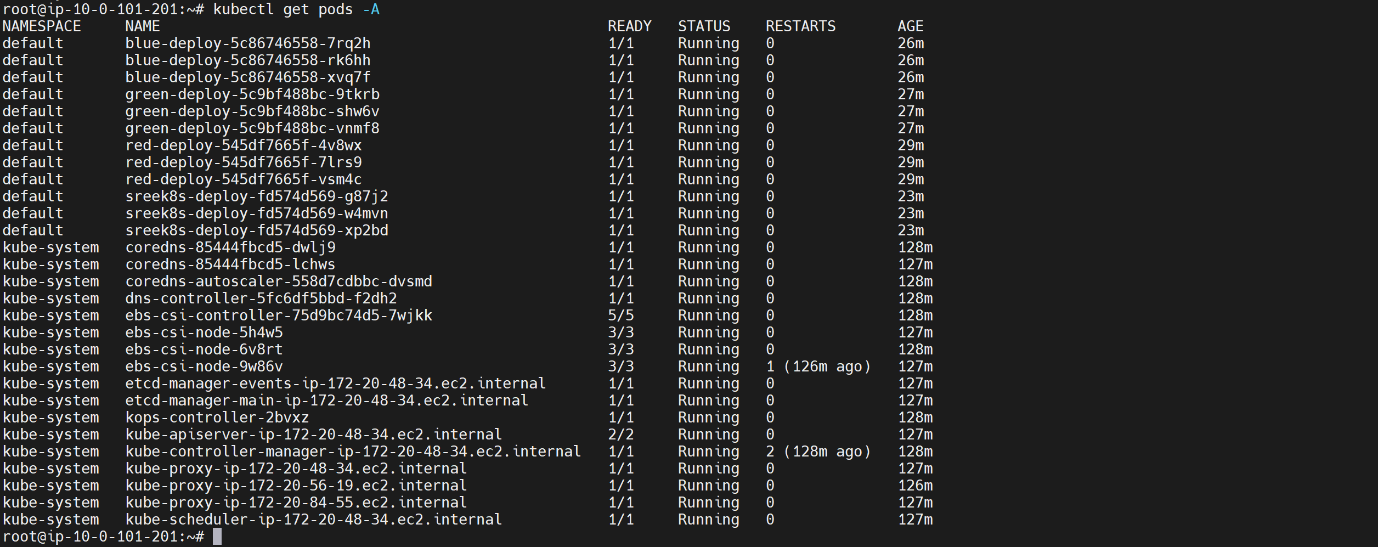
--- kubectl get ns



--- **note** – you can see here, the ingress name spaces is not created.

**# List all pods**

--- kubectl get pods -A

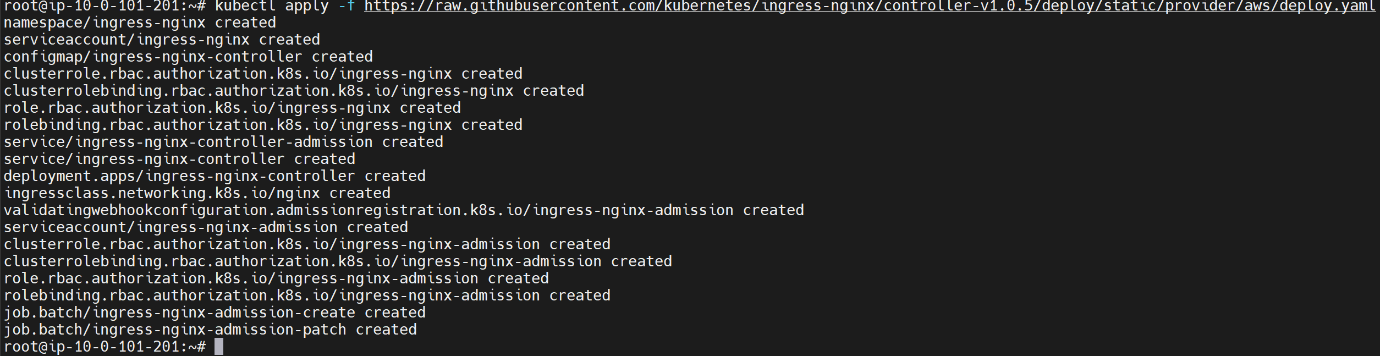


--- **note** – no ingress pod is running.

