lab

lab title

VPC Architecture Design and Deployment with CloudFormation Designer

V1.02

Course title

BackSpace Academy
AWS Certified Associate

BackSpace

Table of Contents

Contents

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 Arcitecture 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 lastest 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

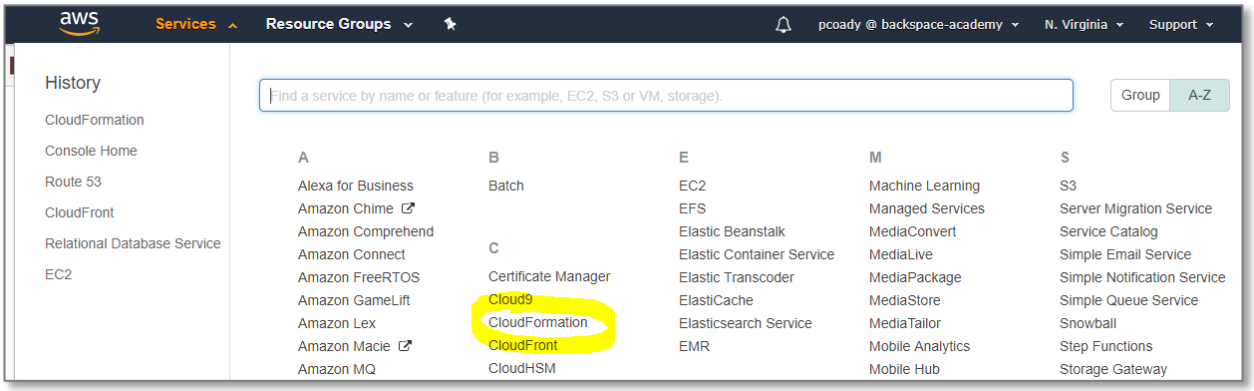
Online solutions:

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

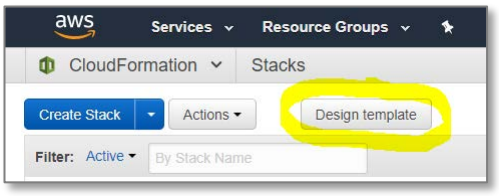
Creating a VPC

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

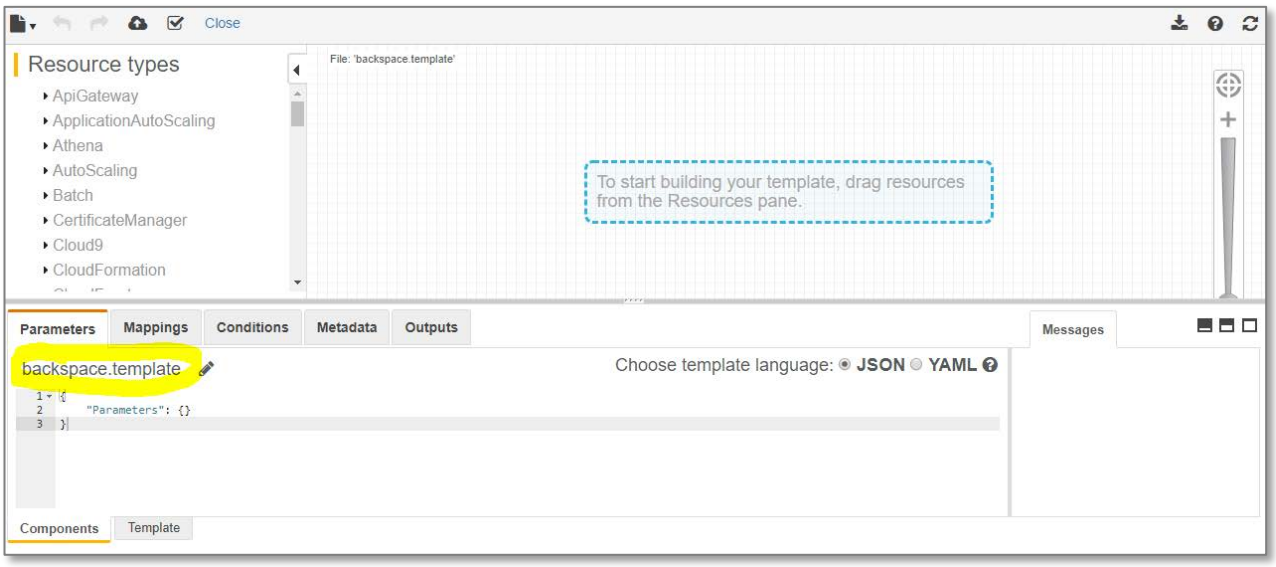
Make sure you are in US-East (N. Virginia) region. From the AWS console select “CloudFormation” from the Application services.



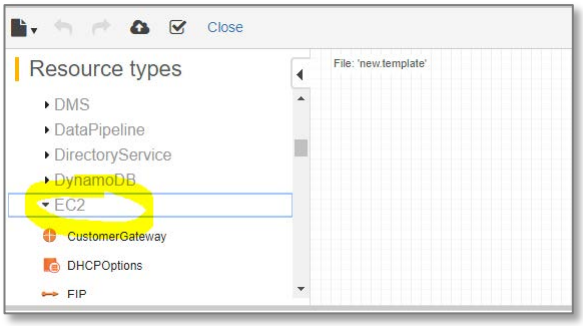
Click “Design Template”



Change the name of the template and press enter

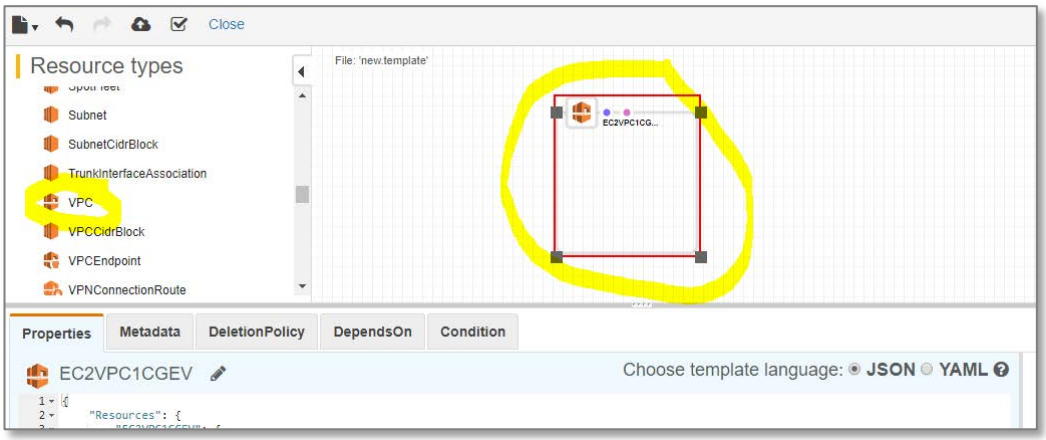


Scroll down to “EC2”

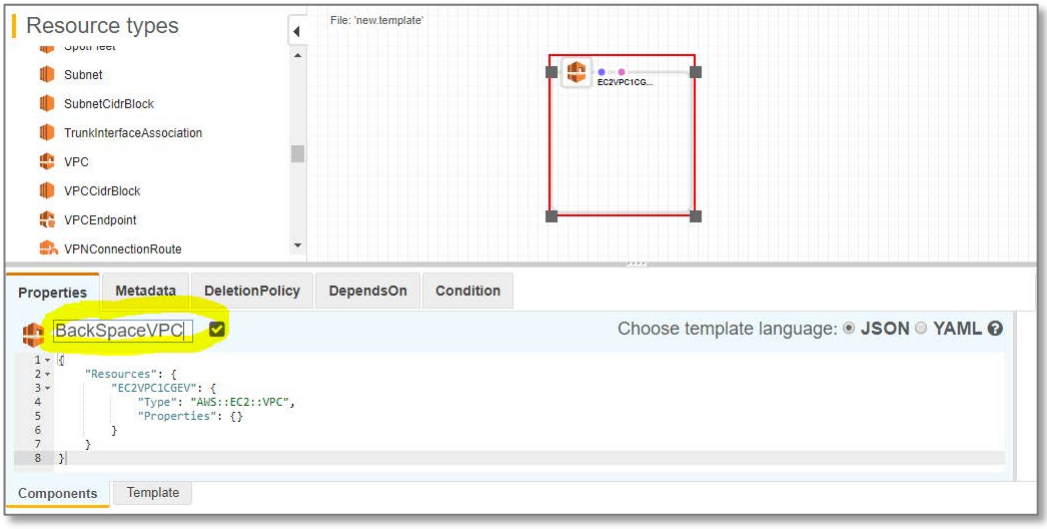


Scroll down to “VPC”

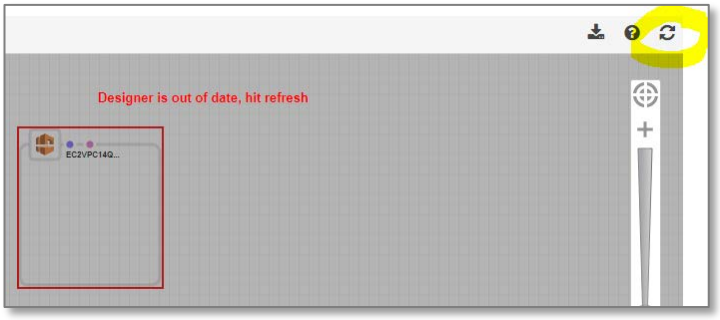
Drag and drop the VPC icon onto the canvas



