**2. Create IAM Policy, k8s Service Account, IAM Role and Verify**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/08-NEW-ELB-Application-LoadBalancers/08-06-Deploy-ExternalDNS-on-EKS>

**Introduction**

--- External DNS: Used for Updating Route53 Record Sets from Kubernetes

--- We need to create IAM Policy, k8s Service Account & IAM Role and associate them together for external-dns pod to add or remove entries in AWS Route53 Hosted Zones.

--- Update External-DNS default manifest to support our needs

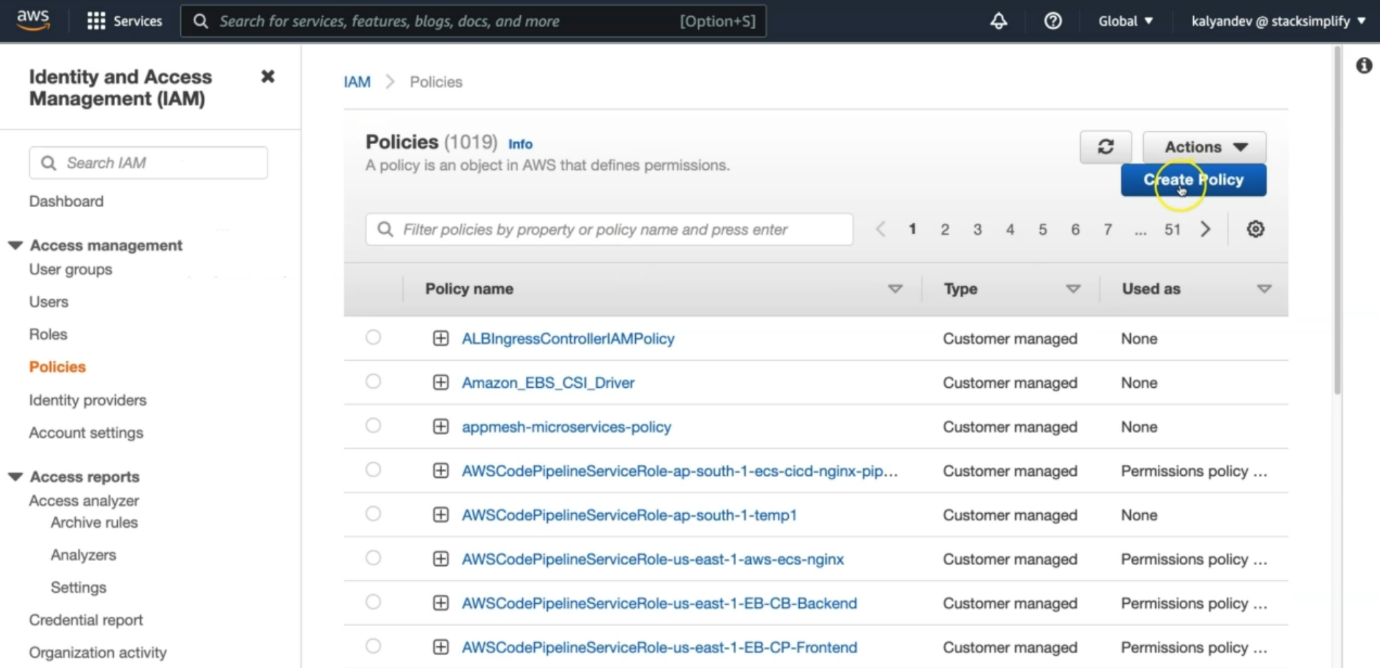
--- Deploy & Verify logs

**Create IAM Policy**

--- This IAM policy will allow external-dns pod to add, remove DNS entries (Record Sets in a Hosted Zone) in AWS Route53 service

--- Go to Services -> IAM -> Policies -> Create Policy

1. Click on JSON Tab and copy paste below JSON
2. Click on Visual editor tab to validate
3. Click on Review Policy
4. Name: AllowExternalDNSUpdates
5. Description: Allow access to Route53 Resources for ExternalDNS
6. Click on Create Policy

