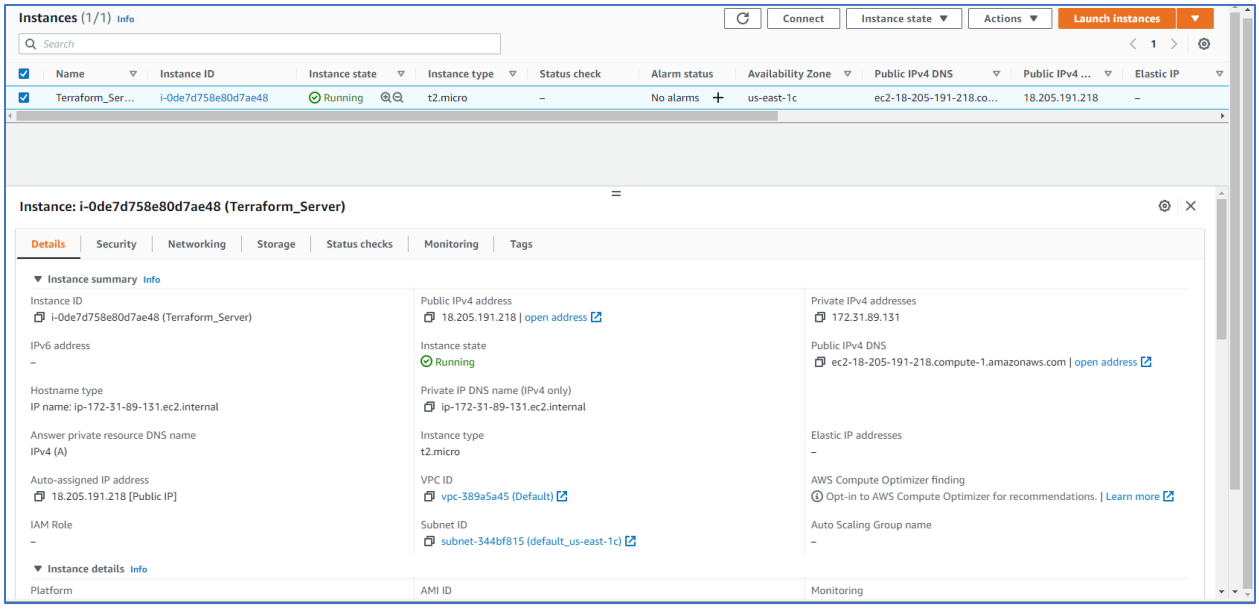
Assignment 10 :: Use terraform to launch 3 ec2 instance and EKS cluster

Step 1 :: Create a EC2 instance and install terraform in it



hostnamectl set-hostname terraform_server
yum update -y
yum install -y wget unzip
wget https://releases.hashicorp.com/terraform/0.12.2/terraform_0.12.2_linux_amd64.zip

```
[root@terraform_server ~]# wget https://releases.hashicorp.com/terraform/0.12.2/terraform_0.12.2_linux_amd64.zip
--2022-07-21 15:05:12--  https://releases.hashicorp.com/terraform/0.12.2/terraform_0.12.2_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 146.75.38.49
Connecting to releases.hashicorp.com (releases.hashicorp.com)|146.75.38.49|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16018372 (15M) [application/zip]
Saving to: 'terraform_0.12.2_linux_amd64.zip'

100%[==========================================================================>] 16,018,372  --.-K/s   in 0.1s

2022-07-21 15:05:12 (112 MB/s) - 'terraform_0.12.2_linux_amd64.zip' saved [16018372/16018372]

[root@terraform_server ~]#
```

unzip ./terraform_0.12.2_linux_amd64.zip -d /usr/local/bin

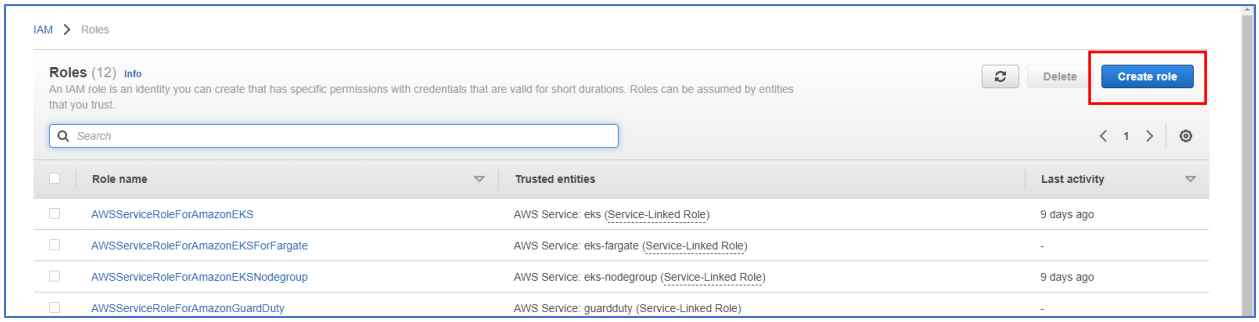
```
[root@terraform_server ~]# unzip ./terraform_0.12.2_linux_amd64.zip -d /usr/local/bin
Archive:  ./terraform_0.12.2_linux_amd64.zip
  inflating: /usr/local/bin/terraform
[root@terraform_server ~]#
```

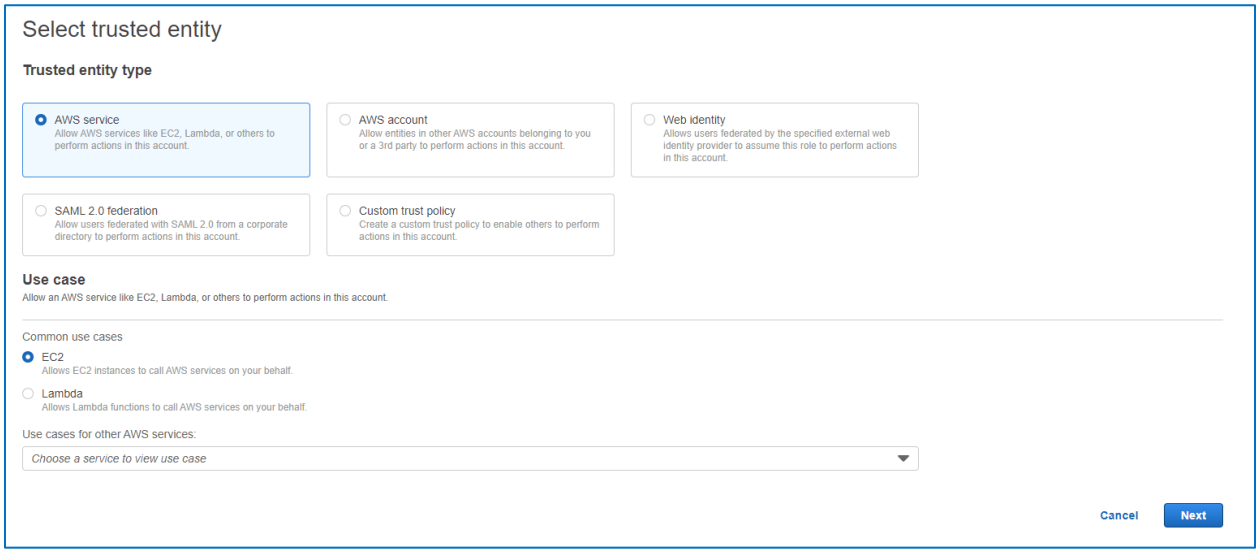
terraform -v

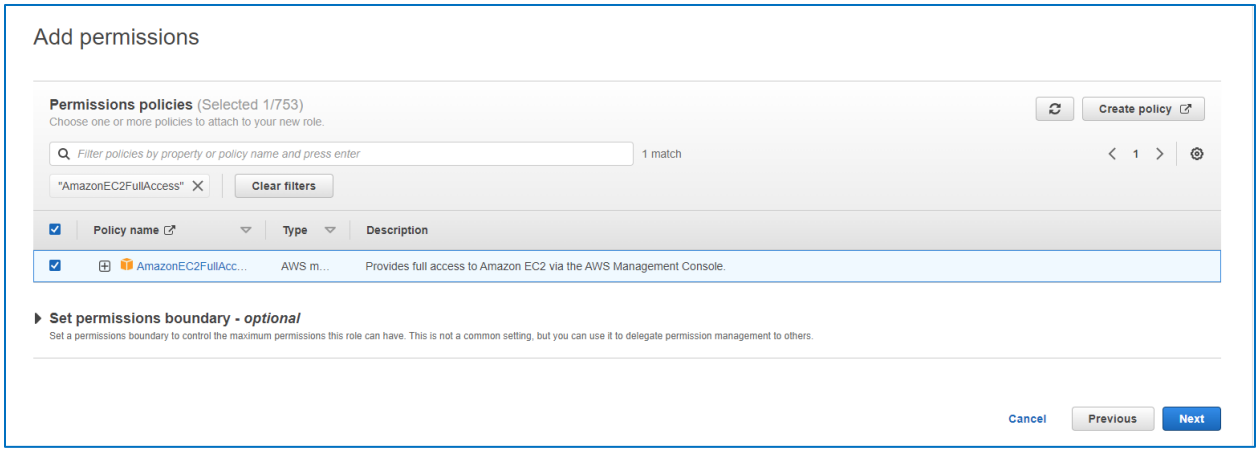
```
[root@terraform_server ~]# terraform -v
Terraform v0.12.2

Your version of Terraform is out of date! The latest version
is 1.2.5. You can update by downloading from www.terraform.io/downloads.html
[root@terraform_server ~]#
```

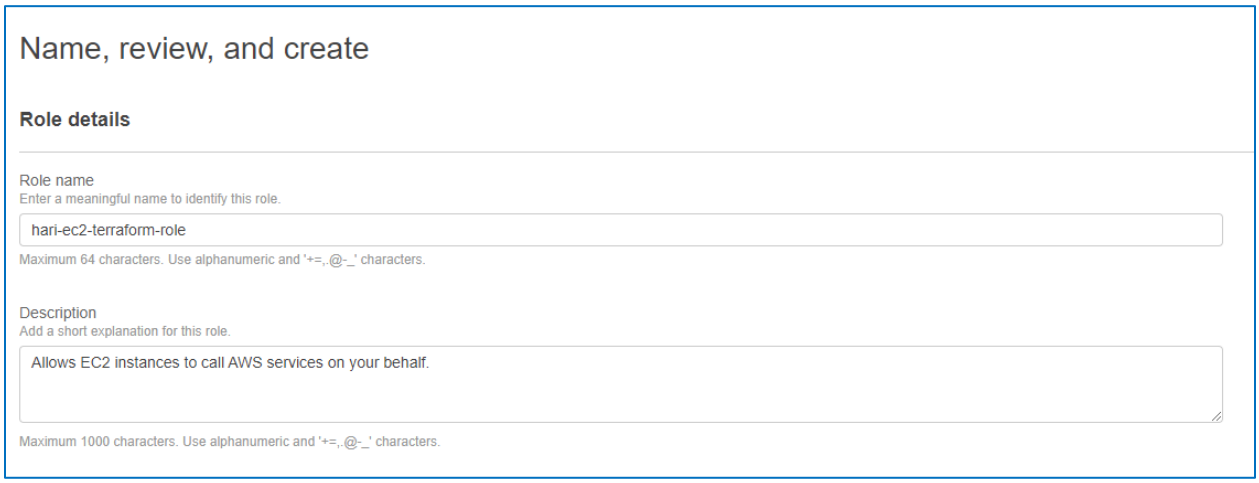
Step 2 :: Create an IAM role with AmazonEC2FullAccess Policy

