**Task 17: Crafting a Virtual Private Cloud Infrastructure via CloudFormation Template**

AWS CloudFormation (CFT) is a service that allows you to model, provision, and manage AWS resources using code templates. With CloudFormation, you can define infrastructure as code (IaC) in JSON or YAML files, describing the resources and their configurations. It automates resource creation and management, making it easier to deploy, update, and scale infrastructure consistently across environments.

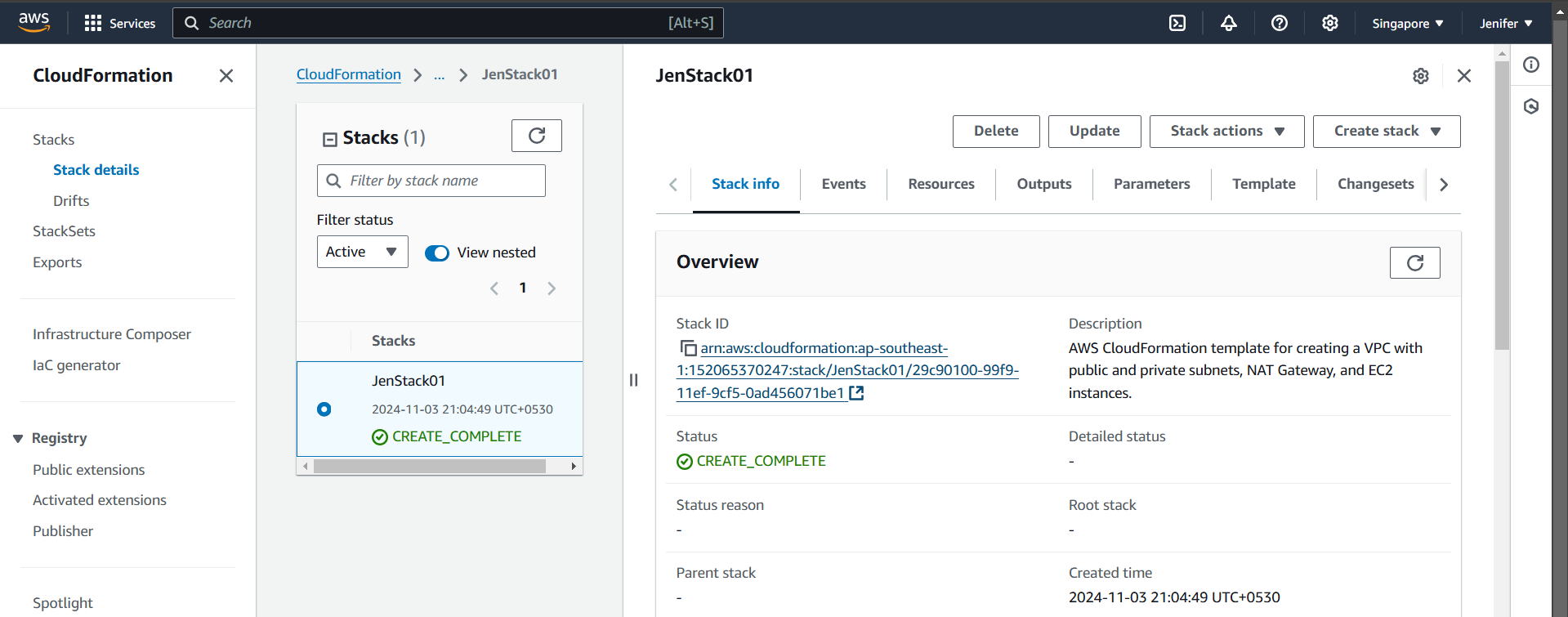
Here**,** I have crafted a Virtual Private Cloud Infrastructure using CFT

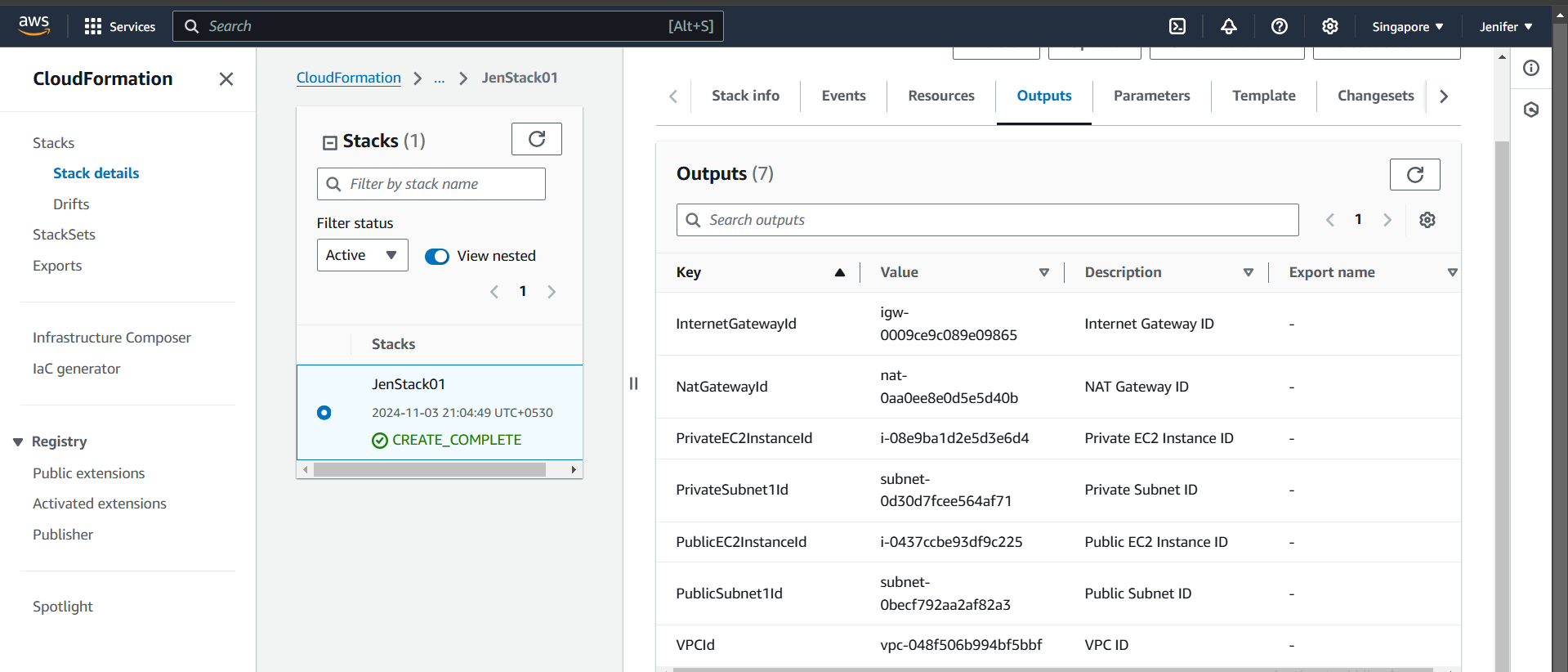
The Components Involved**,**

* + VPC
  + Public & Private Subnets
  + Public & Private Route Tables
  + Internet Gateway
  + NAT Gateway
  + EIP
  + EC2 Instances ( WebServer & AppServer)

