**Application load balancing on Amazon EKS (1.22)**

Configure your Eks with AWS CLI

aws eks --region ap-south-1 update-kubeconfig --name NDDB-Cluster

**Prerequisites for EKS (1.22)**

At least two subnets in different Availability Zones. The AWS Load Balancer Controller chooses one subnet from each Availability Zone. When multiple tagged subnets are found in an Availability Zone, the controller chooses the subnet whose subnet ID comes first lexicographically. Each subnet must have at least eight available IP addresses.

If you're using multiple security groups attached to worker node, exactly one security group must be tagged as follows. Replace *cluster-name* with your cluster name.

**Key** – kubernetes.io/cluster/*cluster-name*

**Value** – shared or owned

If you're using the AWS Load Balancer Controller version 2.1.1 or earlier, subnets must be tagged in the format that follows. If you're using version 2.1.2 or later, tagging is optional. However, we recommend that you tag a subnet if any of the following is the case. You have multiple clusters that are running in the same VPC or have multiple AWS services that share subnets in a VPC. Or you want more control over where load balancers are provisioned for each cluster. Replace *cluster-name* with your cluster name.

**Key** – kubernetes.io/cluster/*cluster-name*

**Value** – shared or owned

**Create an IAM OIDC provider for your cluster.**

**To create an IAM OIDC identity provider for your cluster with the AWS Management Console**

1. Open the Amazon EKS console at <https://console.aws.amazon.com/eks/home#/clusters>.
2. Select the name of your cluster.
3. In the **Details** section on the **Overview** tab, note the value of the **OpenID Connect provider URL**.
4. Open the IAM console at <https://console.aws.amazon.com/iam/>.
5. In the left navigation pane, choose **Identity Providers** under **Access management**. If a **Provider** is listed that matches the URL for your cluster, then you already have a provider for your cluster. If a provider isn't listed that matches the URL for your cluster, then you must create one.
6. To create a provider, choose **Add Provider**.
7. For **Provider Type**, choose **OpenID Connect**.
8. For **Provider URL**, paste the OIDC issuer URL for your cluster, and then choose **Get thumbprint**.
9. For **Audience**, enter sts.amazonaws.com and choose **Add provider**

**Create an IAM policy.**

curl -o iam\_policy.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.4.2/docs/install/iam\_policy.json

**Create an IAM policy using the policy downloaded in the previous step.**

Aws iam create-policy --policy-name AWSLoadBalancerControllerIAMPolicy --policy-document <file://iam_policy.json>

2. Create an IAM role. Create a Kubernetes service account named aws-load-balancer-controller in the kube-system namespace for the AWS Load Balancer Controller and annotate the Kubernetes service account with the name of the IAM role.

You can use eksctl or the AWS CLI and kubectl to create the IAM role and Kubernetes service account.

**Using the AWS CLI and kubectl**

View your cluster's OIDC provider URL.

aws eks describe-cluster --name my-cluster --query "cluster.identity.oidc.issuer" --output text

b) Copy the following contents to a file named *load-balancer-role-trust-policy.json*.

{

"Role": {

"Path": "/",

"RoleName": "AmazonEKSLoadBalancerControllerRole",

"RoleId": "AROASRIAV2IPUOGTT2BIF",

"Arn": "arn:aws:iam::174484476447:role/AmazonEKSLoadBalancerControllerRole",

"CreateDate": "2022-06-24T08:26:45+00:00",

"AssumeRolePolicyDocument": {

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

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Federated": "arn:aws:iam::174484476447:oidc-provider/oidc.eks.ap-south-1.amazonaws.com/id/934CE328164716369B593A4B960B5A33"

},

"Action": "sts:AssumeRoleWithWebIdentity",

"Condition": {

"StringEquals": {

"oidc.eks.ap-south-1.amazonaws.com/id/934CE328164716369B593A4B960B5A33:aud": "sts.amazonaws.com",

"oidc.eks.ap-south-1.amazonaws.com/id/934CE328164716369B593A4B960B5A33:sub": "system:serviceaccount:kube-system:aws-load-balancer-controller"

