**CLOUD FORMATION**

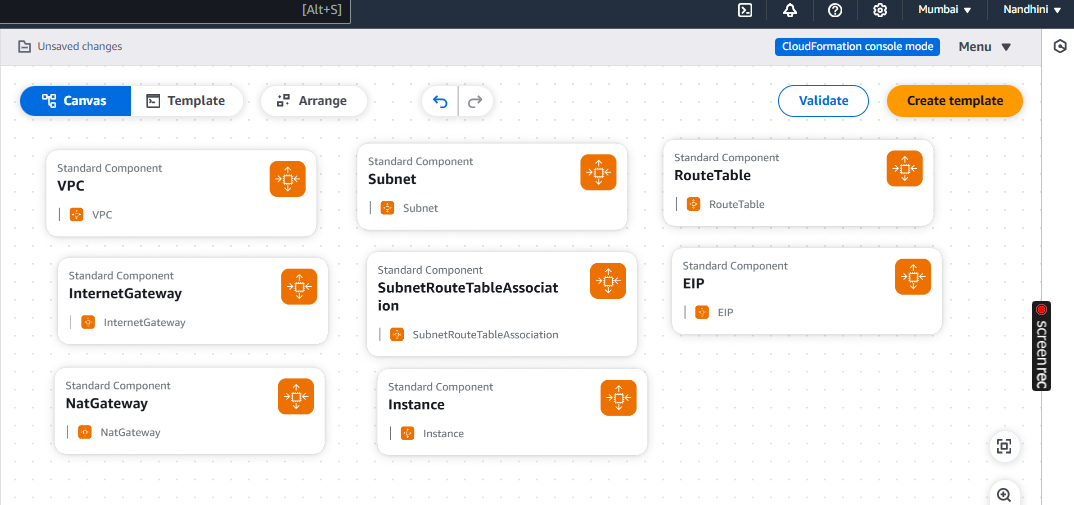
**To build infrastructure: VPC, Public and Private Subnet, Route table, IGW, Nat Gateway, EIP, and EC2 Instance.**

* AWS Cloud Formation is an Amazon Web Services (AWS) service that allows users to build, set up, and manage resources in their cloud infrastructure through code.
* With Cloud Formation, you can create and provision AWS infrastructure resources using templates, and text files written in JSON or YAML format.

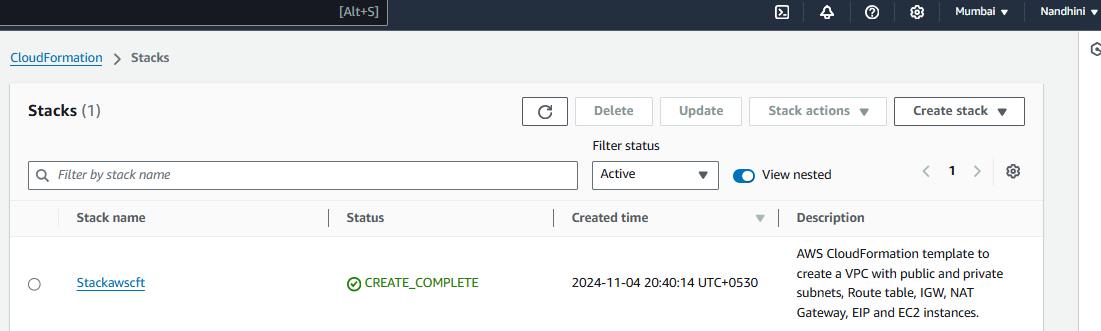
**Step 1: Create stack**

**Stack**: A stack is a collection of AWS resources that can manage as a single unit. If you delete a stack, all the resources within it are also deleted, which helps with easy teardown and clean up.

