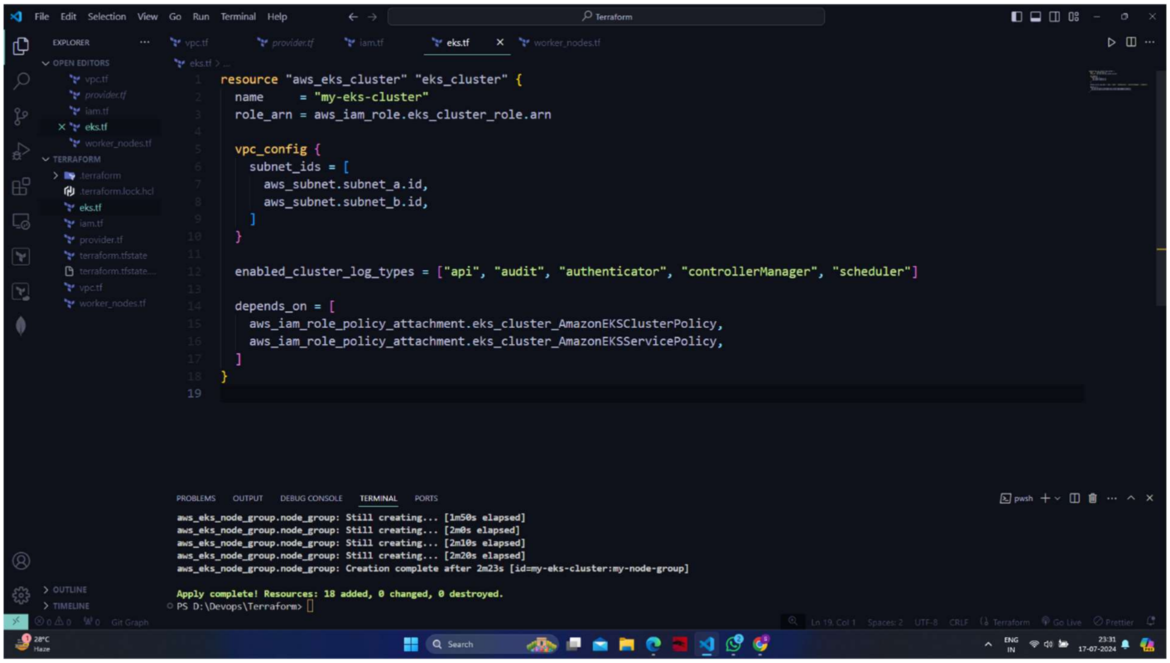
Project-1

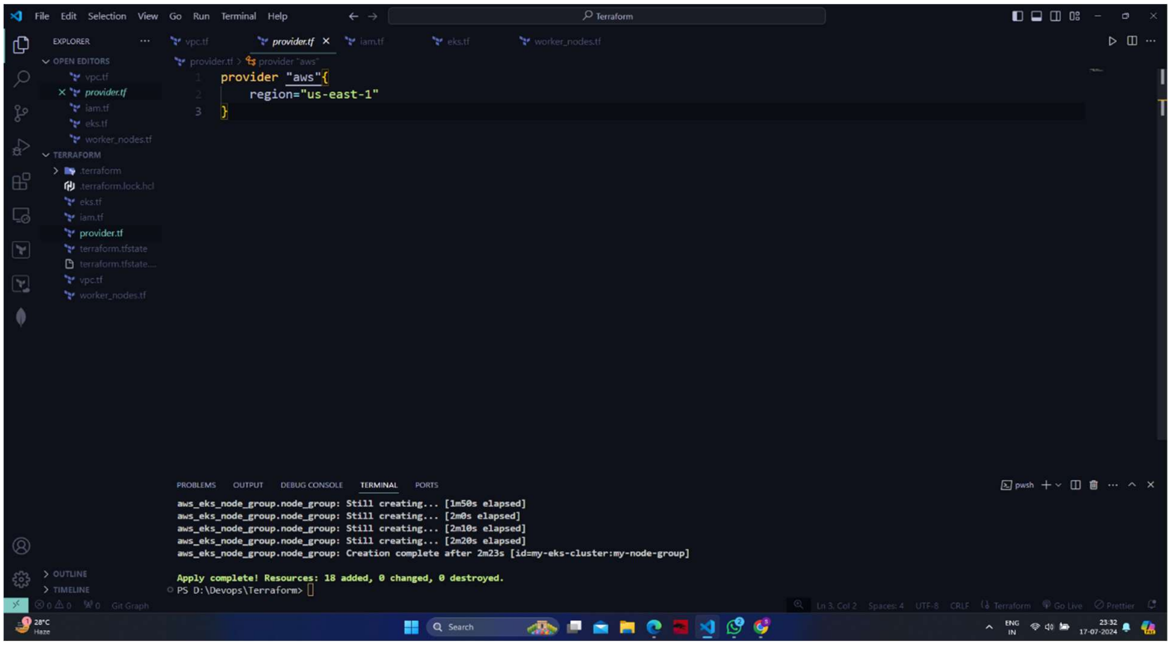
1. Create a eks cluster using terraform and create vpc, iam roles,worker nodes, enabling logging for eks through cloud watch.



```
resource "aws_eks_cluster" "eks_cluster" {
  name     = "my-eks-cluster"
  role_arn = aws_iam_role.eks_cluster_role.arn

  vpc_config {
    subnet_ids = [
      aws_subnet.subnet_a.id,
      aws_subnet.subnet_b.id,
    ]
  }

  enabled_cluster_log_types = ["api", "audit", "authenticator", "controllerManager", "scheduler"]

  depends_on = [
    aws_iam_role_policy_attachment.eks_cluster_AmazonEKSClusterPolicy,
    aws_iam_role_policy_attachment.eks_cluster_AmazonEKSServicePolicy,
  ]
}
```

```
aws_eks_node_group.node_group: Still creating... [1m50s elapsed]
aws_eks_node_group.node_group: Still creating... [2m0s elapsed]
aws_eks_node_group.node_group: Still creating... [2m10s elapsed]
aws_eks_node_group.node_group: Still creating... [2m20s elapsed]
aws_eks_node_group.node_group: Creation complete after 2m23s [id=my-eks-cluster:my-node-group]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
PS D:\Devops\Terraform>
```