**Ingress controller creating**

--- **note** – this deployment will create L4 network load balancer in the aws.

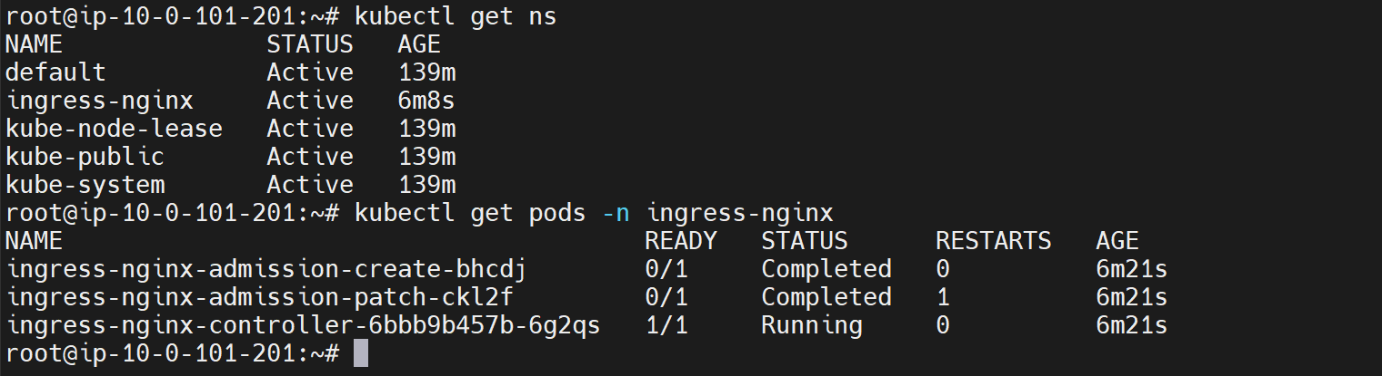
--- kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.0.5/deploy/static/provider/aws/deploy.yaml>



**# Verify the ingress is installed or not.**

--- kubectl get ns

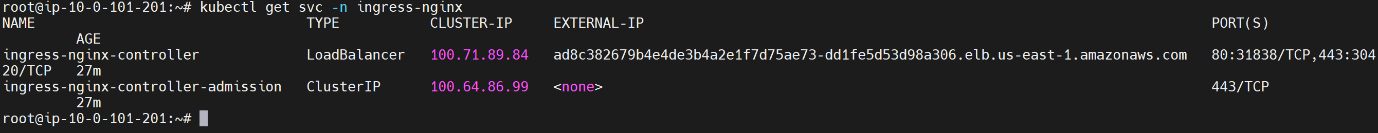
--- kubectl get pods -n ingress-nginx



--- **note** – ingress name space got created and pods also running in that name space.

