Slalom AWS Introduction Workshop

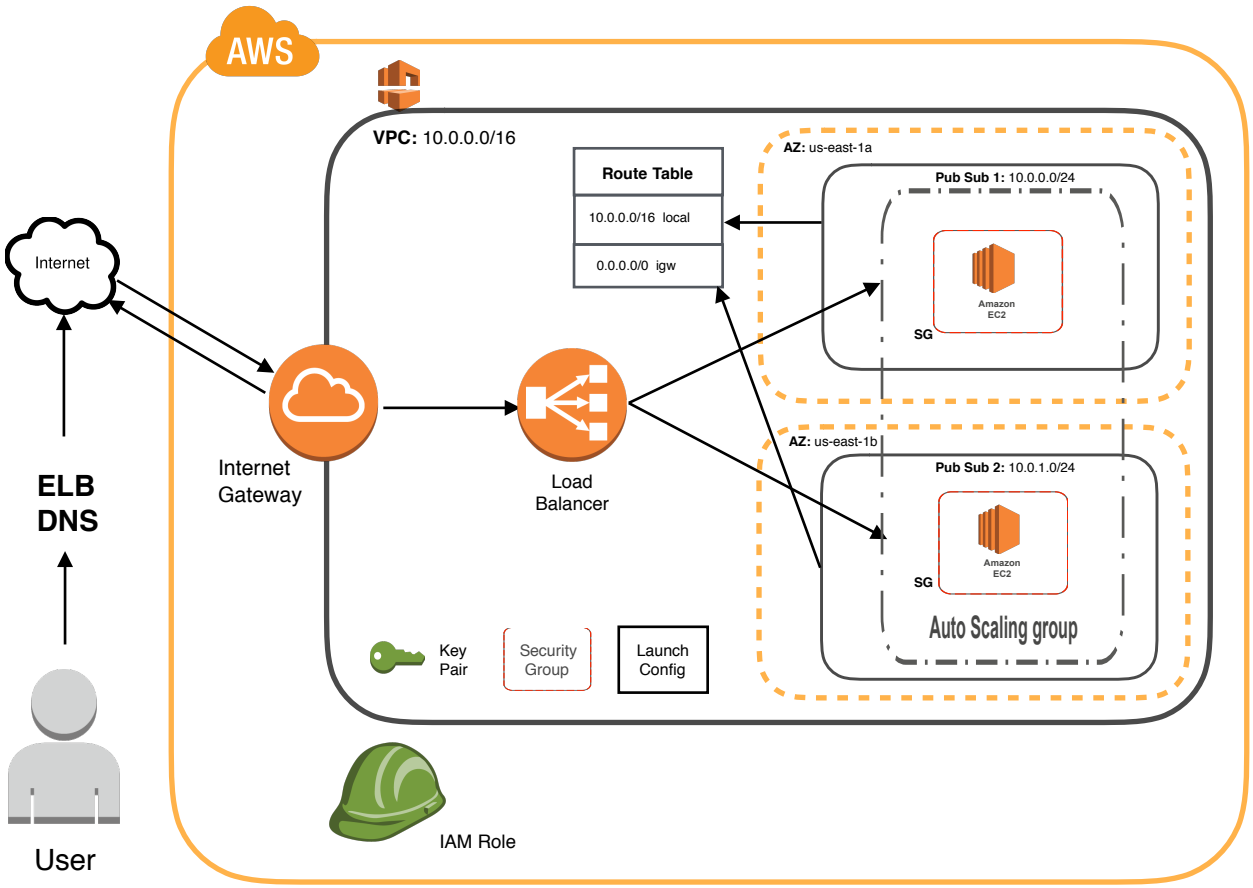
Introduction

Welcome to the lab section of the Slalom AWS Introduction Workshop! This document outlines 11 steps, walking you through the creation of the AWS resources required to setup a simple load-balanced autoscaling group serving a basic webpage.

At the end of each step (excluding Step 1) you will find an architectural summary of the AWS resources that you have built so far.

The next two pages contain a brief summary of what is achieved within each step of the process. In addition, please reach out to Bruce, Jeff or Todd for clarification on any of the content covered within this document!

The following image outlines the final architecture of what we are going to be building within this lab.



Step Overview

Step 1: Logging in to the AWS Console

By now you should have signed up for your own AWS account. This step simply directs you to log in to your account and select the appropriate AWS Region

Step 2: Create a VPC

Amazon Virtual Private Cloud (VPC) lets you provision a logically isolated section of the AWS Cloud. Think of it as your own private data center within the larger AWS cloud. You create your VPC using an IP address block, which describes how many IP addresses you wish your VPC to contain. In our example, we use 10.0.0.0/16, which provides 65531 IP addresses for use.

Step 3: Creation of Additional Public Subnet

Within a VPC, you can create subnets to logically divide the space within your VPC into smaller logical sections. Subnets can be both public (AWS resources created within are reachable over the public Internet) or private (AWS resources created within cannot be reached over the Internet). An important fact to know is that a subnet is tied to an Amazon availability zone such as us-east-1a **or** us-east-1b. **You cannot have a subnet that spans multiple availability zones**.

Step 4: Route Table Association

When you create a subnet, AWS also creates a route table for that subnet. The route table contains entries (routes) that define how to communicate with other AWS resources and resources within specific IP address blocks

Step 5: Create an IAM Role

AWS Identity and Access Management (IAM) is an AWS best practice that helps you to define what a resource can and cannot do in AWS. In our example, we will enable the virtual servers (EC2 instances) in our auto scale group to communicate with another popular AWS Service, Amazon S3. Communication with all other AWS Services will be denied. Unlike many of the other AWS Resources we create within this workshop, IAM is a Global service. This means any IAM resources that you create are visible and available for use in any AWS Region.

Step 6: Create a Key Pair

Generating a key pair allows you to log into an EC2 instance using ssh or RDP when the EC2 instance is created with an association to that key pair.

Step 7: Create a Security Group

Security groups provide an efficient way to assign access to AWS resources within your Virtual Private Cloud (VPC). Think of it as a firewall, where you can allow various network traffic types to pass through specific ports and IP address ranges. In our example, we open up port 80 to allow HTTP traffic from the Internet.

Step 8: Create a load balancer

An Elastic Load Balancer (ELB) is used to help distribute incoming traffic among any connected EC2 instances. In our example, we use it to distribute traffic among the EC2 instances in our auto scaling group. Our autoscaling group is spread across us-east-1a and us-east-1b, so this helps to introduce an element of high availability.

Step 9: Create a launch configuration

A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. Within our launch configuration we will include some of the features we have created earlier, such as the KeyPair and our Security Group.

Step 10: Create an AutoScaling Group

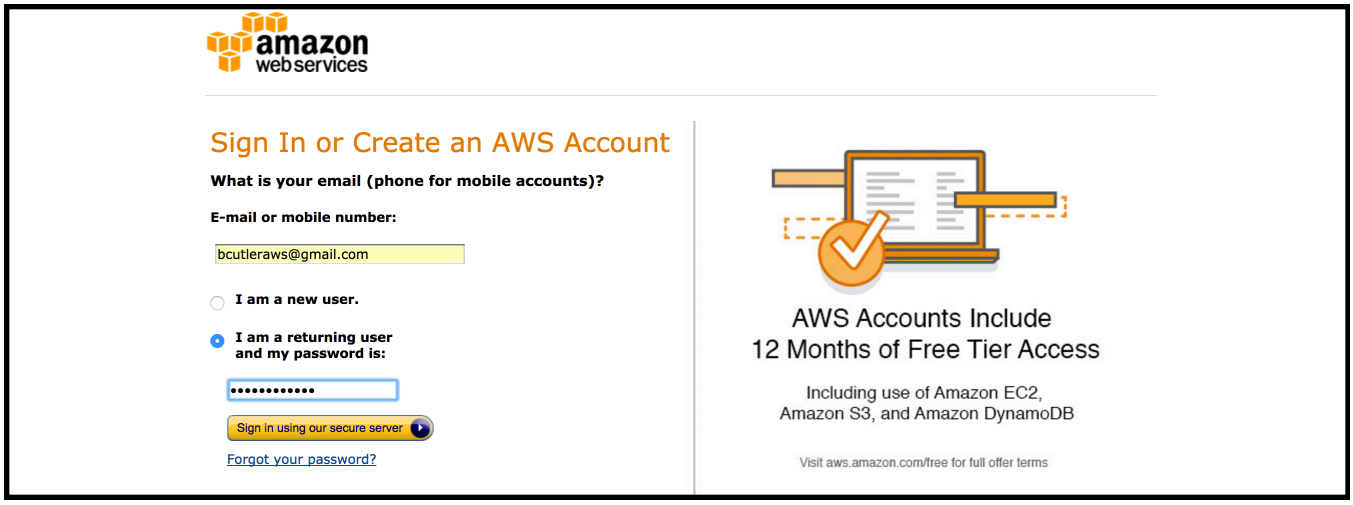
An Auto Scaling group is a collection of EC2 instances that share similar characteristics and are treated as a logical grouping. In our example, we create an Auto Scaling group with a size of two EC2 instances. Our autoscaling group will create instances in us-east-1a and us-east-1b availability zones. The load balancer ensures traffic that enters the load balancer is distributed between these two EC2 instances.

Step 11: Test it out!

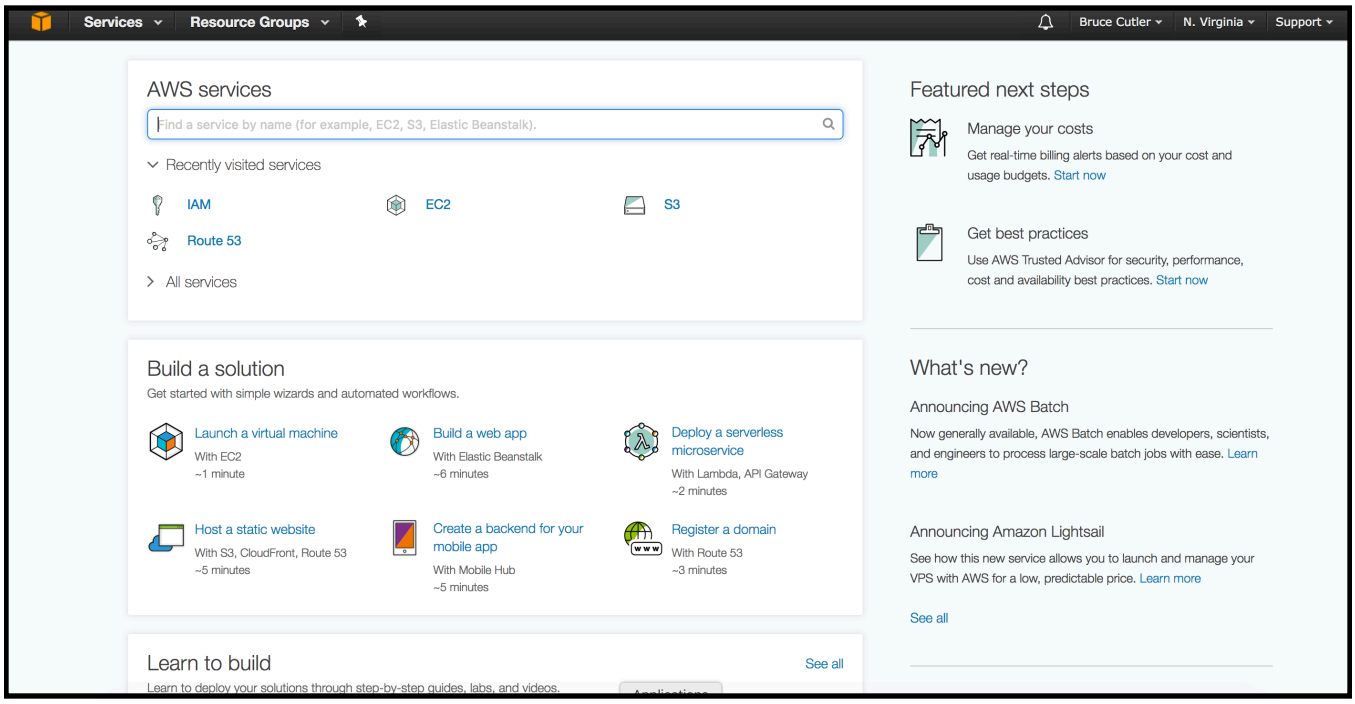
Once we've build all of the above components, we can use the DNS value from the load balancer in a browser to view our simple webpage.

