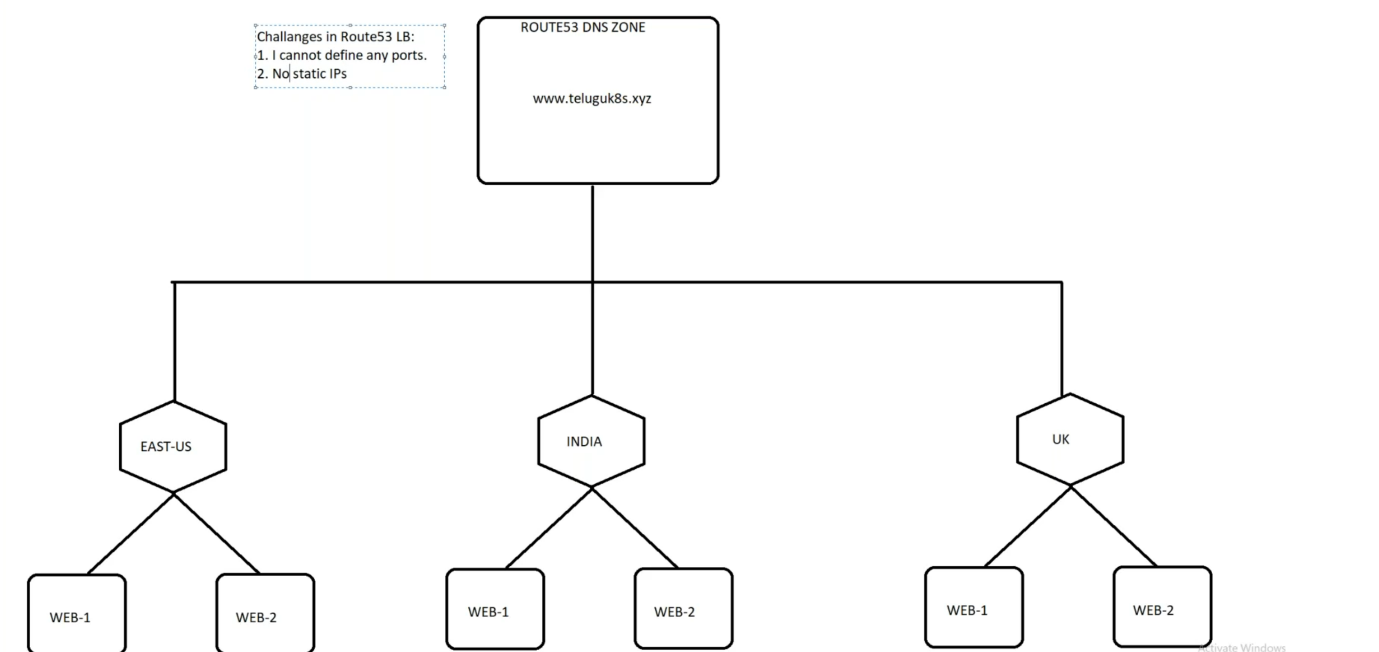
**23.AWS-B30-GlobalAccel-CloudFront**

--- in this session, we will learn about aws global accelerator, cloudfront - CDN (content delivery network).

--- **what are the advantages and disadvantages of AWS rote53…?**



* Route53 is a dns level load balancer.
* I cannot define any port, ports like http, https
* No static ip address.

**--- what are the advantages and disadvantages of aws global accelerator…?**

* It will provide 2 static ip address which can be accessed from anywhere.
* I can also define which ports needs to be open.
* It does not support or provide caching.

--- **note** – AWS global accelerator is a good solution if you have a global applications and load balance between them. We can also regulate the traffic based on weights which can be very helpful for canary deployment.

**global accelerator**

--- note – I will create 3 services in 3 different zones and I will load balance these services using global accelerator.

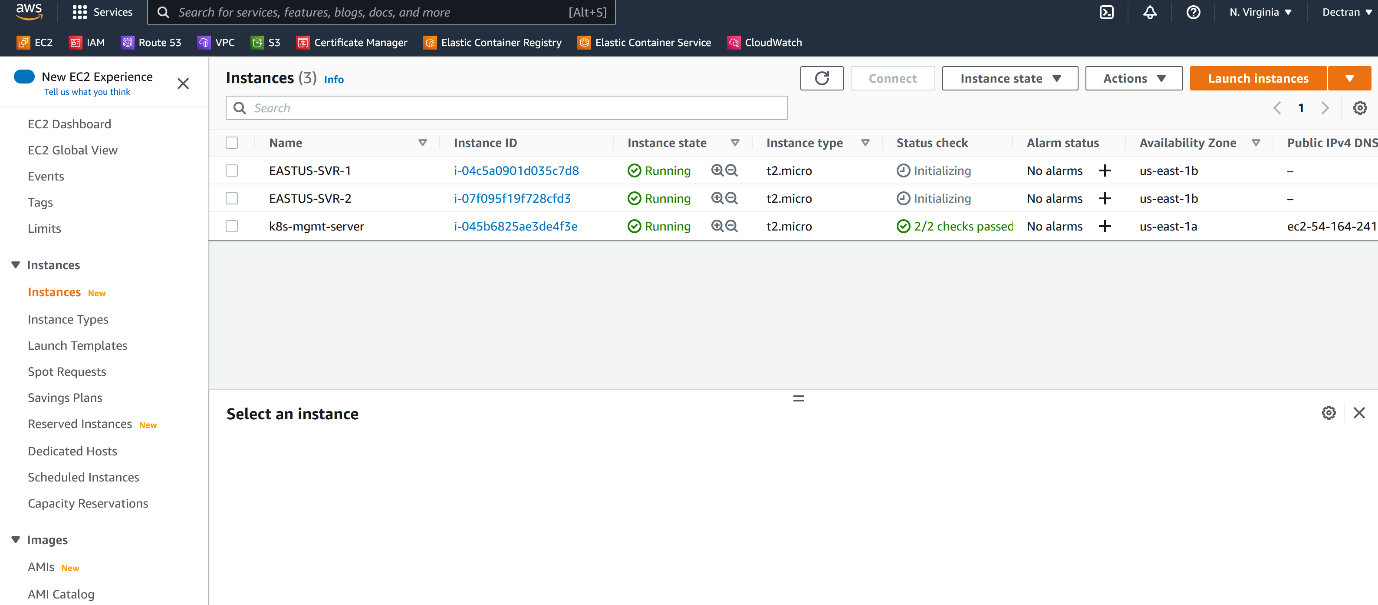
1. In N. Virginia, I will create application load balancer.
2. In London, I will create ec2 service.
3. In Mumbai, I will create ec2 service.

**N.Virginia**

--- **note** – in this region, I will create application load balancer.

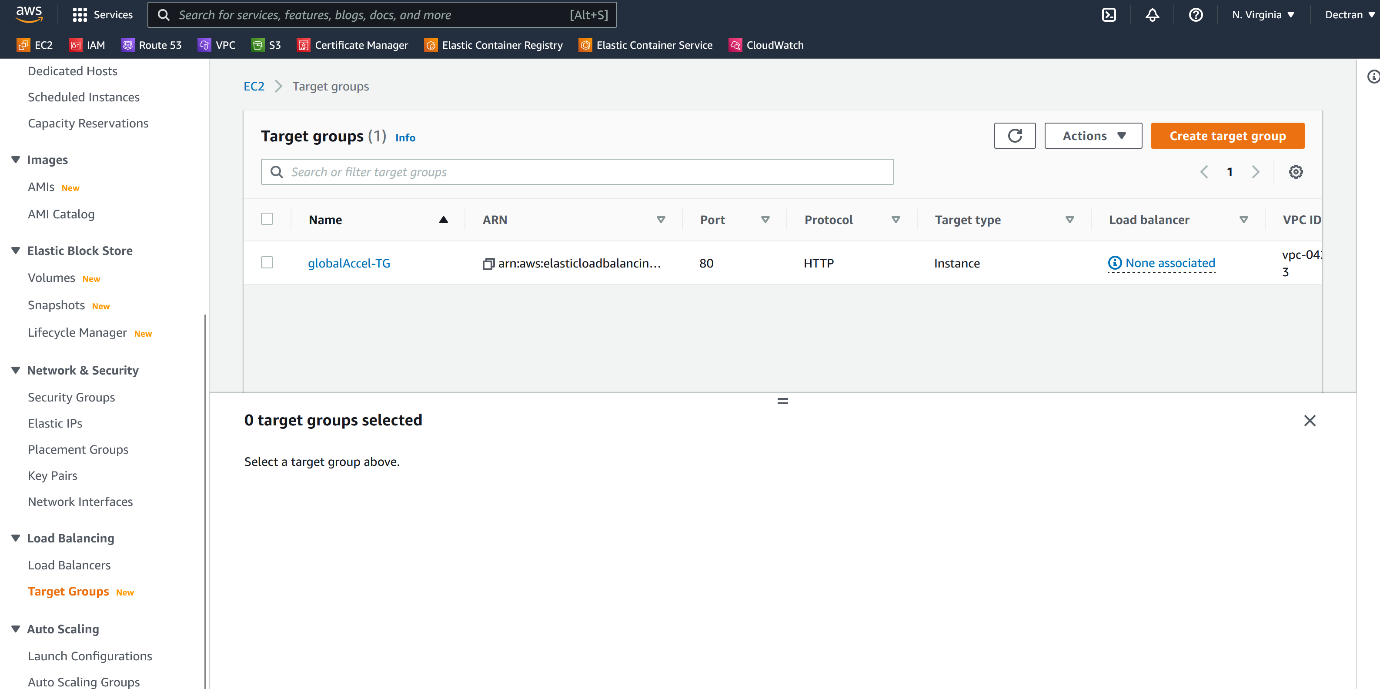
**CREATE 2 EC2 INSTANCE (**

--- create 2 ec2 instance

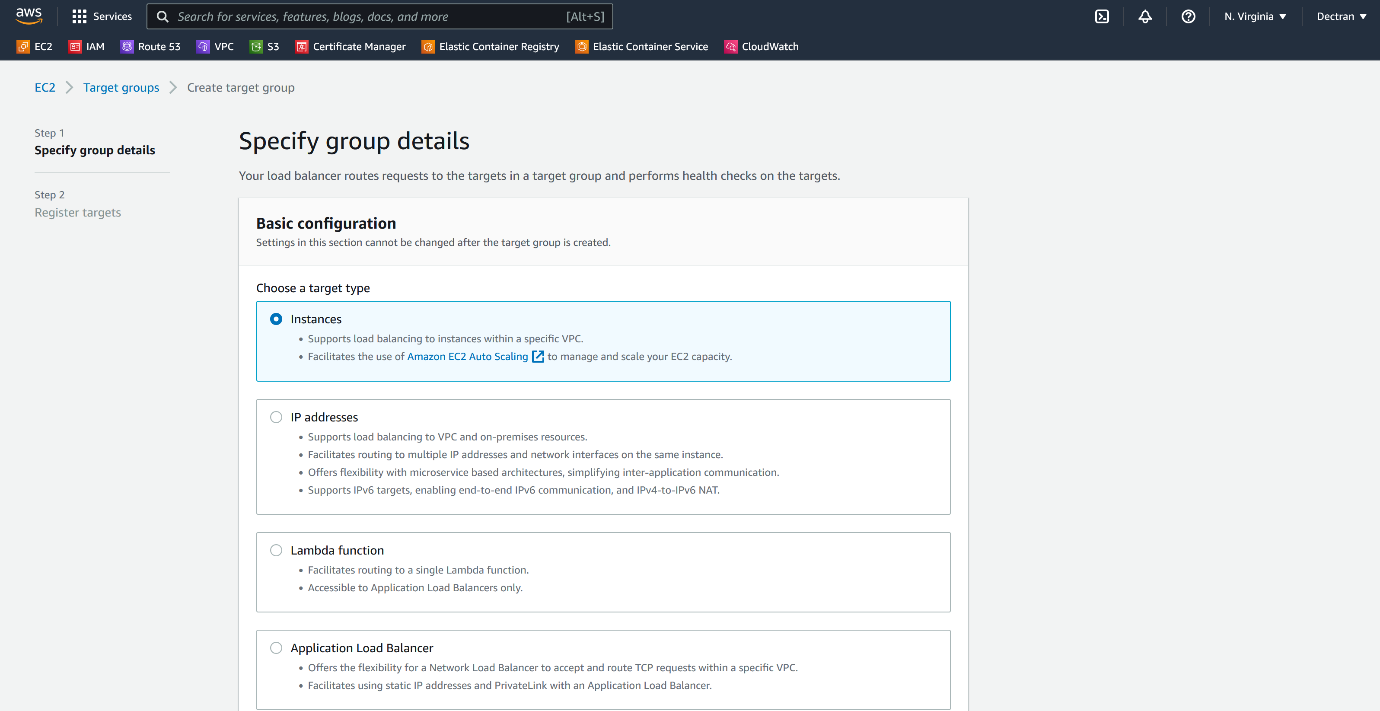


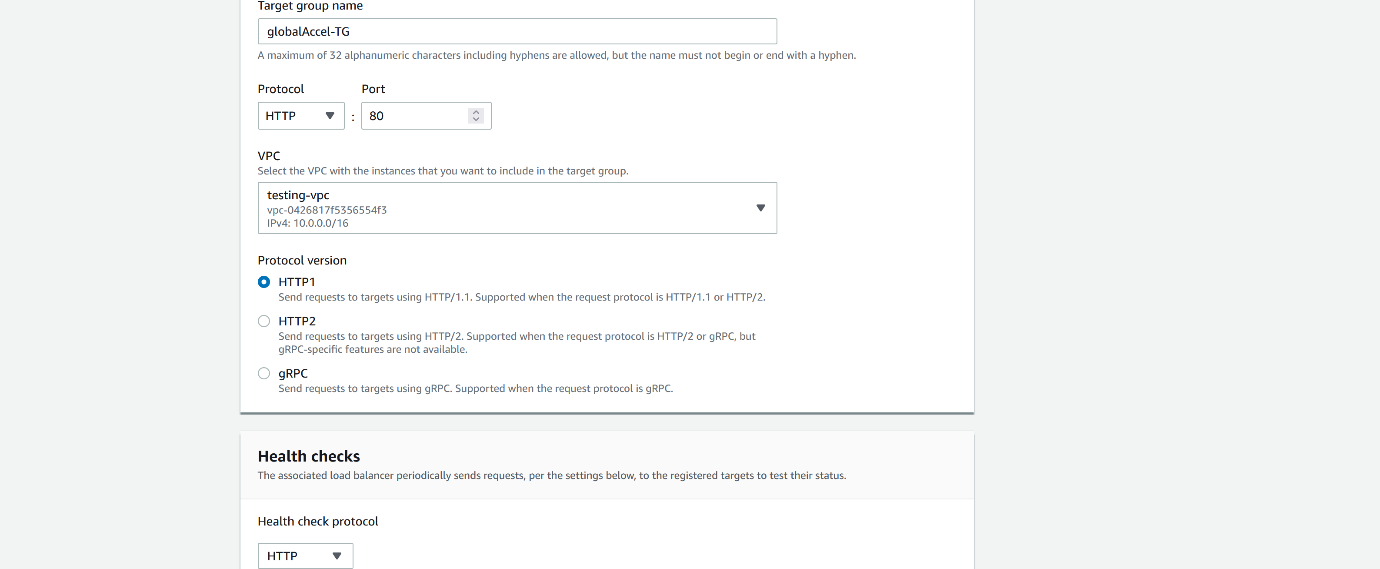
**Application load balancer**

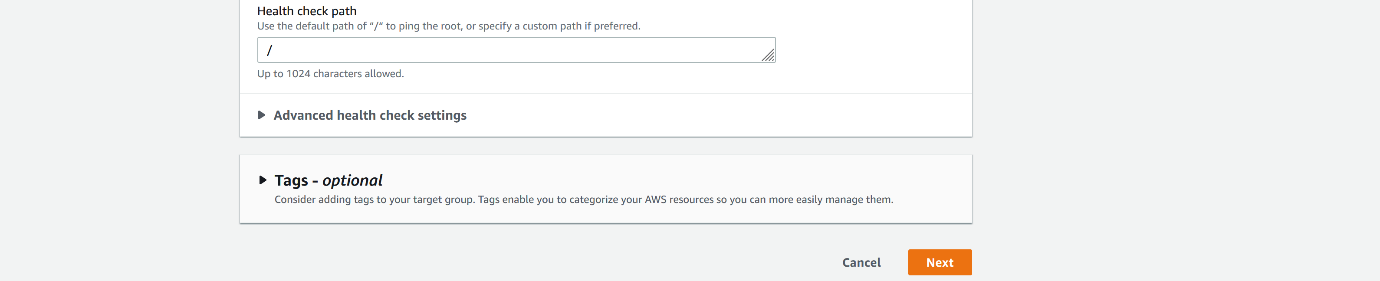
--- **Target group creating**



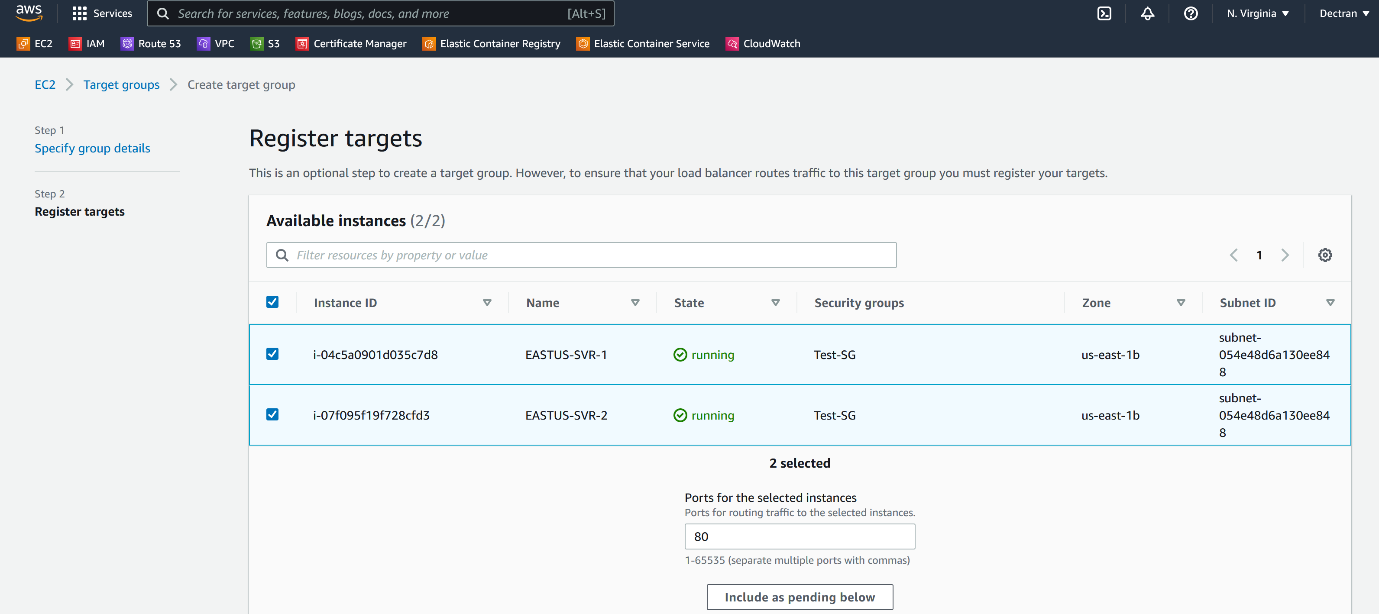
--- click on target groups.

