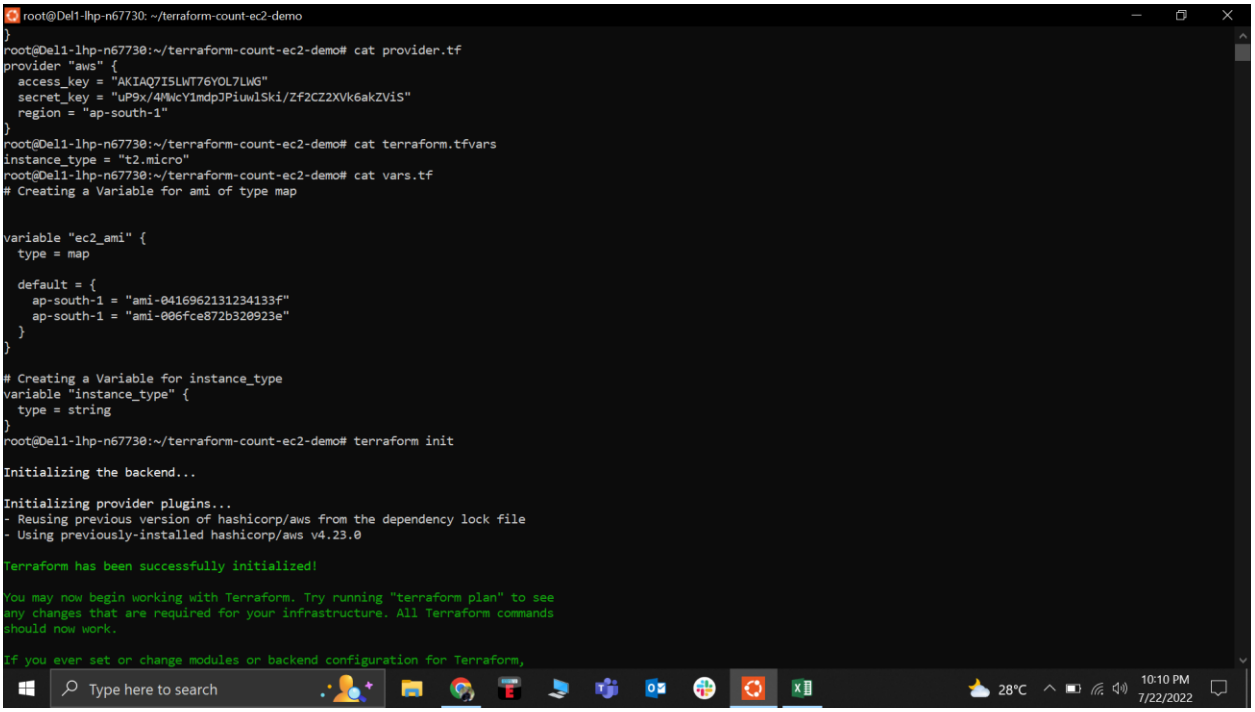
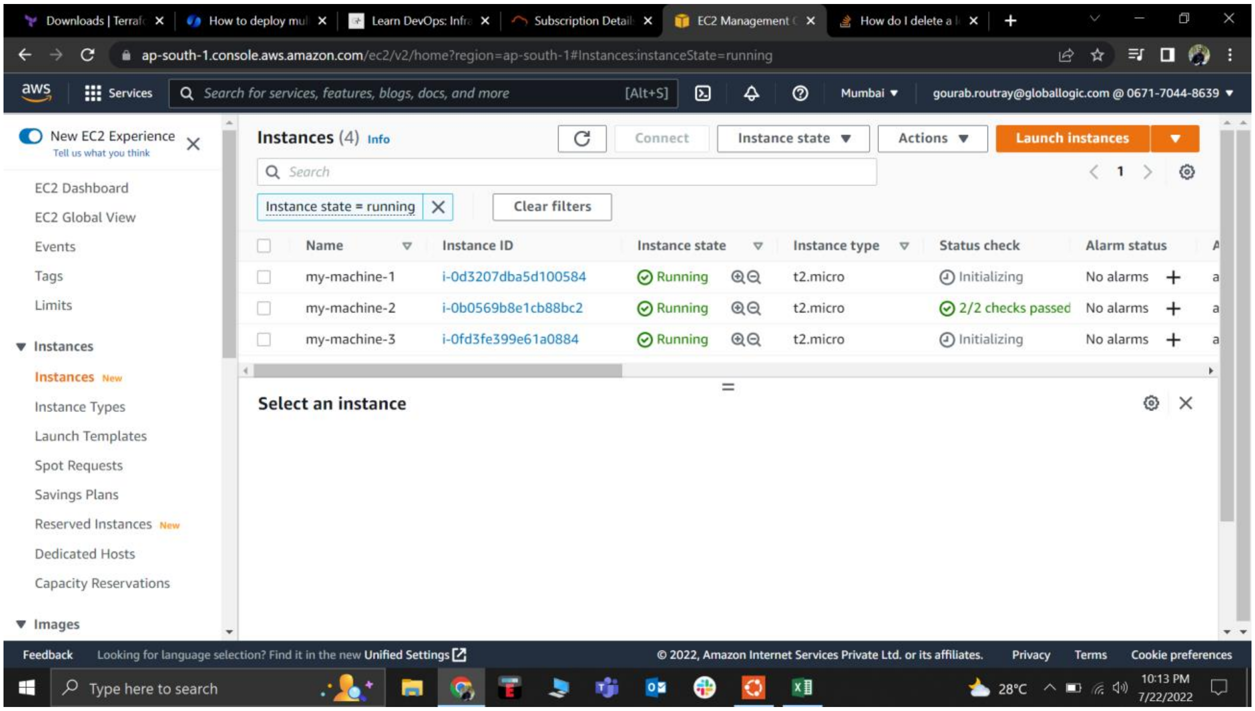
ASSIGNMENT-10

