



lab



lab title

VPC Architecture Design and Deployment with CloudFormation Designer V1.02



Course title

BackSpace Academy
AWS Certified Associate



Table of Contents

Contents

Table of Contents.....	1
About the Lab	2
Tools for Creating Architecture Diagrams of AWS	3
Creating a VPC	4
Deploying the Architecture with CloudFormation.....	9
Creating an Internet Gateway and a Route to the Subnet	12
Updating the Architecture with CloudFormation	17
Creating an EC2 Instance and Security Group	20
Create Security Group	20
Create EC2 Instance.....	21
Deploying the Wordpress Server.....	25
Save the template as a new version.....	25
Viewing your WordPress Server	27
Clean Up.....	28

About the Lab

Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.

These lab notes are to support the hands on instructional videos of the VPC Architecture section of the AWS Certified Associate Architecture Essentials.

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the latest version with any updates or corrections.

Tools for Creating Architecture Diagrams of AWS

The AWS website has a range of tools for creating architecture diagrams.

aws.amazon.com/architecture/icons

Desktop solutions:

- Powerpoint templates
- SVG images for LibreOffice Draw (Free)
- Visio Stencils
- EPS images for Adobe Illustrator
- Sketch templates

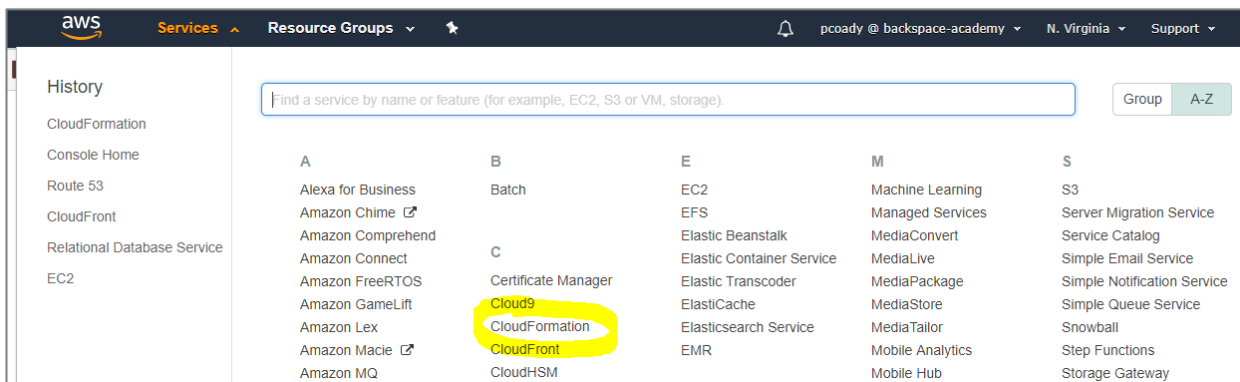
Online solutions:

- Lucidchart
- Cacoo
- Creately
- draw.io
- Cloudcraft
- DC Solution Factory

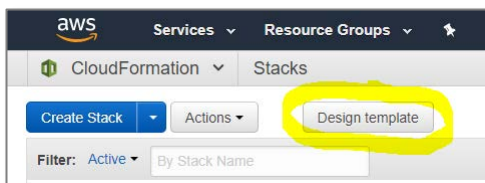
▶ Creating a VPC

In this section, we will use the AWS CloudFormation Designer to create a Virtual Private Cloud with a Subnet.

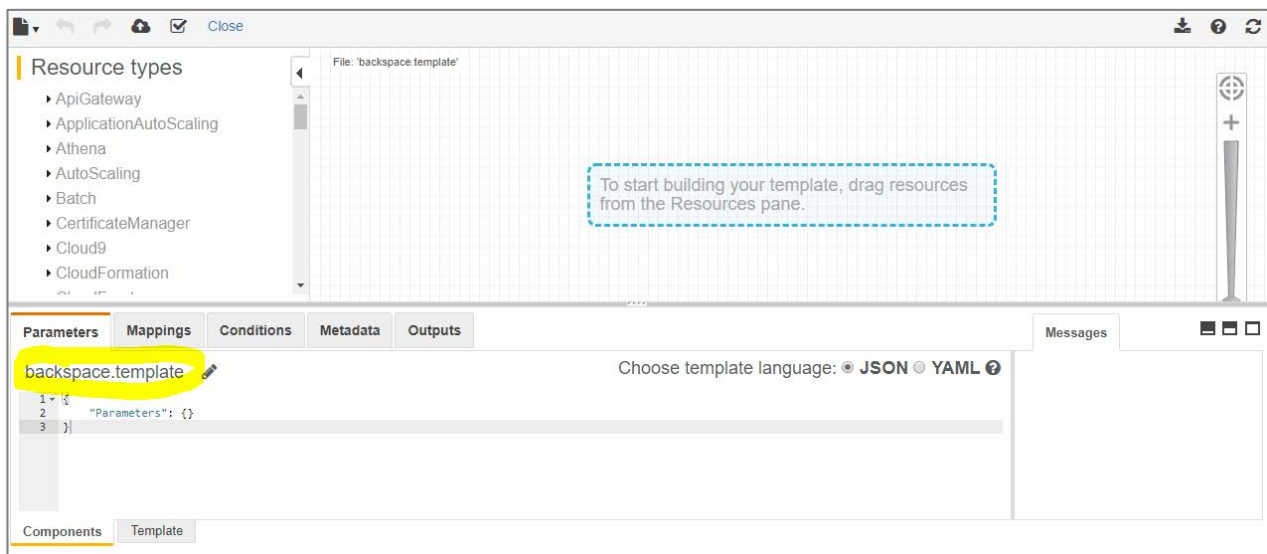
Make sure you are in US-East (N. Virginia) region. From the AWS console select “CloudFormation” from the Application services.



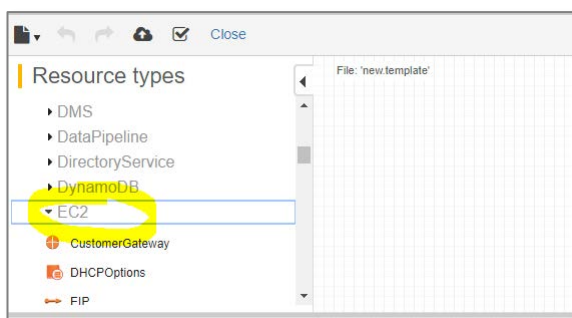
Click “Design Template”



Change the name of the template and press enter

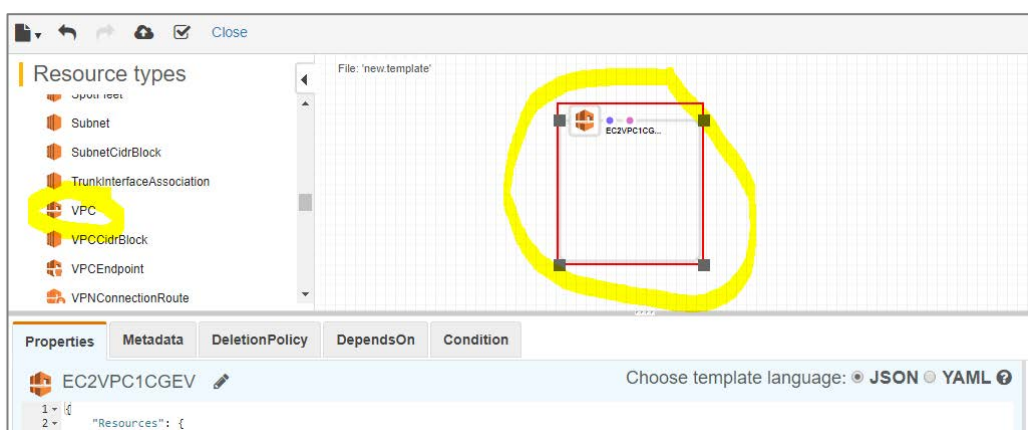


Scroll down to "EC2"

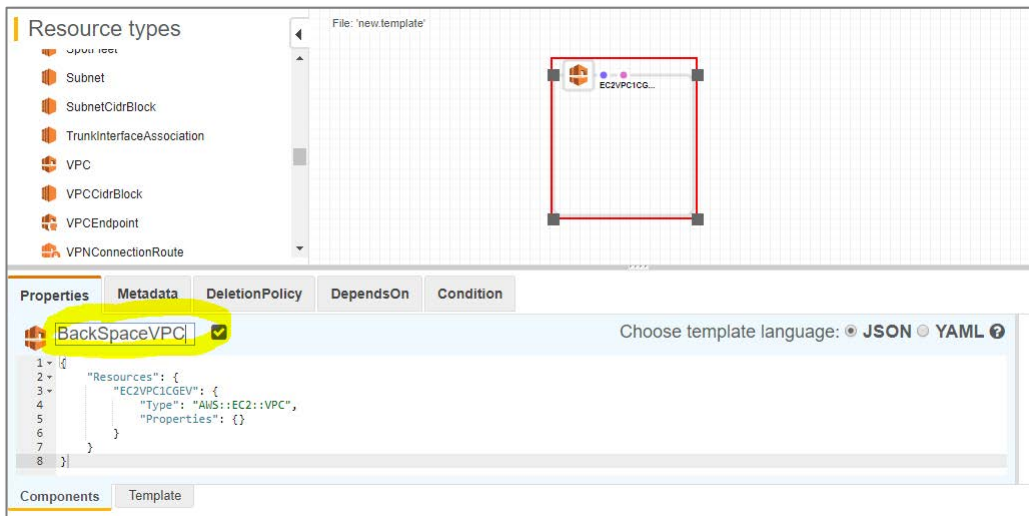


Scroll down to "VPC"