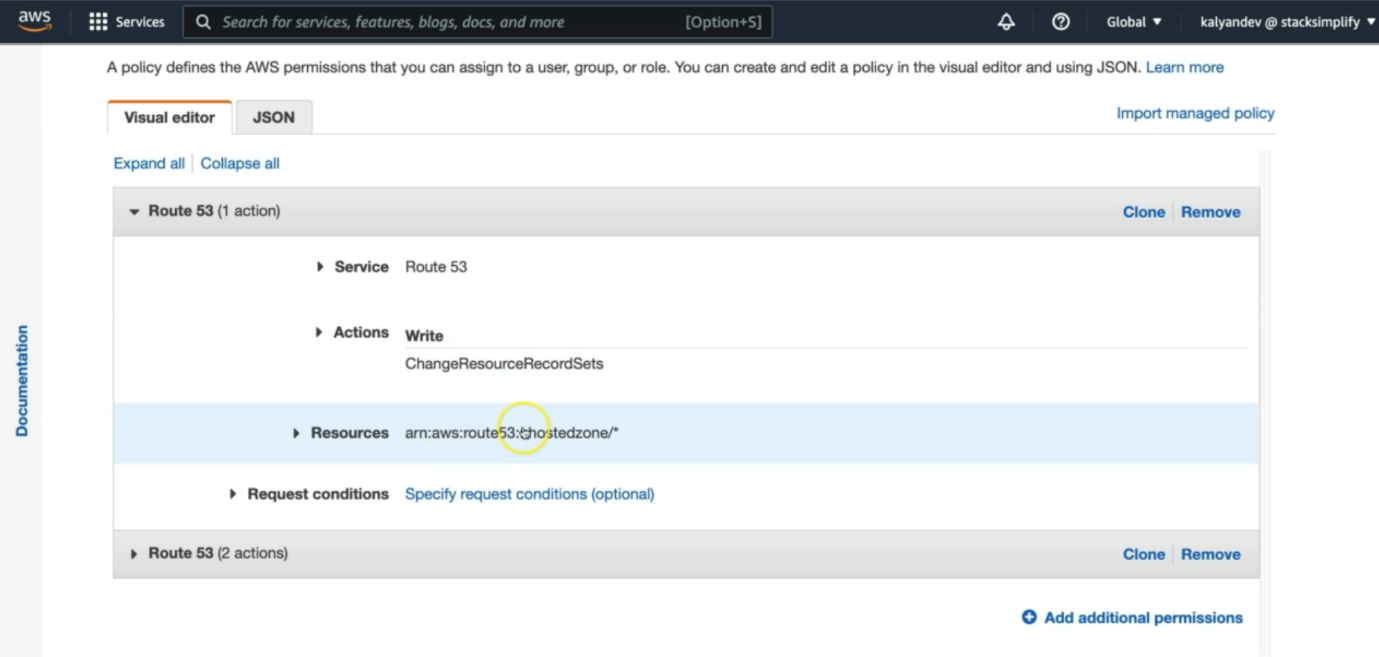


--- click on create policy.

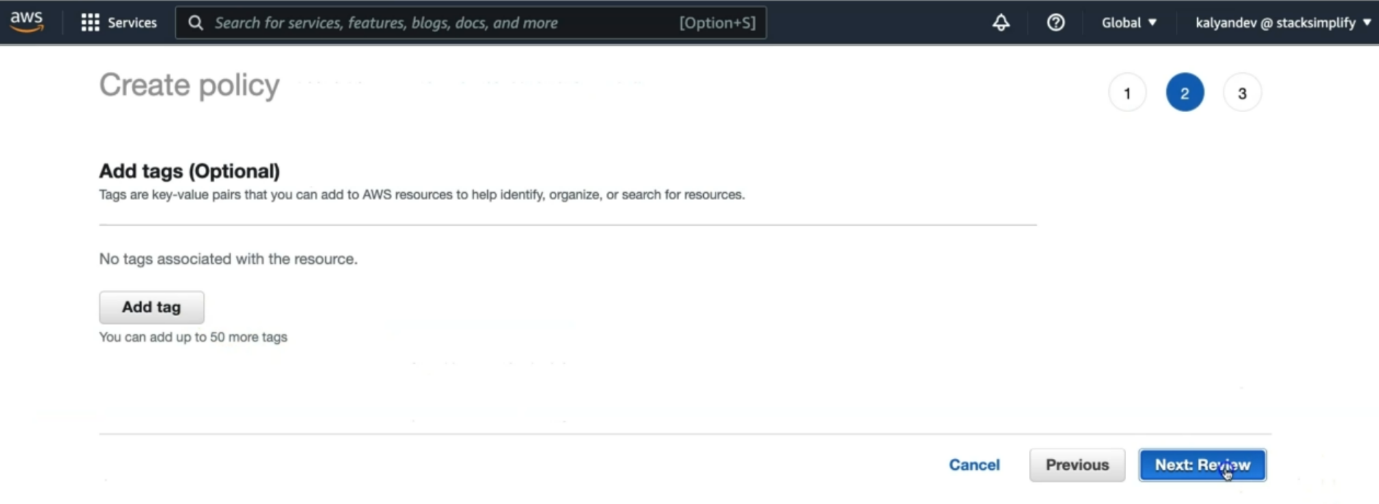


--- copy the below policy in the json field and click on visual editor.

