



# **Amazon Web Services**

## **Build VPC With AWS CloudFormation**

---

*May 2018*

## Table of Contents

Overview .....	3
Explore Initial VPC template .....	3
Create Stack .....	4
Lab Objective:.....	8
References:.....	9
Update Stack and Solution:.....	10
Appendix:.....	13

## Overview

This lab will walk the user through using the AWS CloudFormation to create a VPC with public and private subnets, describe each of the objects created by the AWS CloudFormation, and launch VPC with the public and private VPC subnets, RouteTable, Elastic IP NAT Gateway, and S3 bucket.

The following is a high-level overview of this lab:

- Explore the initial AWS CloudFormation template
- Explore the different VPC objects and what they mean
- Launch AWS CloudFormation by creating Stack from Console.
- Export VPC ID, NAT Gateway ID and S3 bucket URL to output tab

The lab will provide an initial template for users to explore . after creating VPC stack from an initial template, users need to complete provided objective to achieve the final solution.

**Note:** Screenshots are provided to guide you through the steps in the lab. The elements that you will create (e.g. VPC, NAT Gateway, EIP) will be unique to your account, so things such as VPC ID that you see in the console will not necessarily mirror what's seen in the screenshot.

## Explore Initial VPC template

Please browse the initial AWS CloudFormation Template file, You can use any text editor to explore the different elements of VPC mentioned in the template:

*Lab\_Initial\_CloudFormation\_Module\_General\_ImmersionDay.json* (Follow instruction in appendix section to get template)

You will notice following resources in Initial AWS CloudFormation Template:

- VPC
- Internet Gateway
- S3 bucket
- Two public subnets with corresponding route tables
- Two private subnets with corresponding route tables
- Two Elastic IP
- Two NAT Gateway

## Create Stack

Log into the **AWS Console**, and click on **CloudFormation** and below screen will open:

**Create Stack** ▾ Actions ▾ Design template

Filter: Active ▾ By Stack Name

### Create a stack

AWS CloudFormation allows you to quickly and easily deploy your infrastructure resources and applications on AWS. You can use one of the templates we provide to get started quickly with applications like WordPress or Drupal, one of the many sample templates or create your own template.

You do not currently have any stacks. Choose **Create new stack** below to create a new AWS CloudFormation stack.

**Create new stack**

### Create a StackSet

A StackSet is a container for AWS CloudFormation stacks that lets you provision stacks across AWS accounts and regions by using a single AWS CloudFormation template.

**Create new StackSet**

Now click on **Create new stack** and browse your initial template to against choose a template option :

### Create stack

#### Select Template

Specify Details  
Options  
Review

#### Select Template

Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.

**Design a template** Use AWS CloudFormation Designer to create or modify an existing template. [Learn more.](#)

Design template

**Choose a template** A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)

☐ Select a sample template

☒ Upload a template to Amazon S3

Browse... Lab\_Initial\_CloudFormation\_Module\_General\_ImmersionDay.yaml

☐ Specify an Amazon S3 template URL

Cancel

Next

## Build VPC with AWS CloudFormation Lab

Click **Next** and give stack name. Make sure your stack name should be unique to your account. Leave all other option as default

### Create stack

Select Template

Specify Details

Options

Review

#### Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name demo-vpc

#### Parameters

psharedacidr 10.20.0.0/22

psharedbcidr 10.20.4.0/22

vpccidr 10.20.0.0/16

Cancel

Previous

Next

Click **Next**, here you can define a tag for the stack, IAM Role and other advance option like termination protection and rollback trigger. For this lab we will leave this as it and click to **Next** again, where you will have the opportunity to review your stack settings:

### Create stack

Select Template

Specify Details

Options

Review

#### Review

##### Template

Template URL [https://s3-external-1.amazonaws.com/cf-templates-ta7yfdqg9l4-us-east-1/2018144UNP-Lab\\_Initial\\_CloudFormation\\_Module\\_General\\_ImmersionDay.yaml](https://s3-external-1.amazonaws.com/cf-templates-ta7yfdqg9l4-us-east-1/2018144UNP-Lab_Initial_CloudFormation_Module_General_ImmersionDay.yaml)  
Description  
Estimate cost Cost

##### Details

Stack name: demo-vpc

psharedacidr 10.20.0.0/22

psharedbcidr 10.20.4.0/22

vpccidr 10.20.0.0/16

##### Options

###### Tags

No tags provided

###### Rollback Triggers

No monitoring time provided

No rollback triggers provided

###### Advanced

###### Notification

Termination Protection Disabled

Timeout none

Rollback on failure Yes

[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

Cancel

Previous

Create

## Build VPC with AWS CloudFormation Lab

Now Click on **Create** and you will notice your stack creation started with status **CREATE\_IN\_PROGRESS**. Explore the **Events** tab where you can see the progress as your stack get created.

