# AWS Network Design using CloudFormation Template

## PART 1: Understanding of the CloudFormation Template

For this task, we are going to use be the attached CloudFormation Template.

As you can see the entire CloudFormation template is written in the YAML language, you can also write the same template in JSON if you're more comfortable with it.

The Template automates the following Tasks:

- Creation of Custom VPC.
- Creation of Custom InternetGateway and its attachment to the Custom VPC.
- Creation of two Public Subnets in two different Availability Zones.
- Creation of two Private Subnets in two different Availability Zones.
- Creation of two NAT Gateways in two Custom Public Subnets with EIP assignments.
- Creation of Public Route Tables with Route assignments.
- Public Subnets association to Public Route Table.
- Creation of two Private Route Tables with Route assignments.
- Private Subnets association to each Private Route Tables.
- Creation of a Custom Security Group.

The CloudFormation template is divided into three different parts, which are Descriptions, Parameters, Resources and Output.

- Description: It states the entire summary of the CloudFormation template. It just gives an overview of the entire template.

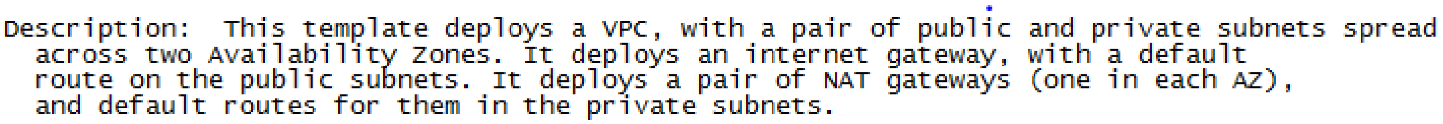
- Parameters: It is not possible to share a same template across multiple regions since Subnet ID, Security Groups Names, Key Pairs, Snapshot IDs differs in each region. You can overcome this problem by specifying PARAMETERS in CloudFormation Template which asks for Custom Values at the start of the template deployment.

- Resources: Specifies the stack of resources that the CloudFormation template is going to get deployed.

- Output: You can use Output_parameters of one CloudFormation template as Input to another CloudFormation template.

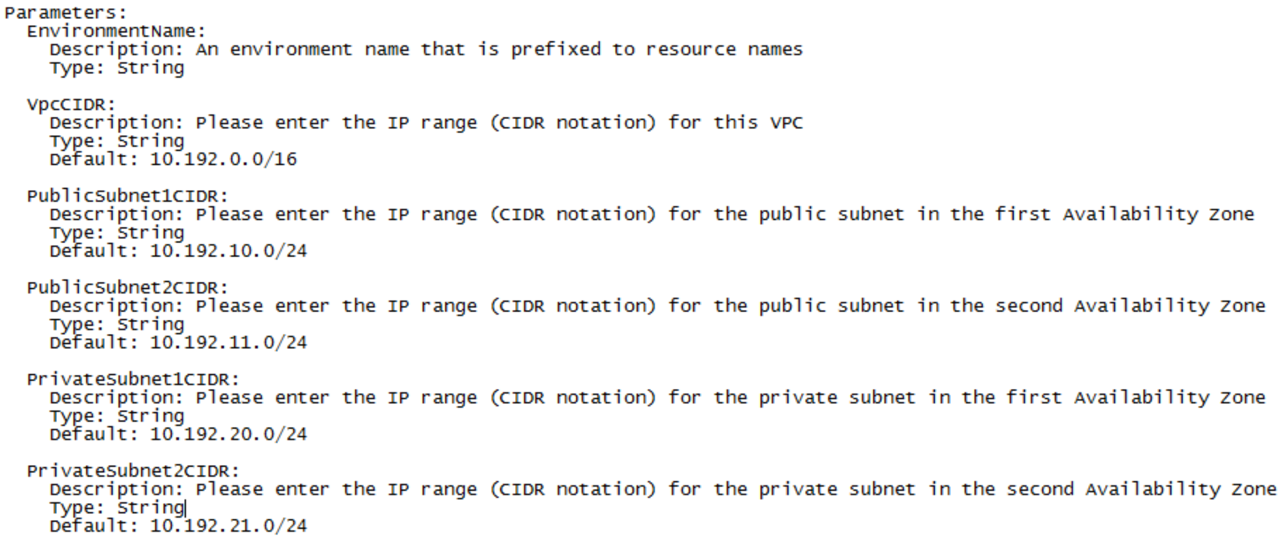
Since we've glance on each part of the CloudFormation Template. Now let's verify each and every thing that we discussed practically.

## Description:



```
Description:  This template deploys a VPC, with a pair of public and private subnets spread
  across two Availability Zones. It deploys an internet gateway, with a default
  route on the public subnets. It deploys a pair of NAT gateways (one in each AZ),
  and default routes for them in the private subnets.
```

## Parameters:

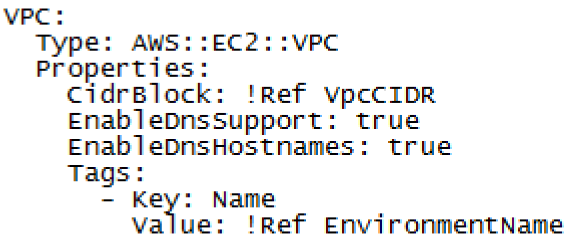


```
Parameters:
  EnvironmentName:
    Description: An environment name that is prefixed to resource names
    Type: String

  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for this VPC
    Type: String
    Default: 10.192.0.0/16

  PublicSubnet1CIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone
    Type: String
    Default: 10.192.10.0/24

  PublicSubnet2CIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone
    Type: String
    Default: 10.192.11.0/24

  PrivateSubnet1CIDR:
    Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone
    Type: String
    Default: 10.192.20.0/24

  PrivateSubnet2CIDR:
    Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone
    Type: String
    Default: 10.192.21.0/24
```

When we'll deploy CloudFormation template later in this lab, you'll notice that the template will ask you to specify some of the Custom Parameters such as VPC CIDR, Public Subnet CIDR and Private Subnet CIDR etc. Because these Custom Parameters are defined in the CloudFormation template as PARAMETERS.

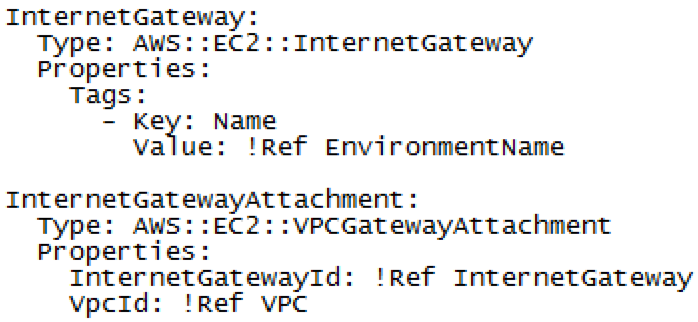
## Resources:

1. Creation of Custom VPC.



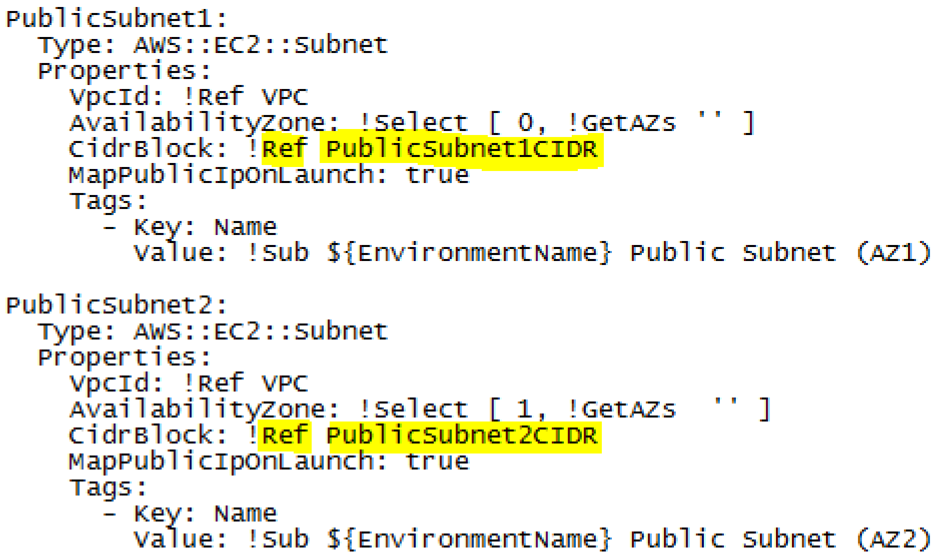
```
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: !Ref VpcCIDR
    EnableDnsSupport: true
    EnableDnsHostnames: true
    Tags:
      - Key: Name
        Value: !Ref EnvironmentName
```

2. Creation of Custom InternetGateway and its attachment to the Custom VPC.



```
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: !Ref EnvironmentName

InternetGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    InternetGatewayId: !Ref InternetGateway
    VpcId: !Ref VPC
```

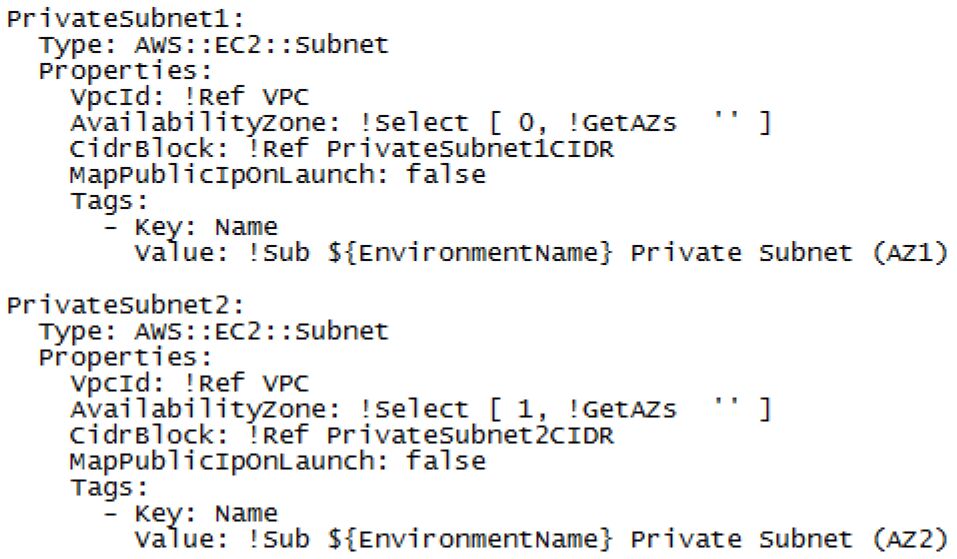
3. Creation of two Public Subnets in two different Availability Zones.



```
PublicSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !Ref PublicSubnet1CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs  '' ]
    CidrBlock: !Ref PublicSubnet2CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ2)
```

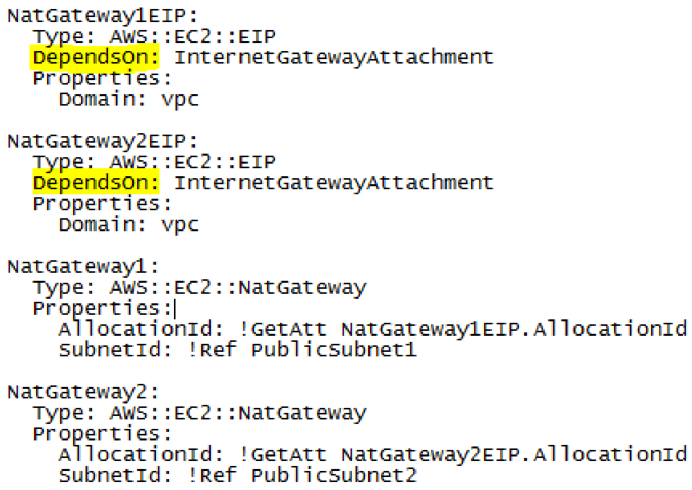
Here, you can see that the **CidrBlock** is referring to the **PublicSubnet1CIDR** that we specified in the PARAMETERS. It takes take the custom specified CIDR block or the default CIDR size of 10.192.10.0/24. Observe and Understand the same while creation of other Public and Private Subnets.

