Use the Cloud9 Environment

The company you work for is constantly deploying applications to AWS. However, the process is currently all manual and involves clicking through the AWS Console. It takes quite a while.

In order to reduce this time and avoid human errors, you were asked to write a CloudFormation template to automate the provisioning and deployment process.

The base infrastructure used for these applications is composed of a Reverse Proxy Server and a Web Server - both powered by Nginx.

The Reverse Proxy Server is responsible for receiving client requests from the internet and routing them to the Web Server. For security purposes, the Proxy and Web servers are deployed to different subnets. The former is hosted on an EC2 instance in a public subnet. The latter, on an EC2 instance in a private subnet. This way, the Web Server is not directly exposed to the internet, which reduces the attack surface.

This will guide us through the steps necessary to write and run CloudFormation templates that create and deploy our infrastructure.

In this first challenge, we will get started with Cloud9, the web-based environment we will use to write our templates.

1. Go to the AWS Console and navigate to the **AWS Cloud9** service.
2. On the AWS Cloud9 dashboard, there should be one environment. Take note and store the name of this environment. We will use it to name our CloudFormation stacks.

Note: The name of the environment should start with ***PS-Labs-*** and be followed by 5 random characters. For example: ***PS-Labs-LFWHN*** or ***PS-Labs-GVUOV***, etc.

3. Click the **Open IDE** button. A new browser tab will open with a browser-based IDE. Wait a few moments so the project files are loaded.

   Note: If you see a pop-up alert saying ***"Unable to register Service Workers due to browser restrictions, Webviews will not work properly"***, it's safe to ignore it. This will not affect the Lab.

4. Once loaded, click the **templates** folder. We'll be editing files inside this folder shortly, but before we do that, let's make sure our AWS CLI is properly configured.

5. There are a few terminal tabs towards the bottom of the screen. Find the one that's open. That's where we'll be running our commands using the AWS CLI.

6. Inside the terminal tab, type the following command to move inside the **templates** folder:

   ```
   cd templates
   ```

7. Next, type the following command to confirm we can communicate with AWS using the AWS CLI:

   ```
   aws ec2 describe-vpcs
   ```

   This command should display a JSON output with a single VPC. If the VPC was successfully displayed in the results, this means the environment is properly configured and we can continue.

8. In order to avoid repeating the name of the stack every time we run the CloudFormation CLI, we'll use the environment name we just copied from the previous tab to run the following command in the terminal:

```
export INFRA_STACK=<name-of-the-environment>
```

So, for example, if the name of the environment was *PS-Labs-GVUOV*, then the command would be:

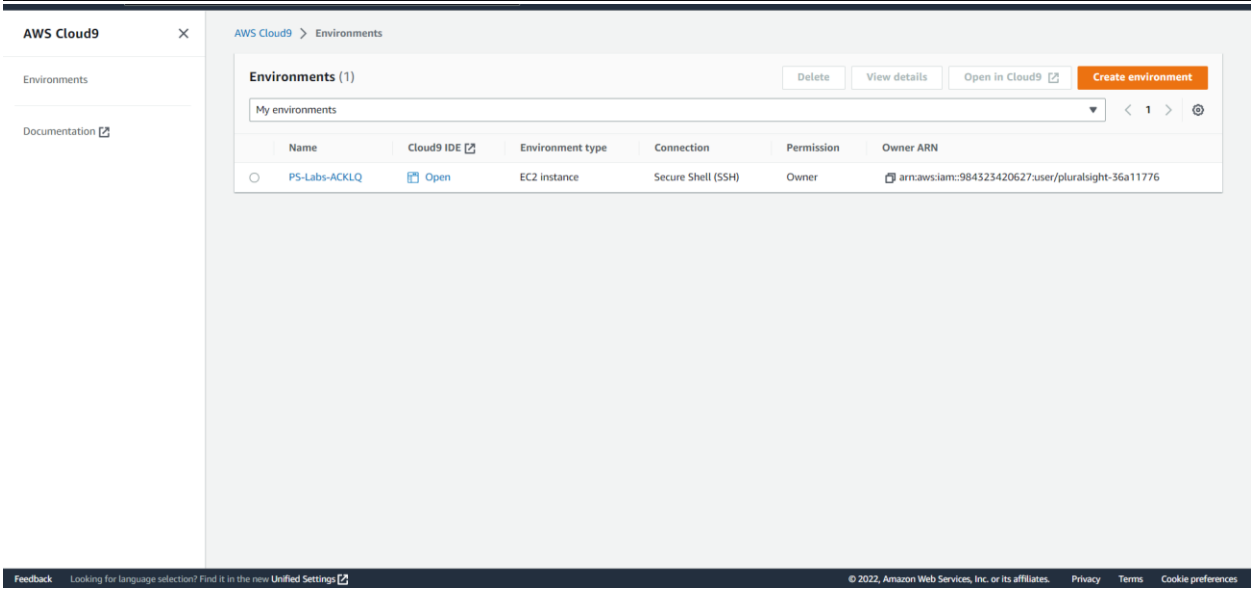
```
export INFRA_STACK=PS-Labs-GVUOV
```

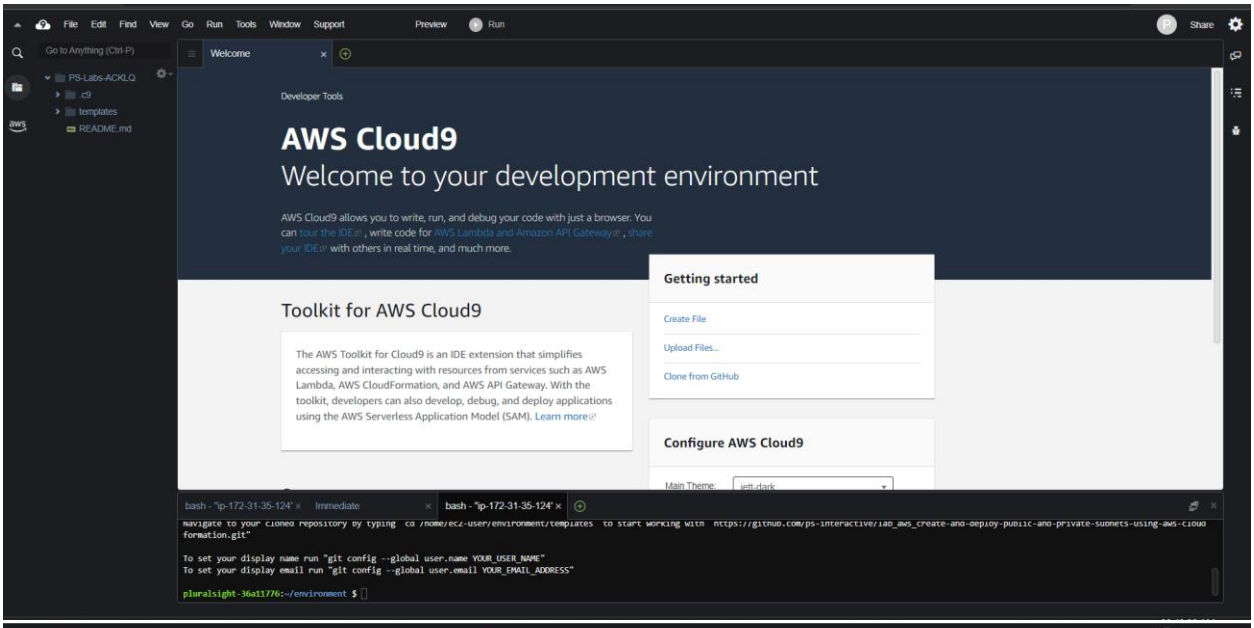
9. To check the previous command, run the following:

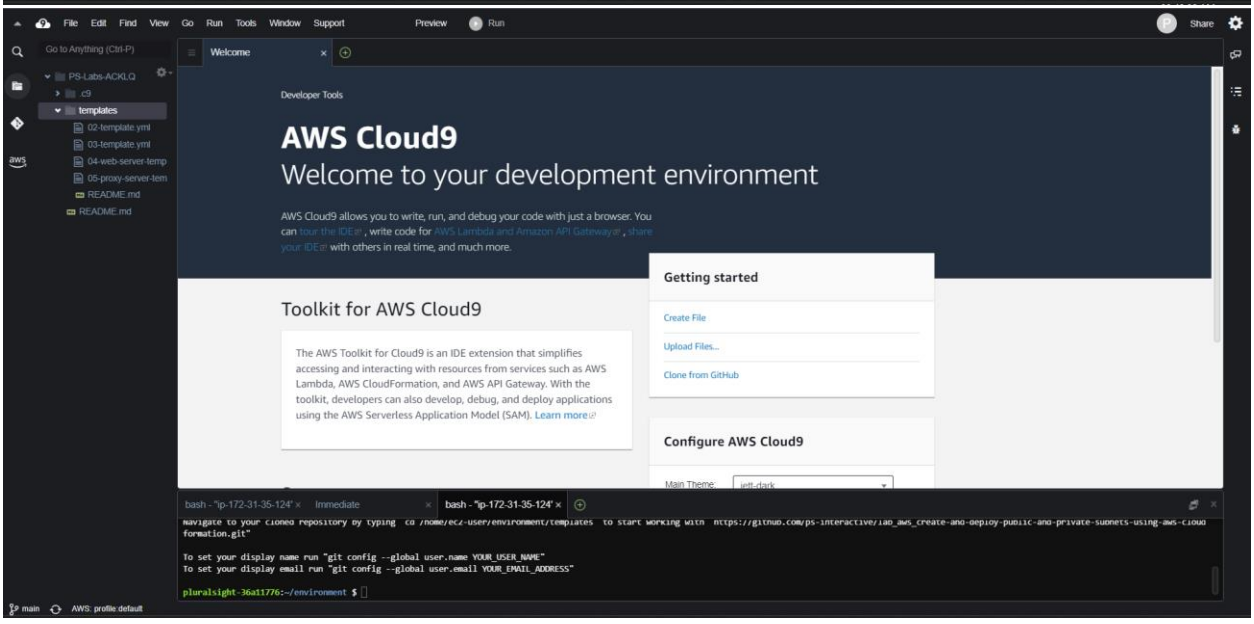
```
echo $INFRA_STACK
```

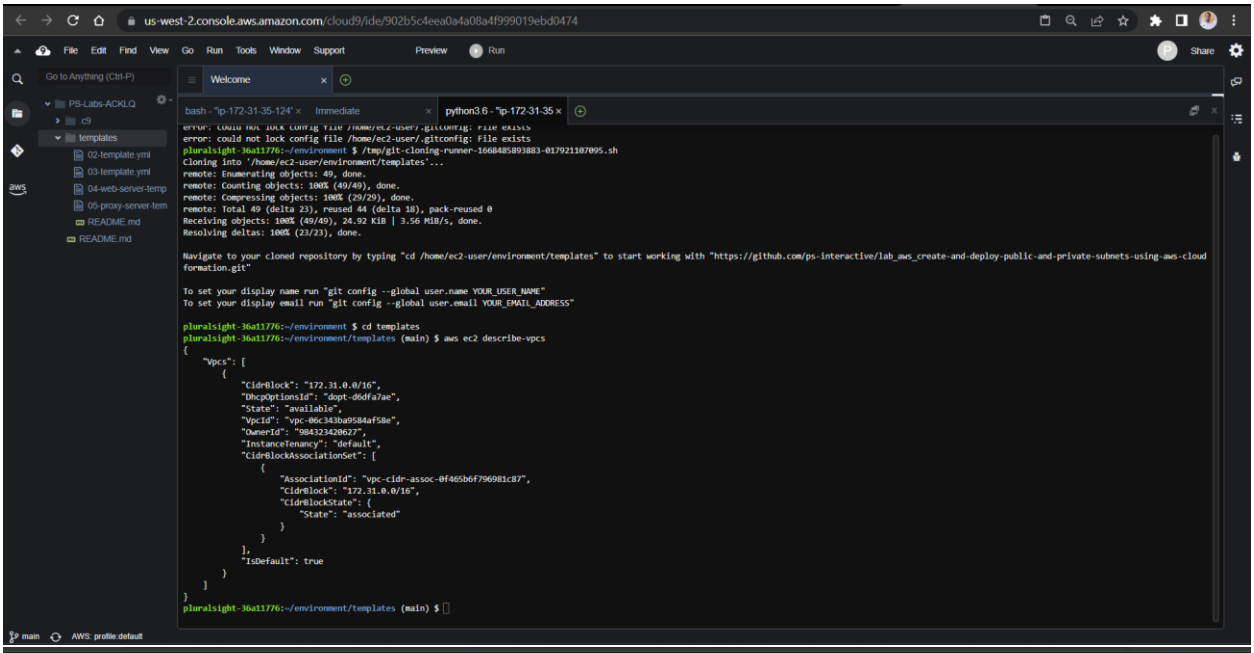
This command should display the name that we just stored.