1. **Created a Stack named JenStack01**

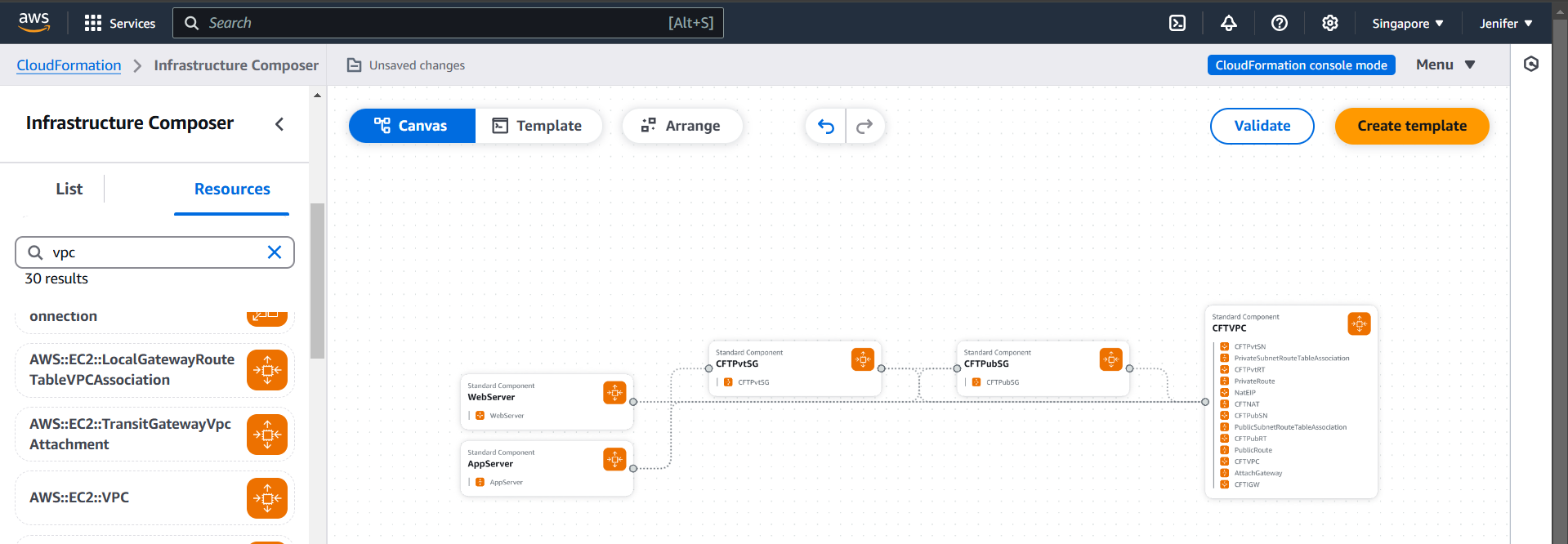
Stacks in AWS CloudFormation provide a powerful way to manage AWS resources collectively, enabling automation, consistency, and ease of management in cloud infrastructure deployment. They simplify the process of creating and managing complex cloud environments while ensuring that resources are created in the correct order and can be updated or deleted together.





1. **After Creating the Stack I have got the Canvas for my VPC Network as below**

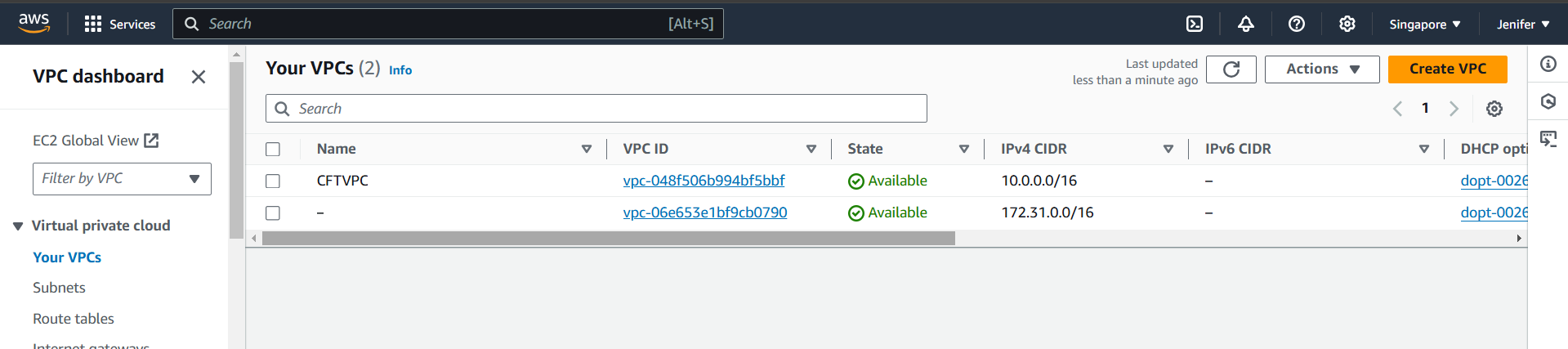
The canvas in AWS CloudFormation refers to the visual representation and design capabilities of the CloudFormation Designer, facilitating easier and more intuitive template creation.