4. Creation of two Private Subnets in two different Availability Zones.



```
PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs  '' ]
    CidrBlock: !Ref PrivateSubnet1CIDR
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Subnet (AZ1)

PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs  '' ]
    CidrBlock: !Ref PrivateSubnet2CIDR
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Subnet (AZ2)
```

5. Creation of two NAT Gateways in two Custom Public Subnets and EIP assignments.

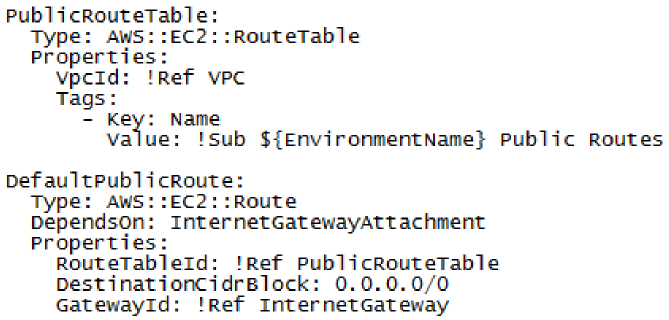


```
NatGateway1EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc

NatGateway2EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc

NatGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway1EIP.AllocationId
    SubnetId: !Ref PublicSubnet1

NatGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway2EIP.AllocationId
    SubnetId: !Ref PublicSubnet2
```

Here makes a note of DependsOn attribute. It means the NatGateway1EIP and NatGateway2EIP tasks are dependent on another task i.e. InternetGatewayAttachment.

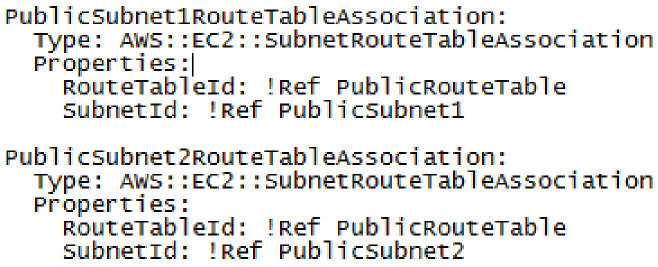
Unless InternetGatewayAttachment task is executed, NatGateway1EIP and NatGateway2EIP won't execute.

6. Creation of Public Route Tables with Route assignments.



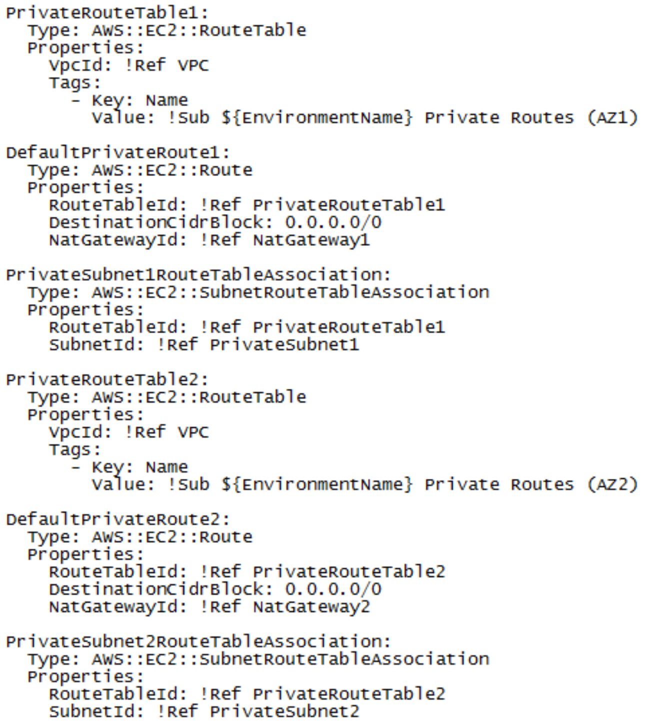
```
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Routes

DefaultPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
```

7. Public Subnets association to Public Route Table.



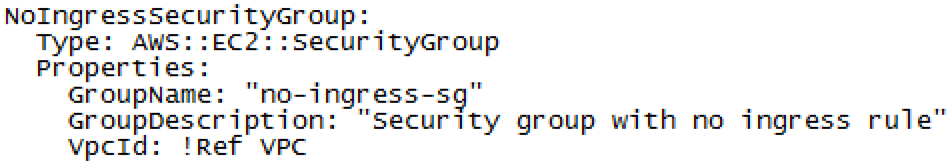
```
PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet1

PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet2
```

8. Creation of two Private Route Tables with Route assignments and Private Subnets association to the Private Route Tables.



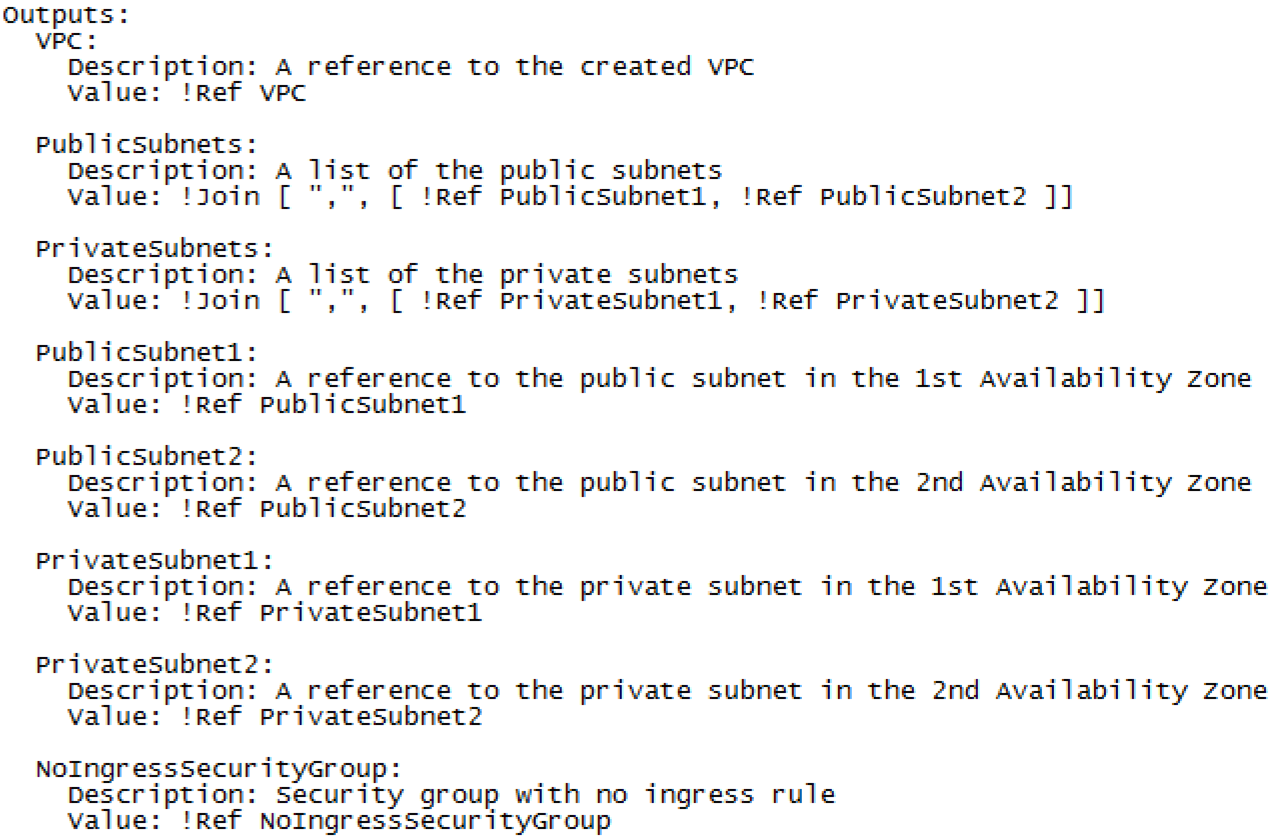
```
PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ1)

DefaultPrivateRoute1:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway1

PrivateSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    SubnetId: !Ref PrivateSubnet1

PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ2)

DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2
```

9. Creation of a Custom Security Group.



```
NoIngressSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "no-ingress-sg"
    GroupDescription: "Security group with no ingress rule"
    VpcId: !Ref VPC
```

## OUTPUTS:



```
Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC

  PublicSubnets:
    Description: A list of the public subnets
    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ]]

  PrivateSubnets:
    Description: A list of the private subnets
    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

  PublicSubnet1:
    Description: A reference to the public subnet in the 1st Availability Zone
    Value: !Ref PublicSubnet1

  PublicSubnet2:
    Description: A reference to the public subnet in the 2nd Availability Zone
    Value: !Ref PublicSubnet2

  PrivateSubnet1:
    Description: A reference to the private subnet in the 1st Availability Zone
    Value: !Ref PrivateSubnet1

  PrivateSubnet2:
    Description: A reference to the private subnet in the 2nd Availability Zone
    Value: !Ref PrivateSubnet2

  NoIngressSecurityGroup:
    Description: Security group with no ingress rule
    Value: !Ref NoIngressSecurityGroup
```