Drag and drop the VPC icon onto the canvas



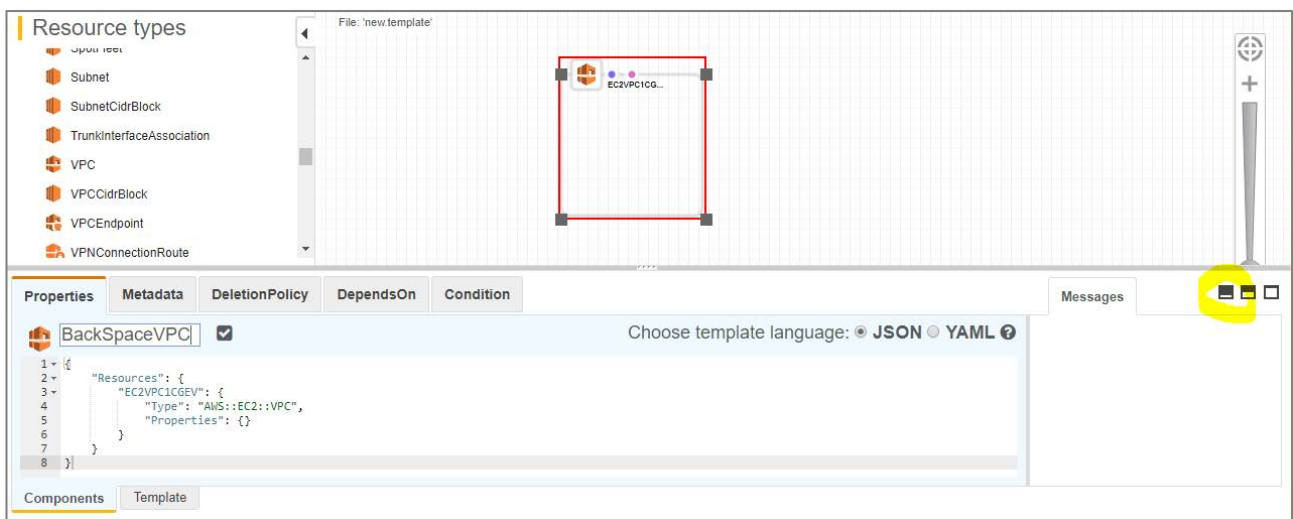
Go to the editor pane and give the VPC a new name



Press enter and refresh the canvas



Minimise the Editor



Resize the VPC

Go back to Split Screen view to see the editor pane

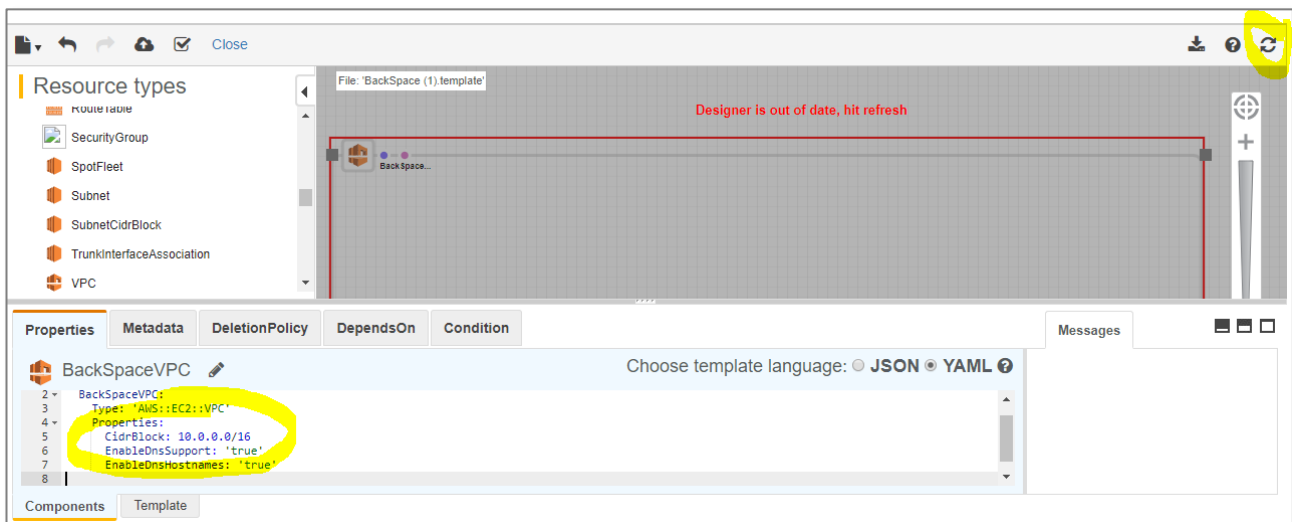
Select YAML

Add the CIDR information to the properties (make sure they are indented correctly)

CidrBlock: 10.0.0.0/16

EnableDnsSupport: 'true'

EnableDnsHostnames: 'true'



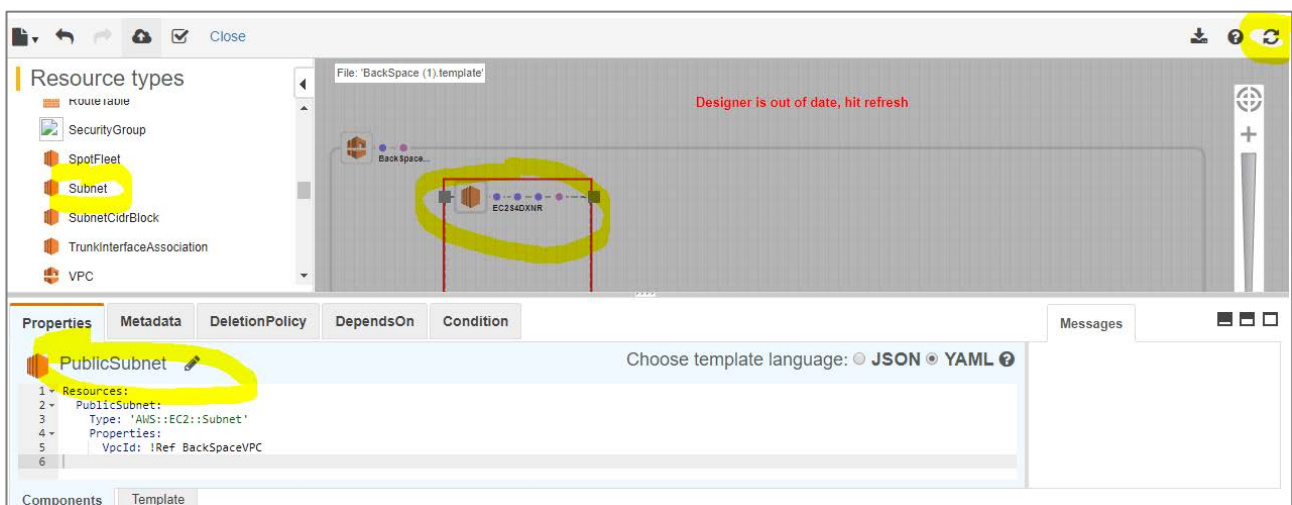
Drag the Subnet icon onto the VPC

Resize the subnet

Change the Subnet name to "PublicSubnet"

Click to Refresh Canvas

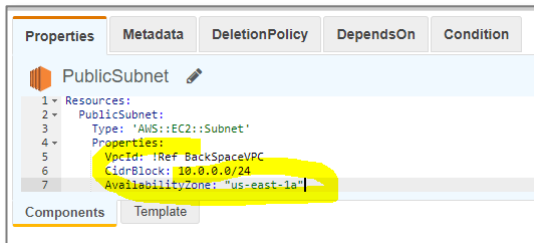
***Troubleshooting Note – If the subnet doesn't appear in the editor, click on the template tab, then the components tab again to reload it.**