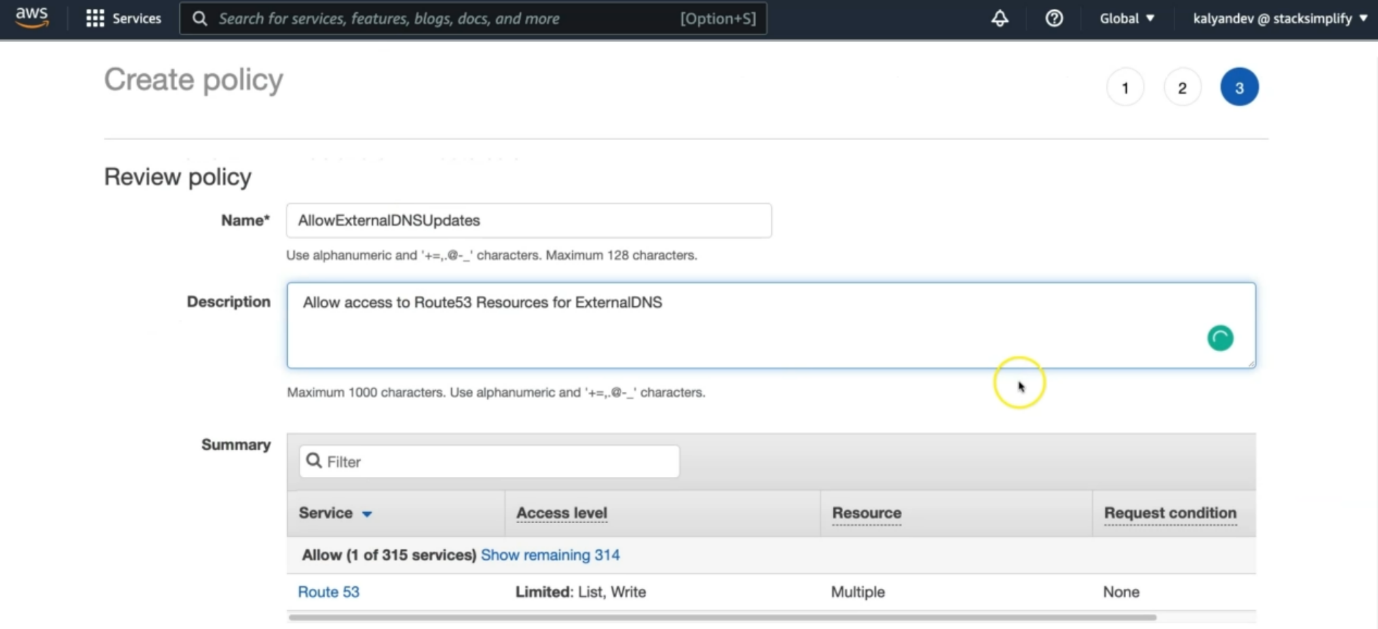
1. {
2. "Version": "2012-10-17",
3. "Statement": [
4. {
5. "Effect": "Allow",
6. "Action": [
7. "route53:ChangeResourceRecordSets"
8. ],
9. "Resource": [
10. "arn:aws:route53:::hostedzone/\*"
11. ]
12. },
13. {
14. "Effect": "Allow",
15. "Action": [
16. "route53:ListHostedZones",
17. "route53:ListResourceRecordSets"
18. ],
19. "Resource": [
20. "\*"
21. ]
22. }
23. ]
24. }



--- these are the permissions currently giving.



--- click on next review.