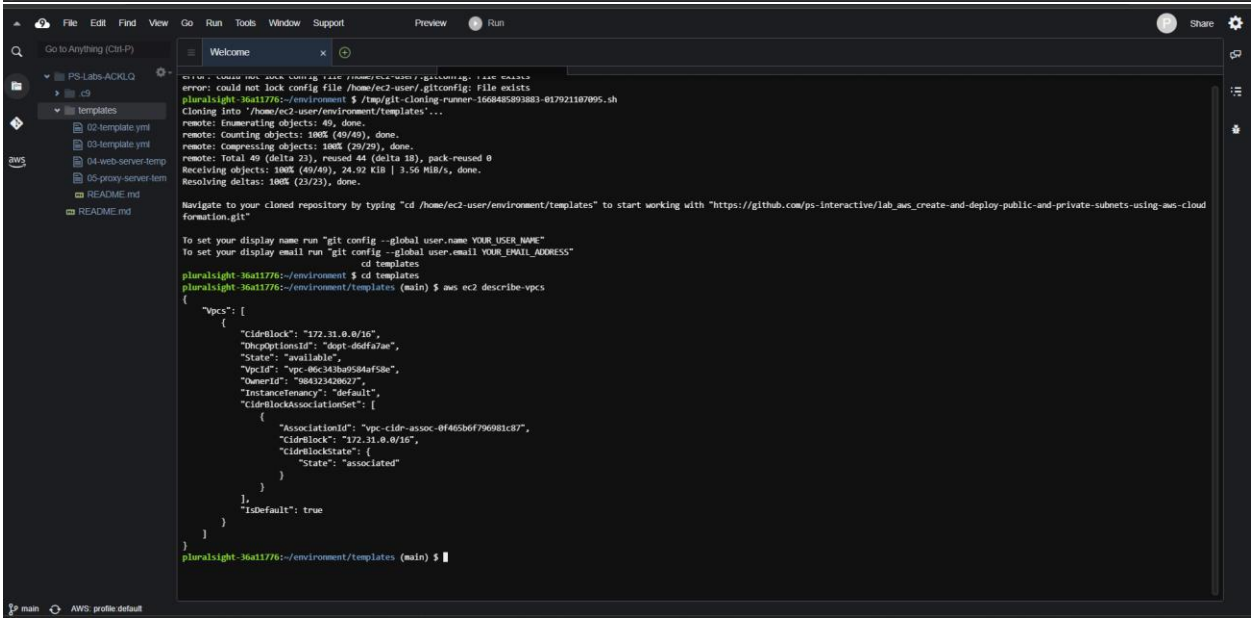
If the previous command successfully displayed back the name of the environment, then congratulations! We are all set up and ready to start creating our infrastructure. Move on to the next challenge.

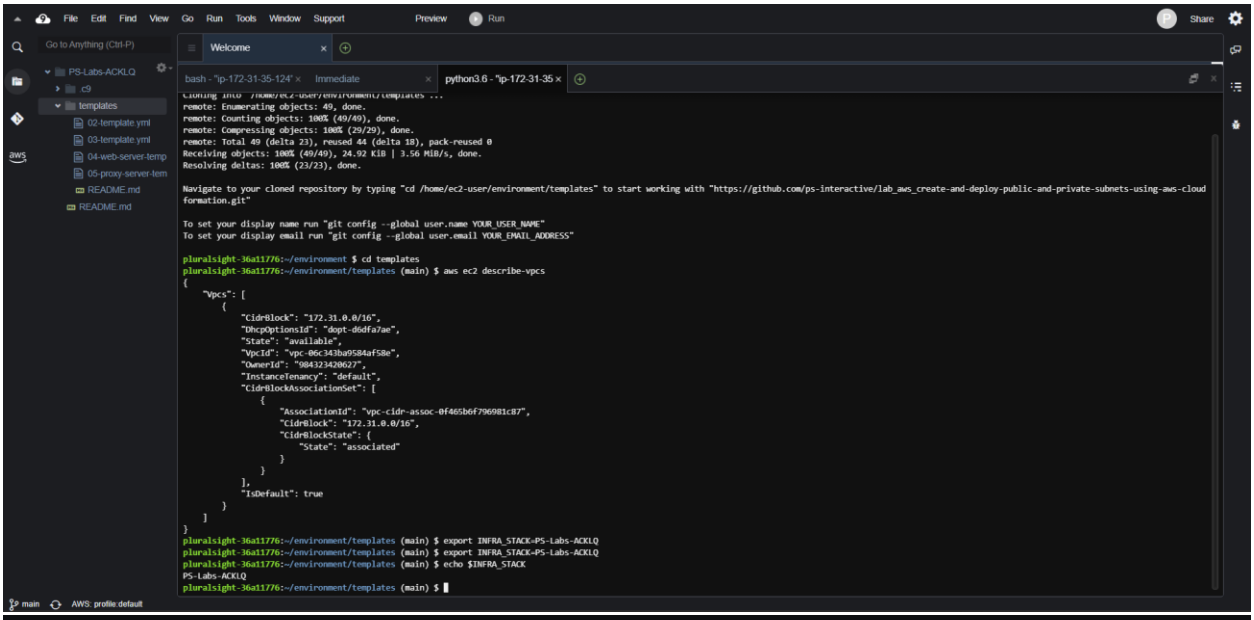



File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
README.md
Welcome
Developer Tools
AWS Cloud9
Welcome to your development environment
AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.
Toolkit for AWS Cloud9
The AWS Toolkit for Cloud9 is an IDE extension that simplifies accessing and interacting with resources from services such as AWS Lambda, AWS CloudFormation, and AWS API Gateway. With the toolkit, developers can also develop, debug, and deploy applications using the AWS Serverless Application Model (SAM). Learn more
Getting started
Create File
Upload Files...
Clone from GitHub
Configure AWS Cloud9
Main Theme:
jett-dark
bash - "ip-172-31-35-124" x
Immediate
bash - "ip-172-31-35-124" x
formation.git"
To set your display name run "git config --global user.name YOUR_USER_NAME"
To set your display email run "git config --global user.email YOUR_EMAIL_ADDRESS"
pluralsight-36a11776:~/environment $


File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
Welcome
Developer Tools
AWS Cloud9
Welcome to your development environment
AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.
Toolkit for AWS Cloud9
The AWS Toolkit for Cloud9 is an IDE extension that simplifies accessing and interacting with resources from services such as AWS Lambda, AWS CloudFormation, and AWS API Gateway. With the toolkit, developers can also develop, debug, and deploy applications using the AWS Serverless Application Model (SAM). Learn more
Getting started
Create File
Upload Files...
Clone from GitHub
Configure AWS Cloud9
Main Theme:
jett-dark
bash - "ip-172-31-35-124" x
Immediate
bash - "ip-172-31-35-124" x
formation.git"
To set your display name run "git config --global user.name YOUR_USER_NAME"
To set your display email run "git config --global user.email YOUR_EMAIL_ADDRESS"
pluralsight-36a11776:~/environment $
main
AWS: profile:default


us-west-2.console.aws.amazon.com/cloud9/ide/902b5c4eea0a4a08a4f999019ebd0474
File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
Welcome
PS-Labs-ACKLQ
c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
bash - "ip-172-31-35-124" Immediate python3.6 - "ip-172-31-35
error: could not lock config file /home/ec2-user/.gitconfig: File exists
pluralsight-36a11776:~/environment $ /tmp/git-cloning-runner-1668485893883-01792110709S.sh
Cloning into '/home/ec2-user/environment/templates'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 49 (delta 23), reused 44 (delta 18), pack-reused 0
Receiving objects: 100% (49/49), 24.92 KiB | 3.56 MiB/s, done.
Resolving deltas: 100% (23/23), done.
Navigate to your cloned repository by typing "cd /home/ec2-user/environment/templates" to start working with "https://github.com/ps-interactive/lab_aws_create-and-deploy-public-and-private-subnets-using-aws-cloud
formation.git"
To set your display name run "git config --global user.name YOUR_USER_NAME"
To set your display email run "git config --global user.email YOUR_EMAIL_ADDRESS"
pluralsight-36a11776:~/environment $ cd templates
pluralsight-36a11776:~/environment/templates (main) $ aws ec2 describe-vpcs
{
"Vpcs": [
{
"CidrBlock": "172.31.0.0/16",
"DhcpOptionsId": "dopt-d6dfa7ae",
"State": "available",
"VpcId": "vpc-06c343ba9584af58e",
"OwnerId": "984323420627",
"InstanceTenancy": "default",
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-0f465b6f796981c87",
"CidrBlock": "172.31.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"IsDefault": true
}
]
}
pluralsight-36a11776:~/environment/templates (main) $
main
AWS: profile:default


File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
Welcome
PS-Labs-ACKLQ
c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
error: could not lock config file /home/ec2-user/.gitconfig: File exists
pluralsight-36a11776:~/environment $ /tmp/git-cloning-runner-1668485893883-01792110709S.sh
Cloning into '/home/ec2-user/environment/templates'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 49 (delta 23), reused 44 (delta 18), pack-reused 0
Receiving objects: 100% (49/49), 24.92 KiB | 3.56 MiB/s, done.
Resolving deltas: 100% (23/23), done.
Navigate to your cloned repository by typing "cd /home/ec2-user/environment/templates" to start working with "https://github.com/ps-interactive/lab_aws_create-and-deploy-public-and-private-subnets-using-aws-cloud
formation.git"
To set your display name run "git config --global user.name YOUR_USER_NAME"
To set your display email run "git config --global user.email YOUR_EMAIL_ADDRESS"
cd templates
pluralsight-36a11776:~/environment $ cd templates
pluralsight-36a11776:~/environment/templates (main) $ aws ec2 describe-vpcs
{
"Vpcs": [
{
"CidrBlock": "172.31.0.0/16",
"DhcpOptionsId": "dopt-d6dfa7ae",
"State": "available",
"VpcId": "vpc-06c343ba9584af58e",
"OwnerId": "984323420627",
"InstanceTenancy": "default",
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-0f465b6f796981c87",
"CidrBlock": "172.31.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"IsDefault": true
}
]
}
pluralsight-36a11776:~/environment/templates (main) $
main
AWS: profile:default



Create a VPC

Oftentimes, deployment is taking place across various different customers' AWS accounts. To ensure we are not interfering with existing services, we want the new infrastructure to be part of a brand new VPC.

In this challenge, we'll be using CloudFormation to create a new VPC from scratch.

1. Inside the templates folder, open the file **02-template.yml**.
2. On line 7, set the value `AWS::EC2::VPC` to the property **Type**. This specifies a resource of type VPC.
3. On line 9, set the value `10.0.0.0/16` to the property **CidrBlock**. This specifies the range of IP addresses which will be available in our VPC.
4. Save the changes to this file.
5. Move to the terminal tab at the bottom of the page. Let's make sure we are in the templates folder. Inside the terminal tab, type the following command:

   ```
   cd ~/environment/templates/
   ```

6. Prior to deploying our CloudFormation stack, it's a good practice to validate the template for syntax errors. To do so, type the following command:

   ```
   aws cloudformation validate-template --template-body file://02-template.yml
   ```

7. This command should return a JSON output with a Parameters key set to an empty array, and a Description key set to the same Description property from the template. If this is the JSON output, then it means our template is valid!