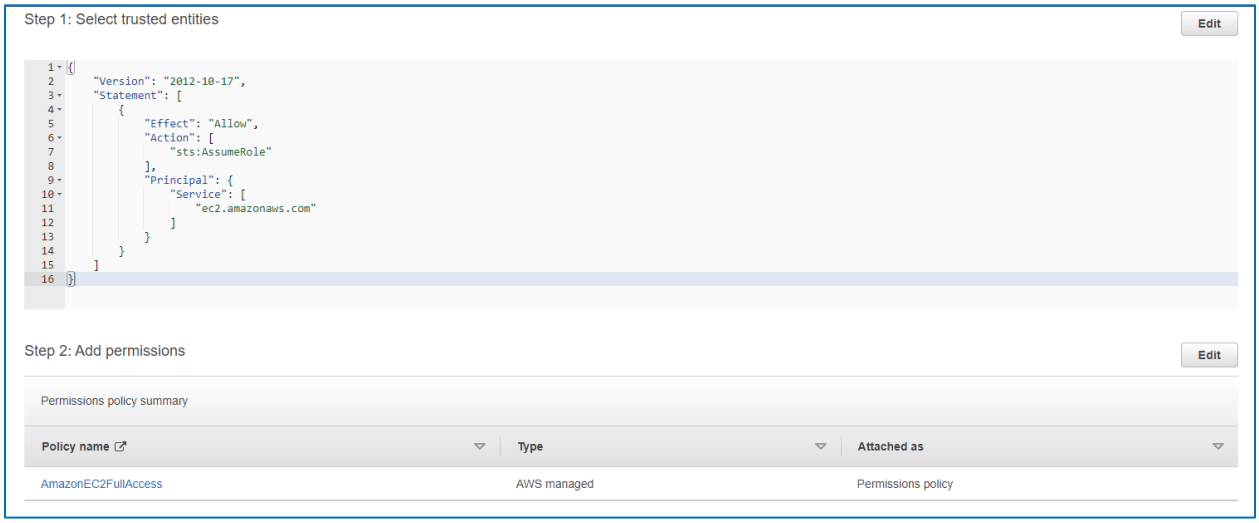
Step 2.1 : Goto IAM dashboard and select “Create role” under Roles page

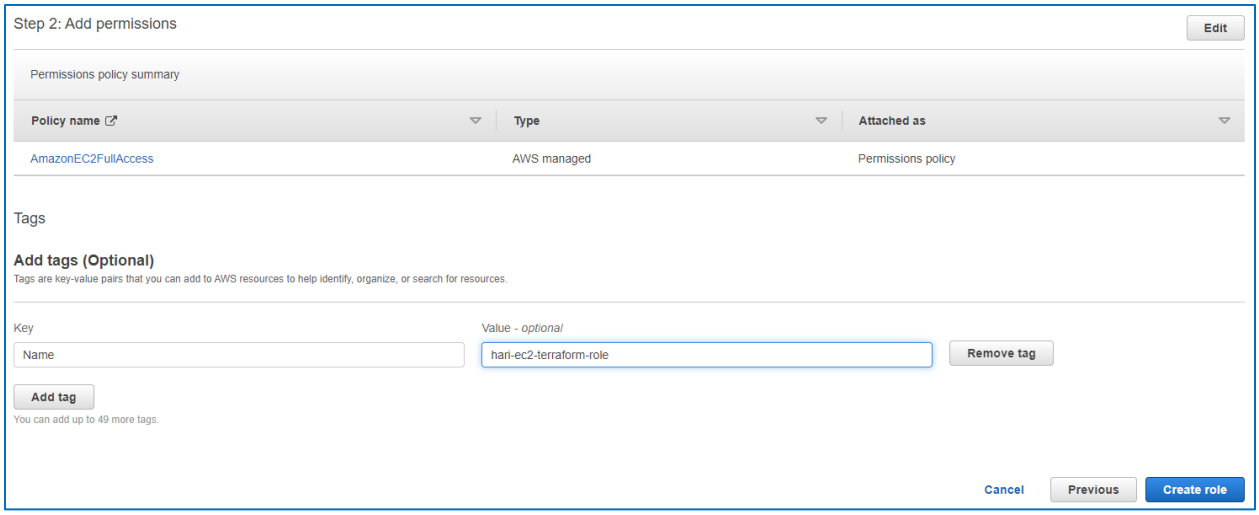




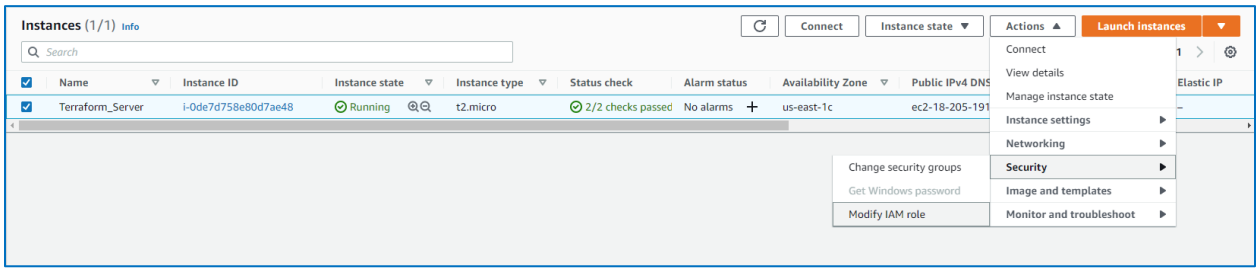

Add permissions
Permissions policies (Selected 1/753)
Choose one or more policies to attach to your new role.
Create policy
Filter policies by property or policy name and press enter
1 match
1
"AmazonEC2FullAccess"
Clear filters
Policy name
Type
Description
AmazonEC2FullAcc...
AWS m...
Provides full access to Amazon EC2 via the AWS Management Console.
Set permissions boundary - optional
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.
Cancel
Previous
Next

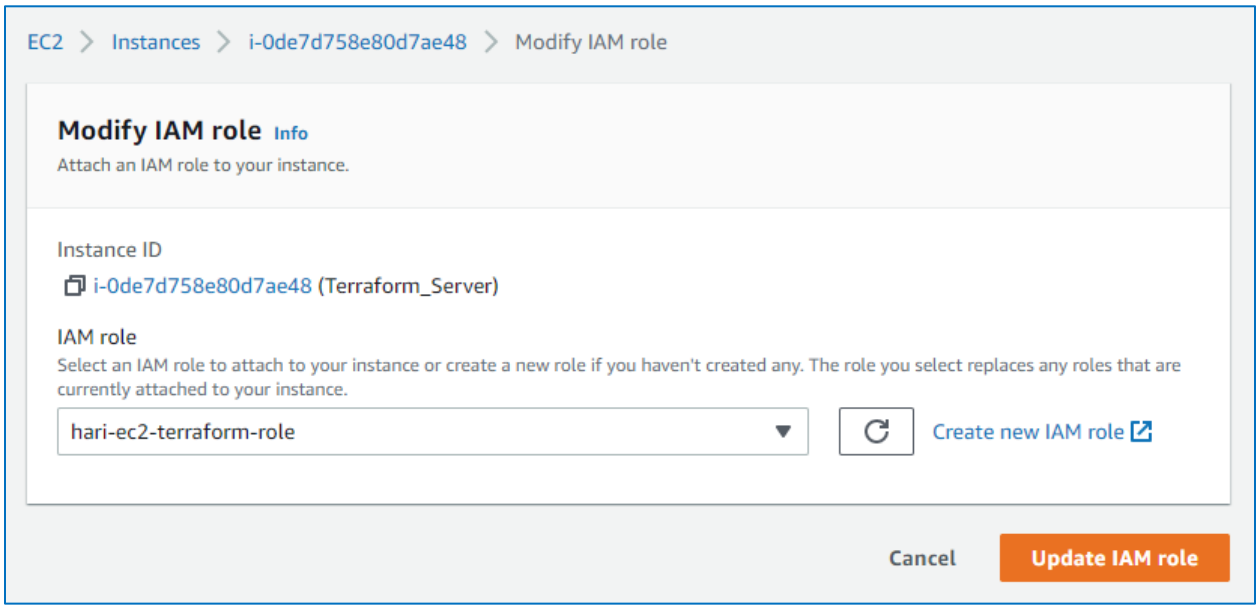

Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
hari-ec2-terraform-role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.


Step 1: Select trusted entities
Edit
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"sts:AssumeRole"
],
"Principal": {
"Service": [
"ec2.amazonaws.com"
]
}
}
]
}
Step 2: Add permissions
Edit
Permissions policy summary
Policy name
Type
Attached as
AmazonEC2FullAccess
AWS managed
Permissions policy



Step 2.2 : Attach the created role to the created terraform instance





Step 3 :: Create terraform files

cd ~
mkdir terraform-dir
cd terraform-dir

```
[root@terraform_server ~]# cd ~
[root@terraform_server ~]# mkdir terraform-dir
[root@terraform_server ~]# cd terraform-dir
[root@terraform_server terraform-dir]# pwd
/root/terraform-dir
[root@terraform_server terraform-dir]#
```

cat variables.tf

```
[root@terraform_server terraform-dir]# cat variables.tf
variable "aws_region" {
       description = "The AWS region to create things in."
       default     = "us-east-1"
}

variable "key_name" {
    description = " SSH keys to connect to ec2 instance"
    default     =  "Terraform_Keypair"
}

variable "instance_type" {
    description = "instance type for ec2"
    default     =  "t2.micro"
}

variable "security_group" {
    description = "Name of security group"
    default     = "terraform-sg"
}

variable "tag_name" {
    description = "Tag Name of for EC2 instance"
    default     = "hari-ec2-instance"
}
variable "ami_id" {
    description = "AMI for Ubuntu Ec2 instance"
    default     = "ami-0cff7528ff583bf9a"
}
[root@terraform_server terraform-dir]#
```

cat main.tf

```
[root@terraform_server terraform-dir]# cat main.tf
provider "aws" {
  region = var.aws_region
}

#Create security group with firewall rules
resource "aws_security_group" "terraform-sg" {
  name        = var.security_group
  description = "security group for jenkins"

  ingress {
    from_port   = 8080
    to_port     = 8080
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

 ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

 # outbound from jenkis server
  egress {
    from_port   = 0
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags= {
    Name = var.security_group
  }
}

resource "aws_instance" "myFirstTerraformInstance" {
  ami           = var.ami_id
  key_name = var.key_name
  instance_type = var.instance_type
  security_groups= [var.security_group]
  tags= {
    Name = var.tag_name
  }
}

# Create Elastic IP address
resource "aws_eip" "myTerraformElasticIP" {
  vpc      = true
  instance = aws_instance.myFirstTerraformInstance.id
tags= {
    Name = "terraform_elastic_ip"
  }
}
[root@terraform_server terraform-dir]#
```

Step 4 :: Execute Terraform Commands

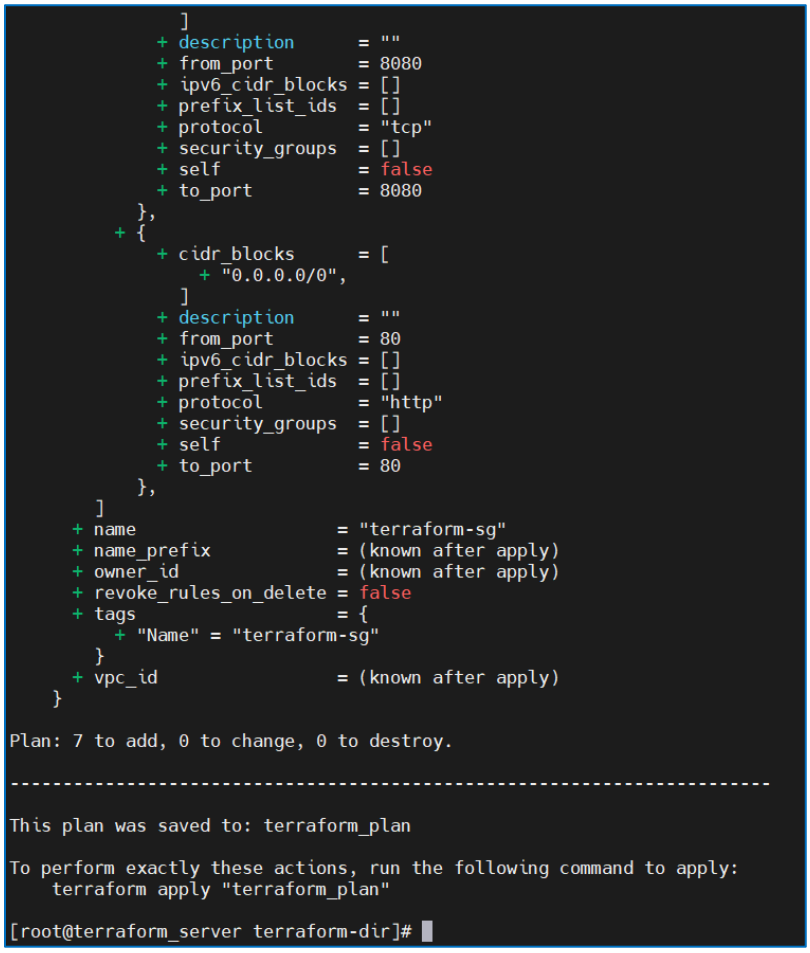
terraform init

```
[root@terraform_server terraform-dir]# terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 3.37.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 3.37"


Warning: registry.terraform.io: This version of Terraform has an outdated GPG key and is unable to verify new provider releases. Please upgrade Terraform to at least
 0.12.31 to receive new provider updates. For details see: https://discuss.hashicorp.com/t/hcsec-2021-12-codecov-security-event-and-hashicorp-gpg-key-exposure/23512


Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[root@terraform_server terraform-dir]#
```

terraform plan

```
[root@terraform_server terraform-dir]# terraform plan -out terraform_plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


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

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_eip.myFirstTerraformElasticIP will be created
  + resource "aws_eip" "myFirstTerraformElasticIP" {
      + allocation_id        = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = (known after apply)
      + id                   = (known after apply)
      + instance             = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags                 = {
          + "Name" = "terraform_elastic_ip_1"
        }
      + vpc                  = true
    }

  # aws_eip.mySecondTerraformElasticIP will be created
  + resource "aws_eip" "mySecondTerraformElasticIP" {
      + allocation_id        = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
```