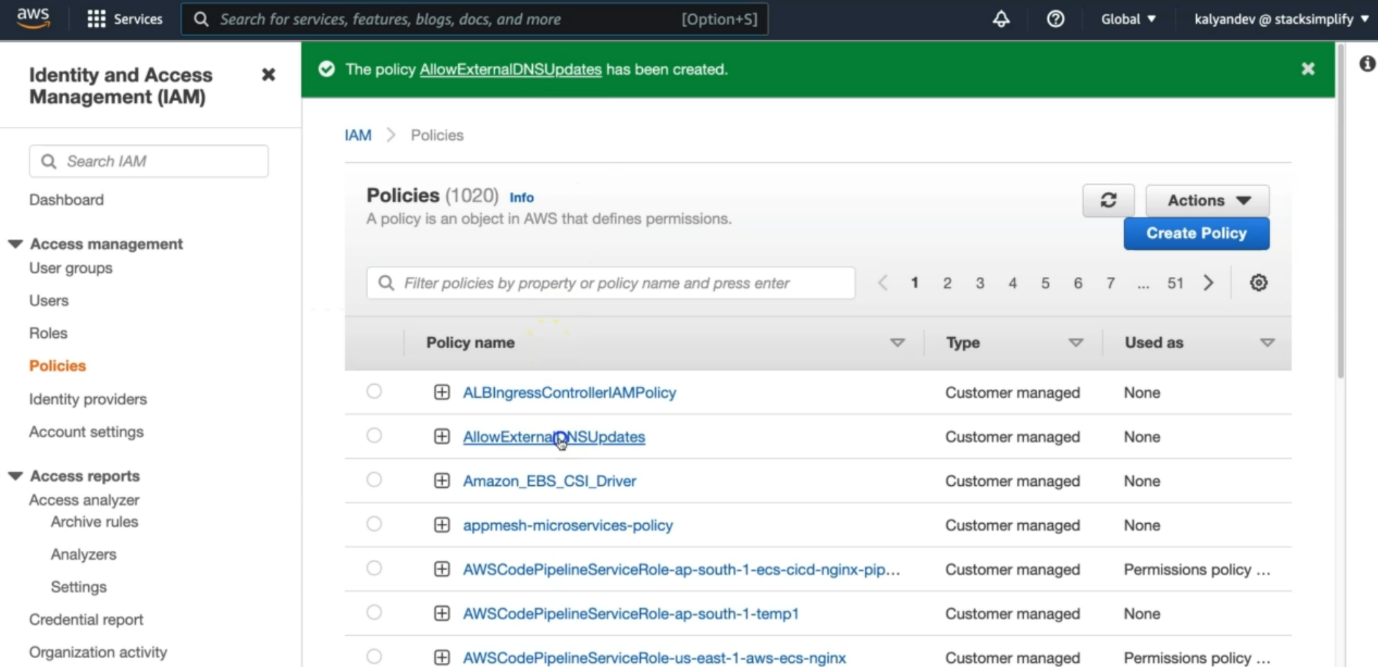


--- click on crate policy.

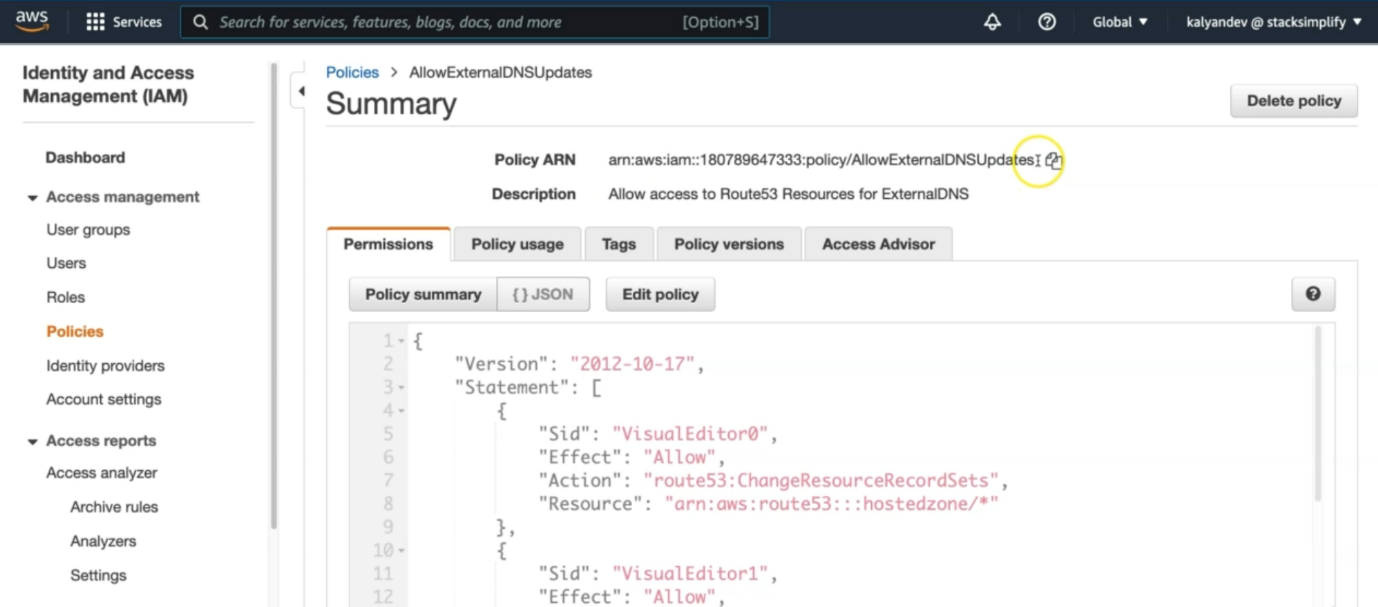
--- Make a note of Policy ARN which we will use in next step

# Policy ARN

arn:aws:iam::180789647333:policy/AllowExternalDNSUpdates



--- for seeing our arn number, we have to click on **AllowExternalDNSUpdates**. This is our policy name.



--- you can copy the arn from here.

**Create IAM Role, k8s Service Account & Associate IAM Policy**

--- As part of this step, we are going to create a k8s Service Account named external-dns and also a AWS IAM role and associate them by annotating role ARN in Service Account.