INSTALLED TERRAFORM AND CREATED TF FILES



```
}
root@Del1-lhp-n67730:~/terraform-count-ec2-demo# cat provider.tf
provider "aws" {
  access_key = "AKIAQ7I5LWT76YOL7LWG"
  secret_key = "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"
  region = "ap-south-1"
}
root@Del1-lhp-n67730:~/terraform-count-ec2-demo# cat terraform.tfvars
instance_type = "t2.micro"
root@Del1-lhp-n67730:~/terraform-count-ec2-demo# cat vars.tf
# Creating a Variable for ami of type map


variable "ec2_ami" {
  type = map

  default = {
    ap-south-1 = "ami-0416962131234133f"
    ap-south-1 = "ami-006fce872b320923e"
  }
}

# Creating a Variable for instance_type
variable "instance_type" {
  type = string
}
root@Del1-lhp-n67730:~/terraform-count-ec2-demo# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.23.0

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,
```

EC-2 INSTANCE CREATING



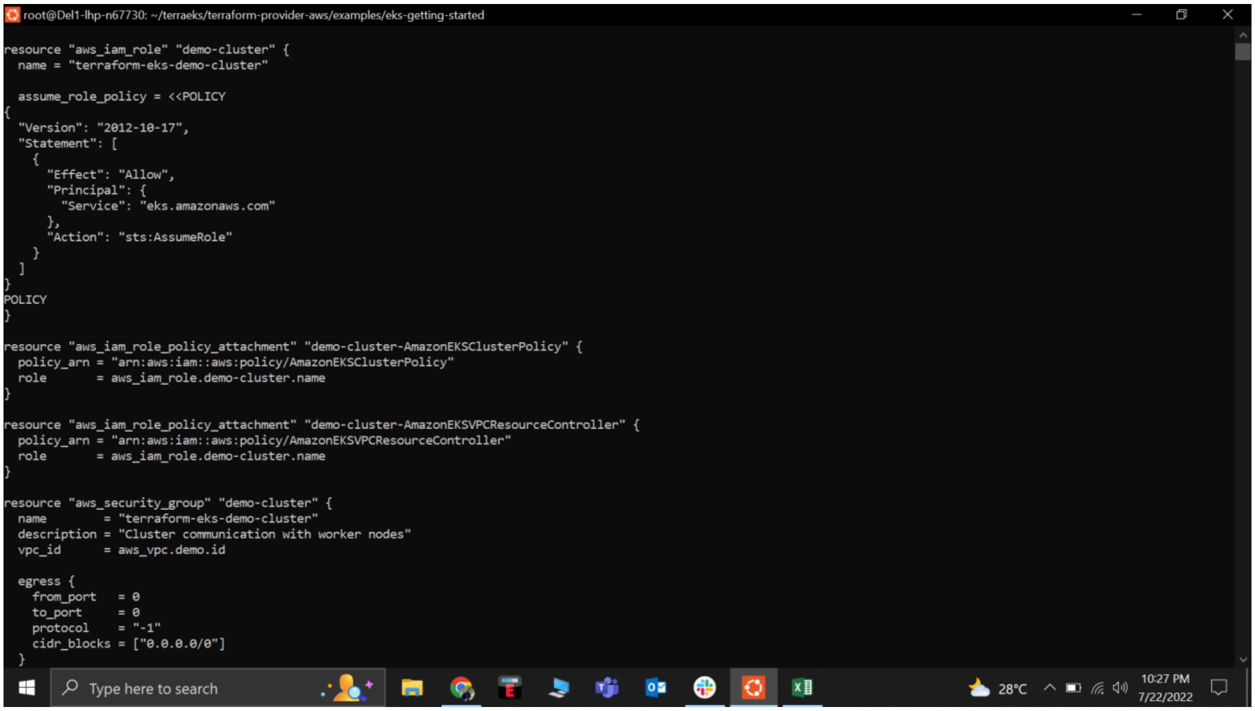
Instances (4) Info

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	A
□	my-machine-1	i-0d3207dba5d100584	Running	t2.micro	Initializing	No alarms	a
□	my-machine-2	i-0b0569b8e1cb88bc2	Running	t2.micro	2/2 checks passed	No alarms	a
□	my-machine-3	i-0fd3fe399e61a0884	Running	t2.micro	Initializing	No alarms	a

