**Create External-DNS for EKS**

This Tutorial Uses AWS Route 53

Note: This tutorial assumes that a Route 53 hosted zone and Domain has already been created.

**Create the IAM Policy**

This policy allows the External-DNS resource to be able to modify Route 53 records.

1. In the AWS console navigate to IAM.
2. Select Policies from the left hand menu.
3. Click the Create policy button.
4. Click the JSON button.
5. Paste the following into the Policy editor:

{

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

"Statement": [

{

"Effect": "Allow",

"Action": [ "route53:ChangeResourceRecordSets"],

"Resource": ["arn:aws:route53:::hostedzone/\*"]

},

{

"Effect": "Allow",

"Action": [

"route53:ListHostedZones",

"route53:ListResourceRecordSets",

"route53:ListTagsForResource"

],

"Resource": ["\*”]

}

]

}

1. Click Next
2. Name the policy “AllowExternalDNSUpdates”

**Create the EKS Cluster**

Provision the EKS Cluster using a shell(to perform these steps the most current version of the AWS CLI and eksctl tools are necessary).

1. Use the following command to provision the cluster

eksctl create cluster \

--name test-cluster \

--region us-east-2 \

--zones us-east-2a,us-east-2b,us-east-2c \

--node-type t2.medium \

--nodes 2 \

--nodes-min 1 \

--nodes-max 2 \

--version 1.27

1. This command creates an EKS cluster named test-cluster, in the us-east-2 region, in three availability zones, with two t2.medium size nodes, using Kubernetes version 1.27.
   1. Modify as necessary
2. Change kubectl context to the new cluster with the following command:  
   aws eks --region us-east-2 update-kubeconfig --name test-cluster

**Create a ServiceAccount**

This will bind the Route 53 policy to a service account to be used by the External-DNS deployment.

1. Run the following command:

eksctl utils associate-iam-oidc-provider \

--cluster test-cluster --approve

1. Run the following command:

eksctl create iamserviceaccount \

--cluster test-cluster \

--name "external-dns" \

--namespace demo \

--attach-policy-arn arn:aws:iam::205832703745:policy/AllowExternalDNSUpdates \

--approve

**Deploy the EKS Resources**

Deploy resources to create the External-DNS deployment with the correct permissions.

1. Create manifest with the following template as externaldns-with-rbac.yaml:

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

name: external-dns

labels:

app.kubernetes.io/name: external-dns

rules:

- apiGroups: [""]

resources: ["services","endpoints","pods","nodes"]

verbs: ["get","watch","list"]

- apiGroups: ["extensions","networking.k8s.io"]

resources: ["ingresses"]

verbs: ["get","watch","list"]

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: external-dns-viewer

labels:

app.kubernetes.io/name: external-dns

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: external-dns

subjects:

- kind: ServiceAccount

name: external-dns

namespace: default # change to desired namespace: externaldns, kube-addons

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: external-dns

labels:

app.kubernetes.io/name: external-dns

spec:

strategy:

type: Recreate

selector:

matchLabels:

app.kubernetes.io/name: external-dns

template:

metadata:

labels:

app.kubernetes.io/name: external-dns

spec:

serviceAccountName: external-dns

containers:

- name: external-dns

image: registry.k8s.io/external-dns/external-dns:v0.13.5

args:

- --source=service

- --source=ingress

- --domain-filter=example.com # will make ExternalDNS see only the hosted zones matching provided domain, omit to process all available hosted zones

- --provider=aws

- --policy=upsert-only # would prevent ExternalDNS from deleting any records, omit to enable full synchronization

- --aws-zone-type=public # only look at public hosted zones (valid values are public, private or no value for both)

- --registry=txt

- --txt-owner-id=external-dns

env:

- name: AWS\_DEFAULT\_REGION

value: us-east-1 # change to region where EKS is installed

1. Run the following command to deploy:

kubectl create -f external-dns-with-rbac.yaml

**Confirm External-DNS is Running**

Create a NGINX Deployment and LoadBalancer to confirm successful External-DNS deployment.

1. Create manifest with the following template as nginx.yaml:

apiVersion: v1

kind: Service

metadata:

name: nginx

annotations:

external-dns.alpha.kubernetes.io/hostname: nginx.example.com

spec:

type: LoadBalancer

ports:

- port: 80

name: http

targetPort: 80

selector:

app: nginx

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx

spec:

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- image: nginx

name: nginx

ports:

- containerPort: 80

name: http

1. Run the following command to deploy:

kubectl create -f nginx.yaml

1. Run the following command to confirm resources are running:

kubectl get all

1. This concludes this tutorial, the nginx welcome page should now be available in the browser at the Route 53 domain.

WHEN USING EASY RSA MAKE SURE YOU USE nopass with build-ca