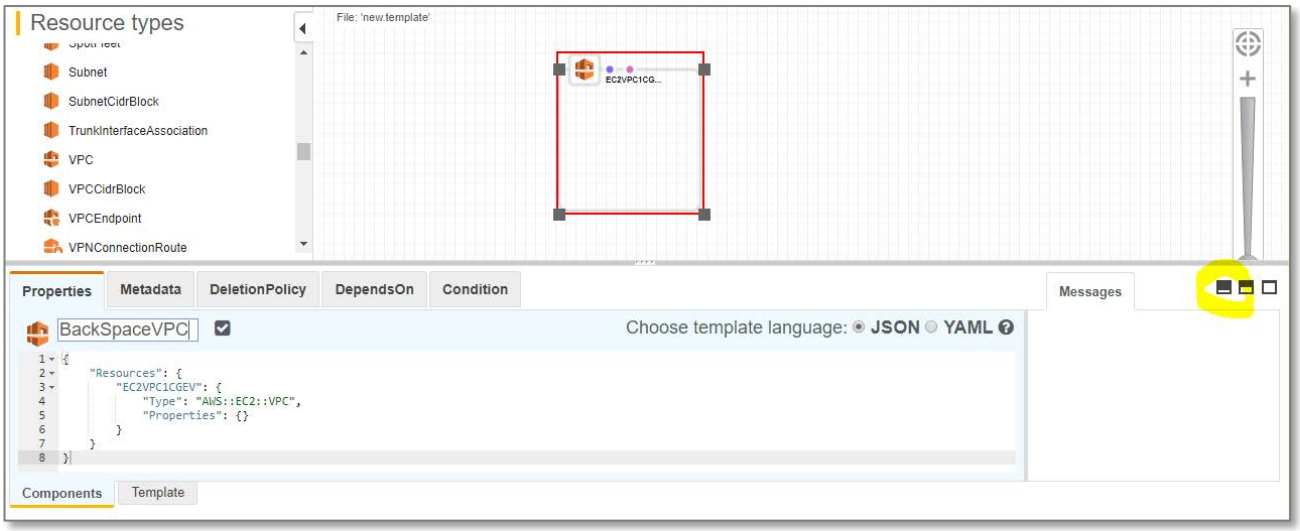
Go to the editor pane and give the VPC a new name



Press enter and refresh the canvas



Minimise the Editor



Resize the VPC

Go back to Split Screen view to see the editor pane

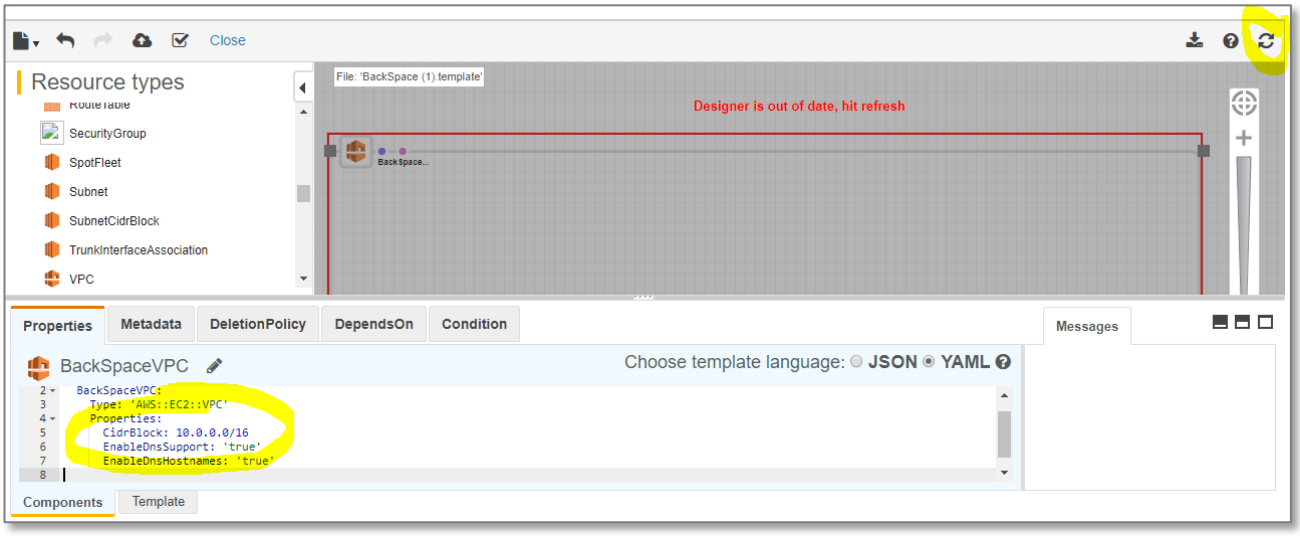
Select YAML

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

CidrBlock: 10.0.0.0/16

EnableDnsSupport: 'true'

EnableDnsHostnames: 'true'



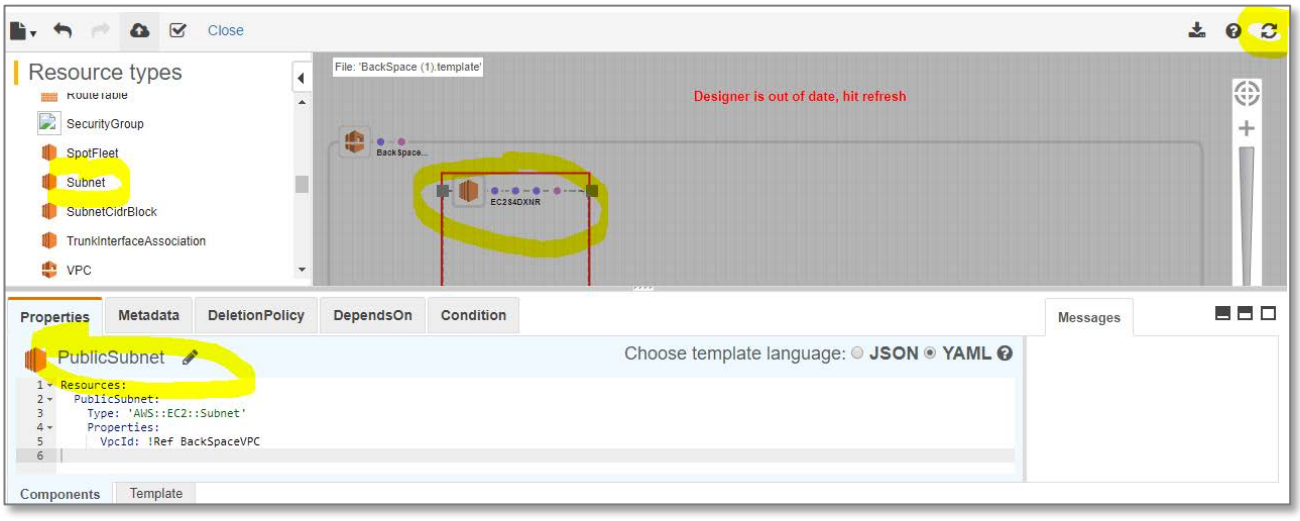
Drag the Subnet icon onto the VPC

Resize the subnet

Change the Subnet name to “PublicSubnet”

Click to Refresh Canvas

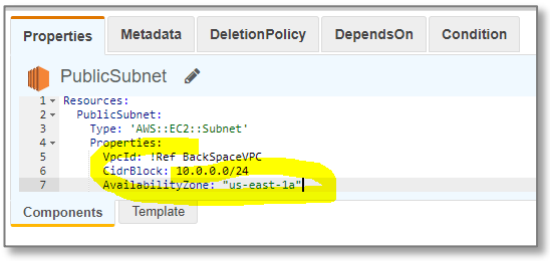
***Troubleshooting Note – If the subnet doesn’t appear in the editor, click on the template tab, then the components tab again to reload it.**



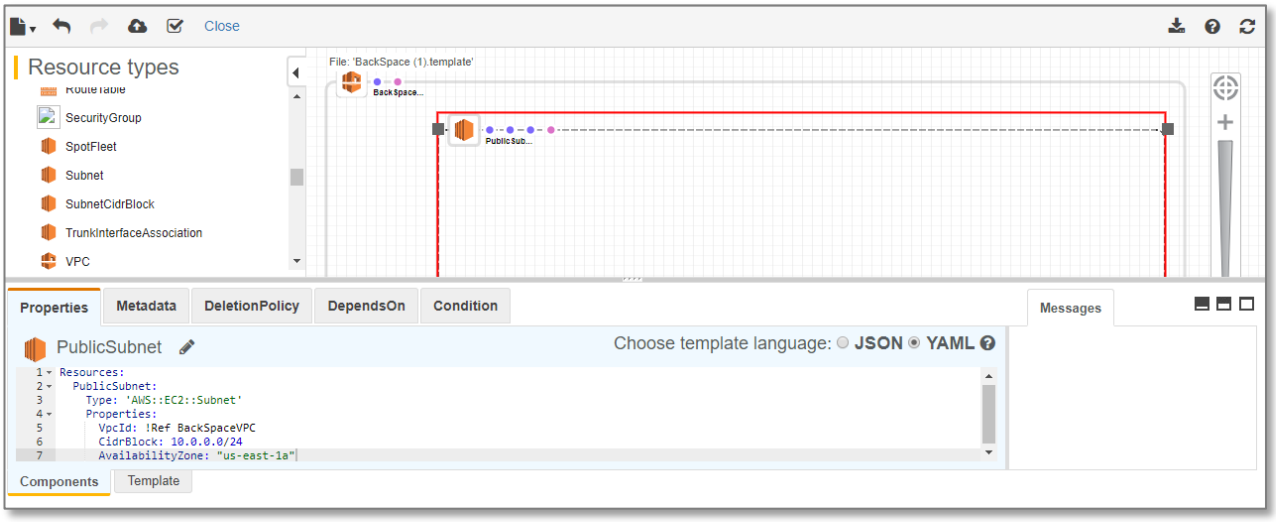
Add the following properties to the Subnet

CidrBlock: 10.0.0.0/24

AvailabilityZone: "us-east-1a"



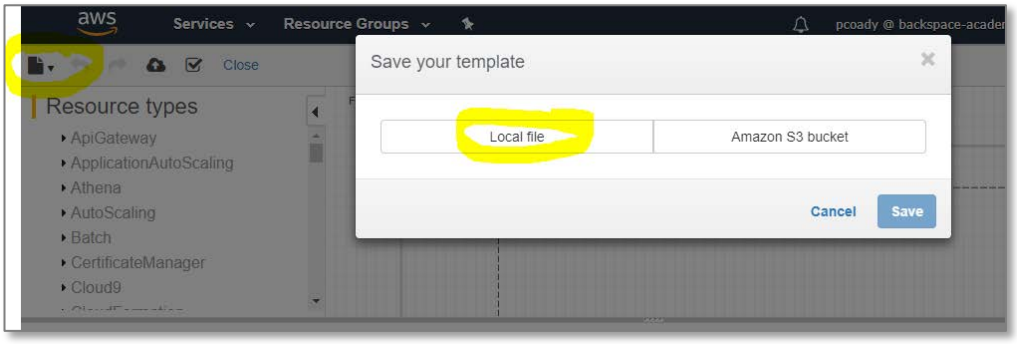
Click to refresh canvas



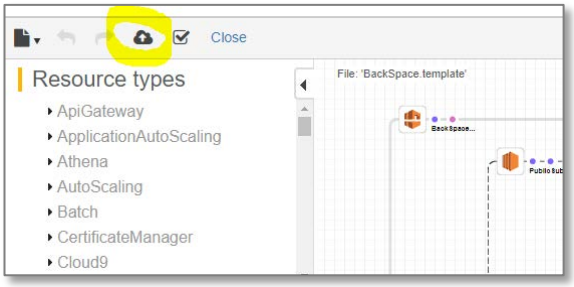
Deploying the Architecture with CloudFormation