ASSIGNMENT-10

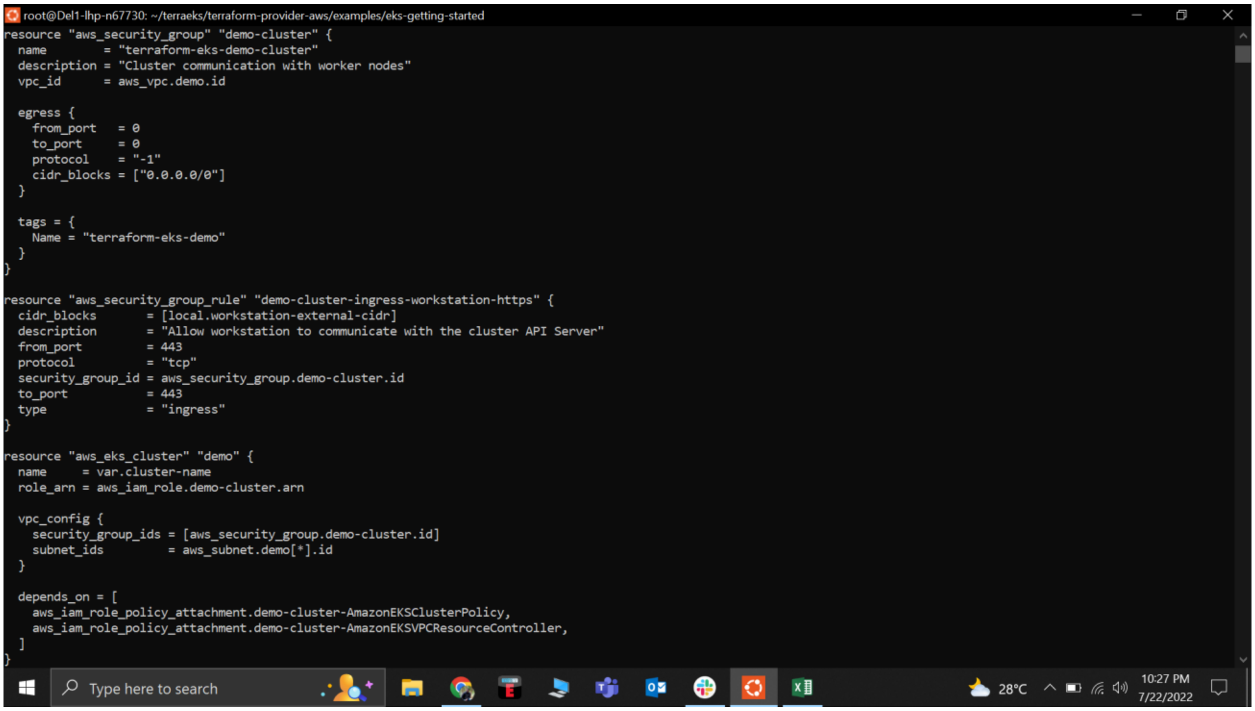
CREATING EKS CLUSTER



root@Del1-lhp-n67730: ~/terraeks/terraform-provider-aws/examples/eks-getting-started

```
resource "aws_iam_role" "demo-cluster" {
  name = "terraform-eks-demo-cluster"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy_attachment" "demo-cluster-AmazonEKSClusterPolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
  role       = aws_iam_role.demo-cluster.name
}

resource "aws_iam_role_policy_attachment" "demo-cluster-AmazonEKSVPCResourceController" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSVPCResourceController"