CloudFormation

Stacks

Create Stack

Actions

Design template

Filter: Active

By Stack Name

Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/> demo-vpc	2018-05-24 15:53:58 UTC-0700	CREATE_IN_PROGRESS	

Overview

Outputs

Resources

Events

Template

Parameters

Tags

Stack Policy

Change Sets

Rollback Triggers

Time	Type	Resource	Message
15:54:43 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::EIP	EIPNatGWA
15:54:43 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::Route	RouteDefaultPublic
15:54:41 UTC-0700	CREATE_COMPLETE	AWS::EC2::VPCGatewayAttachment	GatewayAttach
15:54:41 UTC-0700	CREATE_COMPLETE	AWS::EC2::Subnet	SubnetPublicSharedA
15:54:41 UTC-0700	CREATE_COMPLETE	AWS::EC2::Subnet	SubnetPublicSharedB
15:54:35 UTC-0700	CREATE_COMPLETE	AWS::S3::BucketPolicy	BucketPolicyApp
15:54:35 UTC-0700	CREATE_IN_PROGRESS	AWS::S3::BucketPolicy	BucketPolicyApp
15:54:30 UTC-0700	CREATE_IN_PROGRESS	AWS::S3::BucketPolicy	BucketPolicyApp
15:54:28 UTC-0700	CREATE_COMPLETE	AWS::EC2::RouteTable	RouteTablePrivateB
15:54:27 UTC-0700	CREATE_COMPLETE	AWS::S3::Bucket	S3AppBucket
15:54:27 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::RouteTable	RouteTablePrivateB
15:54:27 UTC-0700	CREATE_COMPLETE	AWS::EC2::RouteTable	RouteTablePrivateA
15:54:26 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::RouteTable	RouteTablePrivateB
15:54:26 UTC-0700	CREATE_COMPLETE	AWS::EC2::RouteTable	RouteTablePublic
15:54:26 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::RouteTable	RouteTablePrivateA
15:54:25 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::VPCGatewayAttachment	GatewayAttach
15:54:25 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::RouteTable	RouteTablePrivateA
15:54:25 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::VPCGatewayAttachment	GatewayAttach
15:54:25 UTC-0700	CREATE_IN_PROGRESS	AWS::EC2::Subnet	SubnetPublicSharedA

While you are waiting to explore all other tabs like **Template** tab to review your template and **Parameters** tab to see parameter value. You will also notice that **Outputs** tab is empty and you are going to modify your template to show values in Outputs tab.

Overview	Outputs	Resources	Events	Template	Parameters	Tags	Stack Policy	Change Sets	Rollback Triggers
Key		Value			Resolved Value				
psharedacidr		10.20.0.0/22							
psharedbcidr		10.20.4.0/22							
vpccidr		10.20.0.0/16							

Overview

Outputs

Resources

Events

Template

Parameters

Tags

Stack Policy

Change Sets

Rollback Triggers

AWSTemplateFormatVersion: '2010-09-09'

Parameters:

vpccidr:

Type: String

Default: 10.20.0.0/16

psharedacidr:

Type: String

Default: 10.20.0.0/22

psharedbcidr:

Type: String

Default: 10.20.4.0/22

View/Edit template in Designer

## Build VPC with AWS CloudFormation Lab

Once stack status changes to **CREATE\_COMPLETE**, you can visit **Resources** tab to see all the resources got created by this AWS CloudFormation template.