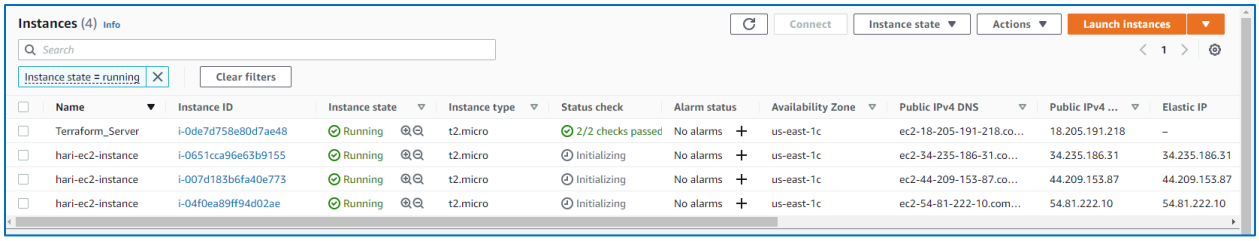
.
.
.


]
+ description = ""
+ from_port = 8080
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 8080
},
+ {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
+ description = ""
+ from_port = 80
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "http"
+ security_groups = []
+ self = false
+ to_port = 80
},
]
+ name = "terraform-sg"
+ name_prefix = (known after apply)
+ owner_id = (known after apply)
+ revoke_rules_on_delete = false
+ tags = {
+ "Name" = "terraform-sg"
}
+ vpc_id = (known after apply)
}
Plan: 7 to add, 0 to change, 0 to destroy.
This plan was saved to: terraform_plan
To perform exactly these actions, run the following command to apply:
terraform apply "terraform_plan"
[root@terraform_server terraform-dir]#

terraform apply

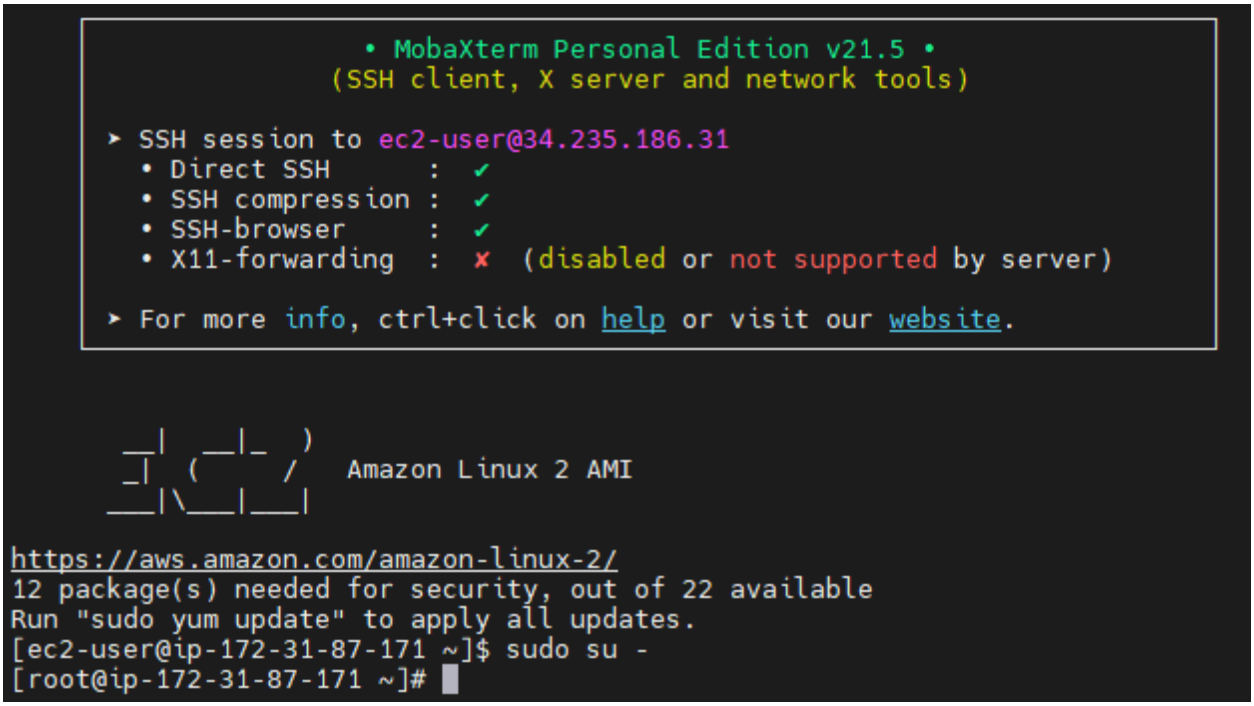
```
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

Plan: 3 to add, 3 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.mySecondTerraformInstance: Creating...
aws_instance.myThirdTerraformInstance: Creating...
aws_instance.myFirstTerraformInstance: Creating...
aws_instance.mySecondTerraformInstance: Still creating... [10s elapsed]
aws_instance.myThirdTerraformInstance: Still creating... [10s elapsed]
aws_instance.myFirstTerraformInstance: Still creating... [10s elapsed]
aws_instance.mySecondTerraformInstance: Still creating... [20s elapsed]
aws_instance.myThirdTerraformInstance: Still creating... [20s elapsed]
aws_instance.myFirstTerraformInstance: Still creating... [20s elapsed]
aws_instance.mySecondTerraformInstance: Still creating... [30s elapsed]
aws_instance.myThirdTerraformInstance: Still creating... [30s elapsed]
aws_instance.myFirstTerraformInstance: Still creating... [30s elapsed]
aws_instance.mySecondTerraformInstance: Creation complete after 31s [id=i-007d183b6fa40e773]
aws_eip.mySecondTerraformElasticIP: Modifying... [id=eipalloc-06bb0152aa18c630e]
aws_instance.myThirdTerraformInstance: Creation complete after 32s [id=i-04f0ea89ff94d02ae]
aws_eip.myThirdTerraformElasticIP: Modifying... [id=eipalloc-07c823fb6319788fe]
aws_instance.myFirstTerraformInstance: Creation complete after 32s [id=i-0651cca96e63b9155]
aws_eip.myFirstTerraformElasticIP: Modifying... [id=eipalloc-0e8d3dc61f4fc6e9f]
aws_eip.mySecondTerraformElasticIP: Modifications complete after 2s [id=eipalloc-06bb0152aa18c630e]
aws_eip.myThirdTerraformElasticIP: Modifications complete after 1s [id=eipalloc-07c823fb6319788fe]
aws_eip.myFirstTerraformElasticIP: Modifications complete after 1s [id=eipalloc-0e8d3dc61f4fc6e9f]

Apply complete! Resources: 3 added, 3 changed, 0 destroyed.
[root@terraform_server terraform-dir]#
```

Result :: Successfully created three EC2 instances



```
[root@terraform_server terraform-dir]# terraform state list
aws_eip.myTerraformElasticIP[0]
aws_eip.myTerraformElasticIP[1]
aws_eip.myTerraformElasticIP[2]
aws_instance.myTerraformInstance[0]
aws_instance.myTerraformInstance[1]
aws_instance.myTerraformInstance[2]
aws_security_group.terraform-sg
[root@terraform_server terraform-dir]#
```

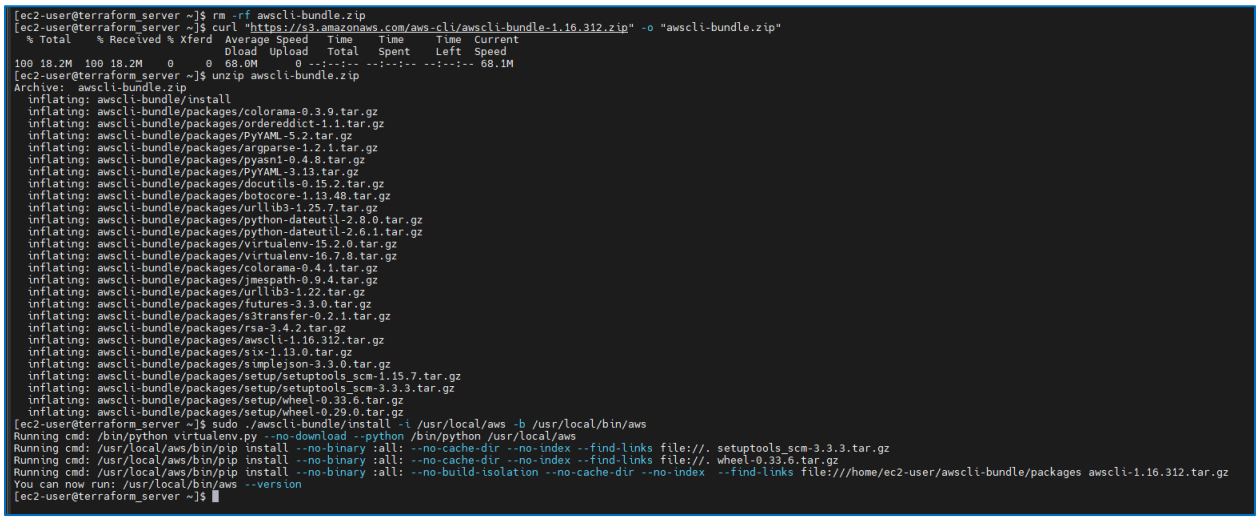
Trying to connect to one of the instances



EKS Cluster using terraform

Step 1 :: Install AWS CLI and aws-iam-authenticator

Step 1.1 : Install aws cli
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
$ unzip awscli-bundle.zip
$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws



Step 1.2 : Install aws-iam-authenticator

$ curl -o aws-iam-authenticator https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.2/2021-07-05/bin/linux/amd64/aws-iam-authenticator
$ curl -o aws-iam-authenticator.sha256 https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.2/2021-07-05/bin/linux/amd64/aws-iam-authenticator.sha256
$ openssl sha1 -sha256 aws-iam-authenticator
$ chmod +x ./aws-iam-authenticator
$ mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$PATH:$HOME/bin
$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
$ aws-iam-authenticator help

```
[ec2-user@terraform_server ~]$ curl -o aws-iam-authenticator https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.2/2021-07-05/bin/linux/amd64/aws-iam-authenticator
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 33.6M  100 33.6M    0     0  7508k      0  0:00:04  0:00:04 --:--:-- 7510k
[ec2-user@terraform_server ~]$ curl -o aws-iam-authenticator.sha256 https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.2/2021-07-05/bin/linux/amd64/aws-iam-authenticator.sha256
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    87  100    87    0     0    310      0 --:--:-- --:--:-- --:--:--   310
[ec2-user@terraform_server ~]$ openssl sha1 -sha256 aws-iam-authenticator
SHA256(aws-iam-authenticator)= fe958eff955bea1499015b45dc53392a33f737630efd841cd574559cc0f41800
[ec2-user@terraform_server ~]$ chmod +x ./aws-iam-authenticator
[ec2-user@terraform_server ~]$ mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$PATH:$HOME/bin
[ec2-user@terraform_server ~]$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
[ec2-user@terraform_server ~]$ aws-iam-authenticator help
A tool to authenticate to Kubernetes using AWS IAM credentials

Usage:
  aws-iam-authenticator [command]

Available Commands:
  help        Help about any command
  init        Pre-generate certificate, private key, and kubeconfig files for the server.
  server      Run a webhook validation server suitable that validates tokens using AWS IAM
  token       Authenticate using AWS IAM and get token for Kubernetes
  verify      Verify a token for debugging purpose
  version     Version will output the current build information

Flags:
  -i, --cluster-id ID                Specify the cluster ID, a unique-per-cluster identifier for your aws-iam-authenticator installation.
  -c, --config filename              Load configuration from filename
      --feature-gates mapStringBool  A set of key=value pairs that describe feature gates for alpha/experimental features. Options are:
                                     AllAlpha=true|false (ALPHA - default=false)
                                     IAMIdentityMappingCRD=true|false (ALPHA - default=false)
  -h, --help                         help for aws-iam-authenticator
  -l, --log-format string            Specify log format to use when logging to stderr [text or json] (default "text")

Use "aws-iam-authenticator [command] --help" for more information about a command.
[ec2-user@terraform_server ~]$
```

Step 2 :: Configure AWS Command Line using Security Credentials

- Go to AWS Management Console --> Services --> IAM --> User --> Select user : abushad
- Click on Security credentials tab
- Click on Create access key
- Copy Access ID and Secret access key
- Go to command line and use
    - $ aws configure
    - $ aws eks list-clusters

```
[ec2-user@terraform_server ~]$ aws configure
AWS Access Key ID [None]: AKIA4YTN4MDXCICCBEW5
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: table
[ec2-user@terraform_server ~]$ aws eks list-clusters
--------------
|ListClusters|
+------------+
[ec2-user@terraform_server ~]$
```

Step 3 :: Install and configure kubectl CLI

Step 3.1 :: Install kubectl CLI

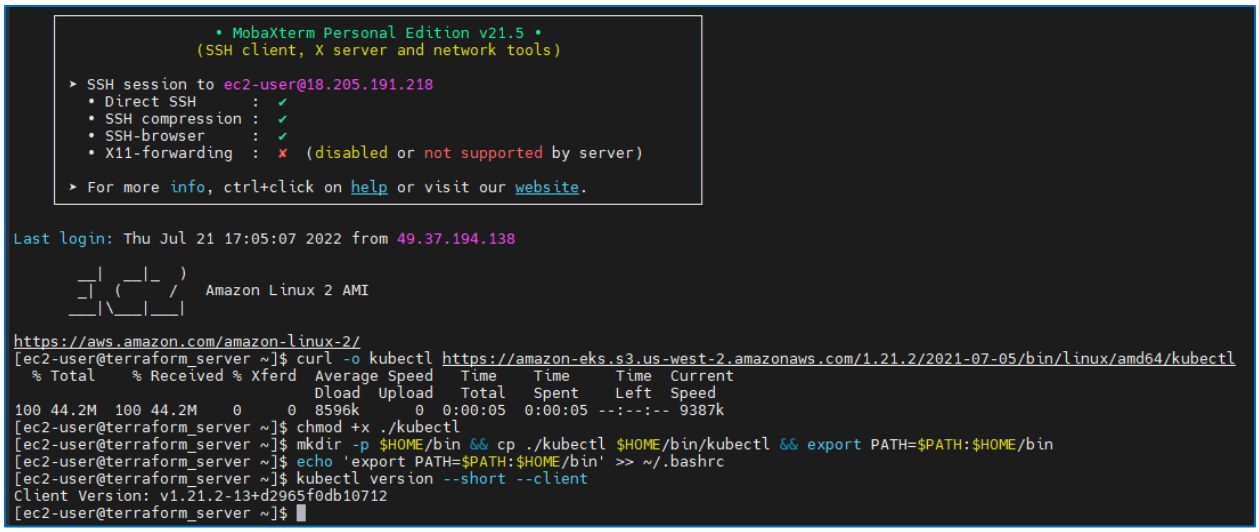
$ curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.21.2/2021-07-05/bin/linux/amd64/kubectl

$ chmod +x ./kubectl

$ mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin

$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc

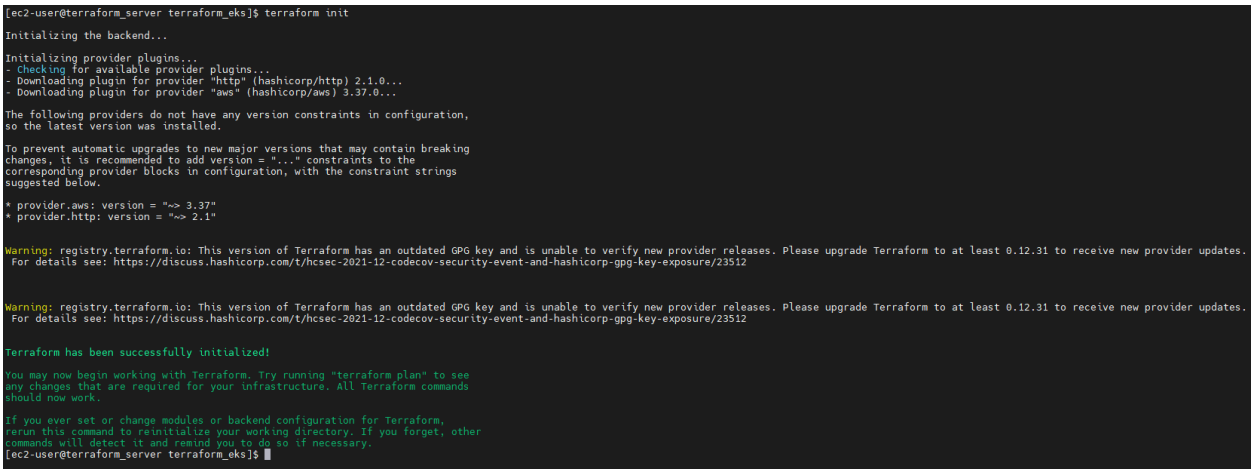
$ kubectl version --short –client



Step 4 :: Creating EKS Cluster

$ mkdir terraform_eks

$ terraform init

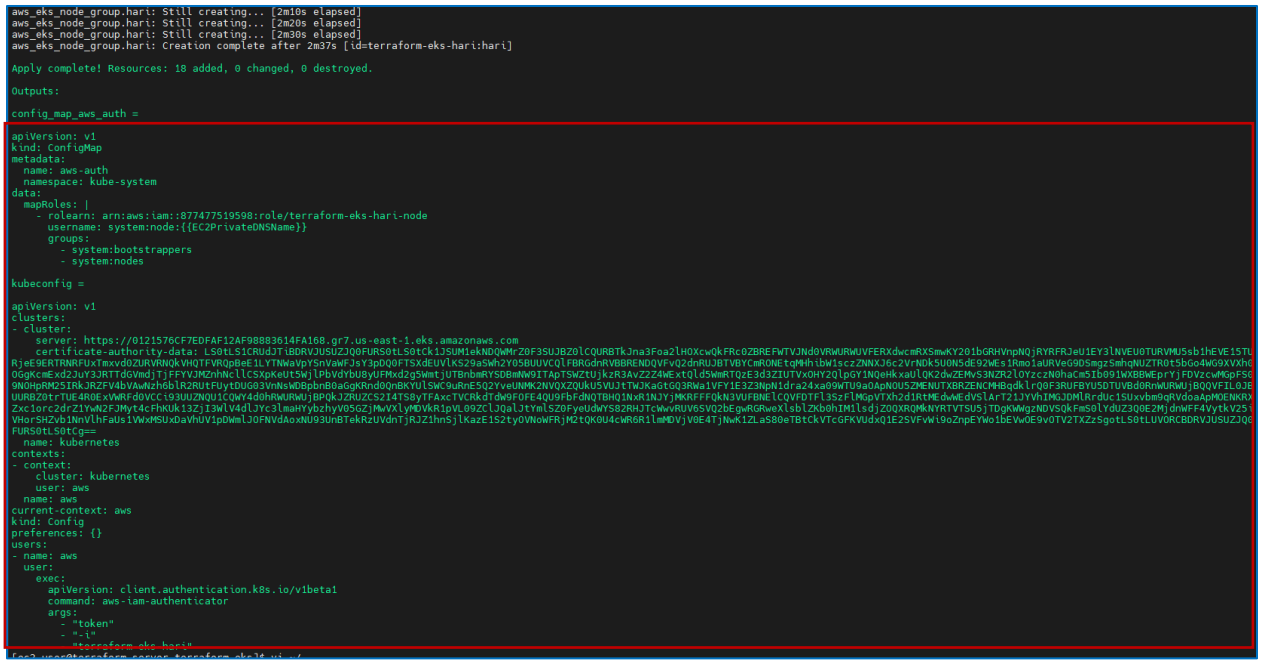


$ terraform plan

```
[ec2-user@terraform_server terraform_eks]$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.http.workstation-external-ip: Refreshing state...
data.aws_availability_zones.available: Refreshing state...

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

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_eks_cluster.hari will be created
  + resource "aws_eks_cluster" "hari" {
      + arn                   = (known after apply)
      + certificate_authority = (known after apply)
      + created_at            = (known after apply)
      + endpoint              = (known after apply)
      + id                    = (known after apply)
      + identity              = (known after apply)
      + name                  = "terraform-eks-hari"
      + platform_version      = (known after apply)
      + role_arn              = (known after apply)
      + status                = (known after apply)
      + version               = "1.20"

      + kubernetes_network_config {
          + service_ipv4_cidr = (known after apply)
        }

      + vpc_config {
          + cluster_security_group_id = (known after apply)
          + endpoint_private_access   = false
          + endpoint_public_access    = true
          + public_access_cidrs       = (known after apply)
          + security_group_ids        = (known after apply)
          + subnet_ids                = (known after apply)
          + vpc_id                    = (known after apply)
        }
    }

  # aws_eks_node_group.hari will be created
  + resource "aws_eks_node_group" "hari" {
      + ami_type        = (known after apply)
      + arn             = (known after apply)
      + capacity_type   = (known after apply)
      + cluster_name    = "terraform-eks-hari"
      + disk_size       = 10
      + id              = (known after apply)
      + instance_types  = [
          + "t2.medium",
        ]
      + node_group_name = "hari"
      + node_role_arn   = (known after apply)
```

```
      + arn                                  = (known after apply)
      + assign_ipv6_address_on_creation      = false
      + availability_zone                    = "us-east-1b"
      + availability_zone_id                 = (known after apply)
      + cidr_block                           = "10.0.1.0/24"
      + id                                   = (known after apply)
      + ipv6_cidr_block_association_id       = (known after apply)
      + map_public_ip_on_launch              = true
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name"                                        = "terraform-eks-hari-node"
          + "kubernetes.io/cluster/terraform-eks-hari" = "shared"
        }
      + tags_all                             = {
          + "Name"                                        = "terraform-eks-hari-node"
          + "kubernetes.io/cluster/terraform-eks-hari" = "shared"
        }
      + vpc_id                               = (known after apply)
    }

  # aws_vpc.hari will be created
  + resource "aws_vpc" "hari" {
      + arn                                = (known after apply)
      + assign_generated_ipv6_cidr_block = false
      + cidr_block                         = "10.0.0.0/16"
      + default_network_acl_id             = (known after apply)
      + default_route_table_id             = (known after apply)
      + default_security_group_id          = (known after apply)
      + dhcp_options_id                    = (known after apply)
      + enable_classiclink                 = (known after apply)
      + enable_classiclink_dns_support     = (known after apply)
      + enable_dns_hostnames               = (known after apply)
      + enable_dns_support                 = true
      + id                                 = (known after apply)
      + instance_tenancy                   = "default"
      + ipv6_association_id                = (known after apply)
      + ipv6_cidr_block                    = (known after apply)
      + main_route_table_id                = (known after apply)
      + owner_id                           = (known after apply)
      + tags                               = {
          + "Name"                                        = "terraform-eks-hari-node"
          + "kubernetes.io/cluster/terraform-eks-hari" = "shared"
        }
      + tags_all                           = {
          + "Name"                                        = "terraform-eks-hari-node"
          + "kubernetes.io/cluster/terraform-eks-hari" = "shared"
        }
    }

Plan: 18 to add, 0 to change, 0 to destroy.

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

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.

[ec2-user@terraform_server terraform_eks]$
```