  role       = aws_iam_role.demo-cluster.name
}

resource "aws_security_group" "demo-cluster" {
  name        = "terraform-eks-demo-cluster"
  description = "Cluster communication with worker nodes"
  vpc_id      = aws_vpc.demo.id

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

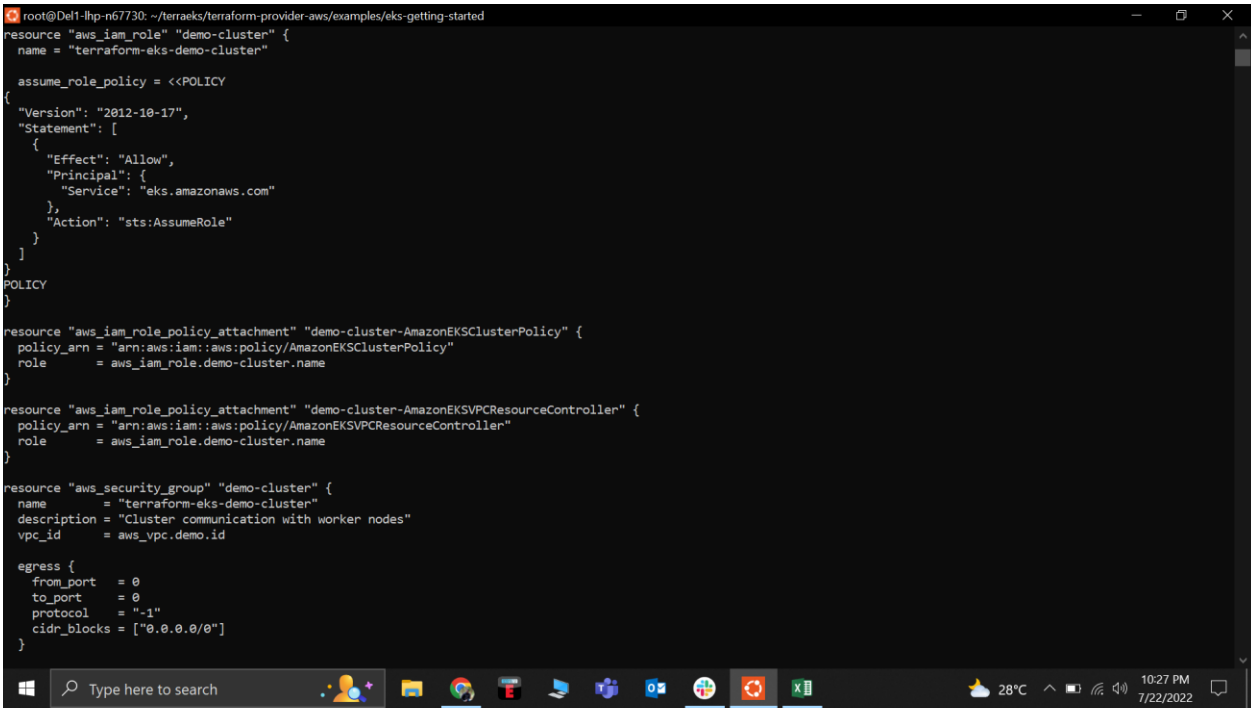
TF FILES



root@Del1-lhp-n67730: ~/terraeks/terraform-provider-aws/examples/eks-getting-started

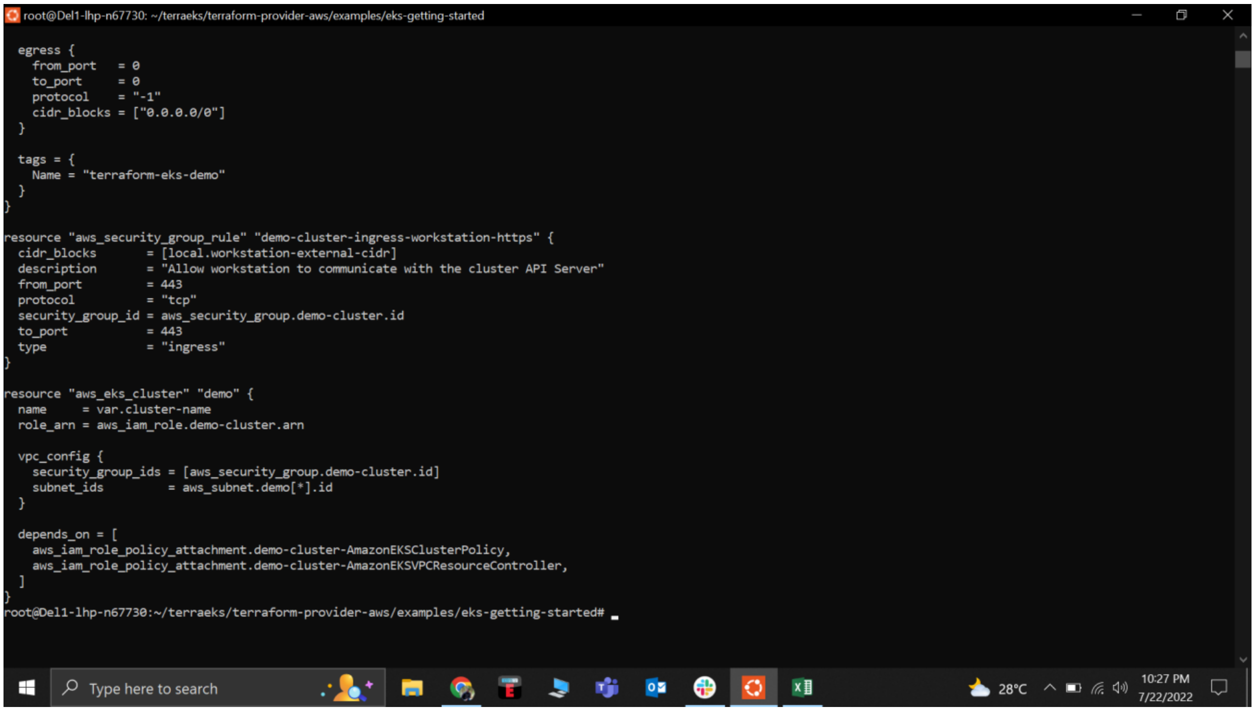
```
resource "aws_security_group" "demo-cluster" {
  name        = "terraform-eks-demo-cluster"
  description = "Cluster communication with worker nodes"
  vpc_id      = aws_vpc.demo.id