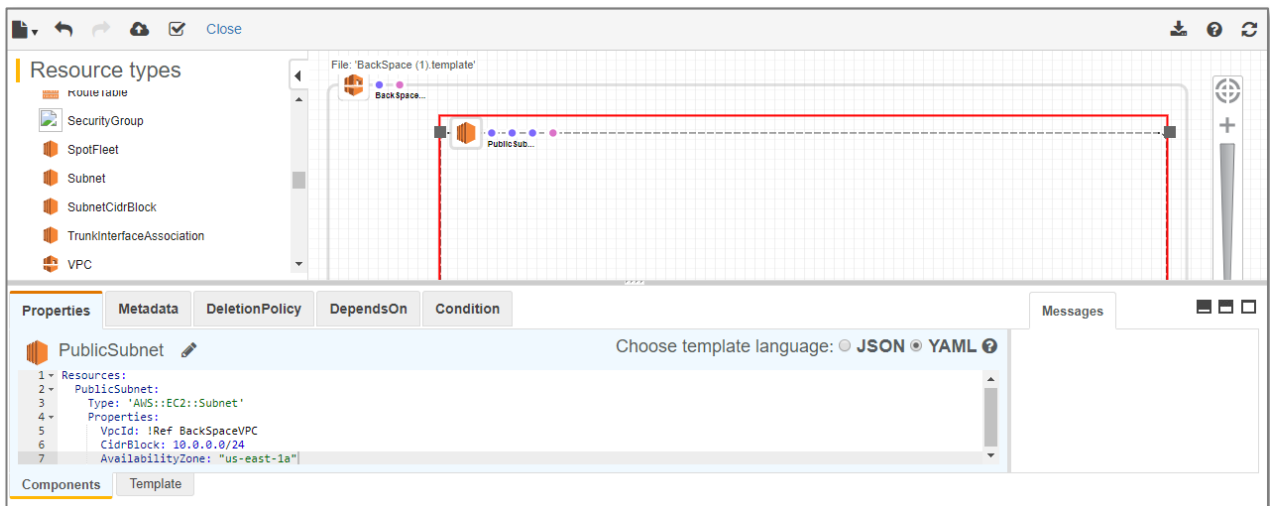
Add the following properties to the Subnet

CidrBlock: 10.0.0.0/24

AvailabilityZone: "us-east-1a"



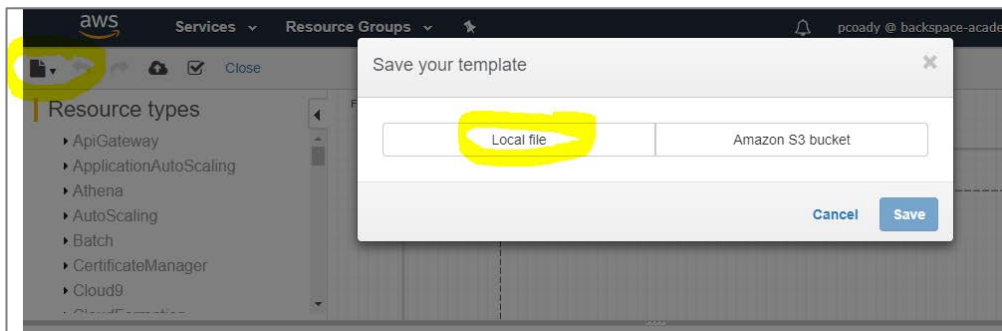
Click to refresh canvas



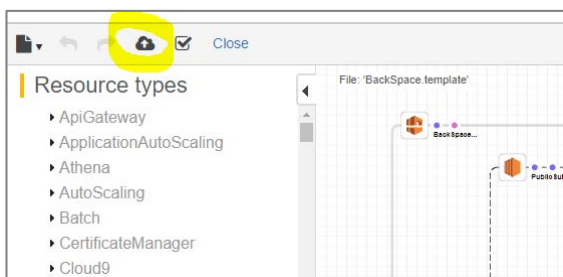
▶ Deploying the Architecture with CloudFormation

In this section, we will use the AWS CloudFormation service to save our template and use it to deploy our architecture.

Select File – Save
Select Local File
Save the File



Click “Create Stack”



Click Next

*Troubleshooting Note – If the S3 URL doesn't work use “Upload a template to Amazon S3” and select the saved template

Select Template

Specify Details
Options
Review

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
[Choose File](#) No file chosen
- ☒ Specify an Amazon S3 template URL
 [View/Edit template in Designer](#)

[Cancel](#) [Next](#)

Give the stack a name
Click Next

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

[Cancel](#) [Previous](#) [Next](#)

Click Next again

Rollback Triggers

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. [Learn more](#)

Monitoring Time Minutes
Minimum value of 0. Maximum value of 180.

Available triggers remaining: 5

	Type	ARN (Amazon Resource Name)	
1	AWS::CloudWatch::Alarm	<input type="text"/>	+

Advanced

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

[Cancel](#) [Previous](#) [Next](#)

Click Create

[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](#)

Click the refresh icon to see status

Create Stack [Actions](#) Design template

Filter: Active Showing 1 stack

	Stack Name	Created Time	Status	Description
<input type="checkbox"/>	BackSpace	2018-02-09 03:56:55 UTC+1100	CREATE_IN_PROGRE...	

If successful you see "CREATE_COMPLETE"

Create Stack [Actions](#) Design template

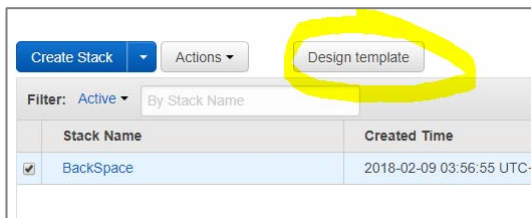
Filter: Active

	Stack Name	Created Time	Status
<input checked="" type="checkbox"/>	BackSpace	2018-02-09 03:56:55 UTC+1100	CREATE_COMPLETE

▶ Creating an Internet Gateway and a Route to the Subnet

In this section, we will use the AWS CloudFormation Designer to create an Internet Gateway and route to the subnet.

Click on “Design Template”



Select File – Open

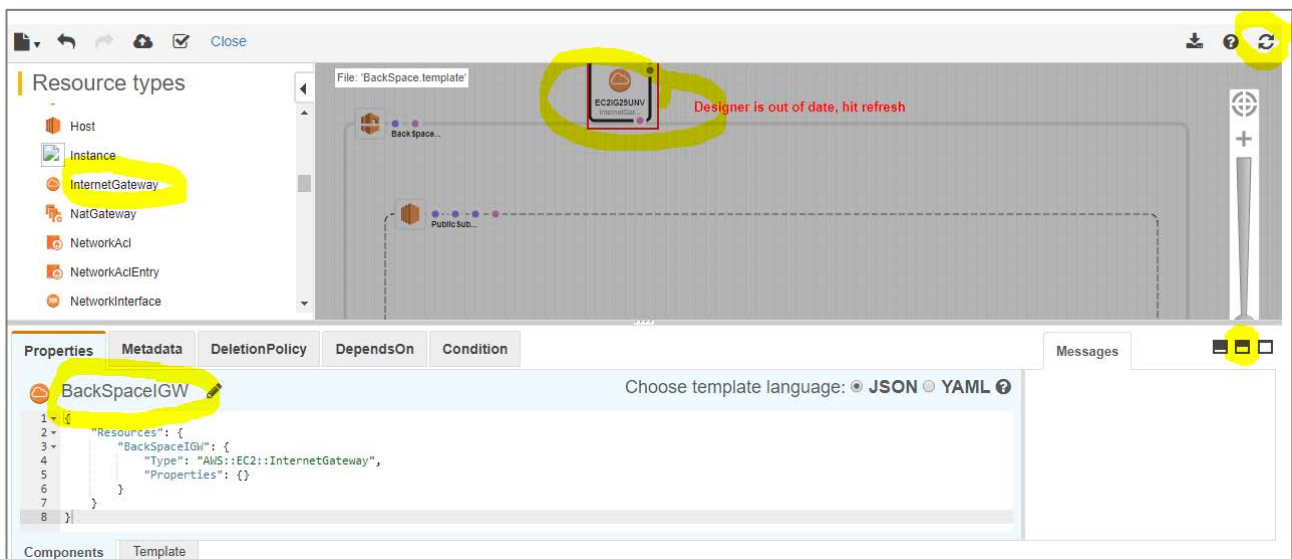
Open the local template file you saved

Drag and Drop an Internet Gateway onto the empty canvas (outside of the VPC).

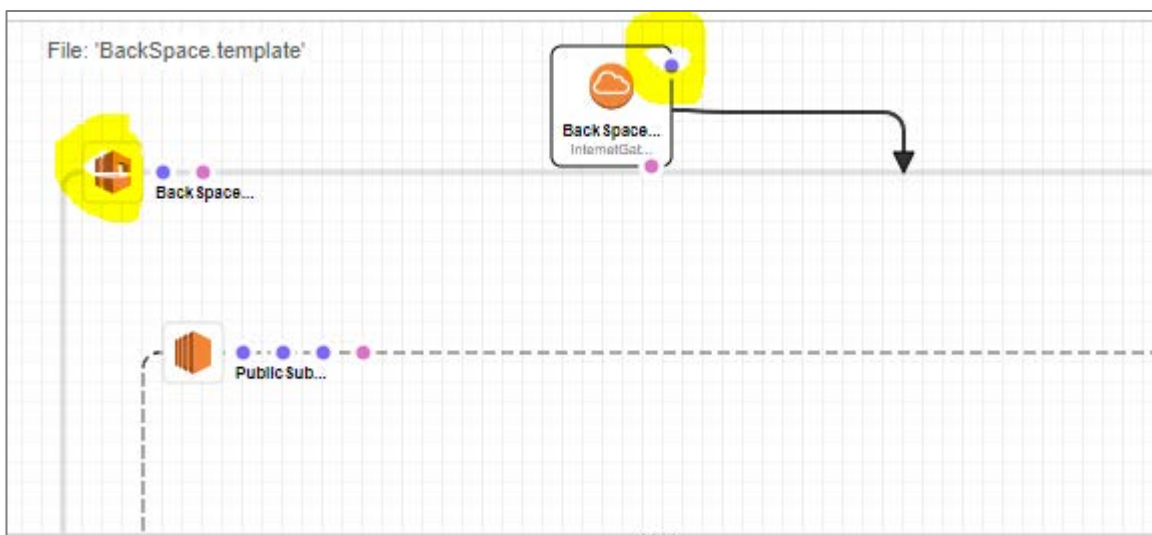
Rename the IGW and press enter

Refresh the canvas

***Troubleshooting Note – If the IGW doesn’t appear in the editor, click on the template tab, then the components tab again to reload it.**



