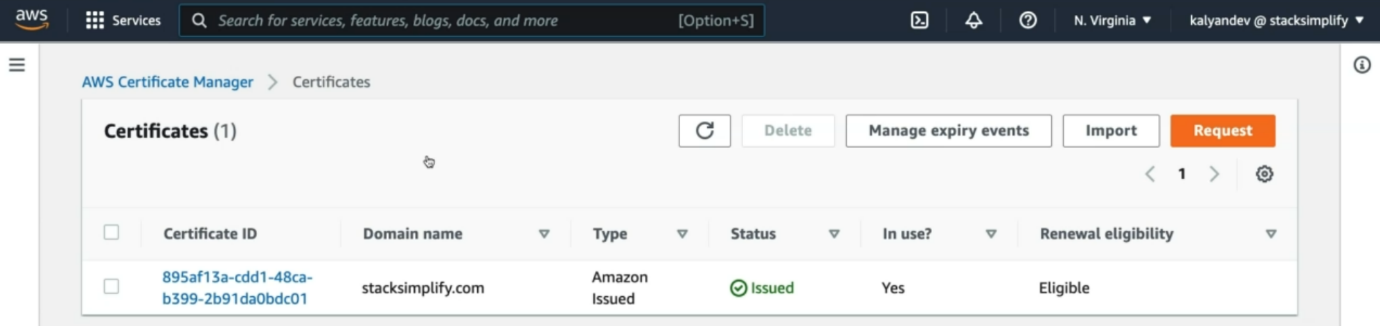
**3. Create SSL Certificate in AWS Certificate Manager**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/08-NEW-ELB-Application-LoadBalancers/08-04-ALB-Ingress-SSL>

**Create an SSL Certificate in Certificate Manager**

--- Pre-requisite: You should have a registered domain in Route53

--- Go to Services -> Certificate Manager -> Create a Certificate

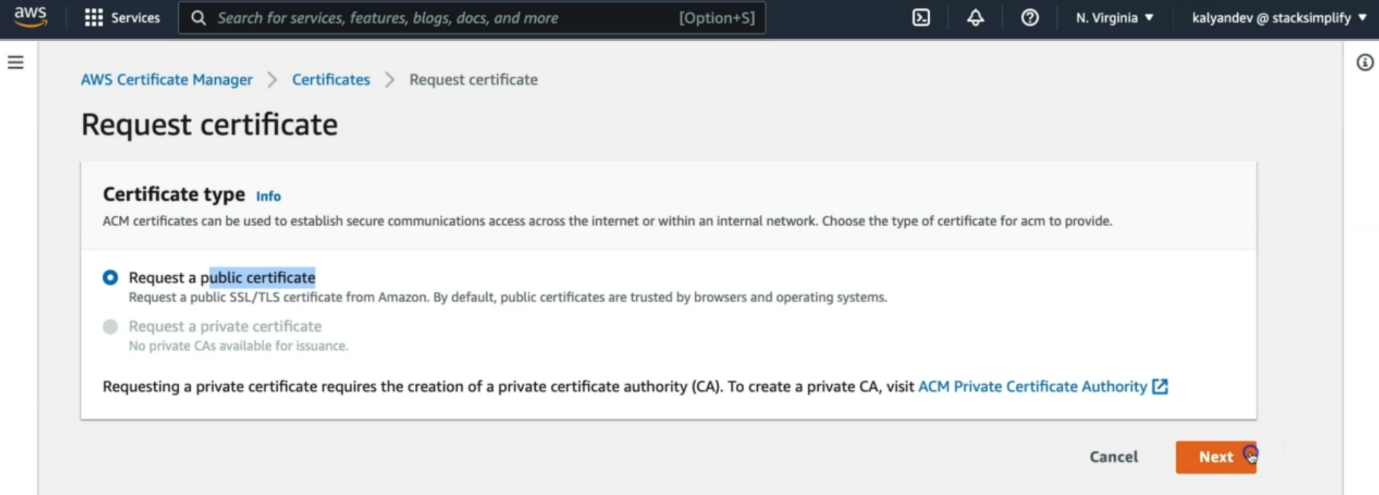


--- **note** – here we will create wild card ssl certificate, which means it will work with any subdomain name.