Step 1: Logging in to the AWS Console

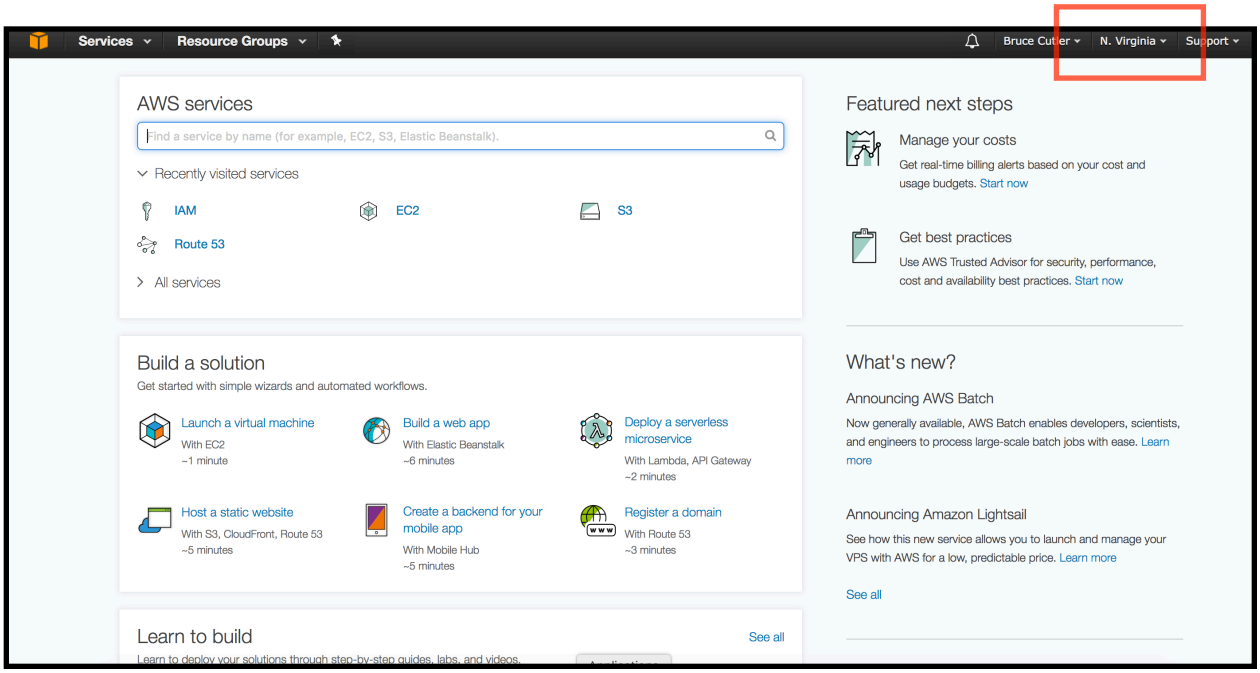
1. In a web browser, navigate to **https://console.aws.amazon.com/console/home**



2. Log in with the email / password combination you signed up for your account with. This will bring you to the AWS Console home page:

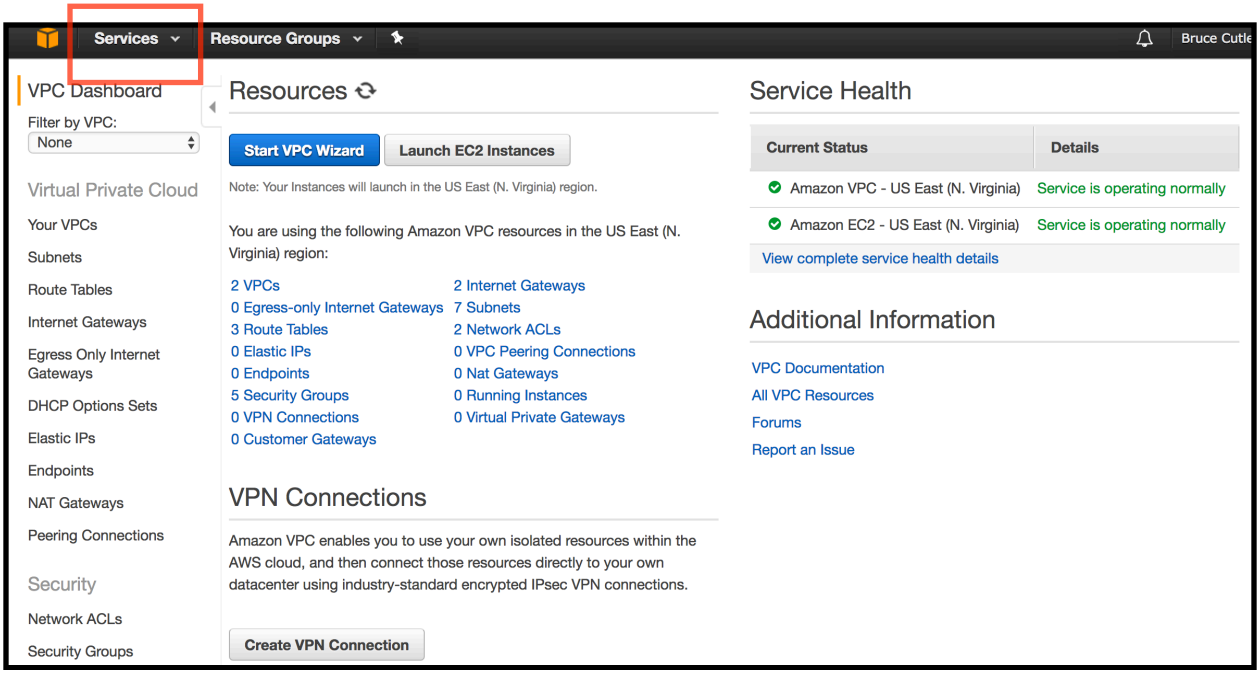


3. At the top right of the AWS Console screen, ensure that the **N. Virginia** is selected as your AWS Region



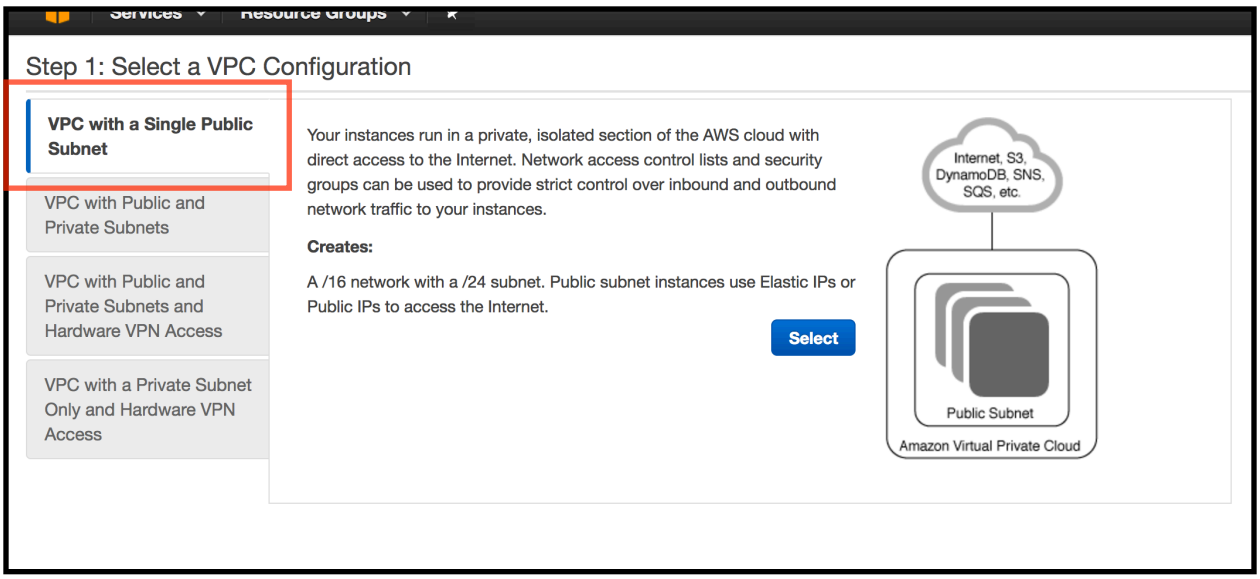
Step 2: Create a VPC (Virtual Private Cloud)

1. In the AWS Console, navigate to **Services —> VPC:**



2. Click on the blue **Start VPC Wizard** button towards the top left of the screen

3. In the first page of the VPC Wizard, select the top option, **VPC with a Single Public Subnet** and click the blue **Select** button:



4. Ensure that the following is entered when configuring your VPC and Public Subnet:

IPv4 CIDR Block: 10.0.0.0/16

VPC Name: slalom-aws-intro

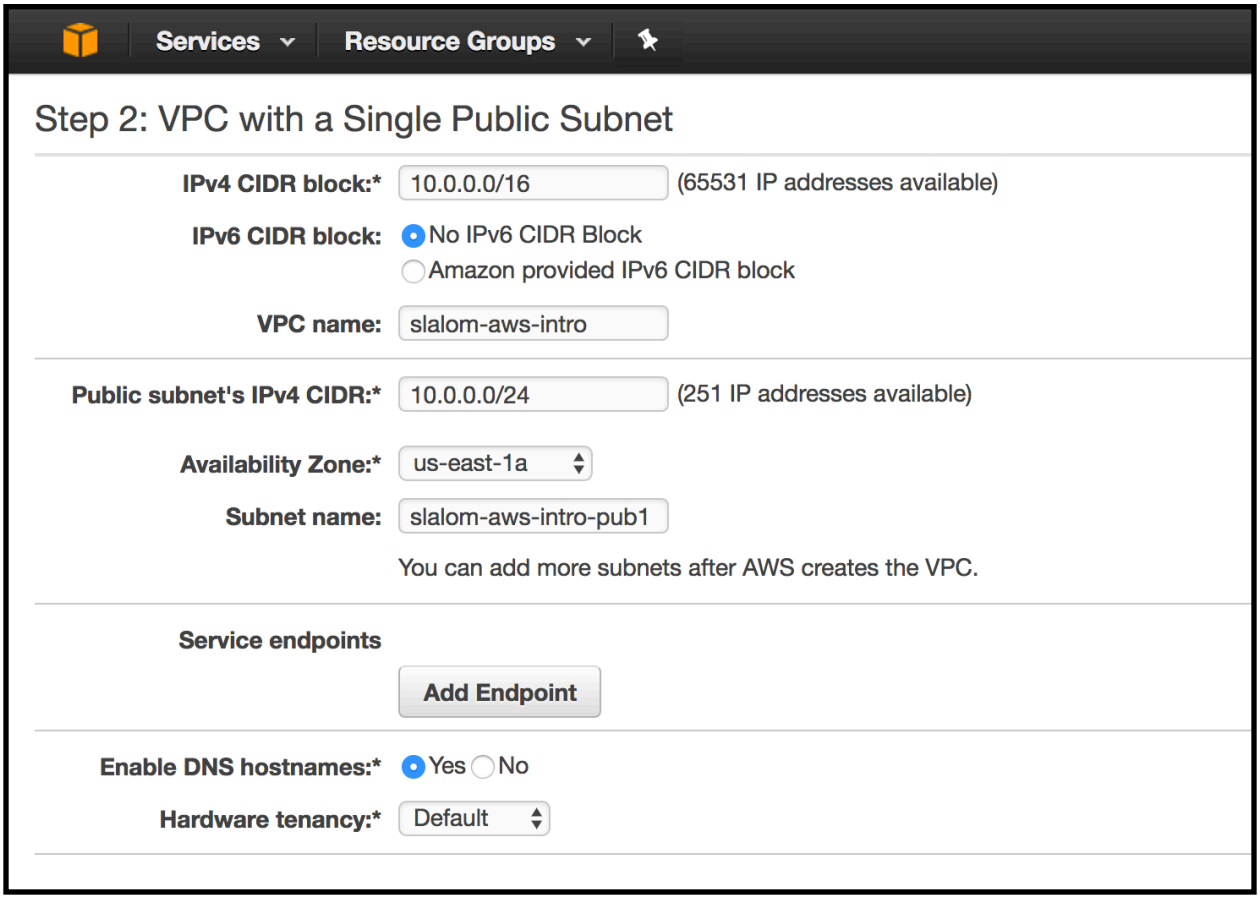
Public subnet's IPv4 CIDR: 10.0.0.0/24

Availability Zone: us-east-1a

Subnet Name: slalom-aws-intro-pub1

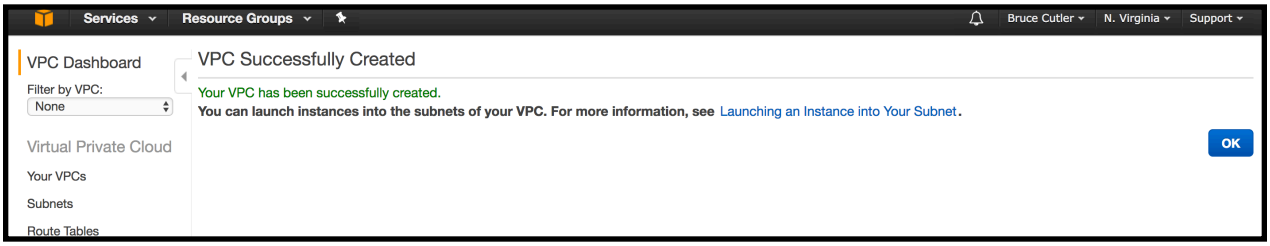
Enable DNS Hostnames: Yes

All other options can be left as default values, leaving you with the following setup:



5. Select the **Create VPC** button when you are ready

6. You should then see the following screen if VPC creation was successful

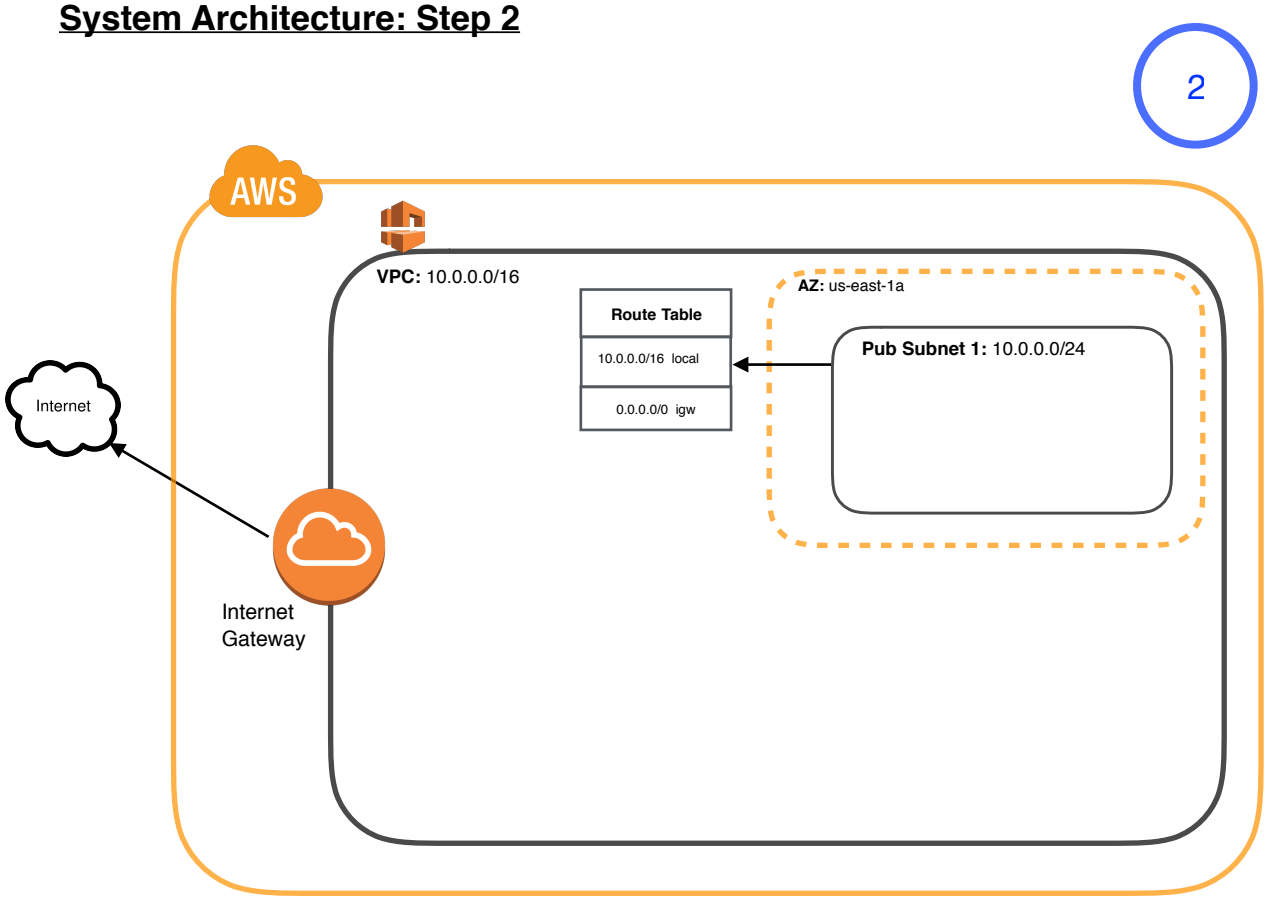


7. In addition to the VPC, the VPC Wizard has automatically created other components that are required for a successful networking setup.

   Click in the following menu items on the left side to see what else has been created with the **slalom-aws-intro** VPC:

   - Subnets
   - Route Tables
   - Internet Gateways

System Architecture: Step 2



Step 3: Creation of Additional Public Subnet

We created the **slalom-aws-intro-pub1** subnet as part of VPC creation in **Step 2**. To show the benefits of distributing resources across multiple AWS zones, we are going to create an additional subnet in a different availability zone

1. Navigate to **Services —> VPC** and click on the **Subnets** link in the left menu
2. Click on the blue **Create Subnet** button
3. Ensure that the following information is entered in the Create Subnet box that appears:

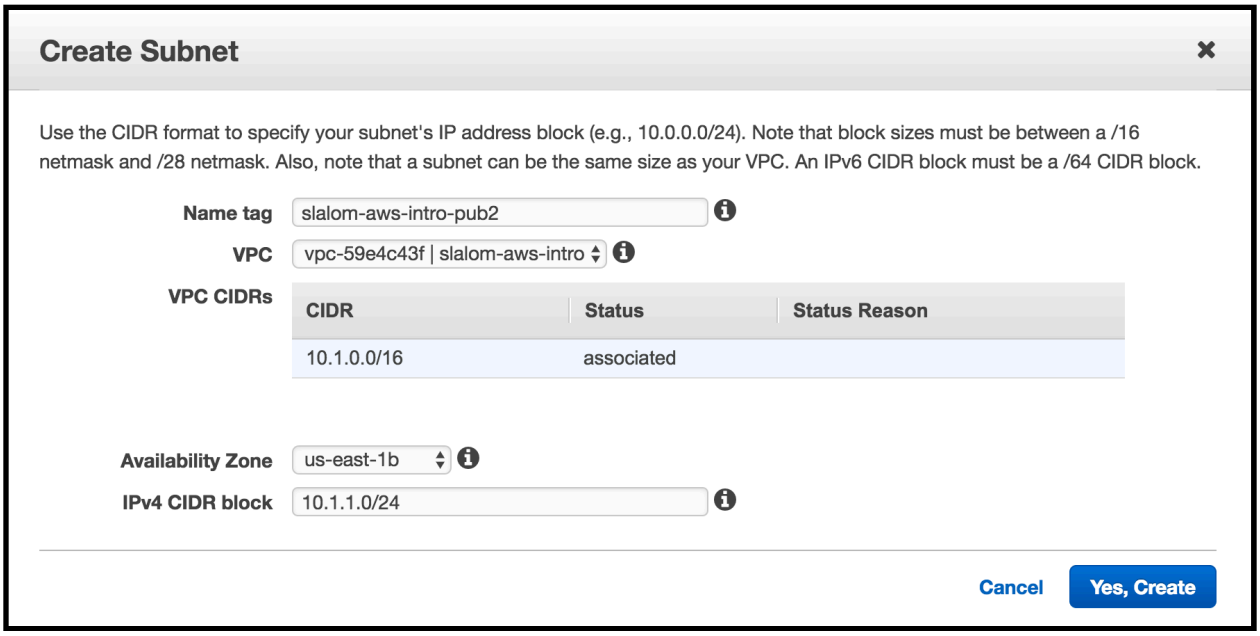
Name tag: slalom-aws-intro-pub2

VPC: slalom-aws-intro

Availability Zone: us-east-1b

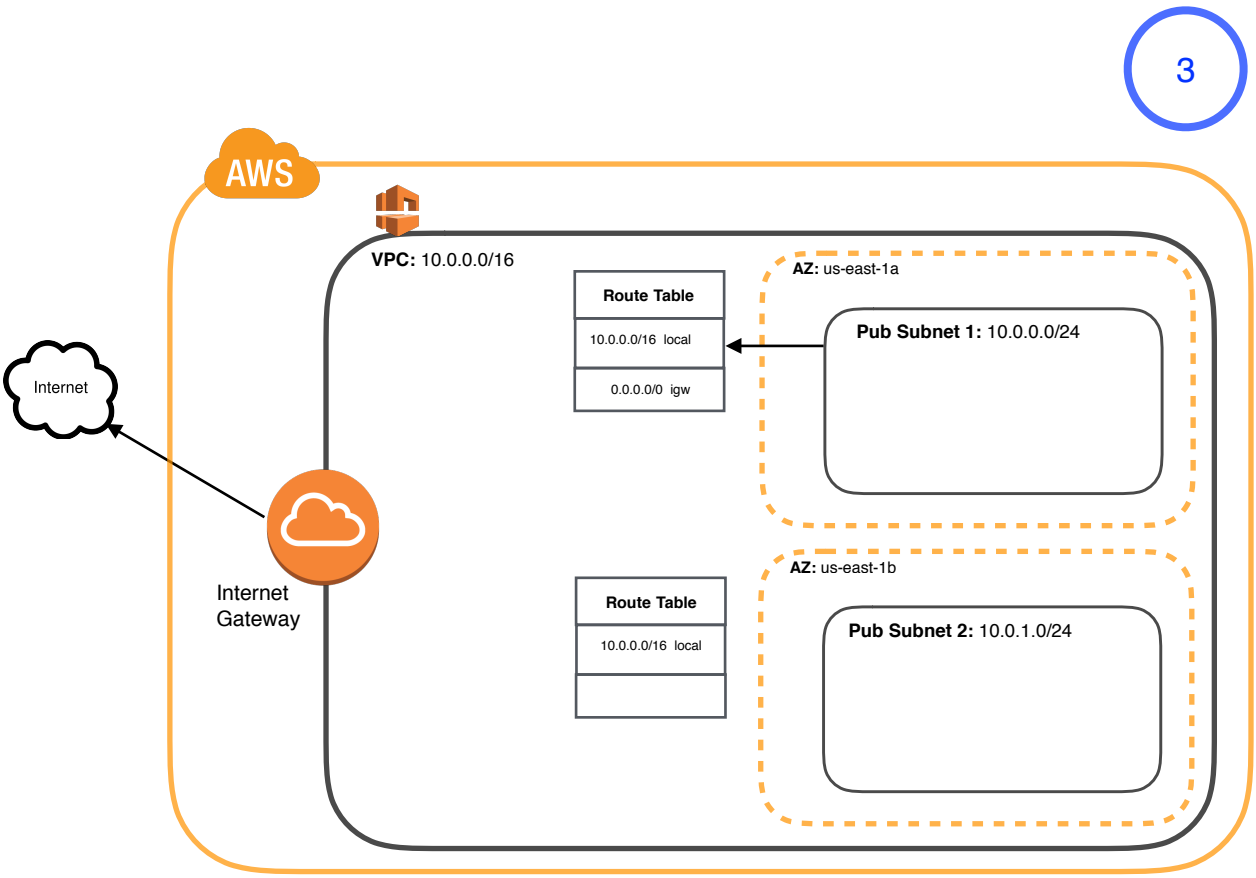
IPv4 CIDR Block: 10.0.1.0/24

This should leave you with something like the screenshot below. Ignore the fact that CIDR values in the image differ to your configuration (This is because VPC's already exist in my account with the values you are using):