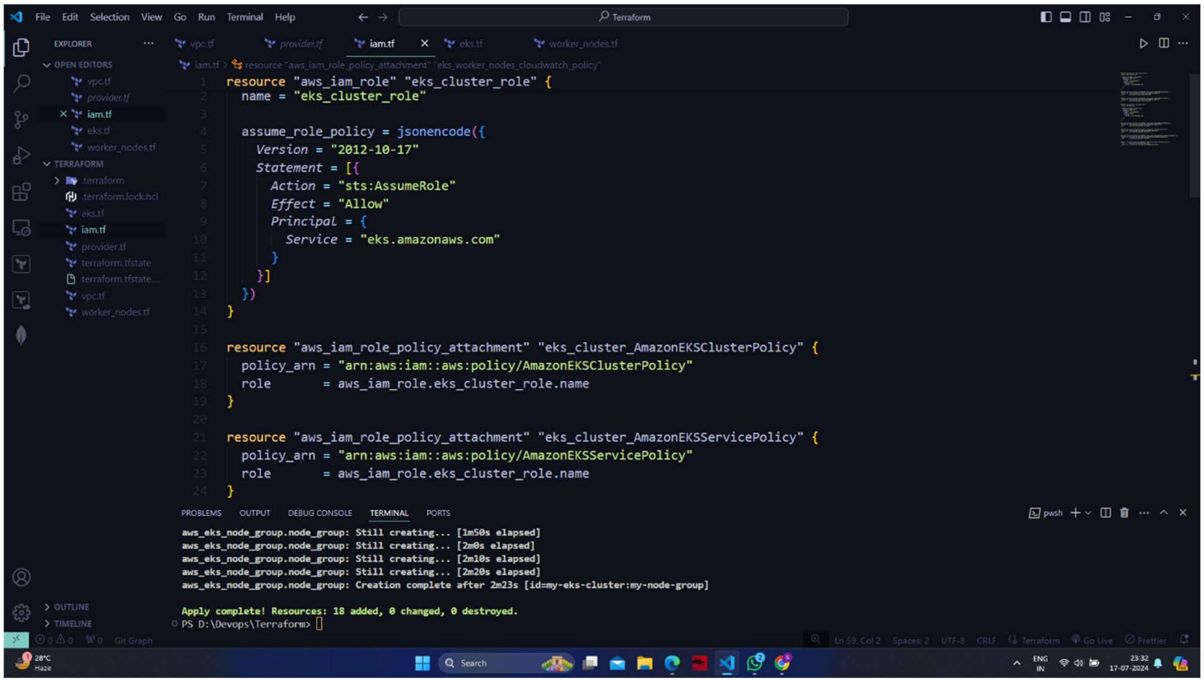
eks.tf (creating eks cluster)



```
provider "aws"{
    region="us-east-1"
}
```

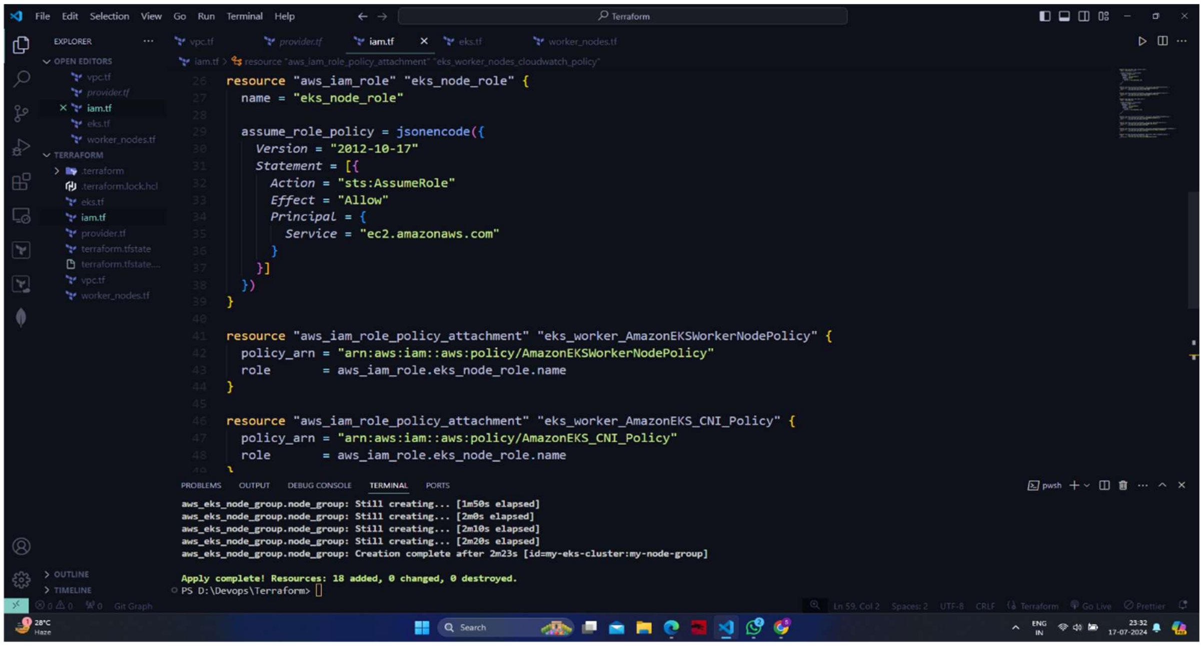
```
aws_eks_node_group.node_group: Still creating... [1m50s elapsed]
aws_eks_node_group.node_group: Still creating... [2m0s elapsed]
aws_eks_node_group.node_group: Still creating... [2m10s elapsed]
aws_eks_node_group.node_group: Still creating... [2m20s elapsed]
aws_eks_node_group.node_group: Creation complete after 2m23s [id=my-eks-cluster:my-node-group]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
PS D:\Devops\Terraform>
```

provider.tf



```
resource "aws_iam_role" "eks_cluster_role" {
  name = "eks_cluster_role"

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

resource "aws_iam_role_policy_attachment" "eks_cluster_AmazonEKSClusterPolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
  role       = aws_iam_role.eks_cluster_role.name
}

resource "aws_iam_role_policy_attachment" "eks_cluster_AmazonEKSServicePolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSServicePolicy"
  role       = aws_iam_role.eks_cluster_role.name
}
```

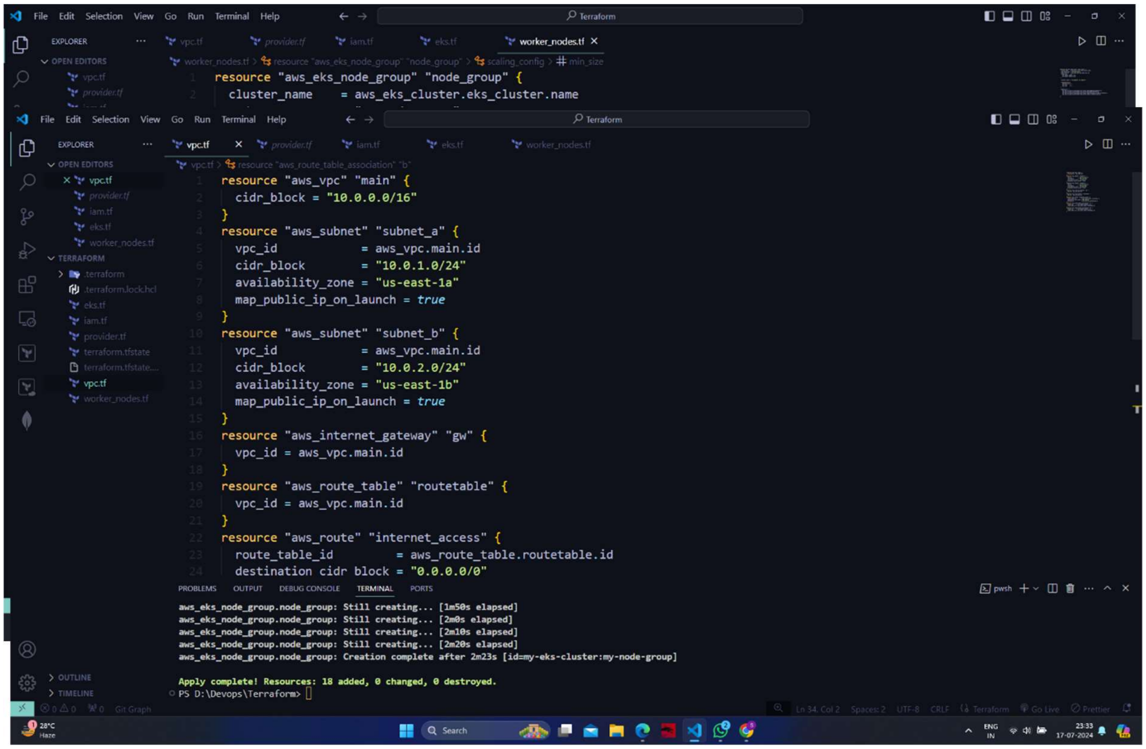
```
aws_eks_node_group.node_group: Still creating... [1m50s elapsed]
aws_eks_node_group.node_group: Still creating... [2m0s elapsed]
aws_eks_node_group.node_group: Still creating... [2m10s elapsed]
aws_eks_node_group.node_group: Still creating... [2m20s elapsed]
aws_eks_node_group.node_group: Creation complete after 2m23s [id=my-eks-cluster:my-node-group]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
PS D:\Devops\Terraform>
```



```
resource "aws_iam_role" "eks_node_role" {
  name = "eks_node_role"

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

resource "aws_iam_role_policy_attachment" "eks_worker_AmazonEKSWorkerNodePolicy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
  role       = aws_iam_role.eks_node_role.name
}

resource "aws_iam_role_policy_attachment" "eks_worker_AmazonEKS_CNI_Policy" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
  role       = aws_iam_role.eks_node_role.name
}
```

```
aws_eks_node_group.node_group: Still creating... [1m50s elapsed]
aws_eks_node_group.node_group: Still creating... [2m0s elapsed]
aws_eks_node_group.node_group: Still creating... [2m10s elapsed]
aws_eks_node_group.node_group: Still creating... [2m20s elapsed]
aws_eks_node_group.node_group: Creation complete after 2m23s [id=my-eks-cluster:my-node-group]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
PS D:\Devops\Terraform>
```