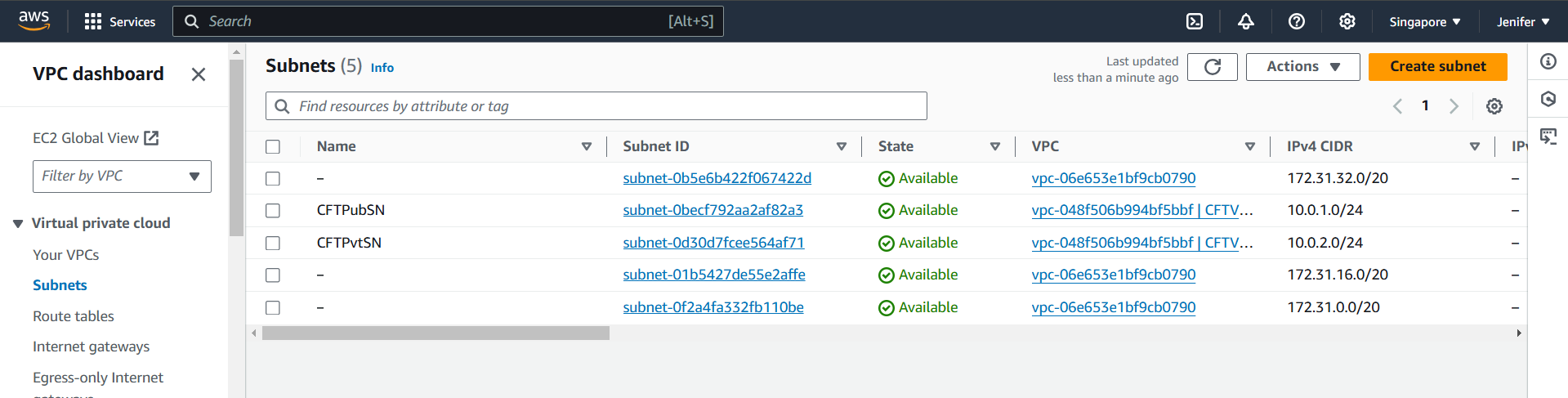
1. **I have used YAML as the language to create the appropriate Template and I have given the entire code in the end of this runbook.**

**Outputs:**

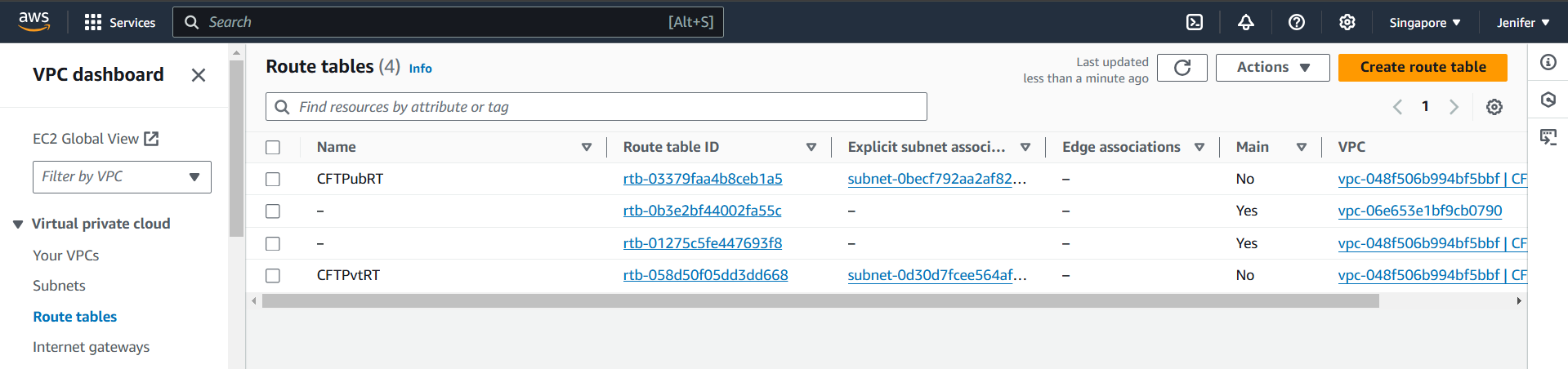
* **VPC got created (Custom Name: CFTVPC)**



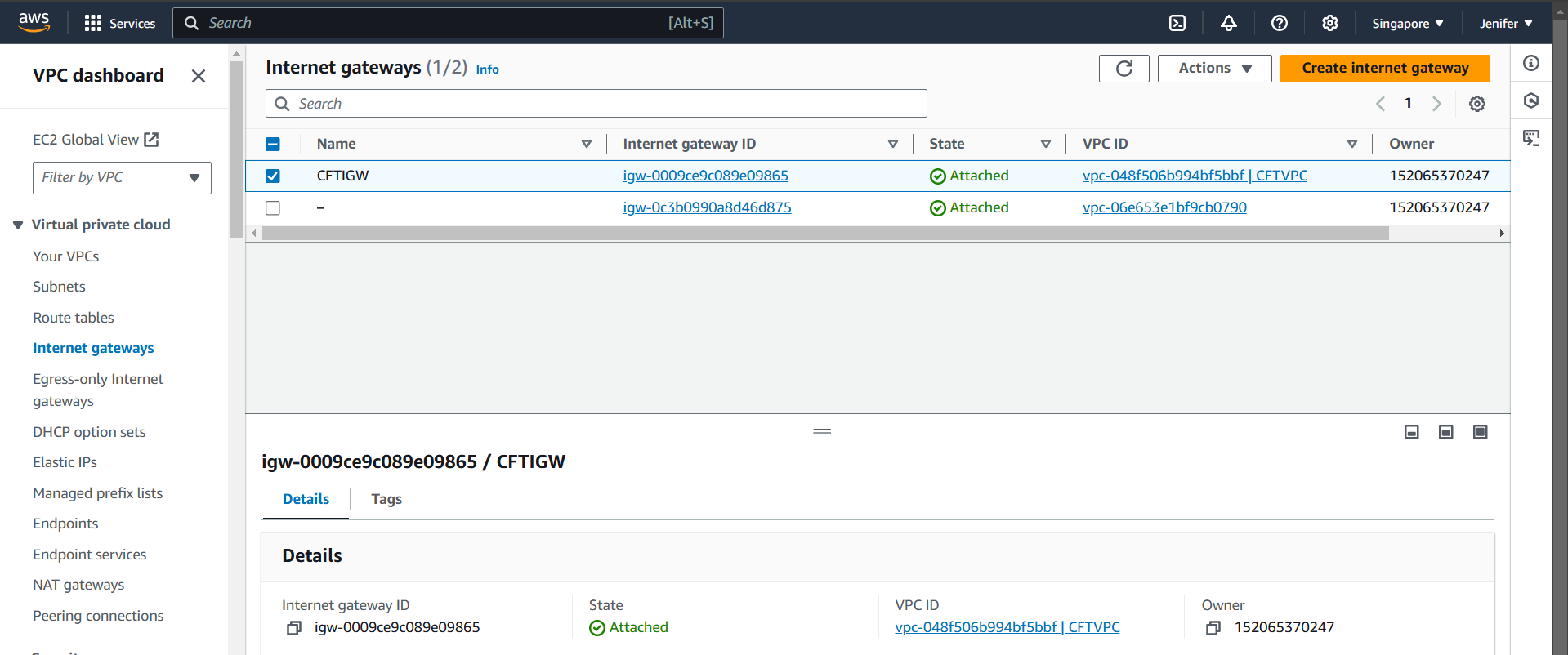
* **Subnets got created (Custom Name: CFTPubSN & CFTPvtSN)**