Click and drag the blue dot on the IGW icon onto the VPC to create a connection to the VPC

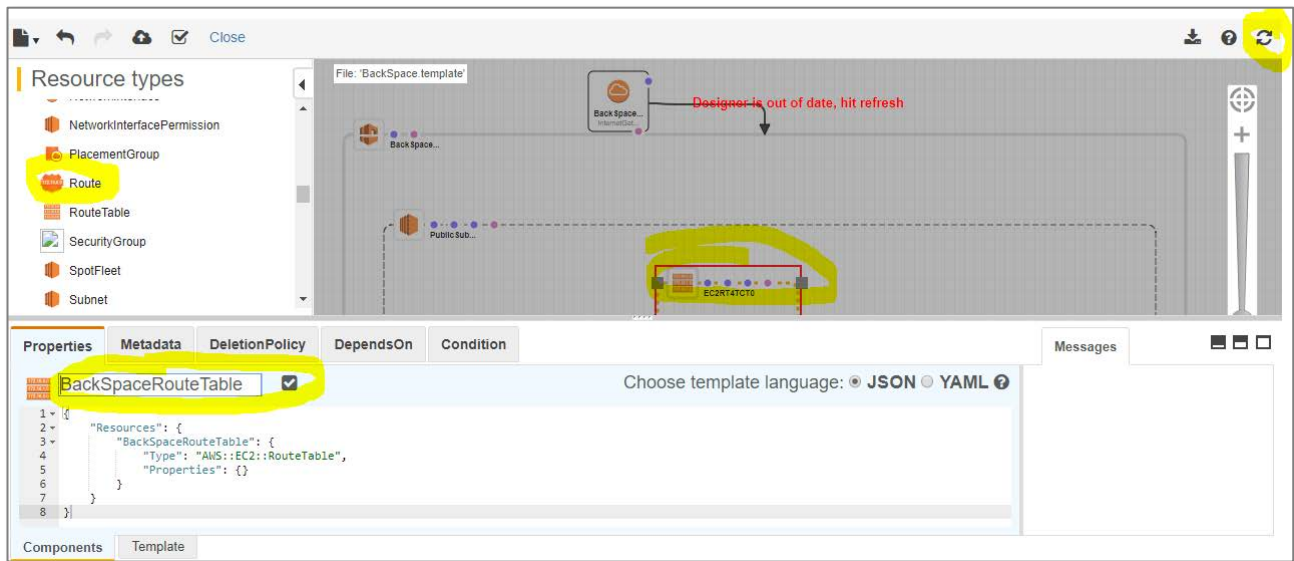


Drag and drop a RouteTable icon onto the subnet

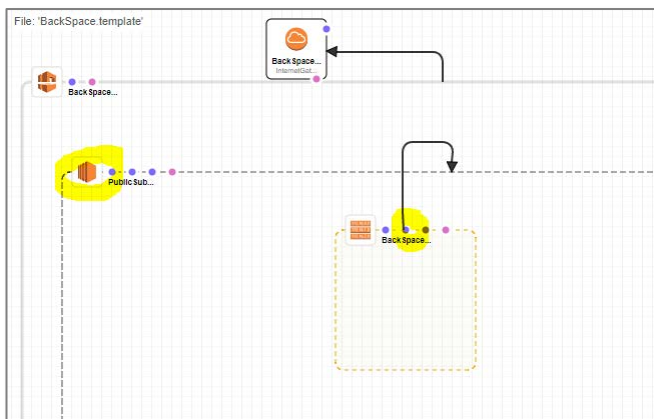
Rename the Route table and press enter

Click to refresh the canvas

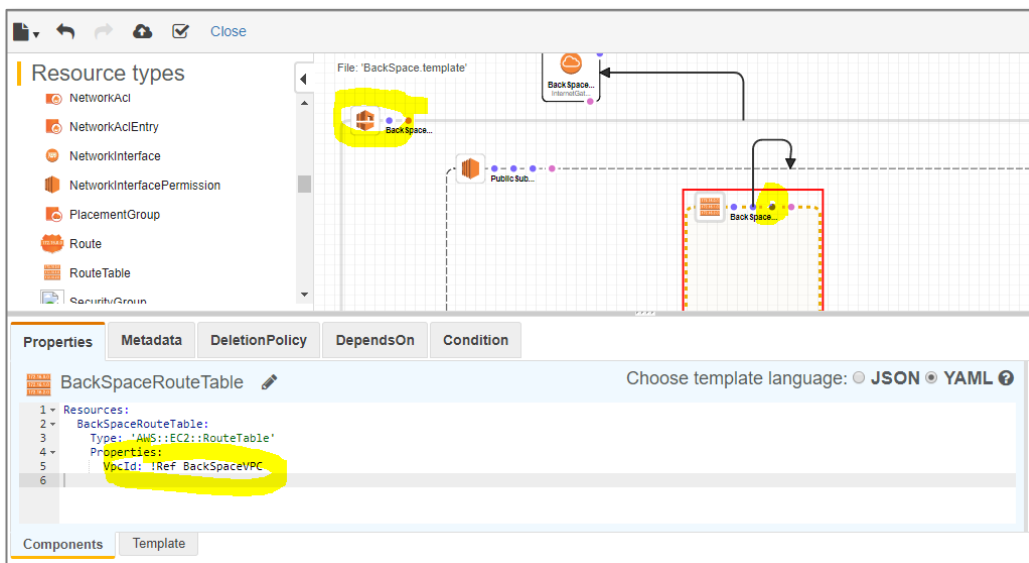
***Troubleshooting Note – If the RouteTable doesn't appear in the editor, click on the template tab, then the components tab again to reload it.**



Drag and drop the second blue dot SubnetRouteTableAssociation on the RouteTable icon onto the subnet to associate the Route Table with the subnet



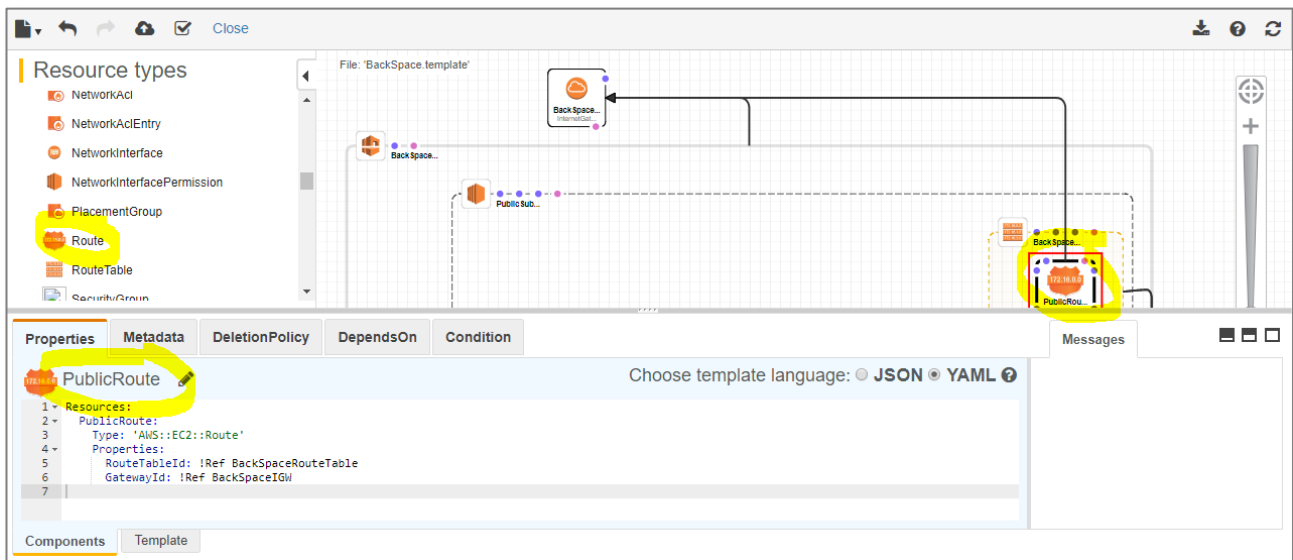
Drag and drop the third blue dot on the RouteTable icon onto the VPC to associate the Route Table with the VPC



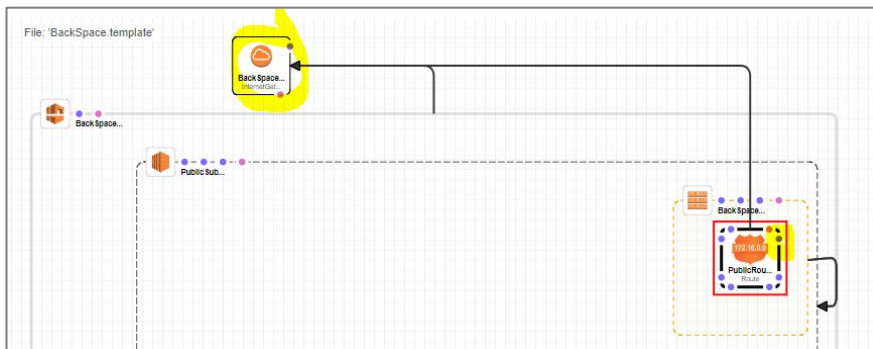
Drag and drop a Route icon onto the RouteTable