iam.tf

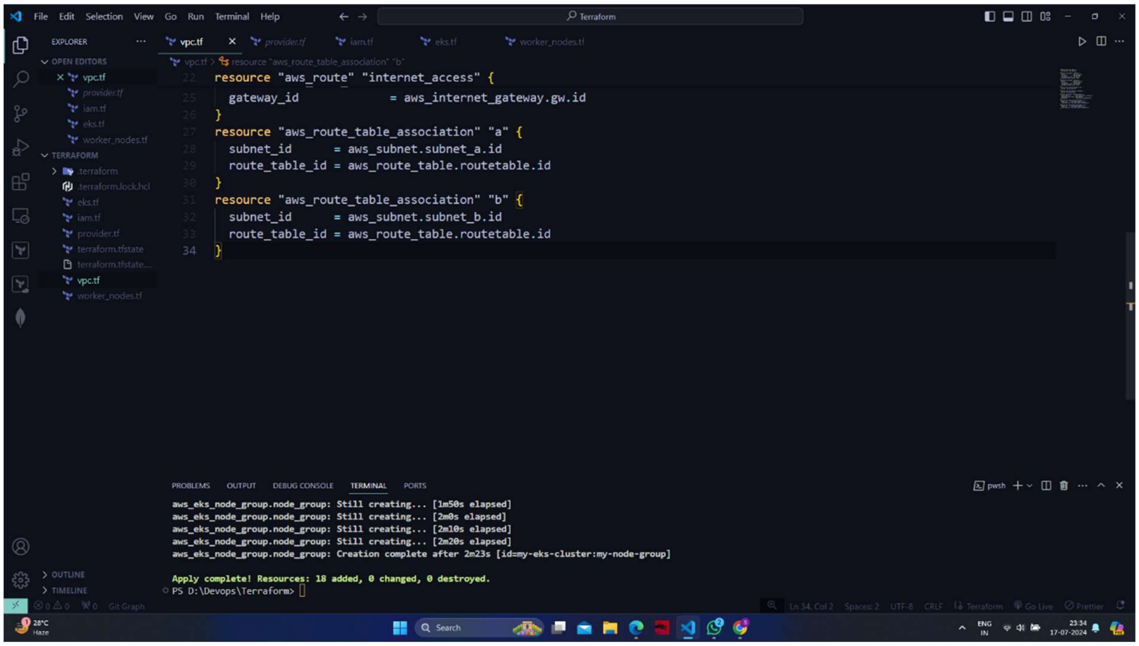


```
resource "aws_eks_node_group" "node_group" {
  cluster_name    = aws_eks_cluster.eks_cluster.name
```

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}
resource "aws_subnet" "subnet_a" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-1a"
  map_public_ip_on_launch = true
}
resource "aws_subnet" "subnet_b" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "us-east-1b"
  map_public_ip_on_launch = true
}
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id
}
resource "aws_route_table" "routetable" {
  vpc_id = aws_vpc.main.id
}
resource "aws_route" "internet_access" {
  route_table_id         = aws_route_table.routetable.id
  destination cidr block = "0.0.0.0/0"
```

```
aws_eks_node_group.node_group: Still creating... [1m50s elapsed]
aws_eks_node_group.node_group: Still creating... [2m0s elapsed]
aws_eks_node_group.node_group: Still creating... [2m10s elapsed]
aws_eks_node_group.node_group: Still creating... [2m20s elapsed]
aws_eks_node_group.node_group: Creation complete after 2m23s [id=my-eks-cluster:my-node-group]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
PS D:\Devops\Terraform> 
```

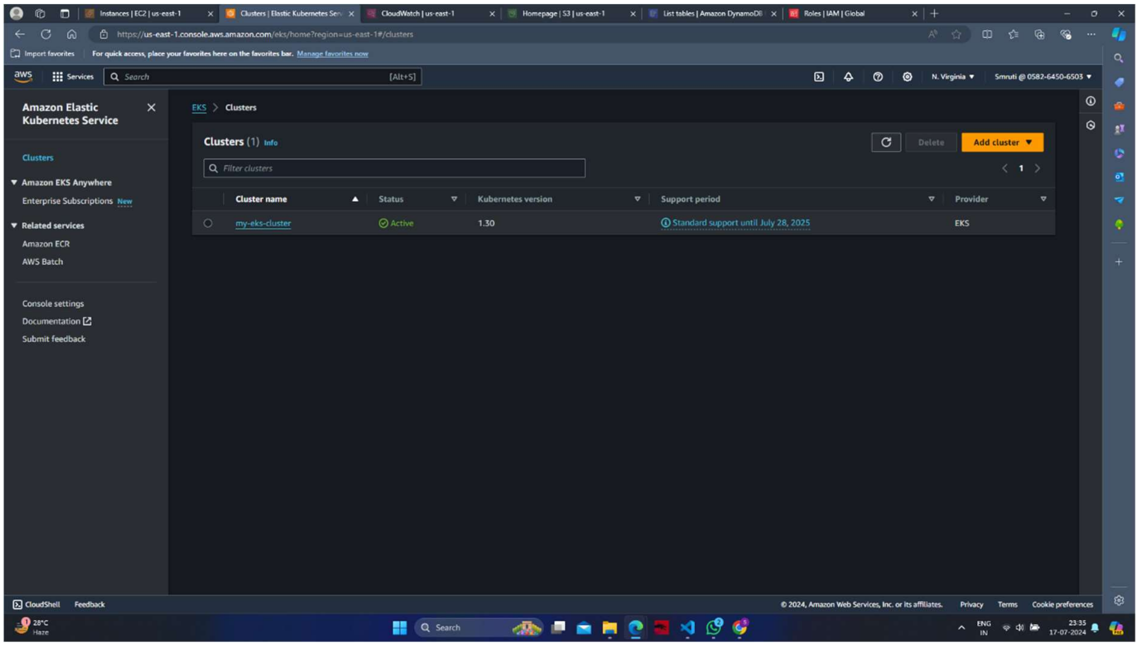
worker_nodes.tf