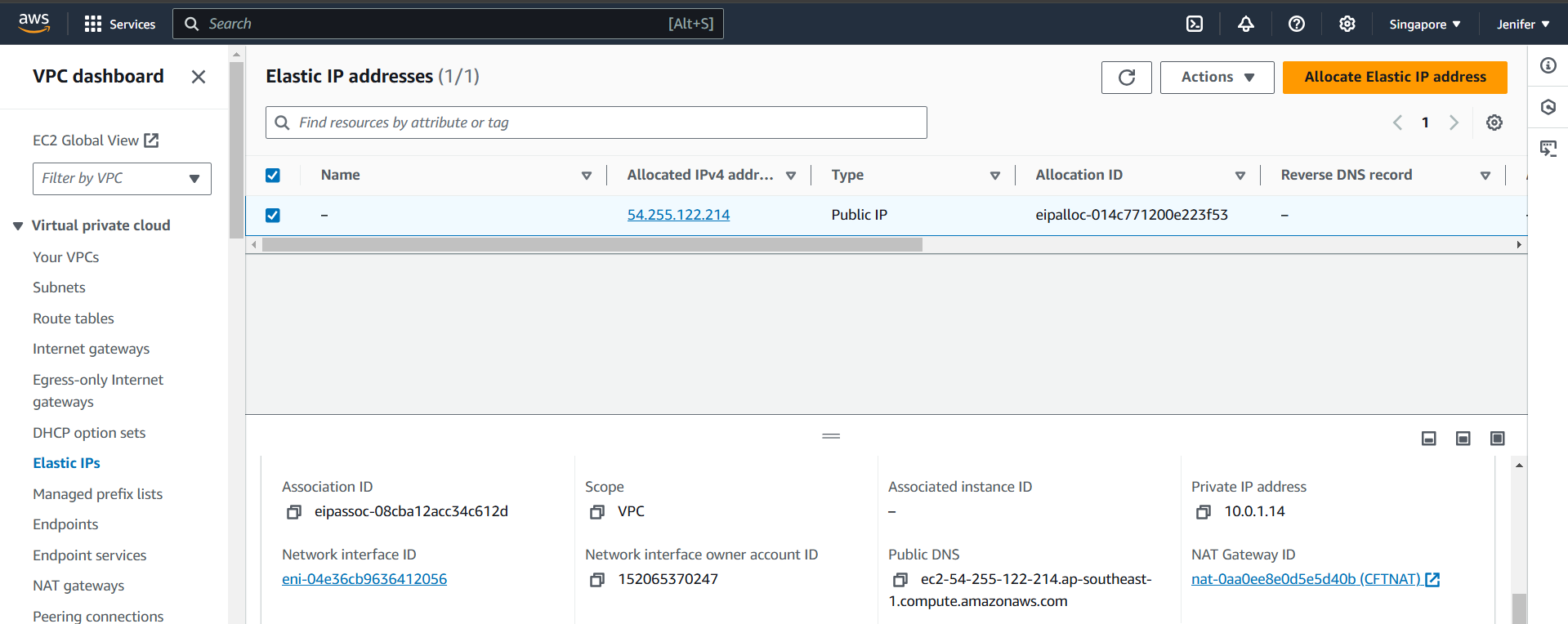
* **Route Tables got created (Custom Name: CFTPubRT & CFTPvtRT)**



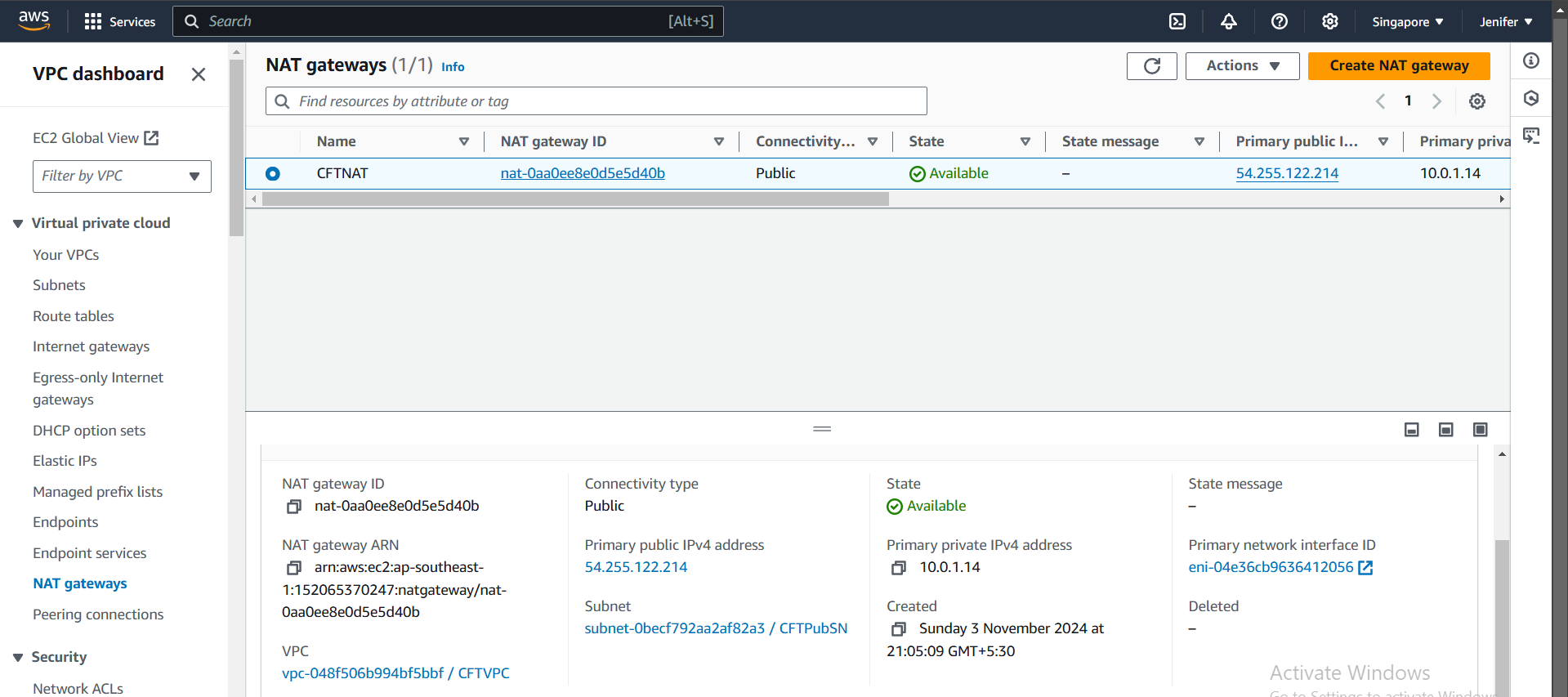
* **Internet Gateway got created (Custom Name: CFTIGW)**