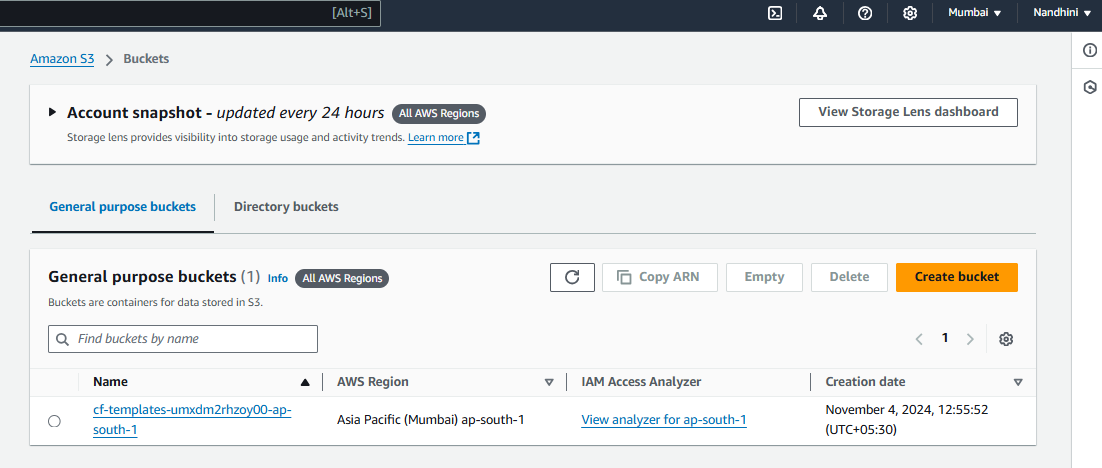
**Canvas of infrastructure:**

****

**Stack Name: Stackawscft**

****

**CF-Template stored in S3 bucket:**

****

***TEMPLATE used to build infrastructures:***

***YAML CODE:***

*AWSTemplateFormatVersion: '2010-09-09'*

*Description: AWS CloudFormation template to create a VPC with public and*

*private subnets, Route table, IGW, NAT Gateway, EIP and EC2 instances.*

*Resources:*

*# VPC*

*CloudVPC:*

*Type: 'AWS::EC2::VPC'*

*Properties:*

*CidrBlock: '10.0.0.0/24'*

*EnableDnsSupport: true*

*EnableDnsHostnames: true*

*Tags:*

*- Key: Name*

*Value: CloudVPC*

*#InternetGateway*

*CloudIGW:*

*Type: 'AWS::EC2::InternetGateway'*

*Properties:*

*Tags:*

*- Key: Name*

*Value: CloudIGW*

*AttachGateway:*

*Type: 'AWS::EC2::VPCGatewayAttachment'*

*Properties:*

*VpcId: !Ref CloudVPC*

*InternetGatewayId: !Ref CloudIGW*

*#Public Subnet*

*PubSN:*

*Type: 'AWS::EC2::Subnet'*

*Properties:*

*VpcId: !Ref CloudVPC*

*CidrBlock: '10.0.0.0/28'*

*MapPublicIpOnLaunch: true*

*AvailabilityZone: !Select [0, !GetAZs '']*

*Tags:*

*- Key: Name*

*Value: PubSN*

*#Private Subnet*

*PriSN:*

*Type: 'AWS::EC2::Subnet'*

*Properties:*

*VpcId: !Ref CloudVPC*

*CidrBlock: '10.0.0.16/28'*

*MapPublicIpOnLaunch: false*

*AvailabilityZone: !Select [1, !GetAZs '']*

*Tags:*

*- Key: Name*

*Value: PriSN*

*# Public RouteTable*

*PubRT:*

*Type: AWS::EC2::RouteTable*

*Properties:*

*VpcId: !Ref CloudVPC*

*Tags:*

*- Key: Name*

*Value: PubRT*

*# Route InternetGateway in Public Route Table*

*PublicRouteTable:*

*Type: AWS::EC2::Route*

*Properties:*

*RouteTableId: !Ref PubRT*

*DestinationCidrBlock: 0.0.0.0/0*