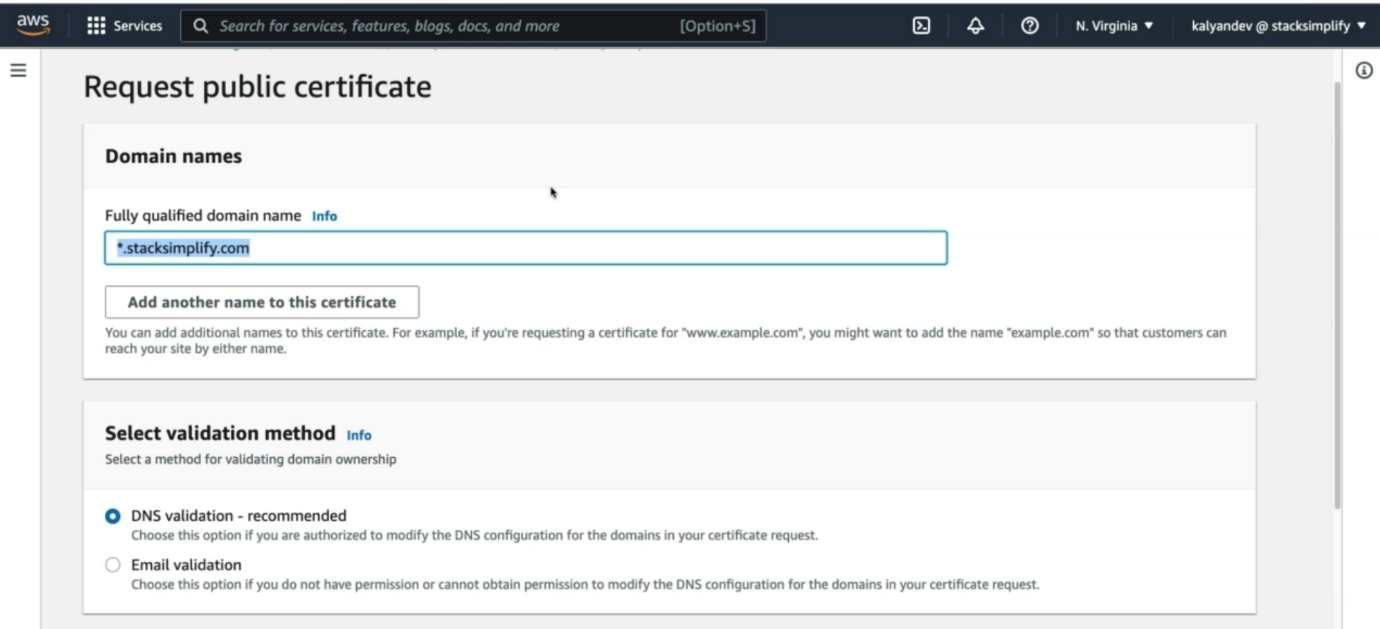
--- **note** - if give like this **\*.stacksimplify.com** then any subdomain with ssl certificate will works.

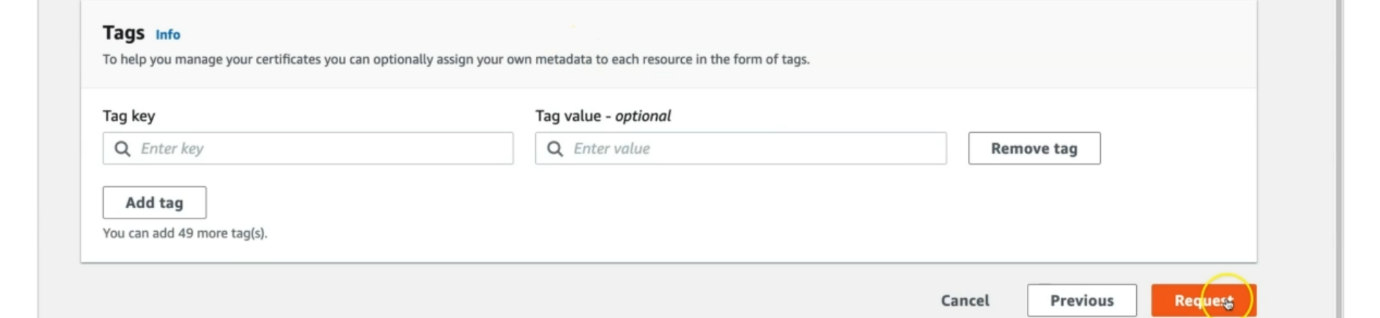
--- Click on Request a Certificate

1. Choose the type of certificate for ACM to provide: Request a public certificate
2. Add domain names: \*.yourdomain.com (in my case it is going to be \*.stacksimplify.com)
3. Select a Validation Method: DNS Validation
4. Click on Confirm & Request