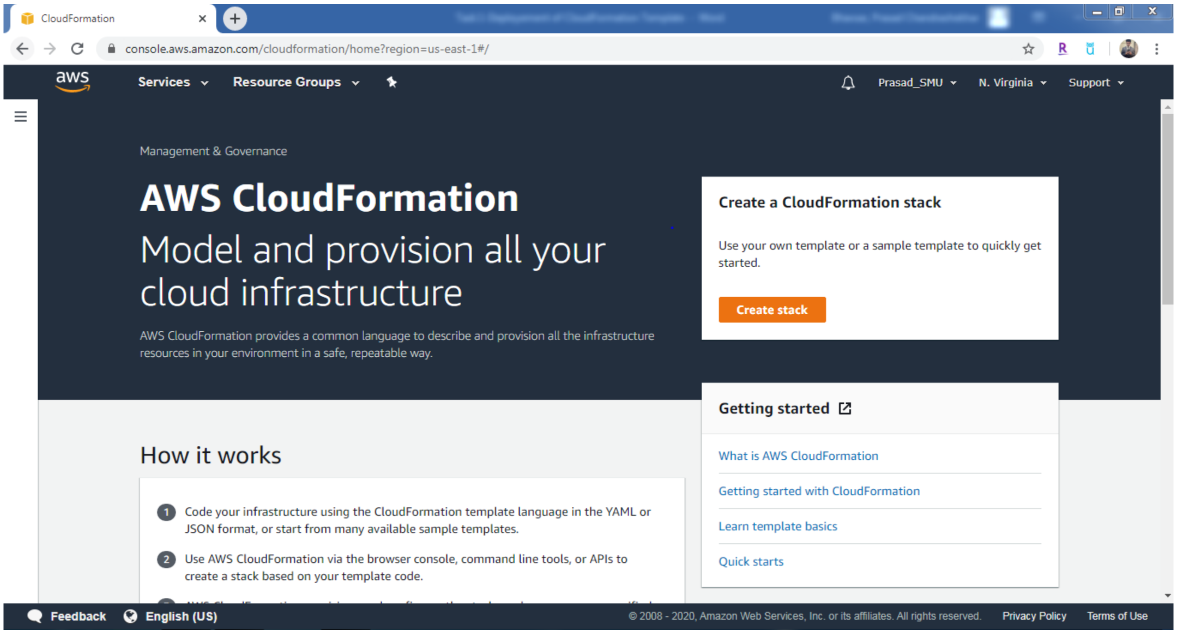
You can use these Output parameters in another CloudFormation Template.

For example, if you're deploying Application Template over this Network Template then you can refer the Output VPC, Security Groups, Public or Private subnets from the original template.

## PART 2: Deploying the CloudFormation Template

Navigate to AWS Management and console and select **US East (N. Virginia) (us-east-1)** region.

Select **CloudFormation** service.


CloudFormation
console.aws.amazon.com/cloudformation/home?region=us-east-1#/
aws
Services
Resource Groups
Prasad_SMU
N. Virginia
Support
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Management & Governance

AWS CloudFormation

Model and provision all your cloud infrastructure

AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your environment in a safe, repeatable way.

Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

Create stack

Getting started

What is AWS CloudFormation

Getting started with CloudFormation

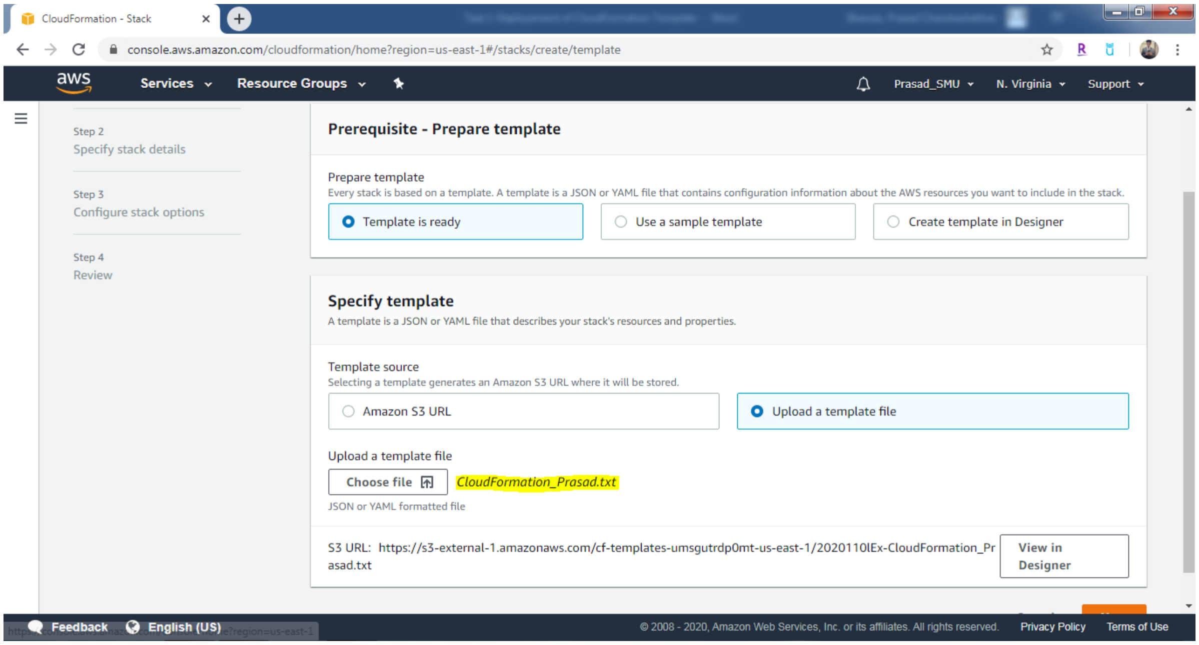
Learn template basics

Quick starts

How it works

1. Code your infrastructure using the CloudFormation template language in the YAML or JSON format, or start from many available sample templates.
2. Use AWS CloudFormation via the browser console, command line tools, or APIs to create a stack based on your template code.

Click on Create **Stack**. Since you've YAML template ready to deploy, you'll select Template is ready and click on Upload a template file and browse to the CloudFormation template file. You can also upload your template from a S3 bucket and click Next.


CloudFormation - Stack
console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/template
aws
Services
Resource Groups
Prasad_SMU
N. Virginia
Support
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

● Template is ready ○ Use a sample template ○ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

○ Amazon S3 URL ● Upload a template file

Upload a template file

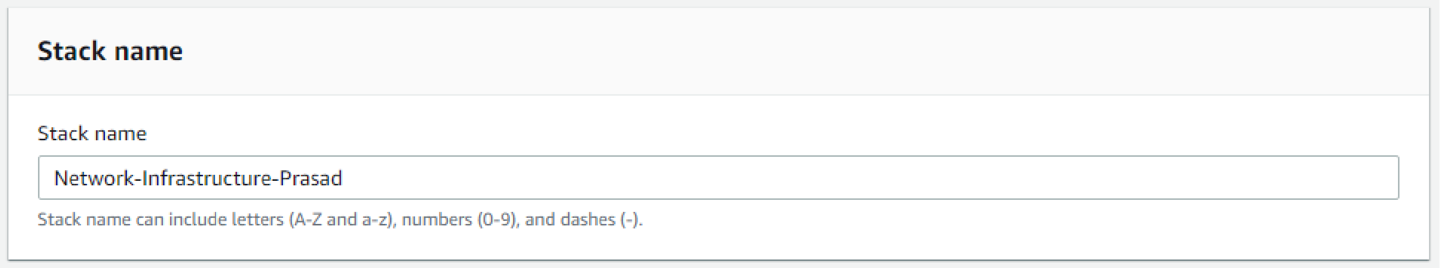
Choose file CloudFormation_Prasad.txt

JSON or YAML formatted file

S3 URL: https://s3-external-1.amazonaws.com/cf-templates-umsgutrdp0mt-us-east-1/2020110lEx-CloudFormation_Prasad.txt

View in Designer

Give Stack Name of your choice.



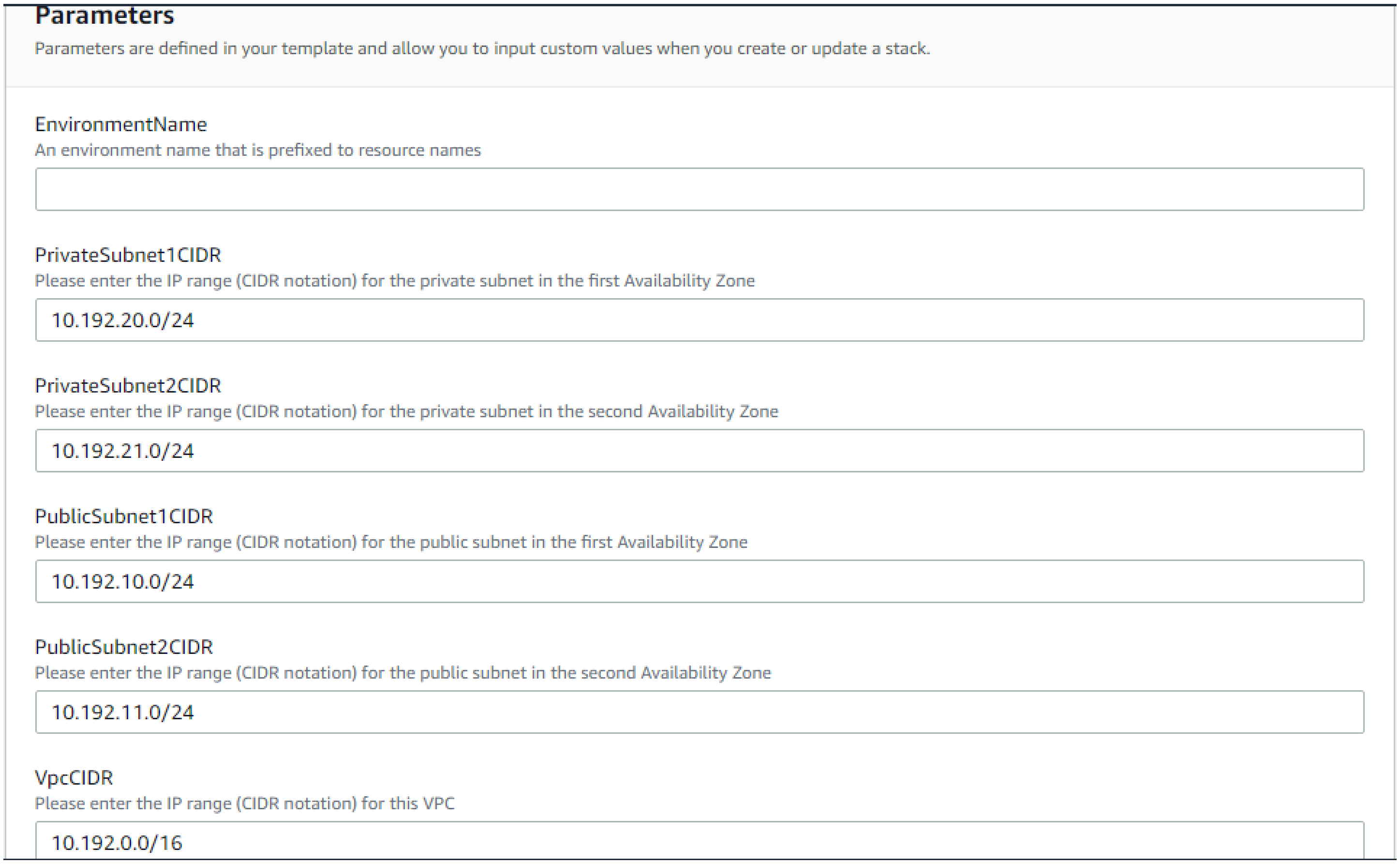
**Stack name**

Stack name

Network-Infrastructure-Prasad

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Now try to remember what we've discussed on CloudFormation template PARAMETERS. As we've specified VPC CIDR, Public Subnet CIDR and Private Subnet CIDR in CloudFormation template, we are now getting an option to put values of our choice, if we skip this option then it will take the default specified Parameter's values.