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

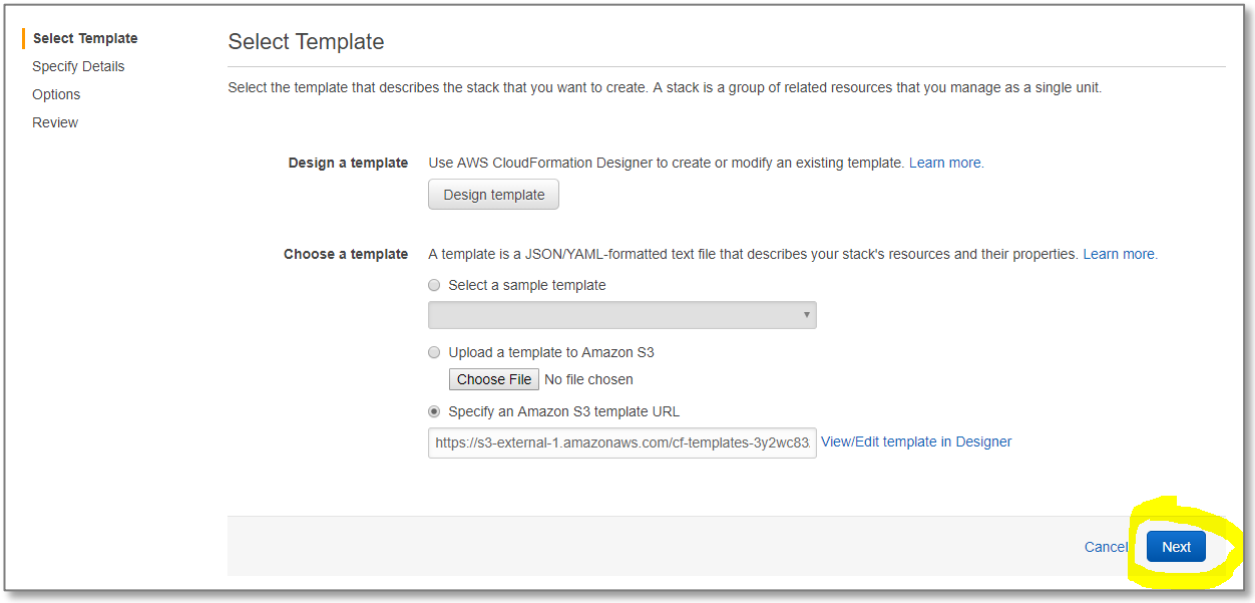


Click “Create Stack”

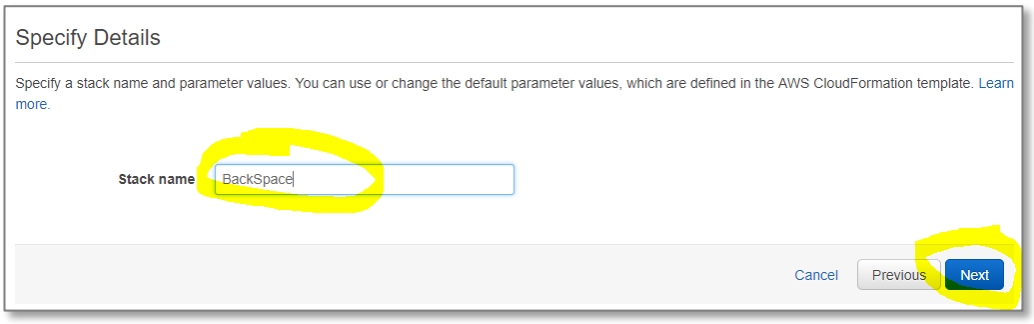


Click Next

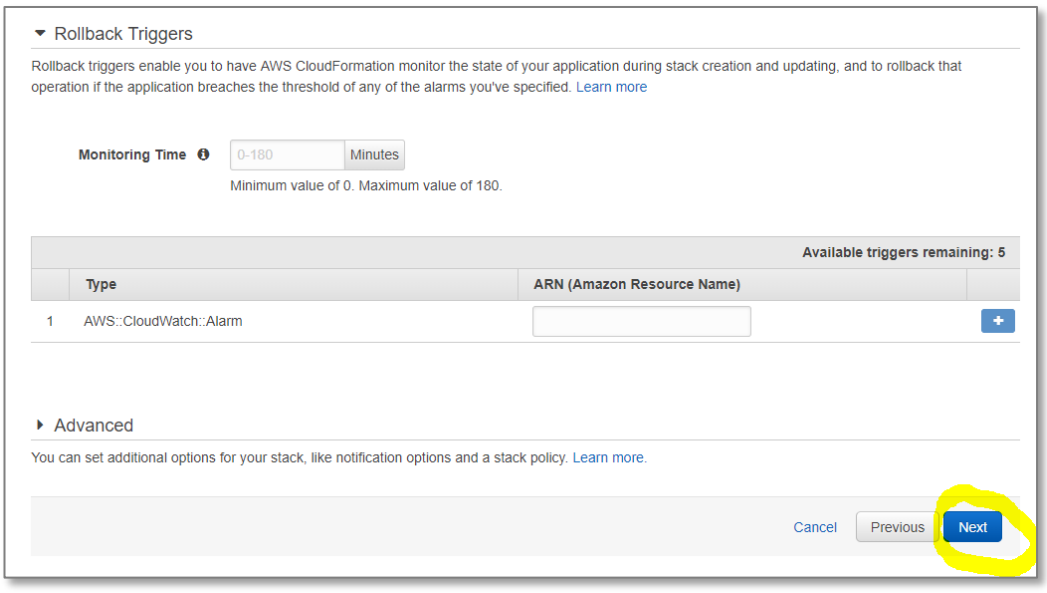
*Troubleshooting Note – If the S3 URL doesn’t work use “Upload a template to Amazon S3” and select the saved template



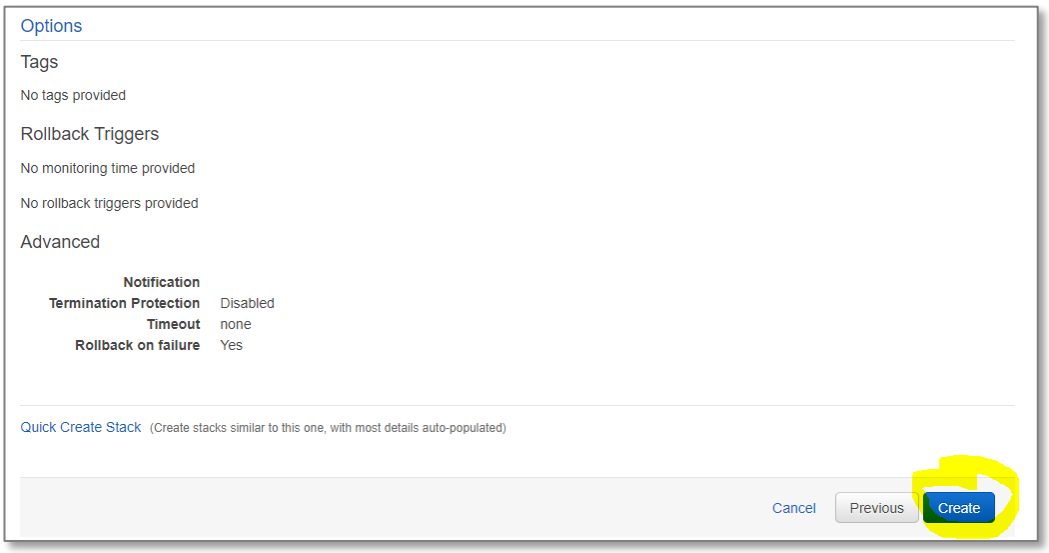
Give the stack a name
Click Next



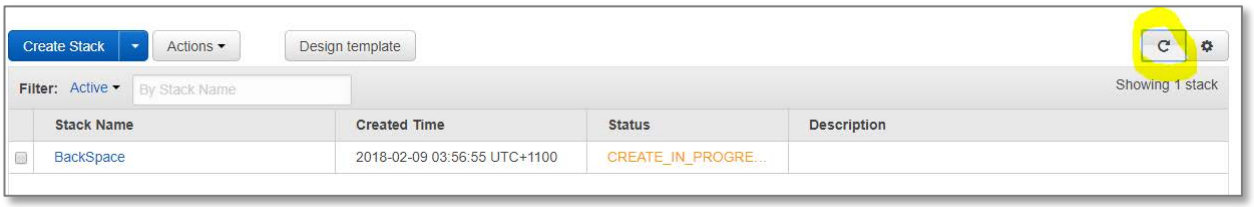
Click Next again



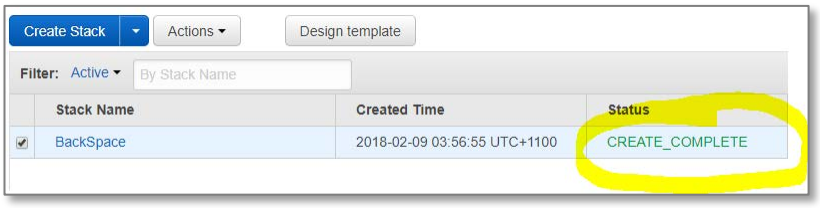
Click Create



Click the refresh icon to see status



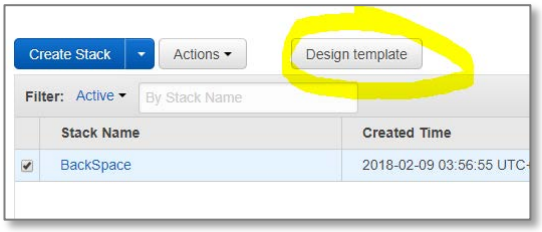
If successful you see “CREATE_COMPLETE”