   If this command does NOT return a JSON output, check the template file for syntax errors and missing values. Once the template is valid, proceed to the next step.

8. Still inside the terminal tab, run the following command to deploy the stack and create the VPC:

   ```
   aws cloudformation deploy --stack-name $INFRA_STACK --template-file 02-template.yml
   ```

9. Wait a few moments until a message like the following is displayed in the terminal

   Waiting for changeset to be created..

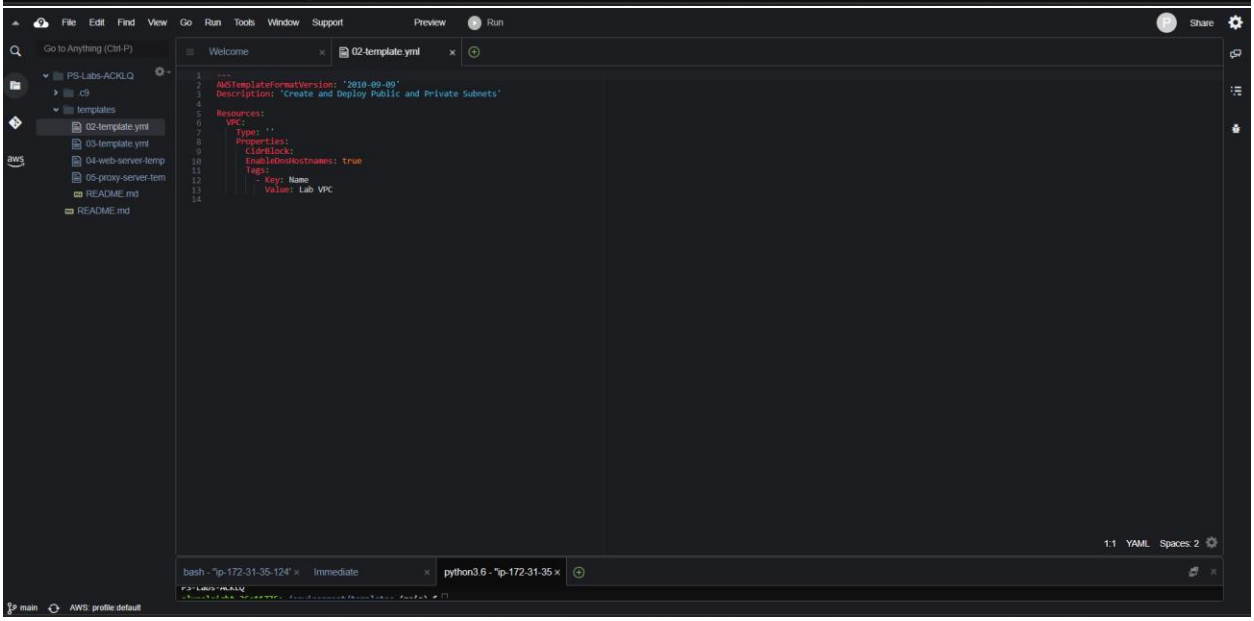
   Waiting for stack create/update to complete

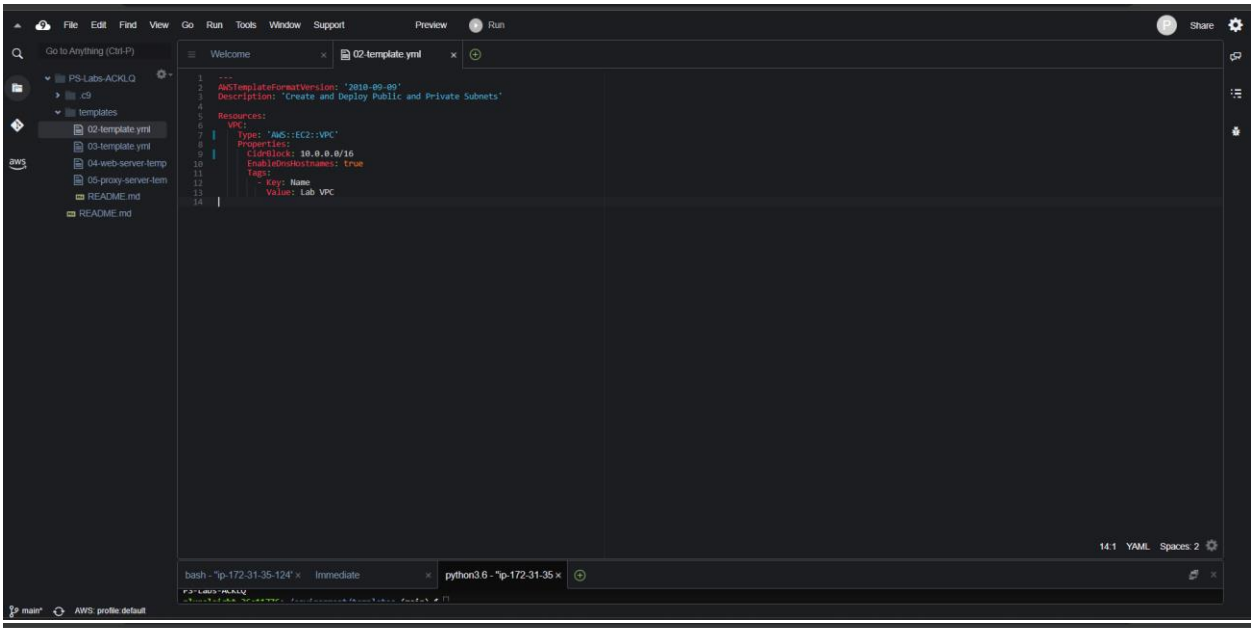
   Successfully created/updated stack...

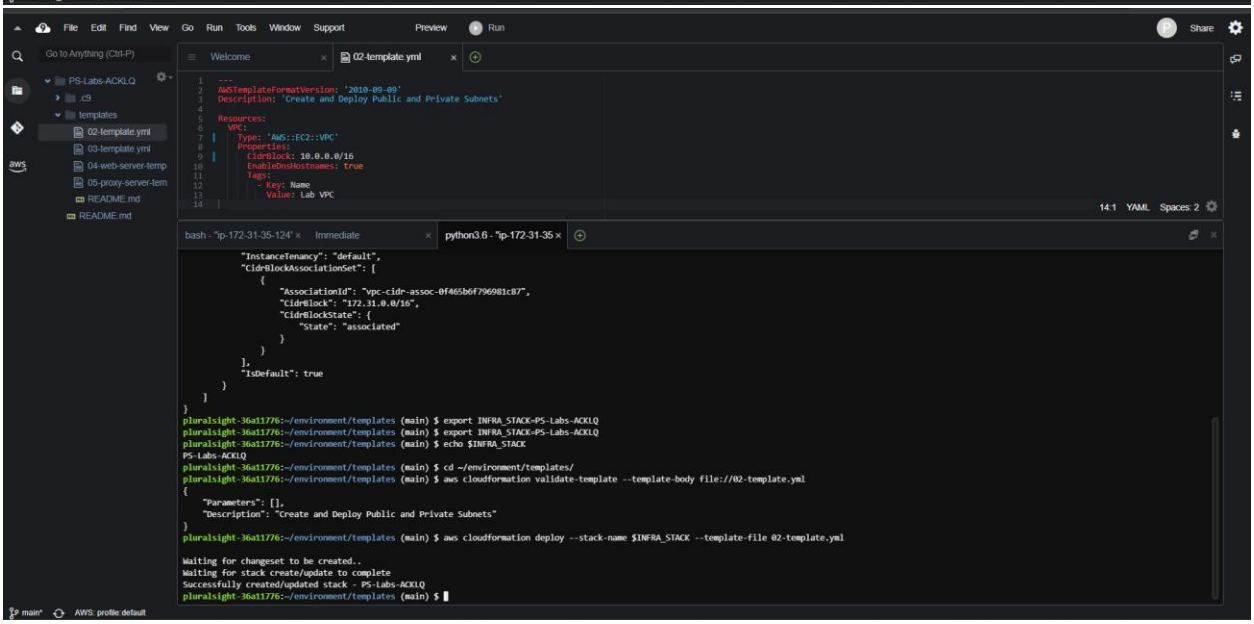
10. To see the newly created VPC, let's visit the VPC page on the AWS Console.

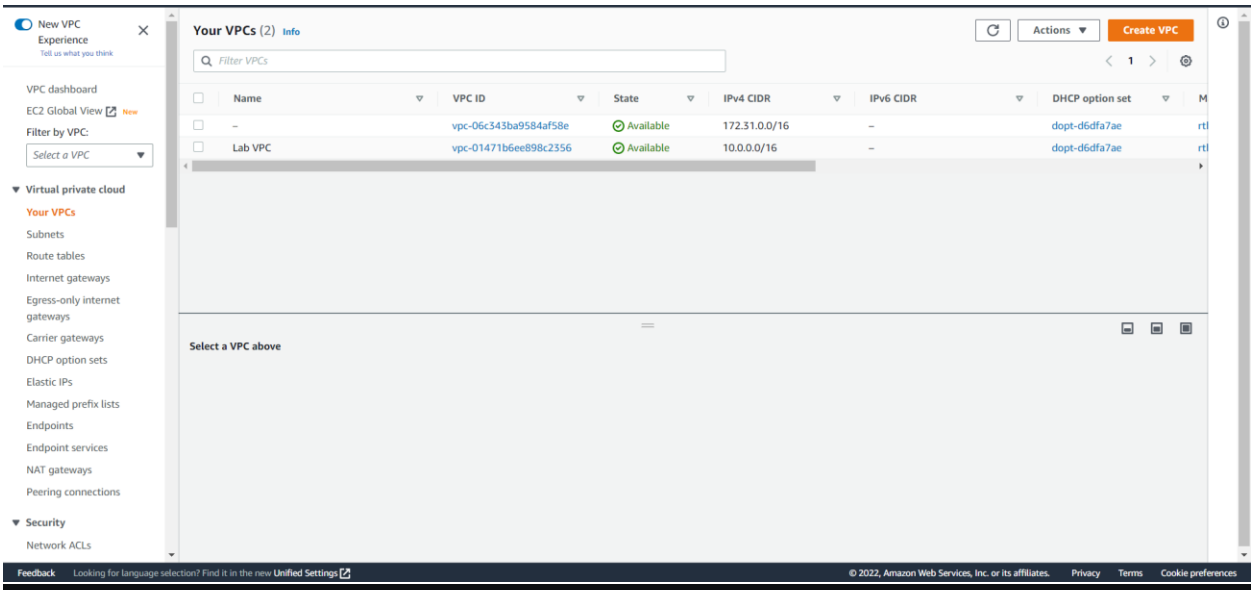
    Leave the Cloud9 tab open and, using a new browser tab, visit the AWS Console. Under the **Services** tab, go to the **VPC** service. Navigate to **Your VCPs**. Under the list of existing VPCs, our newly created VPC named *"Lab VPC"* should be listed with the IPv4 CIDR *10.0.0.0/16*.

If our new VPC is listed, congratulations! The VPC has been successfully created. Move on to the next challenge.