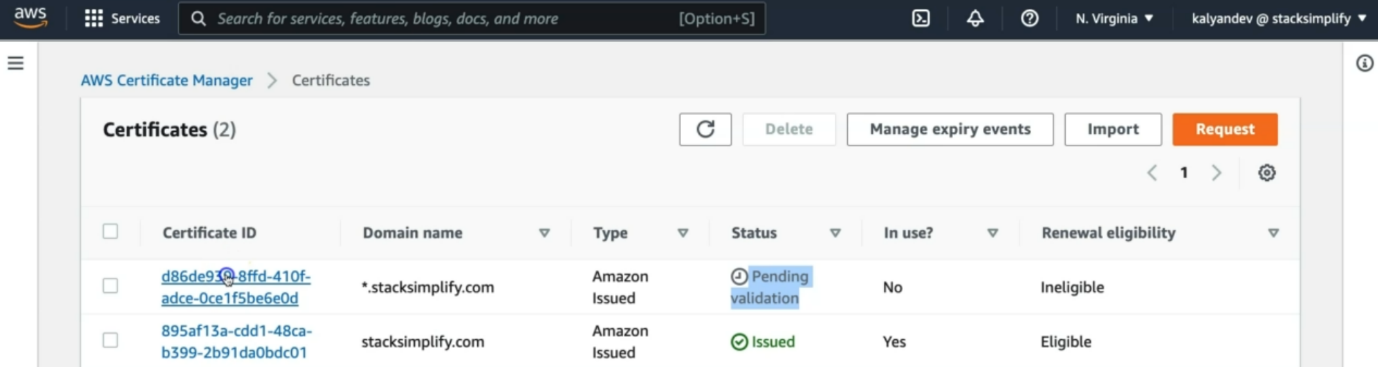


--- click on next

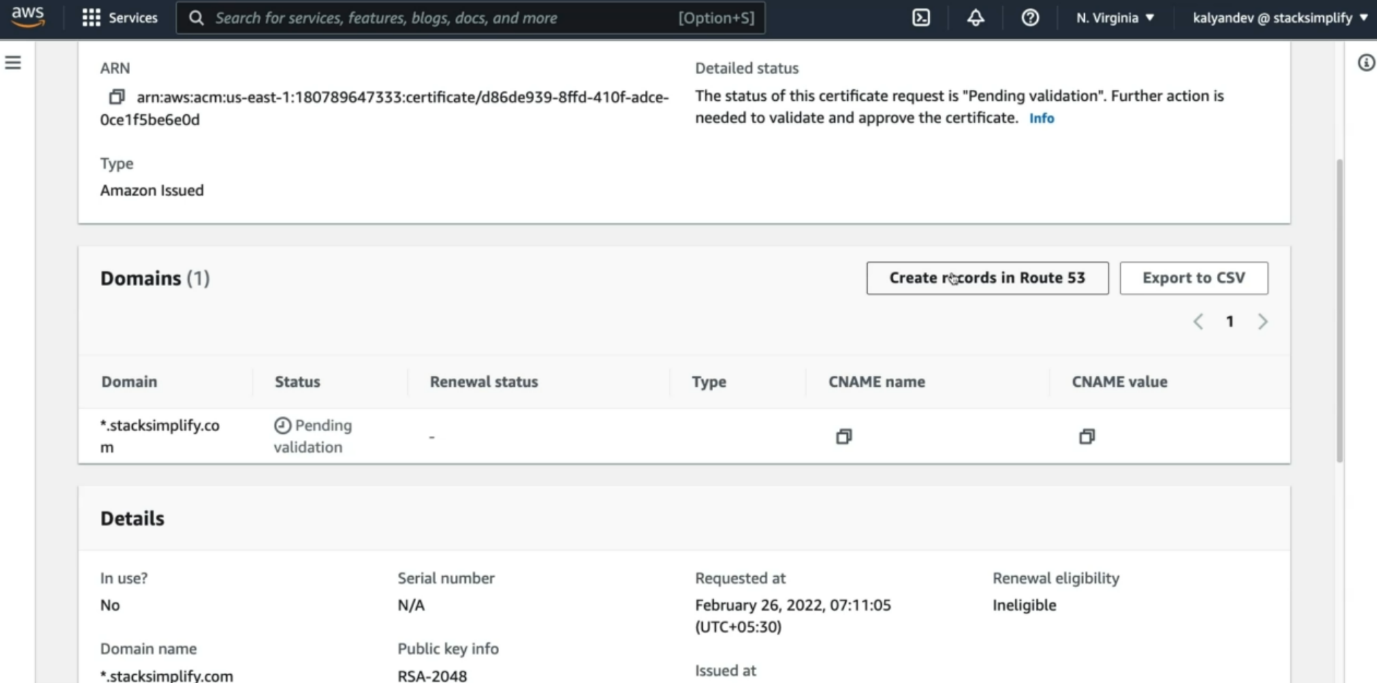




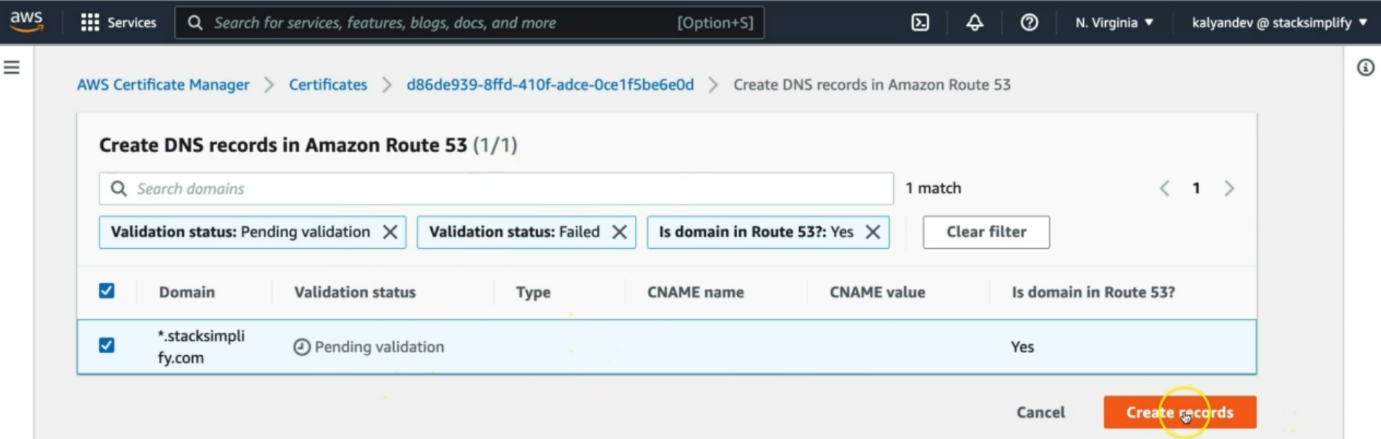