}

}

}

]

}

}

}

**Create the IAM role**.

aws iam create-role --role-name AmazonEKSLoadBalancerControllerRole --assume-role-policy-document file://"load-balancer-role-trust-policy.json"

d) Attach the required Amazon EKS managed IAM policy to the IAM role.

aws iam attach-role-policy --policy-arn arn:aws:iam:: 174484476447:policy/AWSLoadBalancerControllerIAMPolicy --role-name AmazonEKSLoadBalancerControllerRole

e) Save the following contents to a file that's named *aws-load-balancer-controller-service-account.yaml*.

apiVersion: v1

kind: ServiceAccount

metadata:

labels:

app.kubernetes.io/component: controller

app.kubernetes.io/name: aws-load-balancer-controller

name: aws-load-balancer-controller

namespace: kube-system

annotations:

eks.amazonaws.com/role-arn: arn:aws:iam:: 174484476447:role/AmazonEKSLoadBalancerControllerRole

f) Create the Kubernetes service account on your cluster. The Kubernetes service account named aws-load-balancer-controller is annotated with the IAM role that you created named *AmazonEKSLoadBalancerControllerRole*.

kubectl apply -f aws-load-balancer-controller-service-account.yaml

**Verify that the controller is installed**.

kubectl get deployment -n kube-system aws-load-balancer-controller

Apply file to Eks cluster (angular application)

Kubectl apply -f client.depl.yam:- a single file consist deployment and services both.

Apply ingress .yaml file on Eks

After Applying ingress on Eks to verify ALB

kubectl get service ingress-nginx-controller --namespace=ingress-nginx

we can also see in aws console on loader balance left side navigation bar

This is a link of angular application yaml file.

https://nddbcoop.sharepoint.com/:u:/s/TCSLivestackTeam/EfidqZYdpsVLvtwvx8UlxxsBhypv-6WoJ0FTSGnu7oFa1g?e=bdyeZO



------------------------

Problem Statement For Creating ALB On Kubernet 1.22

------------------------

You were facing an issue where after creating an Ingress, a DNS record was not getting added to Route53. You wanted to know why and how to address it.

\"level\":\"error\",\"ts\":1656491480.4026341,\"logger\":\"controller-runtime.manager.controller.ingress\",\"msg\":\"Reconciler error\",\"name\":\"ingress-service\",\"namespace\":\"dev\",\"error\":\"WebIdentityErr: failed to retrieve credentials\\ncaused by: InvalidIdentityToken: Incorrect token audience\\n\\tstatus code: 400, request id: 642ba4b0-9cb9-4f3c-97dd-b205aeb70701\"

--------------

Observations

--------------

During the chat, We observed that the aws-load-balancer-controller service account was not annotated with a role. We annotated the service account with the following command and redeployed the AWS loadbalancer controller pod.

kubectl annotate serviceaccount -n kube-system aws-loadbalancer-controller eks.amazonaws.com/role-arn=arn:aws:iam::174484476447:role/AmazonEKSLoadBalancerControllerRole

We then created a deployment, service, and ingress and we observed the following error again.

{\"level\":\"error\",\"ts\":1656499113.4256,\"logger\":\"controller-runtime.manager.controller.ingress\",\"msg\":\"Reconciler error\",\"name\":\"nddb-ingress-srv\",\"namespace\":\"nddb-dev\",\"error\":\"WebIdentityErr: failed to retrieve credentials\\ncaused by: InvalidIdentityToken: Incorrect token audience\\n\\tstatus code: 400, request id: f731c0be-7199-4775-9933-78a898504e96\"}

We confirmed the Identity provider was not set to correct audience. We updated it on amazon console and after that the ingress was created correctly but we were not able to access the application and started getting 502 bad gateway.

He is not assign a public ip to subnets and these ip are assigned with subnets.

subnet-09c991cd90fe71e1d

subnet-0a885f9ed9857f473

subnet-0d416bed4957dcc9f