File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Go to Anything (Ctrl-P)
Welcome
02-template.yml
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
1 ---
2 AWSTemplateFormatVersion: '2010-09-09'
3 Description: 'Create and Deploy Public and Private Subnets'
4
5 Resources:
6 VPC:
7 Type: 'AWS::EC2::VPC'
8 Properties:
9 CidrBlock: 10.0.0.0/16
10 EnableDnsHostnames: true
11 Tags:
12 - Key: Name
13 Value: Lab VPC
14
14:1 YAML Spaces: 2
bash - "ip-172-31-35-124" x
Immediate
python3.6 - "ip-172-31-35 x
main*
AWS: profile default


File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Go to Anything (Ctrl-P)
Welcome
02-template.yml
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
1 ---
2 AWSTemplateFormatVersion: '2010-09-09'
3 Description: 'Create and Deploy Public and Private Subnets'
4
5 Resources:
6 VPC:
7 Type: 'AWS::EC2::VPC'
8 Properties:
9 CidrBlock: 10.0.0.0/16
10 EnableDnsHostnames: true
11 Tags:
12 - Key: Name
13 Value: Lab VPC
14
14:1 YAML Spaces: 2
bash - "ip-172-31-35-124" x
Immediate
python3.6 - "ip-172-31-35 x
"InstanceTenancy": "default",
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-0f465b6f796981c87",
"CidrBlock": "172.31.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"IsDefault": true
}
]
}
pluralsight-36a11776:~/environment/templates (main) $ export INFRA_STACK=PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ export INFRA_STACK=PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ echo $INFRA_STACK
PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ cd ~/environment/templates/
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation validate-template --template-body file://02-template.yml
{
"Parameters": [],
"Description": "Create and Deploy Public and Private Subnets"
}
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 02-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $
main*
AWS: profile default



Create a Public and Private Subnet

With the VPC in place, we can now proceed with the creation of the subnets.

In this challenge, we will create one public and one private subnet, the routing necessary for both of these subnets, and the Security Groups which will later be used by EC2 instances.

1. Go back to Cloud9. Still inside the templates folder, open the file **03-template.yml**.
2. On lines 16 and 27, set the value `AWS::EC2::Subnet` to the property **Type.**
3. On line 20, set the value `10.0.0.0/24` to the property **CidrBlock**. This sets the CIDR block for the Public Subnet.
4. On line 31, set the value `10.0.1.0/24` to the property **CidrBlock**. This sets the CIDR block for the Private Subnet.
5. On line 61, set the value `!Ref PublicSubnet` to the property **SubnetId**. This creates a reference to the Public Subnet on the Public Subnet Route Table.
6. On line 96, set the value `!Ref PrivateSubnet` to the property **SubnetId.** This creates a reference to the Private Subnet on the Private Subnet Route Table.
7. On line 120, set the value `!Ref ProxyServerSecurityGroup` to the property **SourceSecurityGroupId.** This ensures that only connections originated from the Proxy Server will be allowed to reach the Web Server.
8. Save the changes to this file.

9. Move back to the terminal tab and make sure we are still inside the templates folder. Type the following command to validate the template:

   `aws cloudformation validate-template --template-body file://03-template.yml`

   If this command does NOT return a JSON output, check the template file for syntax errors and missing values. Once the template is valid, proceed to the next step.

10. Run the following command to deploy the stack and create the public subnet:

    `aws cloudformation deploy --stack-name $INFRA_STACK --template-file 03-template.yml`

11. Wait a few moments until a message like the following is displayed in the terminal

    Waiting for changeset to be created..

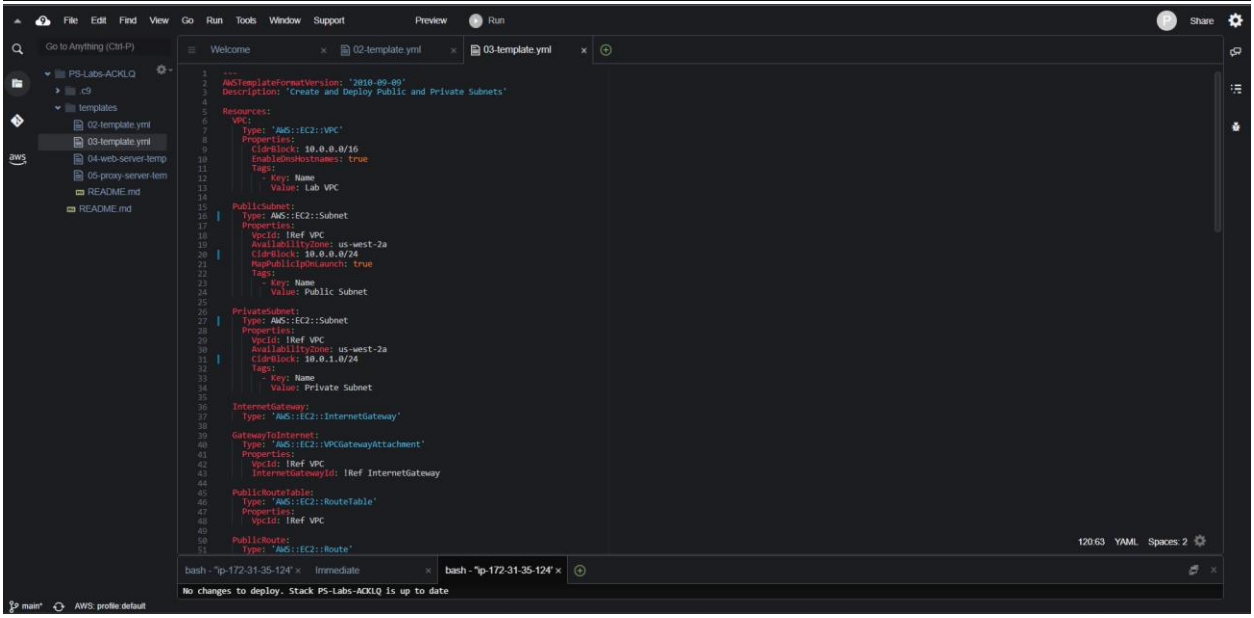
    Waiting for stack create/update to complete

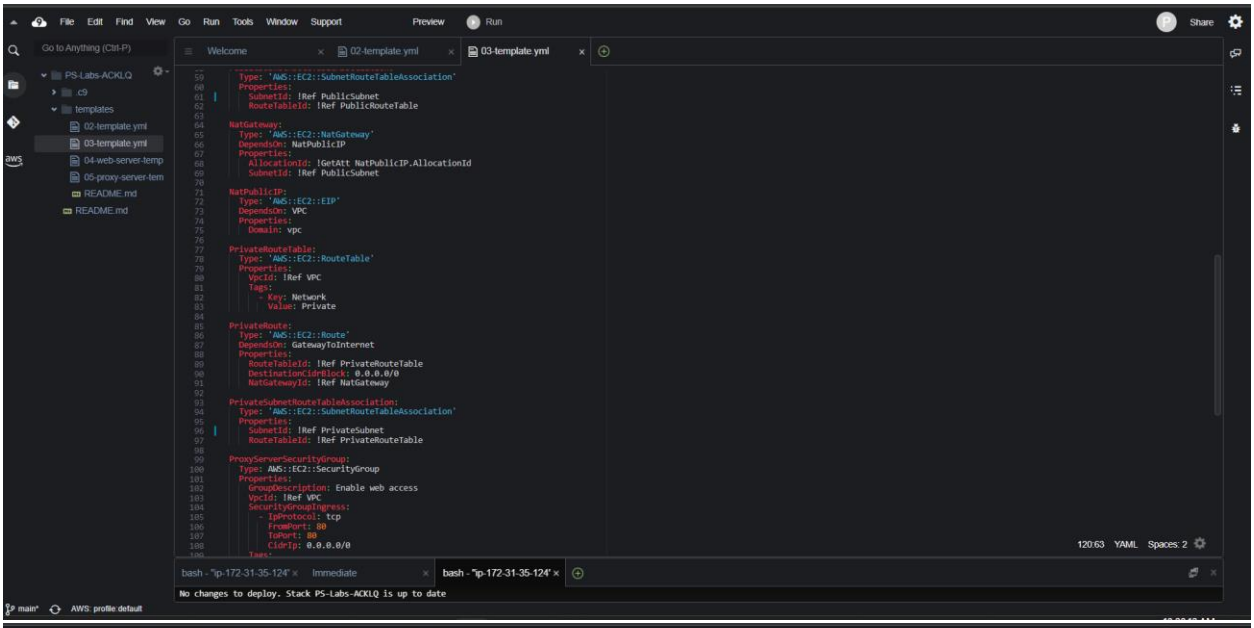
    Successfully created/updated stack...

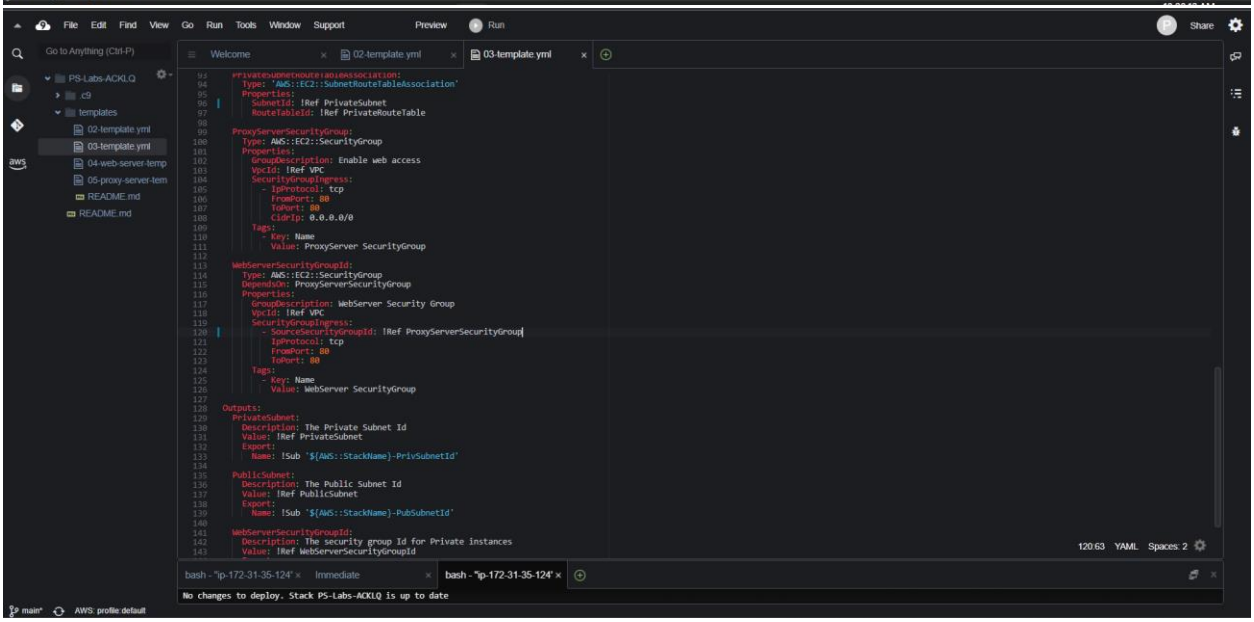
12. To see the newly created subnets, let's once again visit the VPC page on the AWS Console.