--- click on request.



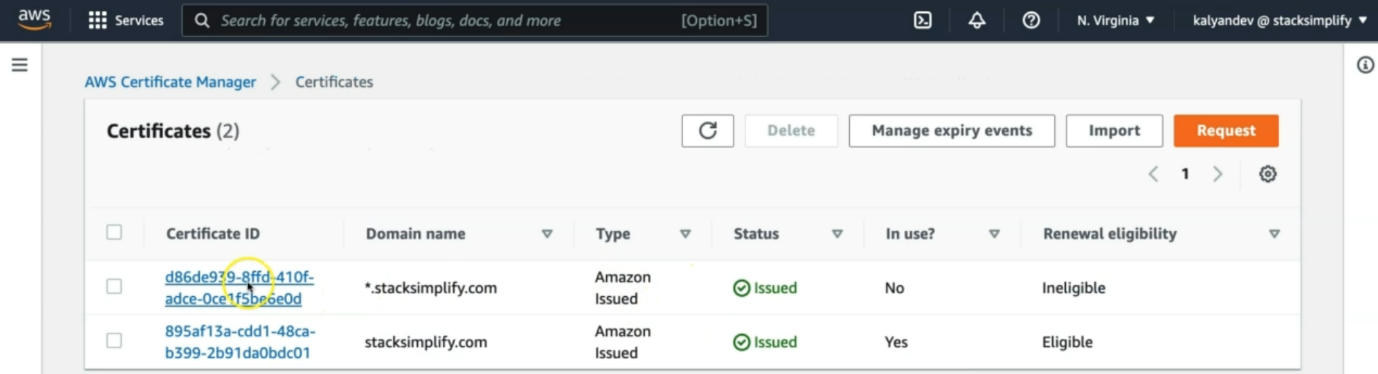
--- the certificate is created and it is pending state, click on certificated ID.