**# List the ingress service in the**

--- kubectl get svc -n ingress-nginx

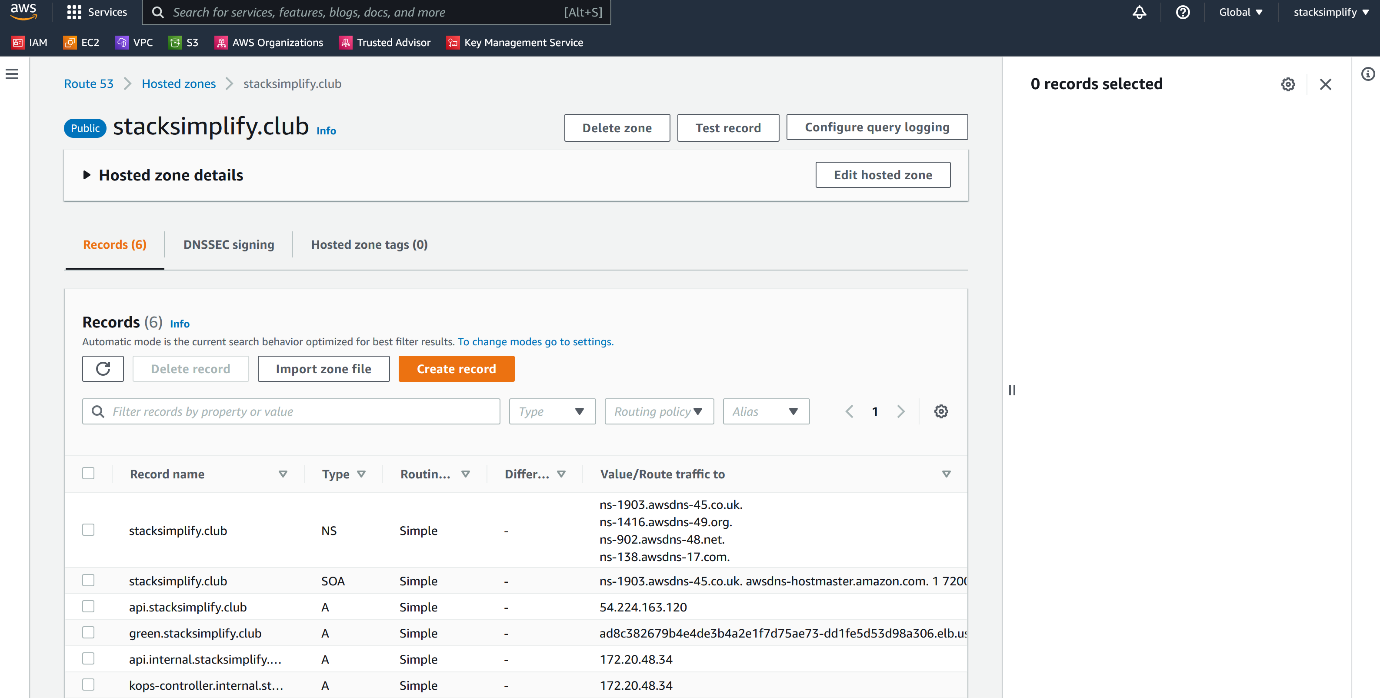


--- **note** – it is deployed load balancer in the aws.