$ terraform apply



$ aws eks update-kubeconfig --name terraform-eks-hari --region us-east-1
Now copy the selected content in above image and paste it in below file
$ vi ~/.kube/config
After pasteing edit the below line
From :: client.authentication.k8s.io/v1beta1
To :: client.authentication.k8s.io/v1alpha1

Result ::

$ kubectl get all

```
[ec2-user@terraform_server ~]$ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   172.20.0.1   <none>        443/TCP   23m
[ec2-user@terraform_server ~]$
```

$ kubectl version --short

```
[ec2-user@terraform_server ~]$ kubectl version --short
Client Version: v1.21.2-13+d2965f0db10712
Server Version: v1.20.15-eks-84b4fe6
[ec2-user@terraform_server ~]$
```

$ kubectl create deployment nginx --image=nginx

```
[ec2-user@terraform_server ~]$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
[ec2-user@terraform_server ~]$
```

$ kubectl create deployment httpd --image=httpd

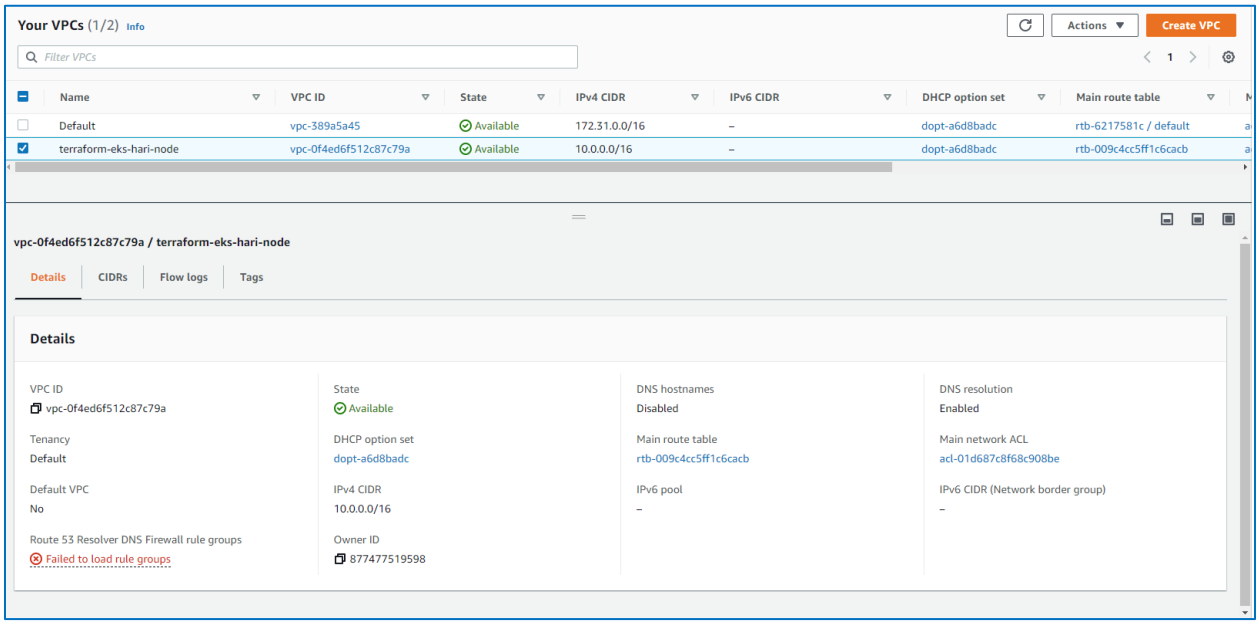
```
[ec2-user@terraform_server terraform_eks]$ kubectl create deployment httpd --image=httpd
deployment.apps/httpd created
[ec2-user@terraform_server terraform_eks]$
```

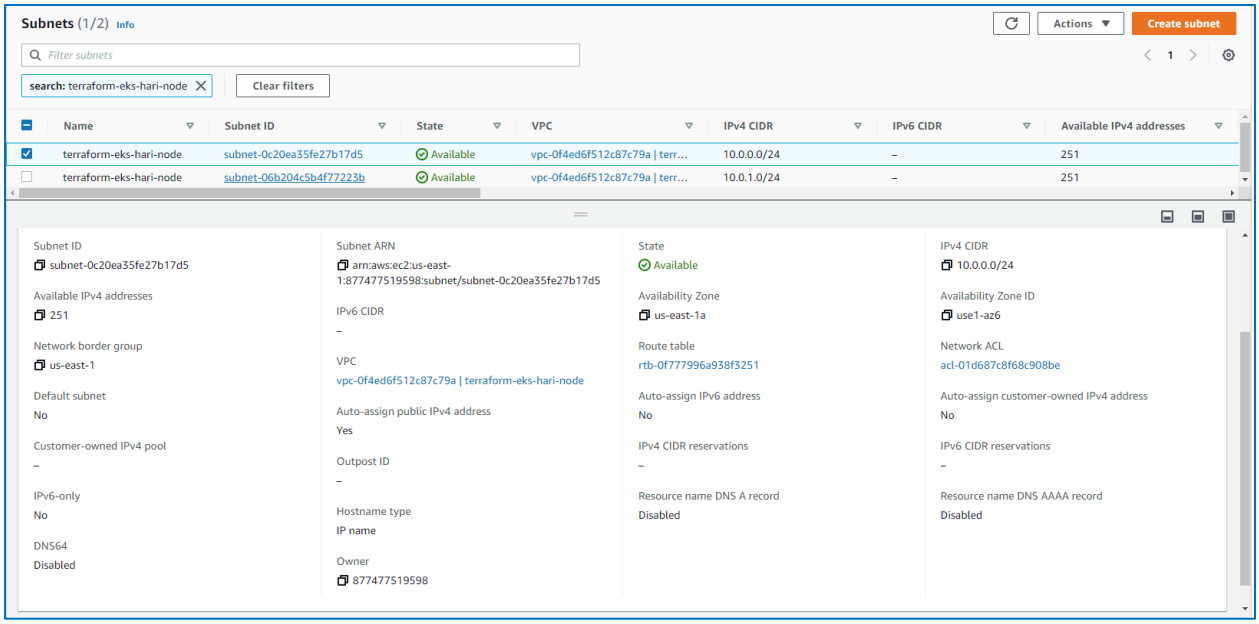
$ kubectl get all -o wide

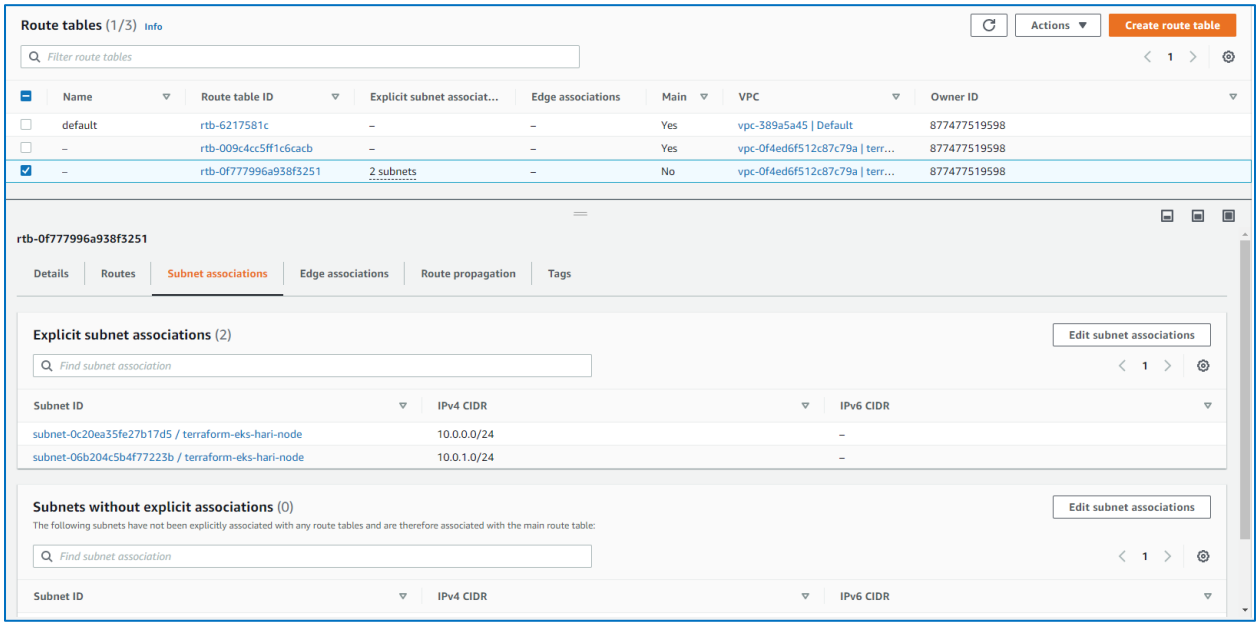
```
[ec2-user@terraform_server terraform_eks]$ kubectl get all -o wide
NAME                         READY   STATUS    RESTARTS   AGE   IP          NODE                           NOMINATED NODE   READINESS GATES
pod/httpd-757fb56c8d-2tkc5   1/1     Running   0          40s   10.0.0.7    ip-10-0-0-85.ec2.internal      <none>           <none>
pod/nginx-6799fc88d8-5cnc5   1/1     Running   0          16m   10.0.1.45   ip-10-0-1-124.ec2.internal     <none>           <none>

NAME                 TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE   SELECTOR
service/kubernetes   ClusterIP   172.20.0.1    <none>        443/TCP   44m   <none>

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES   SELECTOR
deployment.apps/httpd   1/1     1            1           40s   httpd        httpd    app=httpd
deployment.apps/nginx   1/1     1            1           16m   nginx        nginx    app=nginx

NAME                               DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES   SELECTOR
replicaset.apps/httpd-757fb56c8d   1         1         1       40s   httpd        httpd    app=httpd,pod-template-hash=757fb56c8d
replicaset.apps/nginx-6799fc88d8   1         1         1       16m   nginx        nginx    app=nginx,pod-template-hash=6799fc88d8
[ec2-user@terraform_server terraform_eks]$
```