```
resource "aws_route" "internet_access" {
  gateway_id             = aws_internet_gateway.gw.id
}
resource "aws_route_table_association" "a" {
  subnet_id      = aws_subnet.subnet_a.id
  route_table_id = aws_route_table.routetable.id
}
resource "aws_route_table_association" "b" {
  subnet_id      = aws_subnet.subnet_b.id
  route_table_id = aws_route_table.routetable.id
}
```

```
aws_eks_node_group.node_group: Still creating... [1m50s elapsed]
aws_eks_node_group.node_group: Still creating... [2m0s elapsed]
aws_eks_node_group.node_group: Still creating... [2m10s elapsed]
aws_eks_node_group.node_group: Still creating... [2m20s elapsed]
aws_eks_node_group.node_group: Creation complete after 2m23s [id=my-eks-cluster:my-node-group]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
PS D:\Devops\Terraform> 
```

vpc.tf

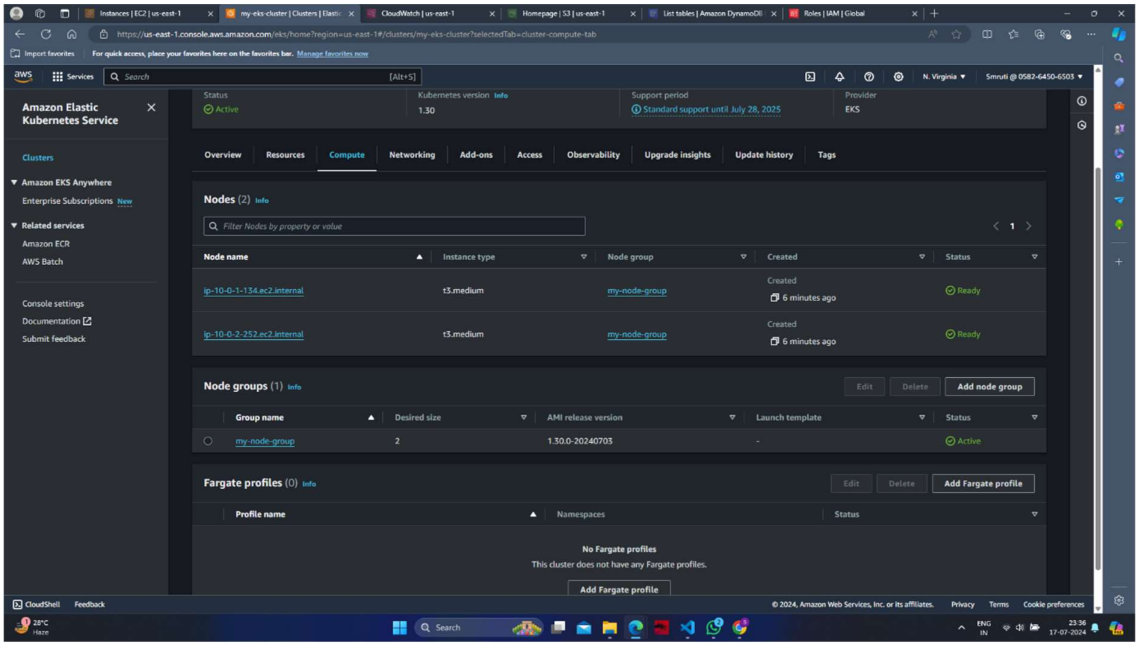


Instances | EC2 | us-east-1
Clusters | Elastic Kubernetes Serv
CloudWatch | us-east-1
Homepage | S3 | us-east-1
List tables | Amazon DynamoDB
Roles | IAM | Global
https://us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#/clusters
Import favorites
For quick access, place your favorites here on the favorites bar. Manage favorites now
Services
Search
[Alt+S]
N. Virginia
Smruti @ 0582-6450-6503
Amazon Elastic Kubernetes Service
Clusters
Amazon EKS Anywhere
Enterprise Subscriptions New
Related services
Amazon ECR
AWS Batch
Console settings
Documentation
Submit feedback
EKS
Clusters
Delete
Add cluster
Filter clusters
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Haze
Search
ENG
IN
23:35
17-07-2024

Clusters (1) Info

	Cluster name	Status	Kubernetes version	Support period	Provider
○	my-eks-cluster	Active	1.30	Standard support until July 28, 2025	EKS

Cluster created



Instances | EC2 | us-east-1
my-eks-cluster | Clusters | Elastic
CloudWatch | us-east-1
Homepage | S3 | us-east-1
List tables | Amazon DynamoDB
Roles | IAM | Global
https://us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#/clusters/my-eks-cluster?selectedTab=cluster-compute-tab
Import favorites
For quick access, place your favorites here on the favorites bar. Manage favorites now
Services
Search
[Alt+S]
N. Virginia