--- In addition, we are also going to associate the AWS IAM Policy AllowExternalDNSUpdates to the newly created AWS IAM Role.

**Create IAM Role, k8s Service Account & Associate IAM Policy**

**# Template**

--- **eksctl create iamserviceaccount \**

**--name service\_account\_name \**

**--namespace service\_account\_namespace \**

**--cluster cluster\_name \**

**--attach-policy-arn IAM\_policy\_ARN \**

**--approve \**

**--override-existing-serviceaccounts**

**# Replaced name, namespace, cluster, IAM Policy arn**

--- **eksctl create iamserviceaccount \**

**--name external-dns \**

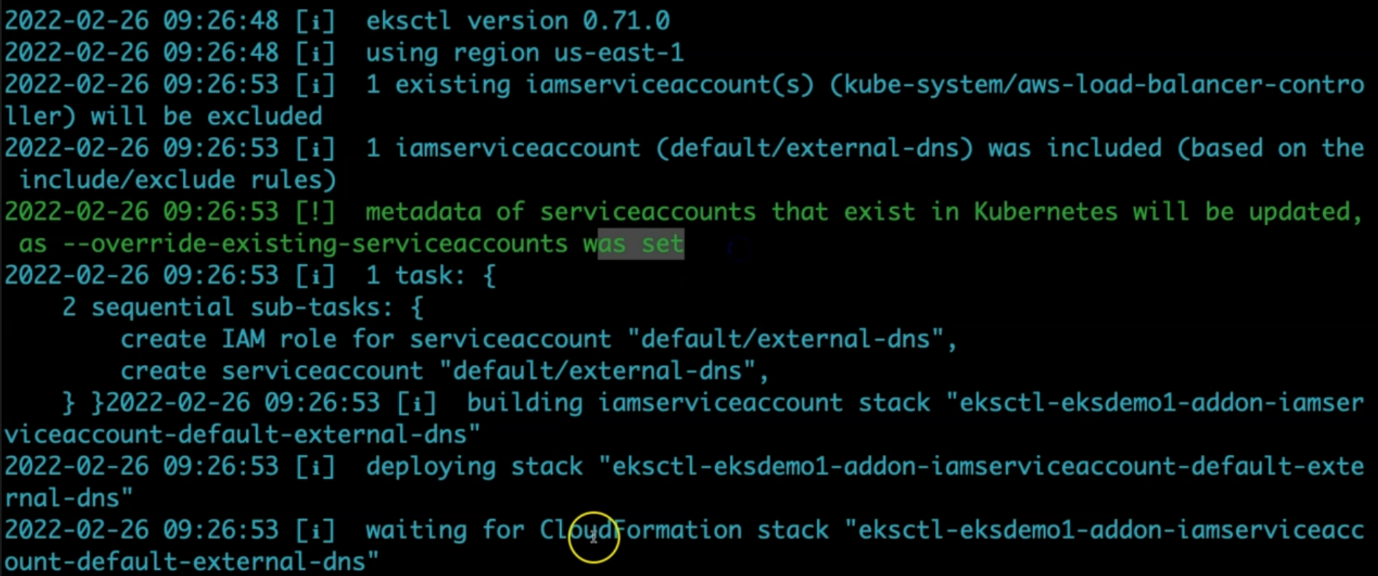
**--namespace default \**

**--cluster eksdemo1 \**

**--attach-policy-arn arn:aws:iam::180789647333:policy/AllowExternalDNSUpdates \**

**--approve \**

**--override-existing-serviceaccounts**



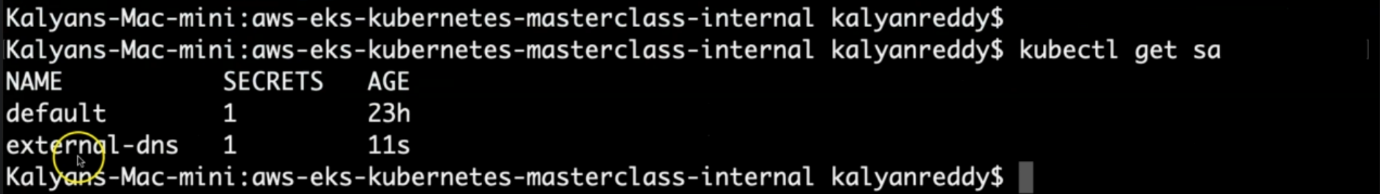
--- it will create cloudformation stack.