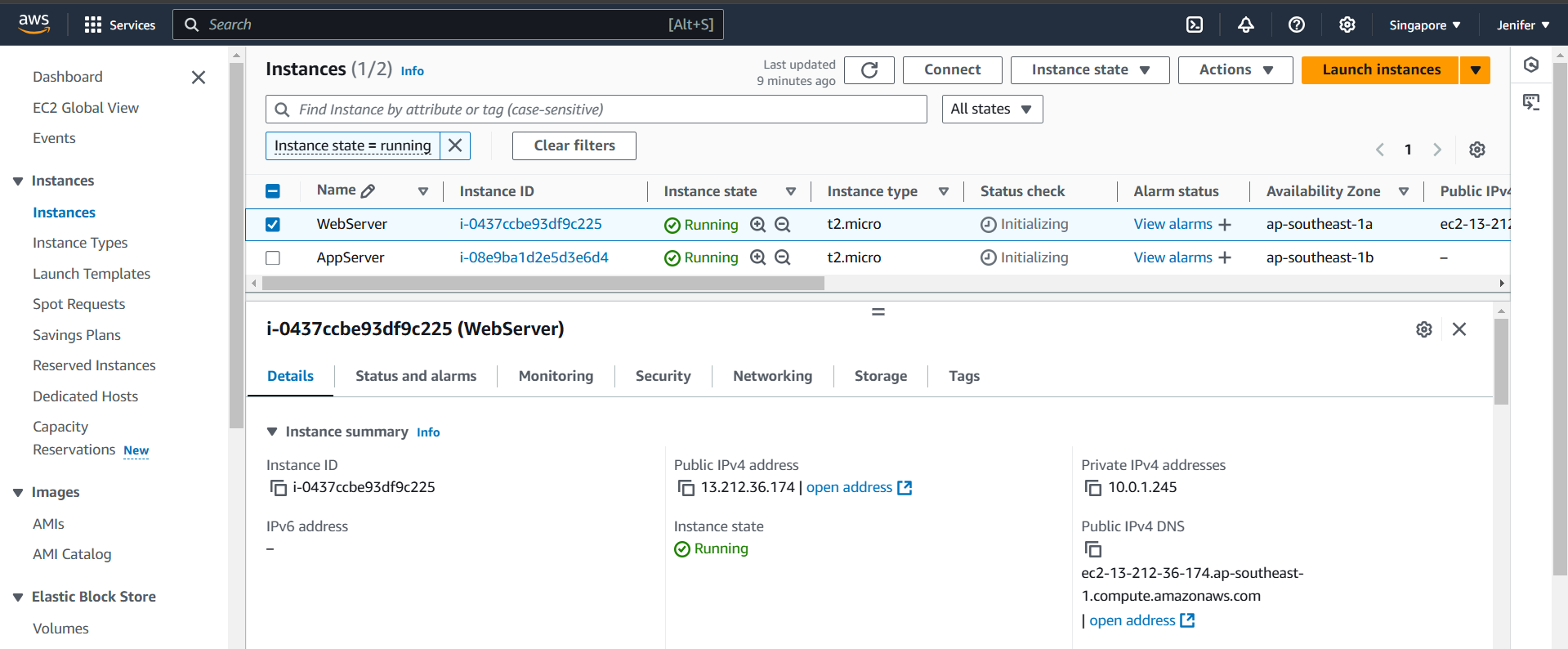
* **Elastic IP got created**

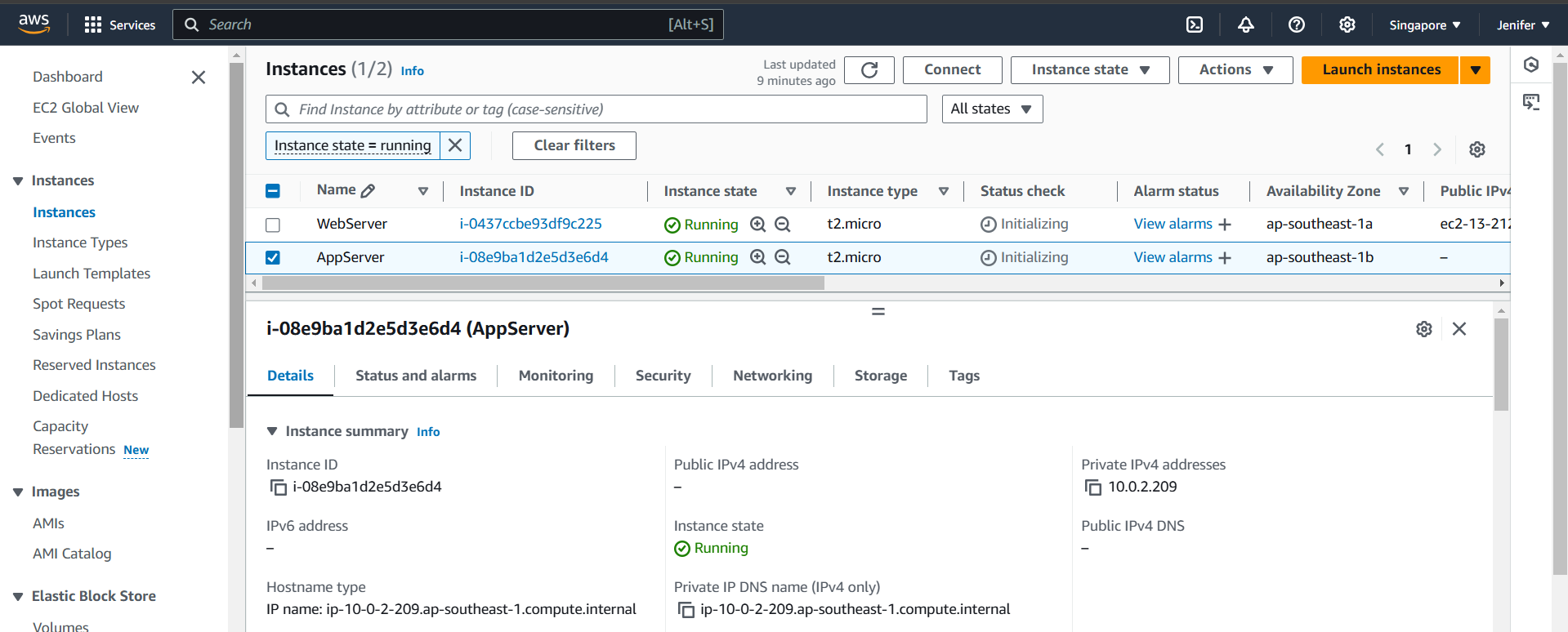


* **NAT Gateway got created(Custom Name: CFTNGW)**

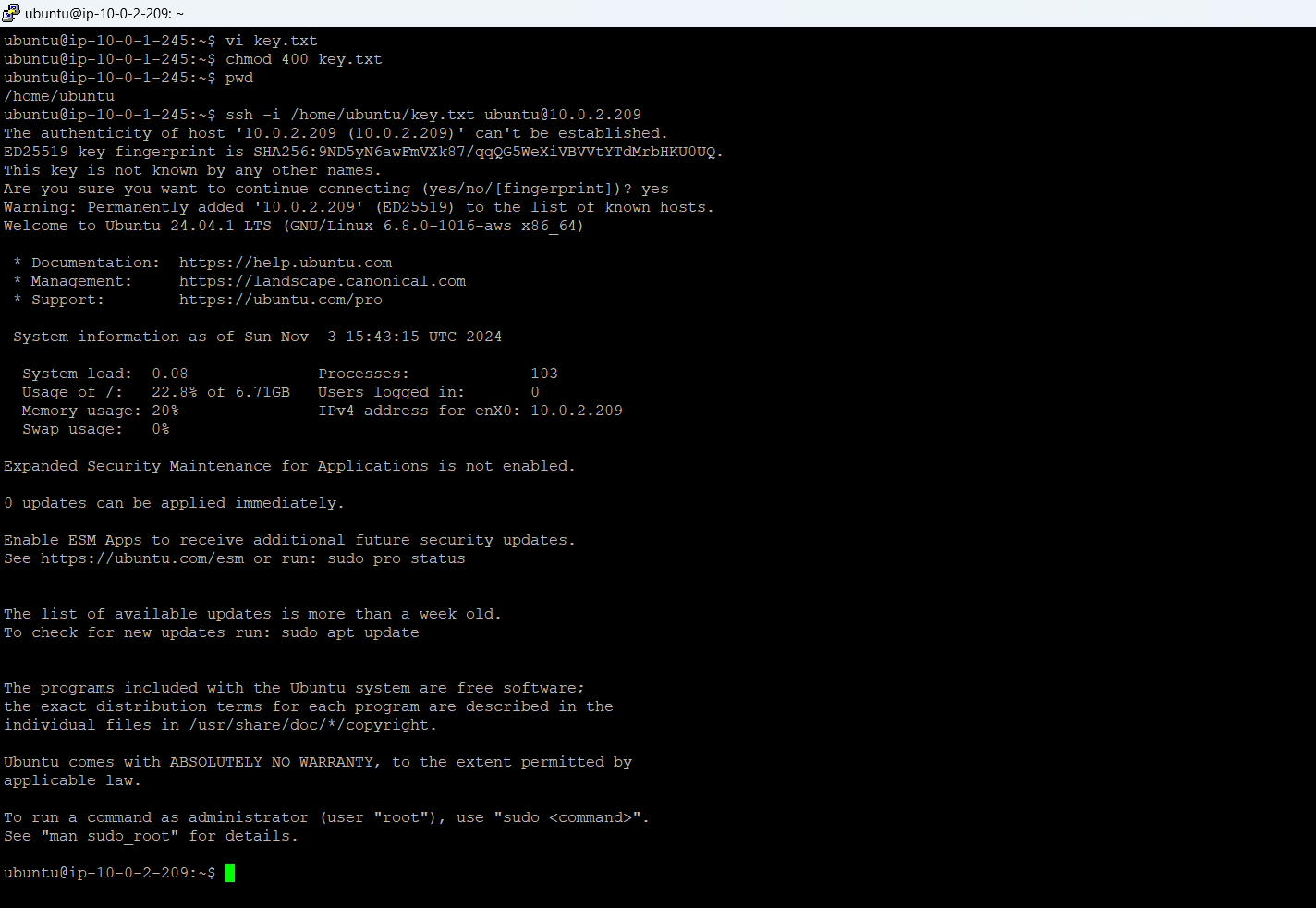