*GatewayId: !Ref CloudIGW*

*# Associate Public Route Table with Public Subnet*

*PublicSubnetRouteTableAssociation:*

*Type: AWS::EC2::SubnetRouteTableAssociation*

*Properties:*

*SubnetId: !Ref PubSN*

*RouteTableId: !Ref PubRT*

*# Private RouteTable*

*PriRT:*

*Type: AWS::EC2::RouteTable*

*Properties:*

*VpcId: !Ref CloudVPC*

*Tags:*

*- Key: Name*

*Value: PriRT*

*# NAT Gateway Elastic IP*

*EIP:*

*Type: AWS::EC2::EIP*

*Properties: {}*

*# NAT Gateway*

*CloudNAT:*

*Type: AWS::EC2::NatGateway*

*Properties:*

*AllocationId: !GetAtt EIP.AllocationId*

*SubnetId: !Ref PubSN*

*Tags:*

*- Key: Name*

*Value: CloudNAT*

*# Route NAT Gateway in Private Route Table*

*PrivateRoute:*

*Type: AWS::EC2::Route*

*Properties:*

*RouteTableId: !Ref PriRT*

*DestinationCidrBlock: 0.0.0.0/0*

*NatGatewayId: !Ref CloudNAT*

*# Associate Private Route Table with Private Subnet*

*PrivateSubnetRouteTableAssociation:*

*Type: AWS::EC2::SubnetRouteTableAssociation*

*Properties:*

*SubnetId: !Ref PriSN*

*RouteTableId: !Ref PriRT*

*# Security Group for Public EC2*

*PubSG:*

*Type: AWS::EC2::SecurityGroup*

*Properties:*

*GroupDescription: Allow SSH and HTTP traffic to the public instance*

*VpcId: !Ref CloudVPC*

*SecurityGroupIngress:*

*- IpProtocol: tcp*

*FromPort: 22*

*ToPort: 22*

*CidrIp: 0.0.0.0/0 # SSH access*

*- IpProtocol: tcp*

*FromPort: 80*

*ToPort: 80*

*CidrIp: 0.0.0.0/0 # HTTP access*

*Tags:*

*- Key: Name*

*Value: PubSG*

*# Security Group for Private EC2*

*PriSG:*

*Type: AWS::EC2::SecurityGroup*

*Properties:*

*GroupDescription: Allow SSH access to the private instance from public instance*