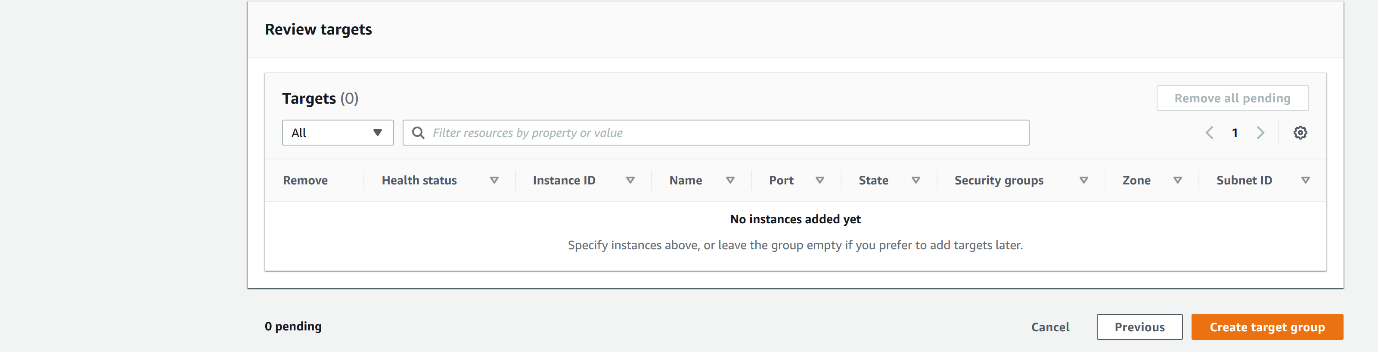






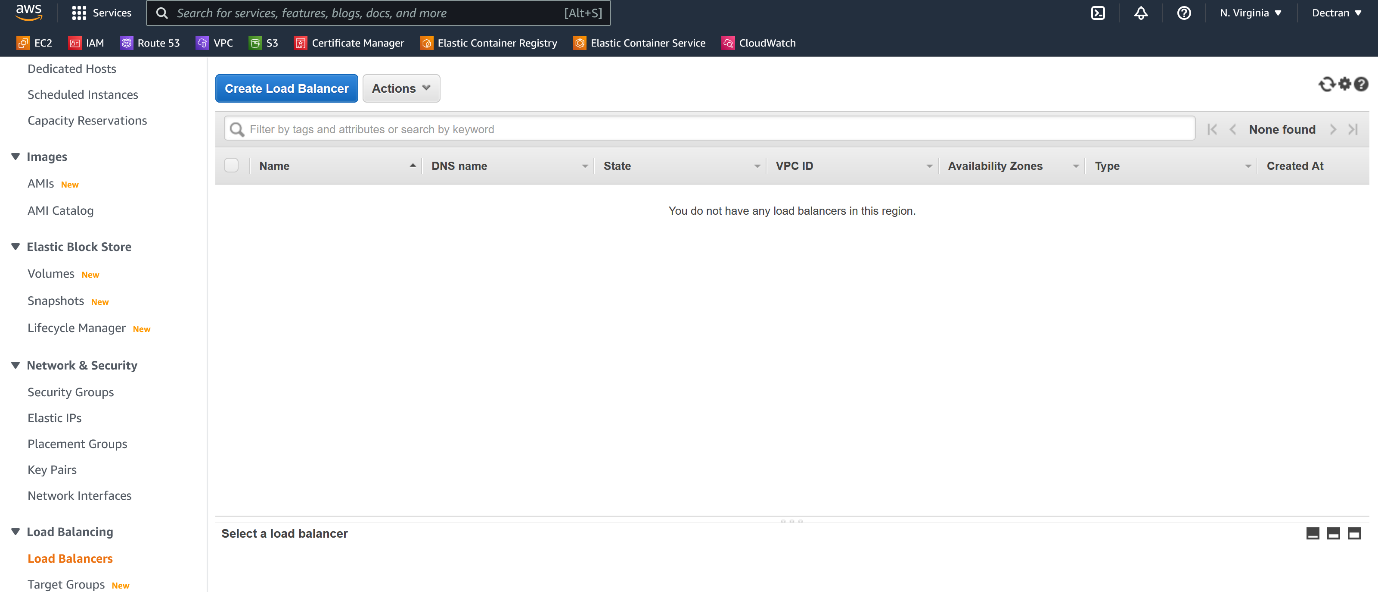
--- click on next.



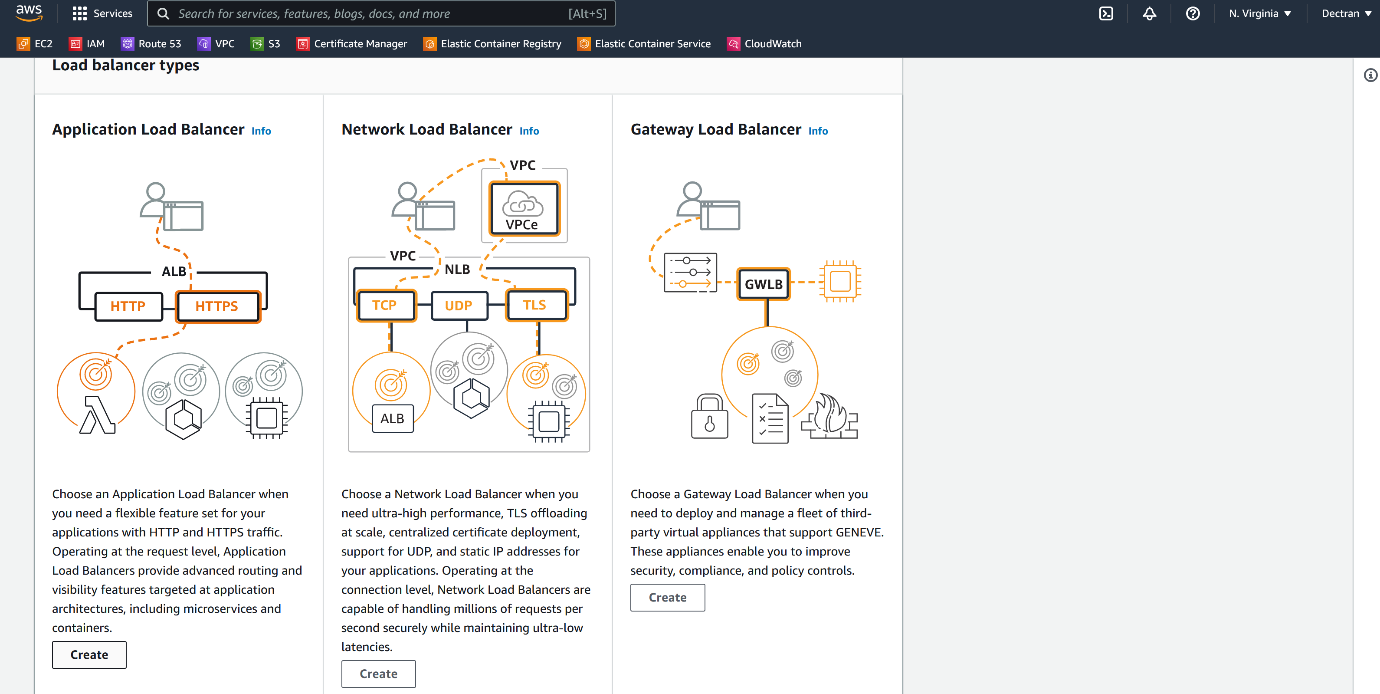


--- click on create target group.

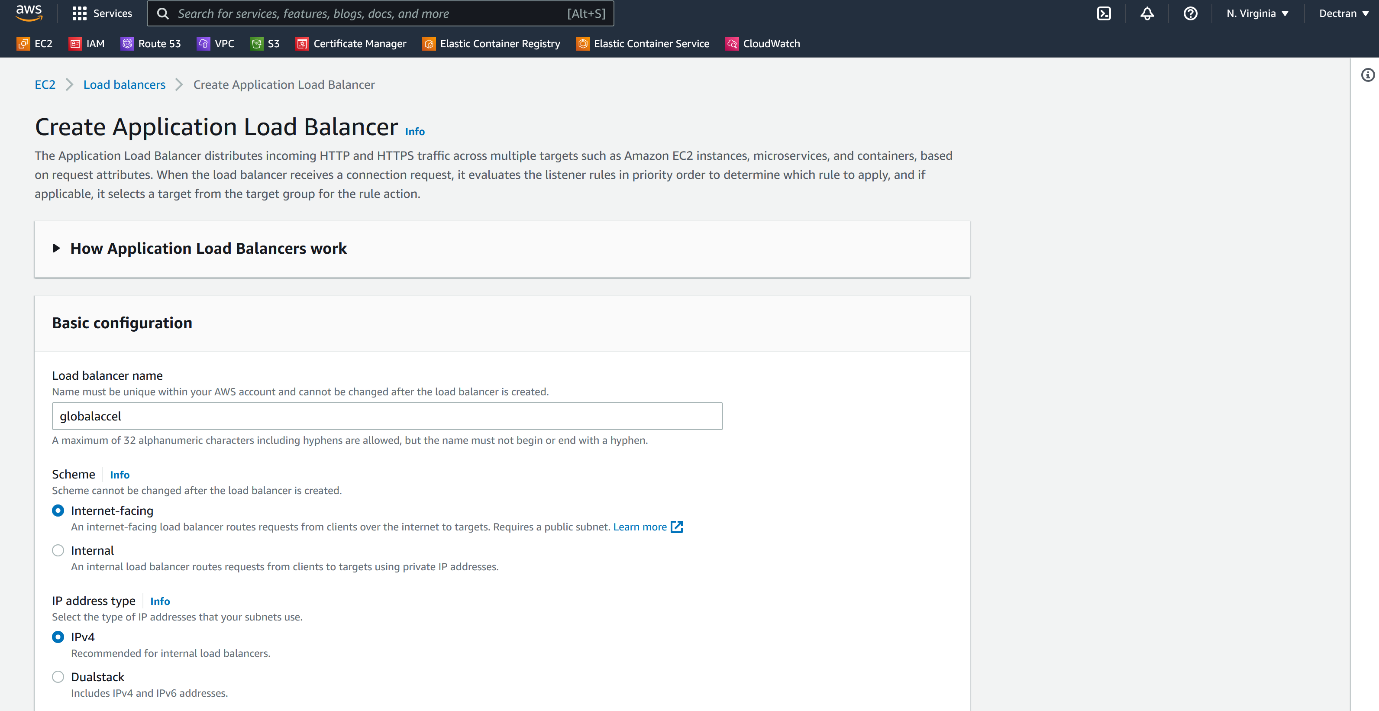
--- **creating application load balancer**

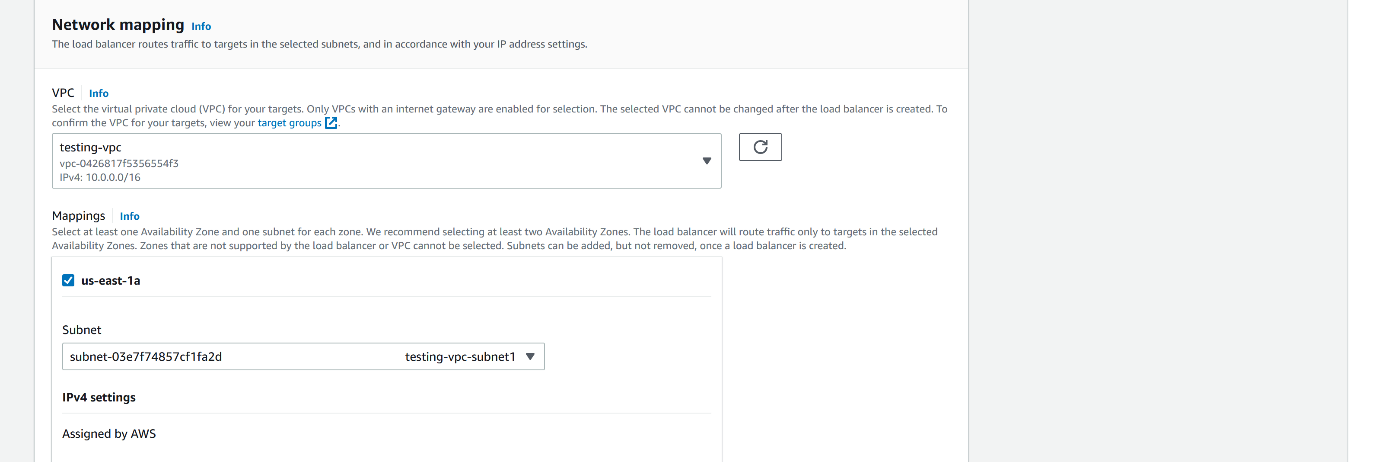


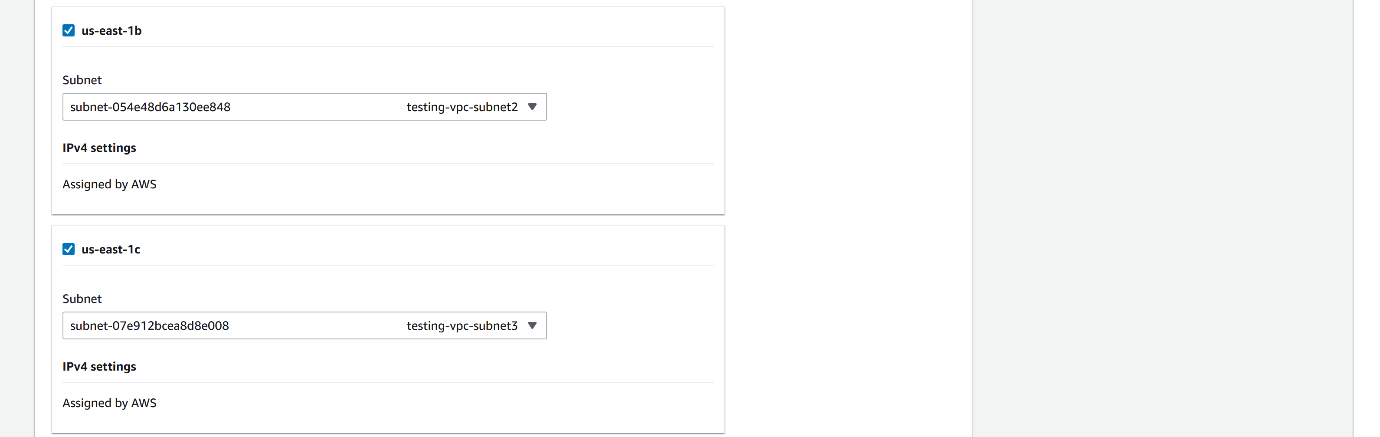
--- click on create load balancer.

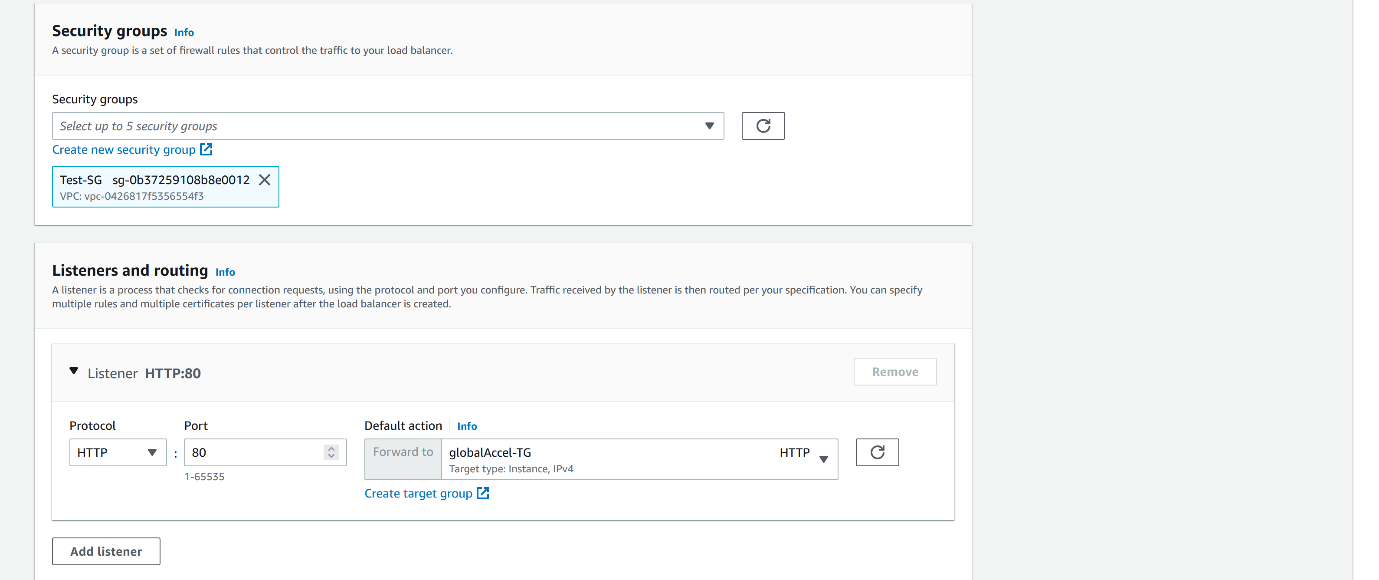


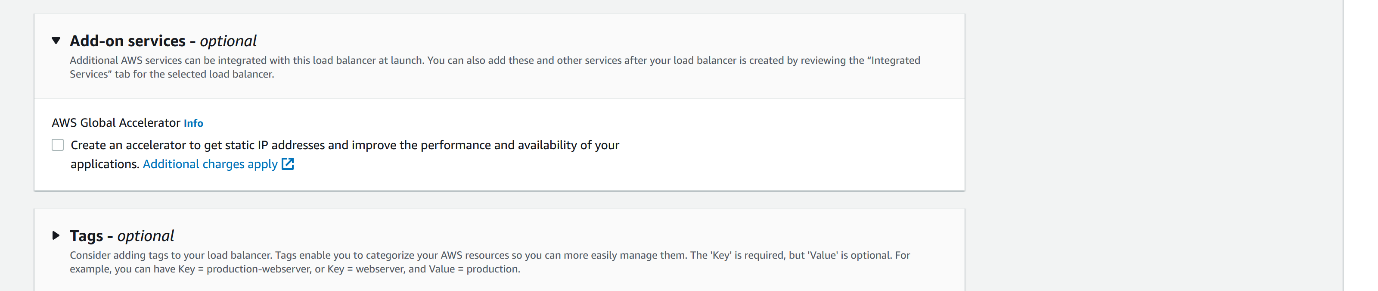
--- click on create.