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "terraform-eks-demo"
  }
}

resource "aws_security_group_rule" "demo-cluster-ingress-workstation-https" {
  cidr_blocks       = [local.workstation-external-cidr]
  description       = "Allow workstation to communicate with the cluster API Server"
  from_port         = 443
  protocol          = "tcp"
  security_group_id = aws_security_group.demo-cluster.id
  to_port           = 443
  type              = "ingress"
}

resource "aws_eks_cluster" "demo" {
  name     = var.cluster-name
  role_arn = aws_iam_role.demo-cluster.arn

  vpc_config {
    security_group_ids = [aws_security_group.demo-cluster.id]
    subnet_ids         = aws_subnet.demo[*].id
  }

  depends_on = [
    aws_iam_role_policy_attachment.demo-cluster-AmazonEKSClusterPolicy,
    aws_iam_role_policy_attachment.demo-cluster-AmazonEKSVPCResourceController,
  ]
}
```

ASSIGNMENT-10



root@Del1-lhp-n67730: ~/terraeks/terraform-provider-aws/examples/eks-getting-started

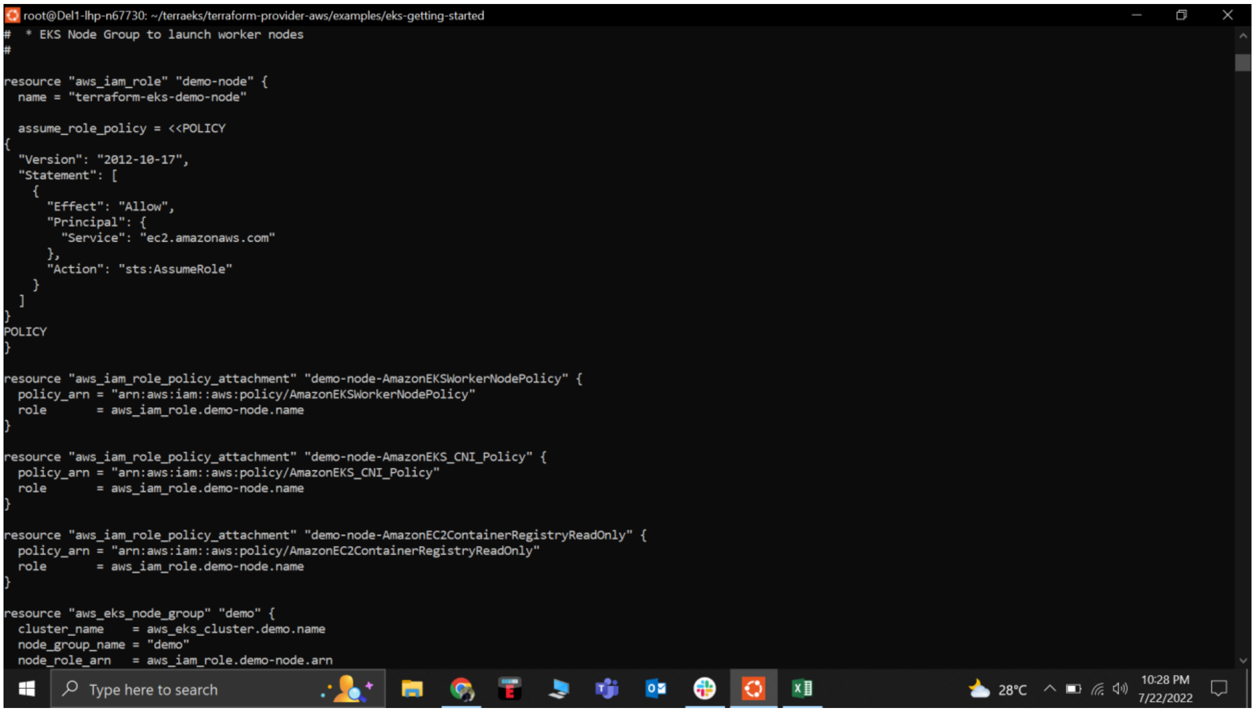
```
resource "aws_iam_role" "demo-cluster" {
  name = "terraform-eks-demo-cluster"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy_attachment" "demo-cluster-AmazonEKSClusterPolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
  role       = aws_iam_role.demo-cluster.name
}

resource "aws_iam_role_policy_attachment" "demo-cluster-AmazonEKSVPCResourceController" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSVPCResourceController"
  role       = aws_iam_role.demo-cluster.name
}

resource "aws_security_group" "demo-cluster" {
  name        = "terraform-eks-demo-cluster"
  description = "Cluster communication with worker nodes"
  vpc_id      = aws_vpc.demo.id