*VpcId: !Ref CloudVPC*

*SecurityGroupIngress:*

*- IpProtocol: tcp*

*FromPort: 22*

*ToPort: 22*

*SourceSecurityGroupId: !Ref PubSG*

*Tags:*

*- Key: Name*

*Value: PriSG*

*# EC2 Instance in Public Subnet*

*CloudInstance1:*

*Type: AWS::EC2::Instance*

*Properties:*

*InstanceType: t2.micro*

*KeyName: Sep*

*ImageId: ami-0e0e417dfa2028266*

*SubnetId: !Ref PubSN*

*SecurityGroupIds:*

*- !Ref PubSG*

*Tags:*

*- Key: Name*

*Value: CloudInstance1*

*# EC2 Instance in Private Subnet*

*CloudInstance2:*

*Type: AWS::EC2::Instance*

*Properties:*

*InstanceType: t2.micro*

*KeyName: Sep*

*ImageId: ami-0e0e417dfa2028266*

*SubnetId: !Ref PriSN*

*SecurityGroupIds:*

*- !Ref PriSG*

*Tags:*

*- Key: Name*

*Value: CloudInstance2*

*Outputs:*

*VPCId:*

*Description: VPC ID*

*Value: !Ref CloudVPC*

*PublicSubnet1Id:*

*Description: Public Subnet ID*

*Value: !Ref PubSN*

*PrivateSubnet1Id:*

*Description: Private Subnet ID*

*Value: !Ref PriSN*

*InternetGatewayId:*

*Description: Internet Gateway ID*

*Value: !Ref CloudIGW*

*NatGatewayId:*

*Description: NAT Gateway ID*

*Value: !Ref CloudNAT*

*PublicEC2InstanceId:*

*Description: Public EC2 Instance ID*

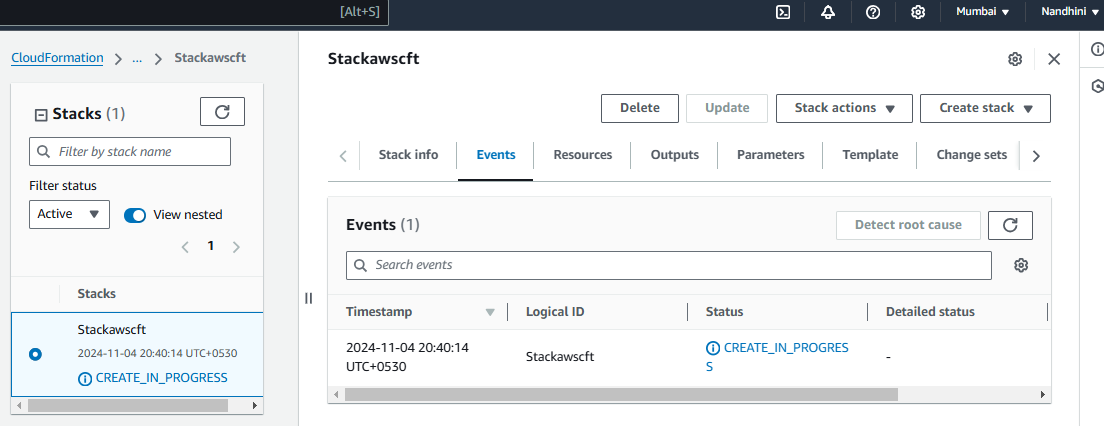
*Value: !Ref CloudInstance1*

*PrivateEC2InstanceId:*

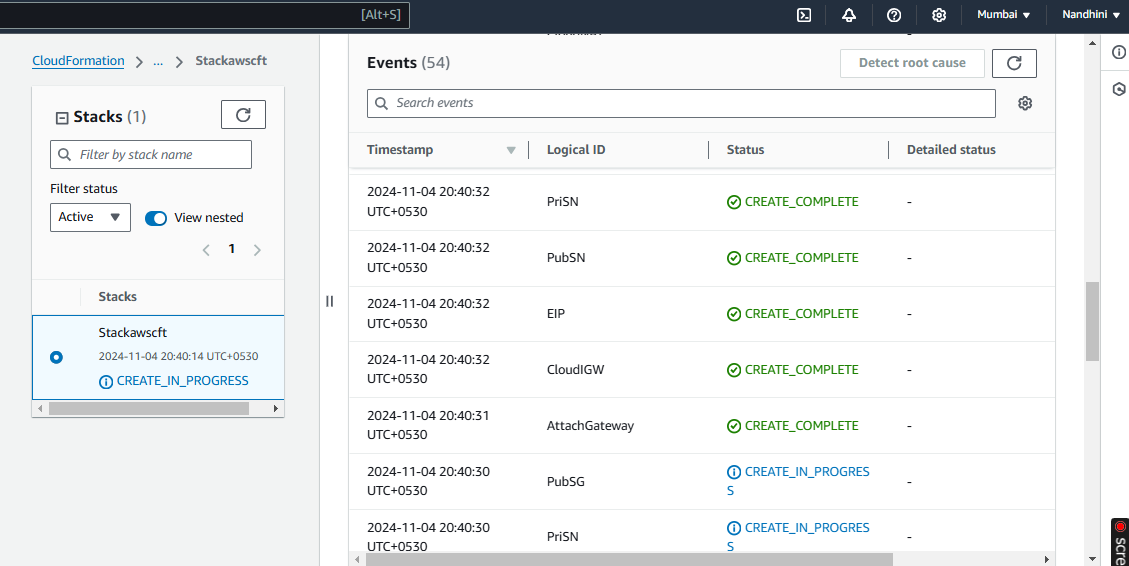