4. Click the blue **Yes, Create** button when you are ready

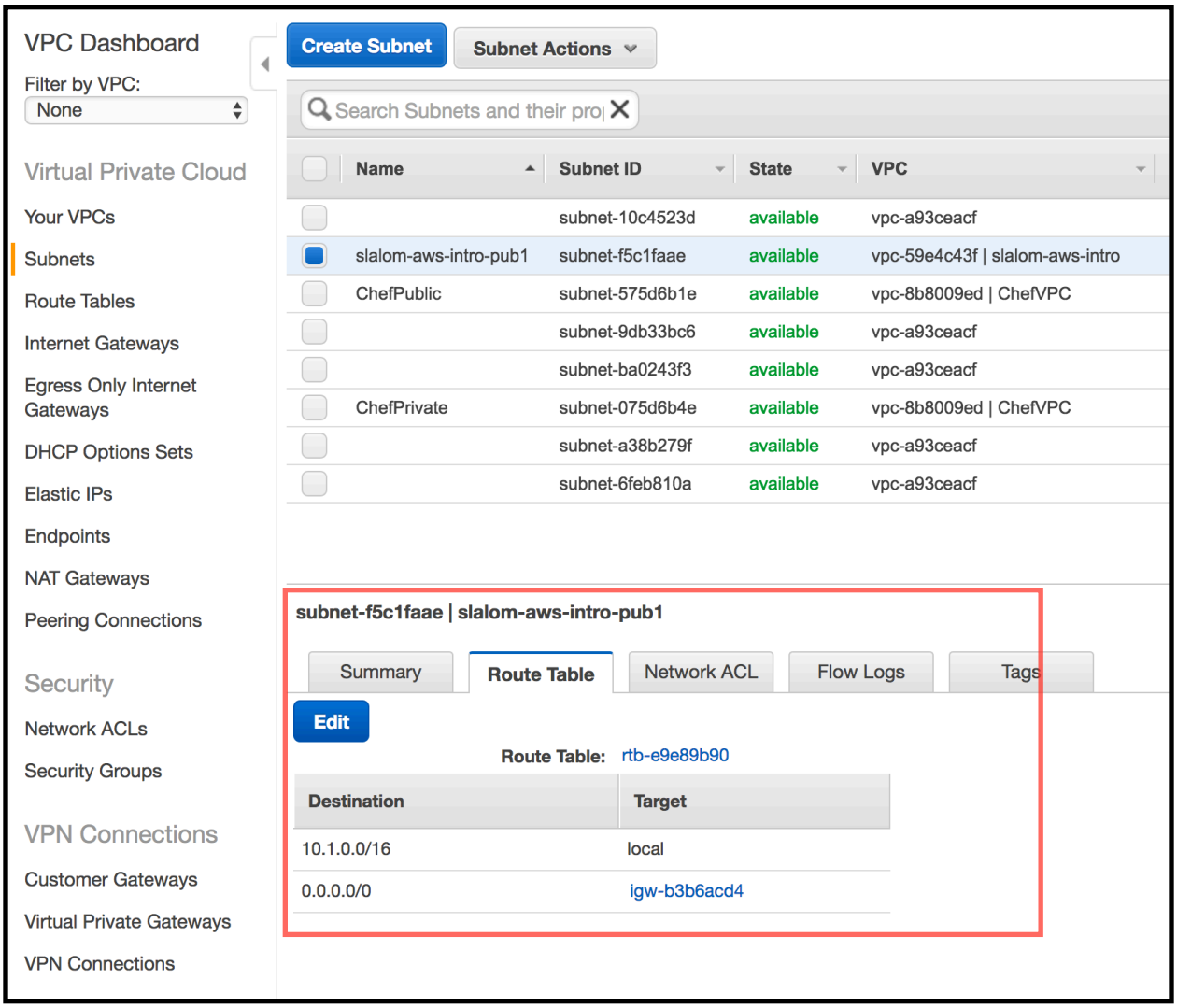
System Architecture: Step 3



Step 4: Route Table Association

1. Navigate to **Services —> VPC** and click on the **Subnets** link in the left hand menu

2. Select the **slalom-aws-intro-pub1** subnet and then navigate to the **Route Table** tab:



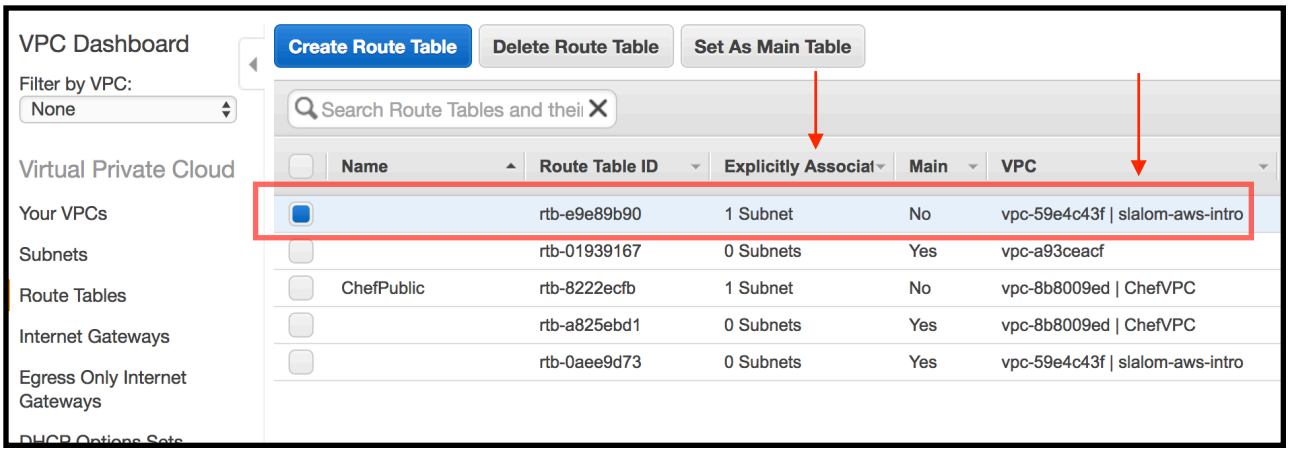
3. Within the Route Table tab above, notice that there are two route entries configured: one for local and one to the igw (Internet Gateway)

4. Now, select the slalom-aws-intro-pub2 subnet and navigate to the Route Table tab. You should notice that this subnet only contains a single route to local.

   To ensure that traffic from the slalom-aws-intro-pub2 subnet can reach the Internet, we need to associate the route table from slalom-aws-intro-pub1 with slalom-aws-intro-pub2

5. Click on the Route Tables link in the left menu

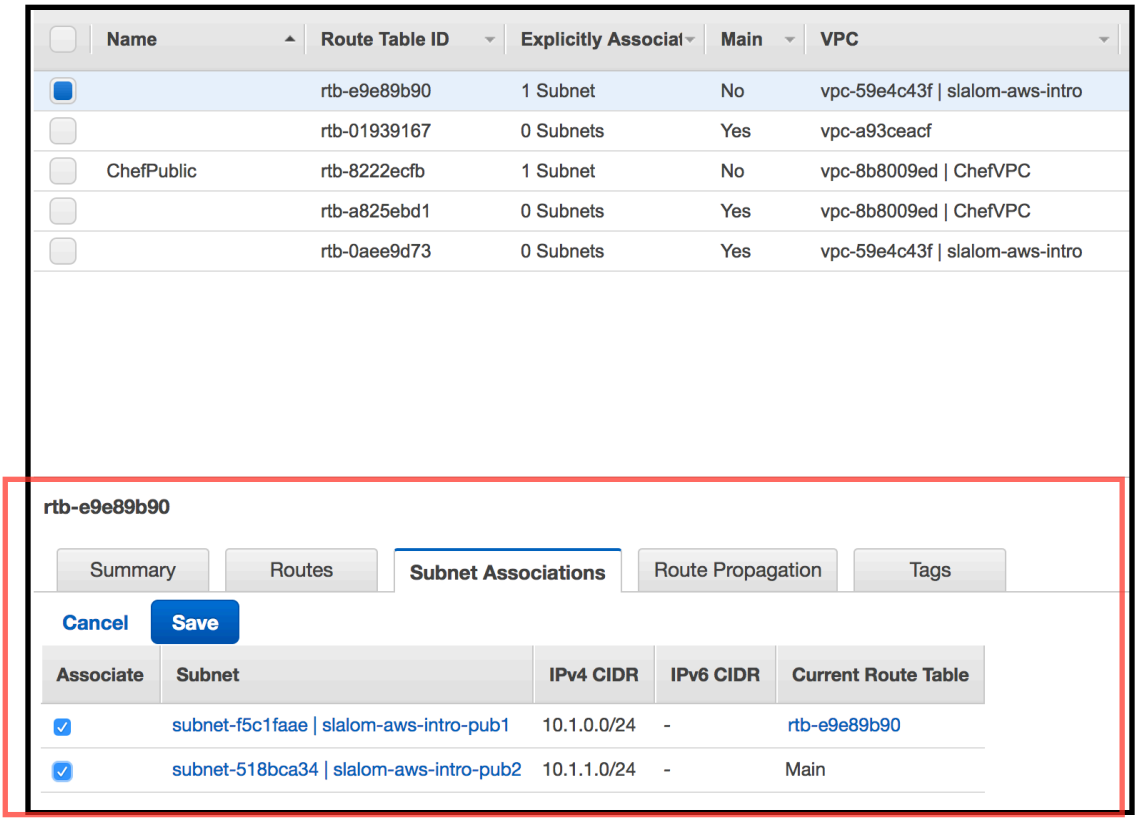
6. Select the Route table with the slalom-aws-intro in the VPC column that is already associated with 1 Subnet. **For example**:



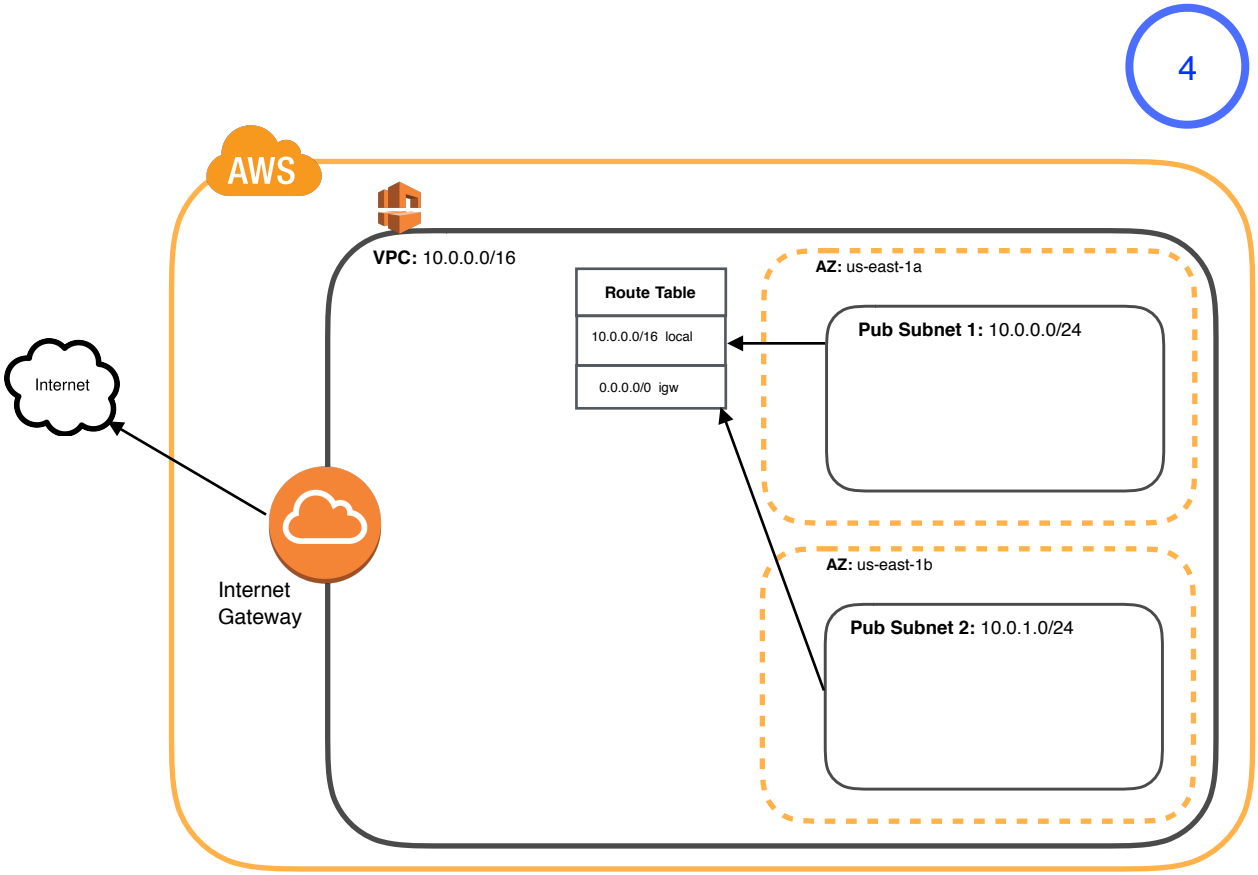
8. With the correct route table selected, navigate to the **Subnet Associations** tab at the bottom (see screenshot below)