Rename the Route as “PublicRoute” and press enter

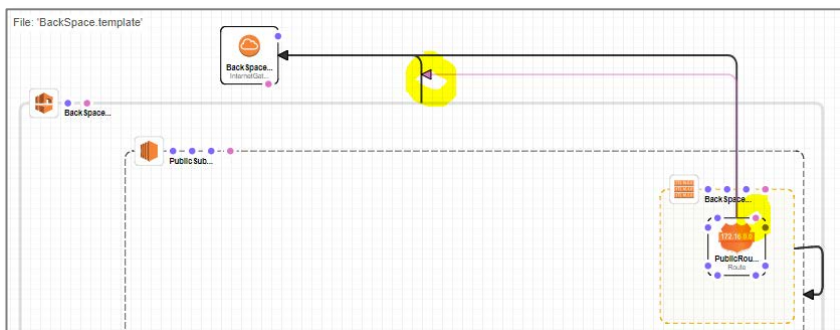
Click to refresh the canvas



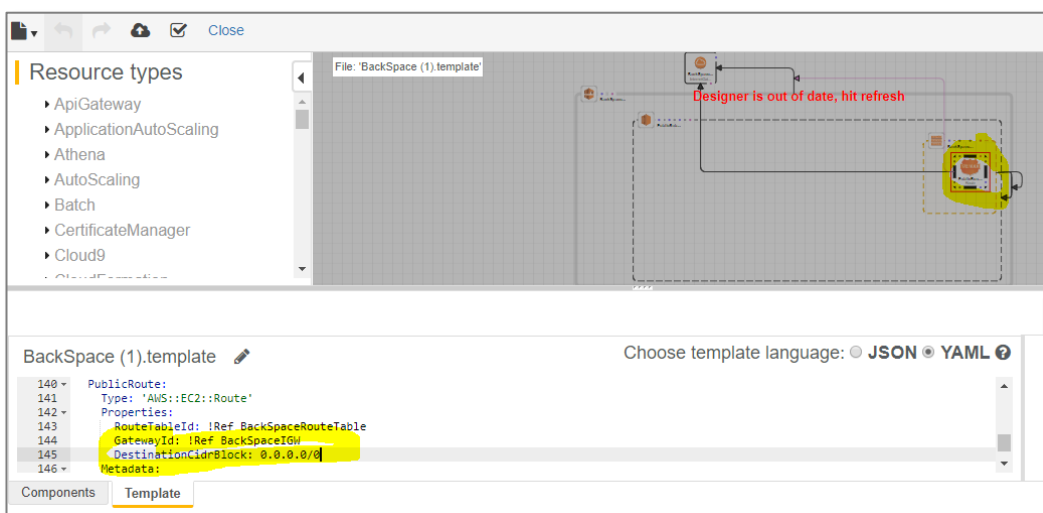
Drag and drop from the blue GatewayID (not the EgressOnlyInternetGateway) dot to the Internet Gateway



Drag and drop the pink “DependsOn” dot from the Route to the IGW / VPC connection



Add the following to the Properties (make sure indentation is correct):
DestinationCidrBlock: 0.0.0.0/0



You have created an Internet gateway (IGW) and a route from the subnet to the IGW. This is now a public subnet.

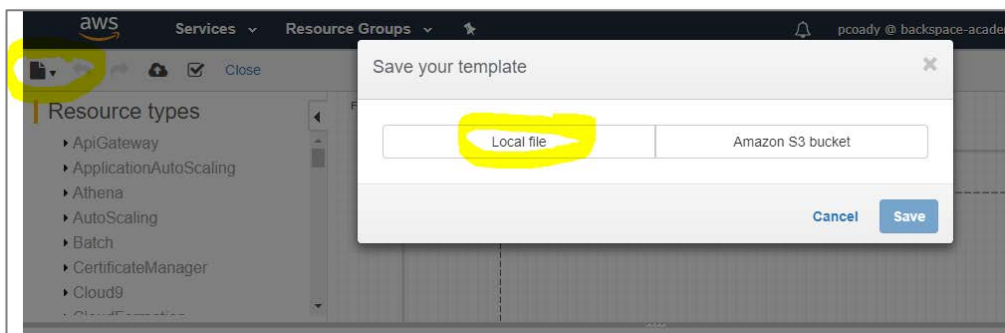
Updating the Architecture with CloudFormation

In this section, we will use the AWS CloudFormation service to save our template and use it to update our architecture.