The screenshot shows the AWS CloudFormation console. At the top, there's a navigation bar with 'CloudFormation' and 'Stacks'. Below this, there are buttons for 'Create Stack', 'Actions', and 'Design template'. A filter dropdown is set to 'Active', and a search box contains 'By Stack Name'. The main table lists stacks, with 'demo-vpc' selected, showing a status of 'CREATE\_COMPLETE' and a creation time of '2018-05-24 15:53:58 UTC-0700'. Below the stack list, the 'Resources' tab is active, showing a table of resources created by the stack:

Resource Name	Logical ID	Physical ID	Type	Status
RouteTablePrivateB	rtb-38896047		AWS::EC2::RouteTable	CREATE_COMPLETE
RouteTablePublic	rtb-a3b45ddc		AWS::EC2::RouteTable	CREATE_COMPLETE
S3AppBucket	demo-vpc-s3appbucket-1tplko9hqotzl		AWS::S3::Bucket	CREATE_COMPLETE
SubnetPublicSharedA	subnet-c768a8e9		AWS::EC2::Subnet	CREATE_COMPLETE
SubnetPublicSharedB	subnet-48900502		AWS::EC2::Subnet	CREATE_COMPLETE
SubnetRouteTableAssoc...	rtbassoc-1cacf663		AWS::EC2::SubnetRouteTableA...	CREATE_COMPLETE
SubnetRouteTableAssoc...	rtbassoc-a8aef4d7		AWS::EC2::SubnetRouteTableA...	CREATE_COMPLETE
VPC	vpc-95370fee		AWS::EC2::VPC	CREATE_COMPLETE

You can click on Amazon S3 bucket link shown in **Resources** tab and explore the bucket. Also, go to VPC from the console and explore different resources got created from AWS CloudFormation stack.

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with 'VPC Dashboard' and a search filter. The main area has a 'Create NAT Gateway' button and a table of existing NAT Gateways:

Name	NAT Gateway ID	Status	Status Message	Elastic IP Address	Private IP Address	Network Interface
	nat-0f1ac3f564cba...	available	-	34.194.195.112	10.20.2.170	eni-7242e9e5
	nat-07cf630f543cf...	available	-	52.20.206.218	10.20.7.95	eni-9f35f30e

## Lab Objective:

Now you need to modify your template with following objectives:

### Add Parameter Constraint :

- Vpccidr
  - Minimum length should be set to 9
  - Maximum length should be set to 18
  - Allowed pattern should be:  
"`(\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})/(\\d{1,2})))`"
  - Add a constraint description
- Psharedacidr
  - Minimum length should be set to 9
  - Maximum length should be set to 18
  - Allowed pattern should be:  
"`(\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})/(\\d{1,2})))`"
  - Add a constraint description
- Psharedbcidr
  - Minimum length should be set to 9
  - Maximum length should be set to 18
  - Allowed pattern should be:  
"`(\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})\\.((\\d{1,3})/(\\d{1,2})))`"
  - Add a constraint description

### Add delete policy constraint :

- Create a Deletion Policy for your S3 bucket to be Retained at deletion

### Add Outputs section to show value in Output tab:

- Vpc id
  - Create a description of your output
  - Reference your VPC as the value using !Ref
- NATGWA
  - Create a description of your output
  - Reference your NAT gateway A as the value using !Ref
- NATGWB
  - Create a description of your output



- Reference your NAT gateway B as the value using !Ref
- App bucket URL
  - Create a description of your output
  - Reference your S3 bucket URL as the value using !Ref

**Add export values in Outputs section for Cross-Stack Reference:**

- Vpc id
  - Export your vpcid Name as 'sharedinf-vpc'
- App bucket URL
  - Export your appbucketurl Name as 'sharedinf-appbucketurl'