**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

EnvironmentName

An environment name that is prefixed to resource names

PrivateSubnet1CIDR

Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

10.192.20.0/24

PrivateSubnet2CIDR

Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

10.192.21.0/24

PublicSubnet1CIDR

Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

10.192.10.0/24

PublicSubnet2CIDR

Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

10.192.11.0/24

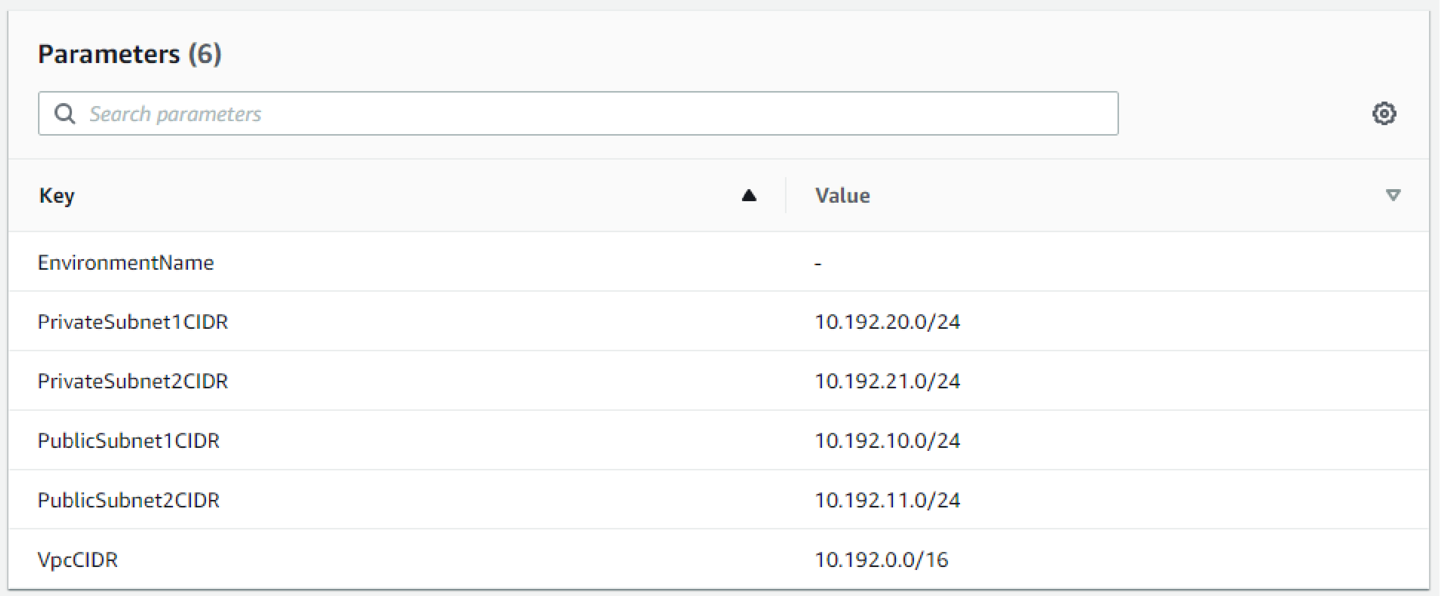
VpcCIDR

Please enter the IP range (CIDR notation) for this VPC

10.192.0.0/16

Stick to the default Parameters values and click Next. On the Configure Stack options, keep the default setting and click Next.

On the Review page, make note of Parameters and click Create Task.

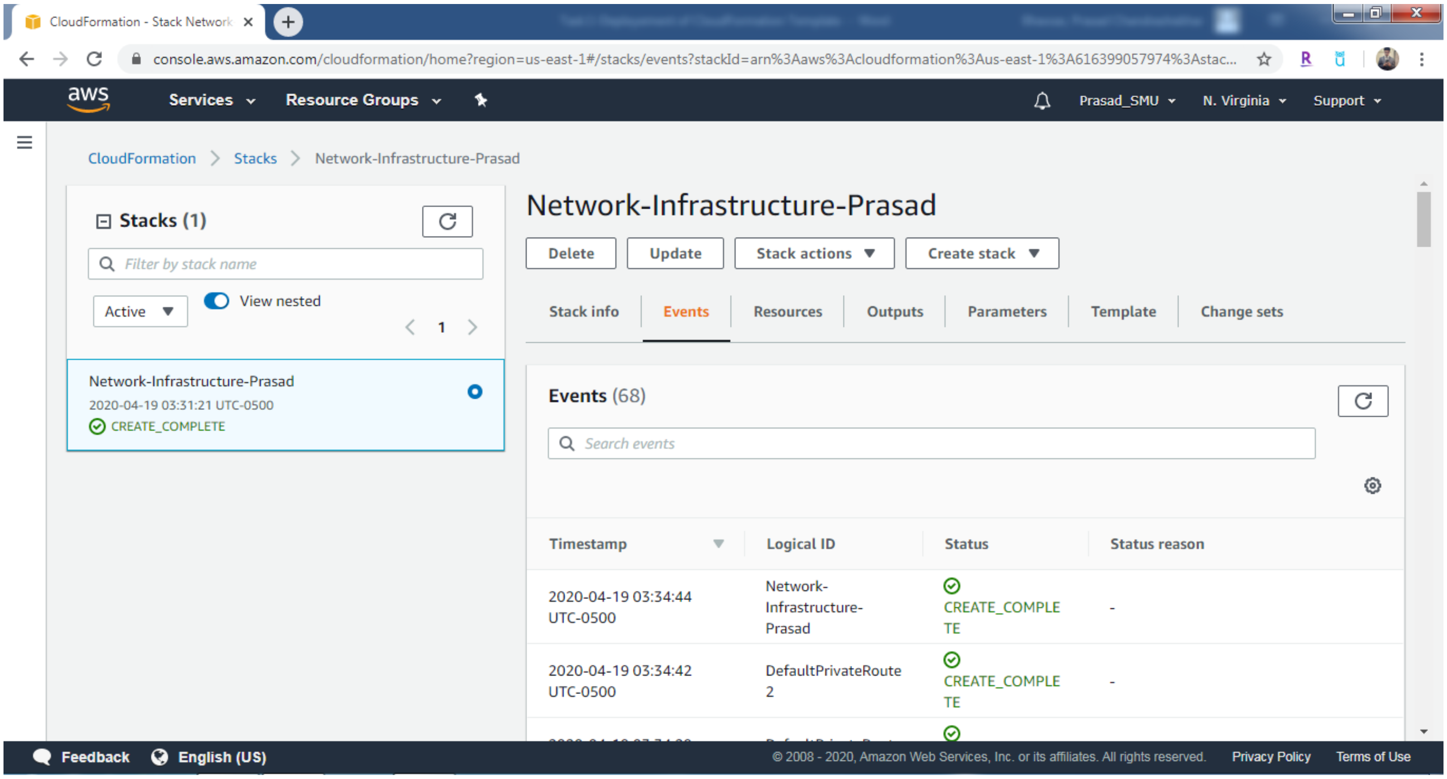


**Parameters (6)**

Search parameters

| Key | Value |
|---|---|
| EnvironmentName | - |
| PrivateSubnet1CIDR | 10.192.20.0/24 |
| PrivateSubnet2CIDR | 10.192.21.0/24 |
| PublicSubnet1CIDR | 10.192.10.0/24 |
| PublicSubnet2CIDR | 10.192.11.0/24 |
| VpcCIDR | 10.192.0.0/16 |

Whatever configurations or resources that you've specified in CloudFormation template's resources field will be deployed for you automatically withing 5 minutes. Keep refreshing.



CloudFormation - Stack Network
console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/events?stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A616399057974%3Astac...
aws
Services
Resource Groups
Prasad_SMU
N. Virginia
Support
CloudFormation > Stacks > Network-Infrastructure-Prasad

Stacks (1)

Filter by stack name

Active ▼ View nested

Network-Infrastructure-Prasad
2020-04-19 03:31:21 UTC-0500
CREATE_COMPLETE

Network-Infrastructure-Prasad

Delete
Update
Stack actions ▼
Create stack ▼
Stack info
Events
Resources
Outputs
Parameters
Template
Change sets

Events (68)

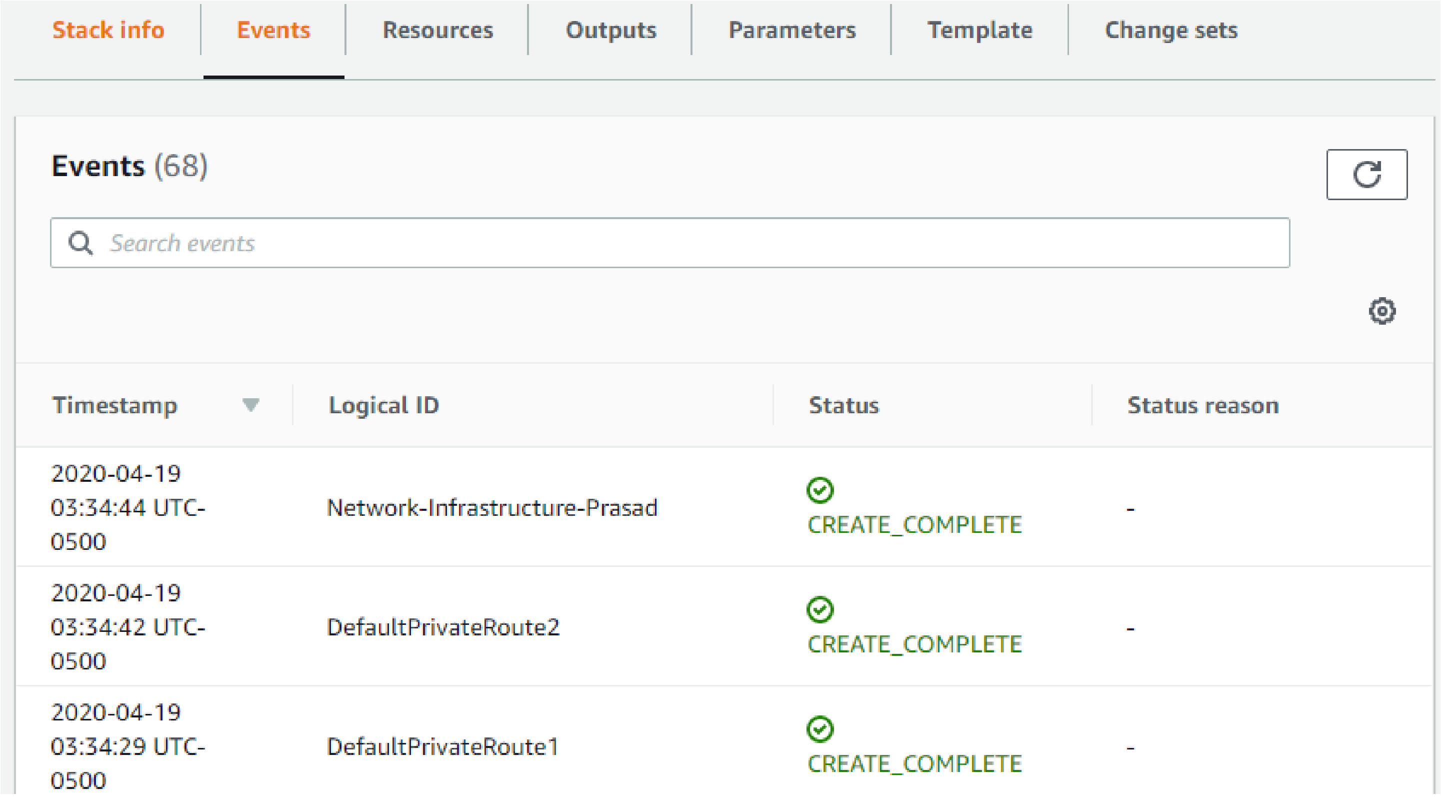
Search events

| Timestamp | Logical ID | Status | Status reason |
|---|---|---|---|
| 2020-04-19 03:34:44 UTC-0500 | Network-Infrastructure-Prasad | ✓ CREATE_COMPLETE | - |
| 2020-04-19 03:34:42 UTC-0500 | DefaultPrivateRoute2 | ✓ CREATE_COMPLETE | - |

Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

You’ve now successfully deployed a CloudFormation template.