* **EC2 Instances got created (Custom Name: WebServer & AppServer)**





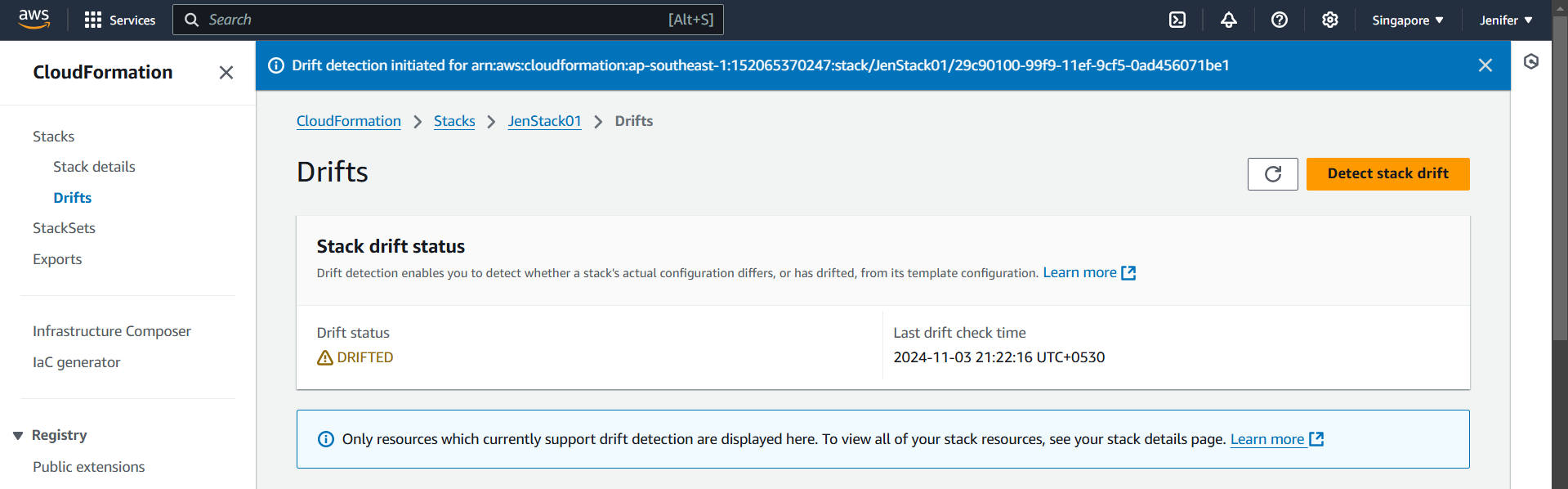
* **Successfully Logged into the WebServer and Jumped into the AppServer via SSH**