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

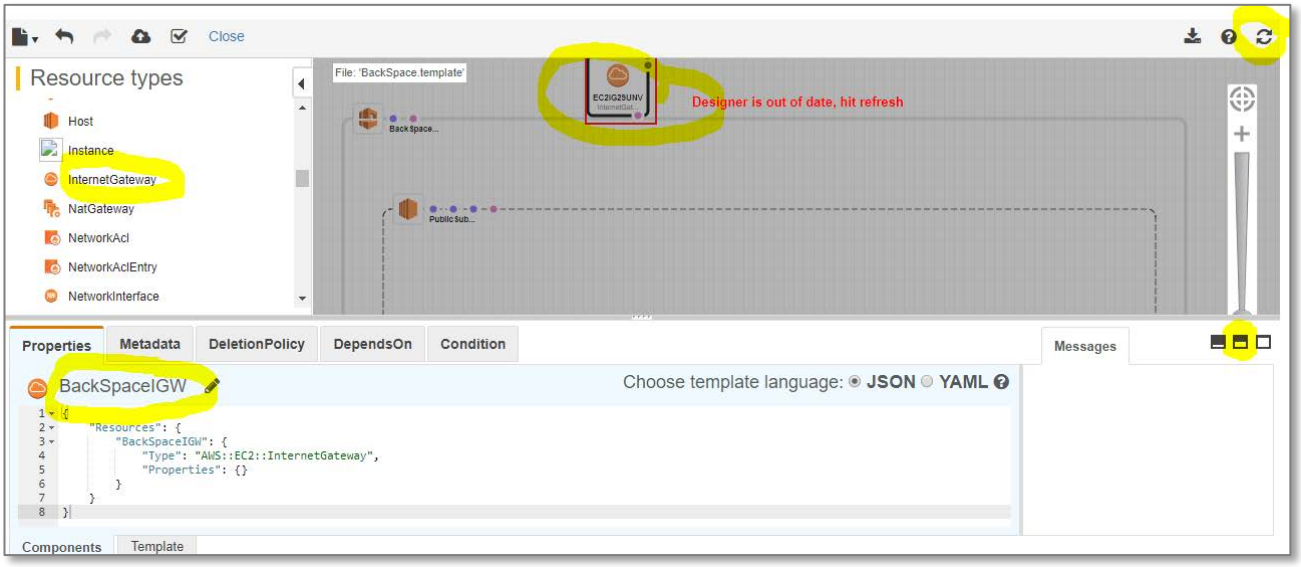
Open the local template file you saved

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

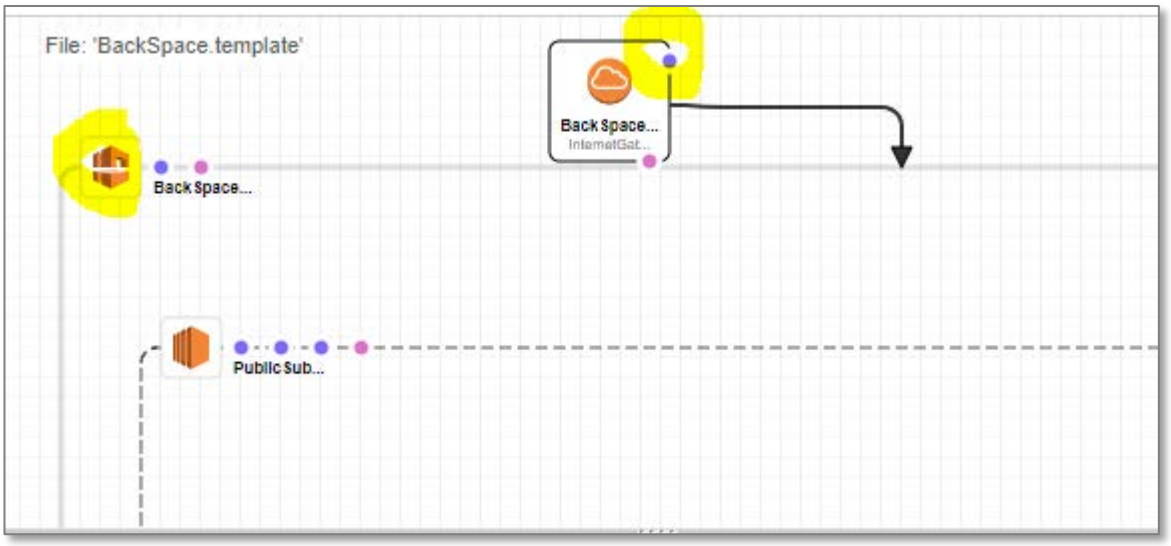
Rename the IGW and press enter

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

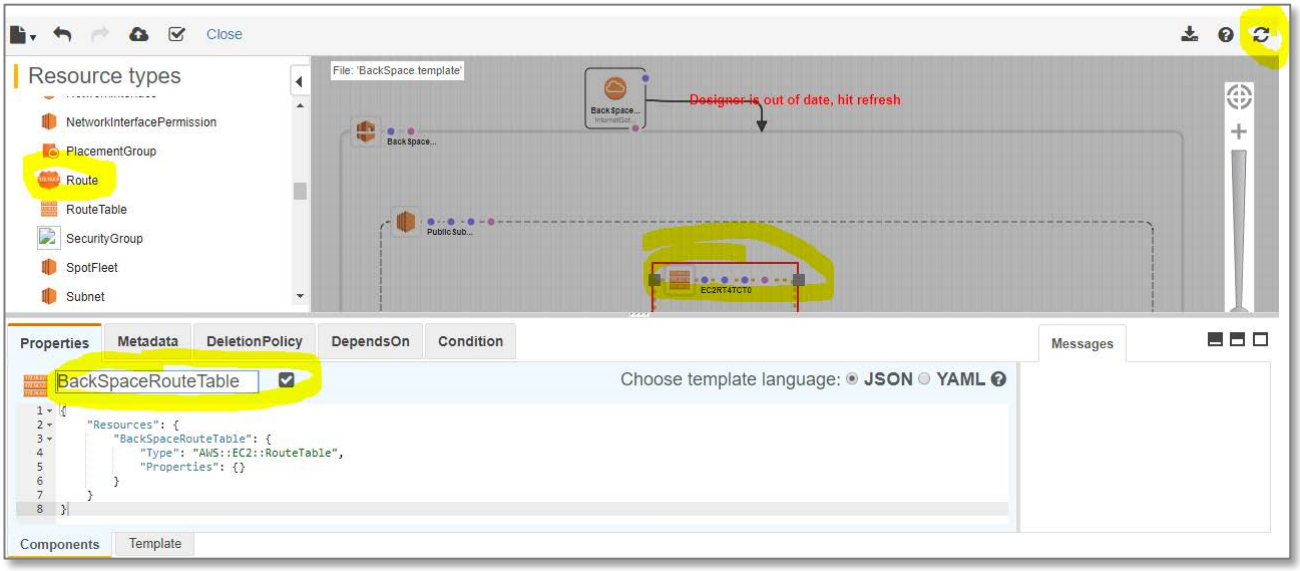


Drag and drop a RouteTable icon onto the subnet

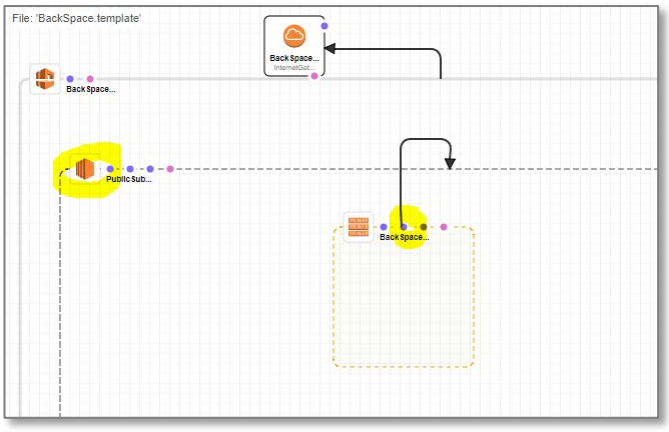
Rename the Route table and press enter

Click to refresh the canvas

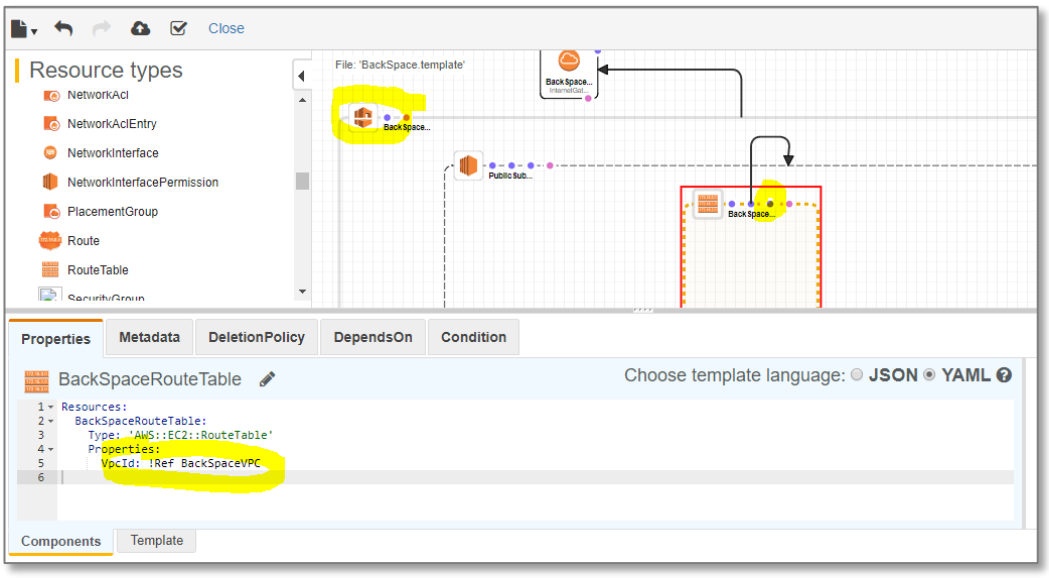
***Troubleshooting Note – If the RouteTable doesn't appear in the editor, click on the template tab, then the components tab again to reload it.**