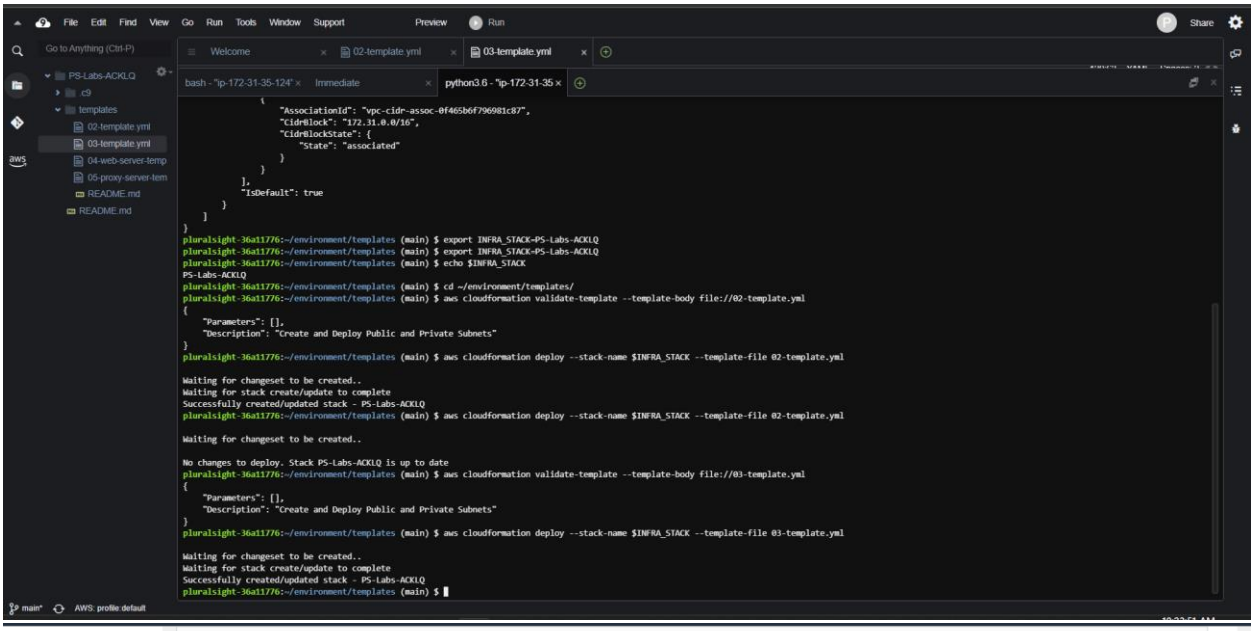
    On the page for VPCs, click the Subnets link on the left side-bar. There should be a list of subnets, including our newly created *"Public Subnet"* and *"Private Subnet"* with IPv4 CIDR *10.0.0.0/24* and *10.0.1.0/24* respectively.

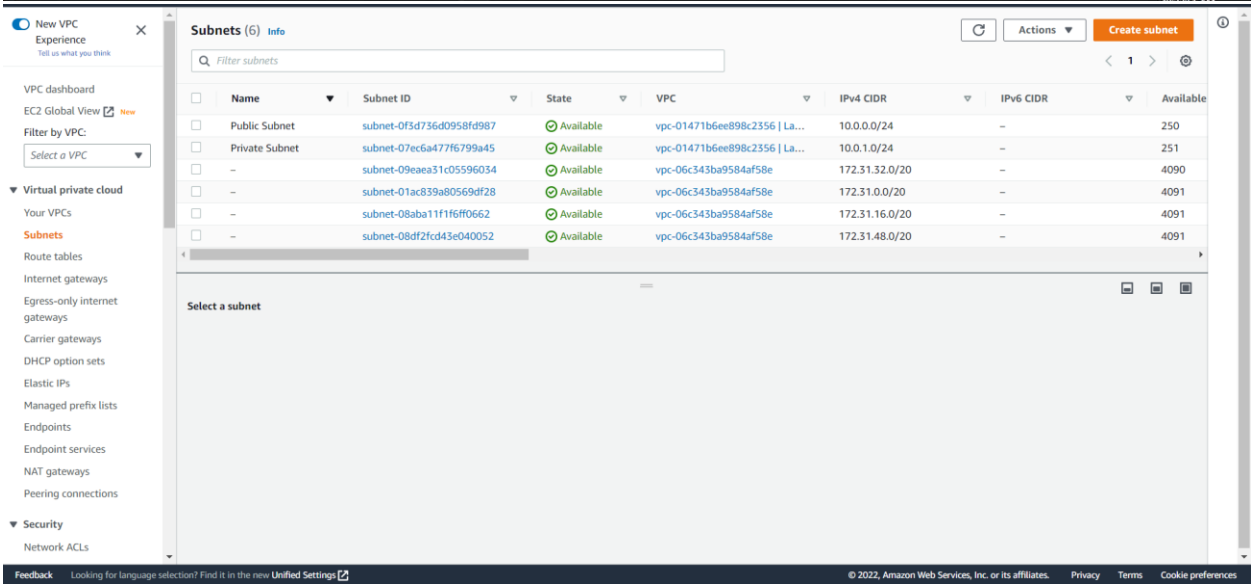
If the subnets are listed, congratulations! Move on to the next challenge.




File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
Welcome
02-template.yml
03-template.yml
Type: 'AWS::EC2::SubnetRouteTableAssociation'
Properties:
SubnetId: !Ref PublicSubnet
RouteTableId: !Ref PublicRouteTable
NatGateway:
Type: 'AWS::EC2::NatGateway'
DependsOn: NatPublicIP
Properties:
AllocationId: !GetAtt NatPublicIP.AllocationId
SubnetId: !Ref PublicSubnet
NatPublicIP:
Type: 'AWS::EC2::EIP'
DependsOn: VPC
Properties:
Domain: vpc
PrivateRouteTable:
Type: 'AWS::EC2::RouteTable'
Properties:
VpcId: !Ref VPC
Tags:
- Key: Network
Value: Private
PrivateRoute:
Type: 'AWS::EC2::Route'
DependsOn: GatewayToInternet
Properties:
RouteTableId: !Ref PrivateRouteTable
DestinationCidrBlock: 0.0.0.0/0
NatGatewayId: !Ref NatGateway
PrivateSubnetRouteTableAssociation:
Type: 'AWS::EC2::SubnetRouteTableAssociation'
Properties:
SubnetId: !Ref PrivateSubnet
RouteTableId: !Ref PrivateRouteTable
ProxyServerSecurityGroup:
Type: AWS::EC2::SecurityGroup
Properties:
GroupDescription: Enable web access
VpcId: !Ref VPC
SecurityGroupIngress:
- IpProtocol: tcp
FromPort: 80
ToPort: 80
CidrIp: 0.0.0.0/0
120:63 YAML Spaces: 2
bash - "ip-172-31-35-124" Immediate bash - "ip-172-31-35-124"
No changes to deploy. Stack PS-Labs-ACKLQ is up to date
main* AWS: profile default


File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
Welcome
02-template.yml
03-template.yml
PrivateSubnetRouteTableAssociation:
Type: 'AWS::EC2::SubnetRouteTableAssociation'
Properties:
SubnetId: !Ref PrivateSubnet
RouteTableId: !Ref PrivateRouteTable
ProxyServerSecurityGroup:
Type: AWS::EC2::SecurityGroup
Properties:
GroupDescription: Enable web access
VpcId: !Ref VPC
SecurityGroupIngress:
- IpProtocol: tcp
FromPort: 80
ToPort: 80
CidrIp: 0.0.0.0/0
Tags:
- Key: Name
Value: ProxyServer SecurityGroup
WebServerSecurityGroupId:
Type: AWS::EC2::SecurityGroup
DependsOn: ProxyServerSecurityGroup
Properties:
GroupDescription: WebServer Security Group
VpcId: !Ref VPC
SecurityGroupIngress:
- SourceSecurityGroupId: !Ref ProxyServerSecurityGroup
IpProtocol: tcp
FromPort: 80
ToPort: 80
Tags:
- Key: Name
Value: WebServer SecurityGroup
Outputs:
PrivateSubnet:
Description: The Private Subnet Id
Value: !Ref PrivateSubnet
Export:
Name: !Sub "${AWS::StackName}-PrivSubnetId"
PublicSubnet:
Description: The Public Subnet Id
Value: !Ref PublicSubnet
Export:
Name: !Sub "${AWS::StackName}-PubSubnetId"
WebServerSecurityGroupId:
Description: The security group Id for Private instances
Value: !Ref WebServerSecurityGroupId
120:63 YAML Spaces: 2
bash - "ip-172-31-35-124" Immediate bash - "ip-172-31-35-124"
No changes to deploy. Stack PS-Labs-ACKLQ is up to date
main* AWS: profile default


File Edit Find View Go Run Tools Window Support Preview Run Share
Welcome 02-template.yml 03-template.yml
bash - "ip-172-31-35-124" Immediate python3.6 - "ip-172-31-35
PS-Labs-ACKLQ
c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
"AssociationId": "vpc-cidr-assoc-0f465b6f796981c87",
"CidrBlock": "172.31.0.0/16",
"CidrBlockState": {
"State": "associated"
"IsDefault": true
pluralsight-36a11776:~/environment/templates (main) $ export INFRA_STACK=PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ export INFRA_STACK=PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ echo $INFRA_STACK
PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ cd ~/environment/templates/
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation validate-template --template-body file://02-template.yml
"Parameters": [],
"Description": "Create and Deploy Public and Private Subnets"
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 02-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 02-template.yml
Waiting for changeset to be created..
No changes to deploy. Stack PS-Labs-ACKLQ is up to date
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation validate-template --template-body file://03-template.yml
"Parameters": [],
"Description": "Create and Deploy Public and Private Subnets"
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 03-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $
main*
AWS: profile:default


New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Subnets (6) Info
Filter subnets
Actions
Create subnet
Name
Subnet ID
State
VPC
IPv4 CIDR
IPv6 CIDR
Available
Public Subnet subnet-0f3d736d0958fd987 Available vpc-01471b6ee898c2356 | La... 10.0.0.0/24 – 250
Private Subnet subnet-07ec6a477f6799a45 Available vpc-01471b6ee898c2356 | La... 10.0.1.0/24 – 251
– subnet-09eaea31c05596034 Available vpc-06c343ba9584af58e 172.31.32.0/20 – 4090
– subnet-01ac839a80569df28 Available vpc-06c343ba9584af58e 172.31.0.0/20 – 4091
– subnet-08aba11f1f6ff0662 Available vpc-06c343ba9584af58e 172.31.16.0/20 – 4091
– subnet-08df2fcd43e040052 Available vpc-06c343ba9584af58e 172.31.48.0/20 – 4091
Select a subnet
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Deploy to the Private Subnet

With the proper subnets in place, now it's time to create our first EC2 instance. We'll deploy an Nginx web server to the private subnet.

Notice, however, that this web server will NOT be reachable until the next challenge.

1. Open the file **04-web-server-template.yml**.
2. On line 13, set the value `AWS::EC2::Instance` to the property **Type.**
3. Save the changes to this file.
4. Move back to the terminal tab and make sure we are still inside the templates folder. Type the following command to validate the template:

   ```
   aws cloudformation validate-template --template-body file://04-web-server-template.yml
   ```

   If this command does NOT return a JSON output, check the template file for syntax errors and missing values. Once the template is valid, proceed to the next step.
5. We'll now create a new CloudFormation stack which references outputs from the previous stack. To do this, we'll use the option –parameter-overrides to pass the name of our previous stack to this new stack.
6. Run the following command to deploy the web server:

   ```
   aws cloudformation deploy --stack-name WEB-$INFRA_STACK --parameter-overrides InfraStackName=$INFRA_STACK --template-file 04-web-server-template.yml
   ```

7. Wait a few moments until a message like the following is displayed in the terminal

   Waiting for changeset to be created..