Smruti @ 0582-6450-6503
Amazon Elastic Kubernetes Service
Clusters
Amazon EKS Anywhere
Enterprise Subscriptions New
Related services
Amazon ECR
AWS Batch
Console settings
Documentation
Submit feedback
Overview
Resources
Compute
Networking
Add-ons
Access
Observability
Upgrade insights
Update history
Tags
Filter Nodes by property or value
Edit
Delete
Add node group
Add Fargate profile
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Haze
Search
ENG
IN
23:36
17-07-2024

Status: Active
Kubernetes version Info: 1.30
Support period: Standard support until July 28, 2025
Provider: EKS

Nodes (2) Info

Node name	Instance type	Node group	Created	Status
ip-10-0-1-134.ec2.internal	t3.medium	my-node-group	Created 6 minutes ago	Ready
ip-10-0-2-252.ec2.internal	t3.medium	my-node-group	Created 6 minutes ago	Ready

Node groups (1) Info

	Group name	Desired size	AMI release version	Launch template	Status
○	my-node-group	2	1.30.0-20240703	-	Active

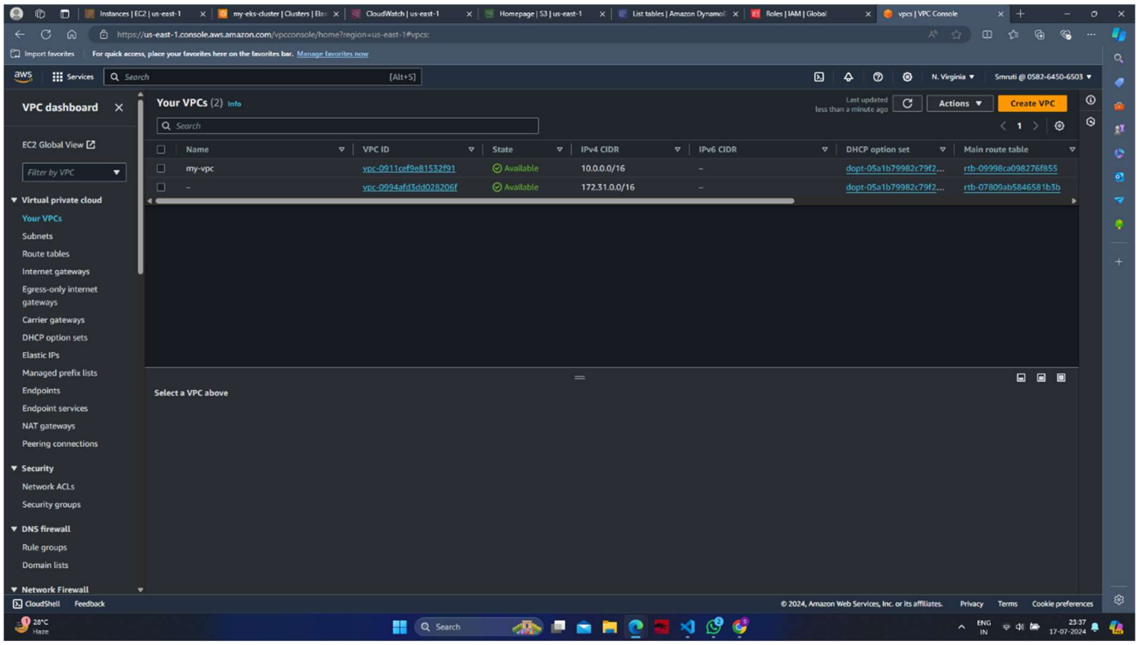
Fargate profiles (0) Info

	Profile name	Namespaces	Status

No Fargate profiles

This cluster does not have any Fargate profiles.

Node group and worker nodes



https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:

Your VPCs (2) Info

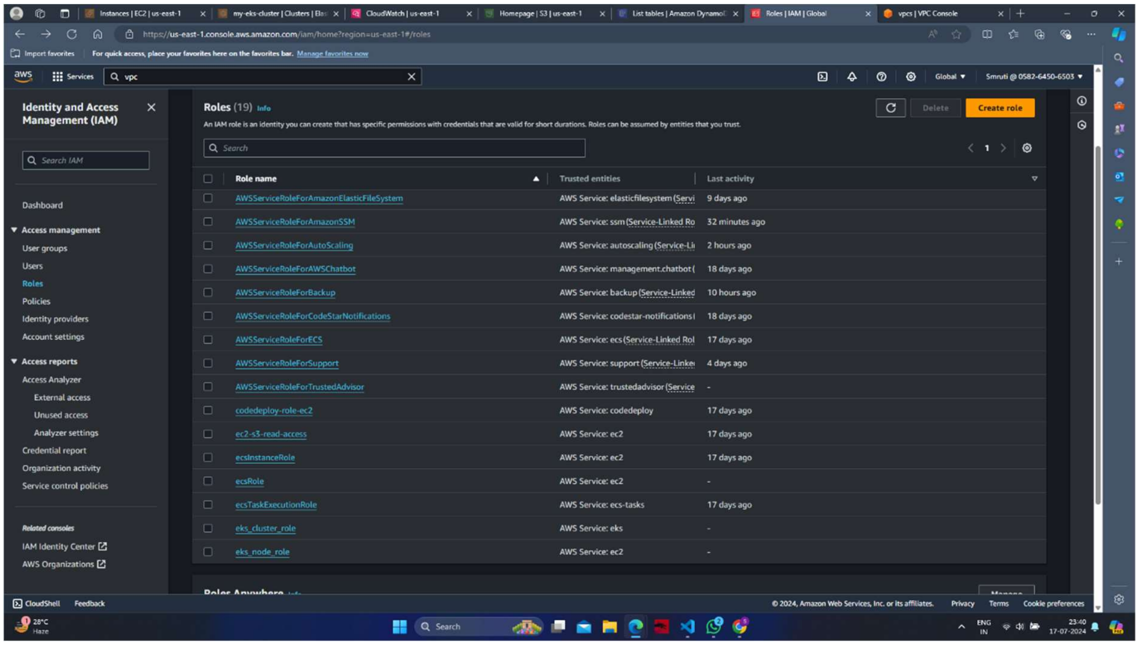
Last updated less than a minute ago

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
my-vpc	vpc-0911cef9e81532f91	Available	10.0.0.0/16	–	dopt-05a1b79982c79f2...	rtb-09998ca098276f855
–	vpc-0994afd3dd028206f	Available	172.31.0.0/16	–	dopt-05a1b79982c79f2...	rtb-07809ab5846581b3b

Select a VPC above

Vpc created

Necessary IAM Roles

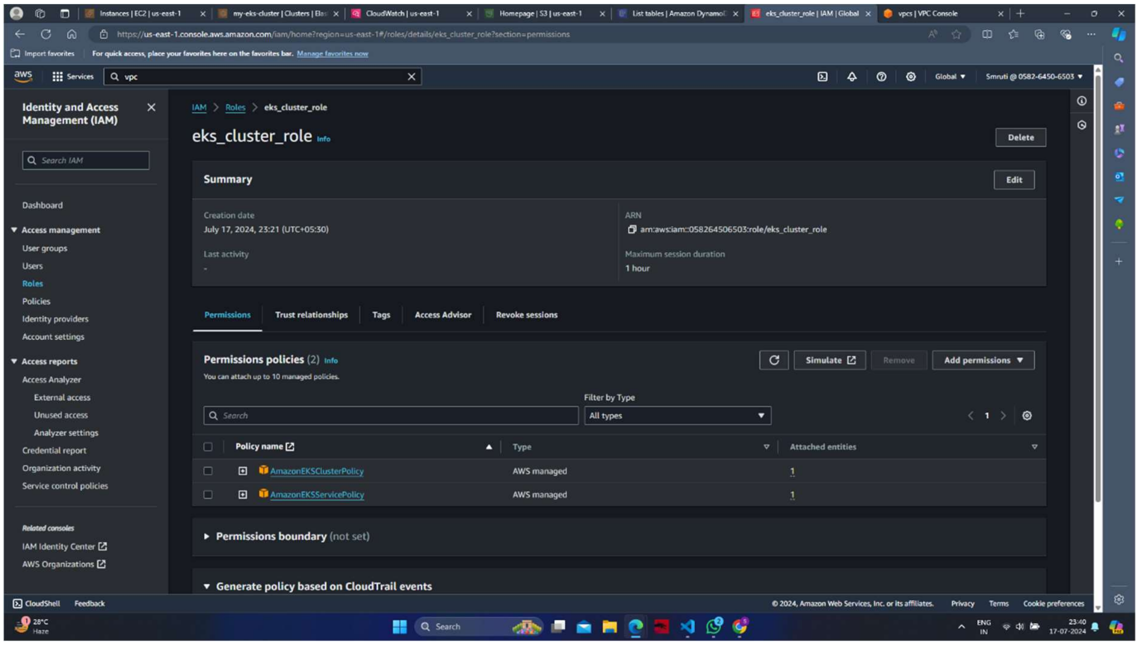


https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles

Roles (19) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Servi	9 days ago
AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Ro	32 minutes ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Li	2 hours ago
AWSServiceRoleForAWSChatbot	AWS Service: management.chatbot (	18 days ago
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked	10 hours ago
AWSServiceRoleForCodeStarNotifications	AWS Service: codestar-notifications (	18 days ago