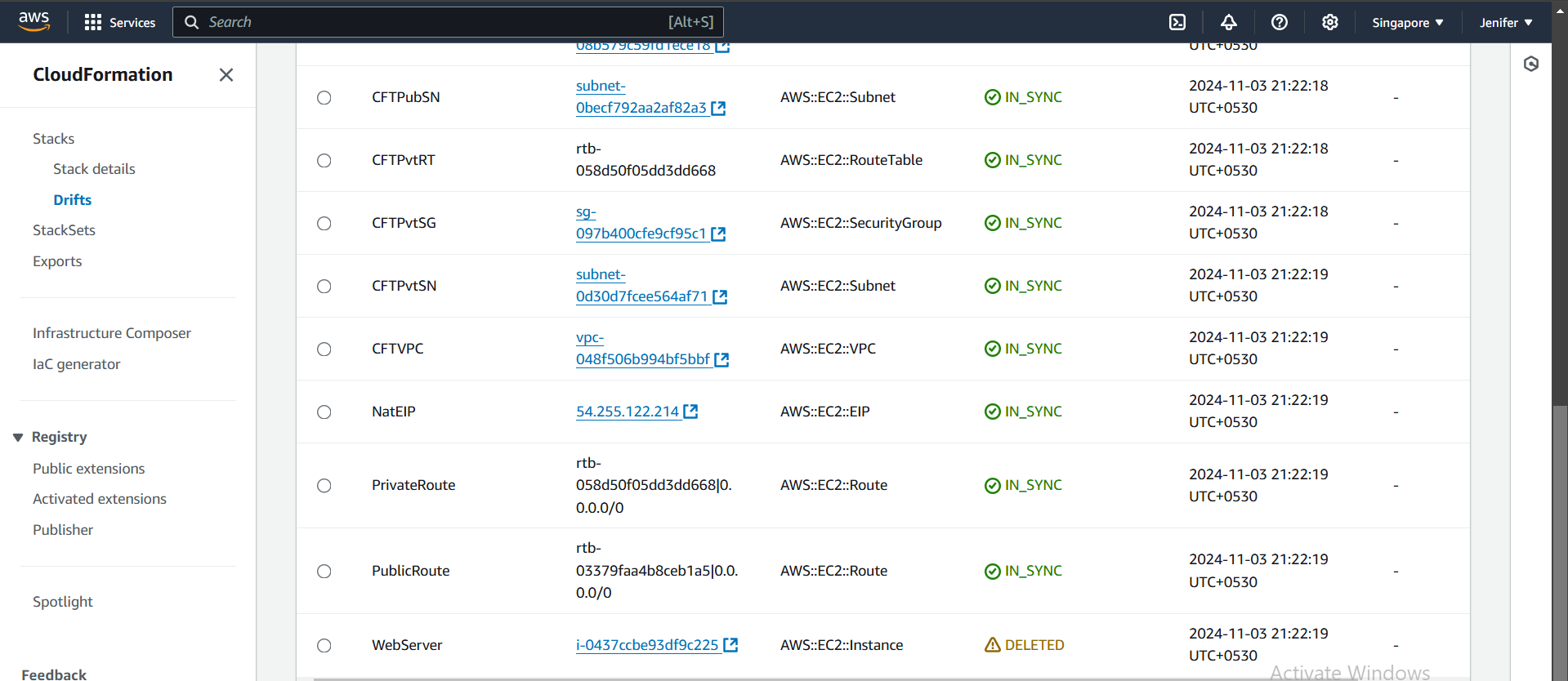


1. **Drift Detection**

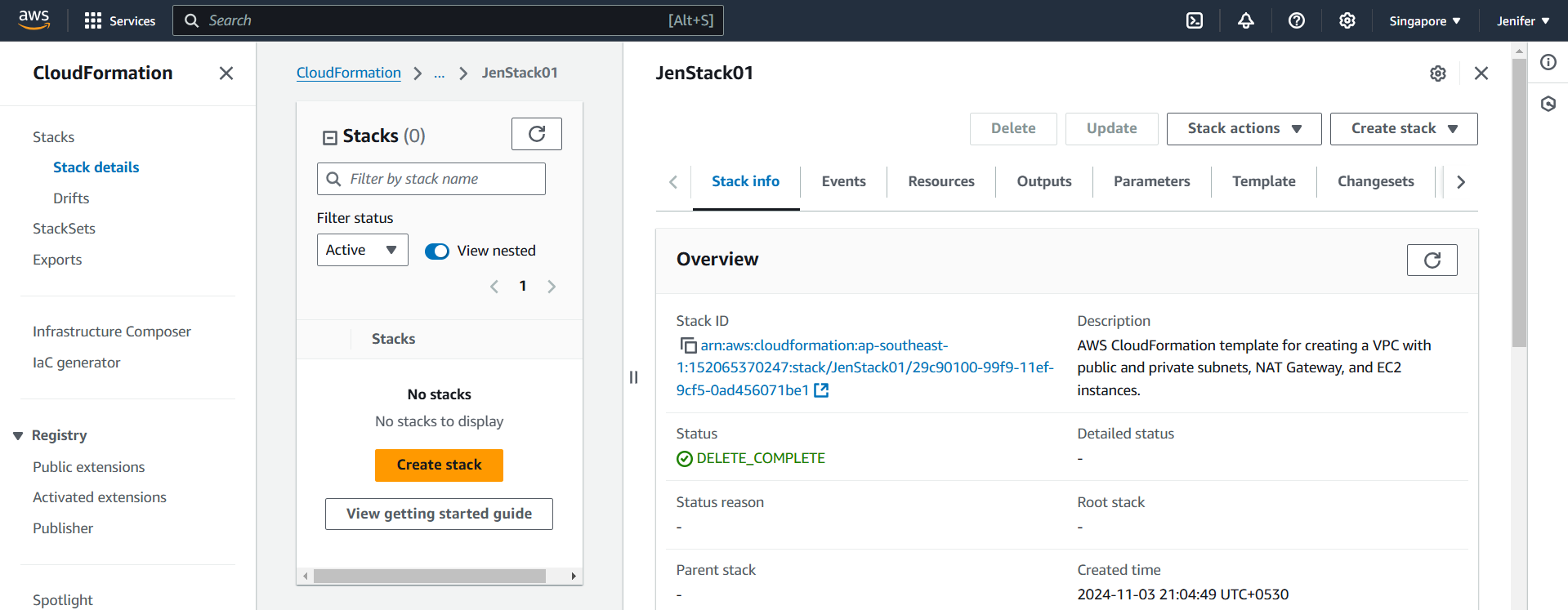
Drift detection in Infrastructure as Code (IaC) is the process of identifying differences, or "drift," between the current state of infrastructure resources in the cloud and the desired state defined in IaC templates.