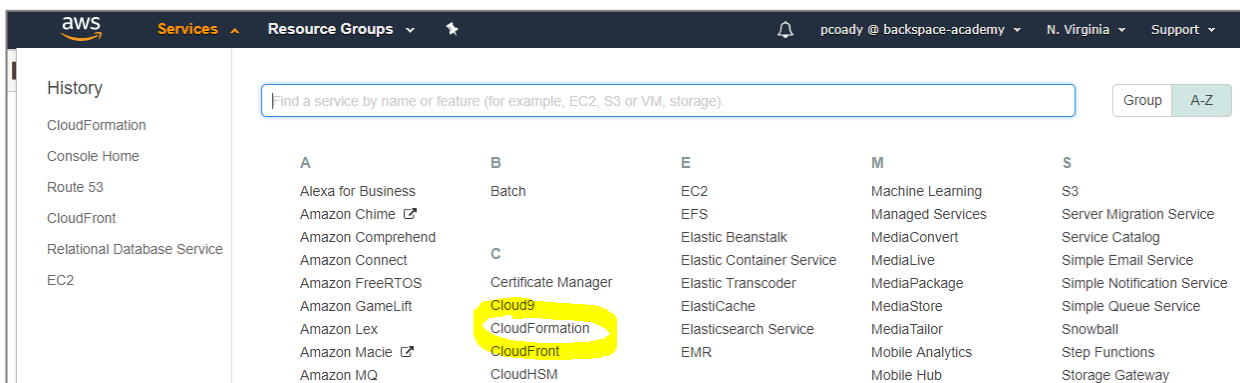
Select File – Save

Select Local File

Save the File as a different version

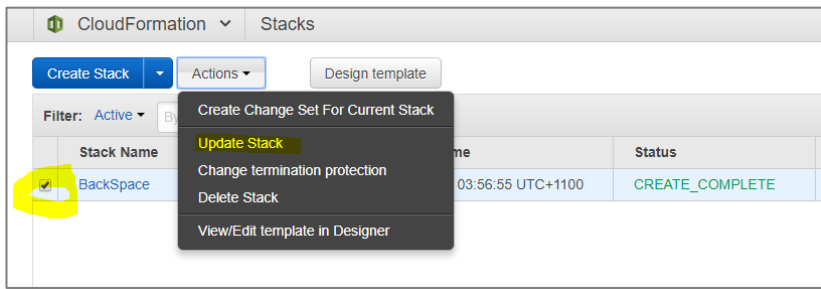


Go to Services – CloudFormation



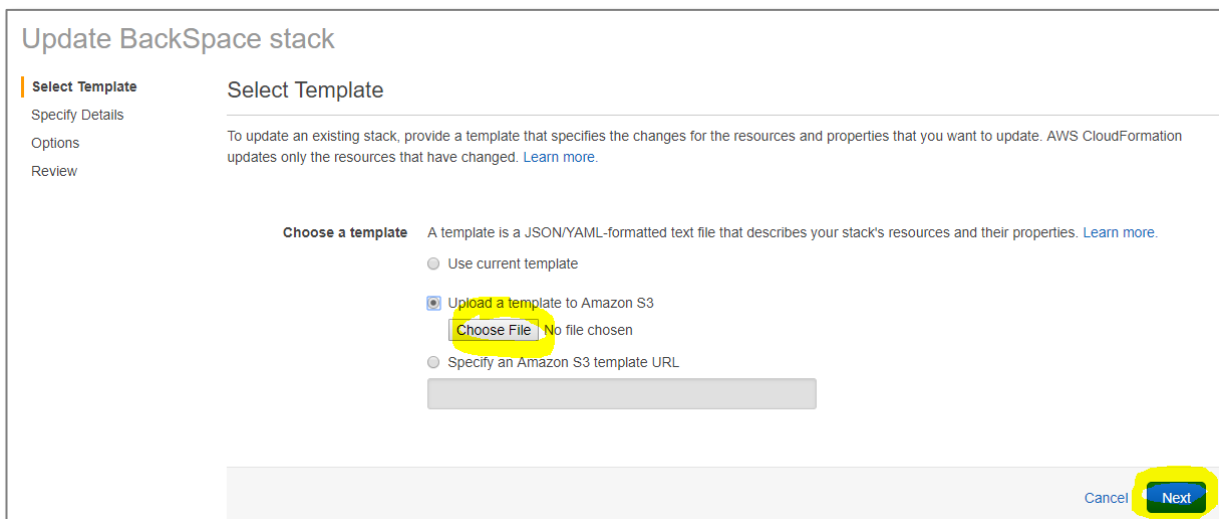
Select the stack

Select Actions – Update Stack

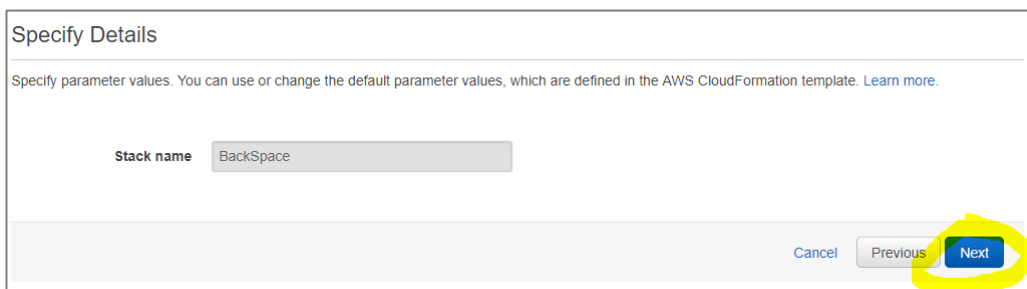


Upload the newly saved template version

Click Next



Click Next again



Click Next again

Stack policy to use during update

☒ Use existing policy
☐ Enter policy

☐ Upload policy file
 No file chosen
[Learn more](#)

Click Update

Preview your changes

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

Action	Logical ID	Physical ID	Resource type	Replacement
Add	BackSpaceIGW		AWS::EC2::InternetGateway	
Add	BackSpaceRouteTable		AWS::EC2::RouteTable	
Add	EC2SRTA3Y58U		AWS::EC2::SubnetRouteTableAssociation	
Add	EC2SRTA4SRS5		AWS::EC2::SubnetRouteTableAssociation	
Add	EC2VPCG2AUS7		AWS::EC2::VPCGatewayAttachment	
Add	PublicRoute		AWS::EC2::Route	

If successful you will see "UPDATE_COMPLETE"

Create Stack ▼ Actions ▼ Design template

Filter: Active ▼ By Stack Name

	Stack Name	Created Time	Status
<input type="checkbox"/>	BackSpace	2018-02-09 03:56:55 UTC+1100	UPDATE_COMPLETE

Creating an EC2 Instance and Security Group

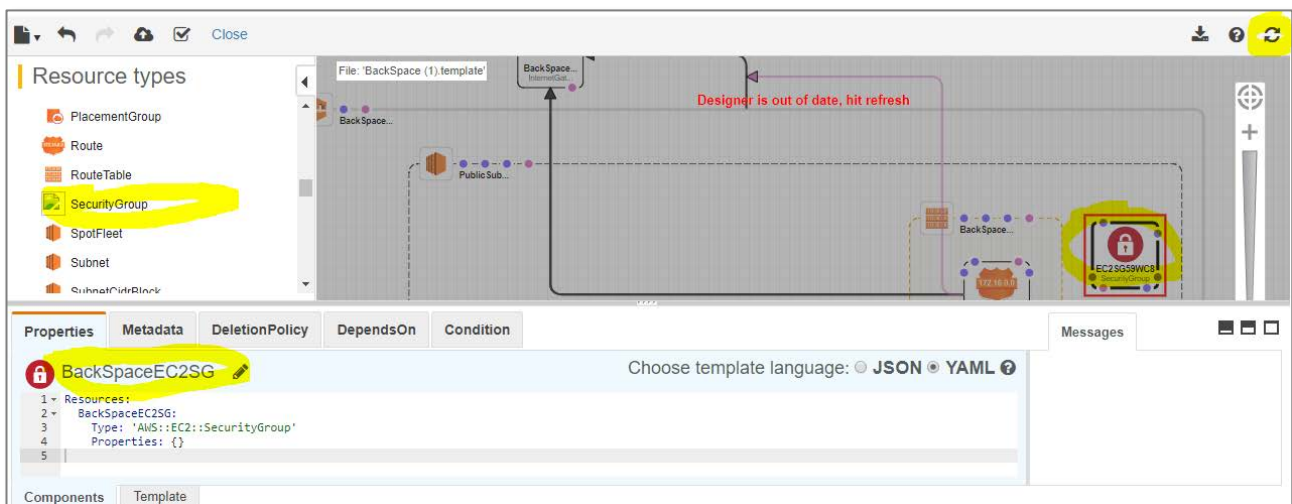
In this section, we will use the **AWS CloudFormation Designer** to create an EC2 Instance and Security Group.

Create Security Group

Resize the subnet to allow room in the VPC

Drag and Drop a SecurityGroup icon onto the VPC

Rename the Security Group “BackSpaceEC2SG” and press enter



Add the following properties to the Security Group:

GroupDescription: Allow access from HTTP and SSH traffic

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: '80'

ToPort: '80'

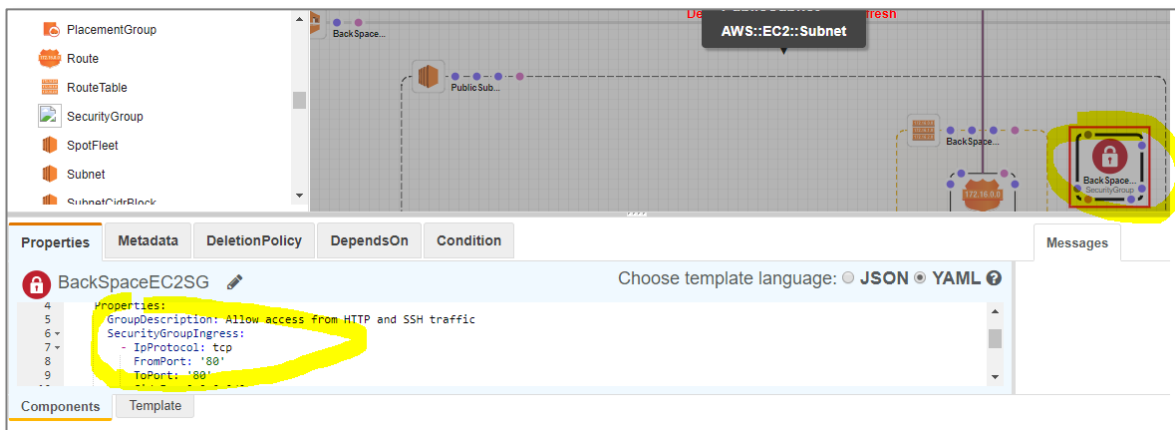
CidrIp: 0.0.0.0/0

- IpProtocol: tcp

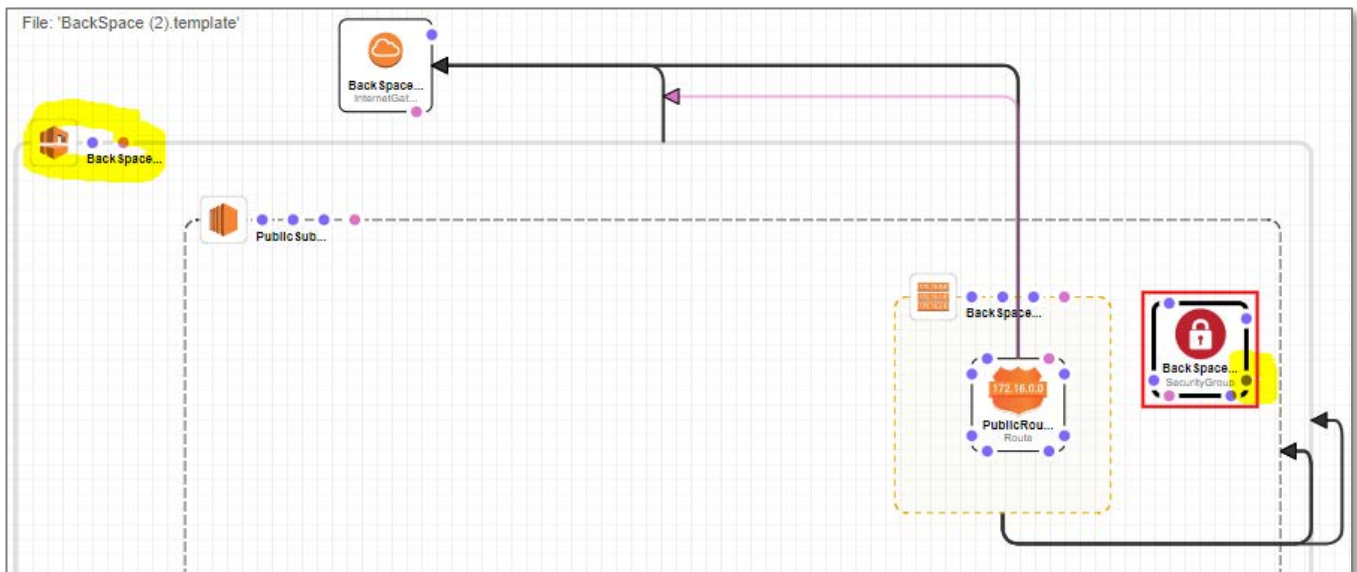
FromPort: '22'

ToPort: '22'