AWSServiceRoleForECS	AWS Service: ecs (Service-Linked Rol	17 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	4 days ago
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
codedeploy-role-ec2	AWS Service: codedeploy	17 days ago
ec2-s3-read-access	AWS Service: ec2	17 days ago
ecsInstanceRole	AWS Service: ec2	17 days ago
ecsRole	AWS Service: ec2	-
ecsTaskExecutionRole	AWS Service: ecs-tasks	17 days ago
eks_cluster_role	AWS Service: eks	-
eks_node_role	AWS Service: ec2	-



https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/details/eks_cluster_role?section=permissions

IAM > Roles > eks_cluster_role

eks_cluster_role

Summary

Creation date
July 17, 2024, 23:21 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::058264506503:role/eks_cluster_role

Maximum session duration
1 hour

Permissions policies (2)

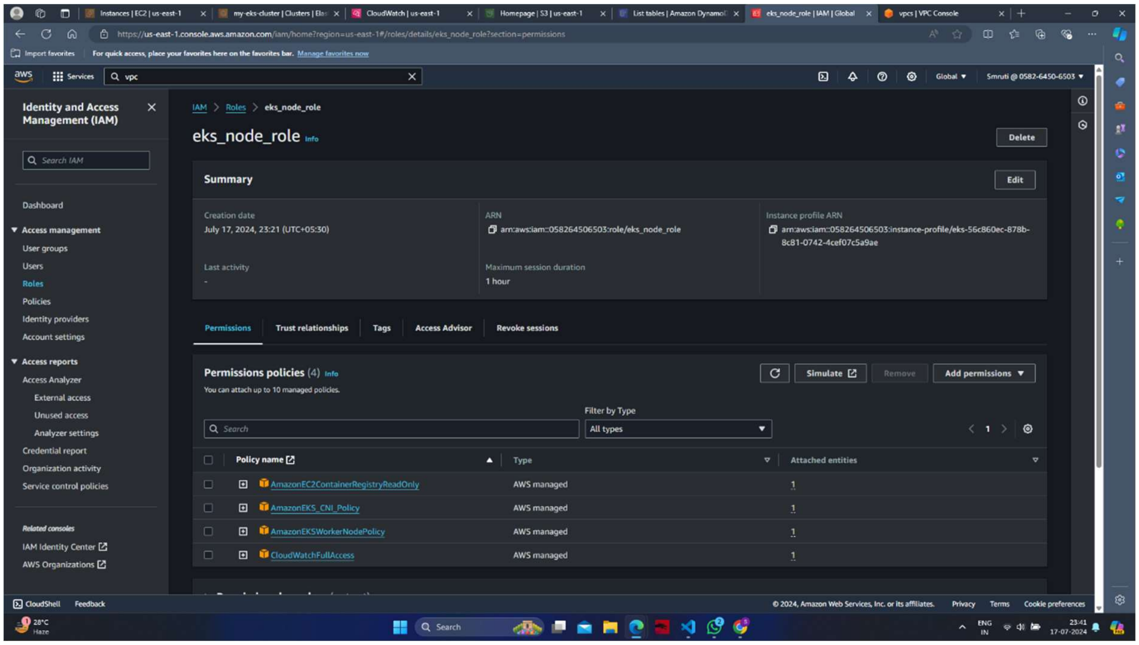
You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonEKSClusterPolicy	AWS managed	1
AmazonEKSServicePolicy	AWS managed	1

Permissions boundary (not set)

Generate policy based on CloudTrail events

eks_cluster_role



https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/details/eks_node_role?section=permissions

IAM > Roles > eks_node_role

eks_node_role

Summary

Creation date
July 17, 2024, 23:21 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::058264506503:role/eks_node_role

Maximum session duration
1 hour

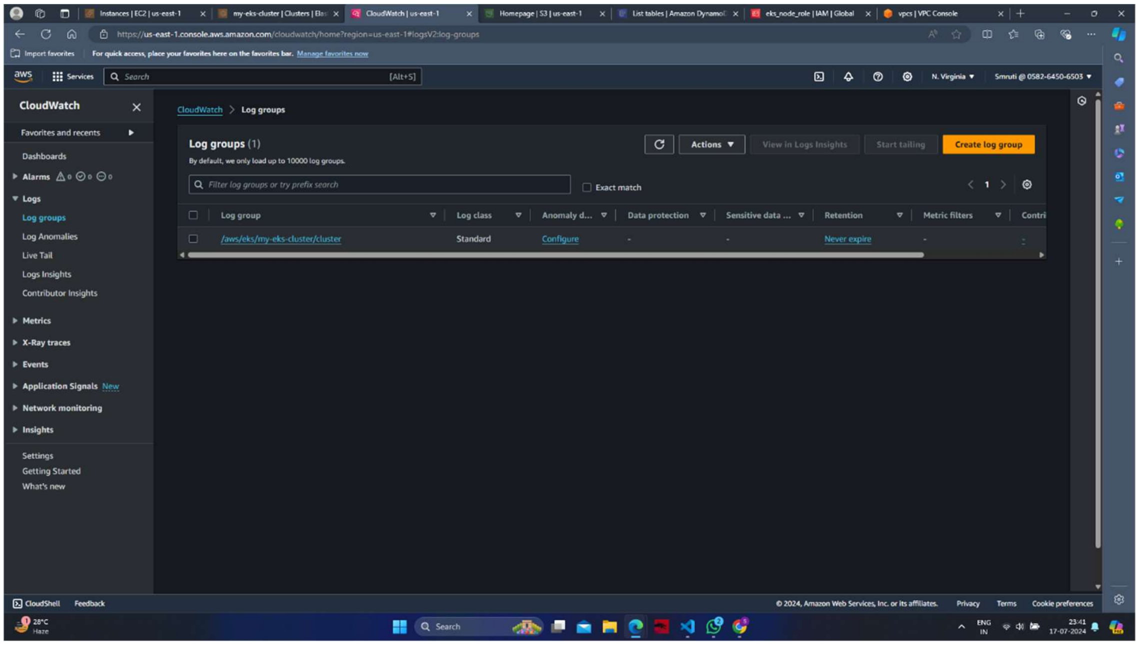
Instance profile ARN
arn:aws:iam::058264506503:instance-profile/eks-56c860ec-878b-8c81-0742-4cef07c5a9ae

Permissions policies (4)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryReadOnly	AWS managed	1
AmazonEKS_CNI_Policy	AWS managed	1
AmazonEKSWorkerNodePolicy	AWS managed	1
CloudWatchFullAccess	AWS managed	1

eks_node_role with CloudWatchFullAccess



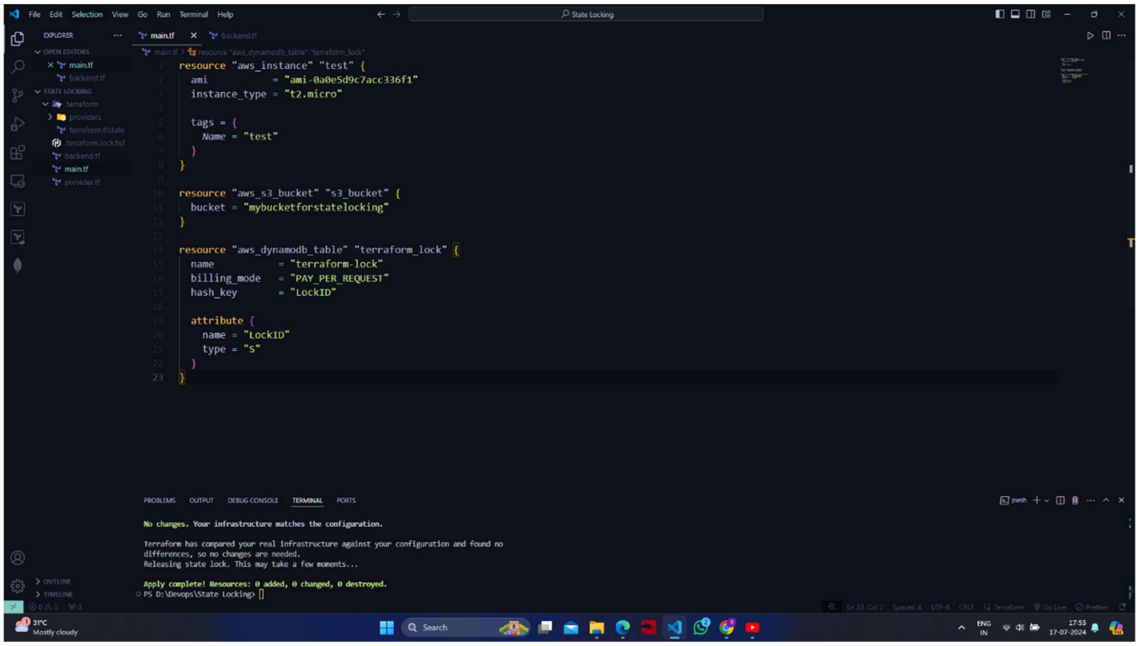
Log groups (1)

By default, we only load up to 10000 log groups.

	Log group	Log class	Anomaly d...	Data protection	Sensitive data ...	Retention	Metric filters	Contri
□	/aws/eks/my-eks-cluster/cluster	Standard	Configure	-	-	Never expire	-	-

Eks logs in CloudWatch

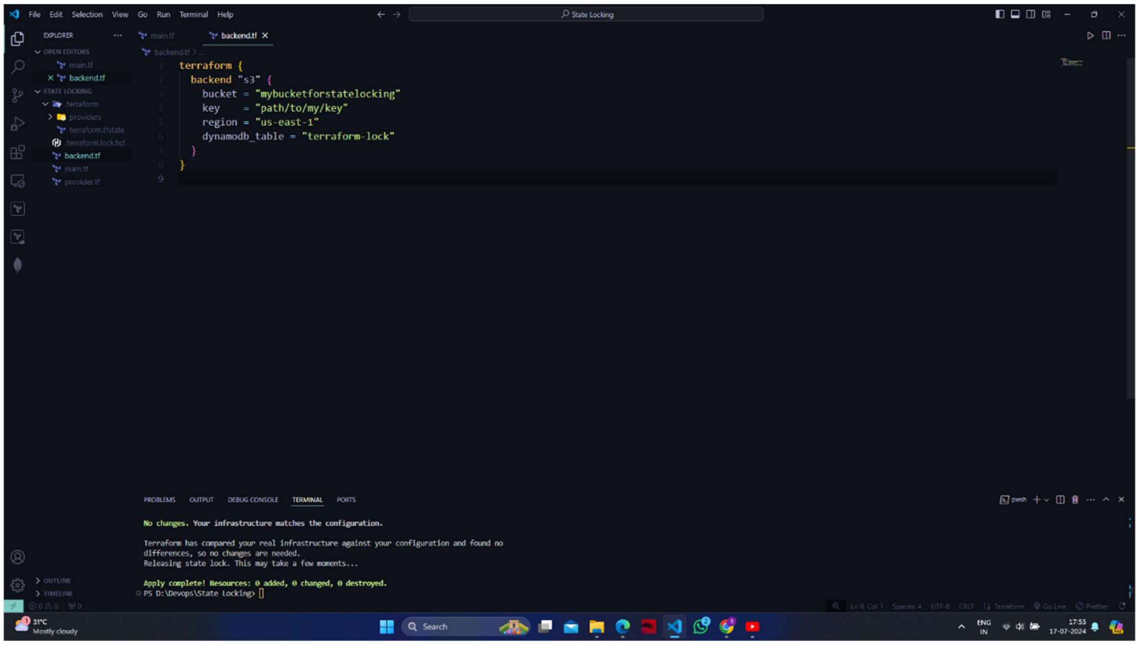
2.Enable state locking using terraform through s3 and DynamoDB.