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
```



root@Del1-lhp-n67730: ~/terraeks/terraform-provider-aws/examples/eks-getting-started

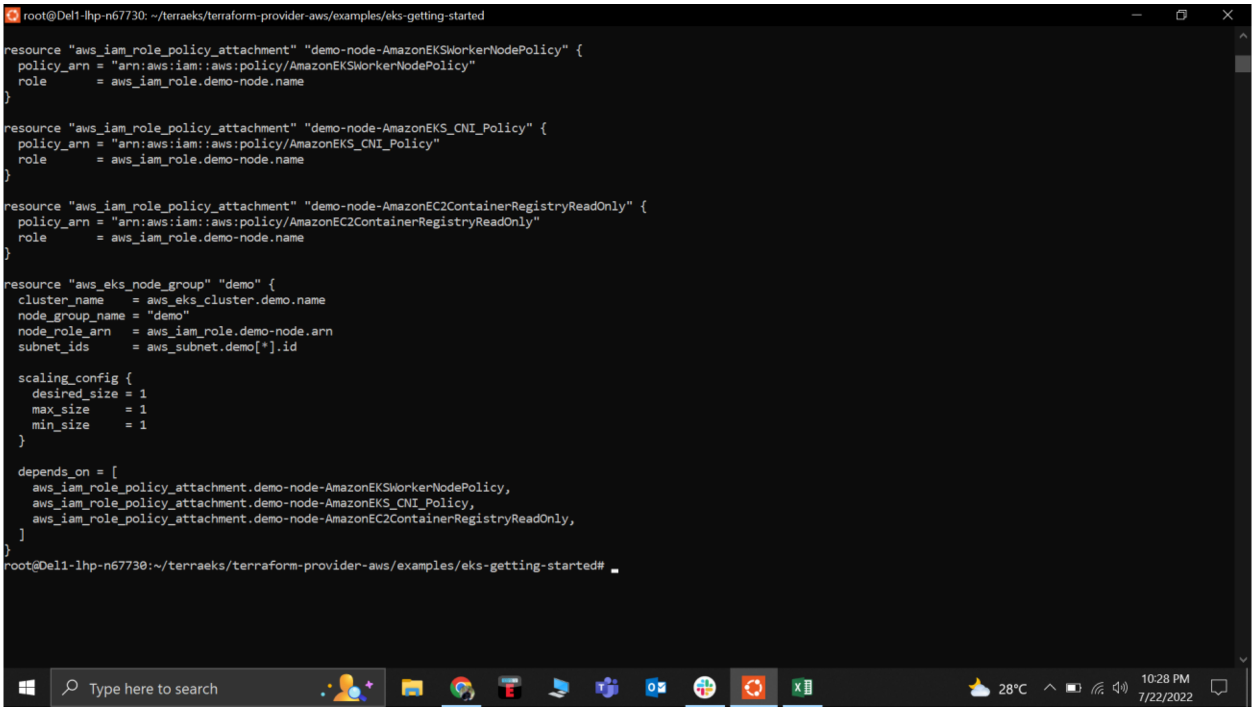
```
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "terraform-eks-demo"
  }
}

resource "aws_security_group_rule" "demo-cluster-ingress-workstation-https" {
  cidr_blocks       = [local.workstation-external-cidr]
  description       = "Allow workstation to communicate with the cluster API Server"
  from_port         = 443
  protocol          = "tcp"
  security_group_id = aws_security_group.demo-cluster.id
  to_port           = 443
  type              = "ingress"
}

resource "aws_eks_cluster" "demo" {
  name     = var.cluster-name
  role_arn = aws_iam_role.demo-cluster.arn

  vpc_config {
    security_group_ids = [aws_security_group.demo-cluster.id]
    subnet_ids         = aws_subnet.demo[*].id
  }

  depends_on = [
    aws_iam_role_policy_attachment.demo-cluster-AmazonEKSClusterPolicy,
    aws_iam_role_policy_attachment.demo-cluster-AmazonEKSVPCResourceController,
  ]
}
root@Del1-lhp-n67730:~/terraeks/terraform-provider-aws/examples/eks-getting-started# _
```

ASSIGNMENT-10



root@Del1-lhp-n67730: ~/terraeks/terraform-provider-aws/examples/eks-getting-started