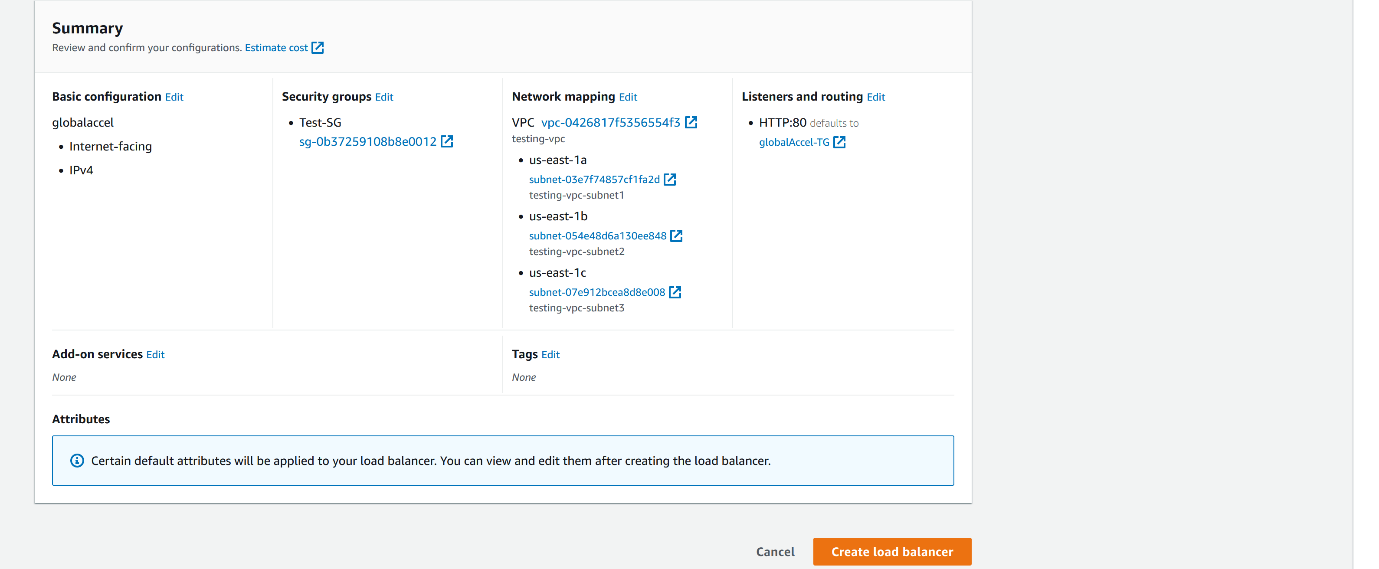




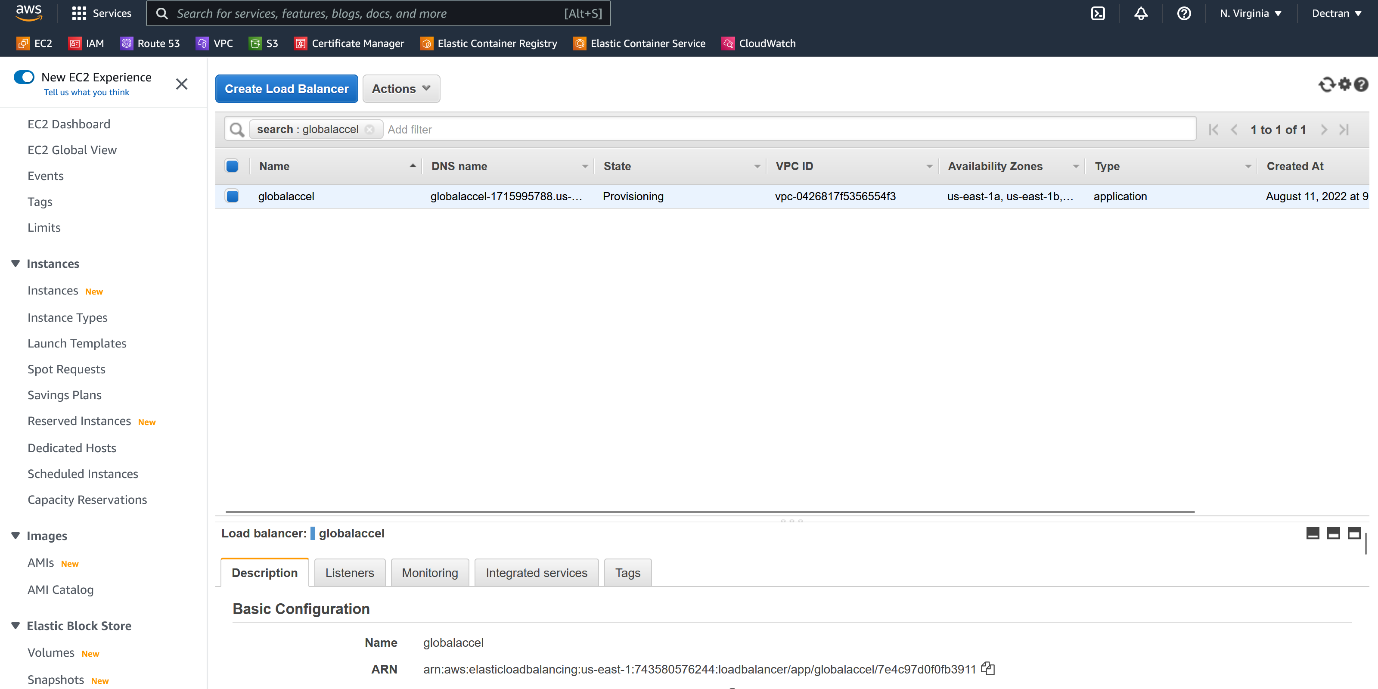




--- **note** – from here, we can also deploy load balancer.

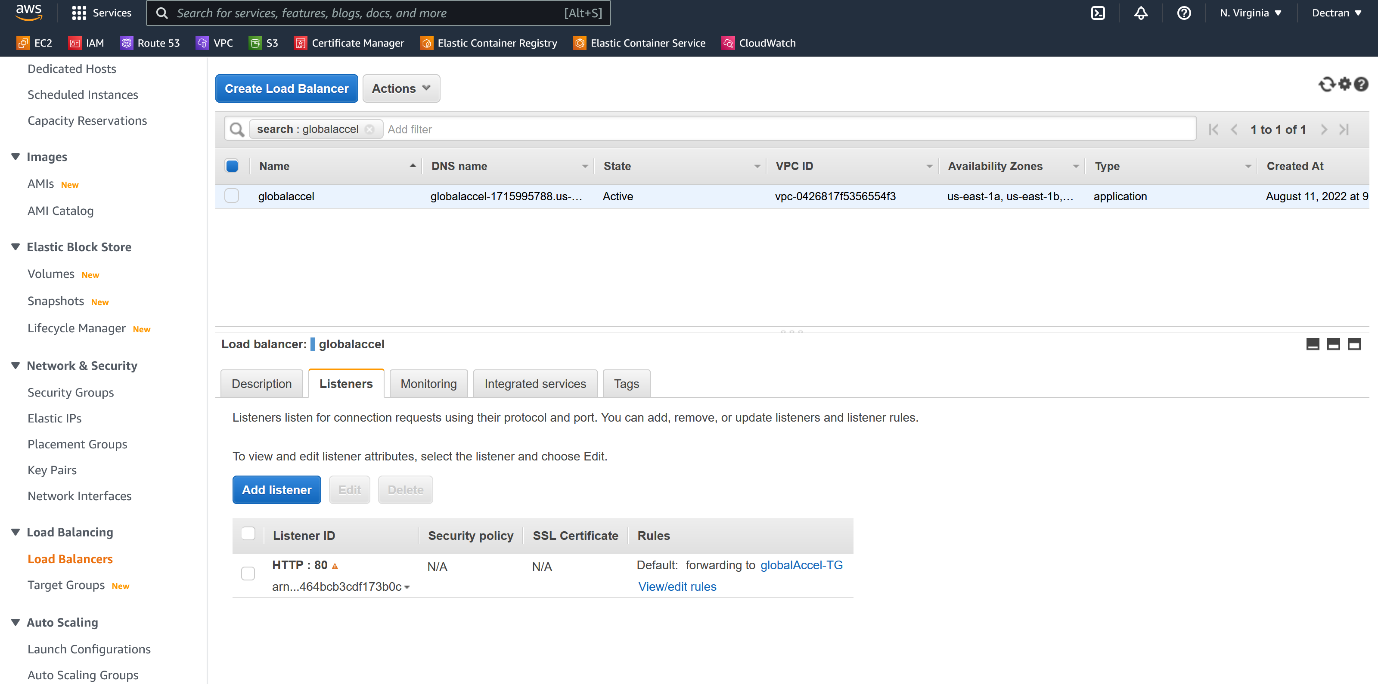


--- click on create load balancer.

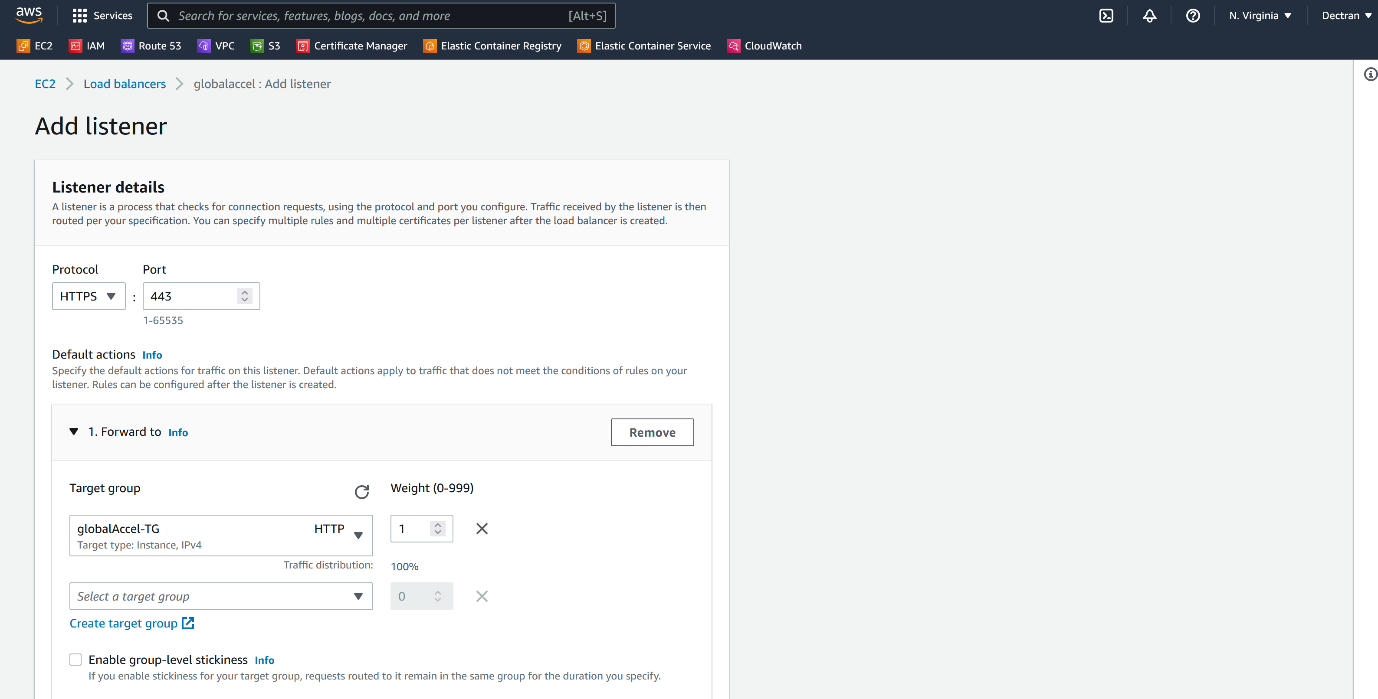


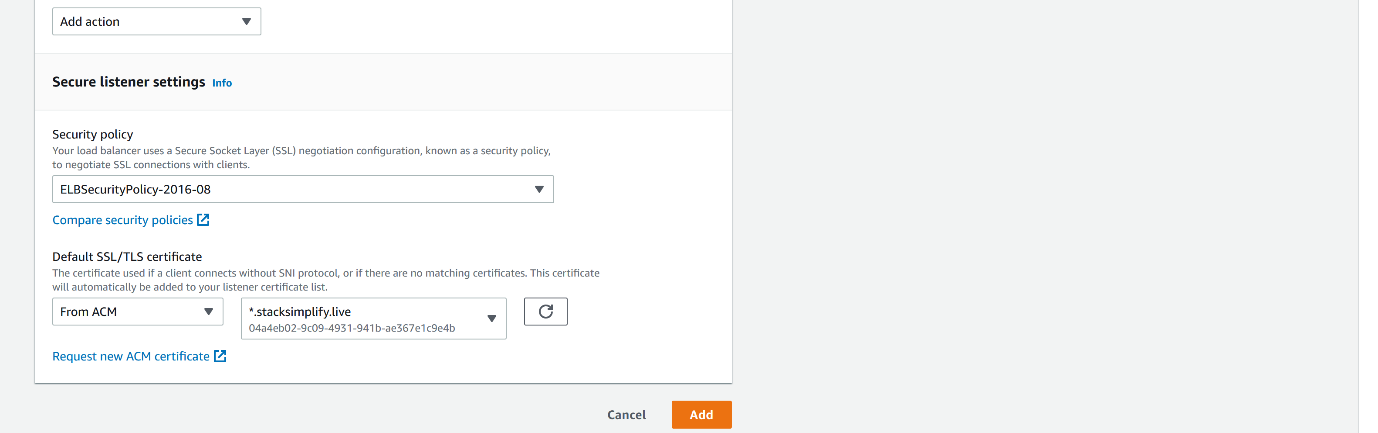
--- finally load balancer got created.

--- **add https listener**



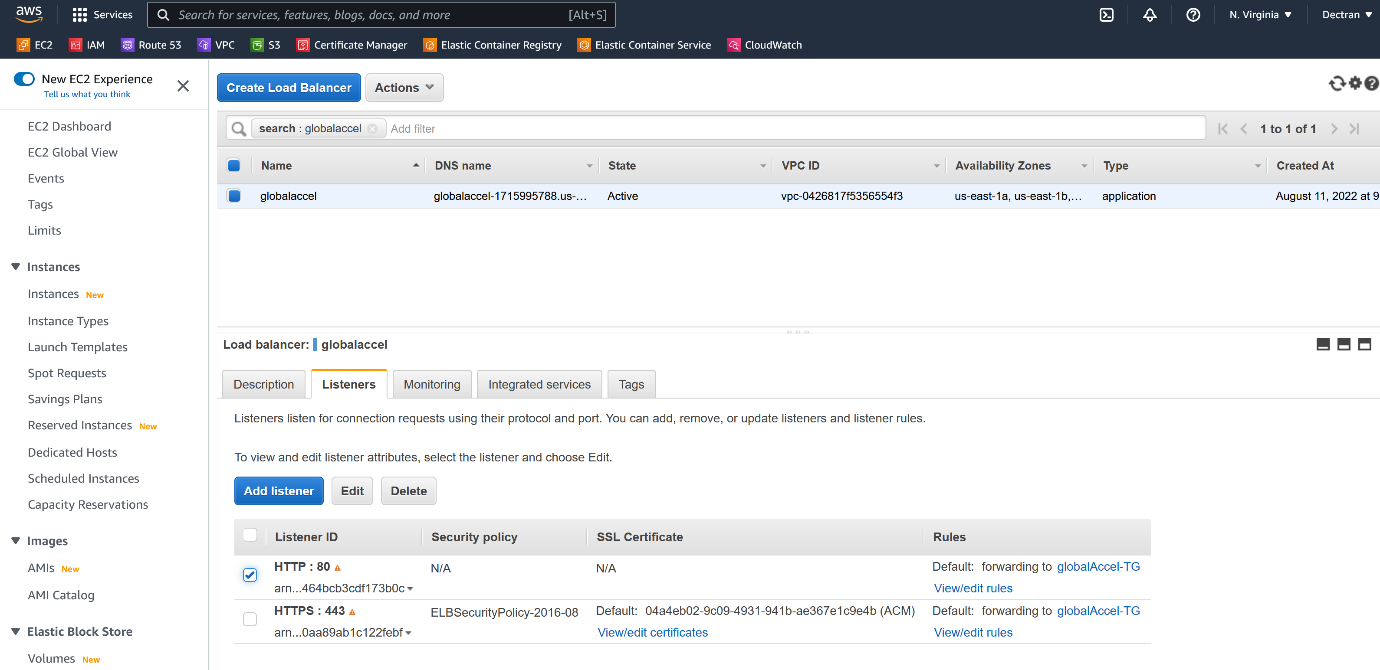
--- click on add listeners.