## References:

### Parameters:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html>

### Intrinsic functions:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference.html>

### Outputs and Export:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/outputs-section-structure.html>

### Mappings:

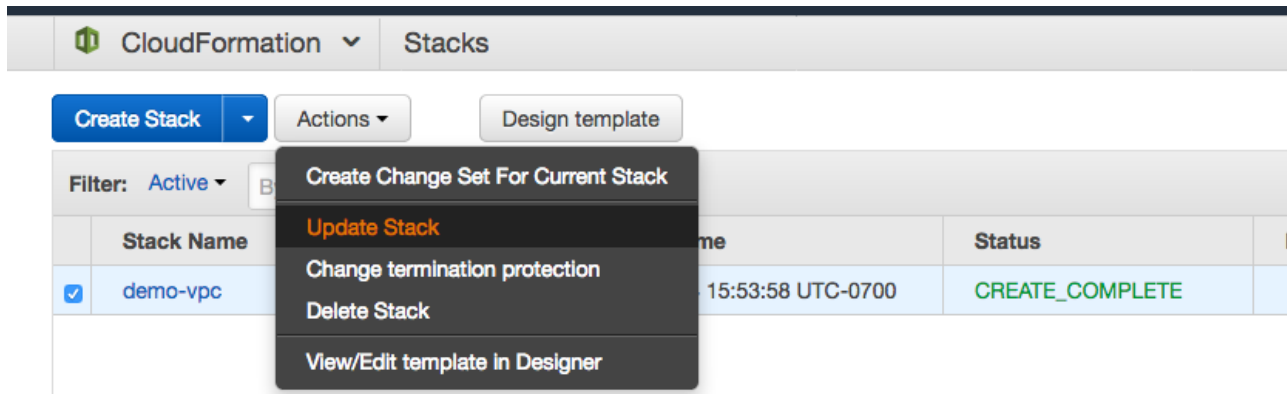
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/mappings-section-structure.html>

### Deletion policy:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-attribute-deletionpolicy.html>

## Update Stack and Solution:

Once you modified your existing template , you can use **Update Stack** option to update your stack. To update select your Stack and click on **Actions drop down and you will find Update Stack** option.



In **Update**, Stack screen select browse your updated template. If you have not figured out a solution yet Follow instruction in appendix section to get template

*Lab\_Solution\_CloudFormation\_Module\_General\_ImmersionDay.json.*

### Update demo-vpc stack

Select Template  
Specify Details  
Options  
Review

#### Select Template

To update an existing stack, provide a template that specifies the changes for the resources and properties that you want to update. AWS CloudFormation updates only the resources that have changed. [Learn more.](#)

**Choose a template** A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)

☐ Use current template

☒ Upload a template to Amazon S3

Lab\_Solution\_CloudFormation\_Module\_General\_ImmersionDay.yaml

☐ Specify an Amazon S3 template URL

## Build VPC with AWS CloudFormation Lab

Now remaining steps are same as you followed in Create stack. Click Next couple of time and you will land up to review summary screen, where you need to click on **Update** button :

[Select Template](#)  
[Specify Details](#)  
[Options](#)  
**Review**

### Review

Review the information that AWS CloudFormation will use to update your stack. If you need to change a value, return to the page that contains the value that you want to change.

---

#### Template

Template URL: [https://s3-external-1.amazonaws.com/cf-templates-ia7yldqg9i4-us-east-1/2018144zmU-Lab\\_Solution\\_CloudFormation\\_Module\\_General\\_ImmersionDay.yaml](https://s3-external-1.amazonaws.com/cf-templates-ia7yldqg9i4-us-east-1/2018144zmU-Lab_Solution_CloudFormation_Module_General_ImmersionDay.yaml)  
Description

---

#### Details

Stack name:	demo-vpc
psharedacidr	10.20.0.0/22
psharedbcidr	10.20.4.0/22
vpccidr	10.20.0.0/16

---

#### Options

##### Tags

No tags provided

##### Rollback Triggers

No monitoring time provided  
No rollback triggers provided

##### Advanced

Notification

---

#### Preview your changes

Based on your input, CloudFormation will change the following resources. For more information, choose [View change set details](#).

No values found

[Cancel](#) [Previous](#) [Update](#)

Now you will find your stack status changed to **UPDATE\_IN\_PROGRESS** and **Events** tab showing the activity performed using update stack.

## Build VPC with AWS CloudFormation Lab

The screenshot shows the AWS CloudFormation console interface. At the top, there are buttons for 'Create Stack', 'Actions', and 'Design template'. Below these is a filter bar with 'Filter: Active' and a search box 'By Stack Name'. The main table lists stacks, with 'demo-vpc' selected and its status 'UPDATE\_IN\_PROGRESS'. Below the table, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', 'Change Sets', and 'Rollback Triggers'. The 'Events' tab is currently active, showing a list of events for the 'demo-vpc' stack.