   Waiting for stack create/update to complete

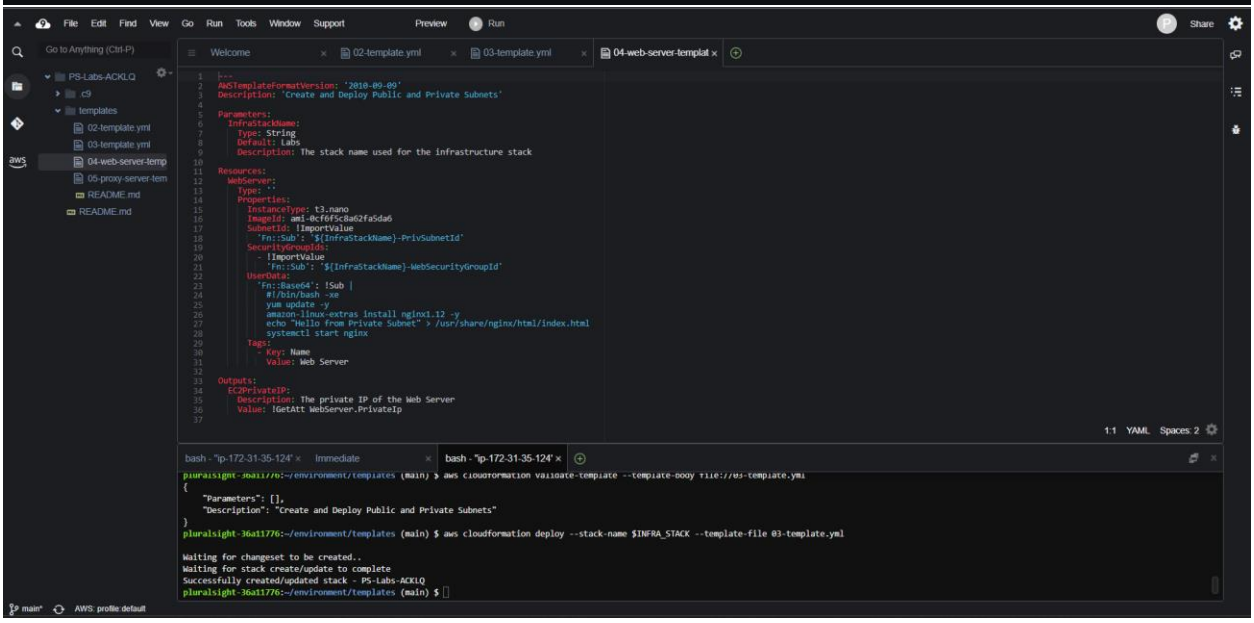
   Successfully created/updated stack...

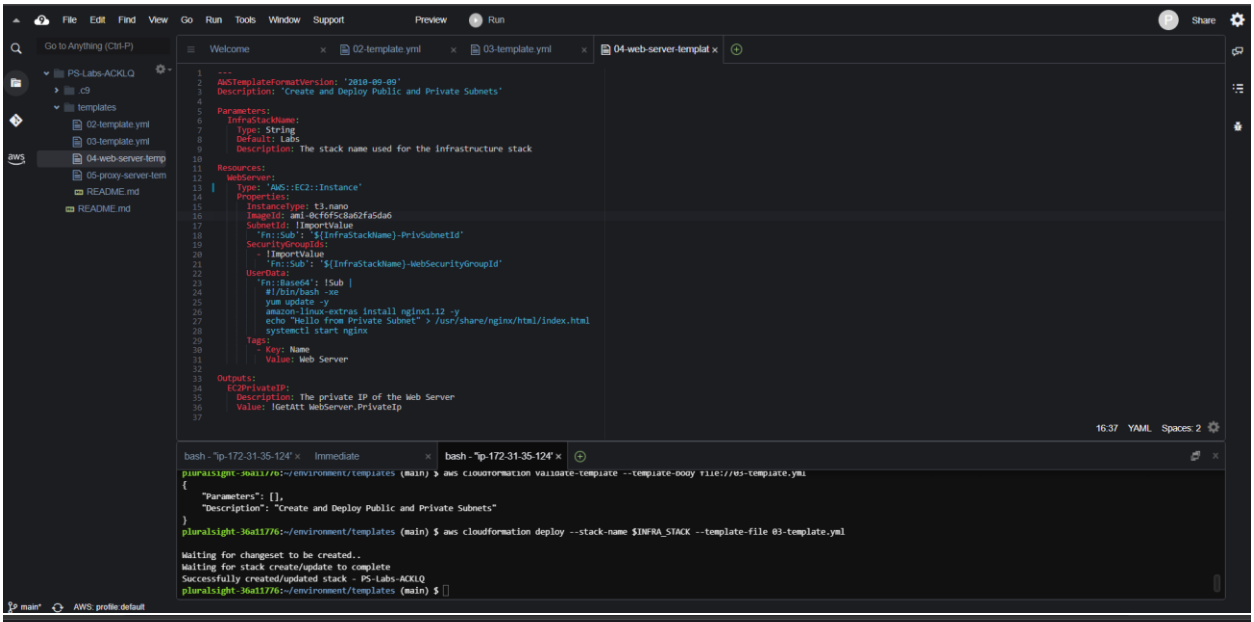
8. To see the newly created EC2 instance, let's one again visit the AWS Console.

   On the page for EC2, click on **Instances (running)**. Click the newly created Web Server instance to expand its details.

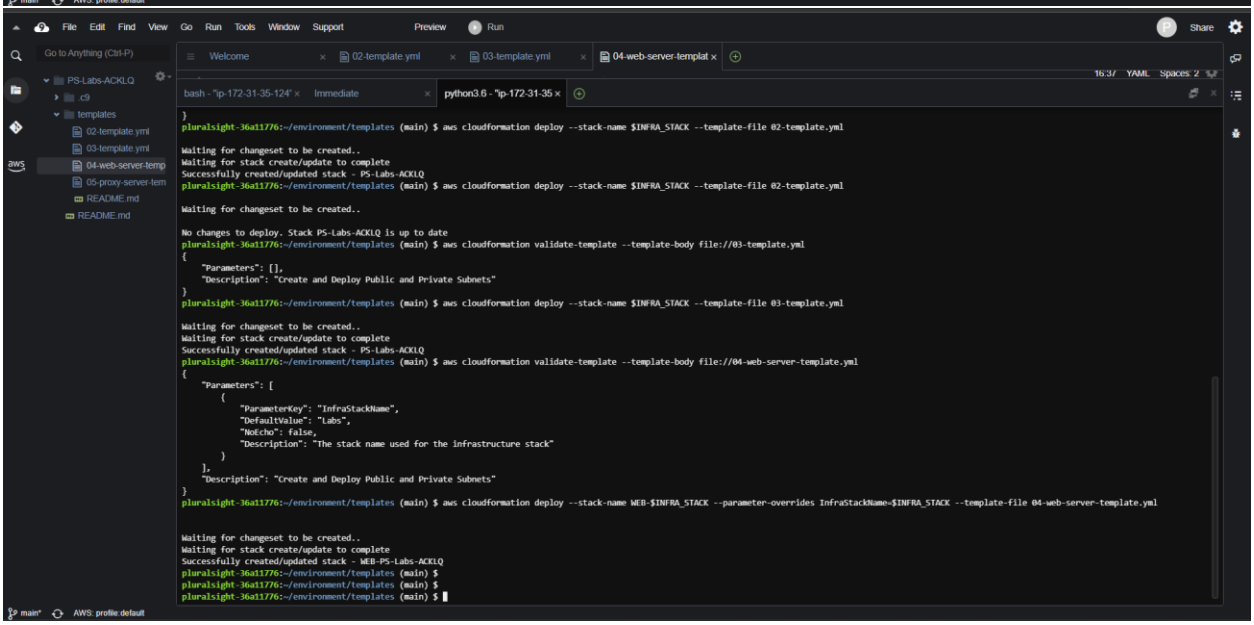
9. Under the **Details** tab for the Web Server instance, find the value for the property Private IPv4 addresses. Save this value. It will be needed in the next challenge to configure Proxy Server to find this particular EC2 instance.

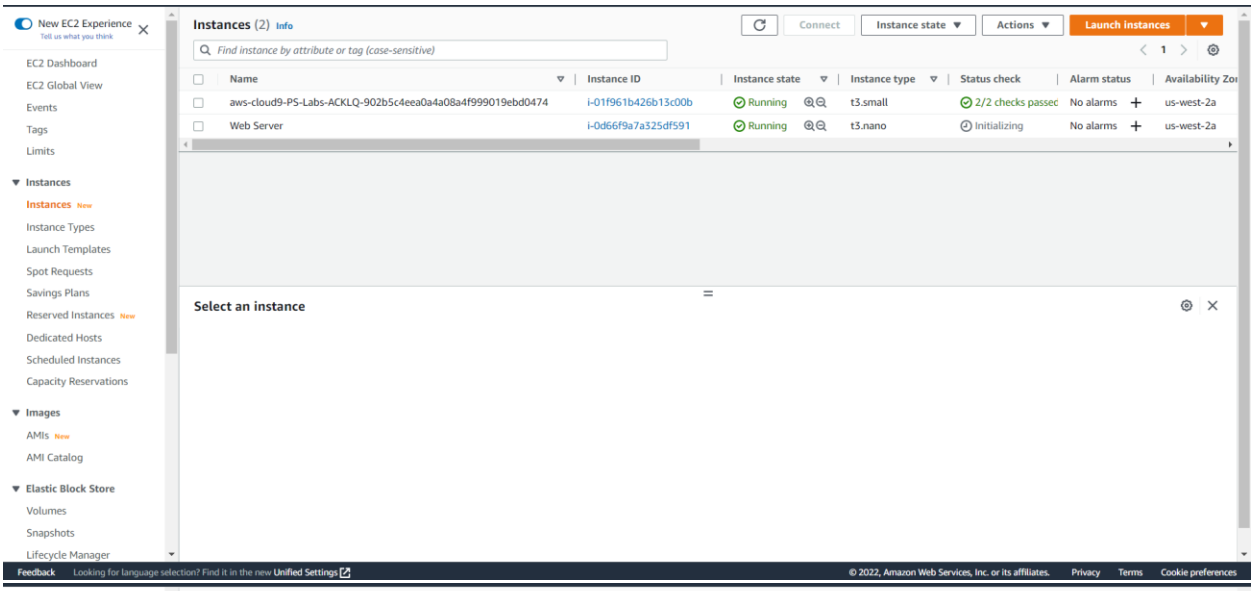
If the Web Server EC2 instance is listed and contains a valid Private IPv4 address, congratulations! Move on to the last challenge.

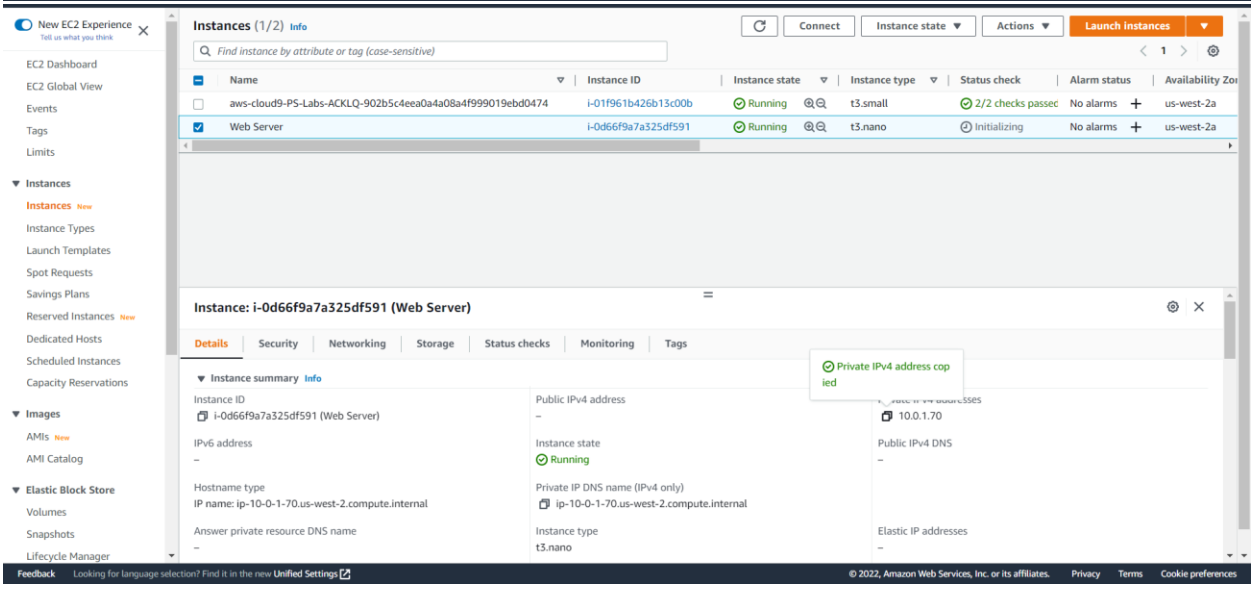



File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
Welcome
02-template.yml
03-template.yml
04-web-server-templat