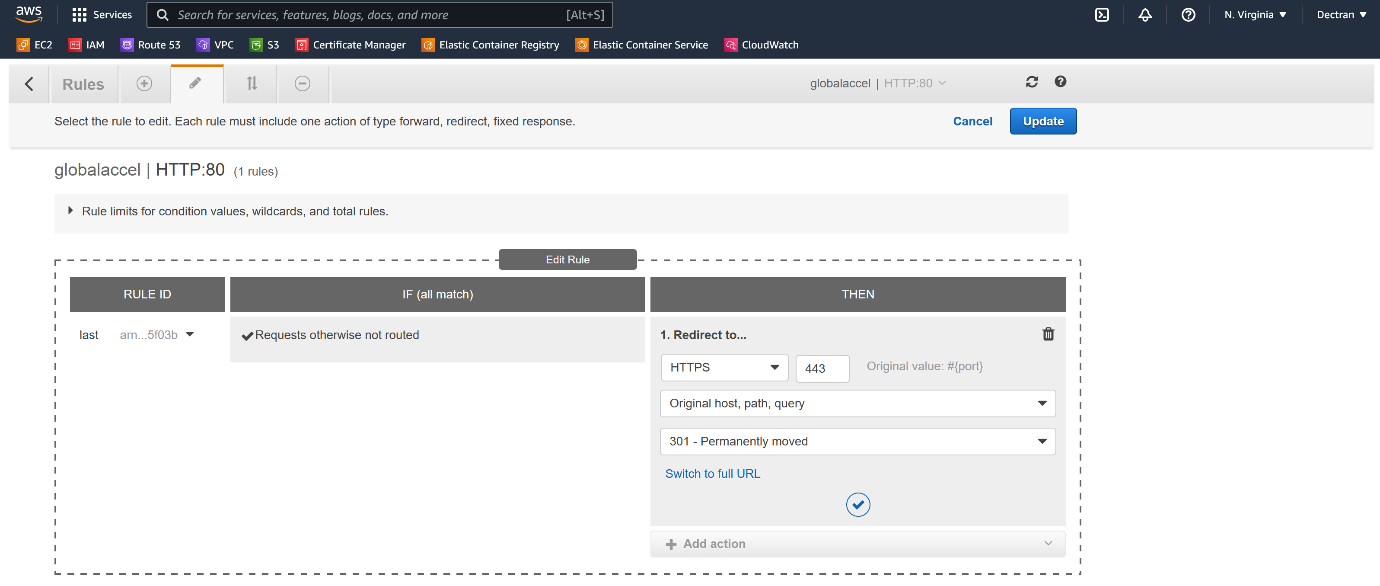


--- here, I selected my ssl certificate form ACM and click on Add.

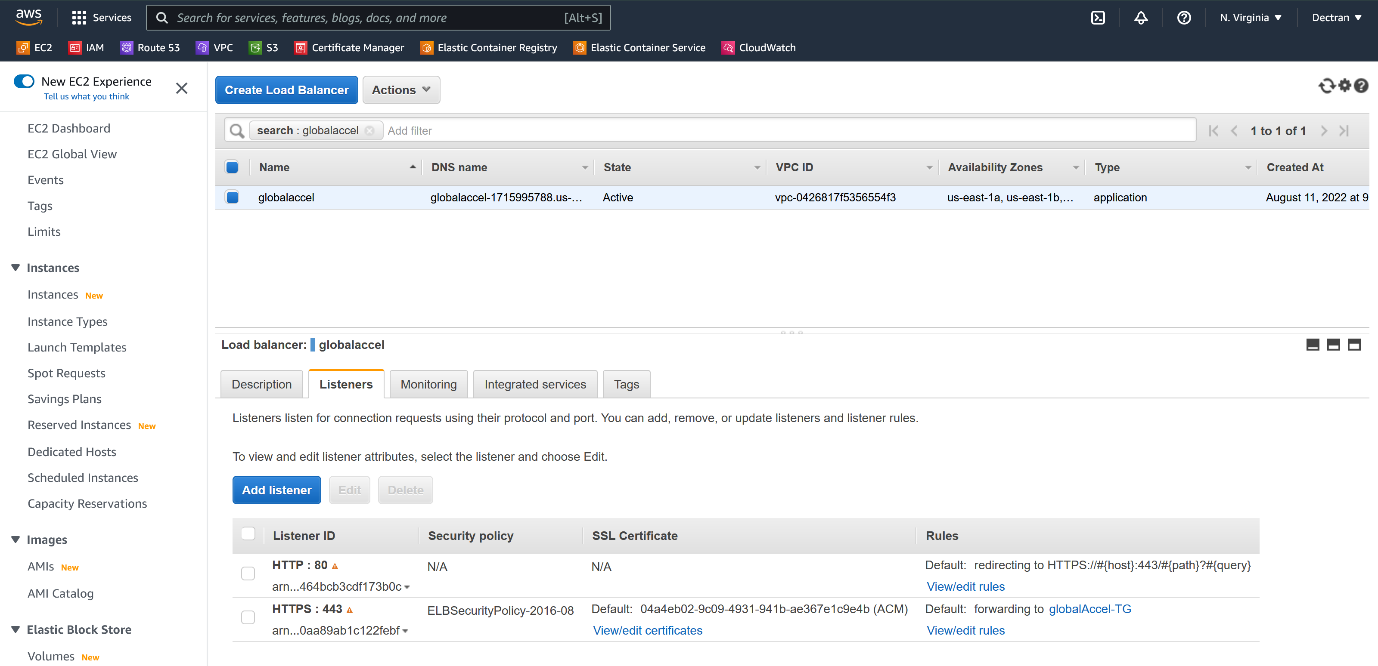
--- **redirect from HTTP to HTTPS.**



--- click on HTTP edit rules.



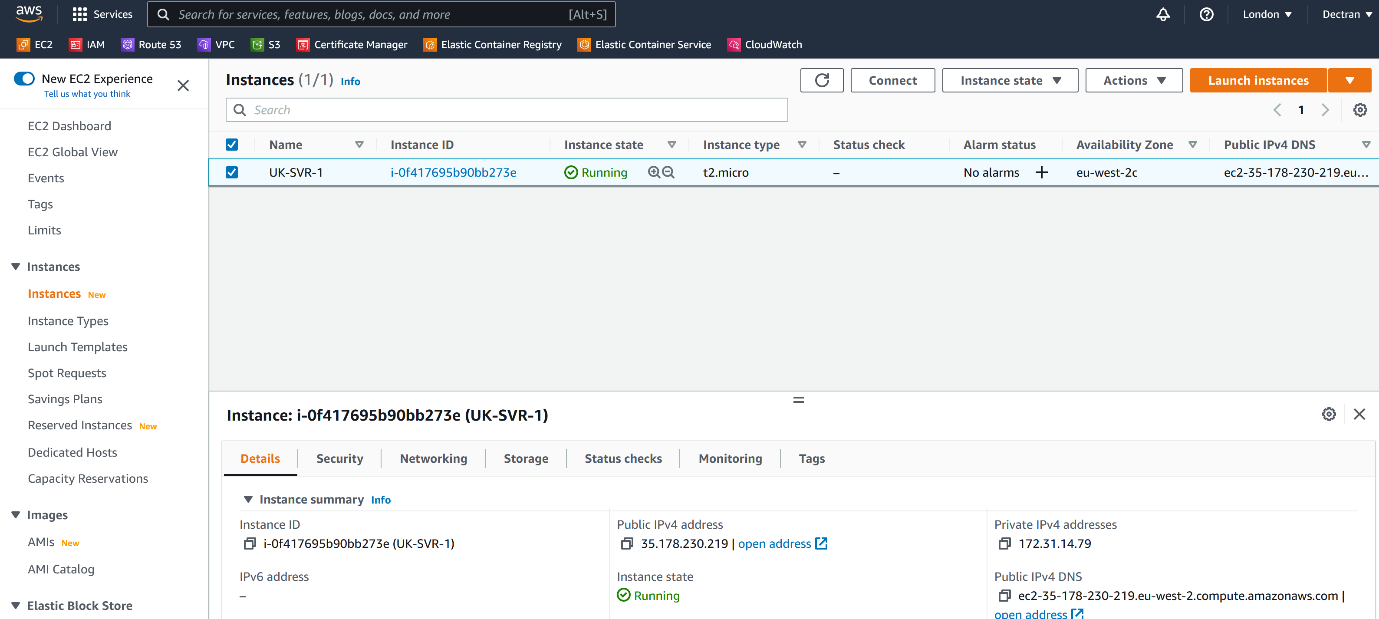
--- click on right symbol and update.



--- note – now you can see that the redirection will happen form HTTP to HTTPS.

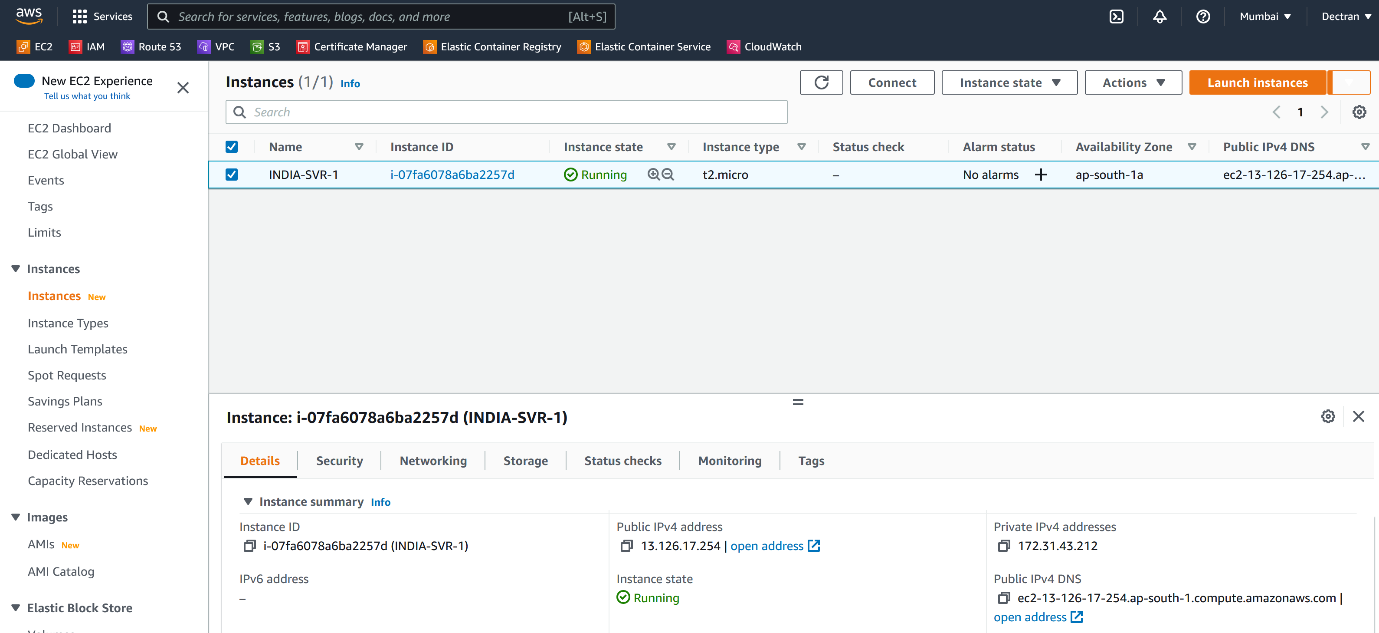
**London region**

--- **note** – in this region, I will create EC2 instance.



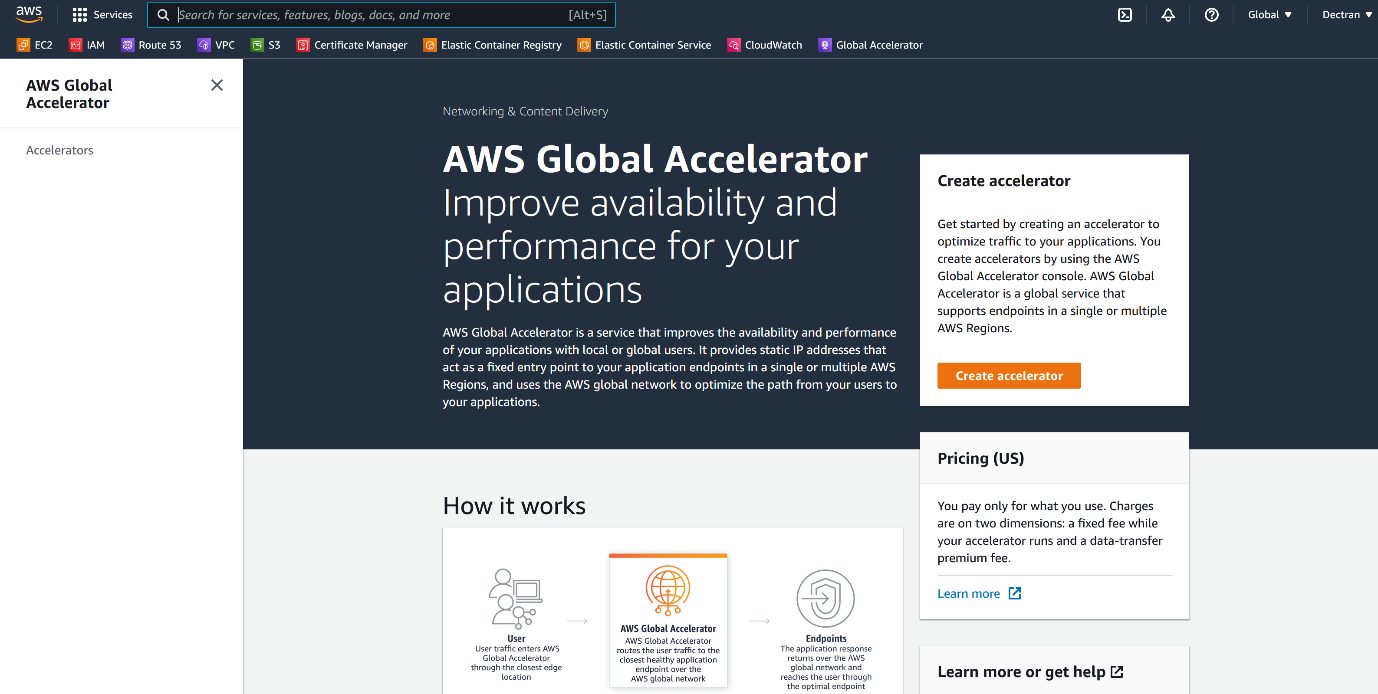
**Mumbai region**

--- in this region, I will create EC2 instance.



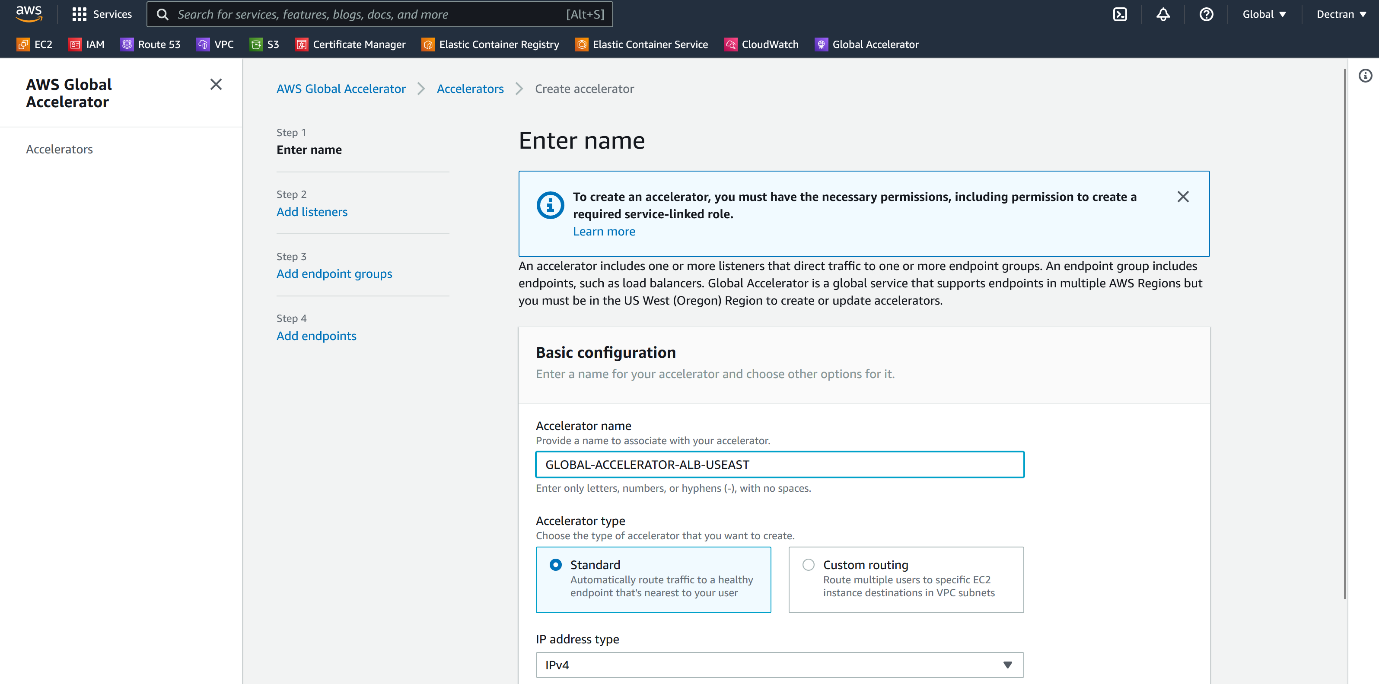
**Global load balancer creating.**

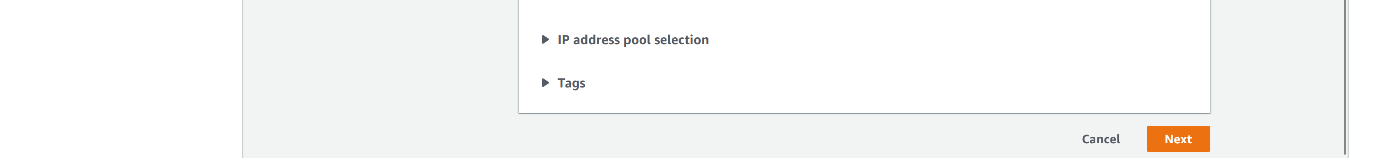
--- AWS Global Accelerator is a service that improves the availability and performance of your applications with local or global users. It provides static IP addresses that act as a fixed entry point to your application endpoints in a single or multiple AWS Regions, and uses the AWS global network to optimize the path from your users to your applications.



--- click on create accelerator.