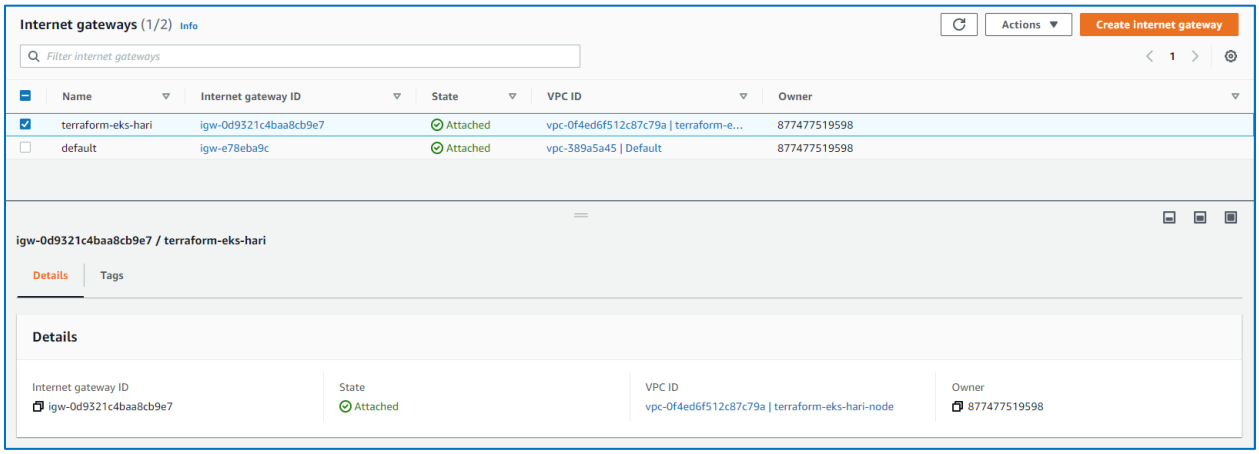
$ kubectl get pods -o wide

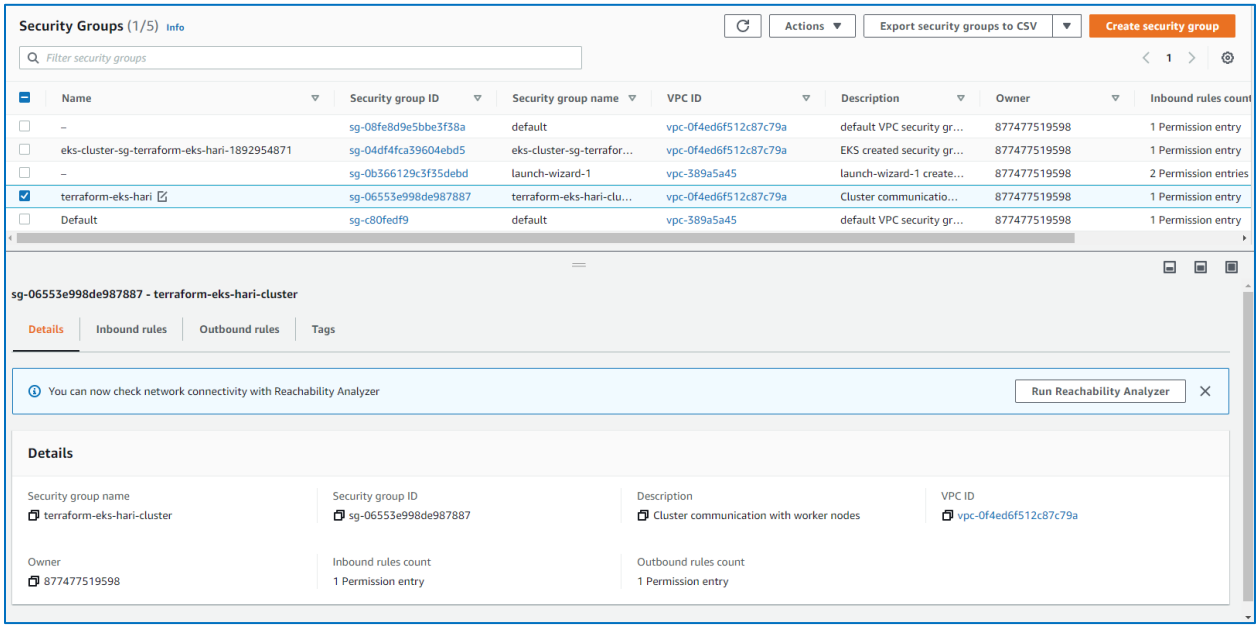
```
[ec2-user@terraform_server terraform_eks]$ kubectl get pods -o wide
NAME                     READY   STATUS    RESTARTS   AGE   IP          NODE                           NOMINATED NODE   READINESS GATES
httpd-757fb56c8d-2tkc5   1/1     Running   0          67s   10.0.0.7    ip-10-0-0-85.ec2.internal      <none>           <none>
nginx-6799fc88d8-5cnc5   1/1     Running   0          17m   10.0.1.45   ip-10-0-1-124.ec2.internal     <none>           <none>
[ec2-user@terraform_server terraform_eks]$
```

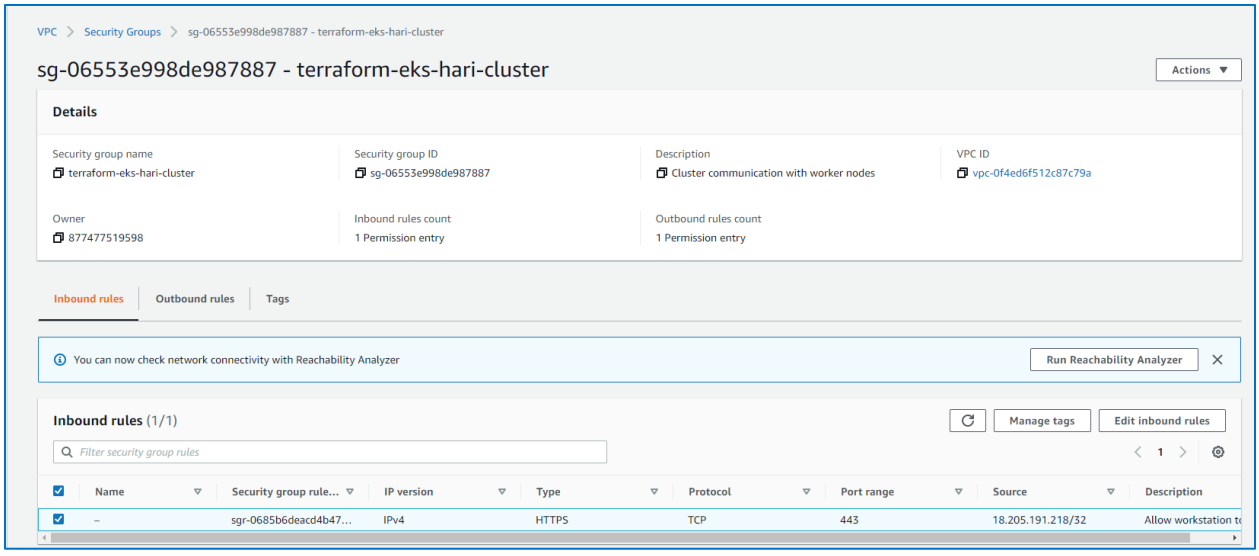


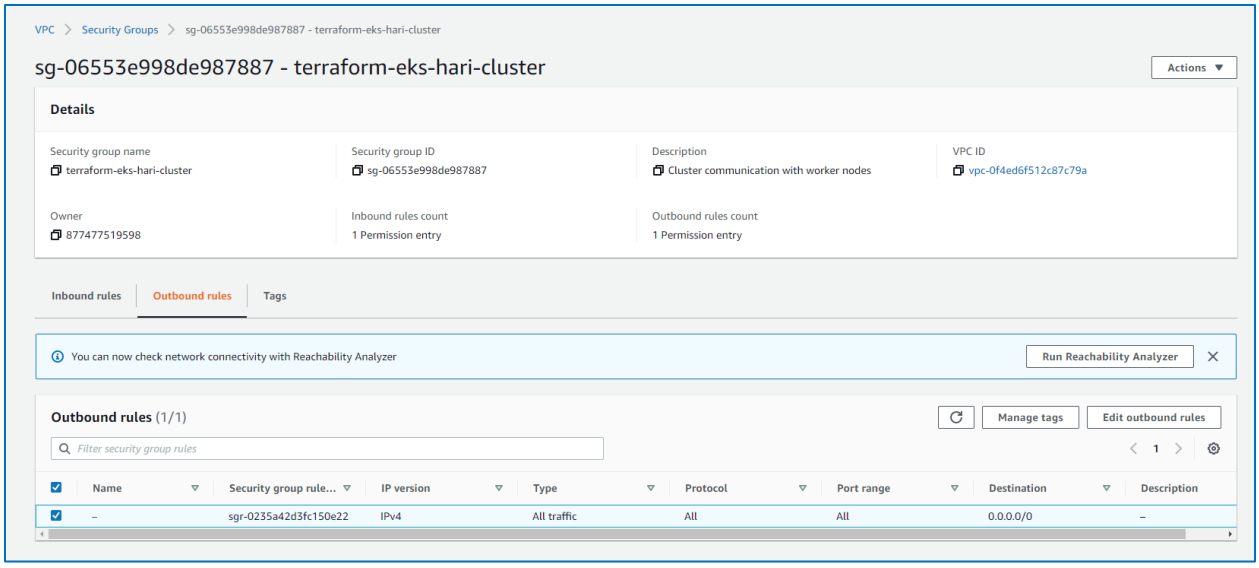

Subnets (1/2) Info
Actions
Create subnet
Filter subnets
search: terraform-eks-hari-node
Clear filters
Name
Subnet ID
State
VPC
IPv4 CIDR
IPv6 CIDR
Available IPv4 addresses
terraform-eks-hari-node
subnet-0c20ea35fe27b17d5
Available
vpc-0f4ed6f512c87c79a | terr...
10.0.0.0/24
–
251
terraform-eks-hari-node
subnet-06b204c5b4f77223b
Available
vpc-0f4ed6f512c87c79a | terr...
10.0.1.0/24
–
251
Subnet ID
subnet-0c20ea35fe27b17d5
Subnet ARN
arn:aws:ec2:us-east-1:877477519598:subnet/subnet-0c20ea35fe27b17d5
State
Available
IPv4 CIDR
10.0.0.0/24
Available IPv4 addresses
251
IPv6 CIDR
–
Availability Zone
us-east-1a
Availability Zone ID
use1-az6
Network border group
us-east-1
VPC
vpc-0f4ed6f512c87c79a | terraform-eks-hari-node
Route table
rtb-0f777996a938f3251
Network ACL
acl-01d687c8f68c908be
Default subnet
No
Auto-assign public IPv4 address
Yes
Auto-assign IPv6 address
No
Auto-assign customer-owned IPv4 address
No
Customer-owned IPv4 pool
–
Outpost ID
–
IPv4 CIDR reservations
–
IPv6 CIDR reservations
–
IPv6-only
No
Hostname type
IP name
Resource name DNS A record
Disabled
Resource name DNS AAAA record
Disabled
DNS64
Disabled
Owner
877477519598

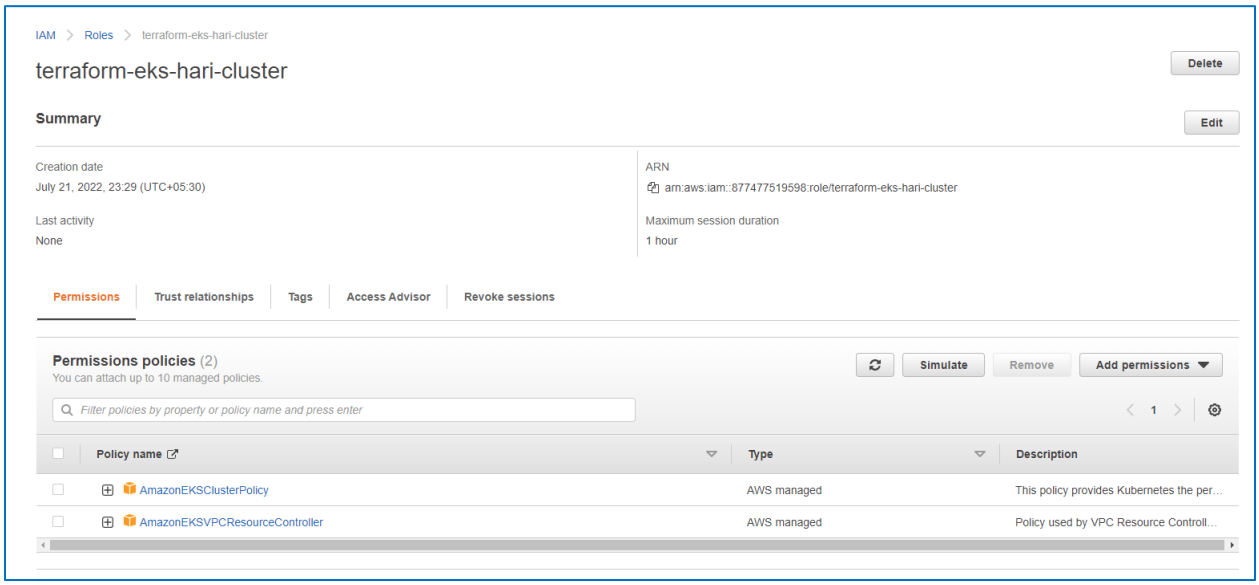

Route tables (1/3) Info
Actions
Create route table
Filter route tables
Name
Route table ID
Explicit subnet associat...
Edge associations
Main
VPC
Owner ID
default
rtb-6217581c
–
–
Yes
vpc-389a5a45 | Default
877477519598
–
rtb-009c4cc5ff1c6cacb
–
–
Yes
vpc-0f4ed6f512c87c79a | terr...
877477519598
–
rtb-0f777996a938f3251
2 subnets
–
No
vpc-0f4ed6f512c87c79a | terr...
877477519598
rtb-0f777996a938f3251
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
Explicit subnet associations (2)
Edit subnet associations
Find subnet association
Subnet ID
IPv4 CIDR
IPv6 CIDR
subnet-0c20ea35fe27b17d5 / terraform-eks-hari-node
10.0.0.0/24
–
subnet-06b204c5b4f77223b / terraform-eks-hari-node
10.0.1.0/24
–
Subnets without explicit associations (0)
Edit subnet associations
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:
Find subnet association
Subnet ID
IPv4 CIDR
IPv6 CIDR