*Description: Private EC2 Instance ID*

*Value: !Ref CloudInstance2*

**Stack creation in progress:**

****

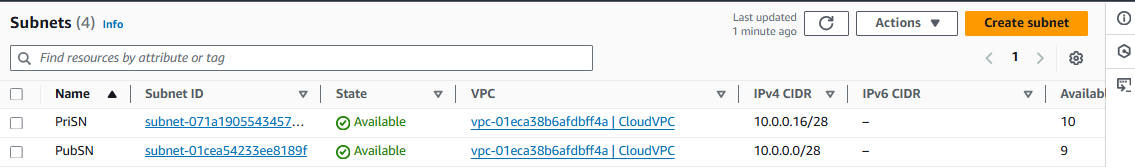
**Events created:**

****

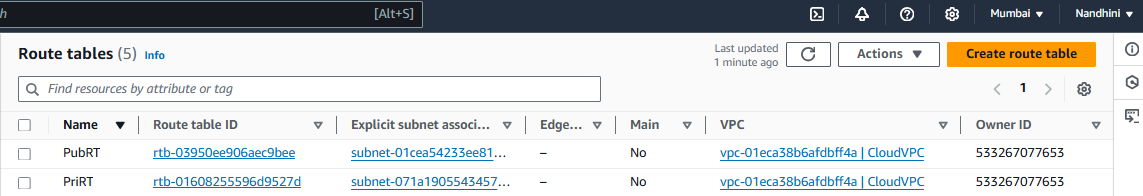
**VPC**

****

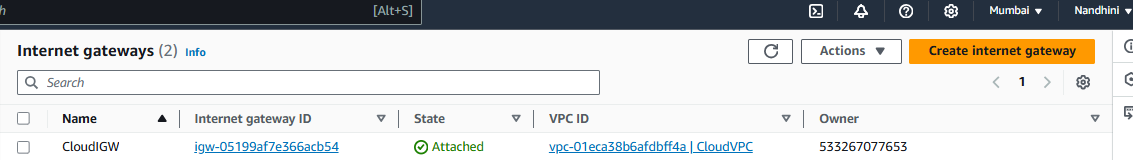
**Subnets**

****

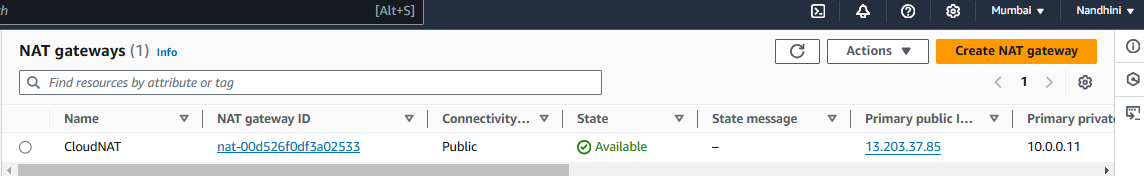
**Route Tables**

****

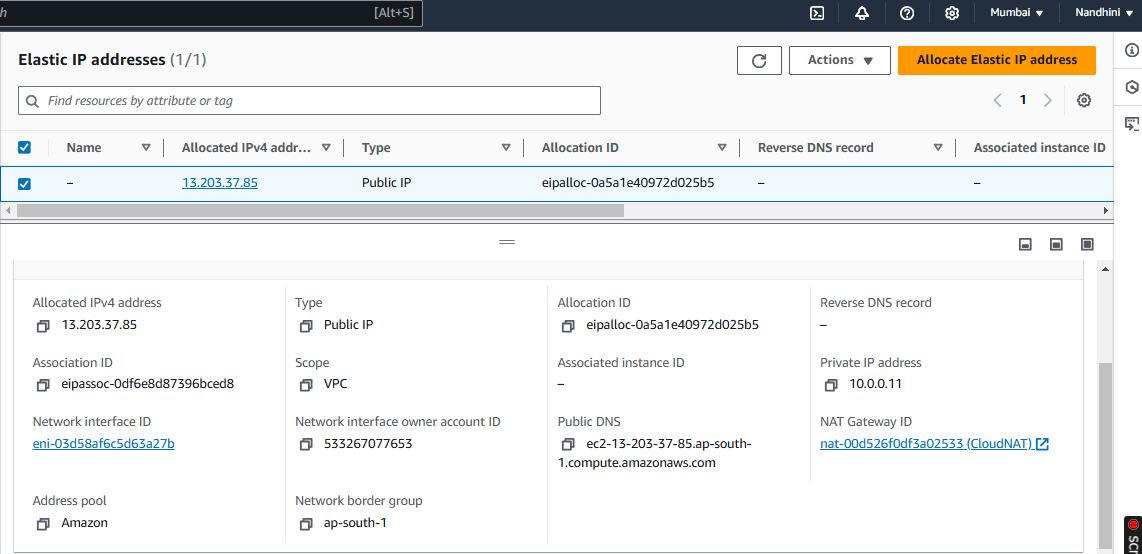
**Internet Gateway**

****

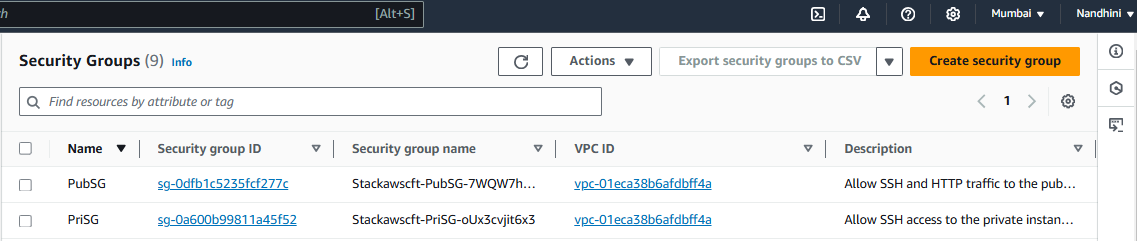
**NAT Gateway**

****

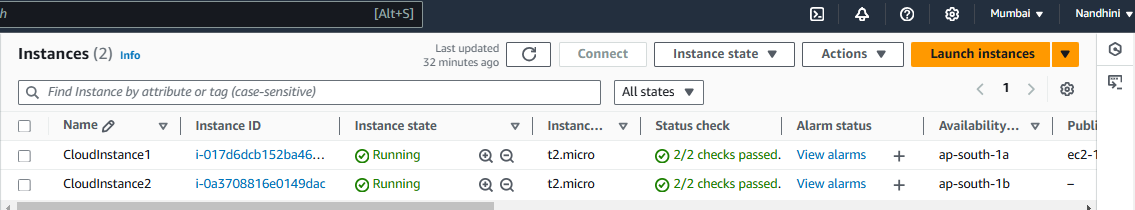
**Elastic IP addresses**

****

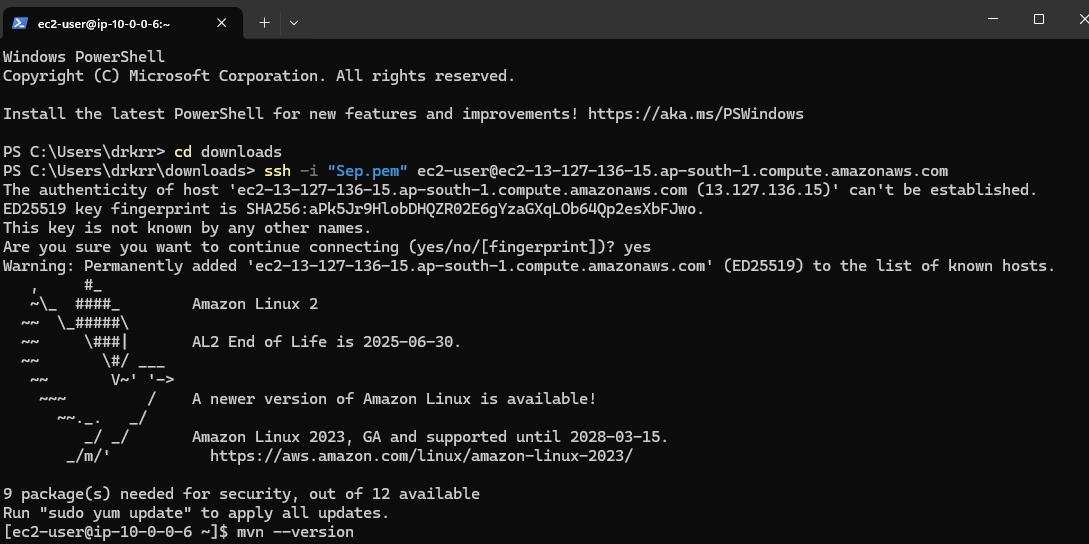
**Security Group**

****

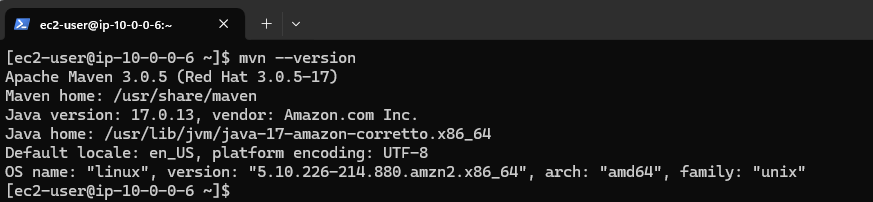