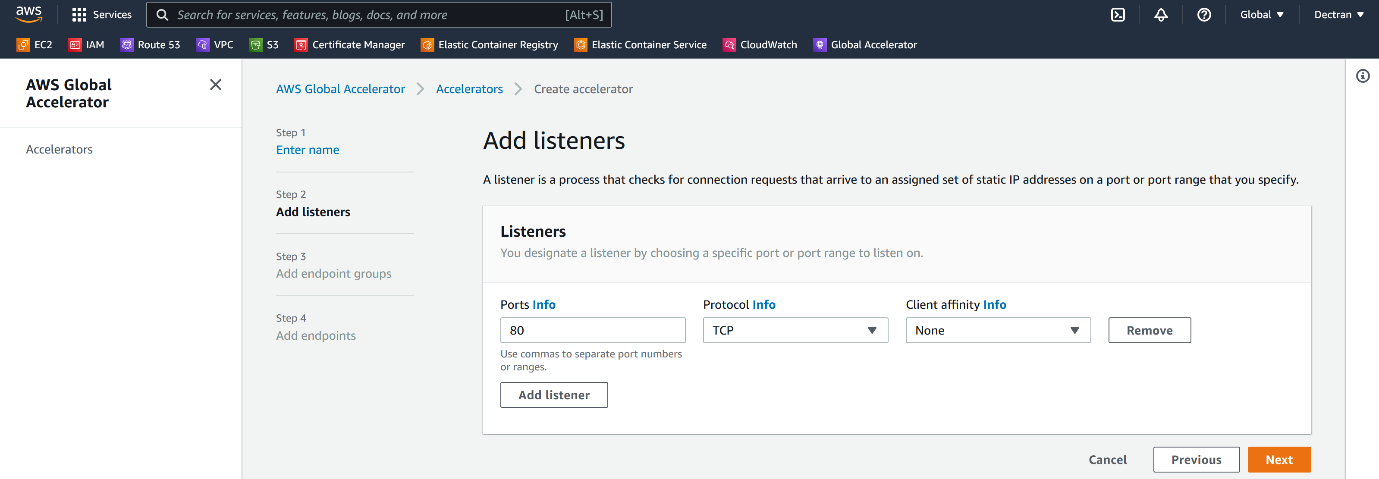
--- **add application load balancer as target.**



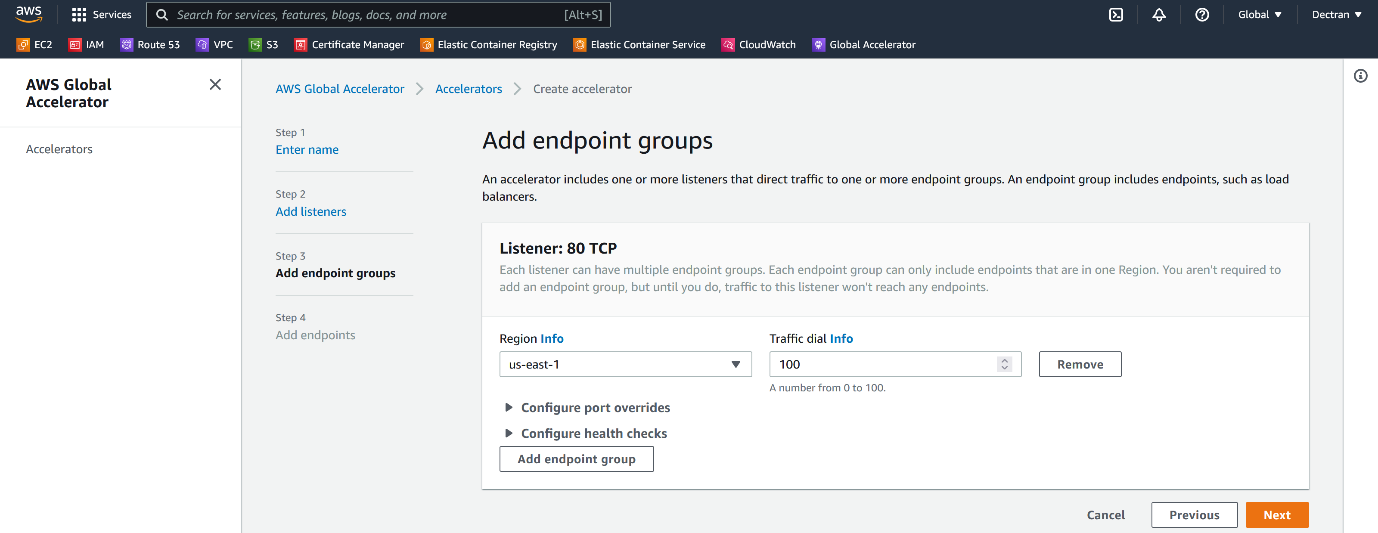


--- **note** – here I am routing traffic to application load balancer.

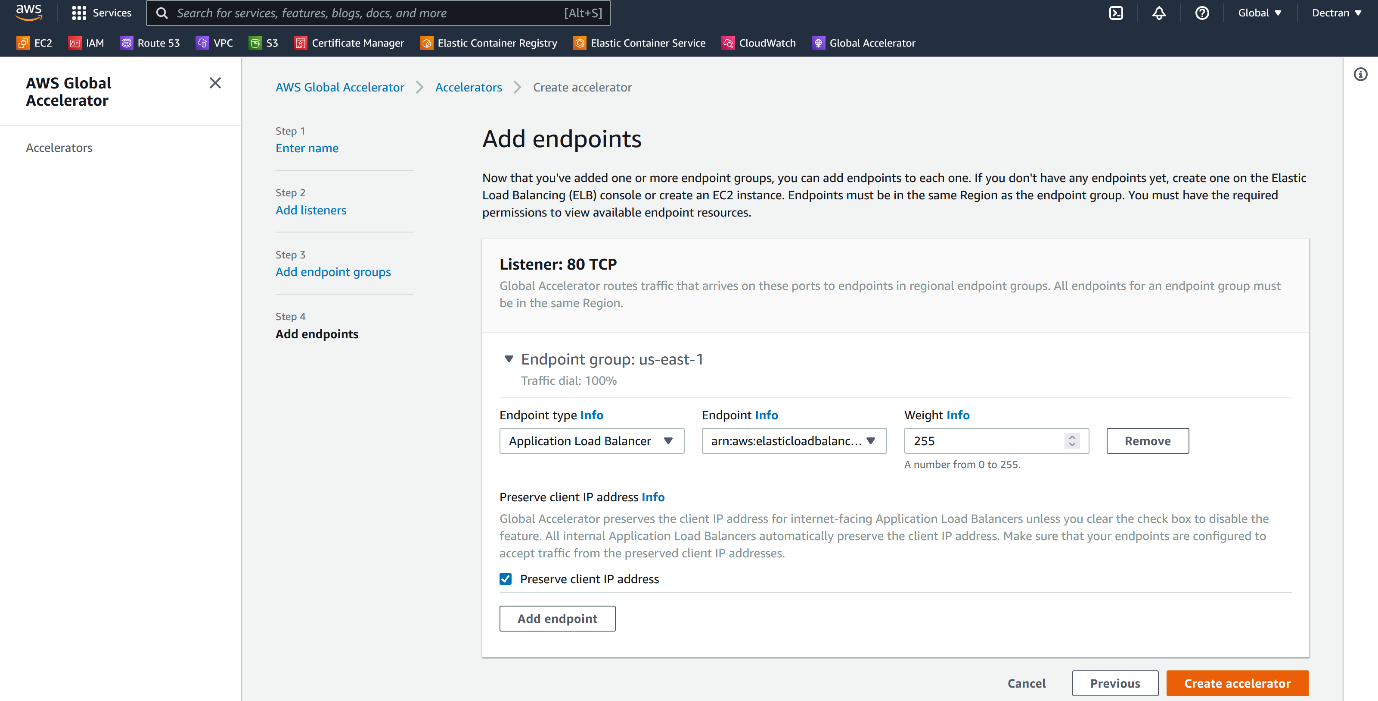
--- click on next.



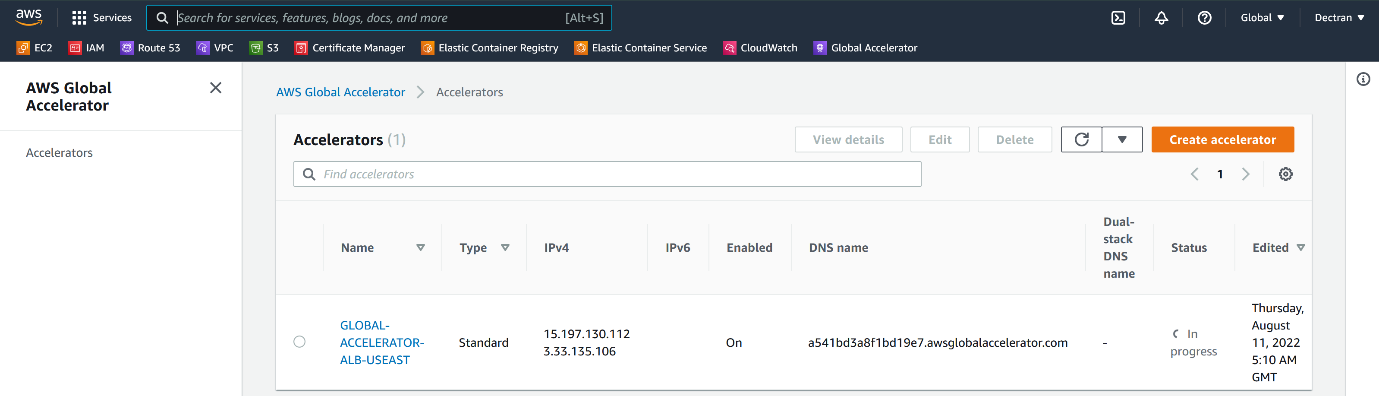
--- click on next. Please add 443 port as well.



--- click on next.



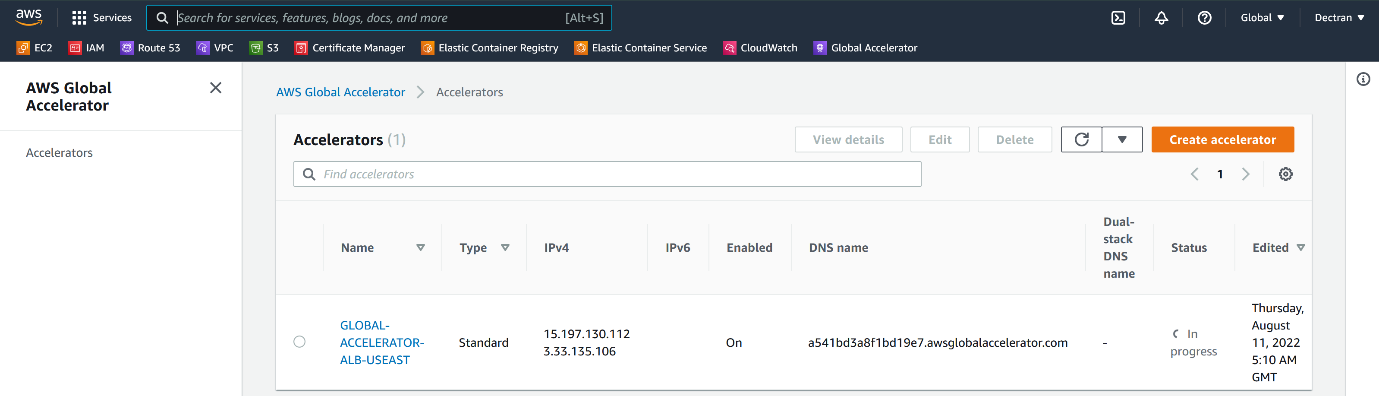
--- click on create accelerator.



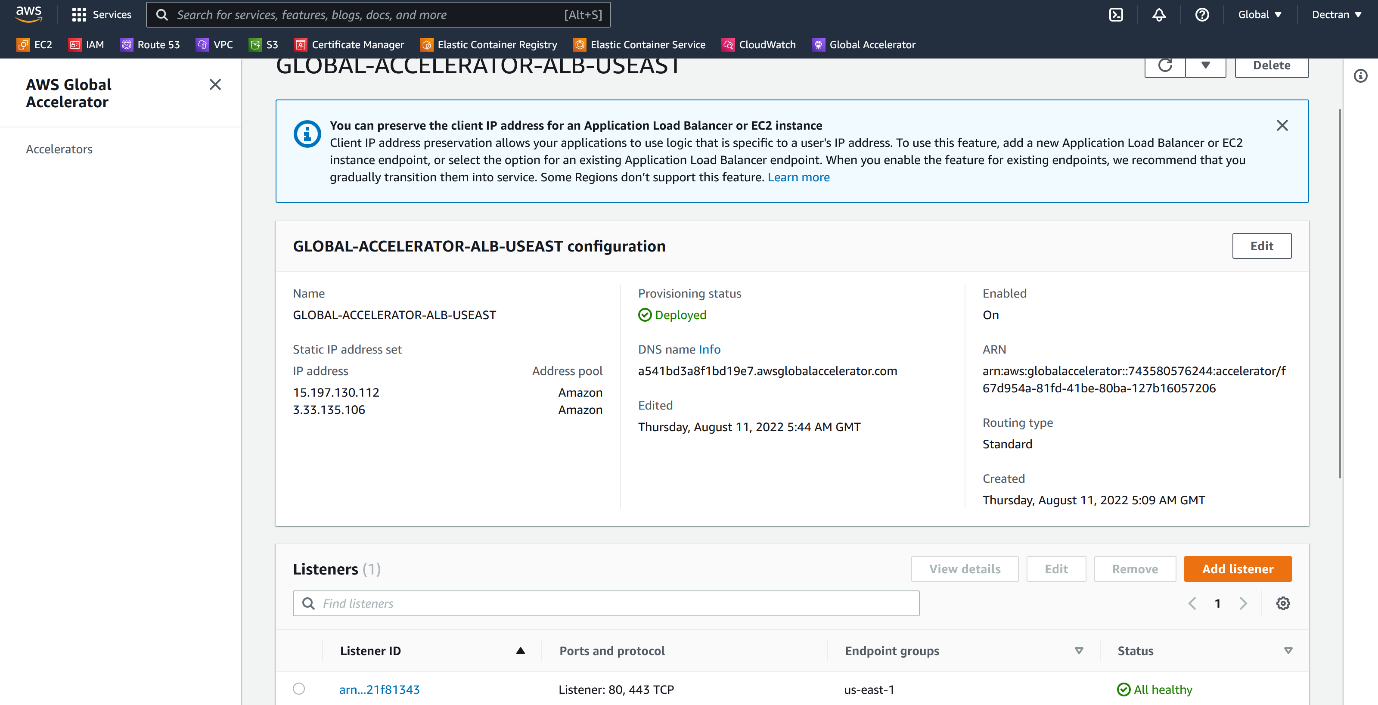
--- **NOTE** – we are successfully created global acceleratory for application load balancer as a target.

--- **important** – so, far I added only single target for global accelerator. We have other 2 service which is running on 2 differed regions. Now I will add those 2 services to global accelerator and distribute the traffic between these 3 targets.

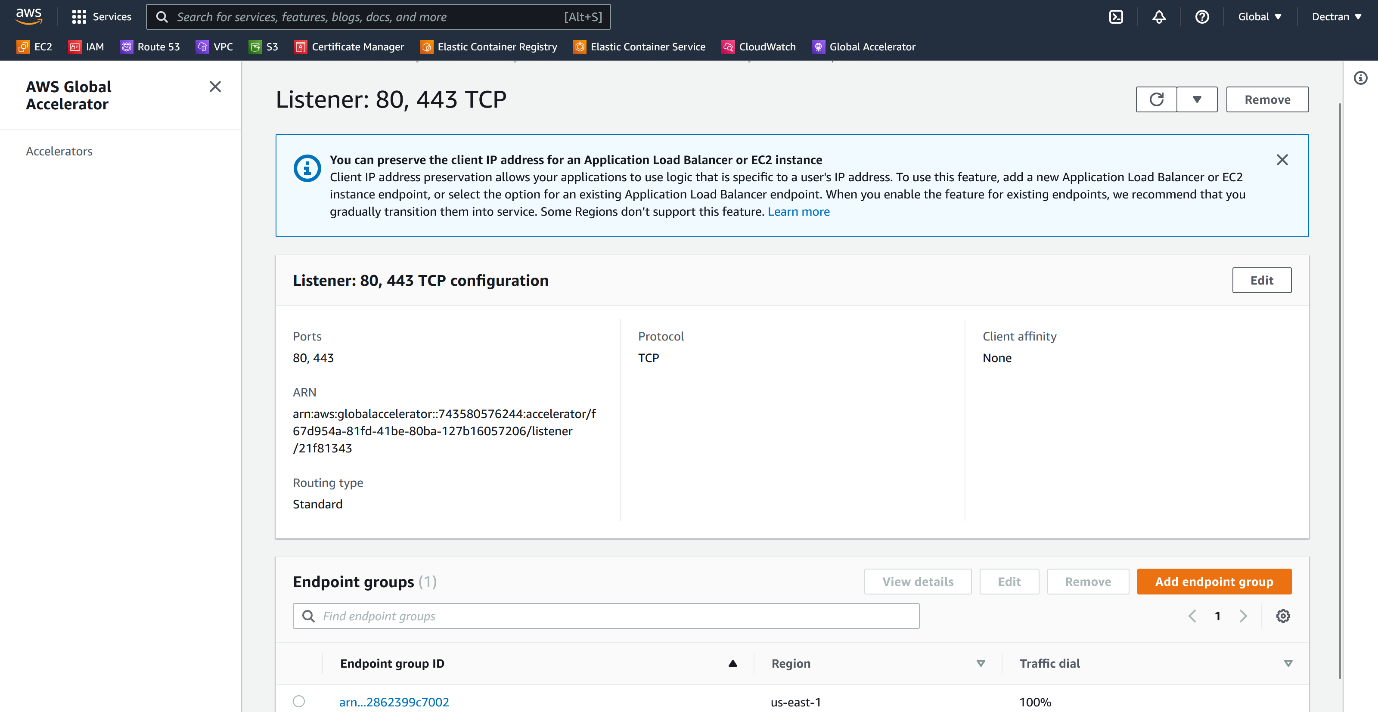
--- **note** – **add EC2 instance as target for global accelerator.**



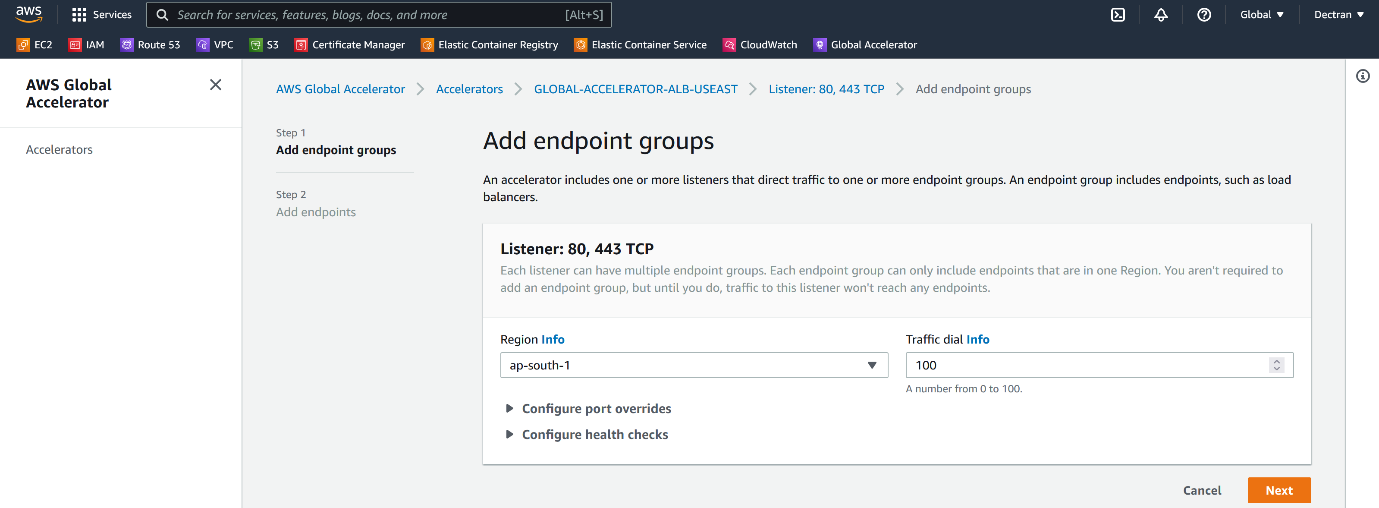
--- click on global accelerator.