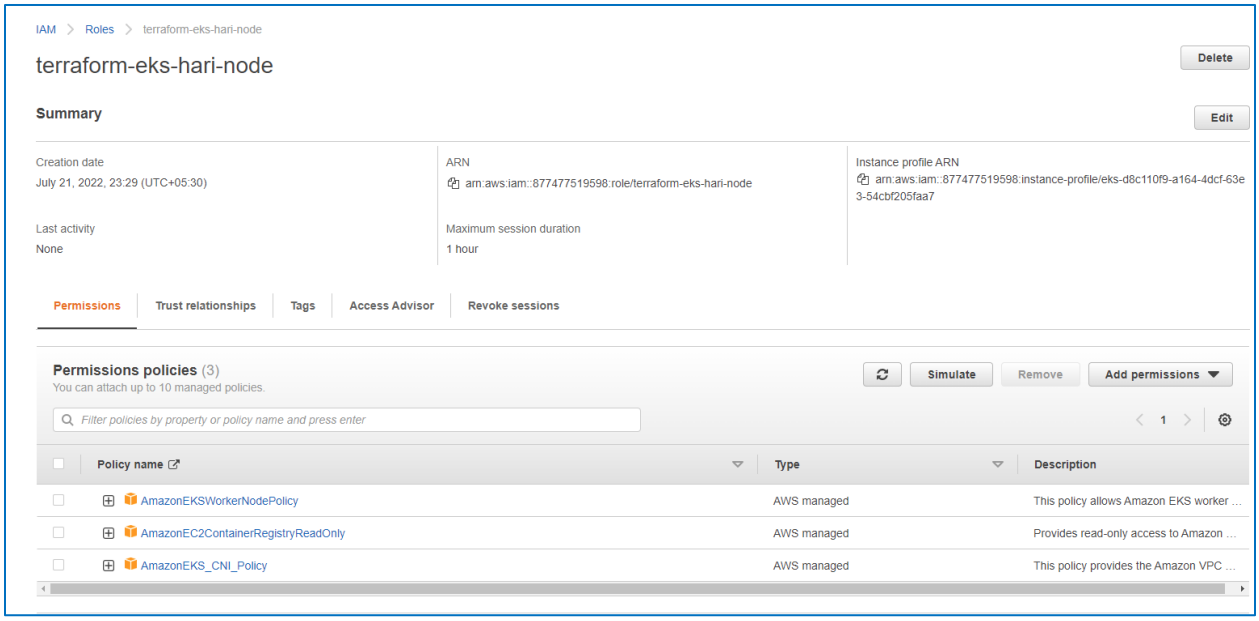

Internet gateways (1/2) Info
Actions
Create internet gateway
Filter internet gateways
Name
Internet gateway ID
State
VPC ID
Owner
terraform-eks-hari
igw-0d9321c4baa8cb9e7
Attached
vpc-0f4ed6f512c87c79a | terraform-e...
877477519598
default
igw-e78eba9c
Attached
vpc-389a5a45 | Default
877477519598
igw-0d9321c4baa8cb9e7 / terraform-eks-hari
Details
Tags
Details
Internet gateway ID
igw-0d9321c4baa8cb9e7
State
Attached
VPC ID
vpc-0f4ed6f512c87c79a | terraform-eks-hari-node
Owner
877477519598

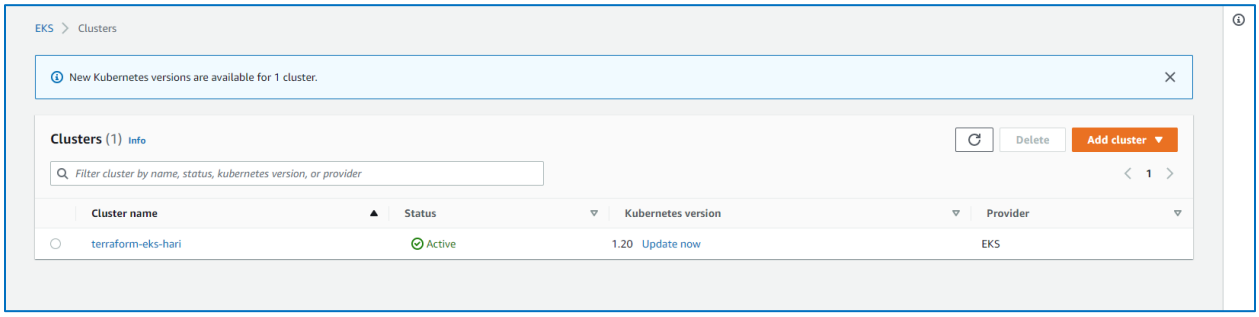

Security Groups (1/5) Info
Actions
Export security groups to CSV
Create security group
Filter security groups
Name
Security group ID
Security group name
VPC ID
Description
Owner
Inbound rules count
sg-08fe8d9e5bbe3f38a
default
vpc-0f4ed6f512c87c79a
default VPC security gr...
877477519598
1 Permission entry
eks-cluster-sg-terraform-eks-hari-1892954871
sg-04df4fca39604ebd5
eks-cluster-sg-terrafor...
EKS created security gr...
launch-wizard-1
sg-0b366129c3f35debd
vpc-389a5a45
launch-wizard-1 create...
2 Permission entries
terraform-eks-hari
sg-06553e998de987887
terraform-eks-hari-clu...
Cluster communicatio...
Default
sg-c80fedf9
sg-06553e998de987887 - terraform-eks-hari-cluster
Details
Inbound rules
Outbound rules
Tags
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Details
Security group name
terraform-eks-hari-cluster
Security group ID
sg-06553e998de987887
Description
Cluster communication with worker nodes
VPC ID
vpc-0f4ed6f512c87c79a
Owner
877477519598
Inbound rules count
1 Permission entry
Outbound rules count
1 Permission entry

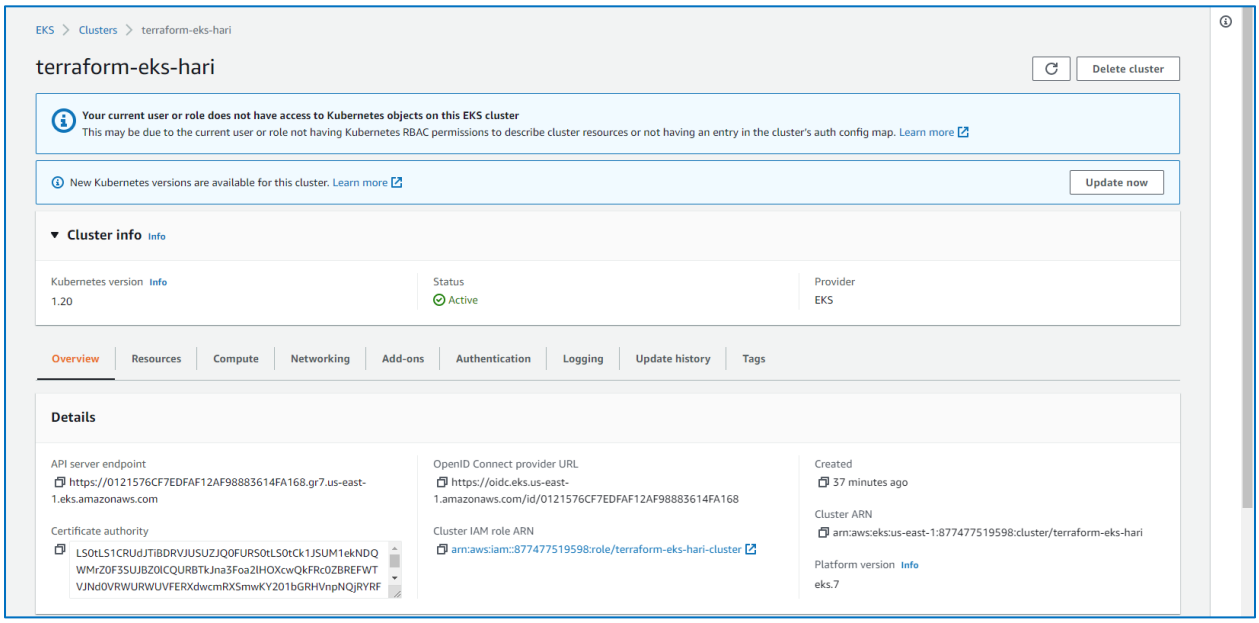

VPC
Security Groups
sg-06553e998de987887 - terraform-eks-hari-cluster
sg-06553e998de987887 - terraform-eks-hari-cluster
Actions
Details
Security group name
terraform-eks-hari-cluster
Security group ID
sg-06553e998de987887
Description
Cluster communication with worker nodes
VPC ID
vpc-0f4ed6f512c87c79a
Owner
877477519598
Inbound rules count
1 Permission entry
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Inbound rules (1/1)
Manage tags
Edit inbound rules
Filter security group rules
Name
Security group rule...
IP version
Type
Protocol
Port range
Source
Description
sgr-0685b6deacd4b47...
IPv4
HTTPS
TCP
443
18.205.191.218/32
Allow workstation t

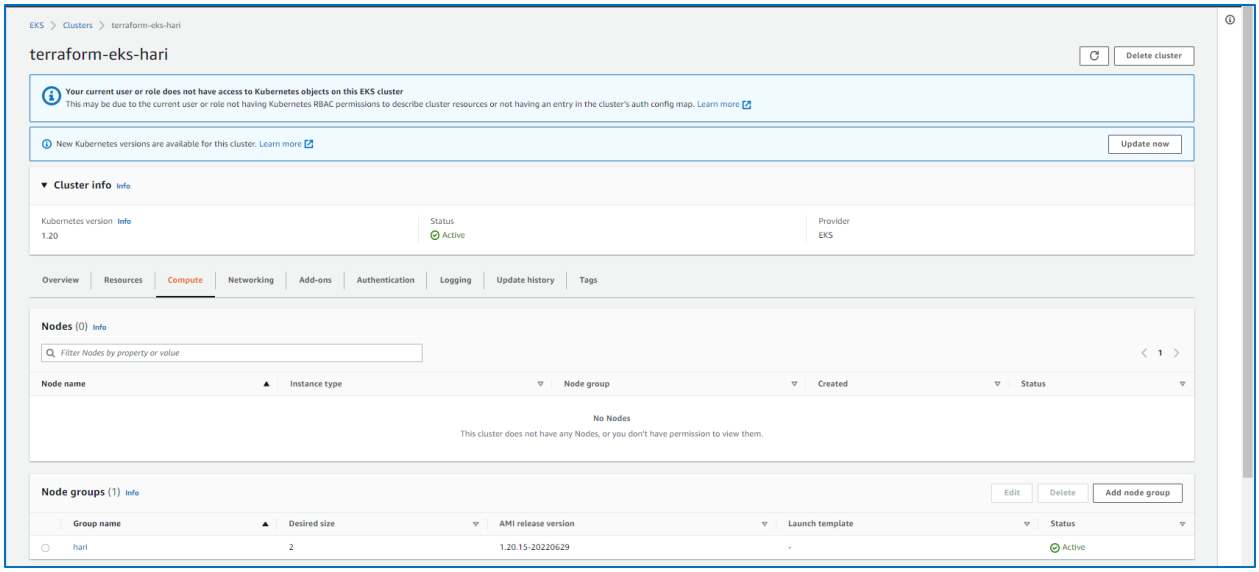

VPC > Security Groups > sg-06553e998de987887 - terraform-eks-hari-cluster
sg-06553e998de987887 - terraform-eks-hari-cluster
Actions
Details
Security group name
terraform-eks-hari-cluster
Security group ID
sg-06553e998de987887
Description
Cluster communication with worker nodes
VPC ID
vpc-0f4ed6f512c87c79a
Owner
877477519598
Inbound rules count
1 Permission entry
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Outbound rules (1/1)
Manage tags
Edit outbound rules
Filter security group rules
Name
Security group rule...
IP version
Type
Protocol
Port range
Destination
Description
–
sgr-0235a42d3fc150e22
IPv4
All traffic
All
All
0.0.0.0/0
–