Here**,** I have deleted the WebServer manually and got the output as below





1. **Deleted the Stack(JenStack01)**



After deleting the stack all the components got deleted automatically.

**YAML Code**

**AWSTemplateFormatVersion: '2010-09-09'Description: AWS CloudFormation template for creating a VPC with public and private subnets, NAT Gateway, and EC2 instances.Resources: # VPC CFTVPC: Type: AWS::EC2::VPC Properties: CidrBlock: 10.0.0.0/16 EnableDnsSupport: true EnableDnsHostnames: true Tags: - Key: Name Value: CFTVPC # Public Subnet CFTPubSN: Type: AWS::EC2::Subnet Properties: VpcId: !Ref CFTVPC CidrBlock: 10.0.1.0/24 AvailabilityZone: !Select - 0 - !GetAZs '' MapPublicIpOnLaunch: true Tags: - Key: Name Value: CFTPubSN # Private Subnet CFTPvtSN: Type: AWS::EC2::Subnet Properties: VpcId: !Ref CFTVPC CidrBlock: 10.0.2.0/24 AvailabilityZone: !Select - 1 - !GetAZs '' MapPublicIpOnLaunch: false Tags: - Key: Name Value: CFTPvtSN**

**# Internet Gateway CFTIGW: Type: AWS::EC2::InternetGateway Properties: Tags: - Key: Name Value: CFTIGW # Attach Internet Gateway to VPC AttachGateway: Type: AWS::EC2::VPCGatewayAttachment Properties: VpcId: !Ref CFTVPC InternetGatewayId: !Ref CFTIGW # Public Route Table CFTPubRT: Type: AWS::EC2::RouteTable Properties: VpcId: !Ref CFTVPC Tags: - Key: Name Value: CFTPubRT # Route to Internet in Public Route Table PublicRoute: Type: AWS::EC2::Route Properties: RouteTableId: !Ref CFTPubRT DestinationCidrBlock: 0.0.0.0/0 GatewayId: !Ref CFTIGW # Associate Public Route Table with Public Subnet PublicSubnetRouteTableAssociation: Type: AWS::EC2::SubnetRouteTableAssociation Properties: SubnetId: !Ref CFTPubSN RouteTableId: !Ref CFTPubRT # Private Route Table CFTPvtRT: Type: AWS::EC2::RouteTable Properties: VpcId: !Ref CFTVPC Tags: - Key: Name Value: CFTPvtRT # NAT Gateway Elastic IP NatEIP: Type: AWS::EC2::EIP Properties: {}**

**# NAT Gateway CFTNAT: Type: AWS::EC2::NatGateway Properties: AllocationId: !GetAtt NatEIP.AllocationId SubnetId: !Ref CFTPubSN Tags: - Key: Name Value: CFTNAT # Route for Private Route Table to use NAT Gateway PrivateRoute: Type: AWS::EC2::Route Properties: RouteTableId: !Ref CFTPvtRT DestinationCidrBlock: 0.0.0.0/0 NatGatewayId: !Ref CFTNAT # Associate Private Route Table with Private Subnet PrivateSubnetRouteTableAssociation: Type: AWS::EC2::SubnetRouteTableAssociation Properties: SubnetId: !Ref CFTPvtSN RouteTableId: !Ref CFTPvtRT # Security Group for Public EC2 CFTPubSG: Type: AWS::EC2::SecurityGroup Properties: GroupDescription: Allow SSH and HTTP traffic to the public instance VpcId: !Ref CFTVPC 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: CFTPubSG # Security Group for Private EC2 CFTPvtSG: Type: AWS::EC2::SecurityGroup Properties: GroupDescription: Allow SSH access to the private instance from public instance VpcId: !Ref CFTVPC SecurityGroupIngress: - IpProtocol: tcp FromPort: 22 ToPort: 22 SourceSecurityGroupId: !Ref CFTPubSG # Allow SSH from public instance Tags: - Key: Name Value: CFTPvtSG # EC2 Instance in Public Subnet WebServer: Type: AWS::EC2::Instance Properties: InstanceType: t2.micro # Change instance type as needed KeyName: T15 # Replace with your key pair name ImageId: ami-047126e50991d067b # Replace with a valid AMI ID for your region SubnetId: !Ref CFTPubSN SecurityGroupIds: - !Ref CFTPubSG Tags: - Key: Name Value: WebServer # EC2 Instance in Private Subnet AppServer: Type: AWS::EC2::Instance Properties: InstanceType: t2.micro # Change instance type as needed KeyName: T15 # Replace with your key pair name ImageId: ami-047126e50991d067b # Replace with a valid AMI ID for your region SubnetId: !Ref CFTPvtSN SecurityGroupIds: - !Ref CFTPvtSG Tags: - Key: Name Value: AppServer**

**Outputs: VPCId: Description: VPC ID Value: !Ref CFTVPC PublicSubnet1Id: Description: Public Subnet ID Value: !Ref CFTPubSN PrivateSubnet1Id: Description: Private Subnet ID Value: !Ref CFTPvtSN InternetGatewayId: Description: Internet Gateway ID Value: !Ref CFTIGW NatGatewayId: Description: NAT Gateway ID Value: !Ref CFTNAT PublicEC2InstanceId: Description: Public EC2 Instance ID Value: !Ref WebServer PrivateEC2InstanceId: Description: Private EC2 Instance ID Value: !Ref AppServer**