**Verify the Service Account**

--- Verify external-dns service account, primarily verify annotation related to IAM Role

**# List Service Account**

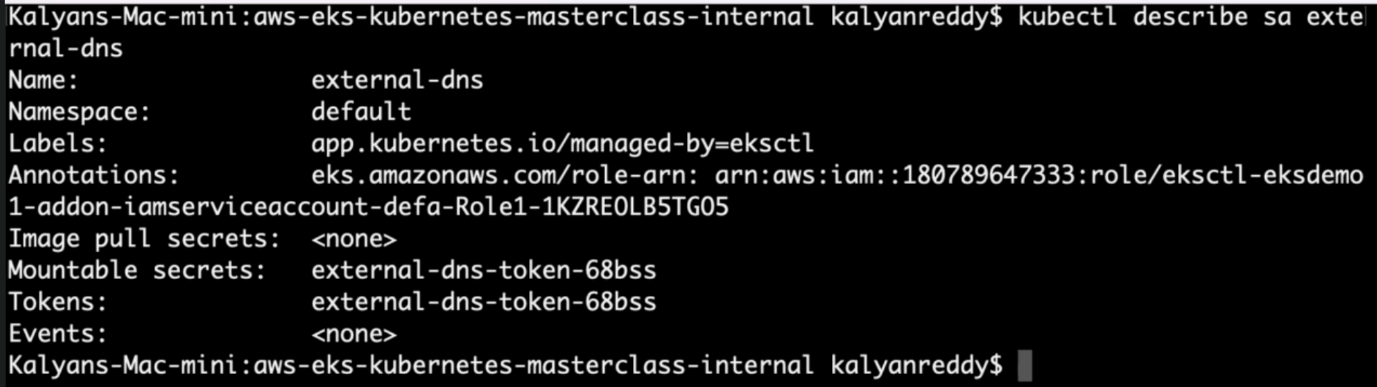
--- kubectl get sa external-dns



--- **note** – an external-dns service account is created at 11 seconds before.

**# Describe Service Account**

--- kubectl describe sa external-dns



--- **note** – inside of annotation, you will find the IAM role arn. This IAM role is created and associated with externa-dns service account.