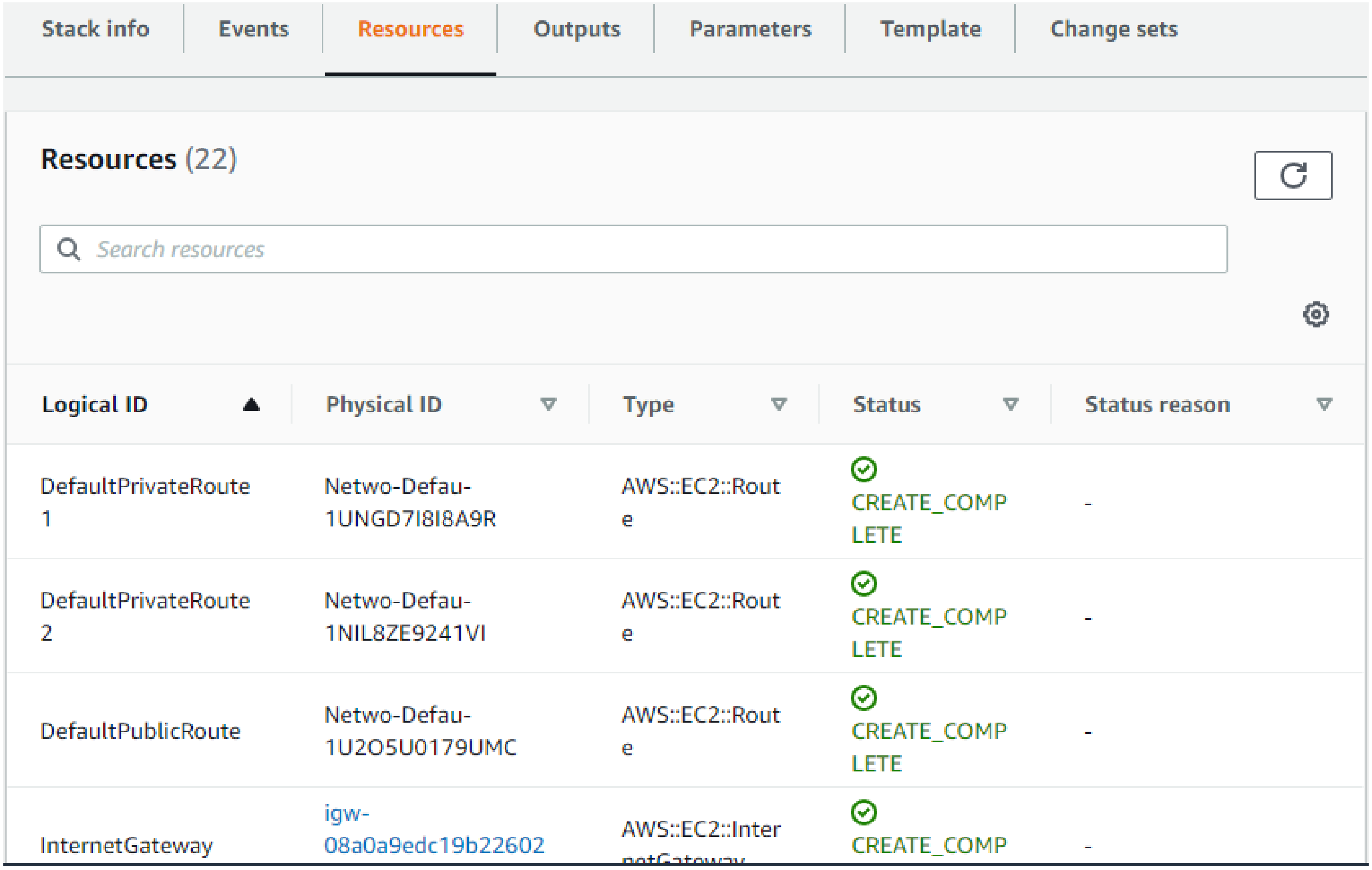
Click on EVENTS and observe the series of tasks executed by the CloudFormation template. For example, first Route Tables and Subnets were created then the Subnet Associations was done. Elastic IP addresses creation and assignment was done just after completion of InternetGatewayAttachment task due to DependsOn attribute etc.



Events (68)

| Timestamp | Logical ID | Status | Status reason |
|---|---|---|---|
| 2020-04-19 03:34:44 UTC-0500 | Network-Infrastructure-Prasad | ✓ CREATE_COMPLETE | - |
| 2020-04-19 03:34:42 UTC-0500 | DefaultPrivateRoute2 | ✓ CREATE_COMPLETE | - |
| 2020-04-19 03:34:29 UTC-0500 | DefaultPrivateRoute1 | ✓ CREATE_COMPLETE | - |

Now click on Resources, you’ll now observe that the Resources that you’ve mentioned in the CloudFormation template’s resources field are now configured and available for you automatically. We could have configured all these resources manually by creating custom VPC and configuring rest of the resources but this is a time consuming and tedious process, takes lot of efforts and manual intervention and chances of human error is more in-case of manual deployment. CloudFormation helped to automate your deployment and save time 😊

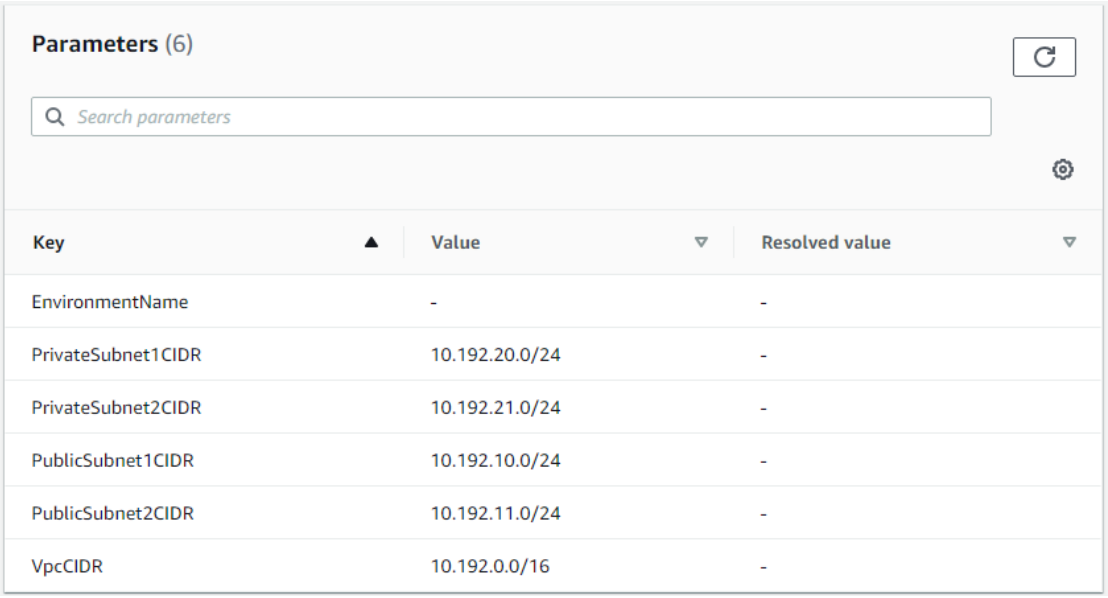


Resources (22)

| Logical ID ▲ | Physical ID ▽ | Type ▽ | Status ▽ | Status reason ▽ |
|---|---|---|---|---|
| DefaultPrivateRoute1 | Netwo-Defau-1UNGD7I8I8A9R | AWS::EC2::Route | ✓ CREATE_COMPLETE | - |
| DefaultPrivateRoute2 | Netwo-Defau-1NIL8ZE9241VI | AWS::EC2::Route | ✓ CREATE_COMPLETE | - |
| DefaultPublicRoute | Netwo-Defau-1U2O5U0179UMC | AWS::EC2::Route | ✓ CREATE_COMPLETE | - |
| InternetGateway | igw-08a0a9edc19b22602 | AWS::EC2::InternetGateway | ✓ CREATE_COMP | - |