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



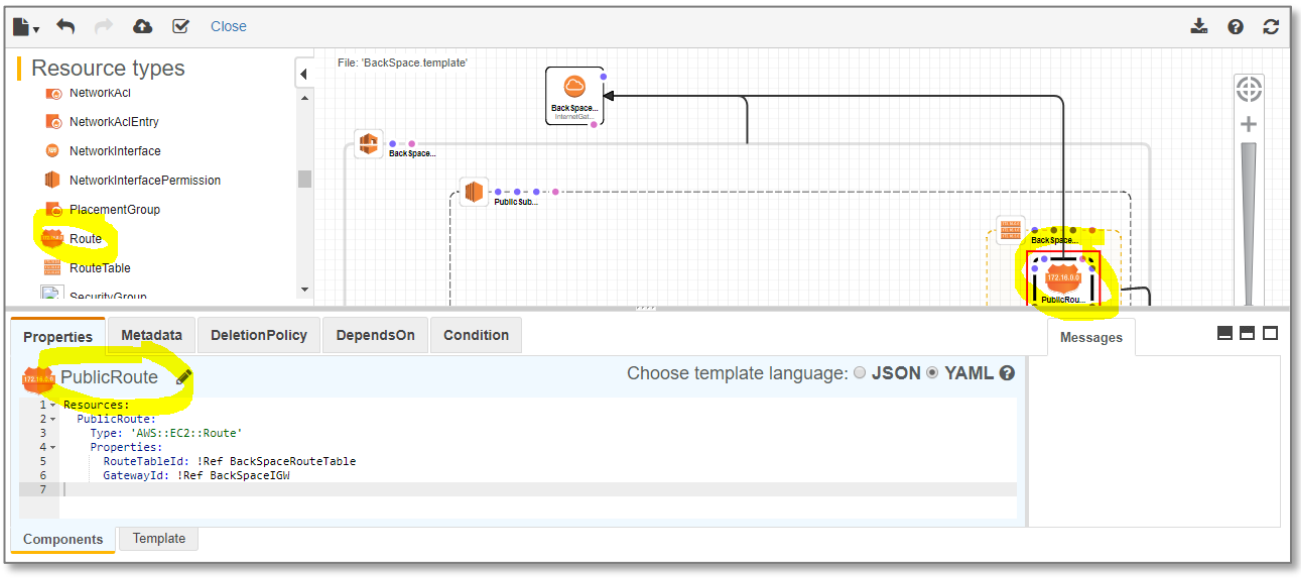
Drag and drop the third blue dot on the RouteTable icon onto the VPC to associate the Route Table with the VPC



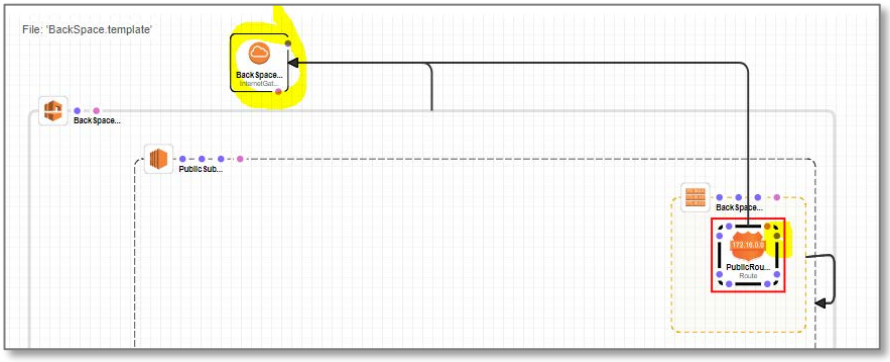
Drag and drop a Route icon onto the RouteTable

Rename the Route as “PublicRoute” and press enter

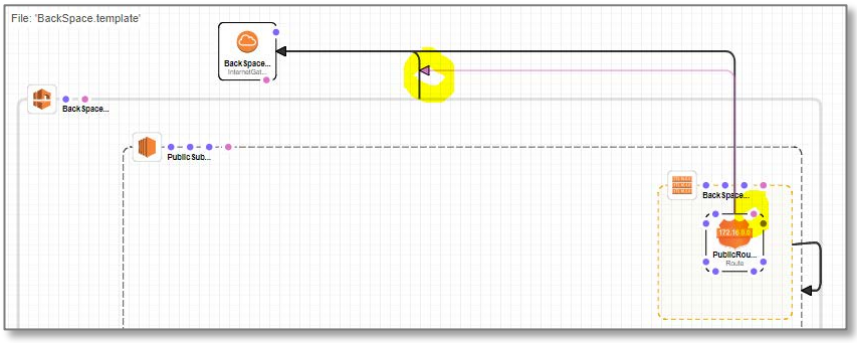
Click to refresh the canvas



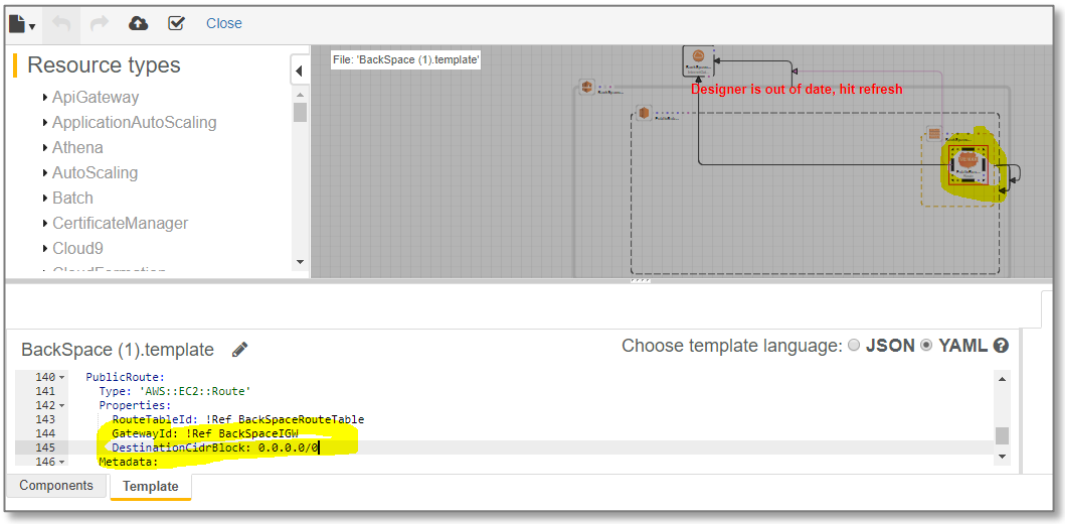
Drag and drop from the blue GatewayID (not the EgressOnlyInternetGateway) dot to the Internet Gateway



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

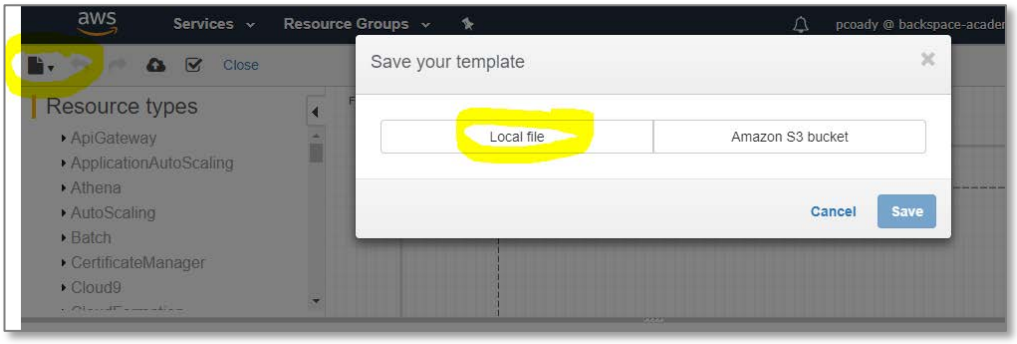


You have created an Internet gateway (IGW) and a route from the subnet to the IGW. This is now a public subnet.

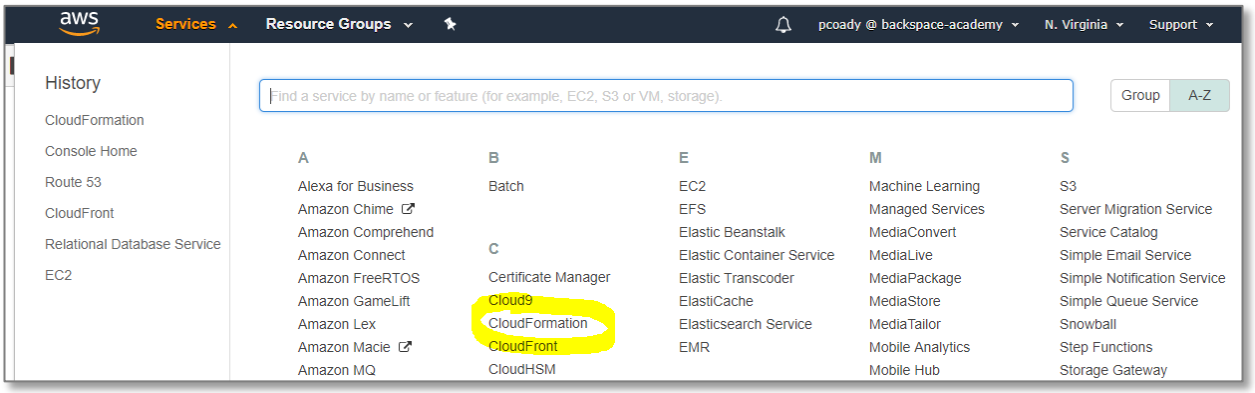
Updating the Architecture with CloudFormation