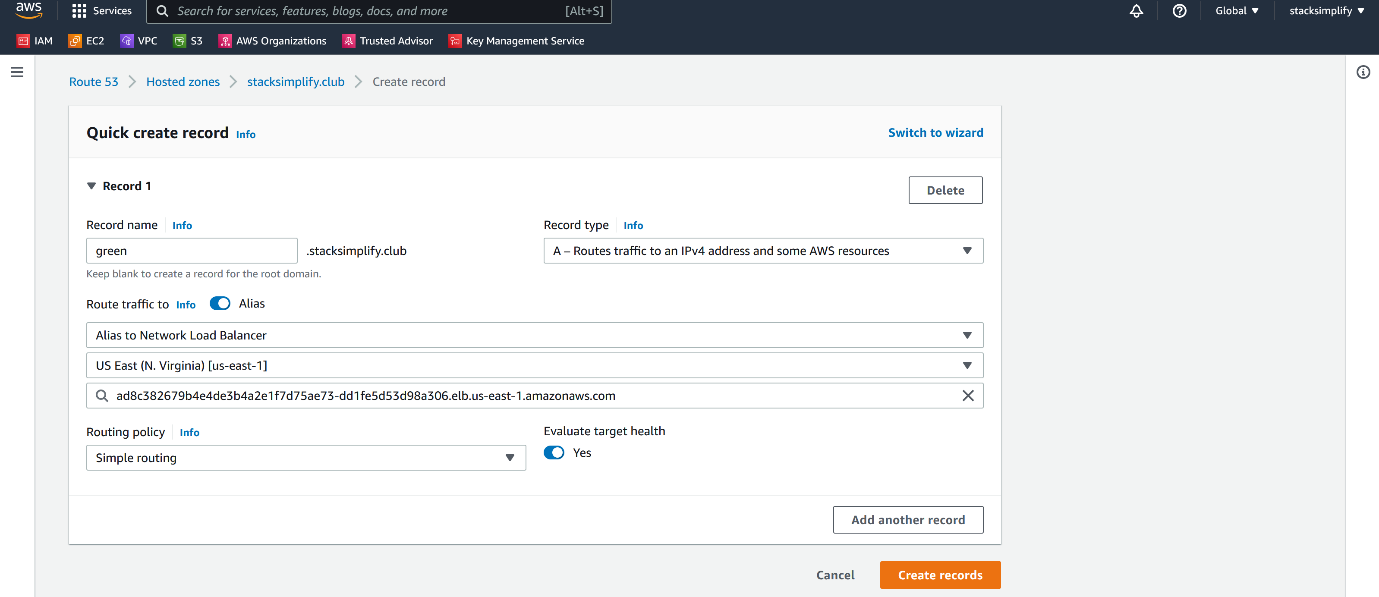
**Create records in route53**

--- **note** – we have to create records for our deployments in order to access application.

**Create green deployment record**

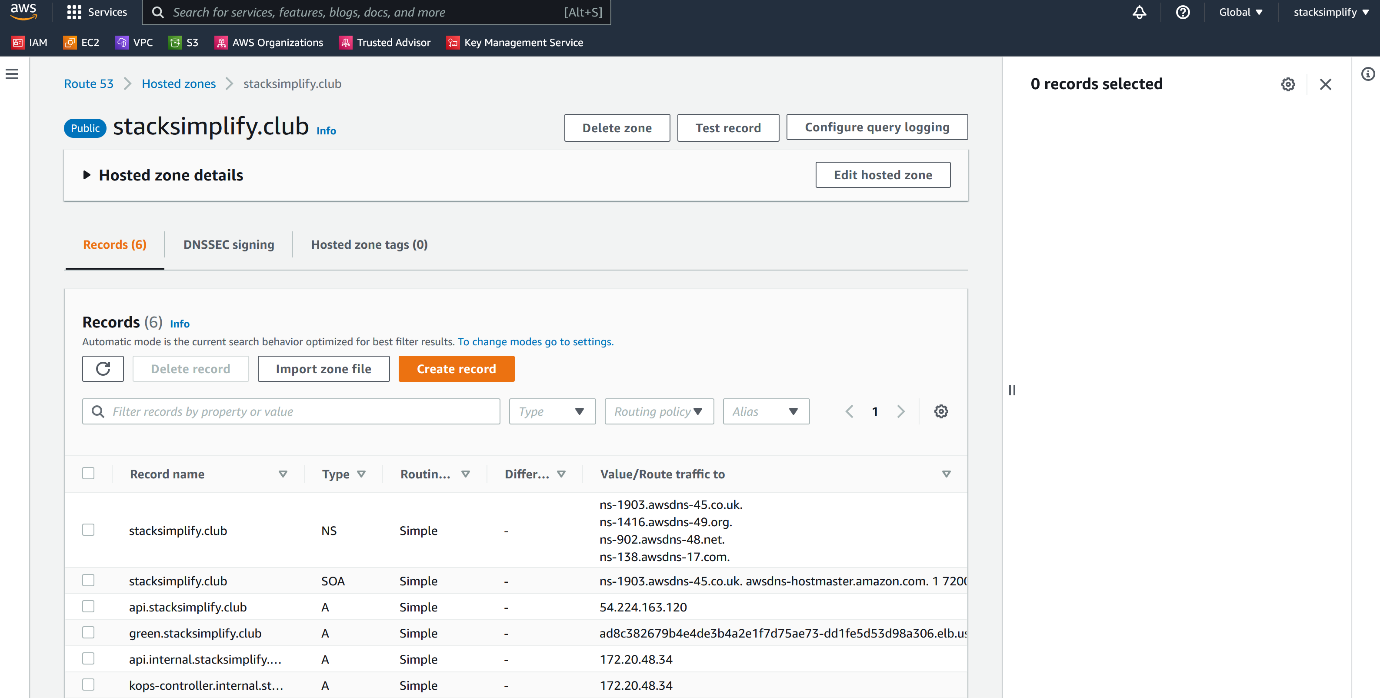


--- click on create record.