```
#  * EKS Node Group to launch worker nodes
#

resource "aws_iam_role" "demo-node" {
  name = "terraform-eks-demo-node"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy_attachment" "demo-node-AmazonEKSWorkerNodePolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
  role       = aws_iam_role.demo-node.name
}

resource "aws_iam_role_policy_attachment" "demo-node-AmazonEKS_CNI_Policy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
  role       = aws_iam_role.demo-node.name
}

resource "aws_iam_role_policy_attachment" "demo-node-AmazonEC2ContainerRegistryReadOnly" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"
  role       = aws_iam_role.demo-node.name
}

resource "aws_eks_node_group" "demo" {
  cluster_name    = aws_eks_cluster.demo.name
  node_group_name = "demo"
  node_role_arn   = aws_iam_role.demo-node.arn
```

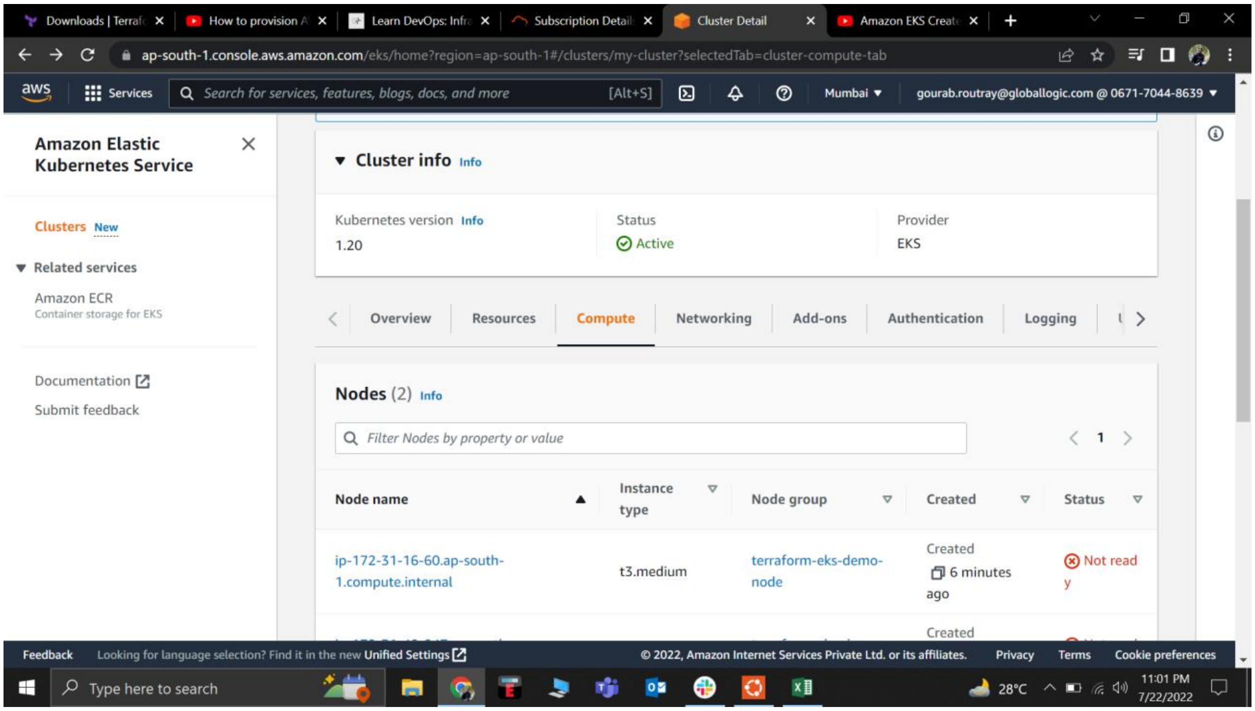


root@Del1-lhp-n67730: ~/terraeks/terraform-provider-aws/examples/eks-getting-started

```
resource "aws_iam_role_policy_attachment" "demo-node-AmazonEKSWorkerNodePolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
  role       = aws_iam_role.demo-node.name
}

resource "aws_iam_role_policy_attachment" "demo-node-AmazonEKS_CNI_Policy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
  role       = aws_iam_role.demo-node.name
}

resource "aws_iam_role_policy_attachment" "demo-node-AmazonEC2ContainerRegistryReadOnly" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"
  role       = aws_iam_role.demo-node.name
}

resource "aws_eks_node_group" "demo" {
  cluster_name    = aws_eks_cluster.demo.name
  node_group_name = "demo"
  node_role_arn   = aws_iam_role.demo-node.arn
  subnet_ids      = aws_subnet.demo[*].id

  scaling_config {
    desired_size = 1
    max_size     = 1
    min_size     = 1
  }