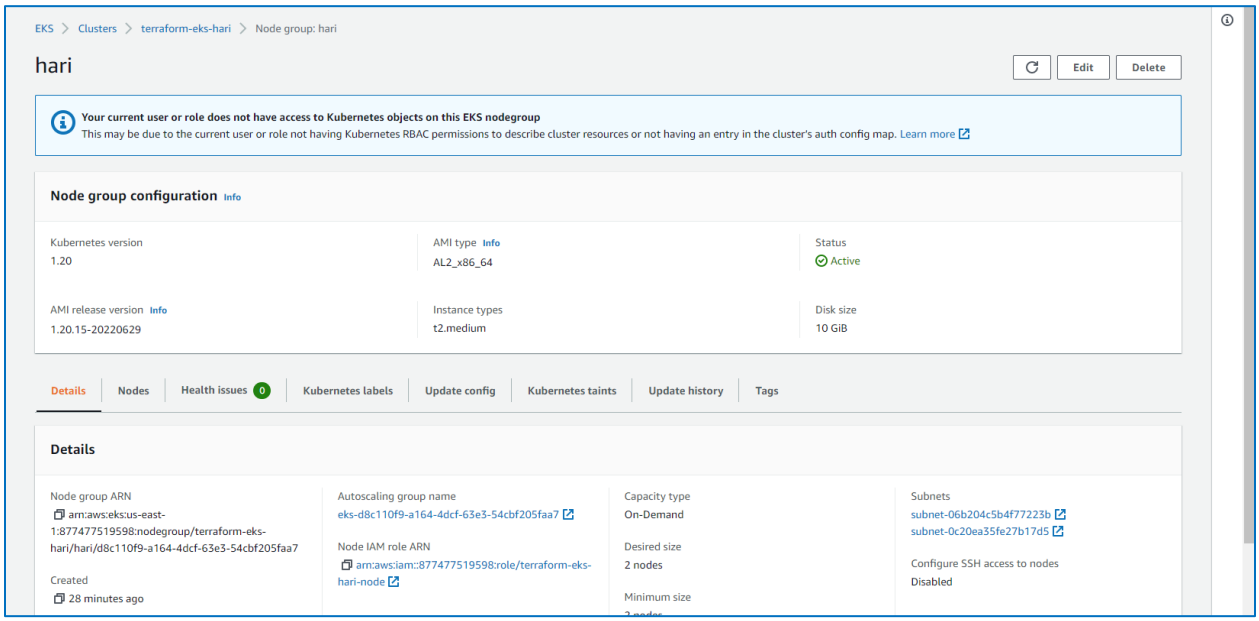

IAM > Roles > terraform-eks-hari-cluster
terraform-eks-hari-cluster
Delete
Summary
Edit
Creation date
July 21, 2022, 23:29 (UTC+05:30)
ARN
arn:aws:iam::877477519598:role/terraform-eks-hari-cluster
Last activity
None
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (2)
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter policies by property or policy name and press enter
Policy name
Type
Description
AmazonEKSClusterPolicy
AWS managed
This policy provides Kubernetes the per...
AmazonEKSVPCResourceController
AWS managed
Policy used by VPC Resource Controll...

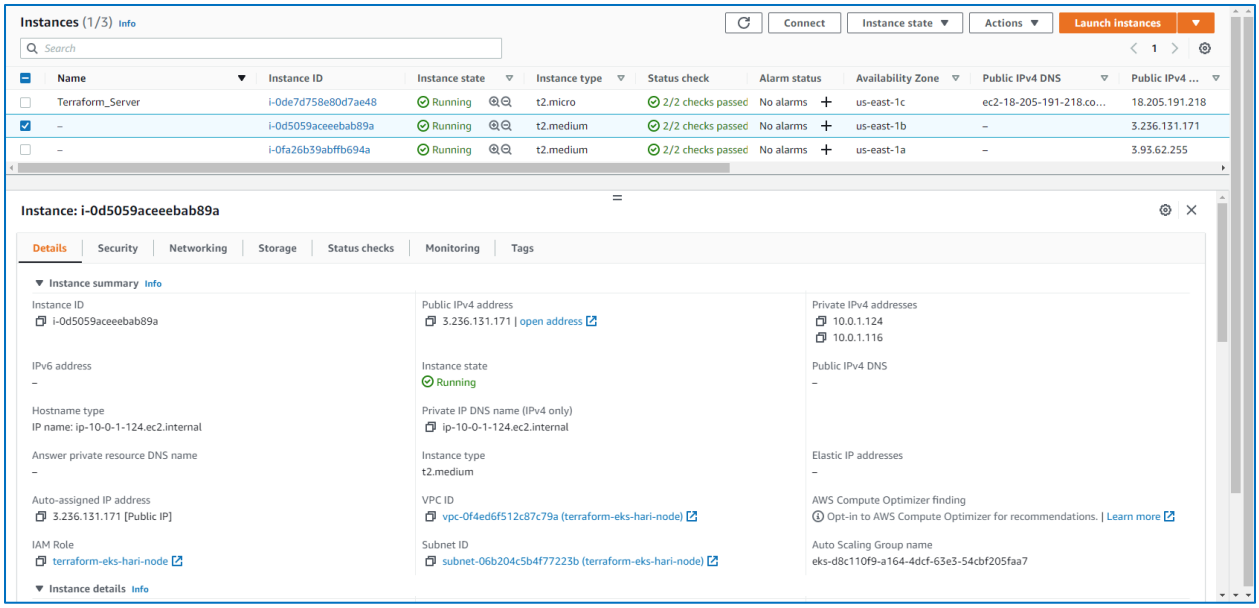

IAM > Roles > terraform-eks-hari-node
terraform-eks-hari-node
Delete
Summary
Edit
Creation date
July 21, 2022, 23:29 (UTC+05:30)
ARN
arn:aws:iam::877477519598:role/terraform-eks-hari-node
Instance profile ARN
arn:aws:iam::877477519598:instance-profile/eks-d8c110f9-a164-4dcf-63e3-54cbf205faa7
Last activity
None
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (3)
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter policies by property or policy name and press enter
Policy name
Type
Description
AmazonEKSWorkerNodePolicy
AWS managed
This policy allows Amazon EKS worker ...
AmazonEC2ContainerRegistryReadOnly
AWS managed
Provides read-only access to Amazon ...
AmazonEKS_CNI_Policy
AWS managed
This policy provides the Amazon VPC ...


EKS > Clusters
New Kubernetes versions are available for 1 cluster.
Clusters (1) Info
Delete
Add cluster
Filter cluster by name, status, kubernetes version, or provider
Cluster name
Status
Kubernetes version
Provider
terraform-eks-hari
Active
1.20 Update now
EKS


EKS > Clusters > terraform-eks-hari
terraform-eks-hari
Delete cluster
Your current user or role does not have access to Kubernetes objects on this EKS cluster
This may be due to the current user or role not having Kubernetes RBAC permissions to describe cluster resources or not having an entry in the cluster's auth config map. Learn more
New Kubernetes versions are available for this cluster. Learn more
Update now
Cluster info Info
Kubernetes version Info
1.20
Status
Active
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Authentication
Logging
Update history
Tags
Details
API server endpoint
https://0121576CF7EDFAF12AF98883614FA168.gr7.us-east-1.eks.amazonaws.com
Certificate authority
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM1ekNDQWMrZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURVUVFERXdwcmRXSmwxY201bGRHVnpNQjRYRF
OpenID Connect provider URL
https://oidc.eks.us-east-1.amazonaws.com/id/0121576CF7EDFAF12AF98883614FA168
Cluster IAM role ARN
arn:aws:iam::877477519598:role/terraform-eks-hari-cluster
Created
37 minutes ago
Cluster ARN
arn:aws:eks:us-east-1:877477519598:cluster/terraform-eks-hari
Platform version Info
eks.7


EKS > Clusters > terraform-eks-hari
terraform-eks-hari
Delete cluster
Your current user or role does not have access to Kubernetes objects on this EKS cluster
This may be due to the current user or role not having Kubernetes RBAC permissions to describe cluster resources or not having an entry in the cluster's auth config map. Learn more
New Kubernetes versions are available for this cluster. Learn more
Update now
Cluster info Info
Kubernetes version Info
1.20
Status
Active
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Authentication
Logging
Update history
Tags
Nodes (0) Info
Filter Nodes by property or value
Node name
Instance type
Node group
Created
Status
No Nodes
This cluster does not have any Nodes, or you don't have permission to view them.
Node groups (1) Info
Edit
Delete
Add node group
Group name
Desired size
AMI release version
Launch template
Status
hari
2
1.20.15-20220629
-
Active


EKS > Clusters > terraform-eks-hari > Node group: hari
hari
Edit
Delete
Your current user or role does not have access to Kubernetes objects on this EKS nodegroup
This may be due to the current user or role not having Kubernetes RBAC permissions to describe cluster resources or not having an entry in the cluster's auth config map. Learn more
Node group configuration Info
Kubernetes version
1.20
AMI type Info
AL2_x86_64
Status
Active
AMI release version Info
1.20.15-20220629
Instance types
t2.medium
Disk size
10 GiB
Details
Nodes
Health issues 0
Kubernetes labels
Update config
Kubernetes taints
Update history
Tags
Details
Node group ARN
arn:aws:eks:us-east-1:877477519598:nodegroup/terraform-eks-hari/hari/d8c110f9-a164-4dcf-63e3-54cbf205faa7
Created
28 minutes ago
Autoscaling group name
eks-d8c110f9-a164-4dcf-63e3-54cbf205faa7
Node IAM role ARN
arn:aws:iam::877477519598:role/terraform-eks-hari-node
Capacity type
On-Demand
Desired size
2 nodes
Minimum size
Subnets
subnet-06b204c5b4f77223b
subnet-0c20ea35fe27b17d5
Configure SSH access to nodes
Disabled


Instances (1/3) Info
Search
Connect
Instance state
Actions
Launch instances
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Terraform_Server
i-0de7d758e80d7ae48
Running
t2.micro
2/2 checks passed
No alarms
us-east-1c
ec2-18-205-191-218.co...
18.205.191.218
–
i-0d5059aceeebab89a
Running
t2.medium
2/2 checks passed
No alarms
us-east-1b
–
3.236.131.171
–
i-0fa26b39abffb694a
Running
t2.medium
2/2 checks passed
No alarms
us-east-1a
–
3.93.62.255
Instance: i-0d5059aceeebab89a
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0d5059aceeebab89a
Public IPv4 address
3.236.131.171 | open address
Private IPv4 addresses
10.0.1.124
10.0.1.116
IPv6 address
–
Instance state
Running
Public IPv4 DNS
–
Hostname type
IP name: ip-10-0-1-124.ec2.internal
Private IP DNS name (IPv4 only)
ip-10-0-1-124.ec2.internal
Answer private resource DNS name
–
Instance type
t2.medium
Elastic IP addresses
–
Auto-assigned IP address
3.236.131.171 [Public IP]
VPC ID
vpc-0f4ed6f512c87c79a (terraform-eks-hari-node)
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role
terraform-eks-hari-node
Subnet ID
subnet-06b204c5b4f77223b (terraform-eks-hari-node)
Auto Scaling Group name
eks-d8c110f9-a164-4dcf-63e3-54cbf205faa7
Instance details Info