```
resource "aws_instance" "test" {
  ami              = "ami-0a0e5d9c7acc336f1"
  instance_type = "t2.micro"

  tags = {
    Name = "test"
  }
}

resource "aws_s3_bucket" "s3_bucket" {
  bucket = "mybucketforstatelocking"
}

resource "aws_dynamodb_table" "terraform_lock" {
  name           = "terraform-lock"
  billing_mode   = "PAY_PER_REQUEST"
  hash_key       = "LockID"

  attribute {
    name = "LockID"
    type = "S"
  }
}
```

```
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no
differences, so no changes are needed.
Releasing state lock. This may take a few moments...

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS D:\Devops\State Locking> 
```

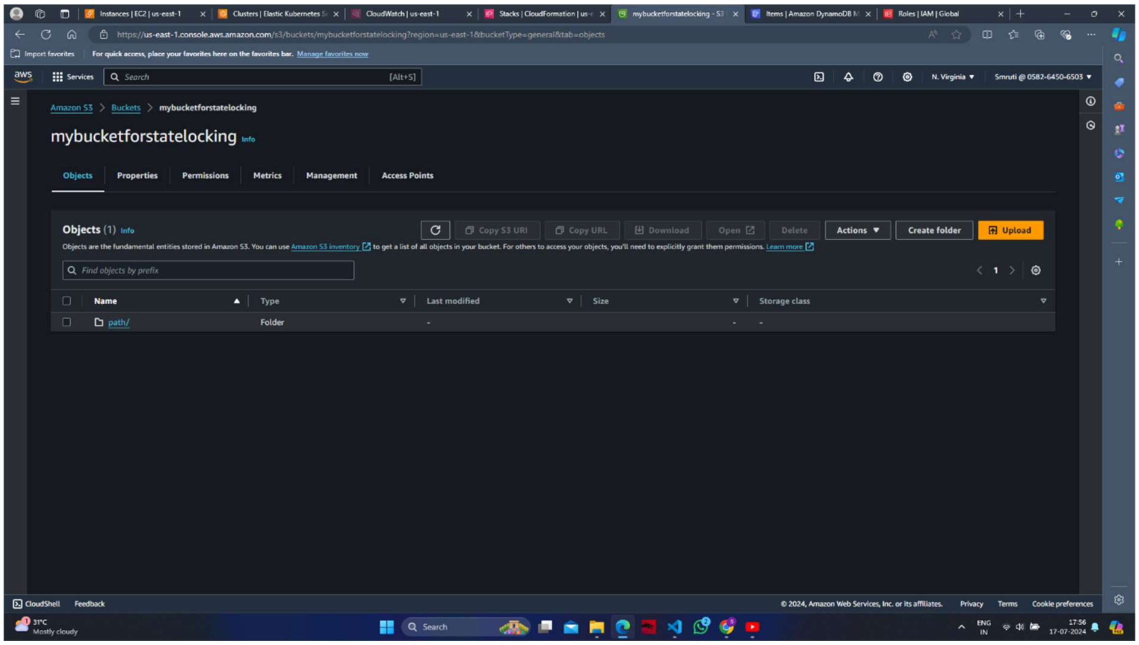
main.tf



```
terraform {
  backend "s3" {
    bucket = "mybucketforstatelocking"
    key    = "path/to/my/key"
    region = "us-east-1"
    dynamodb_table = "terraform-lock"
  }
}
```

```
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no
differences, so no changes are needed.
Releasing state lock. This may take a few moments...