  depends_on = [
    aws_iam_role_policy_attachment.demo-node-AmazonEKSWorkerNodePolicy,
    aws_iam_role_policy_attachment.demo-node-AmazonEKS_CNI_Policy,
    aws_iam_role_policy_attachment.demo-node-AmazonEC2ContainerRegistryReadOnly,
  ]
}
root@Del1-lhp-n67730:~/terraeks/terraform-provider-aws/examples/eks-getting-started# _
```

ASSIGNMENT-10

EKS CLUSTER CREATED


Downloads | Terrafo
How to provision A
Learn DevOps: Infra
Subscription Details
Cluster Detail
Amazon EKS Create
ap-south-1.console.aws.amazon.com/eks/home?region=ap-south-1#/clusters/my-cluster?selectedTab=cluster-compute-tab
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
gourab.routray@globallogic.com @ 0671-7044-8639
Amazon Elastic Kubernetes Service
Clusters New
Related services
Amazon ECR
Container storage for EKS
Documentation
Submit feedback
Overview
Resources
Compute
Networking
Add-ons
Authentication
Logging
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
28°C
11:01 PM
7/22/2022

Cluster info Info

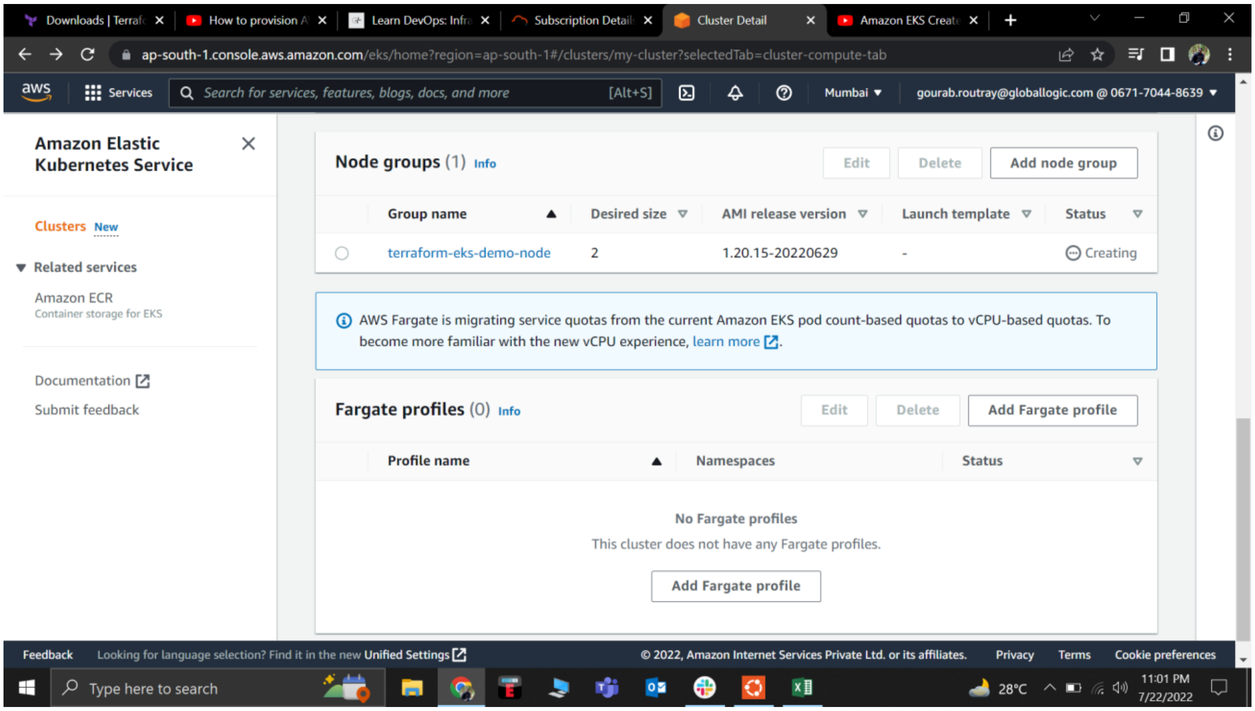
Kubernetes version Info	Status	Provider
1.20	Active	EKS

Nodes (2) Info

Filter Nodes by property or value

Node name	Instance type	Node group	Created	Status
ip-172-31-16-60.ap-south-1.compute.internal	t3.medium	terraform-eks-demo-node	Created 6 minutes ago	Not ready
			Created	

NODE GROUPS


Downloads | Terrafo
How to provision A
Learn DevOps: Infra
Subscription Details
Cluster Detail
Amazon EKS Create
ap-south-1.console.aws.amazon.com/eks/home?region=ap-south-1#/clusters/my-cluster?selectedTab=cluster-compute-tab
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
gourab.routray@globallogic.com @ 0671-7044-8639
Amazon Elastic Kubernetes Service
Clusters New
Related services
Amazon ECR
Container storage for EKS
Documentation
Submit feedback
Edit
Delete
Add node group
Add Fargate profile
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
28°C
11:01 PM
7/22/2022

Node groups (1) Info

	Group name	Desired size	AMI release version	Launch template	Status
○	terraform-eks-demo-node	2	1.20.15-20220629	-	Creating

AWS Fargate is migrating service quotas from the current Amazon EKS pod count-based quotas to vCPU-based quotas. To become more familiar with the new vCPU experience, learn more.

Fargate profiles (0) Info

Profile name	Namespaces	Status

No Fargate profiles

This cluster does not have any Fargate profiles.

Add Fargate profile

ASSIGNMENT-10