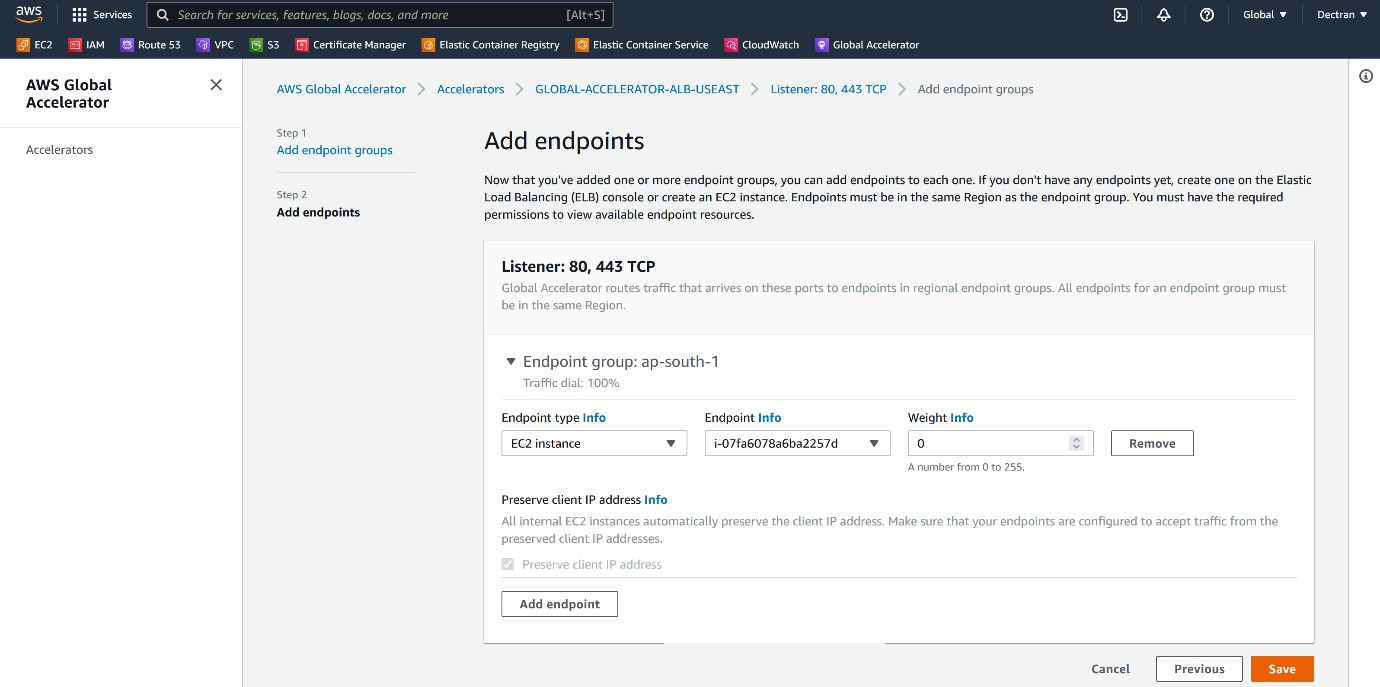
--- click on listener ID.



--- click on add endpoint group.



--- click on next.



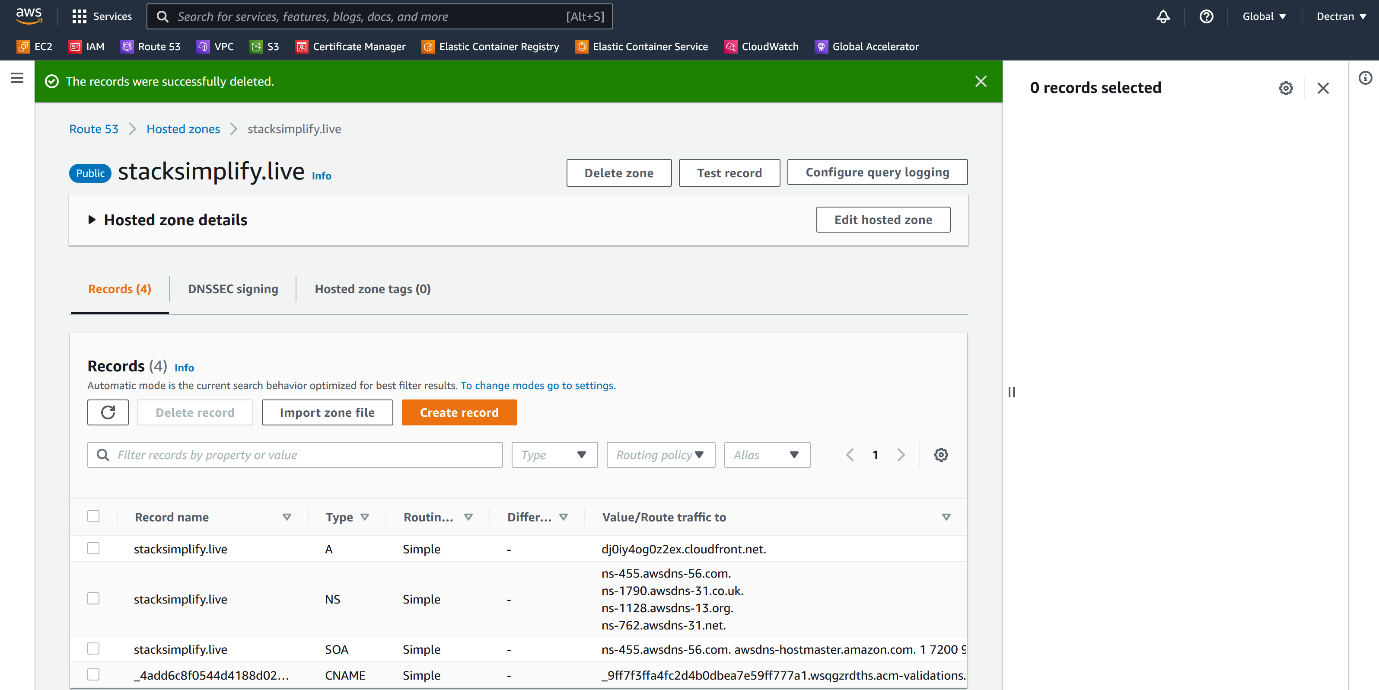
--- **important** – **weight** – based on your traffic requirement, you can give weight to this endpoint.

--- click on save.

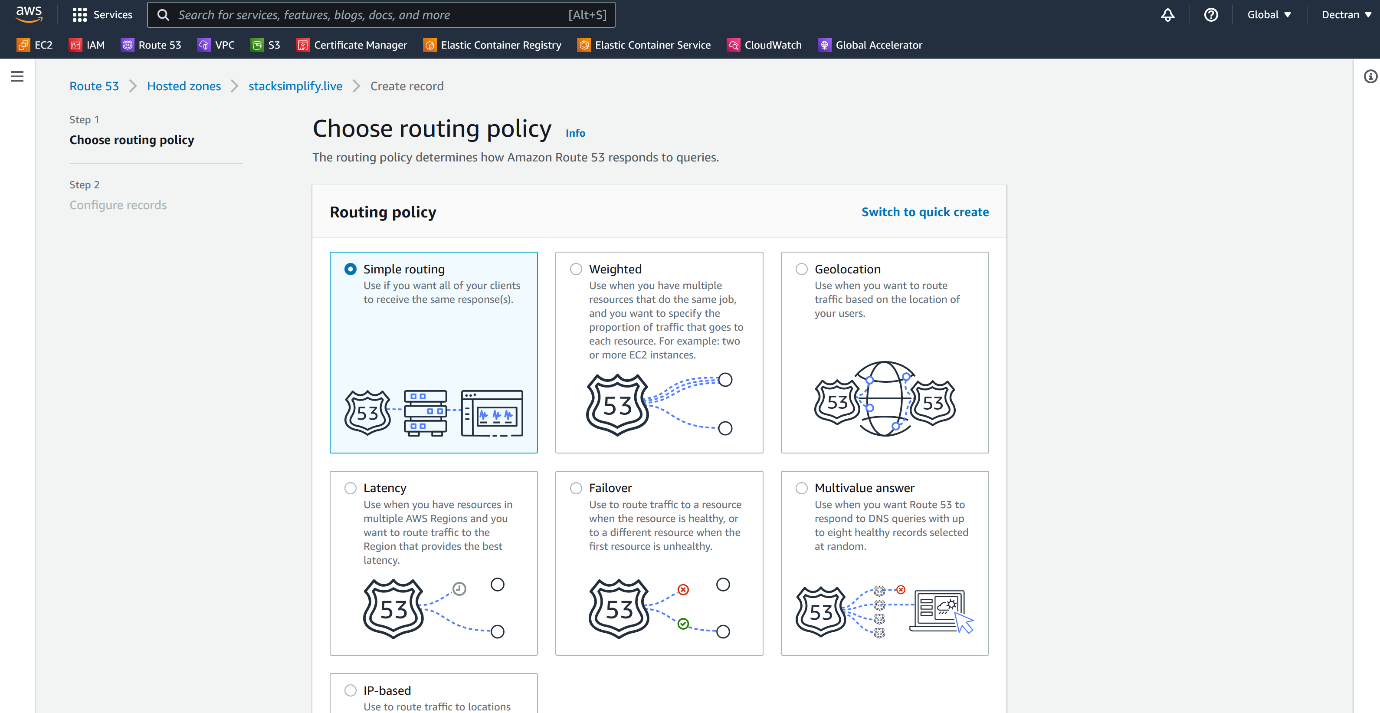
--- now I have added Mumbai end point, please add London endpoint as well.

**Add record in route53**

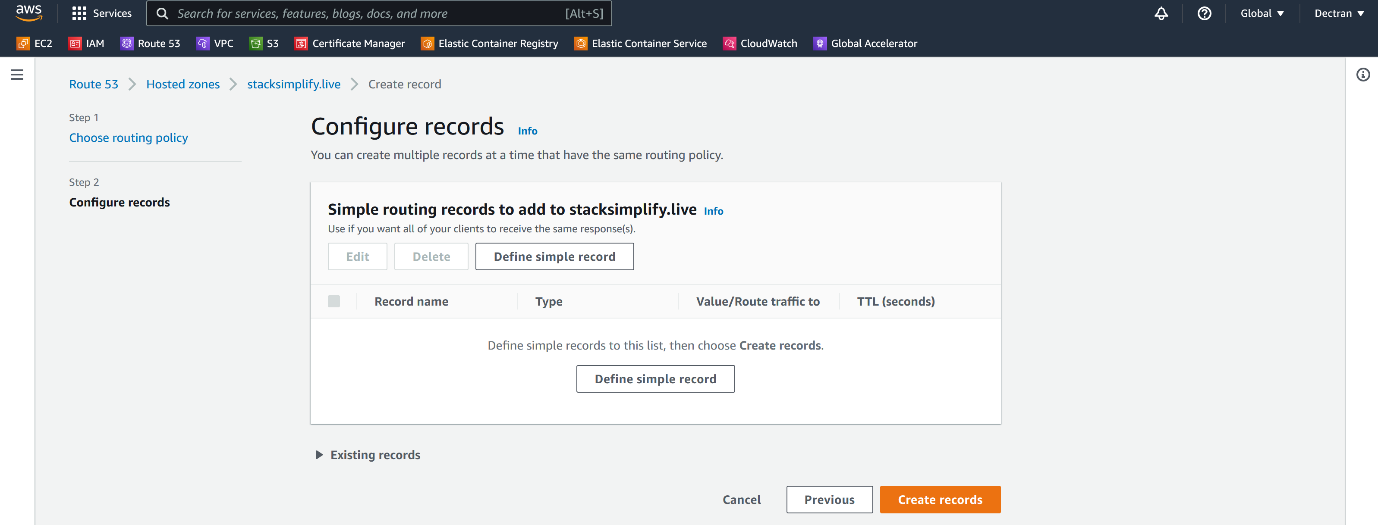
--- if we want, we can add record in route53 for global accelerator DNS name.



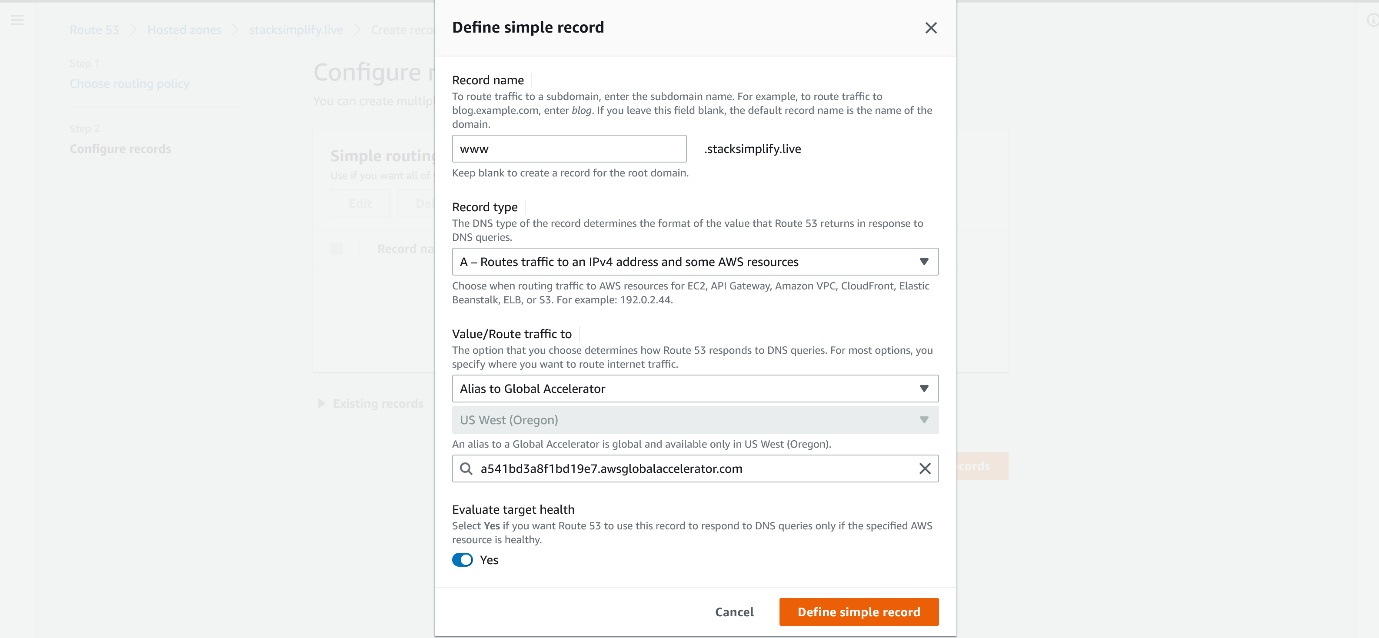
--- click on create record.



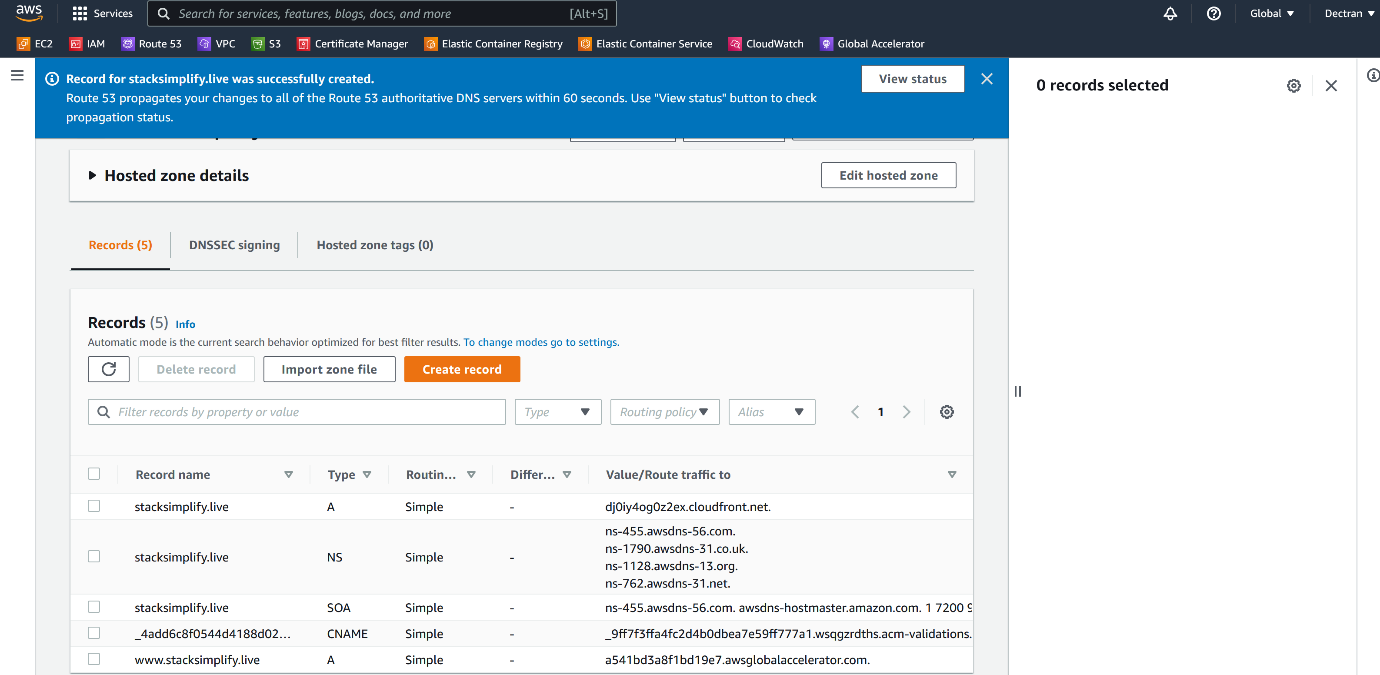
--- select simple routing and click on next.