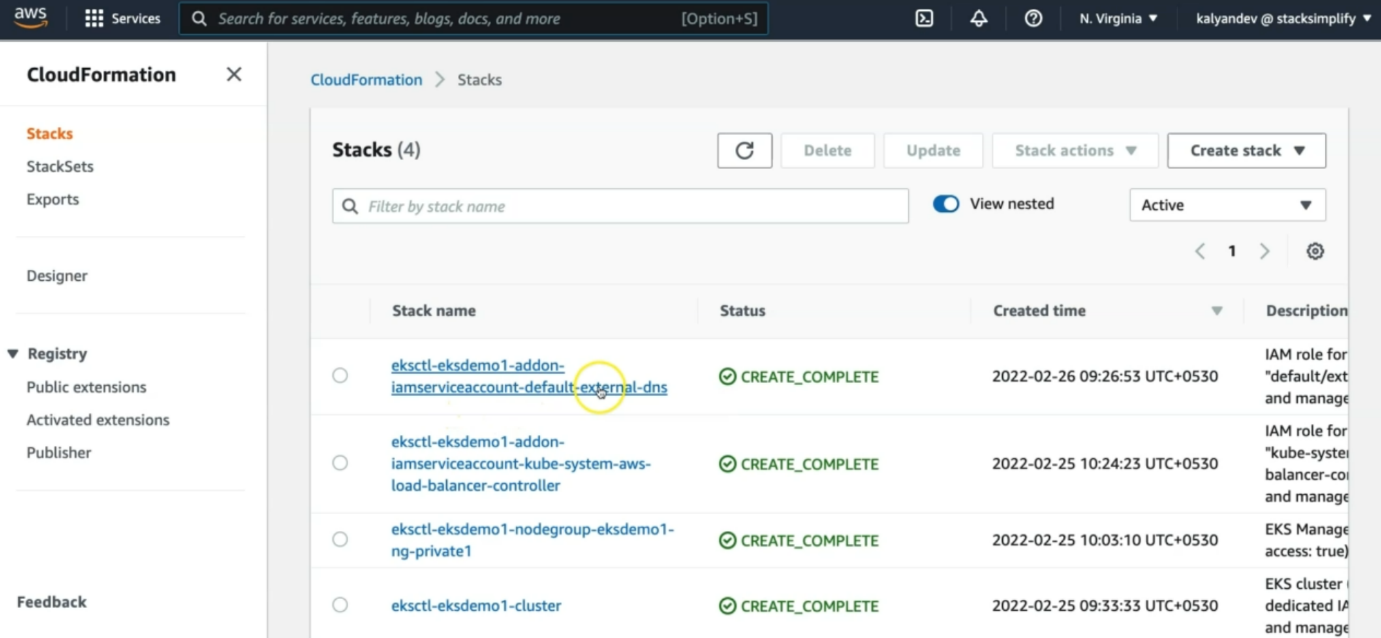
**Observation:**

1. Verify the Annotations and you should see the IAM Role is present on the Service Account

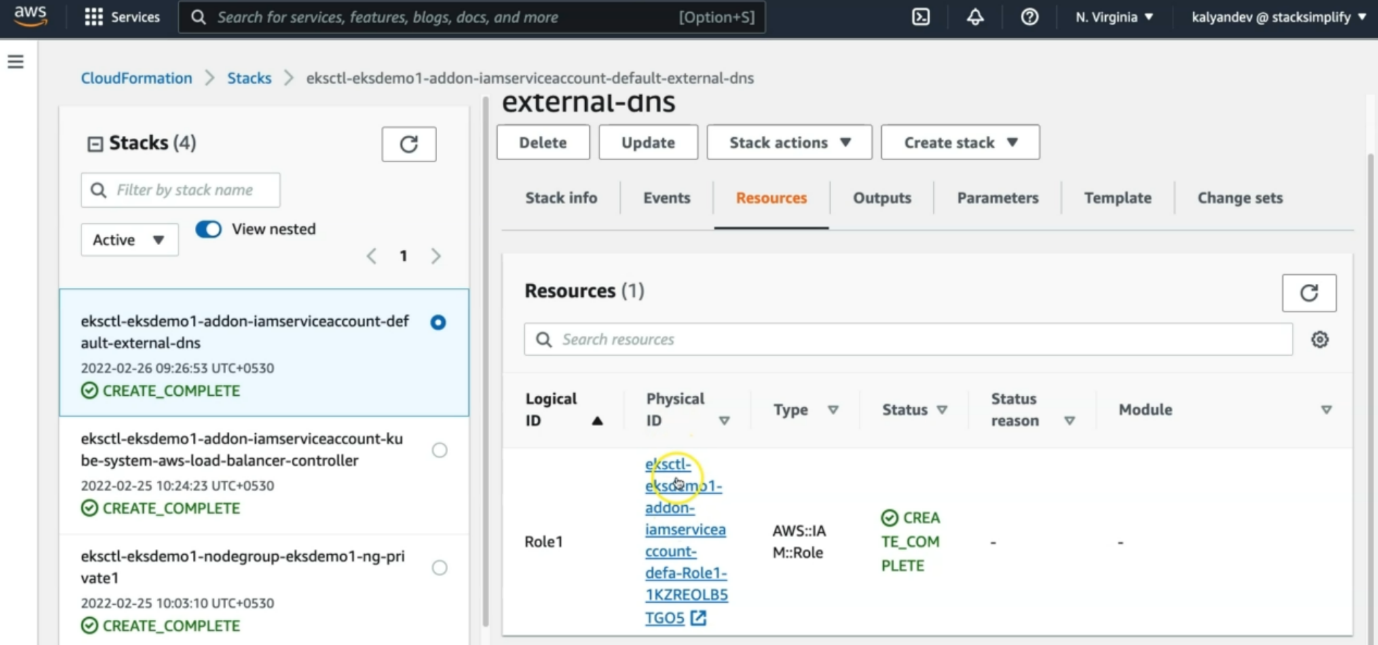
**Verify CloudFormation Stack**

--- Go to Services -> CloudFormation.

--- Verify the latest CFN Stack created.



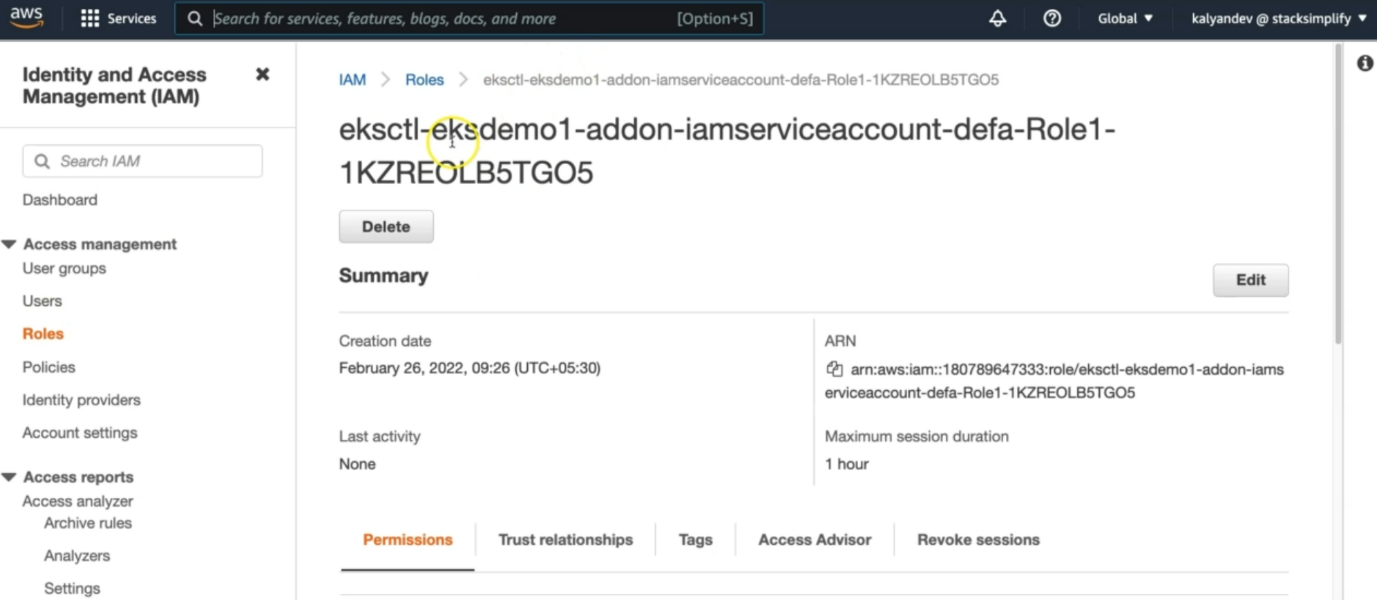
--- that is the cloudformation stack, we created.