**EC2 Instance**

****

**Logged in CloudInstance1 and checked internet connection:**

****

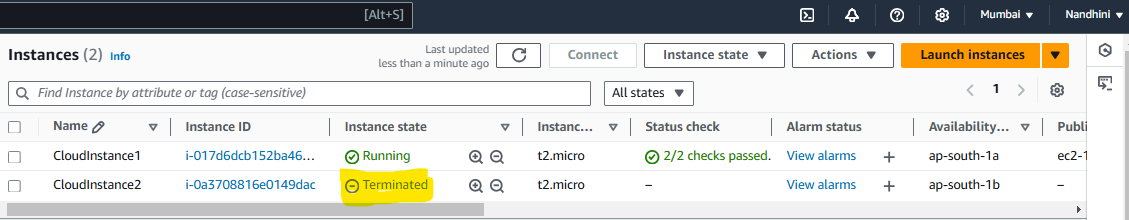
**Installed Java to check network connection:**

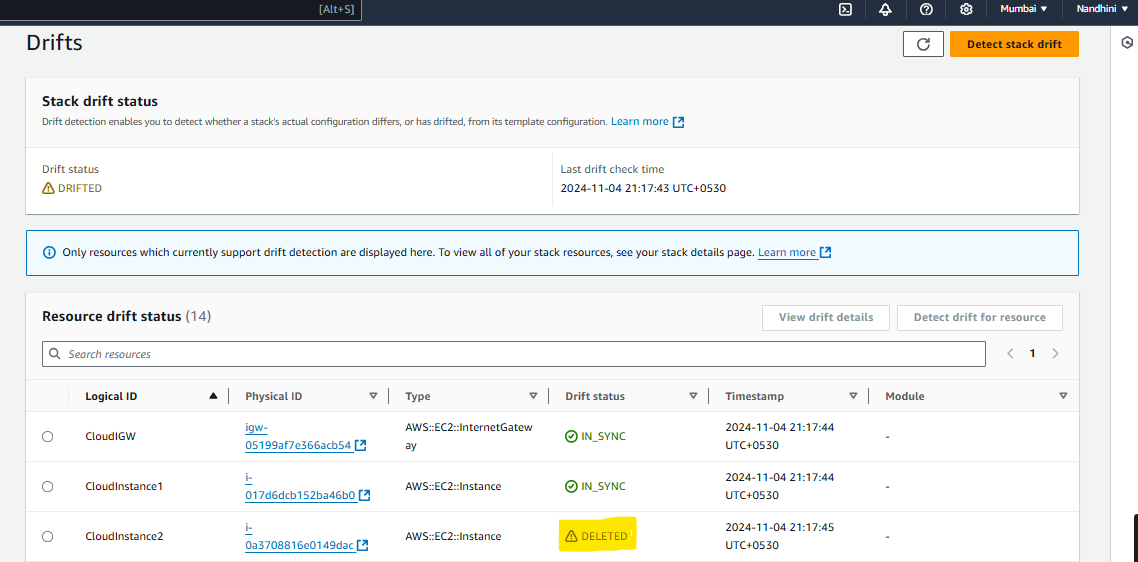
****

**DRIFT DETECTION**

**Drift detection**: Checks if the actual state of resources in a CloudFormation stack differs from the expected state defined in the stack’s template. With drift detection, you can identify, view, and respond to these changes, ensuring your infrastructure stays consistent and compliant with the original template.

**Manually terminated CloudInstance2 and performed Detect stack drift, got the below output.**

****

****