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

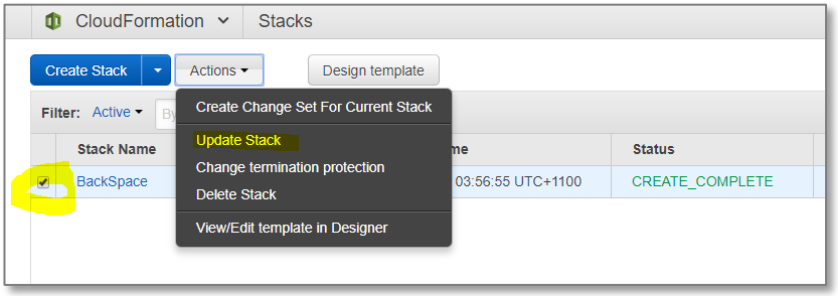


Go to Services – CloudFormation



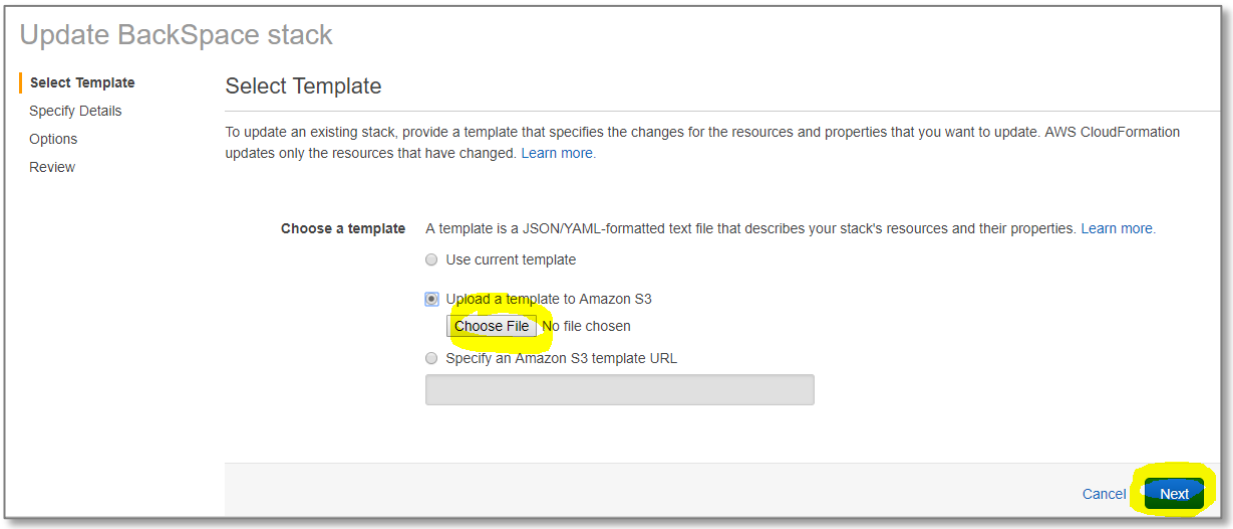
Select the stack

Select Actions – Update Stack

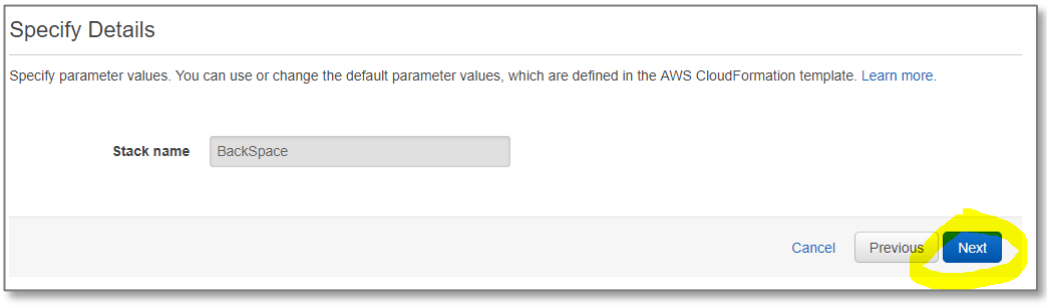


Upload the newly saved template version

Click Next

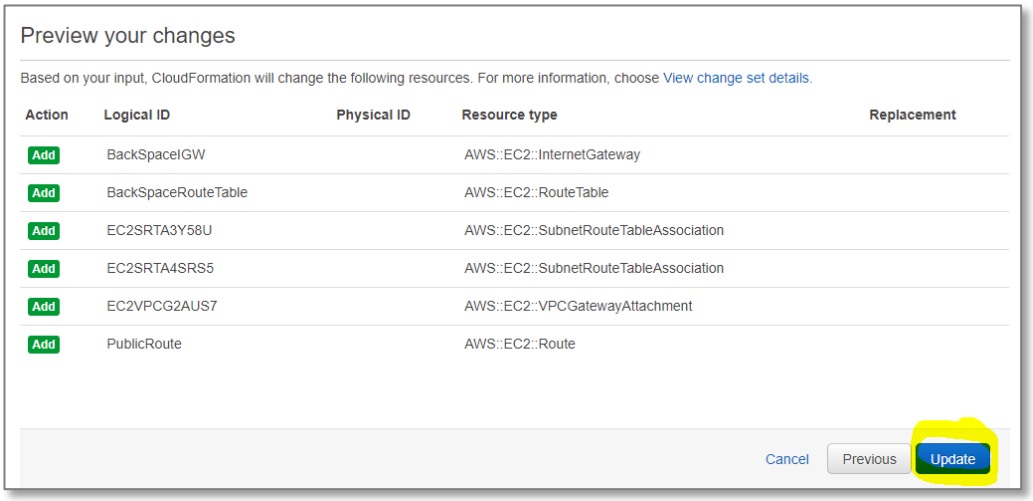


Click Next again

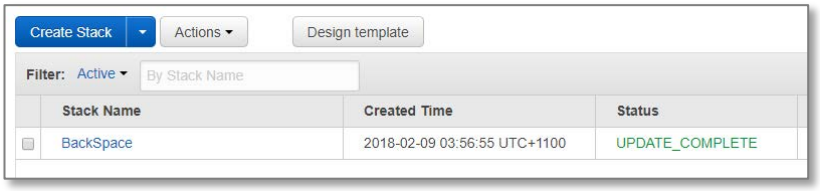


Click Next again

Click Update



If successful you will see “UPDATE_COMPLETE”



Creating an EC2 Instance and Security Group

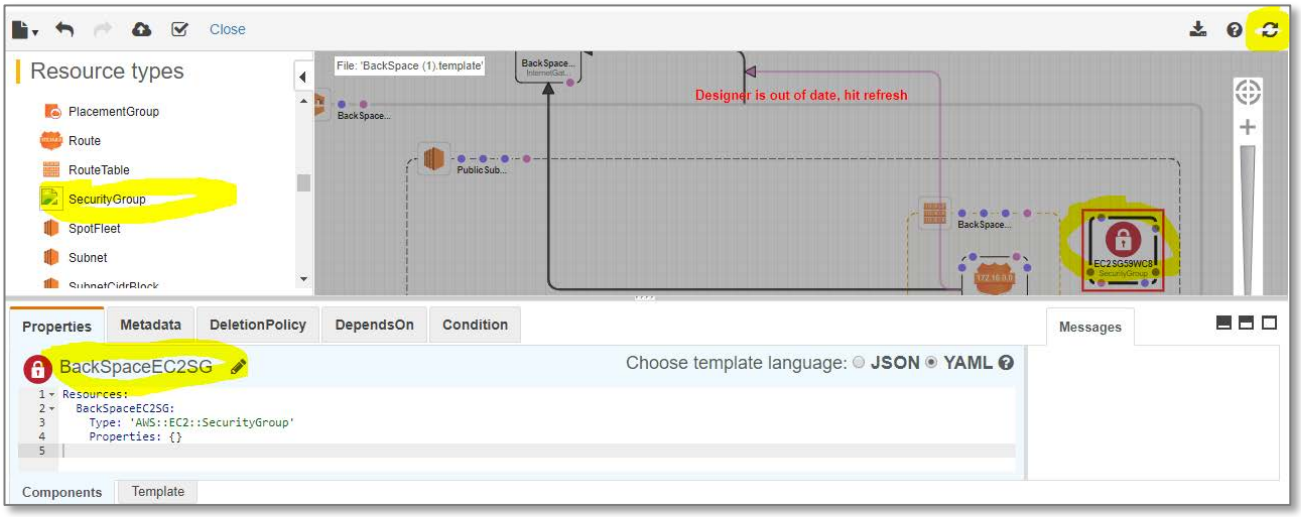
In this section, we will use the AWS CloudFormation Designer to create an EC2 Instance and Security Group.

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'

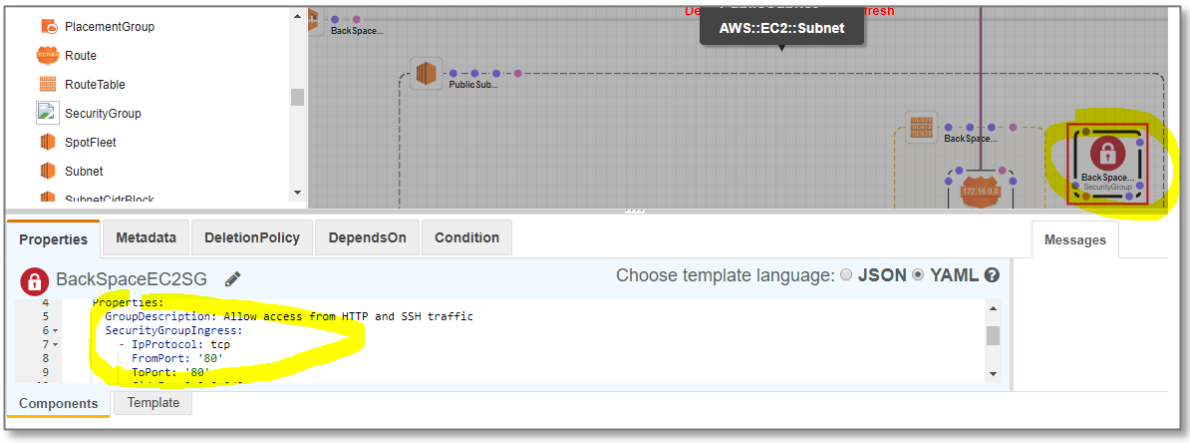
CidrIp: 0.0.0.0/0

- IpProtocol: tcp

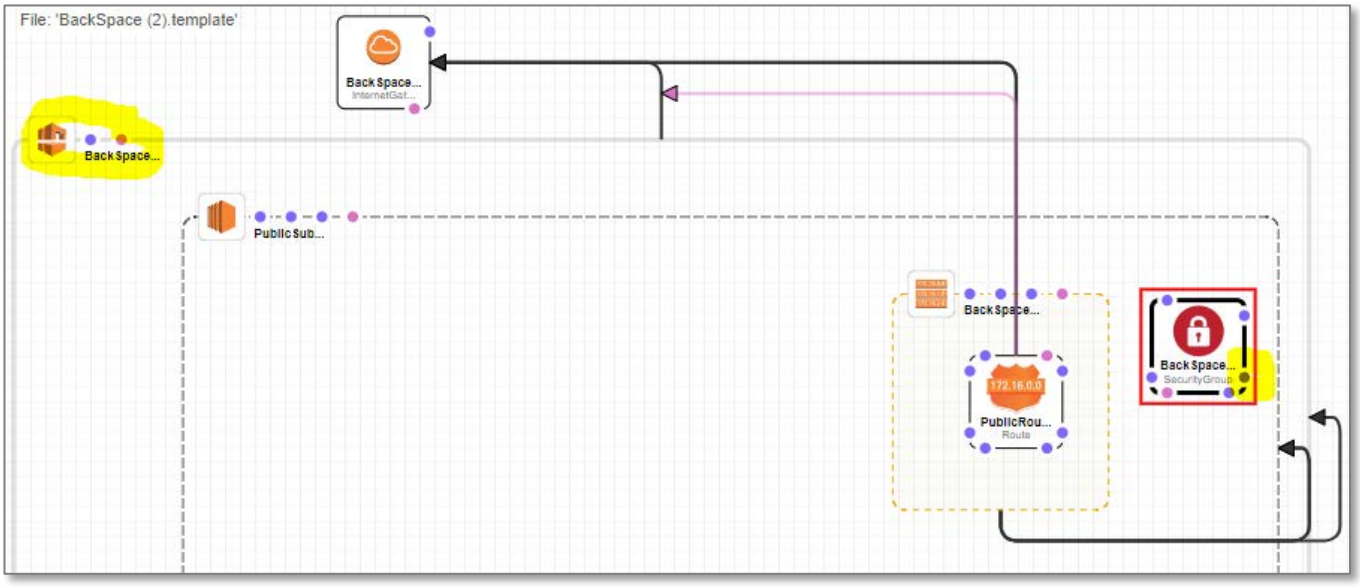
  FromPort: '22'