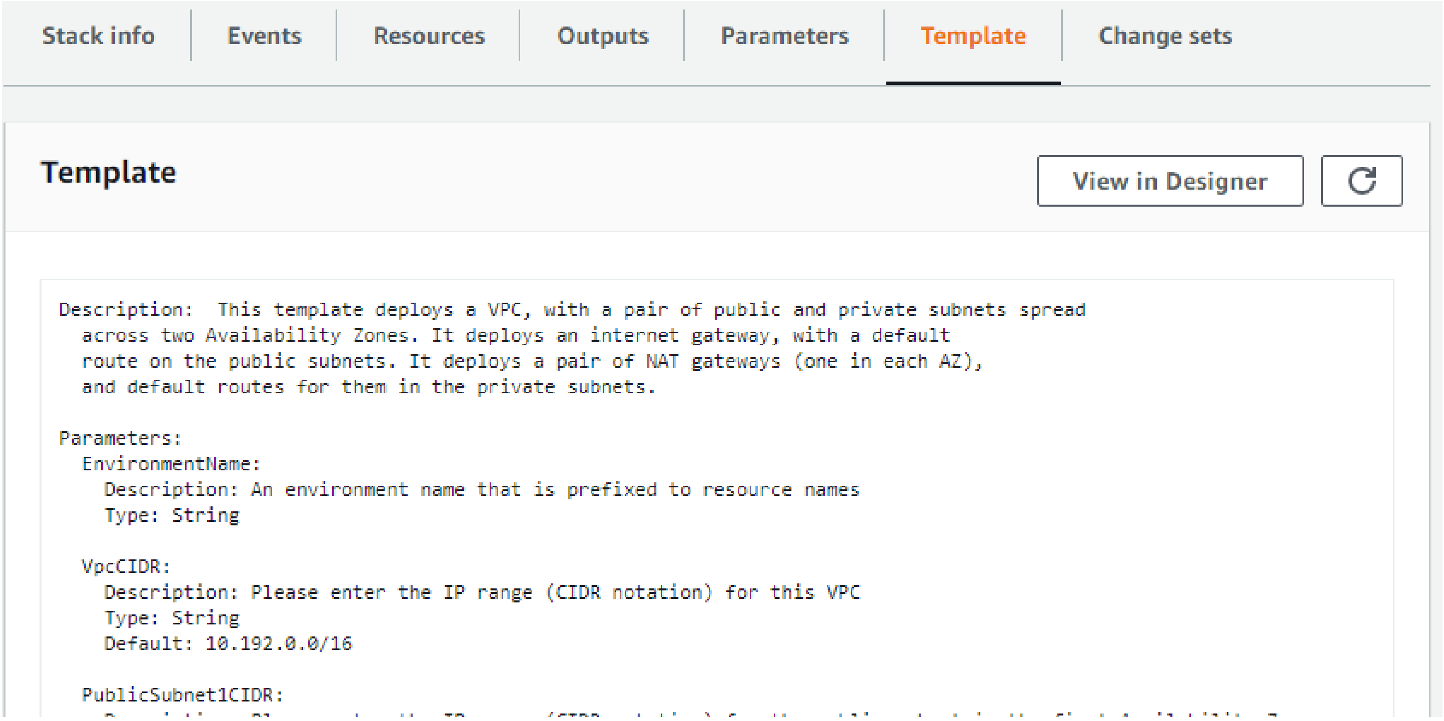
Now click on Outputs. You'll notice that the resources that we've mentioned in CloudFormation template's Output field. These resources can be used by another CloudFormation template. (Eg. If you want to deploy an application design on existing network infrastructure which was deployed by a network template then you can use this Output Resources).

Now click on Parameters, you'll notice the resources that we've specified in the Parameter filed of CloudFormation template. Since we kept all the default values of Parameters, you'll see the below values for each Parameter.



| Key | Value | Resolved value |
|---|---|---|
| EnvironmentName | - | - |
| PrivateSubnet1CIDR | 10.192.20.0/24 | - |
| PrivateSubnet2CIDR | 10.192.21.0/24 | - |
| PublicSubnet1CIDR | 10.192.10.0/24 | - |
| PublicSubnet2CIDR | 10.192.11.0/24 | - |
| VpcCIDR | 10.192.0.0/16 | - |

Now Click on Template, you'll see the complete YAML template.

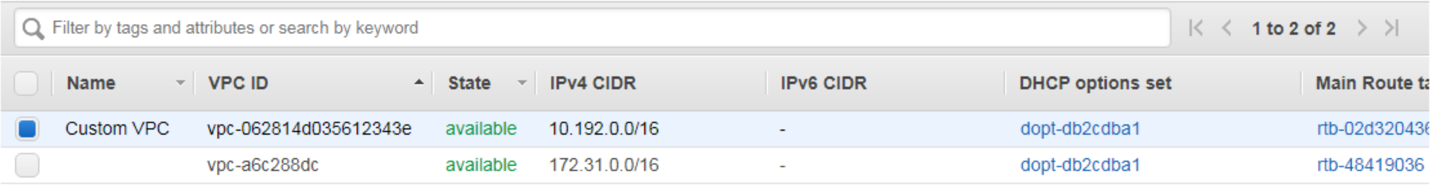


```
Description:  This template deploys a VPC, with a pair of public and private subnets spread
  across two Availability Zones. It deploys an internet gateway, with a default
  route on the public subnets. It deploys a pair of NAT gateways (one in each AZ),
  and default routes for them in the private subnets.

Parameters:
  EnvironmentName:
    Description: An environment name that is prefixed to resource names
    Type: String

  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for this VPC
    Type: String
    Default: 10.192.0.0/16

  PublicSubnet1CIDR:
```

## Part 3: Verifying the Resources.

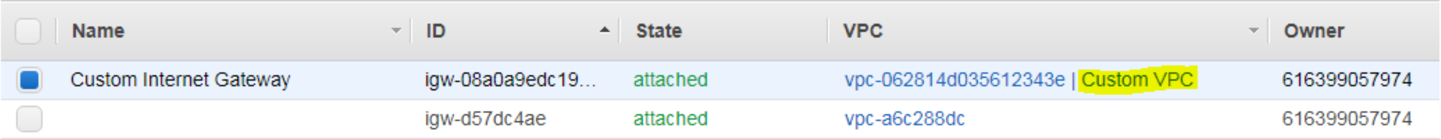
Let’s now very all the resources one-by-one with respect to the CloudFormation template.

Navigate to the VPC service, you’ll notice a Custom VPC is deployed for you with the CIDR of 10.192.0.0/16.



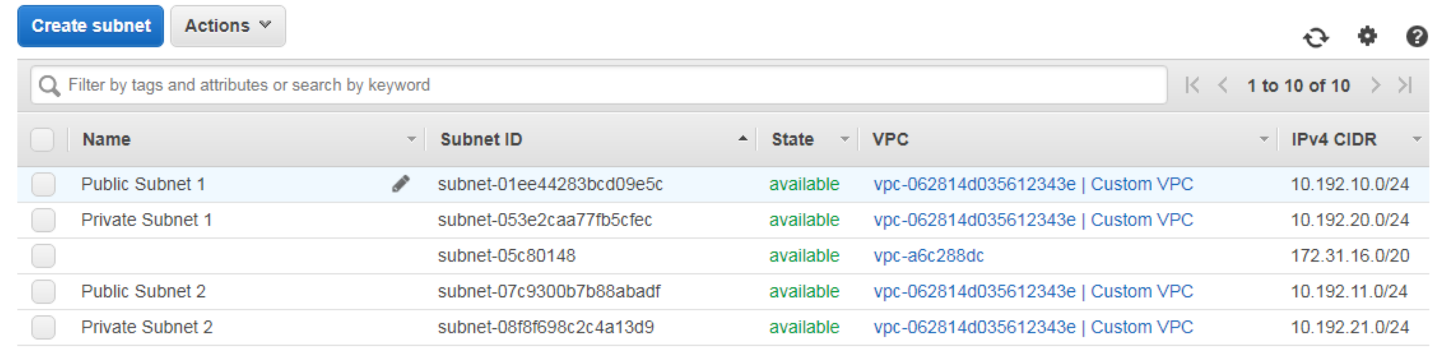
| ☐ | Name | VPC ID | State | IPv4 CIDR | IPv6 CIDR | DHCP options set | Main Route ta |
|---|---|---|---|---|---|---|---|
| ■ | Custom VPC | vpc-062814d035612343e | available | 10.192.0.0/16 | - | dopt-db2cdba1 | rtb-02d320436 |
| ☐ | | vpc-a6c288dc | available | 172.31.0.0/16 | - | dopt-db2cdba1 | rtb-48419036 |

Click on Internet Gateway, you’ll notice that a Custom Internet Gateway is created for you and it has been auto attached to the Custom VPC.



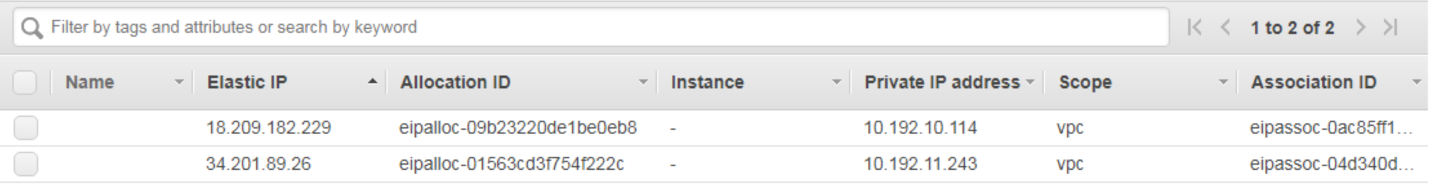
| ☐ | Name | ID | State | VPC | Owner |
|---|---|---|---|---|---|
| ■ | Custom Internet Gateway | igw-08a0a9edc19... | attached | vpc-062814d035612343e \| Custom VPC | 616399057974 |
| ☐ | | igw-d57dc4ae | attached | vpc-a6c288dc | 616399057974 |

Now click on Subnets, you’ll see two Private Subnets and two Public Subnets are created for you in two different Availability Zones along with auto CIDR assignments.



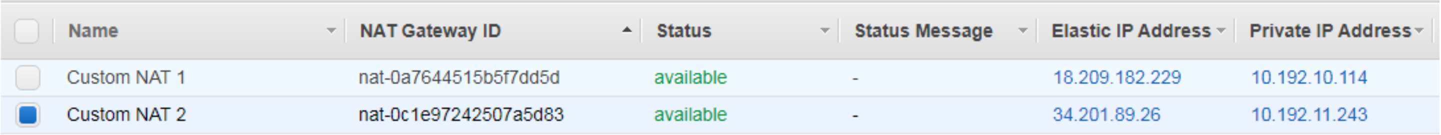
| ☐ | Name | Subnet ID | State | VPC | IPv4 CIDR |
|---|---|---|---|---|---|
| ☐ | Public Subnet 1 | subnet-01ee44283bcd09e5c | available | vpc-062814d035612343e \| Custom VPC | 10.192.10.0/24 |
| ☐ | Private Subnet 1 | subnet-053e2caa77fb5cfec | available | vpc-062814d035612343e \| Custom VPC | 10.192.20.0/24 |
| ☐ | | subnet-05c80148 | available | vpc-a6c288dc | 172.31.16.0/20 |
| ☐ | Public Subnet 2 | subnet-07c9300b7b88abadf | available | vpc-062814d035612343e \| Custom VPC | 10.192.11.0/24 |
| ☐ | Private Subnet 2 | subnet-08f8f698c2c4a13d9 | available | vpc-062814d035612343e \| Custom VPC | 10.192.21.0/24 |

Now click on Elastic IP, you’ll observe that two Elastic IPs have been borrowed for NAT Gateways.



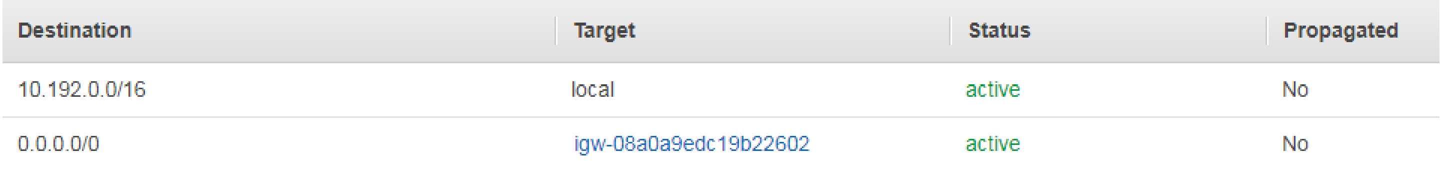
| ☐ | Name | Elastic IP | Allocation ID | Instance | Private IP address | Scope | Association ID |
|---|---|---|---|---|---|---|---|
| ☐ | | 18.209.182.229 | eipalloc-09b23220de1be0eb8 | - | 10.192.10.114 | vpc | eipassoc-0ac85ff1... |
| ☐ | | 34.201.89.26 | eipalloc-01563cd3f754f222c | - | 10.192.11.243 | vpc | eipassoc-04d340d... |

Click on NAT Gateways, you'll observe the deployment of two NAT Gateways in two different Public Subnets with the allocation of Elastic IPs.