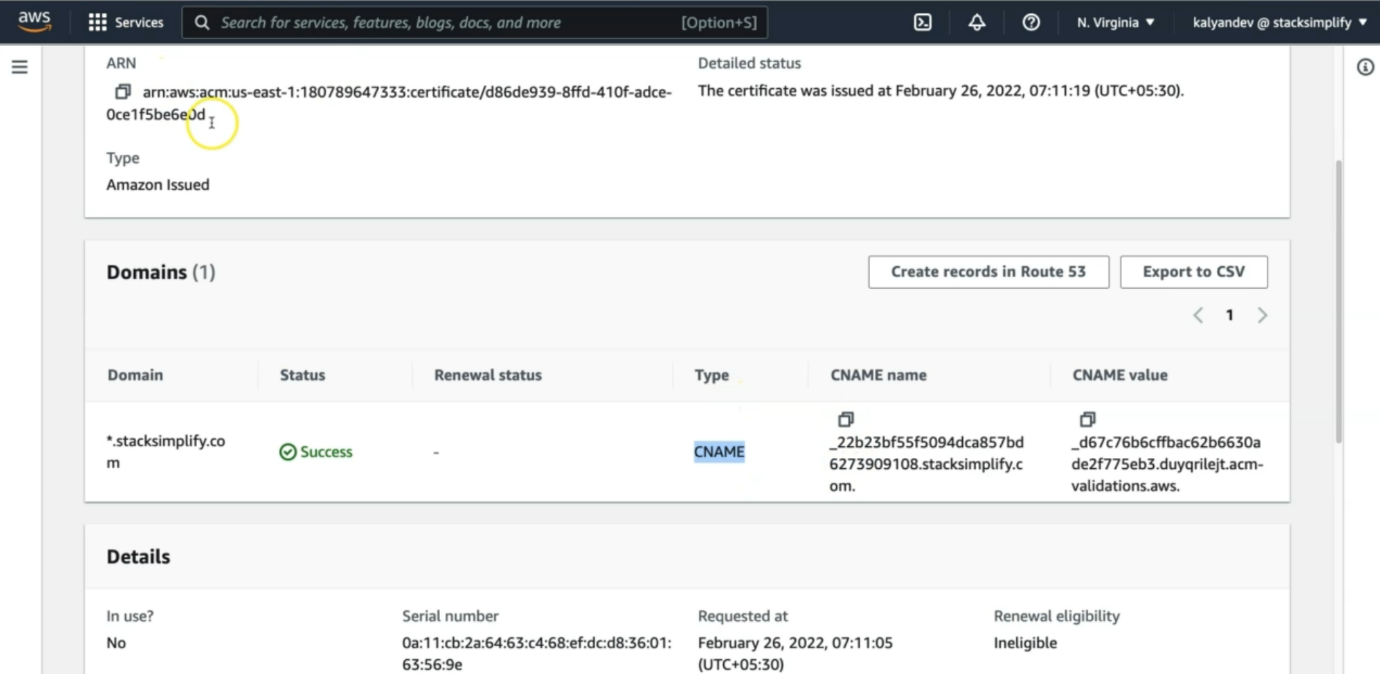
--- **note** – you can see here, we need to create record set in route53 to validate the ssl certificate. So, click on create records in Route 53 option.