  ToPort: '22'

  CidrIp: 0.0.0.0/0



Drag and drop from the blue VpcID dot to the VPC

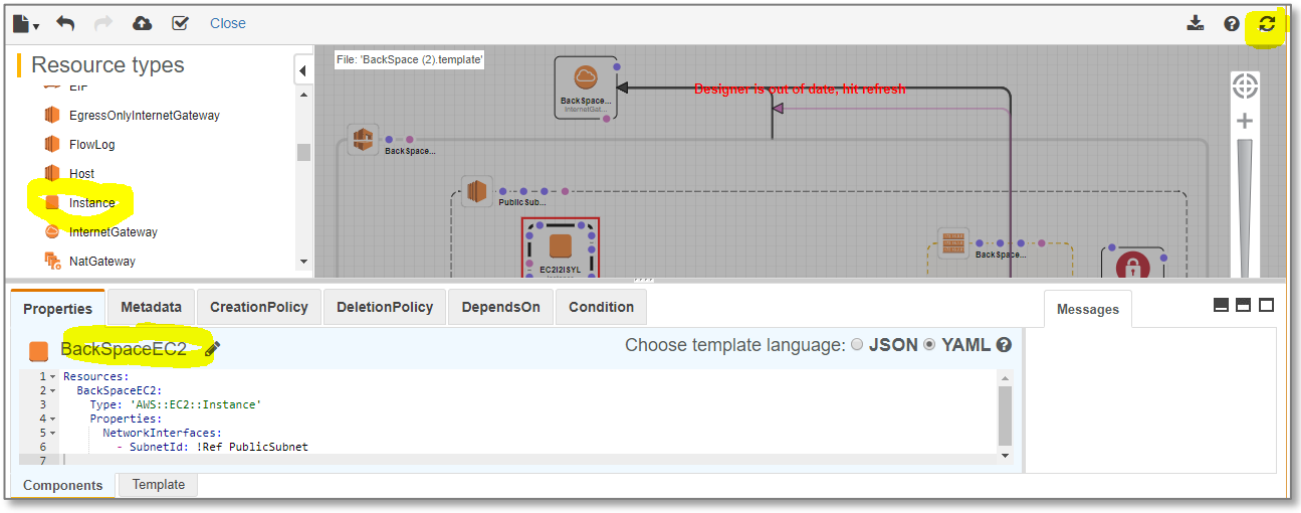


Create EC2 Instance

Drag and drop an Instance onto the subnet.

Rename it “BackSpaceEC2” and press enter

Click to refresh canvas



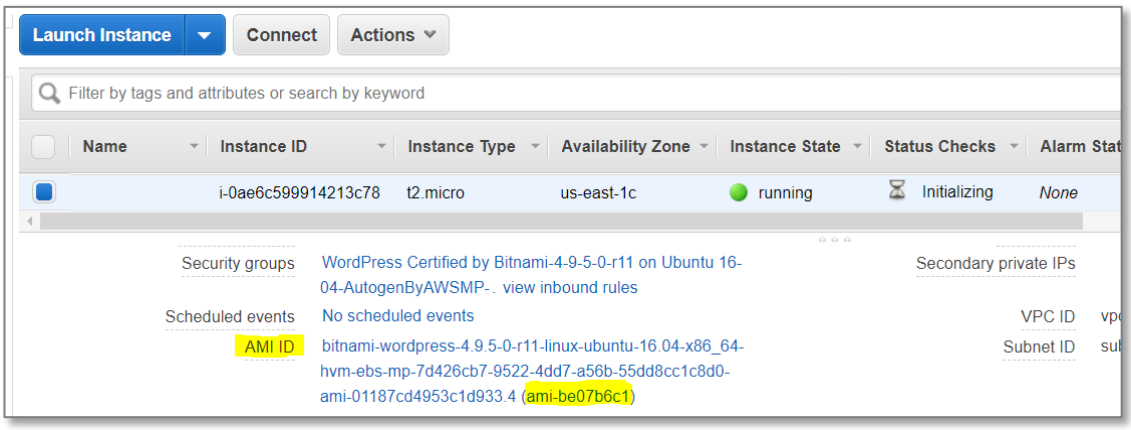
Create a “DependsOn” link from the pink dot to the PublicRoute

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:



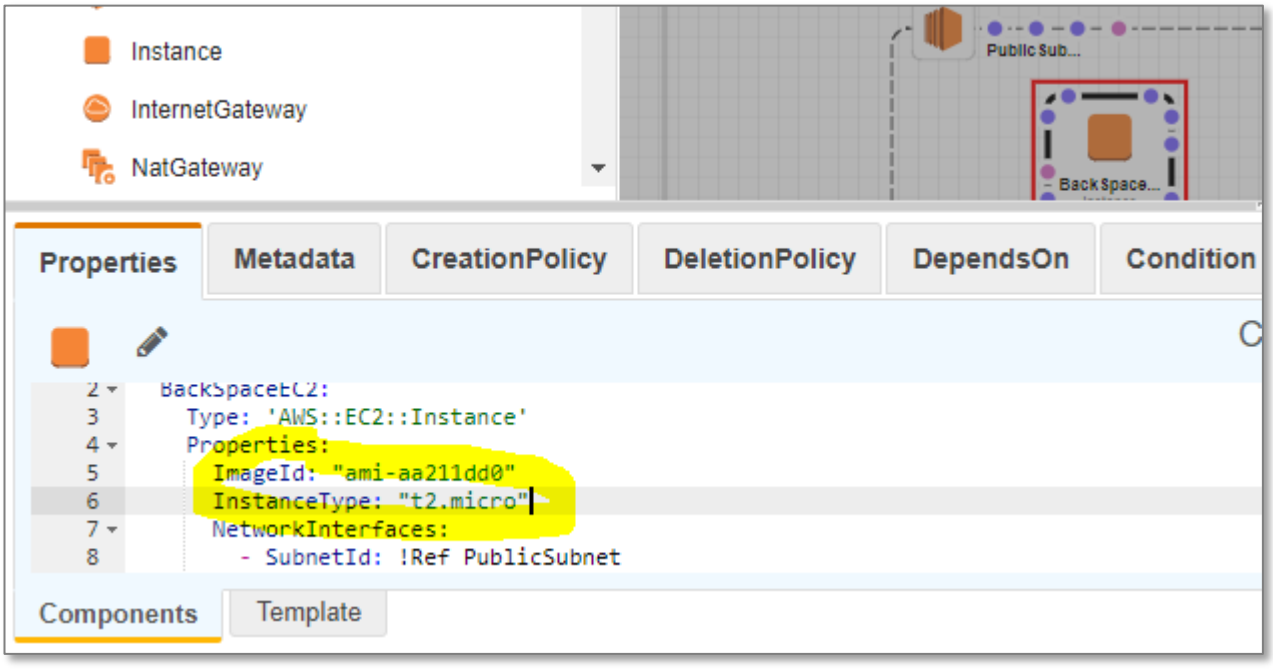
Terminate the instance after you have copied the AMI ID so that you don't get billed.

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'

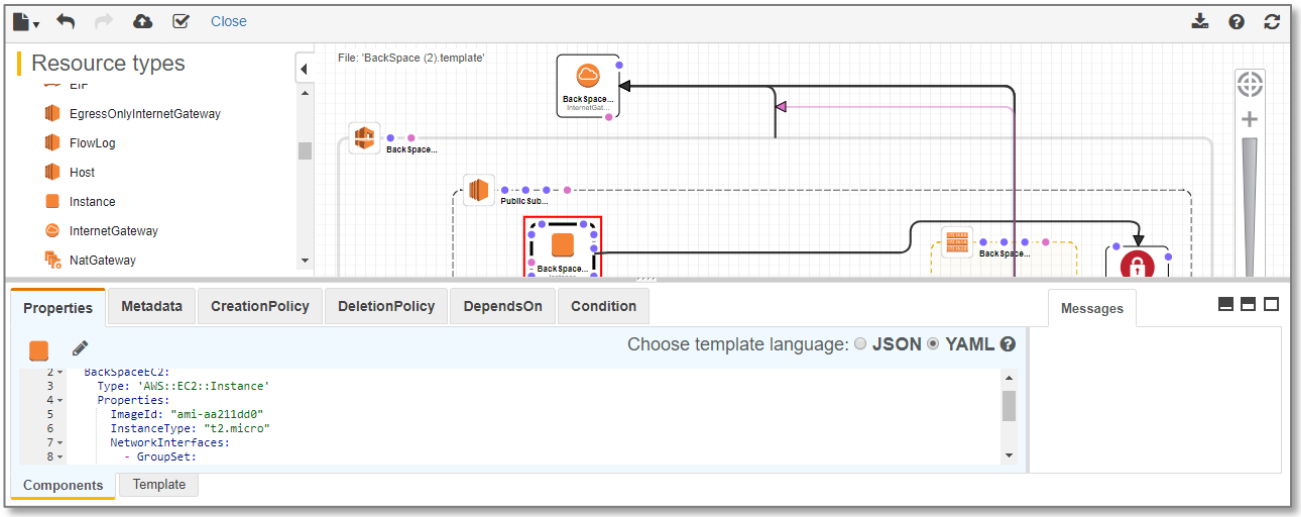
DeviceIndex: '0'

DeleteOnTermination: 'true'

SubnetId: !Ref PublicSubnet

```
BackSpaceEC2:
  Type: 'AWS::EC2::Instance'
  Properties:
    ImageId: "ami-aa211dd0"
    InstanceType: "t2.micro"
    NetworkInterfaces:
      - GroupSet:
          - !Ref BackSpaceEC2SG
        AssociatePublicIpAddress: 'true'
        DeviceIndex: '0'
        DeleteOnTermination: 'true'
        SubnetId: !Ref PublicSubnet
```