--- click on create define simple record.



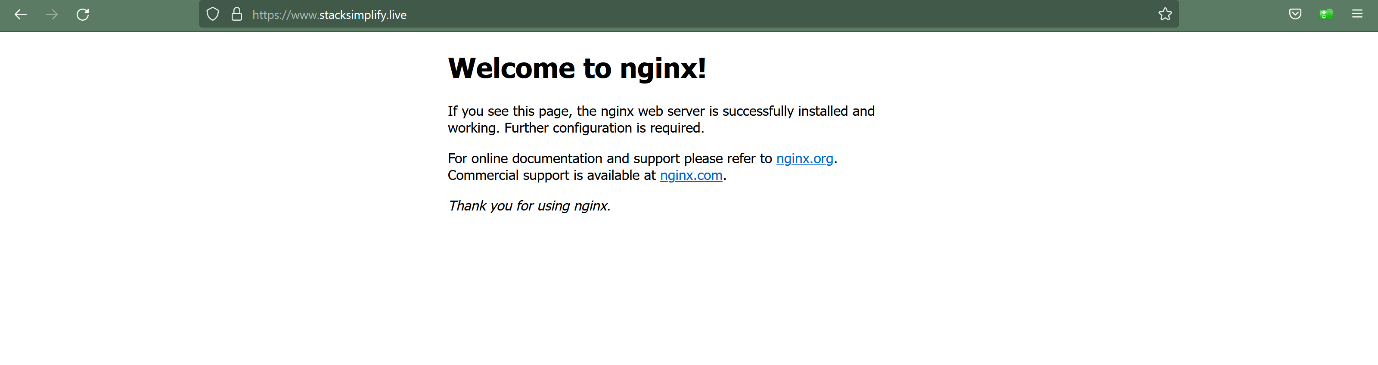
--- give the details like above and click on define simple record.



--- our record got created, it is a single point entry.

**Test application**

--- <https://www.stacksimplify.live>

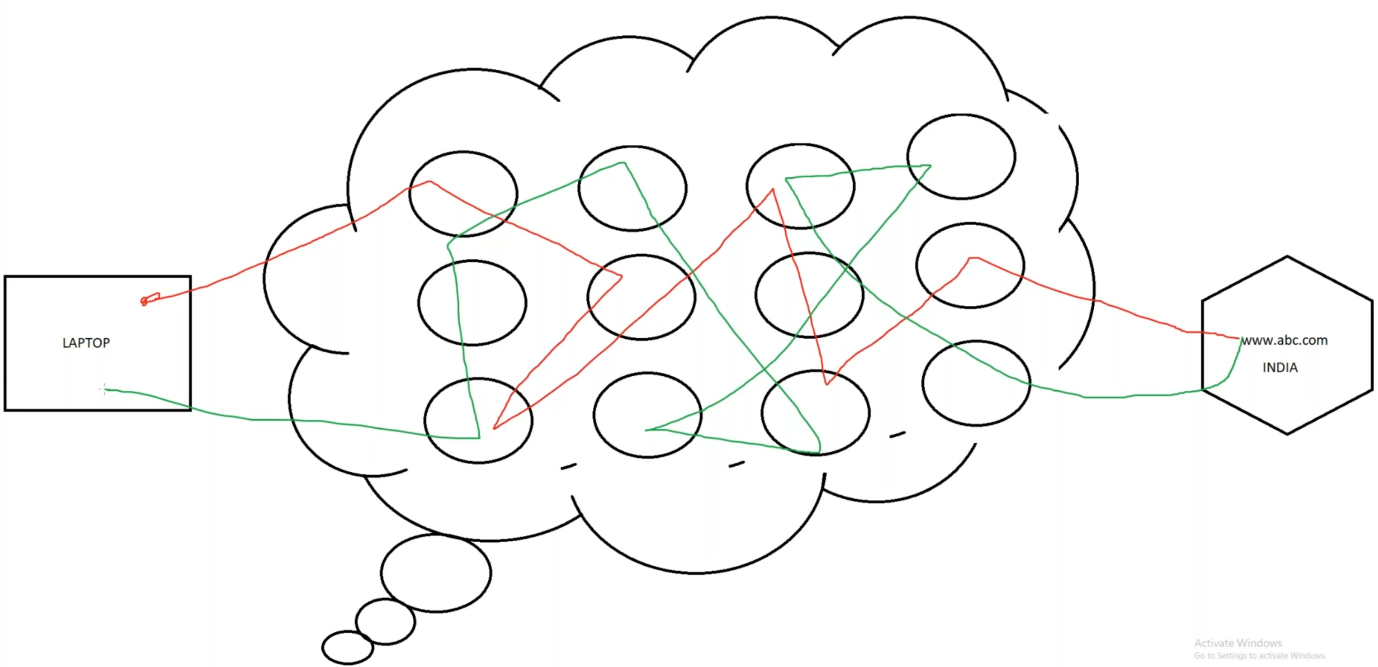


**Cloudfront – CDN (content delivery network)**

--- **what is CDN…?**

In order to improve our website, we need to cache all our website static content date. This is possible using cloudfront.

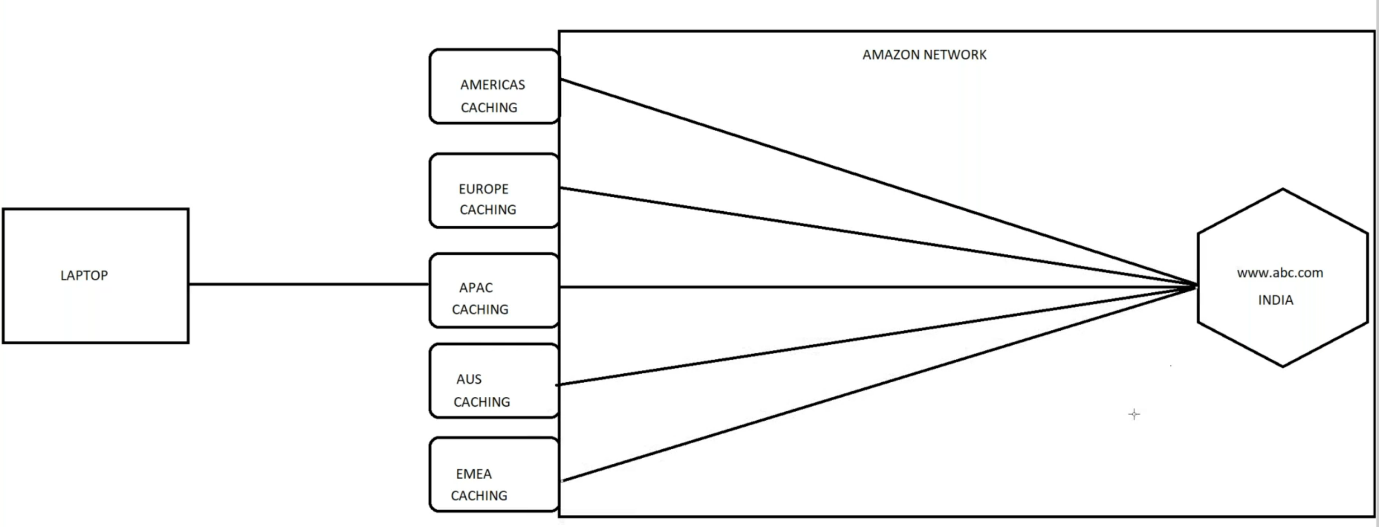
--- **Scenario** – we have website called [www.abc.com](http://www.abc.com), this website servers are present in India. When you trying to connect this website from laptop, the request 1st goes to the networking devices which are present on the internet. From networking devices, the request will reach to domain. The response from site will reaches to the networking devices, from there it will reaches to our laptop.



--- **note** – this will increase the latency. We need to lower the latency time. For that we will use CDN, this will decrease the latency.

--- in cloudfront, we have point of present in multiple regions. In every major country they have **point of presence**.

--- we have a website, in that website, we have some static components are present, those components are changing for every 7 or 15 0r 30…etc. then we can cache them in **point of presence**.



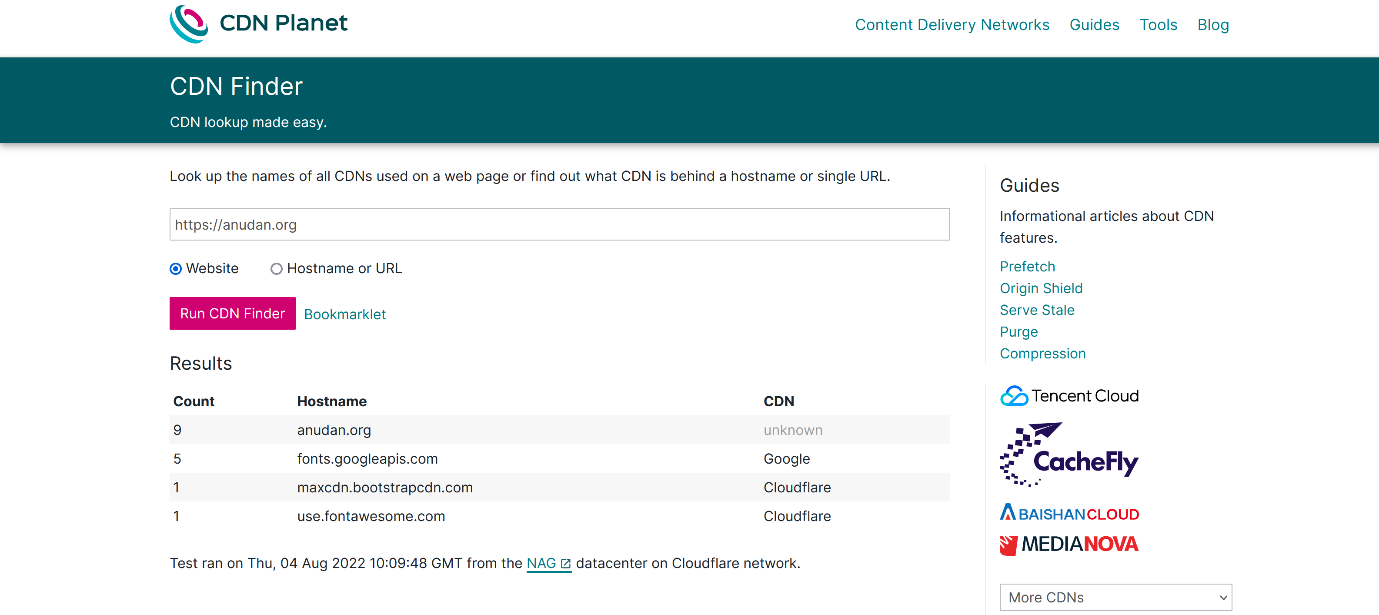
--- even you have cached your website in multiple point of presence. All the point of presence will connect to your domain.

**CDN finder**

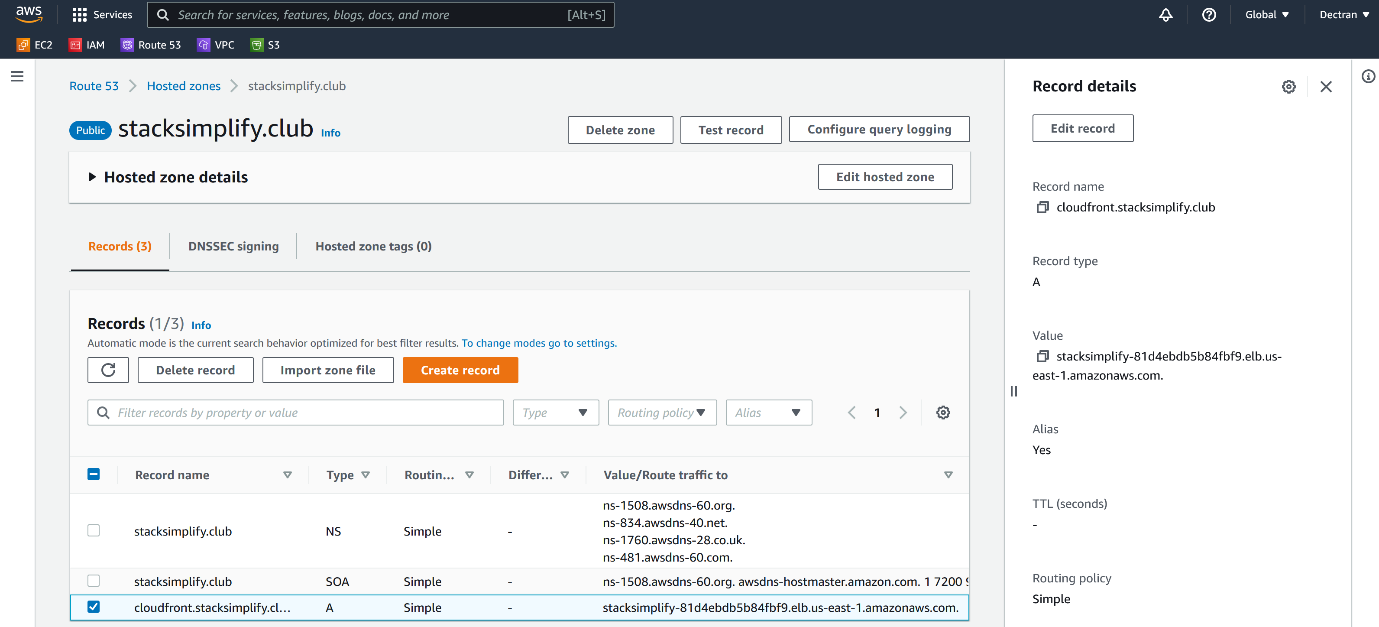
--- Reference - <https://www.cdnplanet.com/tools/cdnfinder/>

--- this tool will help us to check whether the website is using CDN or not.

--- akamai and cloud fare are famous cloudfront.



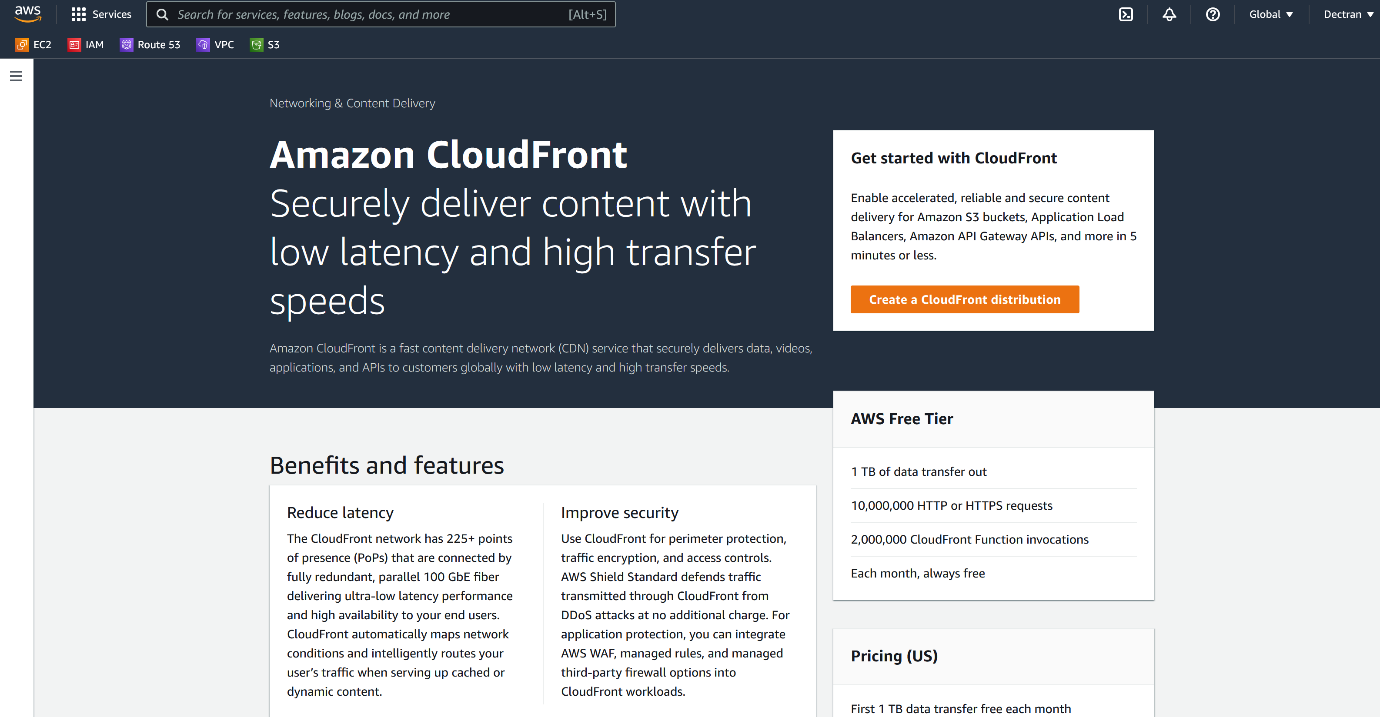
**Route53**



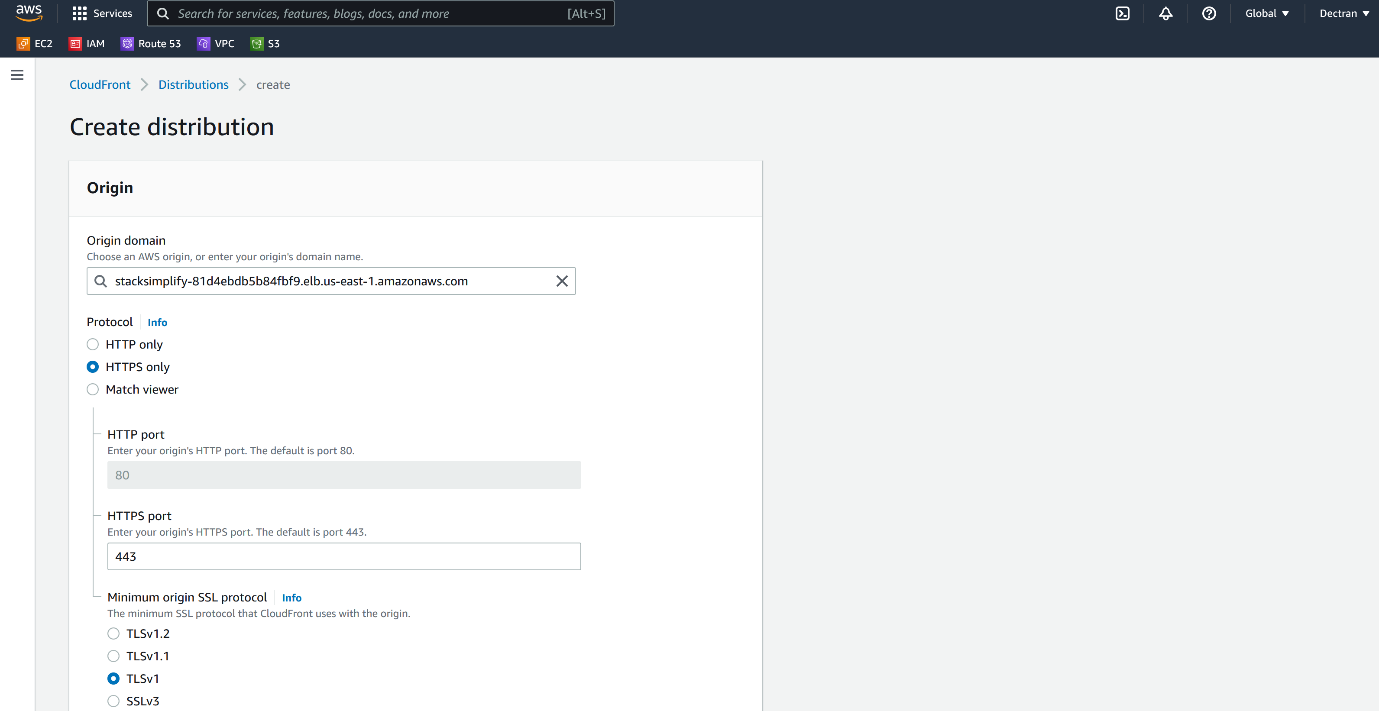
--- **note** – our application is load balanced using application load balancer and I created record for our application**. cloudfront.stacksimplify.club**.