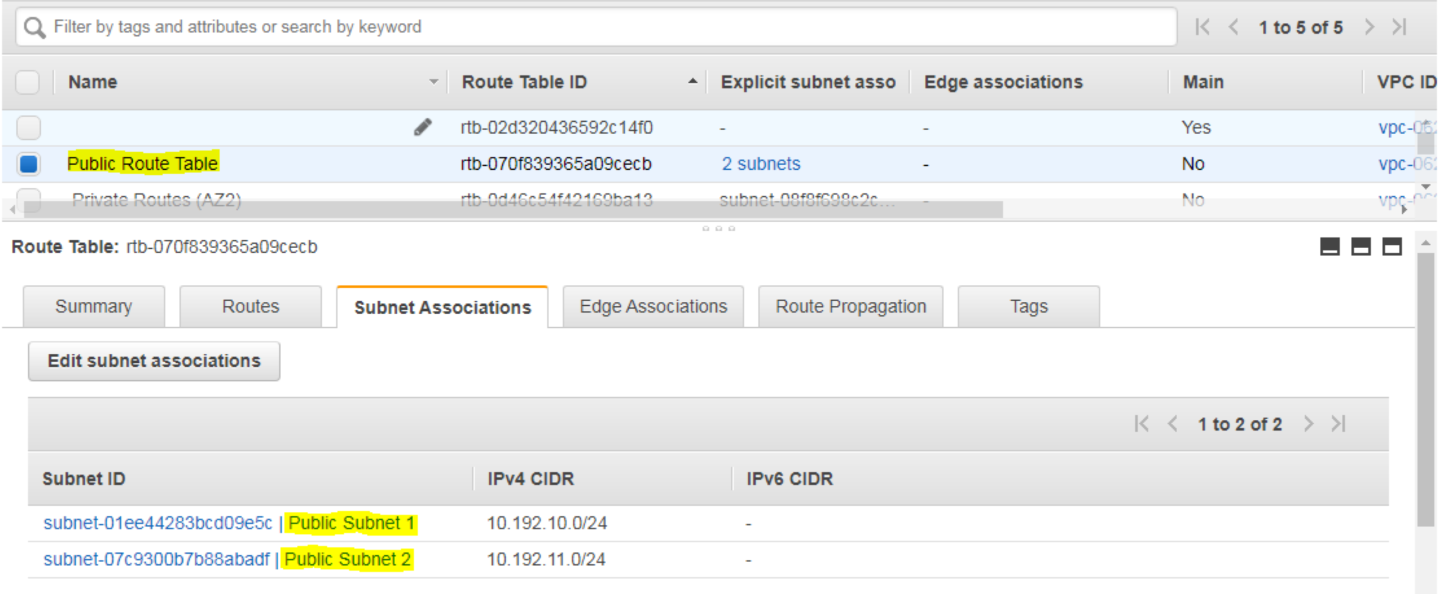


| ☐ | Name | NAT Gateway ID | Status | Status Message | Elastic IP Address | Private IP Address |
|---|---|---|---|---|---|---|
| ☐ | Custom NAT 1 | nat-0a7644515b5f7dd5d | available | - | 18.209.182.229 | 10.192.10.114 |
| ■ | Custom NAT 2 | nat-0c1e97242507a5d83 | available | - | 34.201.89.26 | 10.192.11.243 |

Now click on Route Tables. A single Route Table has been deployed for both the Public Subnets. If you look at the routes, traffic to the destination 10.192.0.0/16 will be local and any unknown traffic will be routed to the Custom Internet Gateway which was deployed by CloudFormation.



| Destination | Target | Status | Propagated |
|---|---|---|---|
| 10.192.0.0/16 | local | active | No |
| 0.0.0.0/0 | igw-08a0a9edc19b22602 | active | No |



Filter by tags and attributes or search by keyword

1 to 5 of 5

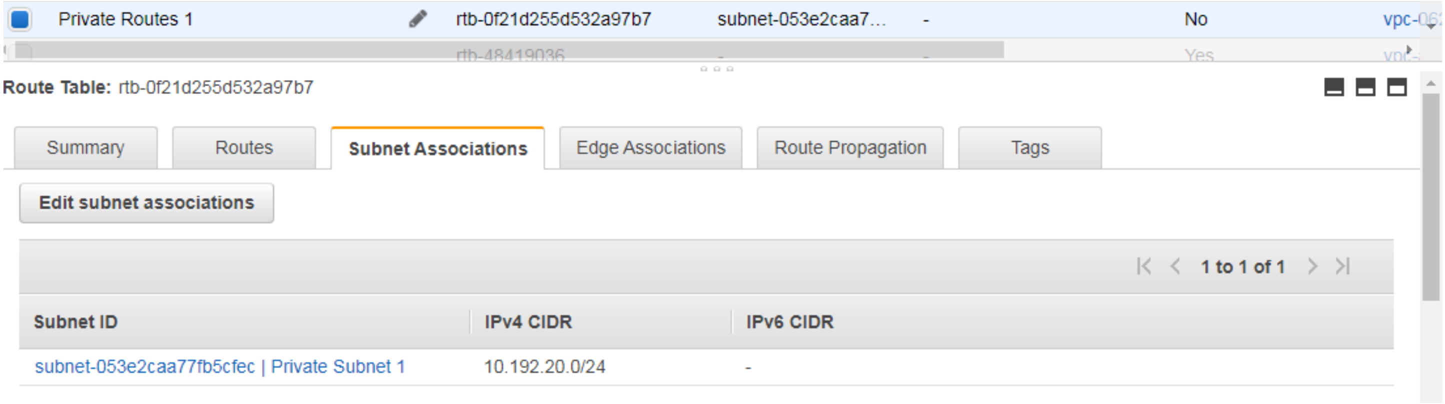
| ☐ | Name | Route Table ID | Explicit subnet asso | Edge associations | Main | VPC ID |
|---|---|---|---|---|---|---|
| ☐ | | rtb-02d320436592c14f0 | - | - | Yes | vpc-06 |
| ■ | Public Route Table | rtb-070f839365a09cecb | 2 subnets | - | No | vpc-06 |
| ☐ | Private Routes (AZ2) | rtb-0d46c54f42169ba13 | subnet-08f8f698c2c... | - | No | vpc-0 |

Route Table: rtb-070f839365a09cecb

1 to 2 of 2

| Subnet ID | IPv4 CIDR | IPv6 CIDR |
|---|---|---|
| subnet-01ee44283bcd09e5c \| Public Subnet 1 | 10.192.10.0/24 | - |
| subnet-07c9300b7b88abadf \| Public Subnet 2 | 10.192.11.0/24 | - |

Similarly, look at the Private Route Table 1 with its subnet association and routes. Traffic to the destination 10.192.0.0/16 will be local and any unknown traffic will be routed to the Custom NAT Gateway 1 which was deployed by CloudFormation.

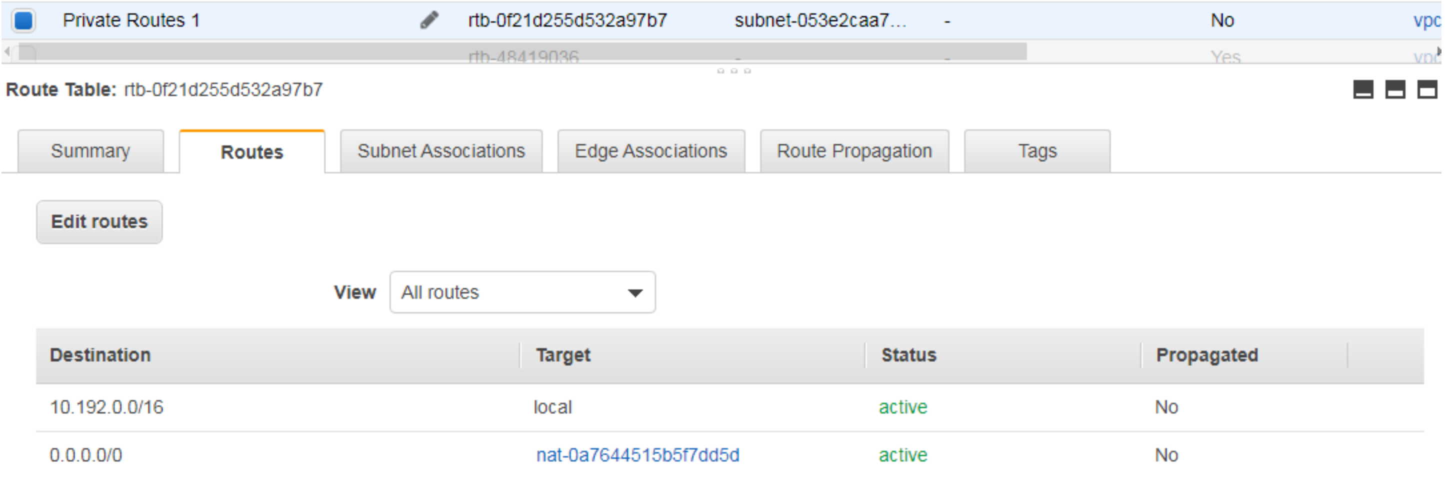


Private Routes 1 rtb-0f21d255d532a97b7 subnet-053e2caa7... - No vpc-06

rtb-48419036 - - Yes vpc-

Route Table: rtb-0f21d255d532a97b7

| Subnet ID | IPv4 CIDR | IPv6 CIDR |
|---|---|---|
| subnet-053e2caa77fb5cfec \| Private Subnet 1 | 10.192.20.0/24 | - |