9. Click **Edit**

10. Select the checkbox next to the **slalom-aws-intro-pub2** subnet so that both slalom-aws-intro public subnets are selected and click **Save**

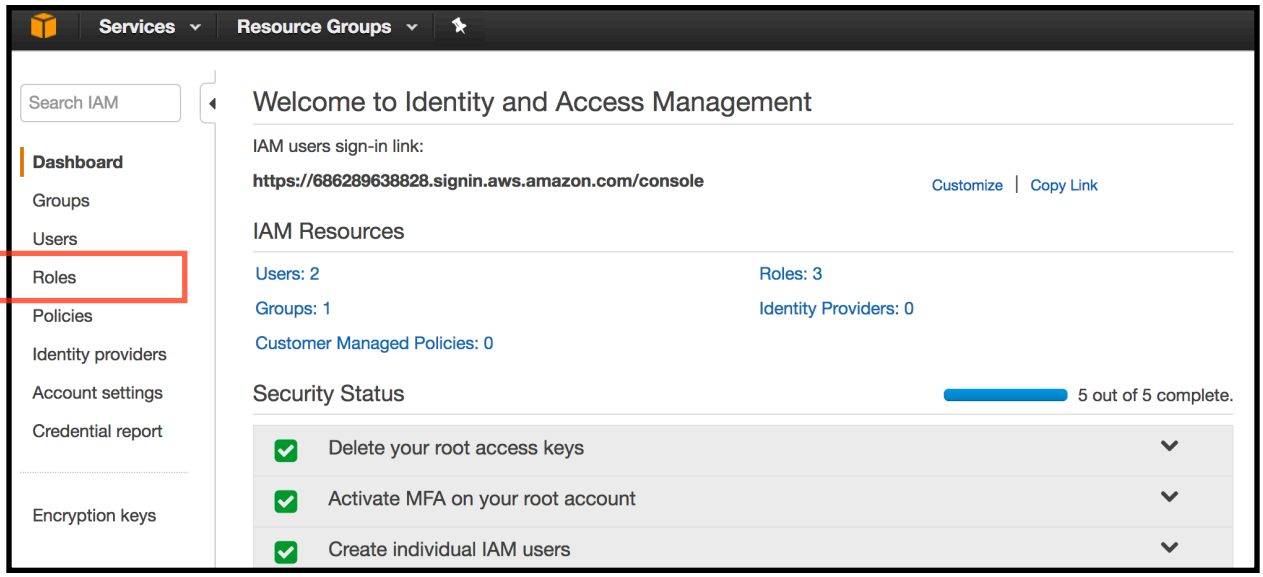


System Architecture: Step 4

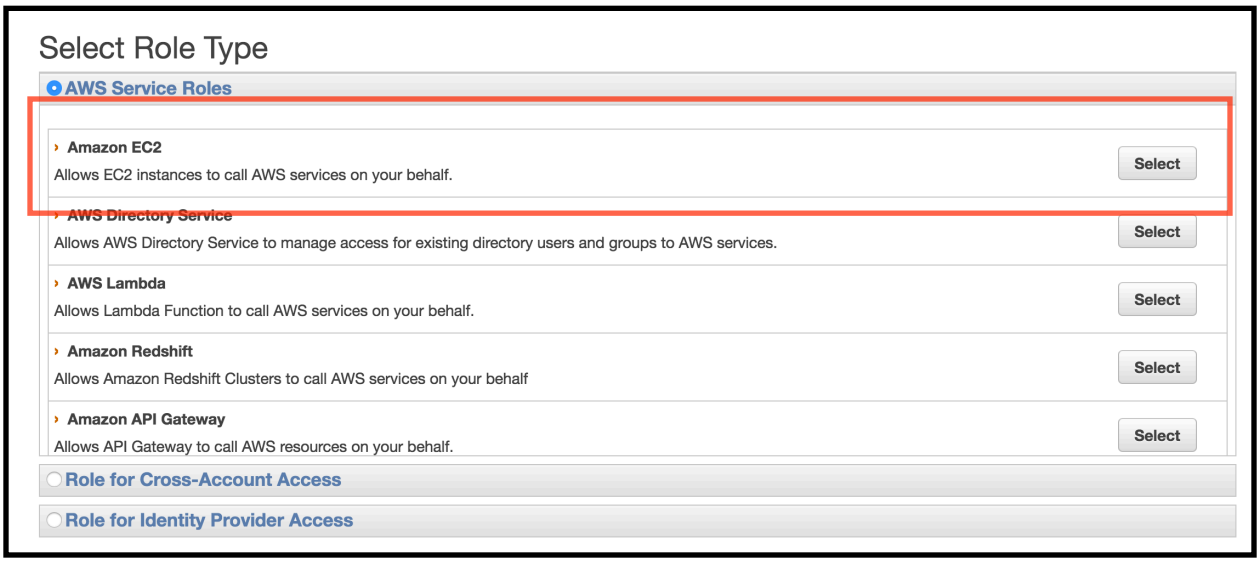


Step 5: Create an IAM Role

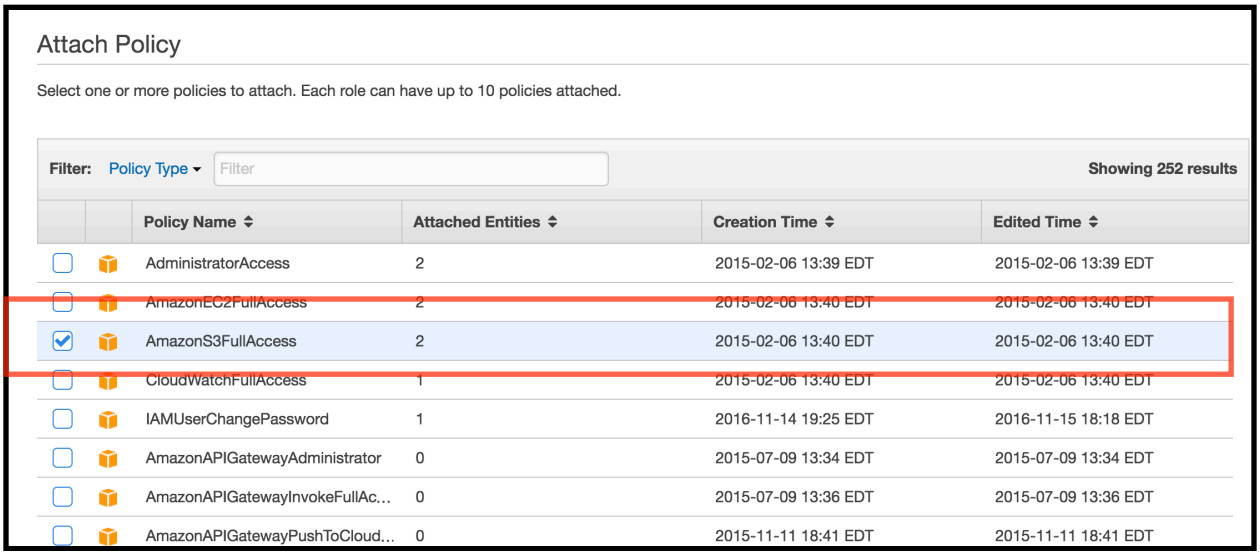
1. Navigate to **Services —> IAM**
2. Select the **Roles** link in the left menu



3. Select the blue **Create New Role** button
4. Enter **slalom_web_server** as the Role Name and click **Next Step**
5. In the Select Role Type screen, click the Select button in the **Amazon EC2** section:

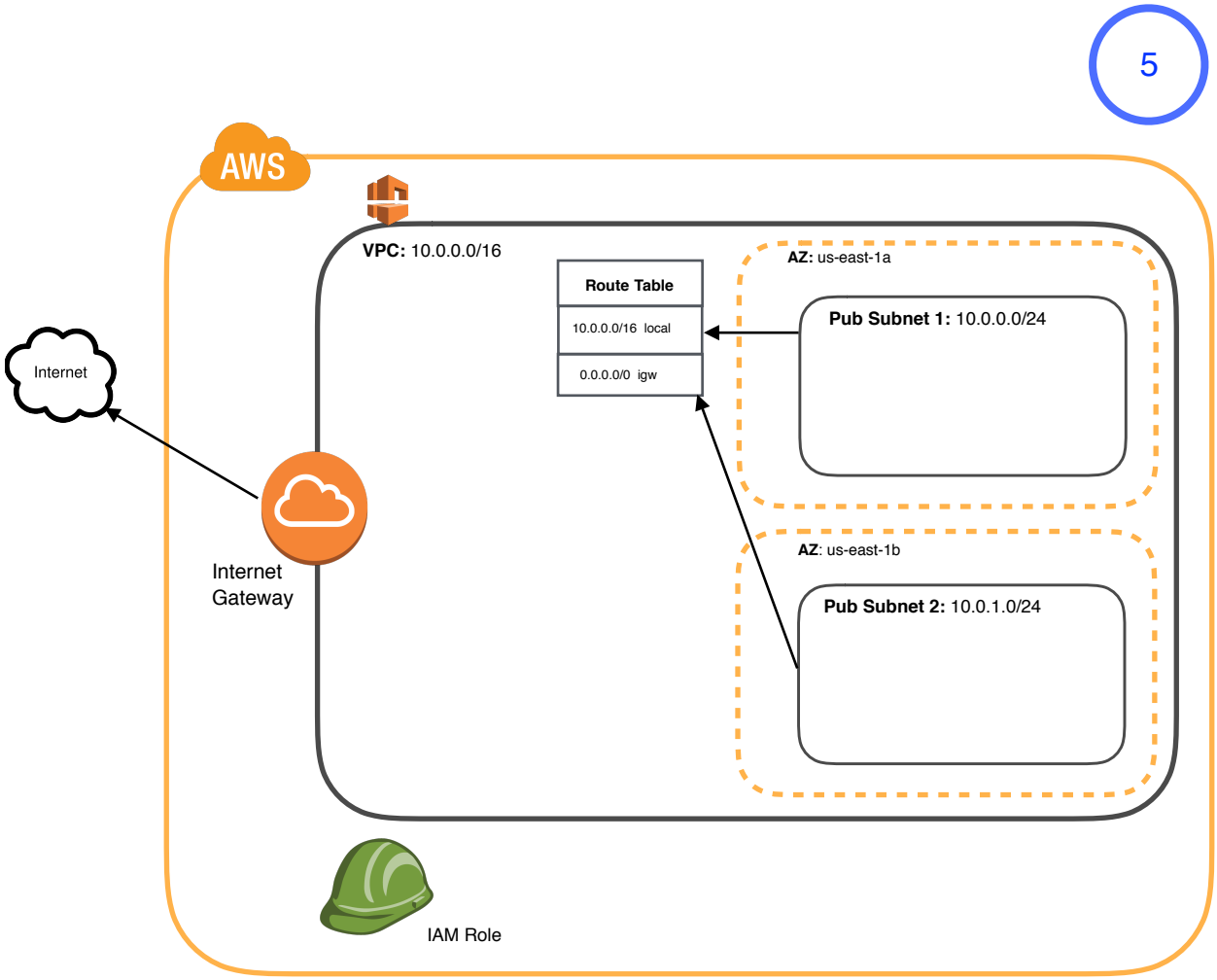


6. In the Attach Policy screen, select **AmazonS3FullAccess** and click Next Step:



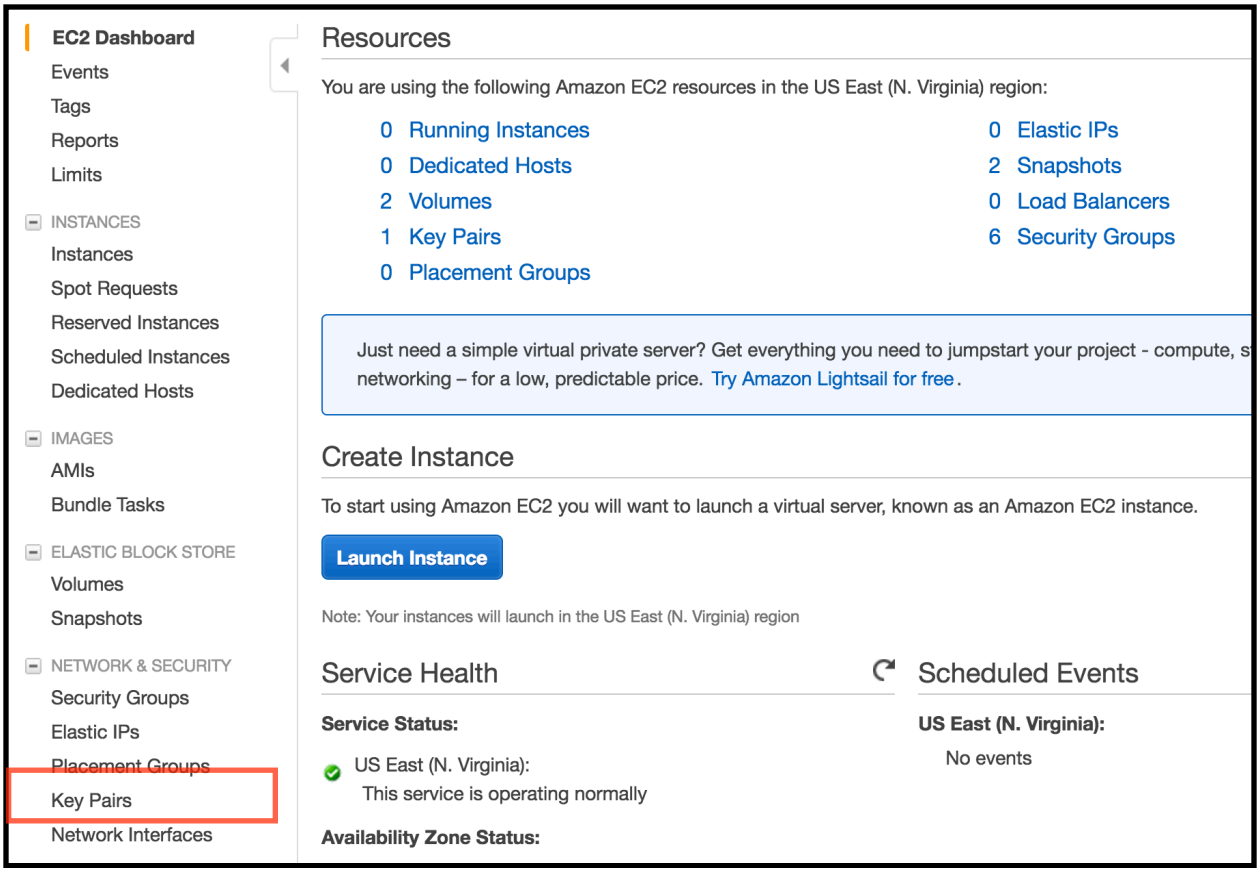
7. Click the **Create Role** button to create the slalom_web_server IAM Role

System Architecture: Step 5



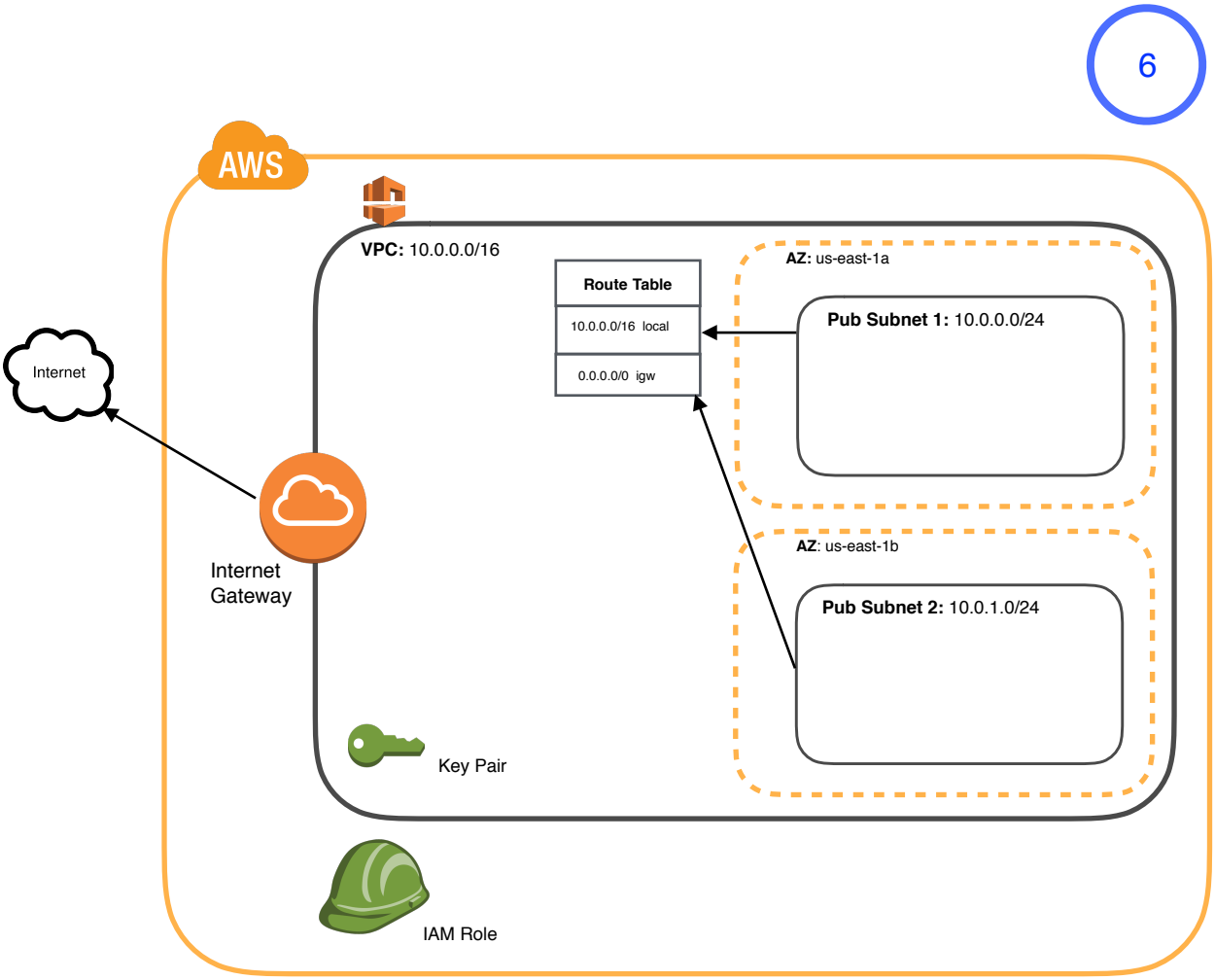
Step 6: Create A Key Pair

1. Navigate to **Services —> EC2**
2. In the left menu, select **Key Pairs**:



3. Select the blue **Create Key Pair** button
4. For Key pair name, enter **slalom_aws_intro** and click the **Create** button
5. This will create a key pair for you in AWS and a file called slalom_aws_intro.pem file will download in your browser
6. Having access to this key file would be important for you if you wanted to log in using ssh (Linux Servers) or RDP (Windows Servers) to any virtual server you create in AWS. **However,** this is beyond the scope of this intro tutorial.