--- in front of application load balancer, I am placing cloudfront.

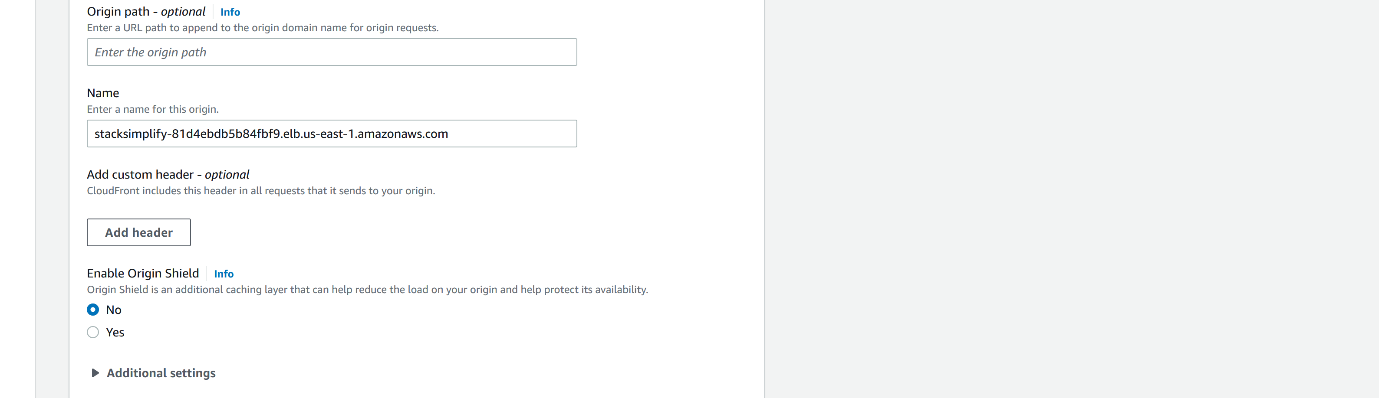
--- go to the cloudfront.

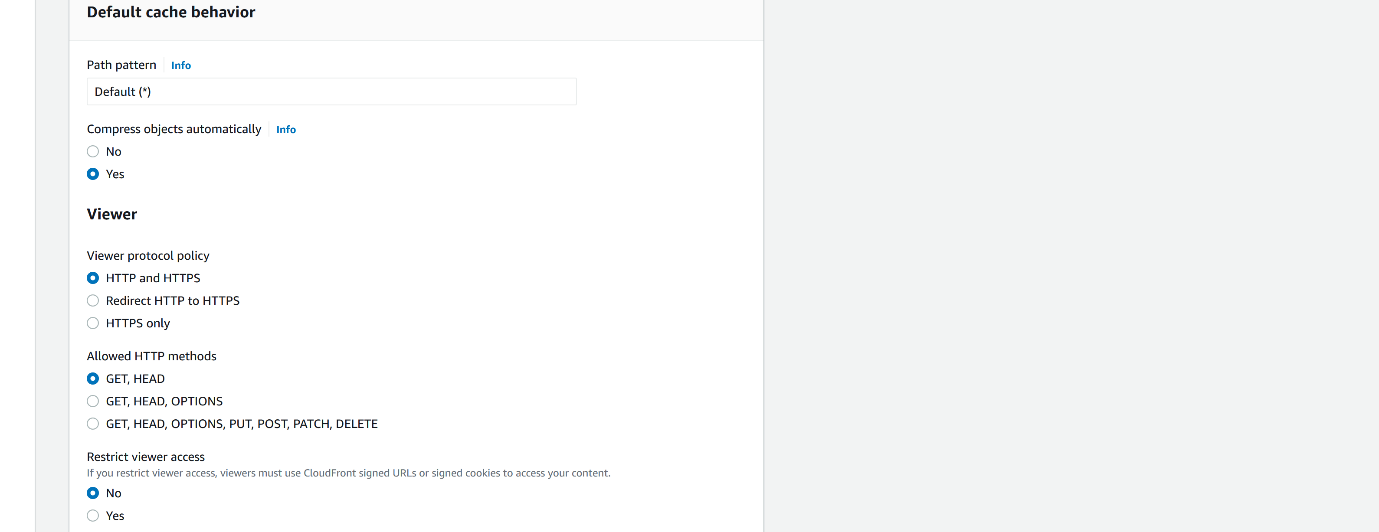


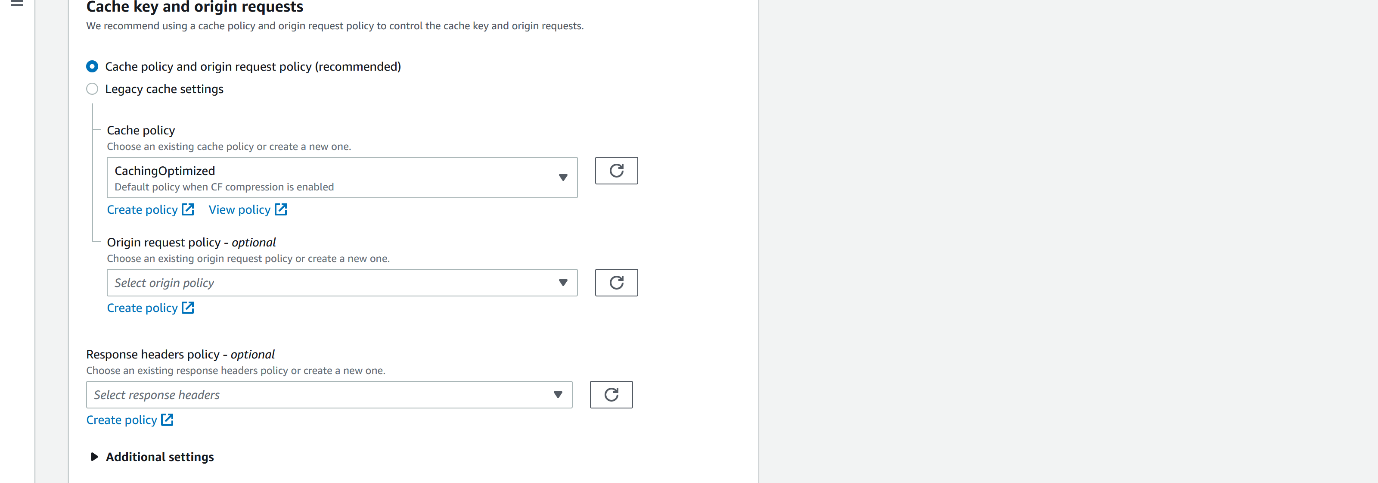
--- click on create a cloudfront distribution.

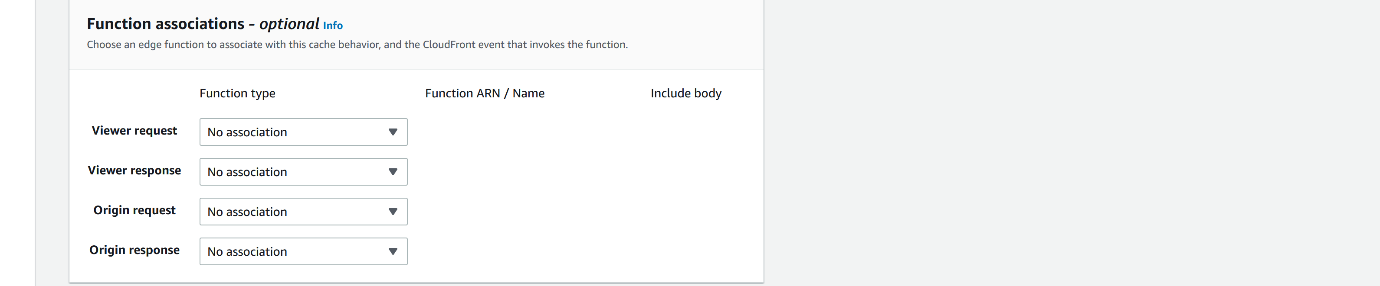


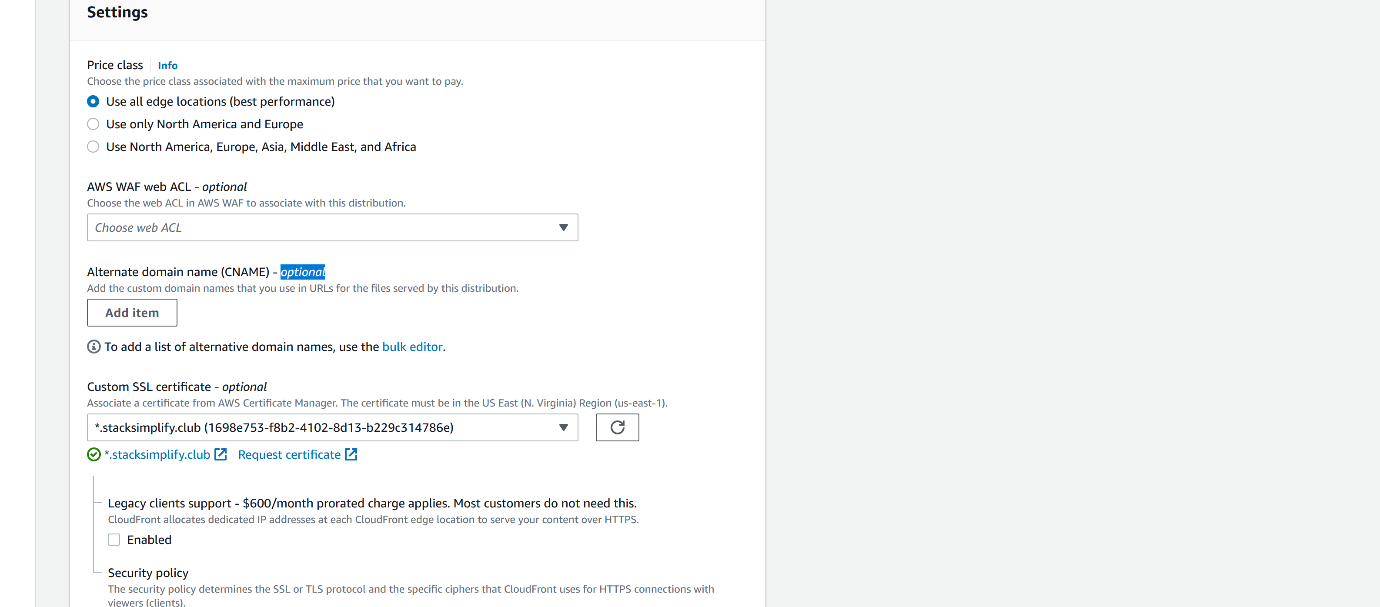
--- please select our load balancer here, if we have hosted website on s3 then select our s3 bucket here.



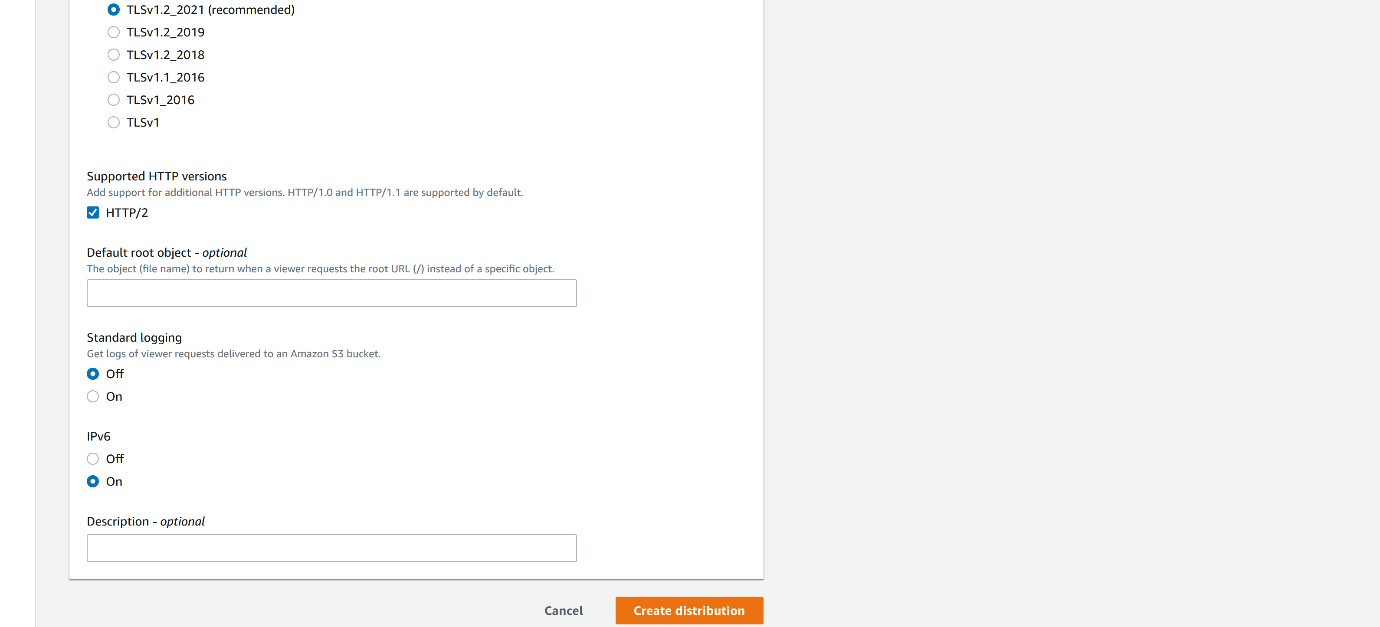




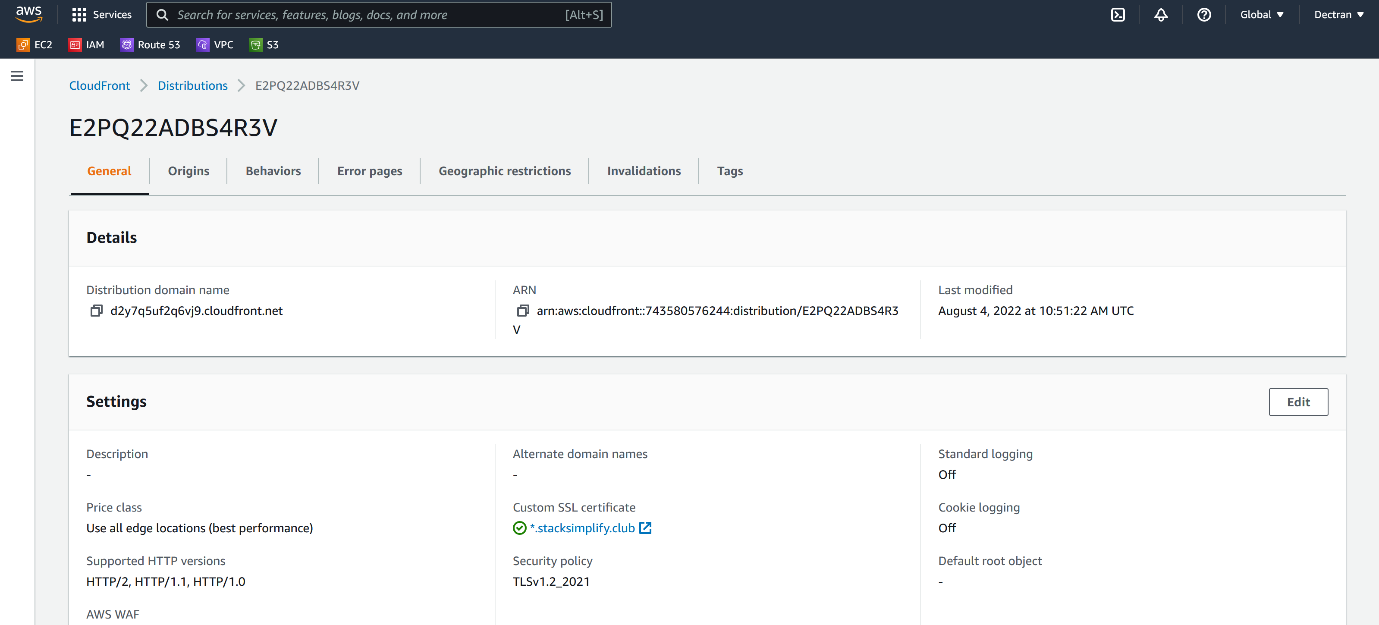




--- choose required edge locations and select the certificate. You can also provide alternate domain name.



--- click on create distribution.



--- **note** - our cloudfront is deployed, here we can upload our static data like html, jpg so when a user tries to access this static data on internet, the latency for user request is very low because this static data come from point of presence.