Private Routes 1 rtb-0f21d255d532a97b7 subnet-053e2caa7... - No vpc

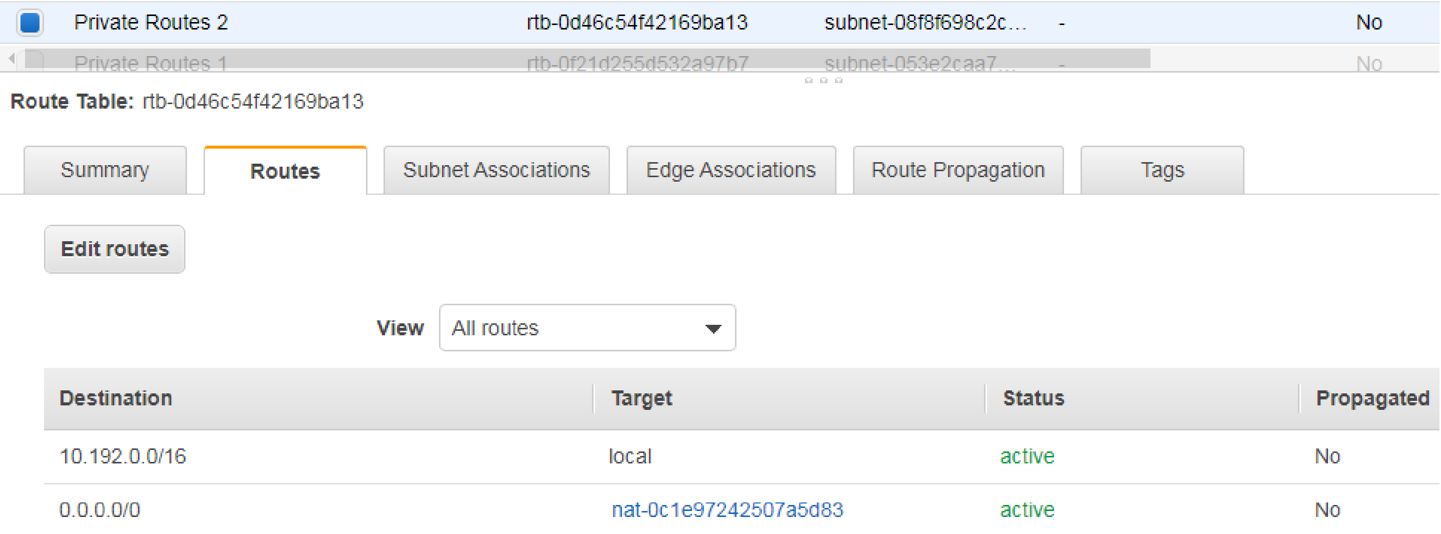
rtb-48419036 - - Yes vpc

Route Table: rtb-0f21d255d532a97b7

View: All routes

| Destination | Target | Status | Propagated |
|---|---|---|---|
| 10.192.0.0/16 | local | active | No |
| 0.0.0.0/0 | nat-0a7644515b5f7dd5d | active | No |

Also, look at the Private Route Table 2 with its subnet association and routes. Traffic to the destination 10.192.0.0/16 will be local and any unknown traffic will be routed to the Custom NAT Gateway 2 which was deployed by CloudFormation.

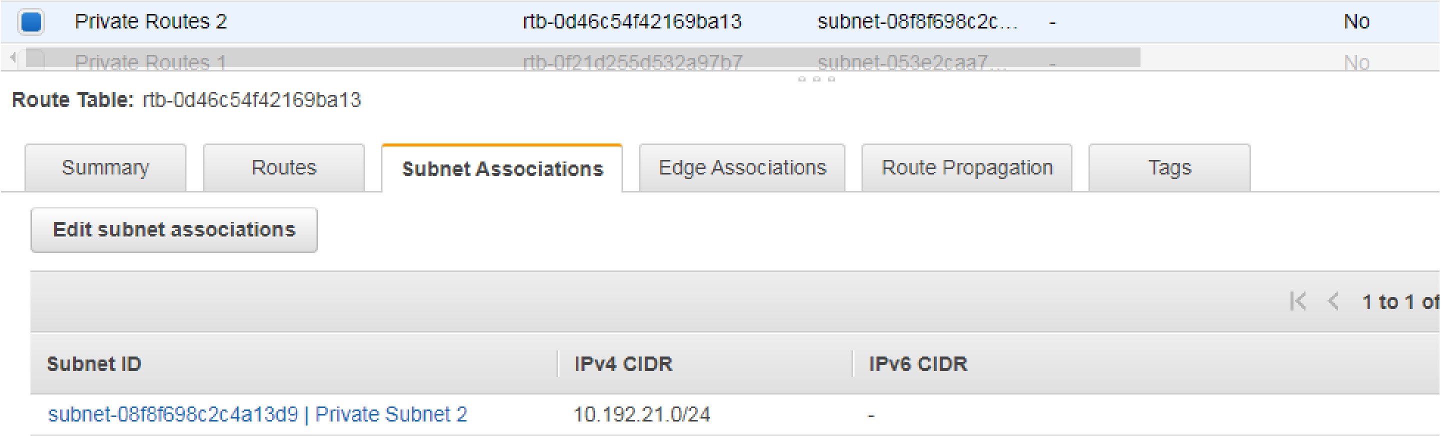


| | | | | |
|---|---|---|---|---|
| ■ Private Routes 2 | rtb-0d46c54f42169ba13 | subnet-08f8f698c2c… | - | No |
| □ Private Routes 1 | rtb-0f21d255d532a97b7 | subnet-053e2caa7… | - | No |

**Route Table:** rtb-0d46c54f42169ba13

**View** All routes

| Destination | Target | Status | Propagated |
|---|---|---|---|
| 10.192.0.0/16 | local | active | No |
| 0.0.0.0/0 | nat-0c1e97242507a5d83 | active | No |

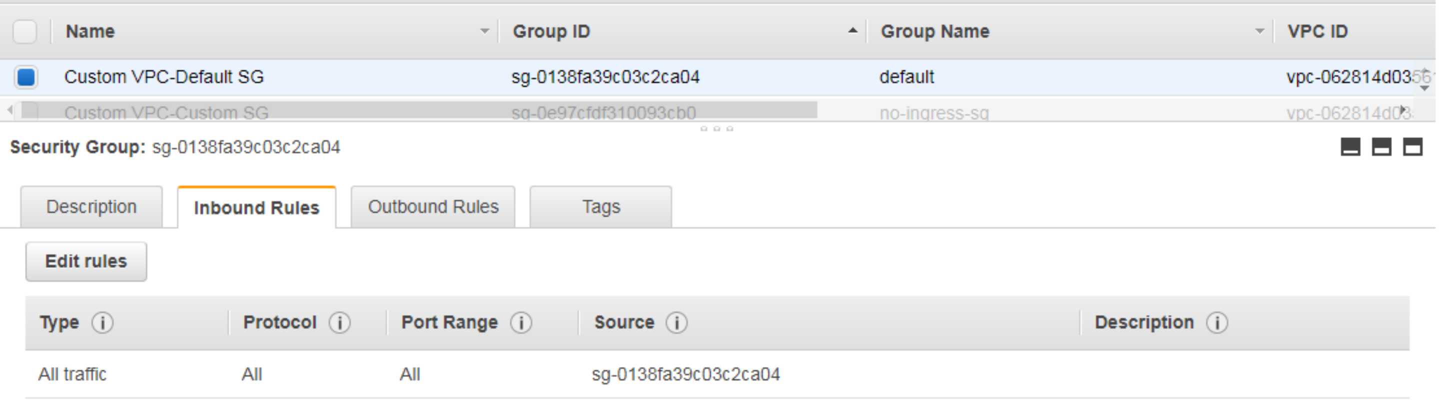


| | | | | |
|---|---|---|---|---|
| ■ Private Routes 2 | rtb-0d46c54f42169ba13 | subnet-08f8f698c2c… | - | No |
| □ Private Routes 1 | rtb-0f21d255d532a97b7 | subnet-053e2caa7… | - | No |

**Route Table:** rtb-0d46c54f42169ba13

| Subnet ID | IPv4 CIDR | IPv6 CIDR |
|---|---|---|
| subnet-08f8f698c2c4a13d9 \| Private Subnet 2 | 10.192.21.0/24 | - |

Finally, navigate to Security Groups. You'll notice that a Default Security group is auto created since SECURITY GROUPS are part of VPCs. Verify the Inbound and Outbound rules of this SG.

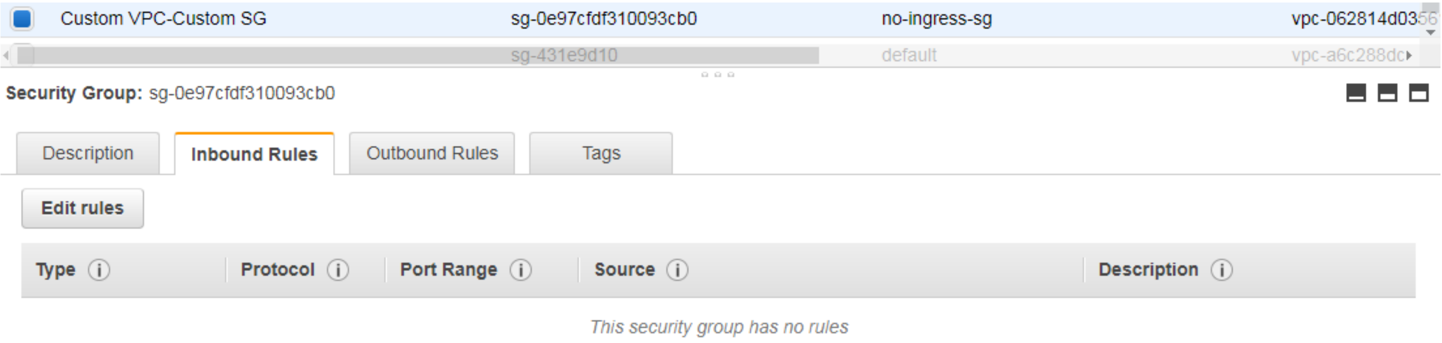


| | Name | Group ID | Group Name | VPC ID |
|---|---|---|---|---|
| ■ | Custom VPC-Default SG | sg-0138fa39c03c2ca04 | default | vpc-062814d03 |
| □ | Custom VPC-Custom SG | sg-0e97cfdf310093cb0 | no-ingress-sg | vpc-062814d0 |

**Security Group:** sg-0138fa39c03c2ca04

| Type | Protocol | Port Range | Source | Description |
|---|---|---|---|---|
| All traffic | All | All | sg-0138fa39c03c2ca04 | |

You'll also notice that a Custom Security Group in Custom VPC has been deployed by a CloudFormation template. Verify Inbound and Outbound rules of this Custom SG.



| | | | |
|---|---|---|---|
| Custom VPC-Custom SG | sg-0e97cfdf310093cb0 | no-ingress-sg | vpc-062814d0356 |
| | sg-431e9d10 | default | vpc-a6c288dc |

**Security Group:** sg-0e97cfdf310093cb0

| Type | Protocol | Port Range | Source | Description |
|---|---|---|---|---|

*This security group has no rules*

Note that for a Default Security group, Inbound traffic from the same Security Group will always be allowed and all the outbound traffic to the Internet will always be allowed by default.

For a Custom Security group, there is no Inbound traffic stated and outbound traffic to the Internet is also allowed by default.

**Recommendations:**

It is always a best practice to configure all the resources in a CUSTOM VPC due to security purpose as default VPC CIDR is well known to everyone and chances of security breach is more in Default VPC.

This completes the entire deployment of Network Infrastructure on AWS using CloudFormation Template. For the next upcoming labs, we will be using the same Network Infrastructure.

I hope you have now gained better understanding of CloudFormation Automation.

For Questions, contact me on pbhavsar@smu.edu.