Stack Name	Created Time	Status	Description
demo-vpc	2018-05-24 15:53:58 UTC-0700	UPDATE_IN_PROGRESS	

Filter by: Status	Search events			
2018-05-24	Status	Type	Logical ID	Status Reason
16:32:13 UTC-0700	UPDATE_COMPLETE	AWS::CloudFormation::Stack	demo-vpc	
16:32:12 UTC-0700	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	demo-vpc	
16:32:06 UTC-0700	UPDATE_COMPLETE	AWS::S3::Bucket	S3AppBucket	
16:32:01 UTC-0700	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	demo-vpc	User Initiated
15:57:42 UTC-0700	CREATE_COMPLETE	AWS::CloudFormation::Stack	demo-vpc	
15:57:40 UTC-0700	CREATE_COMPLETE	AWS::EC2::Route	RouteDefaultPrivateA	

Once stack status changed to UPDATE\_COMPLETE status, you can browse to **Outputs** tab and find out our changes has reflected now Outputs tab has four values compare to earlier it was empty :

The screenshot shows the AWS CloudFormation console interface. At the top, there are buttons for 'Create Stack', 'Actions', and 'Design template'. Below these is a filter bar with 'Filter: Active' and a search box 'By Stack Name'. The main table lists stacks, with 'demo-vpc' selected and its status 'UPDATE\_COMPLETE'. Below the table, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', 'Change Sets', and 'Rollback Triggers'. The 'Outputs' tab is currently active, showing a list of outputs for the 'demo-vpc' stack.

Stack Name	Created Time	Status	Description
demo-vpc	2018-05-24 15:53:58 UTC-0700	UPDATE_COMPLETE	

Overview	Outputs	Resources	Events	Template	Parameters	Tags	Stack Policy	Change Sets	Rollback Triggers
Key	Value	Description	Export Name						
appbucketurl	<a href="http://demo-vpc-s3appbucket-1tpko9hqotzl.s3-website-us-east-1.amazonaws.com">http://demo-vpc-s3appbucket-1tpko9hqotzl.s3-website-us-east-1.amazonaws.com</a>	Shared Infrastructure App Bucket	sharedinf-appbucketurl						
vpcid	vpc-95370fee	ID of Shared Infrastructure VPC	sharedinf-vpcid						
natgatewayaid	nat-0f1ac3f564cbad943	ID of NAT Gateway A							
natgatewaybid	nat-07cf630f543cf33f8	ID of NAT Gateway B							

Also, click on **CloudFormation** icon on the right top corner of the screen and select **Exports** option, you will find two exported value shown in here which can be utilized for cross-stack reference.

## Build VPC with AWS CloudFormation Lab

CloudFormation ^ Exports			
Stacks Exports StackSets			
	Showing 2 exports		
	Export Value	Stack Name	Stack ID
	sharedinf-appbucketurl	demo-vpc	arn:aws:cloudformation:us-east-1:789211807855:stack/demo-vpc/58064a90-...
	sharedinf-vpcid	demo-vpc	arn:aws:cloudformation:us-east-1:789211807855:stack/demo-vpc/58064a90-...

To create a cross-stack reference, use the **Export** output field to flag the value of a resource-output for export. Then, use the **Fn:: ImportValue** intrinsic function to import the value.

## Appendix:

### Initial AWS CloudFormation Template for lab exercise:

Create a file `Lab_Initial_CloudFormation_Module_General_ImmersionDay.yaml` and copy paste following code :

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  vpccidr:
    Type: String
    Default: 10.20.0.0/16
  psharedacidr:
    Type: String
    Default: 10.20.0.0/22
  psharedbcidr:
    Type: String
    Default: 10.20.4.0/22

Resources:
  VPC:
    Type: "AWS::EC2::VPC"
    Properties:
      CidrBlock: !Ref vpccidr
  IGW:
    Type: "AWS::EC2::InternetGateway"
  S3AppBucket:
    Type: "AWS::S3::Bucket"
    Properties:
      AccessControl: PublicRead
      WebsiteConfiguration:
        ErrorDocument: index.html
        IndexDocument: index.html
  BucketPolicyApp:
```

```
Type: "AWS::S3::BucketPolicy"
Properties:
  Bucket: !Ref S3AppBucket
  PolicyDocument:
    Statement:
      -
        Sid: "ABC123"
        Action:
          - "s3:GetObject"
        Effect: Allow
        Resource: !Join ["", ["arn:aws:s3:::", !Ref S3AppBucket, "/*"]]
        Principal:
          AWS:
            - "*"

```

GatewayAttach:

```
Type: "AWS::EC2::VPCGatewayAttachment"
Properties:
  InternetGatewayId: !Ref IGW
  VpcId: !Ref VPC

```

SubnetPublicSharedA:

```
Type: "AWS::EC2::Subnet"
Properties:
  AvailabilityZone: !Select [0, !GetAZs ]
  CidrBlock: !Ref psharedacidr
  MapPublicIpOnLaunch: true
  VpcId: !Ref VPC

```

SubnetPublicSharedB:

```
Type: "AWS::EC2::Subnet"
Properties:
  AvailabilityZone: !Select [1, !GetAZs ]
  CidrBlock: !Ref psharedbcidr
  MapPublicIpOnLaunch: true
  VpcId: !Ref VPC

```

SubnetRouteTableAssociatePublicA:

```
Type: "AWS::EC2::SubnetRouteTableAssociation"
Properties:
  RouteTableId: !Ref RouteTablePublic
  SubnetId: !Ref SubnetPublicSharedA

```

SubnetRouteTableAssociatePublicB:

```
Type: "AWS::EC2::SubnetRouteTableAssociation"
Properties:
  RouteTableId: !Ref RouteTablePublic
  SubnetId: !Ref SubnetPublicSharedB

```

RouteDefaultPublic:

```
Type: "AWS::EC2::Route"
DependsOn: GatewayAttach
Properties:
  DestinationCidrBlock: 0.0.0.0/0

```

```
    GatewayId: !Ref IGW
    RouteTableId: !Ref RouteTablePublic
RouteDefaultPrivateA:
  Type: "AWS::EC2::Route"
  Properties:
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGatewayA
    RouteTableId: !Ref RouteTablePrivateA
RouteDefaultPrivateB:
  Type: "AWS::EC2::Route"
  Properties:
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGatewayB
    RouteTableId: !Ref RouteTablePrivateB
RouteTablePublic:
  Type: "AWS::EC2::RouteTable"
  Properties:
    VpcId: !Ref VPC
RouteTablePrivateA:
  Type: "AWS::EC2::RouteTable"
  Properties:
    VpcId: !Ref VPC
RouteTablePrivateB:
  Type: "AWS::EC2::RouteTable"
  Properties:
    VpcId: !Ref VPC
EIPNatGWA:
  DependsOn: GatewayAttach
  Type: "AWS::EC2::EIP"
  Properties:
    Domain: vpc
EIPNatGWB:
  DependsOn: GatewayAttach
  Type: "AWS::EC2::EIP"
  Properties:
    Domain: vpc
NatGatewayA:
  Type: "AWS::EC2::NatGateway"
  Properties:
    AllocationId: !GetAtt EIPNatGWA.AllocationId
    SubnetId: !Ref SubnetPublicSharedA
NatGatewayB:
  Type: "AWS::EC2::NatGateway"
  Properties:
    AllocationId: !GetAtt EIPNatGWB.AllocationId
    SubnetId: !Ref SubnetPublicSharedB
```

**Solution AWS CloudFormation Template to review at end of the lab:**

## Build VPC with AWS CloudFormation Lab

Create a file `Lab_Solution_CloudFormation_Module_General_ImmersionDay.yaml` and copy paste following code :

```
AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  vpccidr:
    Type: String
    MinLength: 9
    MaxLength: 18
    AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"
    ConstraintDescription: Must be a valid CIDR range in the form x.x.x.x/16
    Default: 10.20.0.0/16
  psharedacidr:
    Type: String
    MinLength: 9
    MaxLength: 18
    AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"
    ConstraintDescription: Must be a valid CIDR range in the form x.x.x.x/22
    Default: 10.20.0.0/22
  psharedbcidr:
    Type: String
    MinLength: 9
    MaxLength: 18
    AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"
    ConstraintDescription: Must be a valid CIDR range in the form x.x.x.x/22
    Default: 10.20.4.0/22
Resources:
  VPC:
    Type: "AWS::EC2::VPC"
    Properties:
      CidrBlock: !Ref vpccidr
  IGW:
    Type: "AWS::EC2::InternetGateway"
  S3AppBucket:
    DeletionPolicy: Retain
    Type: "AWS::S3::Bucket"
    Properties:
      AccessControl: PublicRead
      WebsiteConfiguration:
        ErrorDocument: index.html
        IndexDocument: index.html
  BucketPolicyApp:
    Type: "AWS::S3::BucketPolicy"
    Properties:
      Bucket: !Ref S3AppBucket
      PolicyDocument:
```



Statement:

```
-
  Sid: "ABC123"
  Action:
    - "s3:GetObject"
  Effect: Allow
  Resource: !Join [ "", [ "arn:aws:s3:::", !Ref S3AppBucket, "/" ] ]
  Principal:
    AWS:
      - "*"

```

GatewayAttach:

```
Type: "AWS::EC2::VPCGatewayAttachment"
Properties:
  InternetGatewayId: !Ref IGW
  VpcId: !Ref VPC

```

SubnetPublicSharedA:

```
Type: "AWS::EC2::Subnet"
Properties:
  AvailabilityZone: !Select [ 0, !GetAZs ]
  CidrBlock: !Ref psharedacidr
  MapPublicIpOnLaunch: true
  VpcId: !Ref VPC

```

SubnetPublicSharedB:

```
Type: "AWS::EC2::Subnet"
Properties:
  AvailabilityZone: !Select [ 1, !GetAZs ]
  CidrBlock: !Ref psharedbcidr
  MapPublicIpOnLaunch: true
  VpcId: !Ref VPC

```

SubnetRouteTableAssociatePublicA:

```
Type: "AWS::EC2::SubnetRouteTableAssociation"
Properties:
  RouteTableId: !Ref RouteTablePublic
  SubnetId: !Ref SubnetPublicSharedA

```

SubnetRouteTableAssociatePublicB:

```
Type: "AWS::EC2::SubnetRouteTableAssociation"
Properties:
  RouteTableId: !Ref RouteTablePublic
  SubnetId: !Ref SubnetPublicSharedB

```

RouteDefaultPublic:

```
Type: "AWS::EC2::Route"
DependsOn: GatewayAttach
Properties:
  DestinationCidrBlock: 0.0.0.0/0
  GatewayId: !Ref IGW
  RouteTableId: !Ref RouteTablePublic

```

RouteDefaultPrivateA:

```
Type: "AWS::EC2::Route"

```

```
Properties:
  DestinationCidrBlock: 0.0.0.0/0
  NatGatewayId: !Ref NatGatewayA
  RouteTableId: !Ref RouteTablePrivateA
RouteDefaultPrivateB:
  Type: "AWS::EC2::Route"
Properties:
  DestinationCidrBlock: 0.0.0.0/0
  NatGatewayId: !Ref NatGatewayB
  RouteTableId: !Ref RouteTablePrivateB
RouteTablePublic:
  Type: "AWS::EC2::RouteTable"
Properties:
  VpcId: !Ref VPC
RouteTablePrivateA:
  Type: "AWS::EC2::RouteTable"
Properties:
  VpcId: !Ref VPC
RouteTablePrivateB:
  Type: "AWS::EC2::RouteTable"
Properties:
  VpcId: !Ref VPC
EIPNatGWA:
  DependsOn: GatewayAttach
  Type: "AWS::EC2::EIP"
Properties:
  Domain: vpc
EIPNatGWB:
  DependsOn: GatewayAttach
  Type: "AWS::EC2::EIP"
Properties:
  Domain: vpc
NatGatewayA:
  Type: "AWS::EC2::NatGateway"
Properties:
  AllocationId: !GetAtt EIPNatGWA.AllocationId
  SubnetId: !Ref SubnetPublicSharedA
NatGatewayB:
  Type: "AWS::EC2::NatGateway"
Properties:
  AllocationId: !GetAtt EIPNatGWB.AllocationId
  SubnetId: !Ref SubnetPublicSharedB
Outputs:
  vpcid:
    Description: ID of Shared Infrastructure VPC
    Value: !Ref VPC
    Export: # added to export
    Name: sharedinf-vpcid
```

## Build VPC with AWS CloudFormation Lab

```
natgatewayaid:
  Description: ID of NAT Gateway A
  Value: !Ref NatGatewayA
natgatewaybid:
  Description: ID of NAT Gateway B
  Value: !Ref NatGatewayB
appbucketurl:
  Description: Shared Infrastructure App Bucket
  Value: !GetAtt S3AppBucket.WebsiteURL
  Export: # added to export
  Name: sharedinf-appbucketurl
```