Click to refresh canvas

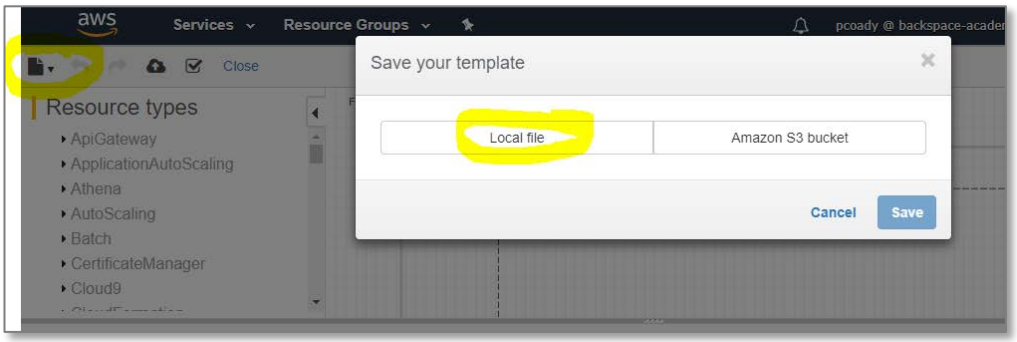


Deploying the Wordpress Server

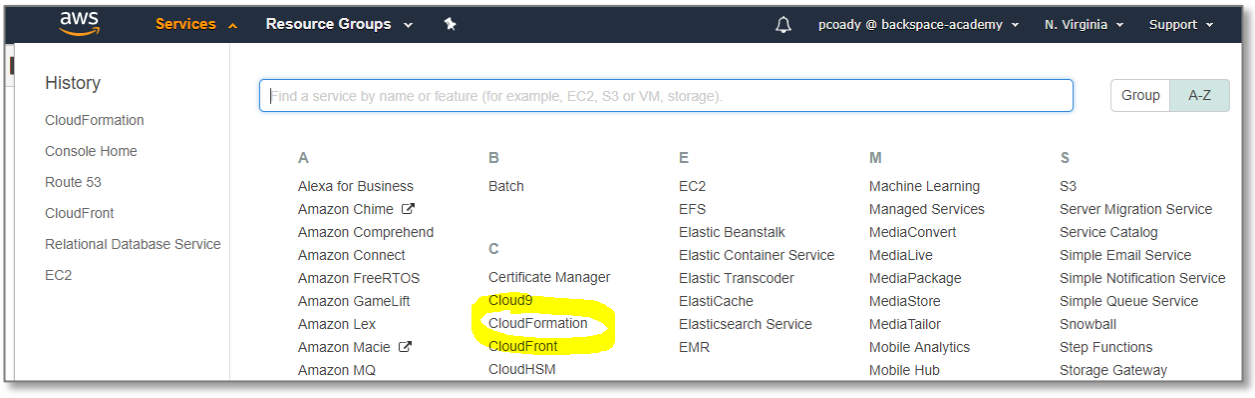
In this section, we will use the AWS CloudFormation to update our stack deploy our WordPress Server.

Save the template as a new version

Select File – Save
Select Local File
Save the File as a different version

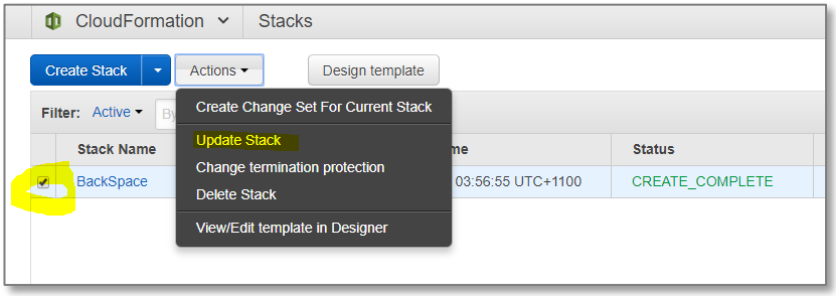


Go to Services – CloudFormation



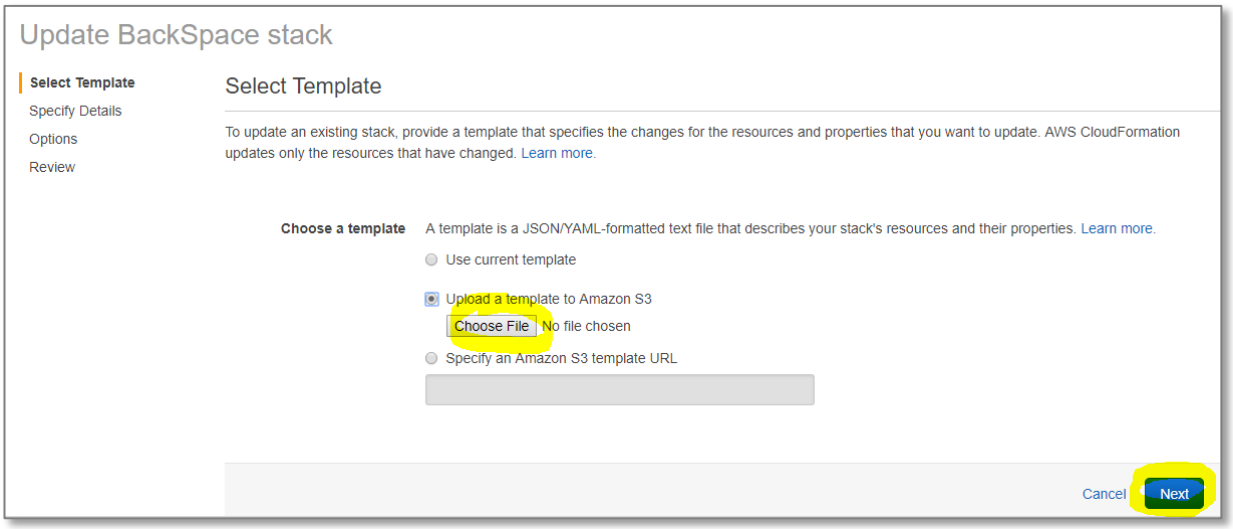
Select the stack

Select Actions – Update Stack

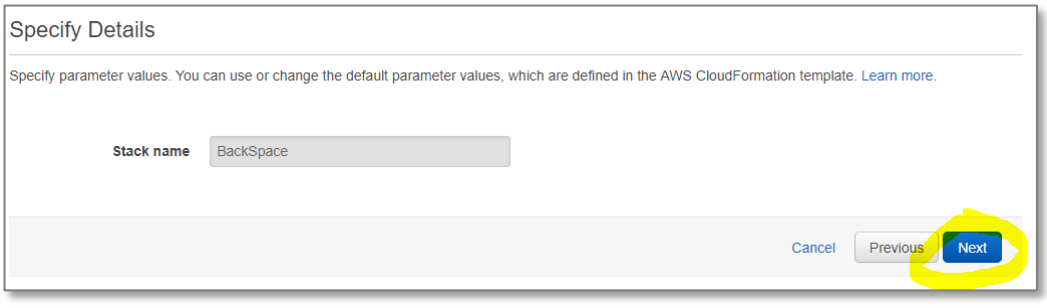


Upload the newly saved template version

Click Next



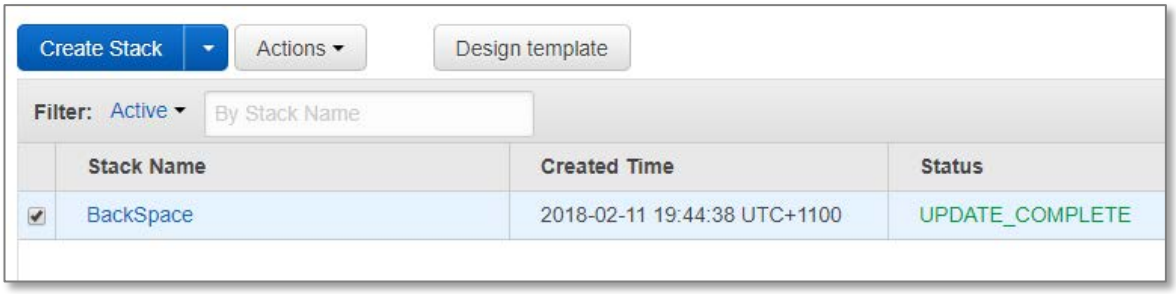
Click Next again



Click Next again

Click Update

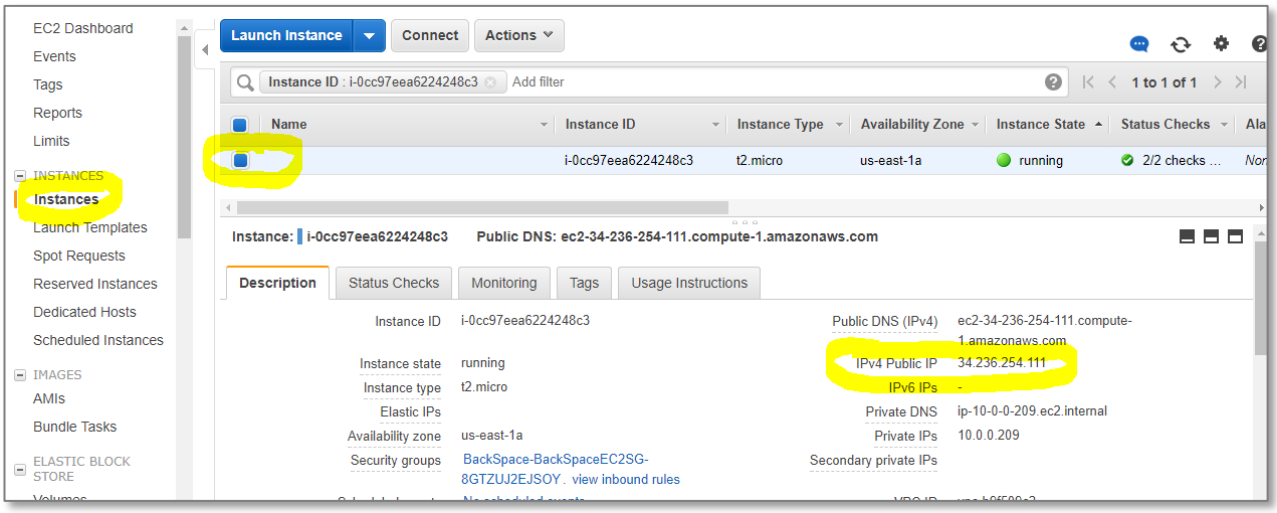
Wait until stack has been updated.



Viewing your WordPress Server

Go to the EC2 console and view the instance

Copy the public IP address



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.