CidrIp: 0.0.0.0/0



Drag and drop from the blue VpcID dot to the VPC

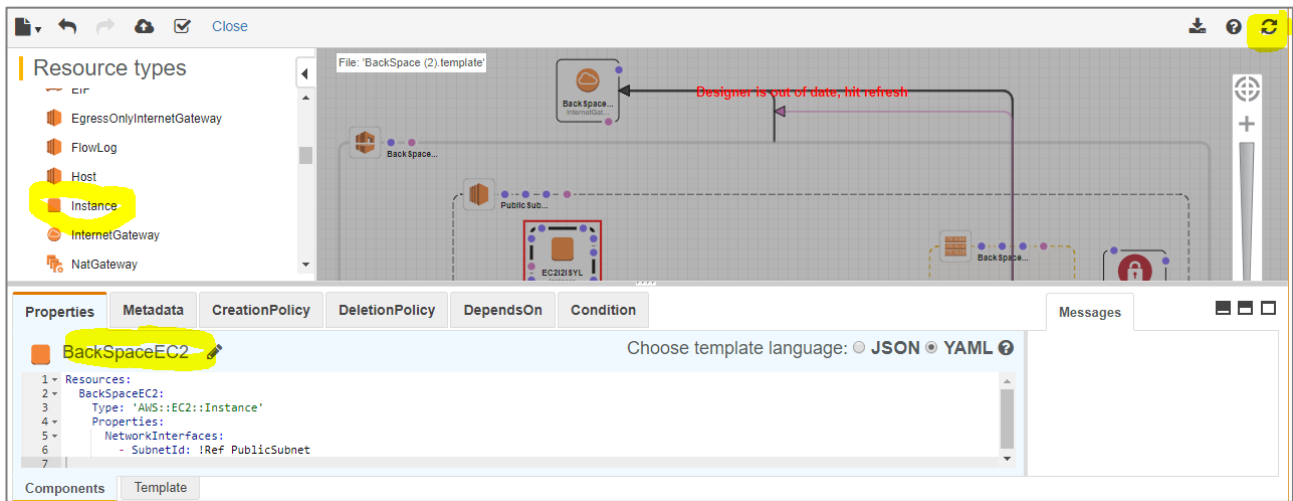


Create EC2 Instance

Drag and drop an Instance onto the subnet.

Rename it "BackSpaceEC2" and press enter

Click to refresh canvas



Create a “DependsOn” link from the pink dot to the PublicRoute

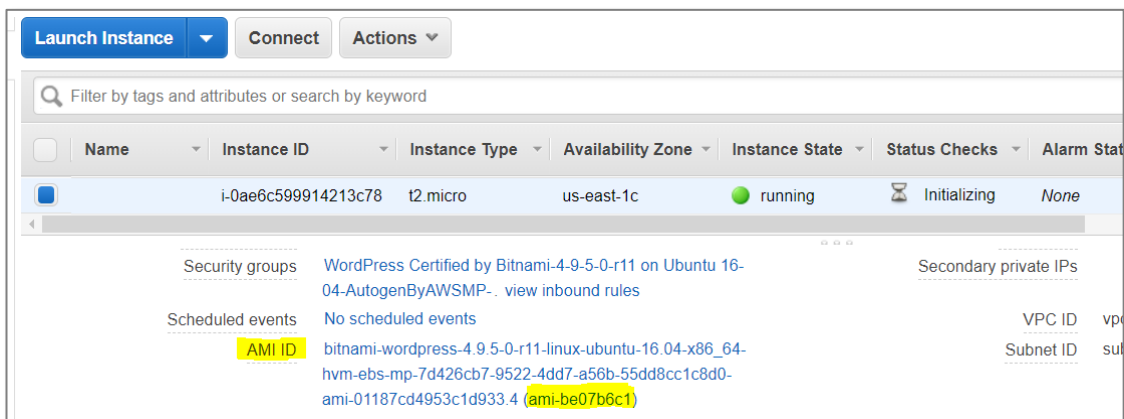


Now we will find a suitable Bitnami WordPress AMI id for our instance

Open the EC2 console in another window.

Launch an instance using a Bitnami Wordpress HVM AMI from the AWS Marketplace

When the instance has started launching copy the AMI id:



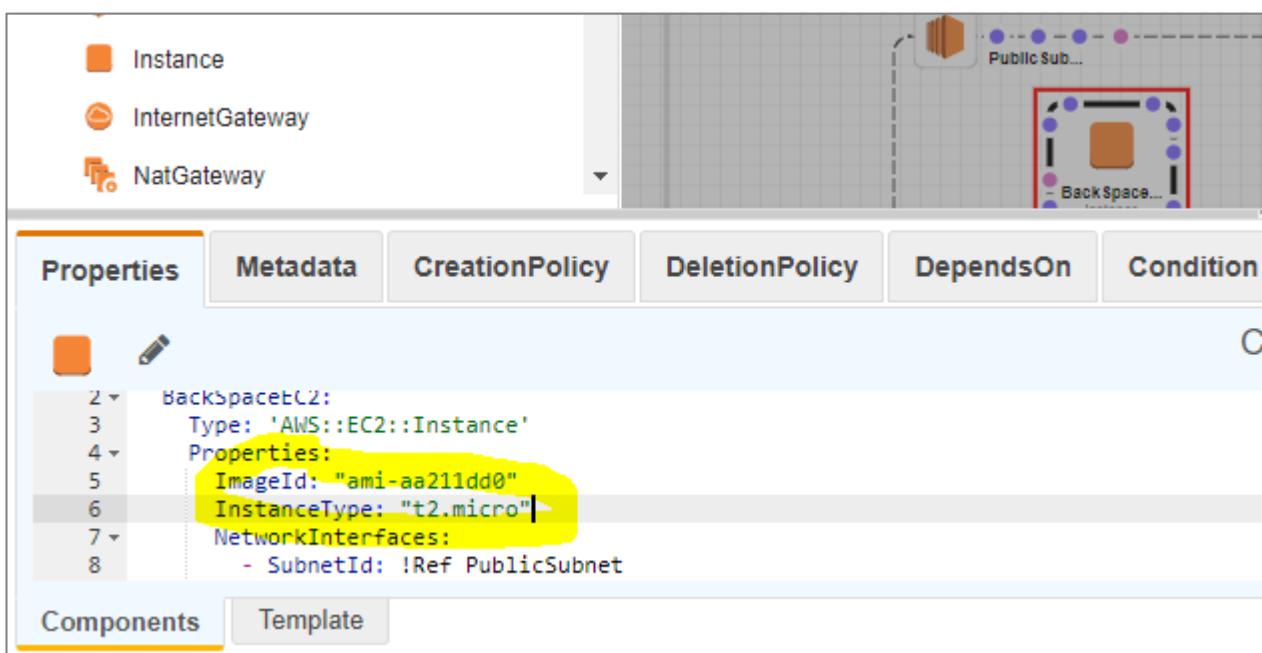
Terminate the instance after you have copied the AMI ID so that you don't get billed.

Go back to CloudFormation Designer

Click on the Components tab and add the following to the instance properties (make sure you paste in the AMI id from before):

ImageId: "ami-your-ami-here"

InstanceType: "t2.micro"



Change the "NetworkInterfaces" section of the instance properties to:

NetworkInterfaces:

- GroupSet:
- !Ref BackSpaceEC2SG

AssociatePublicIpAddress: 'true'

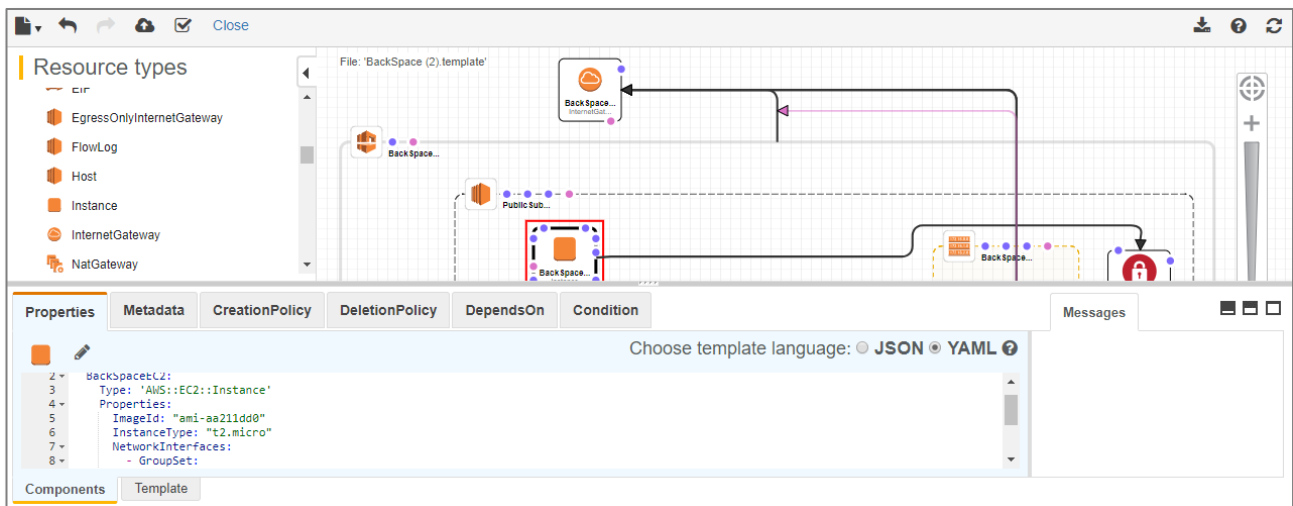
DeviceIndex: '0'