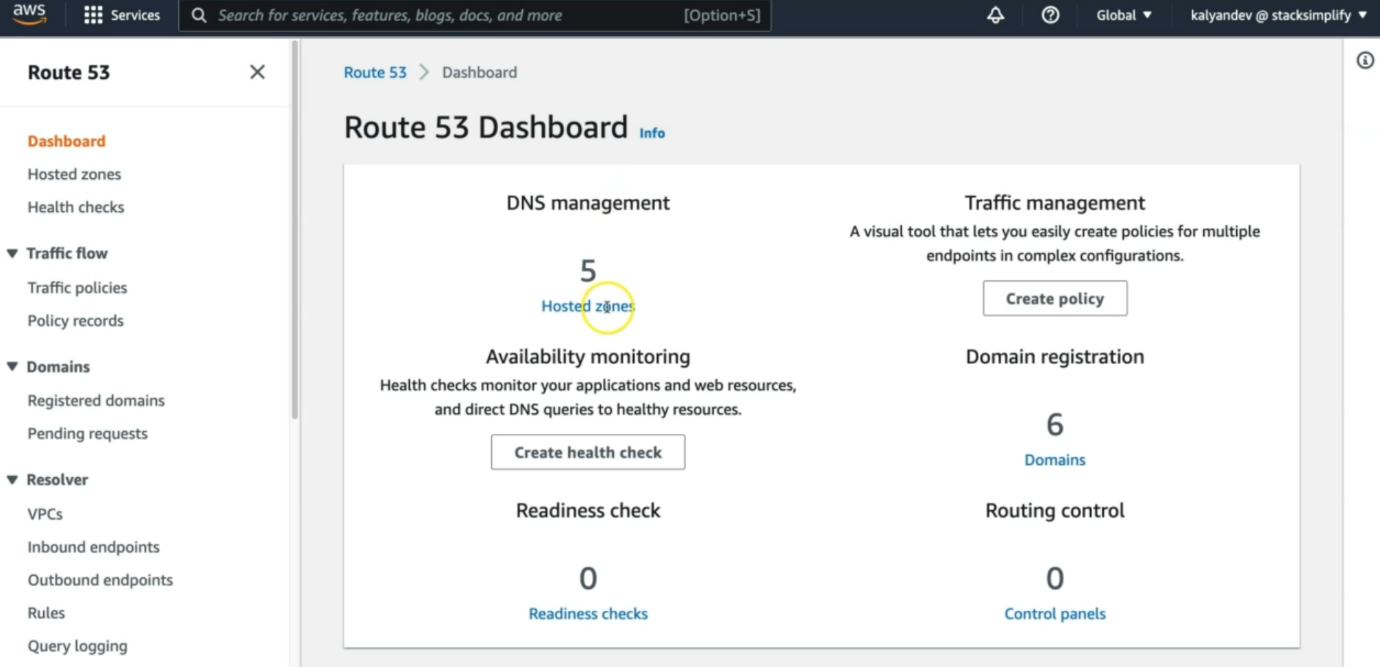
--- it will automatically add records in route 53, we need to click on create records.



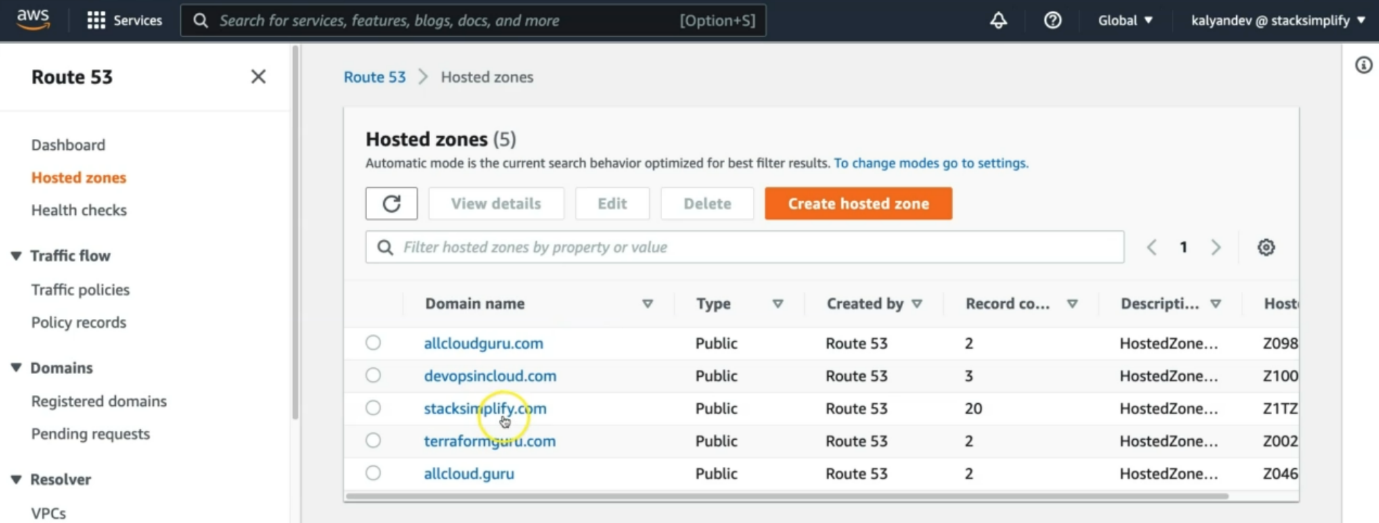
--- the record is successfully added in route53. Click on the certificate id.