AWSTemplateFormatVersion: '2010-09-09'
Description: 'Create and Deploy Public and Private Subnets'
Parameters:
InfraStackName:
Type: String
Default: Labs
Description: The stack name used for the infrastructure stack
Resources:
WebServer:
Type: 'AWS::EC2::Instance'
Properties:
InstanceType: t3.nano
ImageId: ami-0cf6f5c8a62fa5da6
SubnetId: !ImportValue
'Fn::Sub': '${InfraStackName}-PrivSubnetId'
SecurityGroupIds:
- !ImportValue
'Fn::Sub': '${InfraStackName}-WebSecurityGroupId'
UserData:
'Fn::Base64': !Sub |
#!/bin/bash -xe
yum update -y
amazon-linux-extras install nginx1.12 -y
echo "Hello from Private Subnet" > /usr/share/nginx/html/index.html
systemctl start nginx
Tags:
- Key: Name
Value: Web Server
Outputs:
EC2PrivateIP:
Description: The private IP of the Web Server
Value: !GetAtt WebServer.PrivateIp
16:37 YAML Spaces: 2
bash - "ip-172-31-35-124" Immediate bash - "ip-172-31-35-124"
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation validate-template --template-body file://03-template.yml
{
"Parameters": [],
"Description": "Create and Deploy Public and Private Subnets"
}
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 03-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $
main* AWS: profile:default


File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
Welcome
02-template.yml
03-template.yml
04-web-server-templat
16:37 YAML Spaces: 2
bash - "ip-172-31-35-124" Immediate python3.6 - "ip-172-31-35
}
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 02-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 02-template.yml
Waiting for changeset to be created..
No changes to deploy. Stack PS-Labs-ACKLQ is up to date
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation validate-template --template-body file://03-template.yml
{
"Parameters": [],
"Description": "Create and Deploy Public and Private Subnets"
}
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name $INFRA_STACK --template-file 03-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation validate-template --template-body file://04-web-server-template.yml
{
"Parameters": [
{
"ParameterKey": "InfraStackName",
"DefaultValue": "Labs",
"NoEcho": false,
"Description": "The stack name used for the infrastructure stack"
}
],
"Description": "Create and Deploy Public and Private Subnets"
}
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name WEB-$INFRA_STACK --parameter-overrides InfraStackName=$INFRA_STACK --template-file 04-web-server-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - WEB-PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $
pluralsight-36a11776:~/environment/templates (main) $
pluralsight-36a11776:~/environment/templates (main) $
main* AWS: profile:default


New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Instances (2) Info
Find instance by attribute or tag (case-sensitive)
Connect
Instance state
Actions
Launch instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zo
aws-cloud9-PS-Labs-ACKLQ-902b5c4eea0a4a08a4f999019ebd0474
i-01f961b426b13c00b
Running
t3.small
2/2 checks passed
No alarms
us-west-2a
Web Server
i-0d66f9a7a325df591
Running
t3.nano
Initializing
No alarms
us-west-2a
Select an instance
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Instances (1/2) Info
Find instance by attribute or tag (case-sensitive)
Connect
Instance state
Actions
Launch instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zo
aws-cloud9-PS-Labs-ACKLQ-902b5c4eea0a4a08a4f999019ebd0474
i-01f961b426b13c00b
Running
t3.small
2/2 checks passed
No alarms
us-west-2a
Web Server
i-0d66f9a7a325df591
Running
t3.nano
Initializing
No alarms
us-west-2a
Instance: i-0d66f9a7a325df591 (Web Server)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Private IPv4 address copied
Instance summary Info
Instance ID
i-0d66f9a7a325df591 (Web Server)
Public IPv4 address
–
10.0.1.70
IPv6 address
–
Instance state
Running
Public IPv4 DNS
–
Hostname type
IP name: ip-10-0-1-70.us-west-2.compute.internal
Private IP DNS name (IPv4 only)
ip-10-0-1-70.us-west-2.compute.internal
Answer private resource DNS name
–
Instance type
t3.nano
Elastic IP addresses
–
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Deploy to a Public Subnet

The web server residing in the private subnet is currently unreachable by the internet. In this challenge, we will deploy Nginx as a Reverse Proxy to the public subnet, so that it can route requests to the web server.

1. Open the file **05-proxy-server-template.yml**.
2. On line 52, set the `proxy_pass` property, part of nginx's configuration, to the Web Server's IPv4 address copied from the previous challenge.

   For example, if the App Server's private IP address was a.b.c.d, then line 214 would look like so:

   ```
   proxy_pass http://a.b.c.d;
   ```

   Attention to a few details:

   Do not forget to include the **http://** protocol.

   Remember to include a semicolon (`;`) at the end of the line.
3. Save the changes to this file.
4. Move back to the terminal tab and type the following command to validate the template:

   ```
   aws cloudformation validate-template --template-body file://05-proxy-server-template.yml
   ```

If this command does NOT return a JSON output, check the template file for syntax errors and missing values. Once the template is valid, proceed to the next step.

5. Run the following command to deploy the web server:

```
aws cloudformation deploy --stack-name PROXY-$INFRA_STACK --parameter-overrides InfraStackName=$INFRA_STACK --template-file 05-proxy-server-template.yml
```

6. Wait a few moments until a message like the following is displayed in the terminal

   Waiting for changeset to be created..

   Waiting for stack create/update to complete

   Successfully created/updated stack..

7. Run the following command to print the stack's information, including the URL for the Web Server:

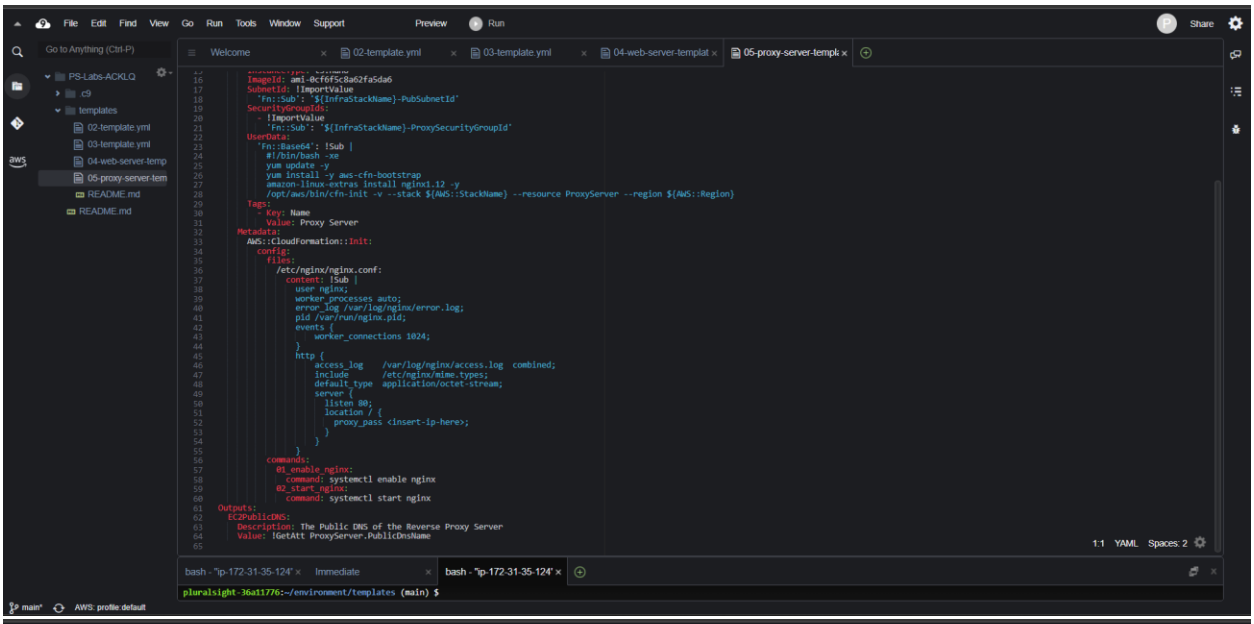
```
aws cloudformation describe-stacks --stack-name PROXY-$INFRA_STACK
```

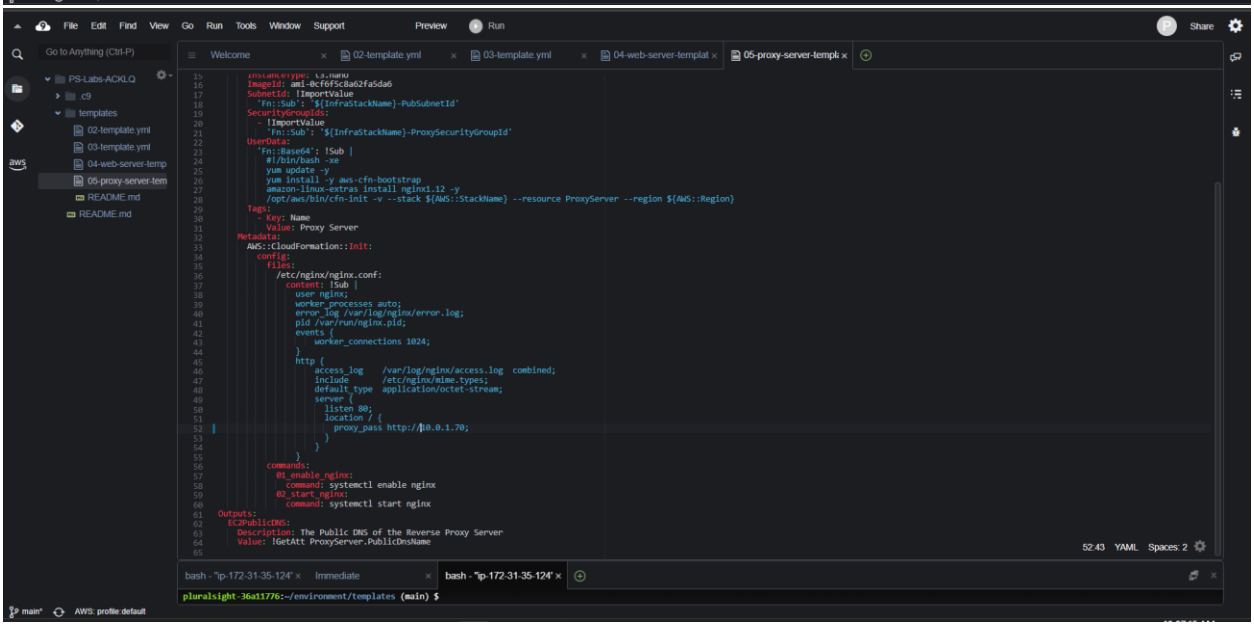
8. The result of running the previous command should be a JSON output. In this output, find the entry for **Outputs**, and then **OutputValue**. This value should be a URL that ends with *"...compute.amazonaws.com"*.