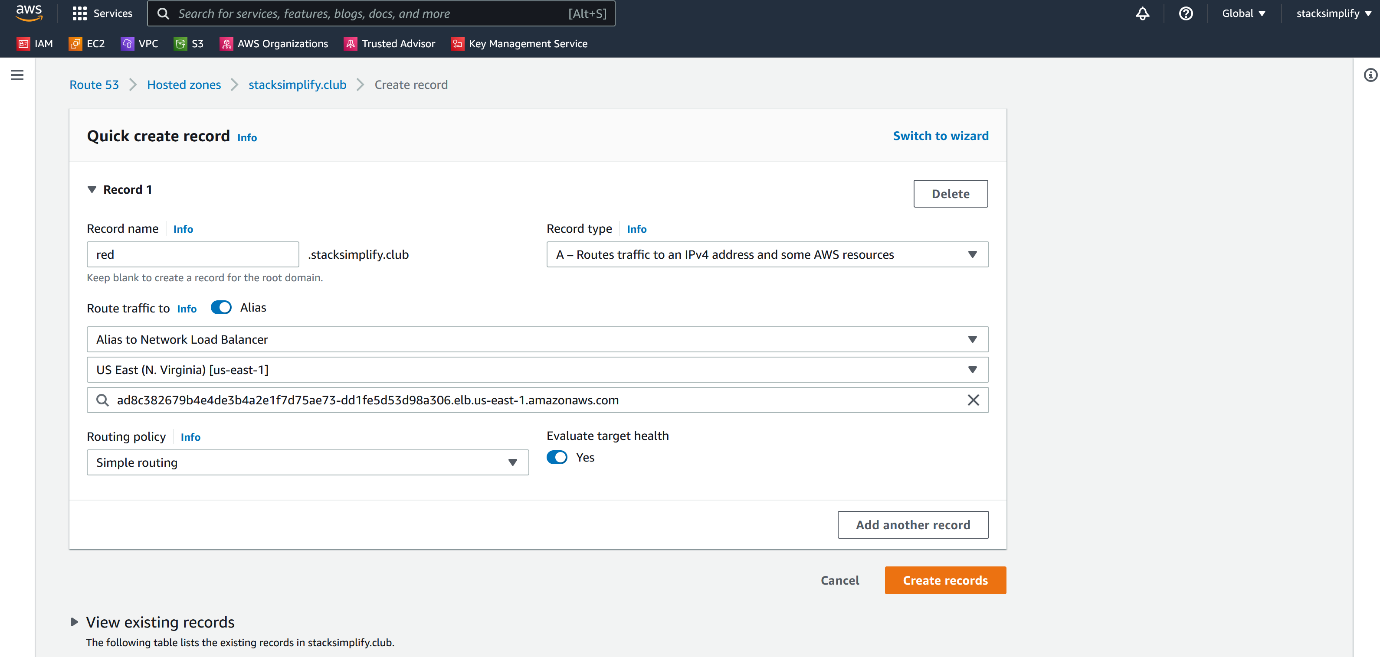


---

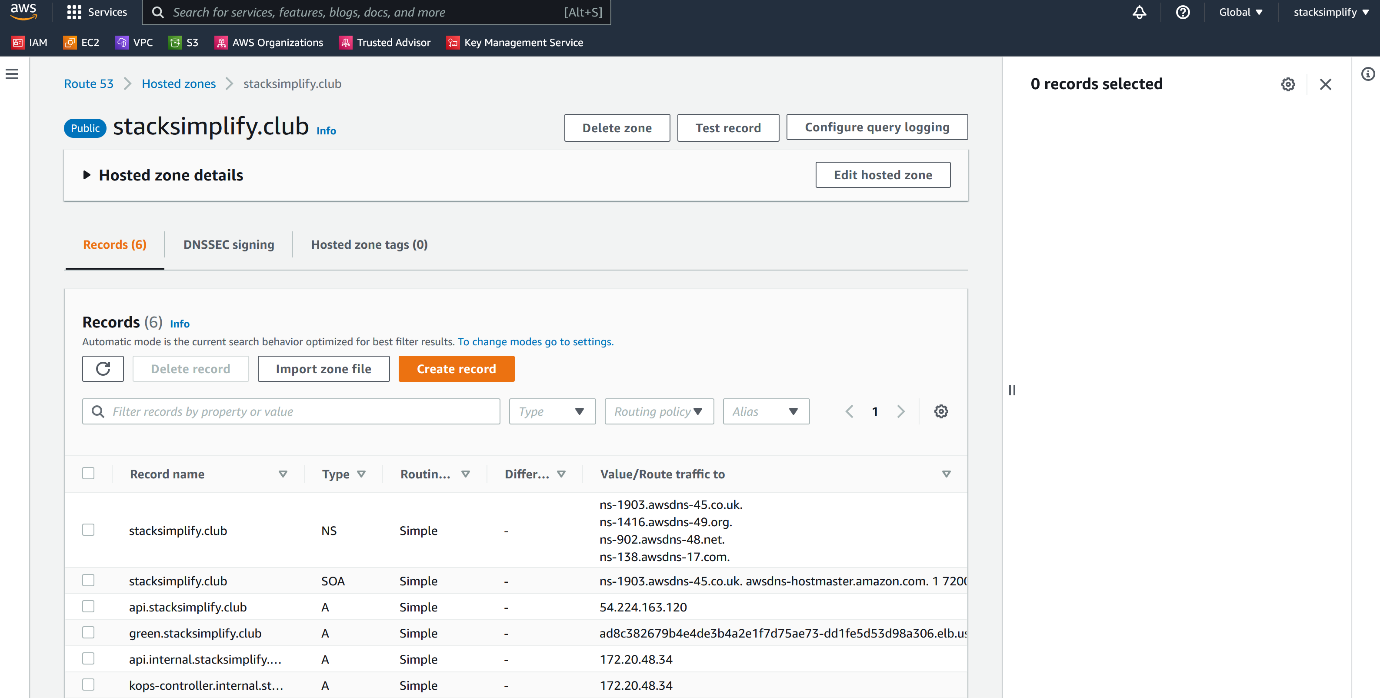