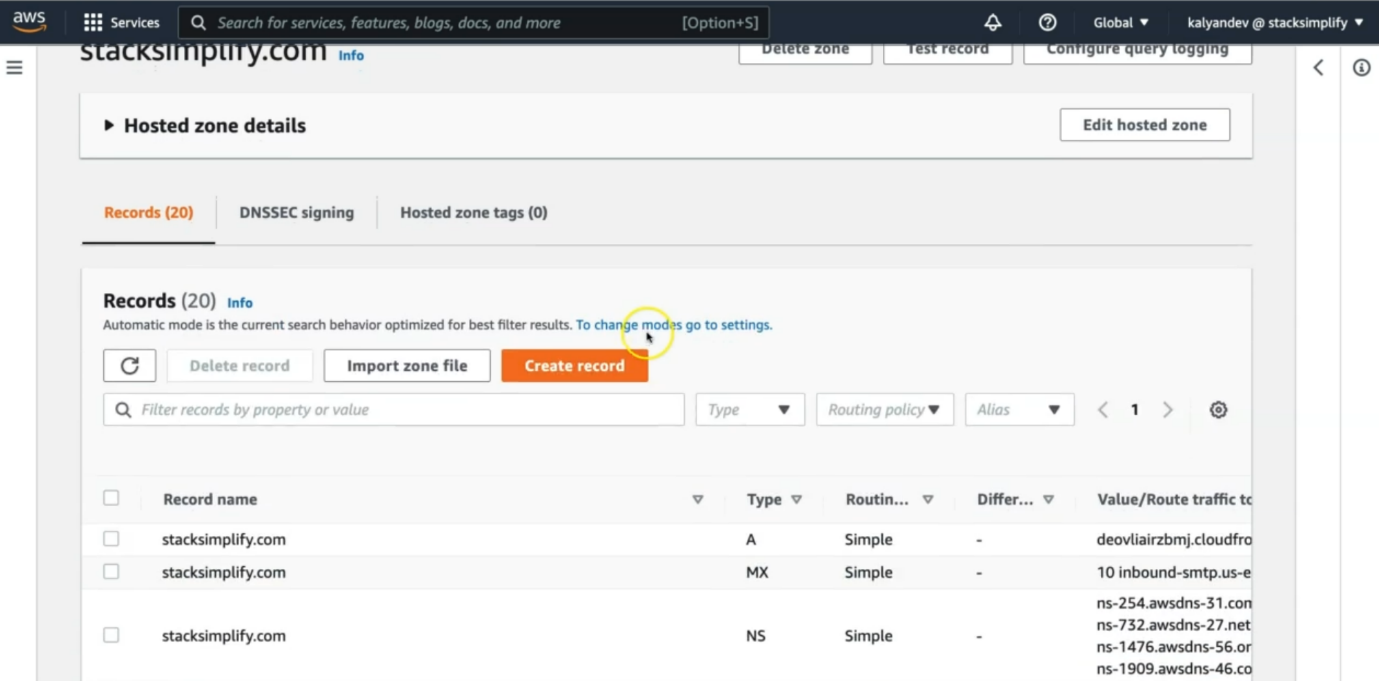
--- it created CNAME with CNAME name. this CNAME name id should match with route53, so , go to route53 verify the same.



--- go to the hosted zones,



--- go inside of stacksimplify.com



--- search for the CNAME name and this CNAM name values should match with IAM certificate CNAME name value.

--- **note** - Wait for 5 to 10 minutes and check the Validation Status

--- this completes the creation of ssl certificate.