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS D:\Devops\State Locking> 
```

backend.tf



https://us-east-1.console.aws.amazon.com/s3/buckets/mybucketforstatelocking?region=us-east-1&bucketType=general&tab=objects

Amazon S3 > Buckets > mybucketforstatelocking

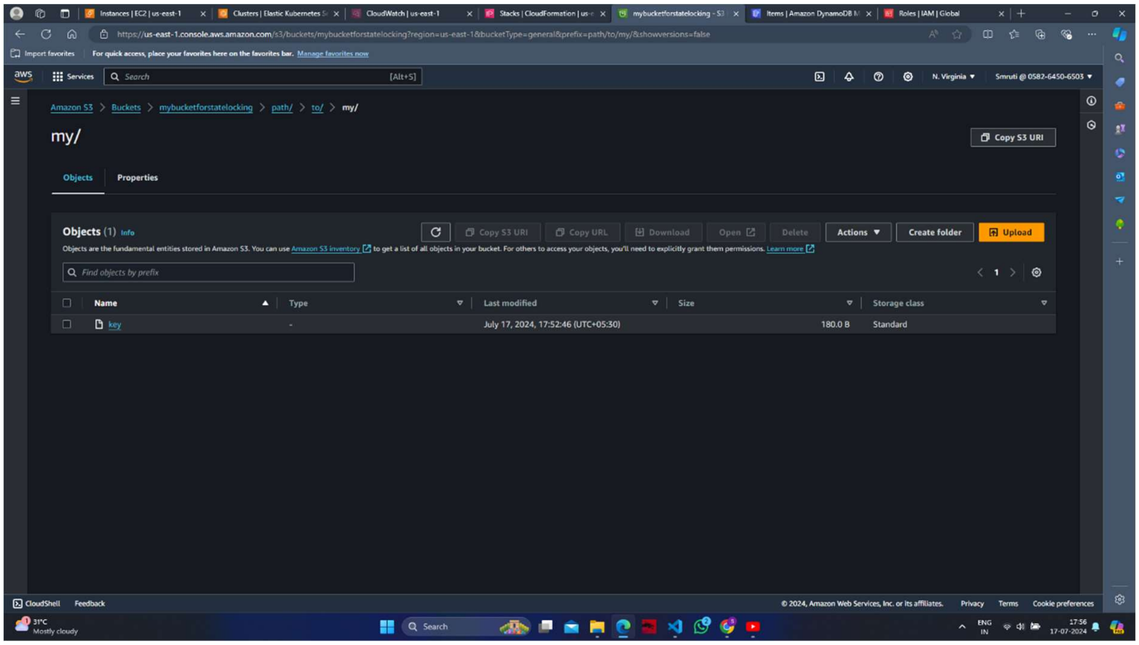
mybucketforstatelocking Info

Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
path/	Folder	-	-	-

Created s3 bucket



https://us-east-1.console.aws.amazon.com/s3/buckets/mybucketforstatelocking?region=us-east-1&bucketType=general&prefix=path/to/my/&showversions=false

Amazon S3 > Buckets > mybucketforstatelocking > path/ > to/ > my/

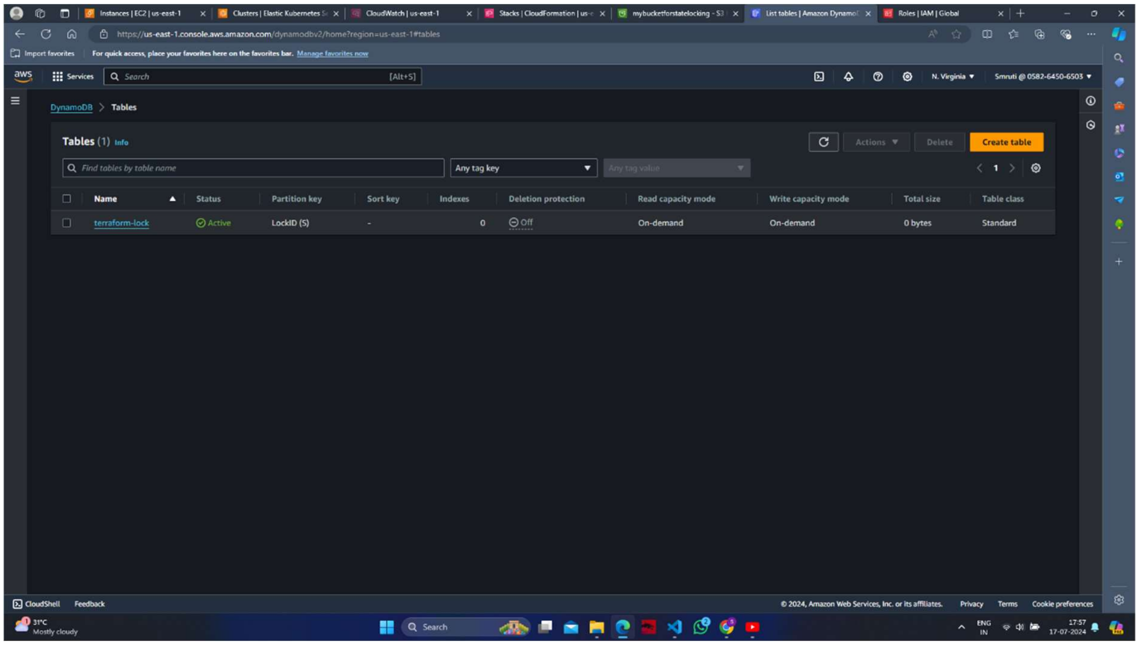
my/

Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
key	-	July 17, 2024, 17:52:46 (UTC+05:30)	180.0 B	Standard

State file inside s3 bucket



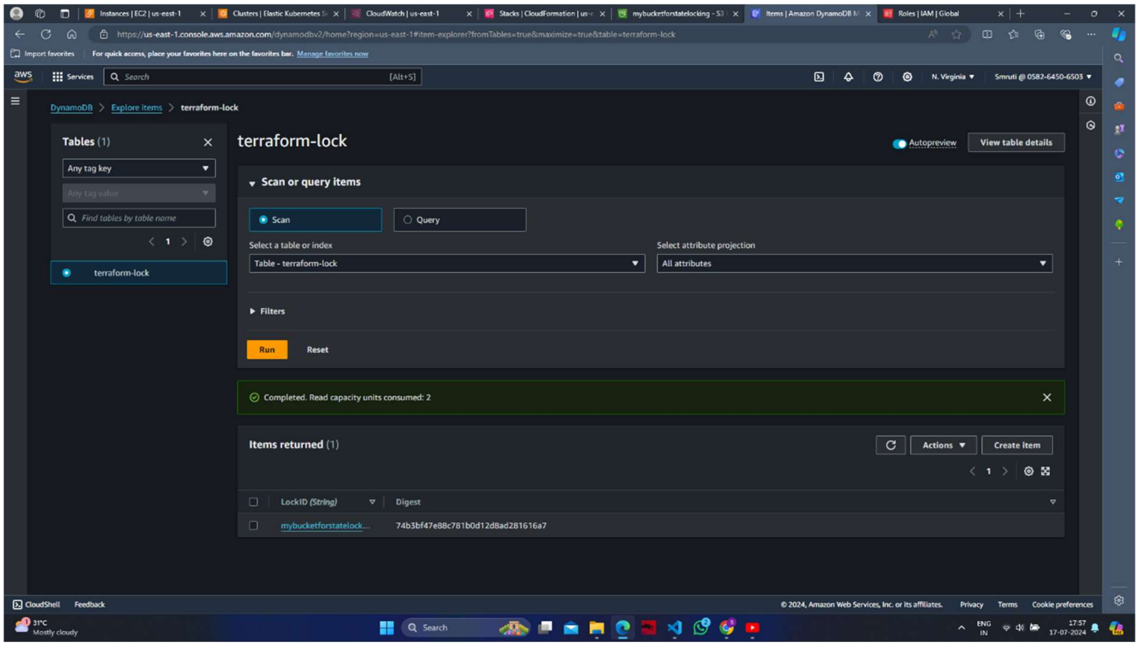
https://us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#tables

DynamoDB > Tables

Tables (1) Info

Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode	Total size	Table class
terraform-lock	Active	LockID (S)	-	0	Off	On-demand	On-demand	0 bytes	Standard

Table created in DynamoDB



terraform-lock

Scan or query items

● Scan ○ Query

Select a table or index: Table - terraform-lock

Select attribute projection: All attributes

Filters

Completed. Read capacity units consumed: 2

Items returned (1)

LockID (String)	Digest
mybucketforstatelock...	74b3bf47e88c781b0d12d8ad281616a7

State Locking(Lock Id)