**Create red deployment record**



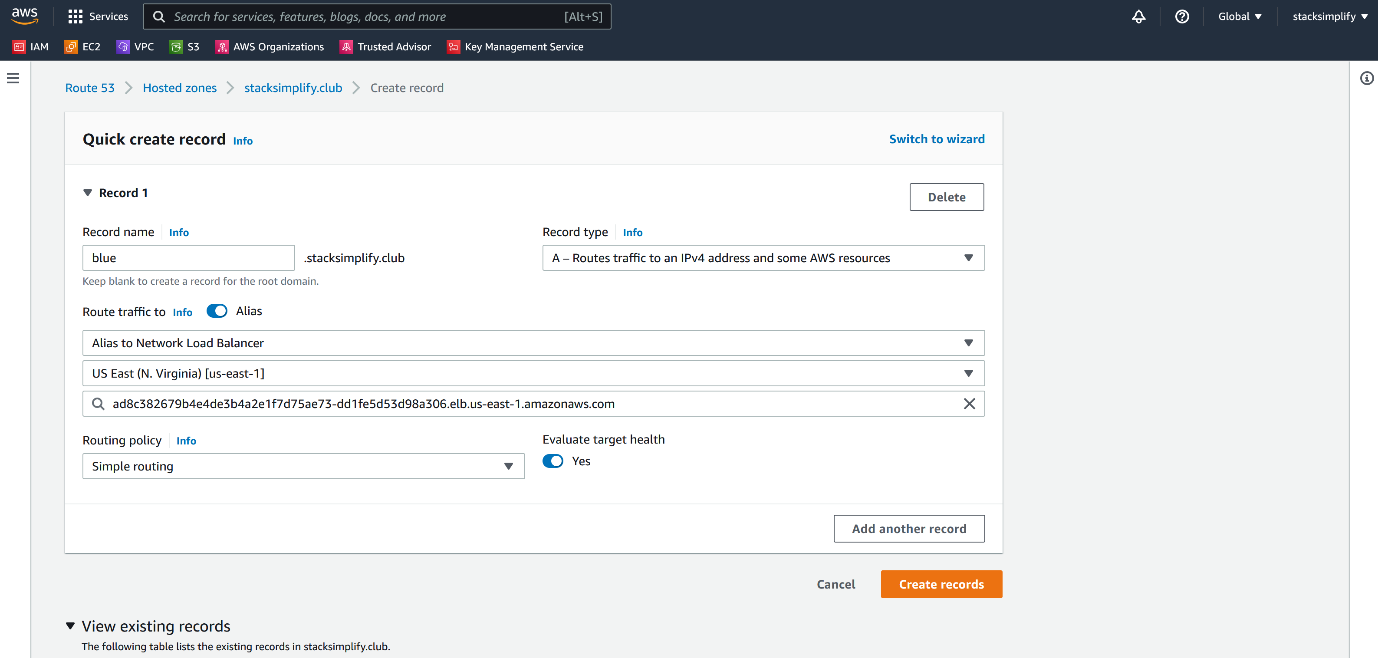
--- click on create record.