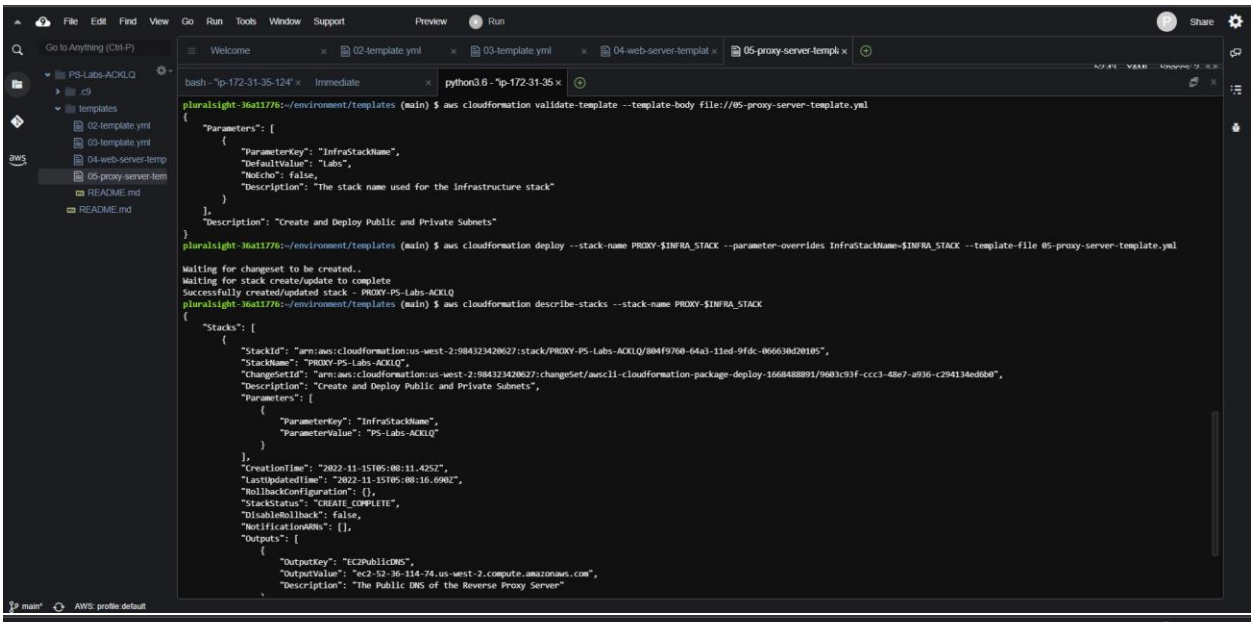
9. Visit this URL, and make sure the protocol used is HTTP and NOT HTTPS. Visiting the URL should display the message "Hello from Private Subnet". This content is actually coming from the Web Server running on the Private Subnet!

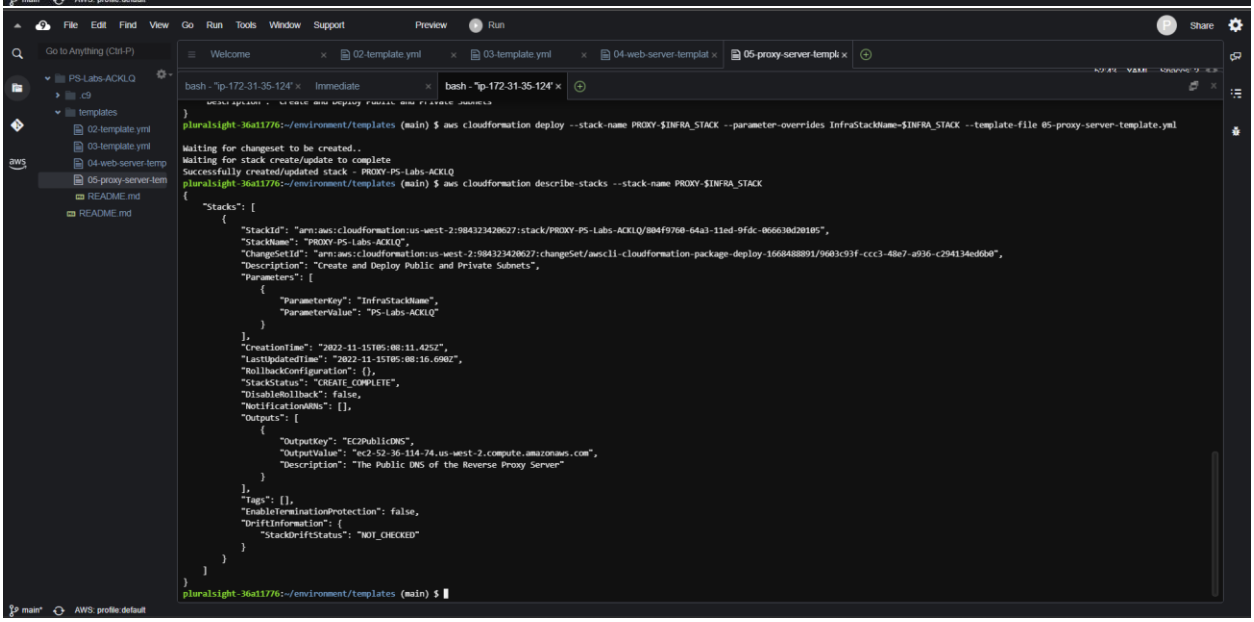
   Note: After the last CloudFormation command is finished running, it still takes a few more minutes for the proxy server to start. If the URL does not immediately display the message, wait a few minutes and refresh the page.

If you can see the message *"Hello from Private Subnet"*, then congratulations! You've successfully created and deployed a Public and Private Subnets using AWS CloudFormation and reached the end of this lab.


File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
Welcome
02-template.yml
03-template.yml
04-web-server-templat
05-proxy-server-templ
PS-Labs-ACKLQ
c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
ImageId: ami-0cf6f5c8a62fa5da6
SubnetId: !ImportValue
'Fn::Sub': '${InfraStackName}-PubSubnetId'
SecurityGroupIds:
- !ImportValue
'Fn::Sub': '${InfraStackName}-ProxySecurityGroupId'
UserData:
'Fn::Base64': !Sub |
#!/bin/bash -xe
yum update -y
yum install -y aws-cfn-bootstrap
amazon-linux-extras install nginx1.12 -y
/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource ProxyServer --region ${AWS::Region}
Tags:
- Key: Name
Value: Proxy Server
Metadata:
AWS::CloudFormation::Init:
config:
files:
/etc/nginx/nginx.conf:
content: !Sub |
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;
events {
worker_connections 1024;
}
http {
access_log /var/log/nginx/access.log combined;
include /etc/nginx/mime.types;
default_type application/octet-stream;
server {
listen 80;
location / {
proxy_pass <insert-ip-here>;
}
}
}
commands:
01_enable_nginx:
command: systemctl enable nginx
02_start_nginx:
command: systemctl start nginx
Outputs:
EC2PublicDNS:
Description: The Public DNS of the Reverse Proxy Server
Value: !GetAtt ProxyServer.PublicDnsName
1:1 YAML Spaces: 2
bash - "ip-172-31-35-124" Immediate
bash - "ip-172-31-35-124"
pluralsight-36a11776:~/environment/templates (main) $
main*
AWS: profile:default


File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
Welcome
02-template.yml
03-template.yml
04-web-server-templat
05-proxy-server-templ
PS-Labs-ACKLQ
c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
ImageId: ami-0cf6f5c8a62fa5da6
SubnetId: !ImportValue
'Fn::Sub': '${InfraStackName}-PubSubnetId'
SecurityGroupIds:
- !ImportValue
'Fn::Sub': '${InfraStackName}-ProxySecurityGroupId'
UserData:
'Fn::Base64': !Sub |
#!/bin/bash -xe
yum update -y
yum install -y aws-cfn-bootstrap
amazon-linux-extras install nginx1.12 -y
/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource ProxyServer --region ${AWS::Region}
Tags:
- Key: Name
Value: Proxy Server
Metadata:
AWS::CloudFormation::Init:
config:
files:
/etc/nginx/nginx.conf:
content: !Sub |
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;
events {
worker_connections 1024;
}
http {
access_log /var/log/nginx/access.log combined;
include /etc/nginx/mime.types;
default_type application/octet-stream;
server {
listen 80;
location / {
proxy_pass http://10.0.1.70;
}
}
}
commands:
01_enable_nginx:
command: systemctl enable nginx
02_start_nginx:
command: systemctl start nginx
Outputs:
EC2PublicDNS:
Description: The Public DNS of the Reverse Proxy Server
Value: !GetAtt ProxyServer.PublicDnsName
52:43 YAML Spaces: 2
bash - "ip-172-31-35-124" Immediate
bash - "ip-172-31-35-124"
pluralsight-36a11776:~/environment/templates (main) $
main*
AWS: profile:default


File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
Welcome
02-template.yml
03-template.yml
04-web-server-templat
05-proxy-server-templ
bash - "ip-172-31-35-124"
Immediate
python3.6 - "ip-172-31-35
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation validate-template --template-body file://05-proxy-server-template.yml
{
"Parameters": [
{
"ParameterKey": "InfraStackName",
"DefaultValue": "Labs",
"NoEcho": false,
"Description": "The stack name used for the infrastructure stack"
}
],
"Description": "Create and Deploy Public and Private Subnets"
}
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name PROXY-$INFRA_STACK --parameter-overrides InfraStackName=$INFRA_STACK --template-file 05-proxy-server-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PROXY-PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation describe-stacks --stack-name PROXY-$INFRA_STACK
{
"Stacks": [
{
"StackId": "arn:aws:cloudformation:us-west-2:984323420627:stack/PROXY-PS-Labs-ACKLQ/804f9760-64a3-11ed-9fdc-066630d20105",
"StackName": "PROXY-PS-Labs-ACKLQ",
"ChangeSetId": "arn:aws:cloudformation:us-west-2:984323420627:changeSet/awscli-cloudformation-package-deploy-1668488891/9603c93f-ccc3-48e7-a936-c294134ed6b0",
"Description": "Create and Deploy Public and Private Subnets",
"Parameters": [
{
"ParameterKey": "InfraStackName",
"ParameterValue": "PS-Labs-ACKLQ"
}
],
"CreationTime": "2022-11-15T05:08:11.425Z",
"LastUpdatedTime": "2022-11-15T05:08:16.690Z",
"RollbackConfiguration": {},
"StackStatus": "CREATE_COMPLETE",
"DisableRollback": false,
"NotificationARNs": [],
"Outputs": [
{
"OutputKey": "EC2PublicDNS",
"OutputValue": "ec2-52-36-114-74.us-west-2.compute.amazonaws.com",
"Description": "The Public DNS of the Reverse Proxy Server"
main*
AWS: profile:default


File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
PS-Labs-ACKLQ
.c9
templates
02-template.yml
03-template.yml
04-web-server-temp
05-proxy-server-tem
README.md
README.md
Welcome
02-template.yml
03-template.yml
04-web-server-templat
05-proxy-server-templ
bash - "ip-172-31-35-124"
Immediate
bash - "ip-172-31-35-124"
}
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation deploy --stack-name PROXY-$INFRA_STACK --parameter-overrides InfraStackName=$INFRA_STACK --template-file 05-proxy-server-template.yml
Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - PROXY-PS-Labs-ACKLQ
pluralsight-36a11776:~/environment/templates (main) $ aws cloudformation describe-stacks --stack-name PROXY-$INFRA_STACK
{
"Stacks": [
{
"StackId": "arn:aws:cloudformation:us-west-2:984323420627:stack/PROXY-PS-Labs-ACKLQ/804f9760-64a3-11ed-9fdc-066630d20105",
"StackName": "PROXY-PS-Labs-ACKLQ",
"ChangeSetId": "arn:aws:cloudformation:us-west-2:984323420627:changeSet/awscli-cloudformation-package-deploy-1668488891/9603c93f-ccc3-48e7-a936-c294134ed6b0",
"Description": "Create and Deploy Public and Private Subnets",
"Parameters": [
{
"ParameterKey": "InfraStackName",
"ParameterValue": "PS-Labs-ACKLQ"
}
],
"CreationTime": "2022-11-15T05:08:11.425Z",
"LastUpdatedTime": "2022-11-15T05:08:16.690Z",
"RollbackConfiguration": {},
"StackStatus": "CREATE_COMPLETE",
"DisableRollback": false,
"NotificationARNs": [],
"Outputs": [
{
"OutputKey": "EC2PublicDNS",
"OutputValue": "ec2-52-36-114-74.us-west-2.compute.amazonaws.com",
"Description": "The Public DNS of the Reverse Proxy Server"
}
],
"Tags": [],
"EnableTerminationProtection": false,
"DriftInformation": {
"StackDriftStatus": "NOT_CHECKED"
}
}
]
}
pluralsight-36a11776:~/environment/templates (main) $
main*
AWS: profile:default

Hello from Private Subnet