DeleteOnTermination: 'true'

SubnetId: !Ref PublicSubnet

```
2 BackSpaceEC2:
3   Type: 'AWS::EC2::Instance'
4   Properties:
5     ImageId: "ami-aa211dd0"
6     InstanceType: "t2.micro"
7     NetworkInterfaces:
8       - GroupSet:
9         - !Ref BackSpaceEC2SG
10        AssociatePublicIpAddress: 'true'
11        DeviceIndex: '0'
12        DeleteOnTermination: 'true'
13        SubnetId: !Ref PublicSubnet
14
```

Click to refresh canvas



▶ Deploying the Wordpress Server

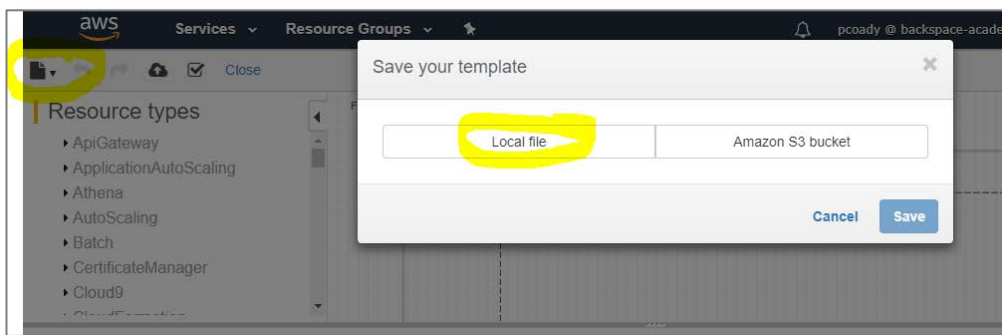
In this section, we will use the AWS CloudFormation to update our stack deploy our WordPress Server.

Save the template as a new version

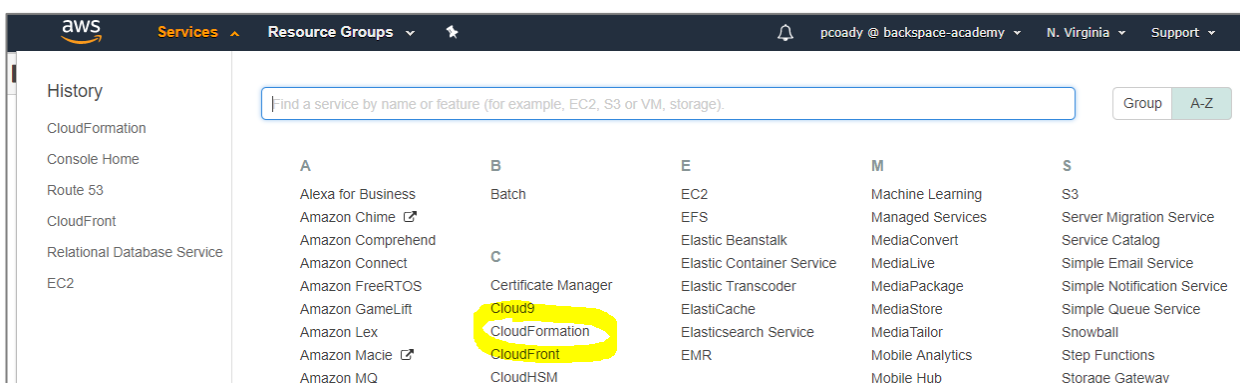
Select File – Save

Select Local File

Save the File as a different version

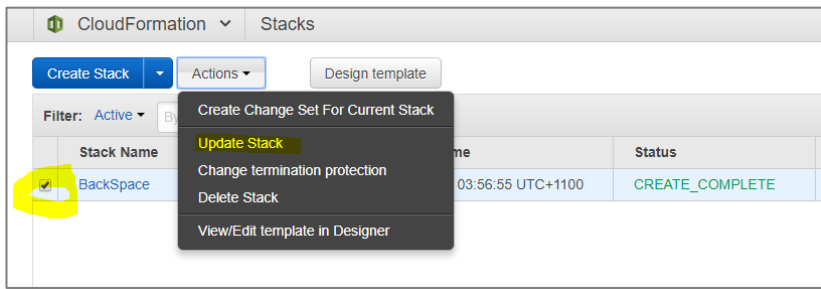


Go to Services – CloudFormation



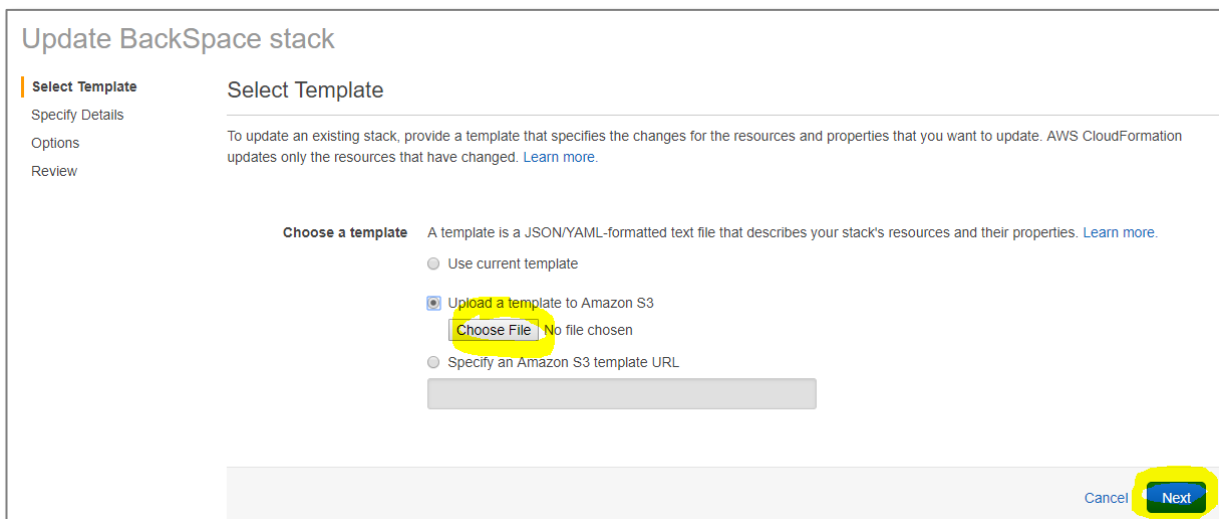
Select the stack

Select Actions – Update Stack

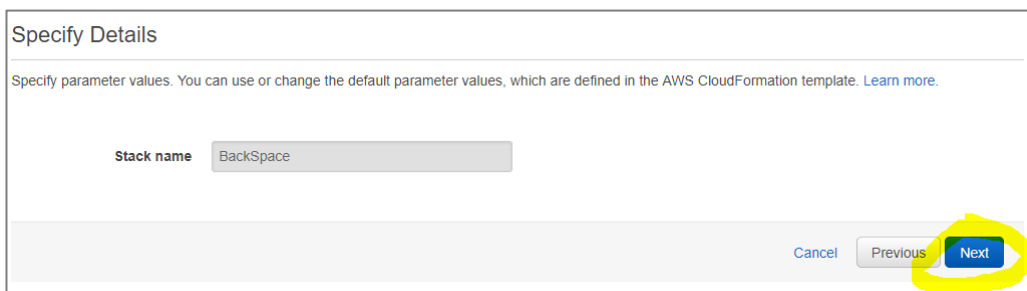


Upload the newly saved template version

Click Next



Click Next again



Click Next again

Stack policy to use during update

☒ Use existing policy

☐ Enter policy

☐ Upload policy file

No file chosen

[Learn more](#)

Click Update

Wait until stack has been updated.

Filter: Active

	Stack Name	Created Time	Status
<input checked="" type="checkbox"/>	BackSpace	2018-02-11 19:44:38 UTC+1100	UPDATE_COMPLETE

Viewing your WordPress Server

Go to the EC2 console and view the instance

Copy the public IP address

EC2 Dashboard

Instance ID: i-0cc97eea6224248c3

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Ala
<input checked="" type="checkbox"/>	i-0cc97eea6224248c3	t2.micro	us-east-1a	running	2/2 checks ...	Nor

Instance: i-0cc97eea6224248c3 Public DNS: ec2-34-236-254-111.compute-1.amazonaws.com

Description

Instance ID	i-0cc97eea6224248c3	Public DNS (IPv4)	ec2-34-236-254-111.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	34.236.254.111
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-10-0-0-209.ec2.internal
Availability zone	us-east-1a	Private IPs	10.0.0.209
Security groups	BackSpace-BackSpaceEC2SG-8GTZUJ2EJSOY. view inbound rules	Secondary private IPs	

Go to the public IP address in your browser

You will now see your WordPress website



Clean Up

Now that we have finished the lab we can delete the stack to avoid costs.

DO NOT DELETE STACK RESOURCES DIRECTLY!

Delete the stack from the CloudFormation console to delete all resources created in the stack.