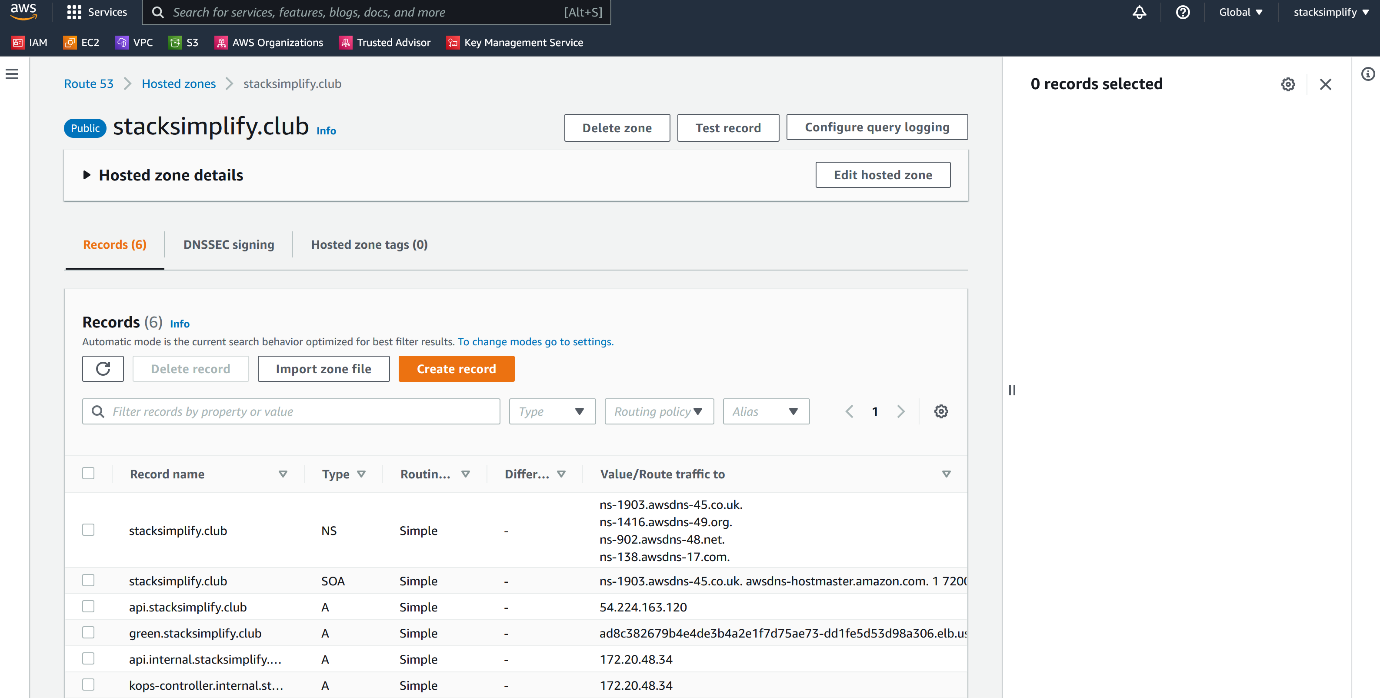
**Create blue deployment record**



--- click on create record.



**Create sreek8s deployment record**



--- click on create record.