--- Click on Resources tab

--- that is IAM policy arn.

--- Click on link in Physical ID field which will take us to IAM Role directly.



--- this is the IAM role, it had created and that is the iam Role arn.

**Verify IAM Role & IAM Policy**

--- With above step in CFN, we will be landed in IAM Role created for external-dns.

--- Verify in Permissions tab we have a policy named AllowExternalDNSUpdates

--- Now make a note of that Role ARN, this we need to update in External-DNS k8s manifest

**# Make a note of Role ARN**

# Make a note of Role ARN

arn:aws:iam::180789647333:role/eksctl-eksdemo1-addon-iamserviceaccount-defa-Role1-JTO29BVZMA2N

**Verify IAM Service Accounts using eksctl**

--- You can also make a note of External DNS Role ARN from here too. This is nothing but IAM role arn. We can this arn by 2 methods.

1. From aws account itself
2. From eksctl command line.

**# List IAM service accounts using eksctl**

--- **eksctl get iamserviceaccount --cluster eksdemo1**

# List IAM Service Accounts using eksctl

eksctl get iamserviceaccount --cluster eksdemo1

# Sample Output

Kalyans-Mac-mini:08-06-ALB-Ingress-ExternalDNS kalyanreddy$ eksctl get iamserviceaccount --cluster eksdemo1

2022-02-11 09:34:39 [ℹ] eksctl version 0.71.0

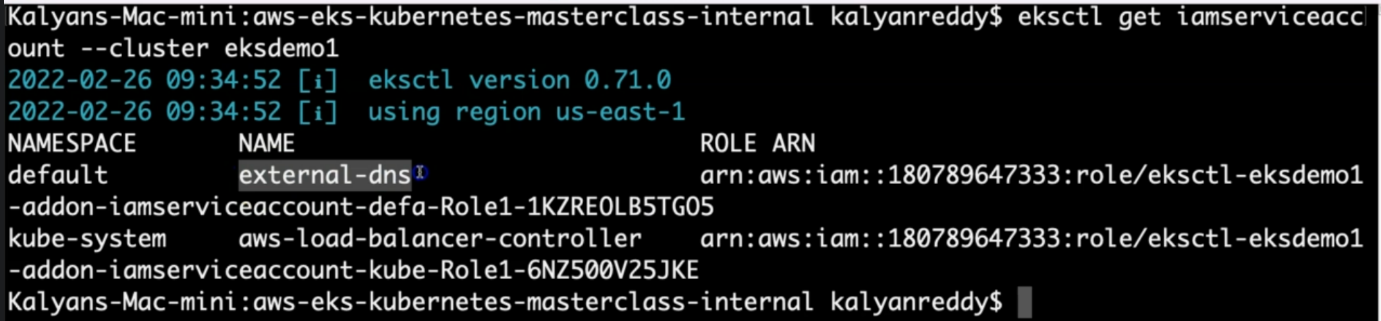
2022-02-11 09:34:39 [ℹ] using region us-east-1

NAMESPACE NAME        ROLE ARN

default   external-dnsarn:aws:iam::180789647333:role/eksctl-eksdemo1-addon-iamserviceaccount-defa-Role1-JTO29BVZMA2N

kube-system aws-load-balancer-controller arn:aws:iam::180789647333:role/eksctl-eksdemo1-addon-iamserviceaccount-kube-Role1-EFQB4C26EALH

Kalyans-Mac-mini:08-06-ALB-Ingress-ExternalDNS kalyanreddy$



--- note – it will give us 2 IAM role arn

1. One is for aws load balancer controller
2. Another one is for external dns service account.

--- **note** - why do we need this arn is, whenever we need to update the external dns manifest, we need this IAM role arn. So the external dns pod will use IAM role arn and do required work on aws.