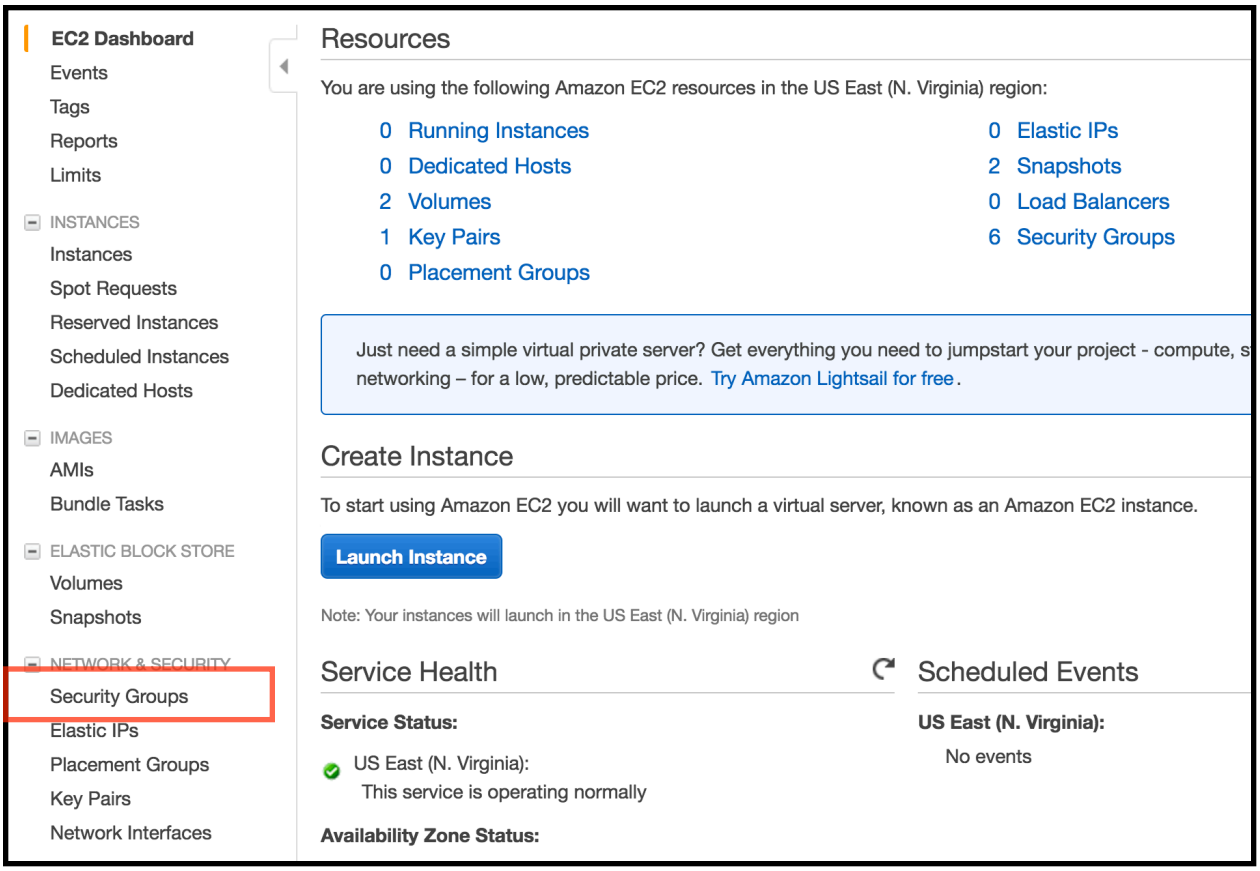
System Architecture: Step 6



Step 7: Create a security group

1. Navigate to **Services —> EC2**

2. In the left menu, click the Security Groups link



3. Click the blue **Create Security Groups** button

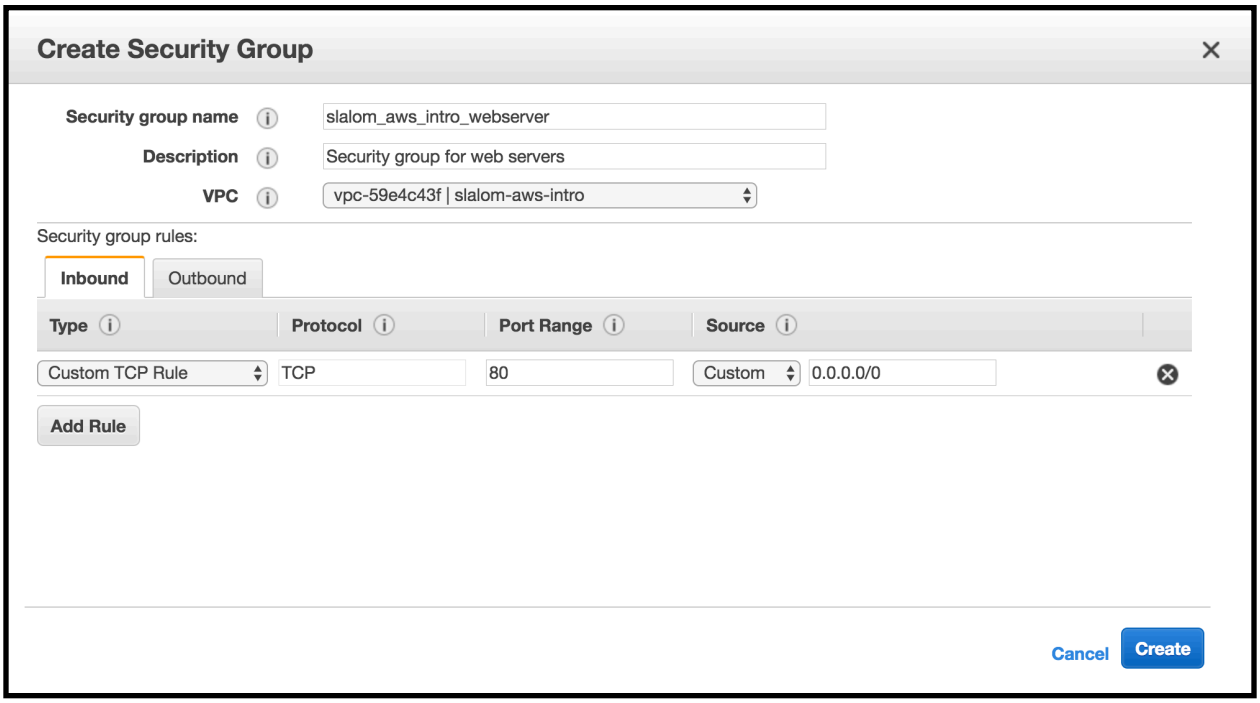
4. In the Create Security Group screen, enter the following information:

Security group name: slalom_aws_intro_webserver
Description: Security group for web servers
VPC: slalom-aws-intro

5. On the Inbound tab, click **Add Rule** and add the following:

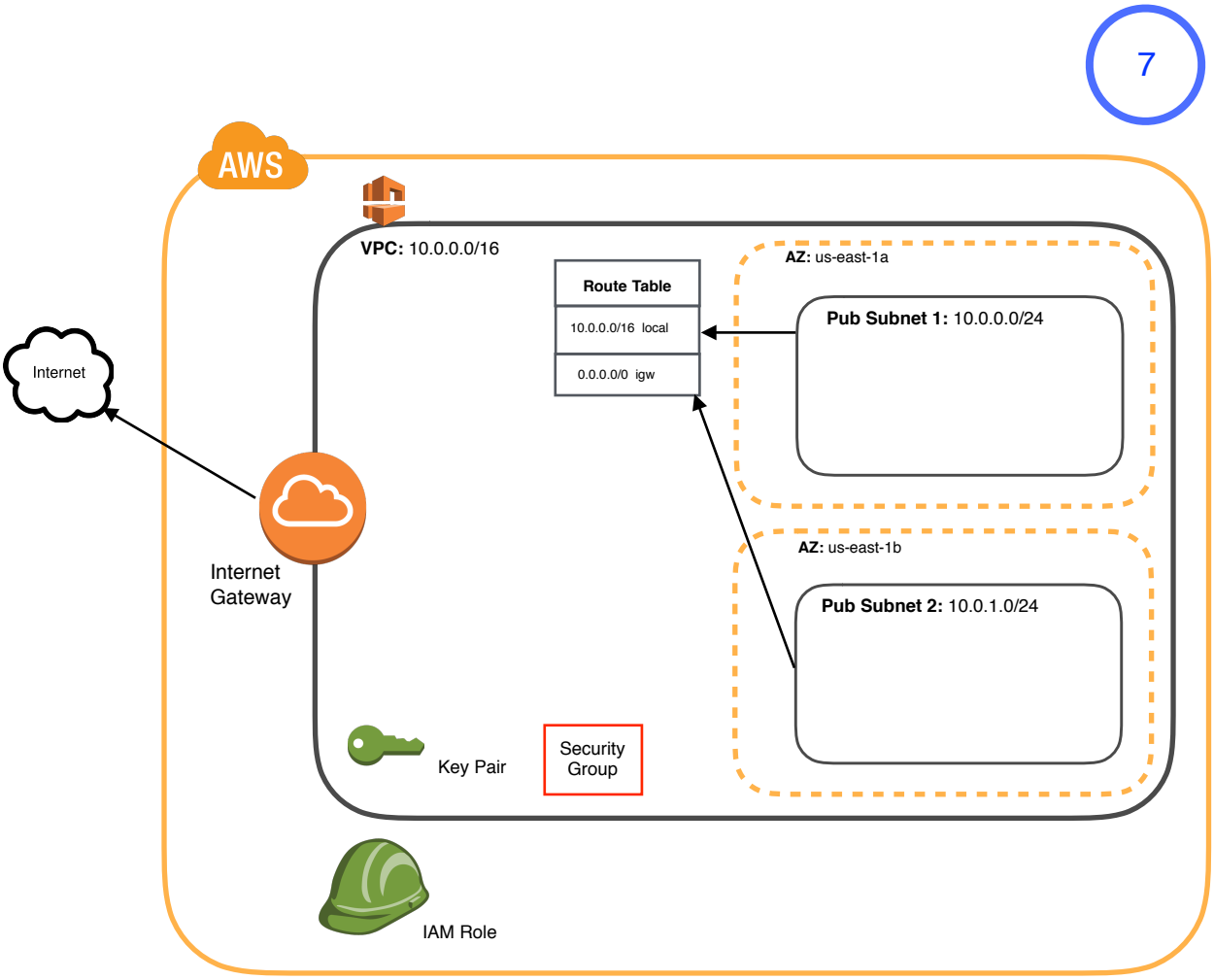
Type: Custom TCP Rule
Protocol: TCP
Port Range: 80
Source: Custom 0.0.0.0/0

This should look like the following:



6. Click the blue **Create** button when you are ready

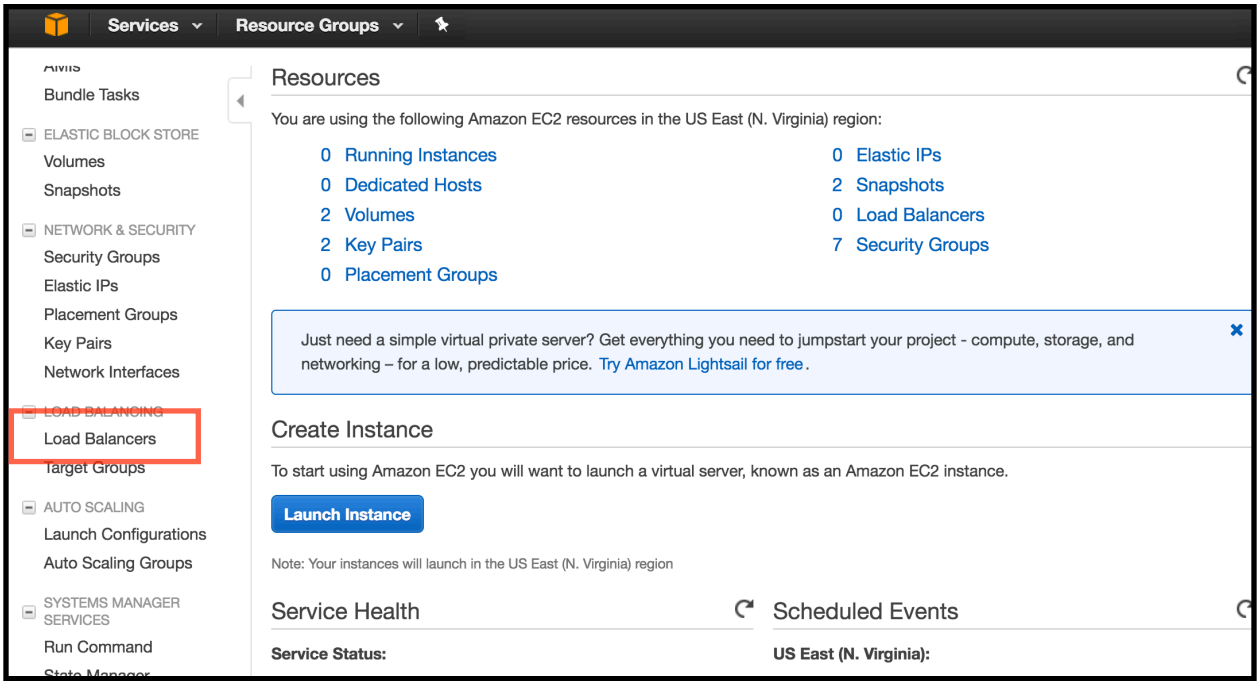
System Architecture: Step 7



Step 8: Create a Load Balancer

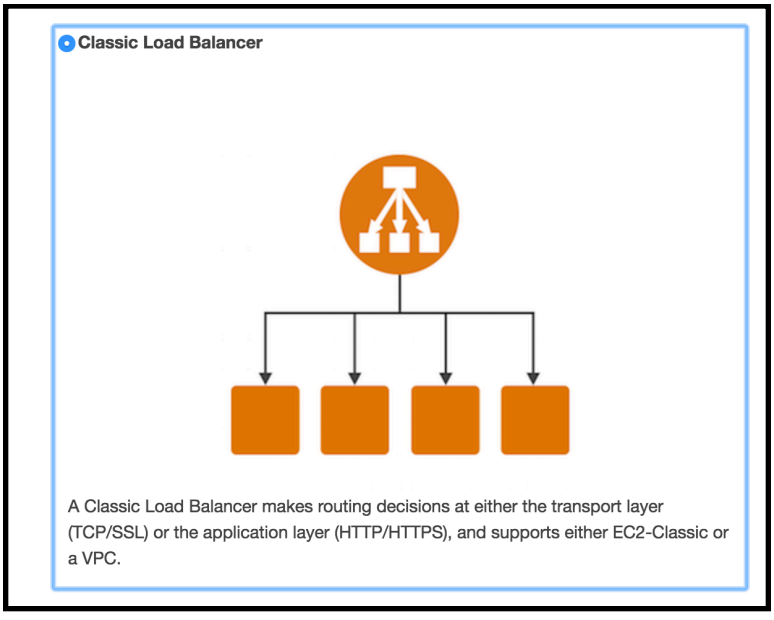
1. Navigate to **Services —> EC2**

2. In the left menu, select **Load Balancers**



3. Select the blue **Create Load Balancer** button

4. In the following screen, select the **Classic Load Balancer** option and click the blue **Continue** button

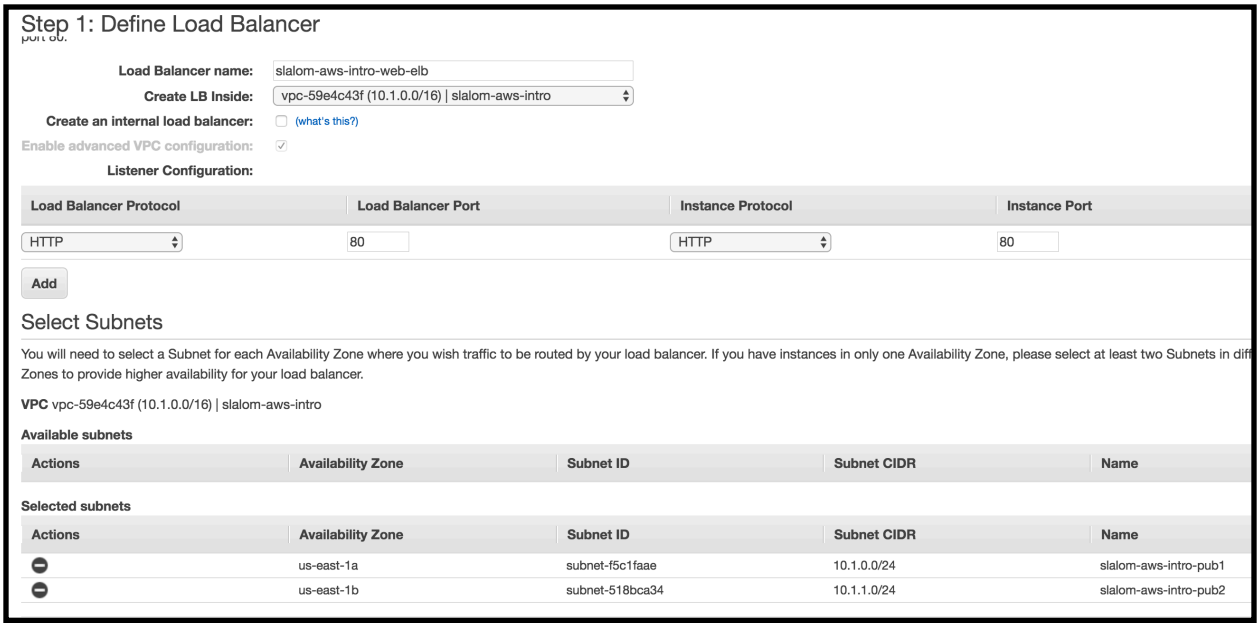


5. In the Define Load Balancer page, enter the following information:

Load Balancer Name: slalom-aws-intro-web-elb
Create LB Inside: slalom-aws-intro

6. In the Select Subnets section at the bottom, click the + symbols beside both **slalom-aws-intro-pub1** and **slalom-aws-intro-pub2**

   This should leave you with the following setup:



7. Click the **Next: Assign Security Groups** button at the bottom right

8. Use the **Select an existing security group** option and select **slalom_aws_intro_webserver** from the list of security groups

9. Click the **Next: Configure Security Settings** button

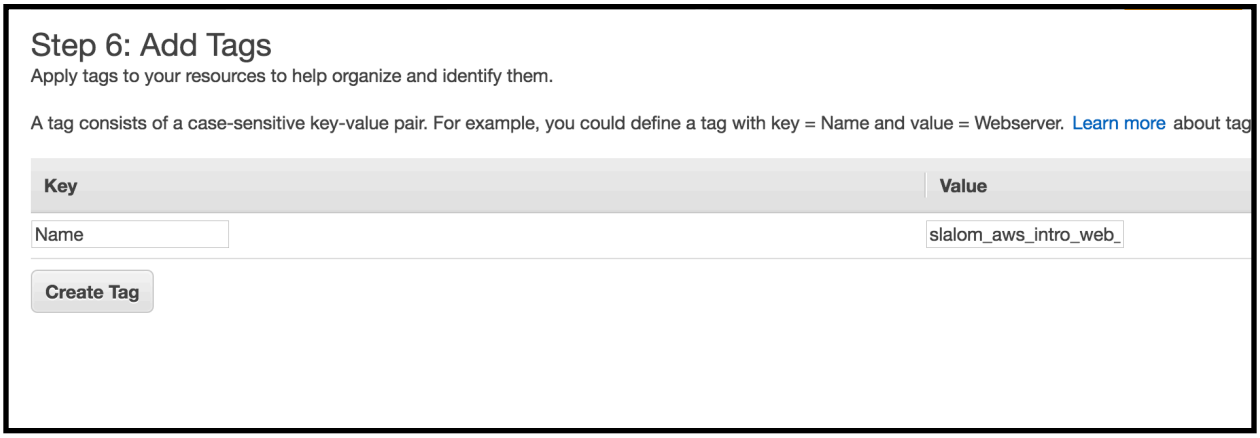
10. A warning message will appear, informing us that we can improve the load balancers security by using a secure listener. For the purposes of this workshop we will be using HTTP instead of HTTPS, so we can continue by clicking **Next: Configure Health Check**

11. We can leave the configured health check as it is, so click **Next: Add EC2 Instances**

12. We don't have any EC2 instances running currently, so continue to **Next: Add Tags**

13. Add a tag with the following values:

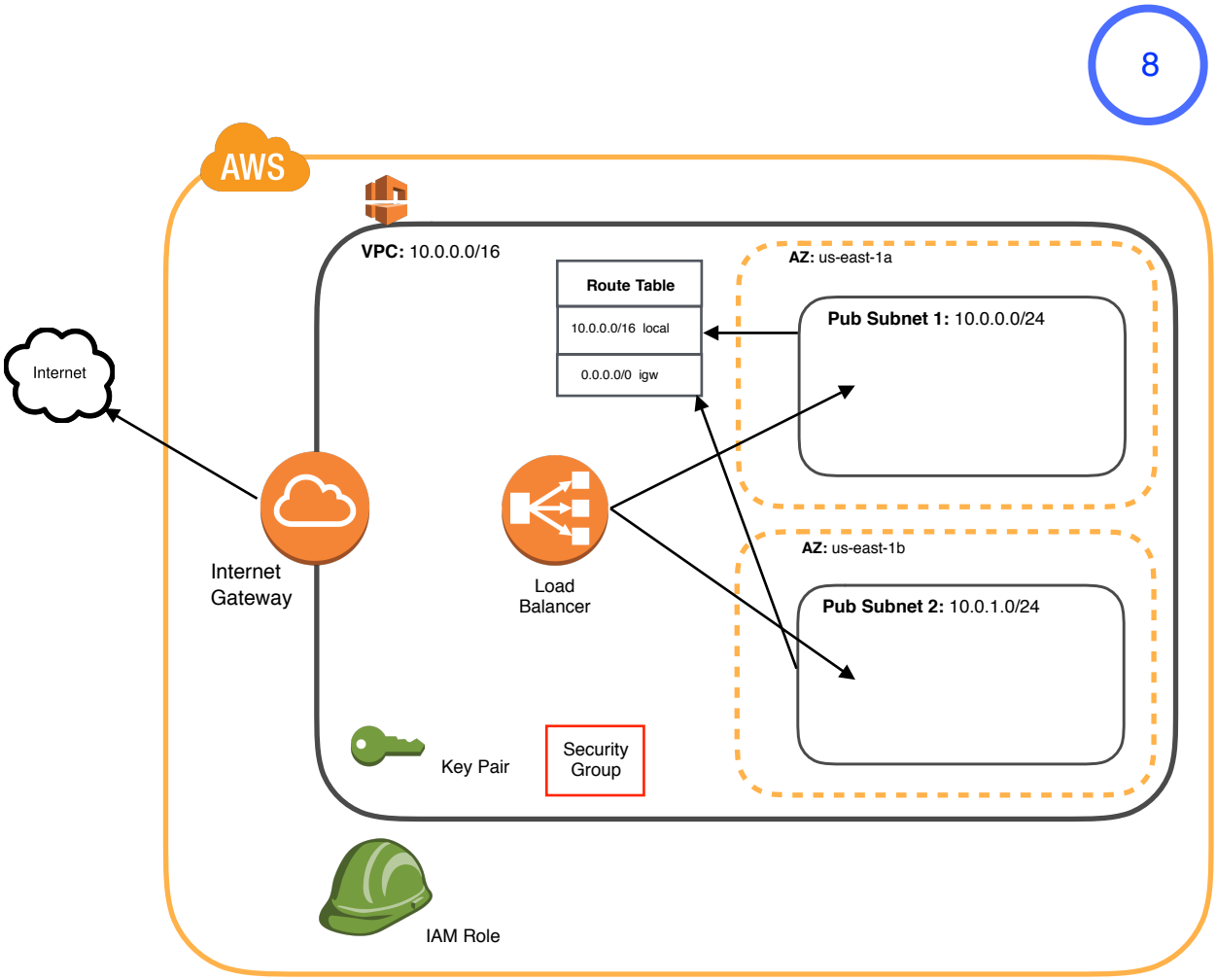
Key = Name
Value = slalom_aws_intro_web_elb



14. Click the blue **Review and Create** button at the bottom right

15. Click the blue **Create** button at the bottom right

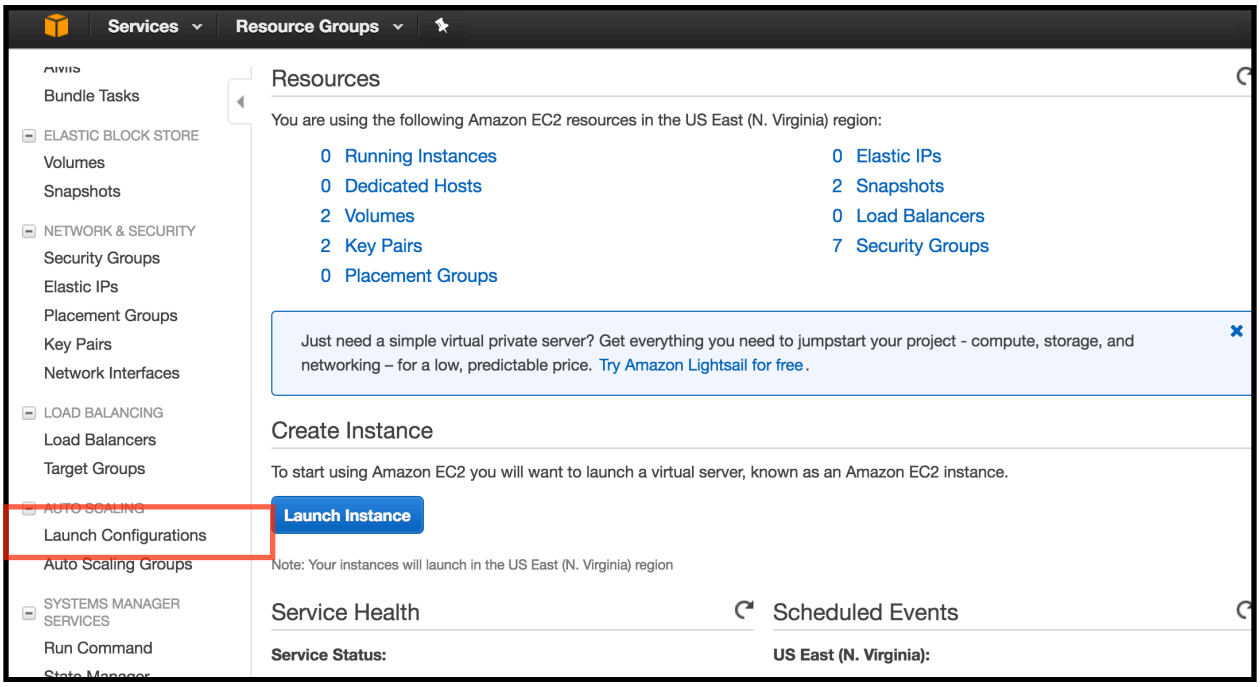
System Architecture: Step 8



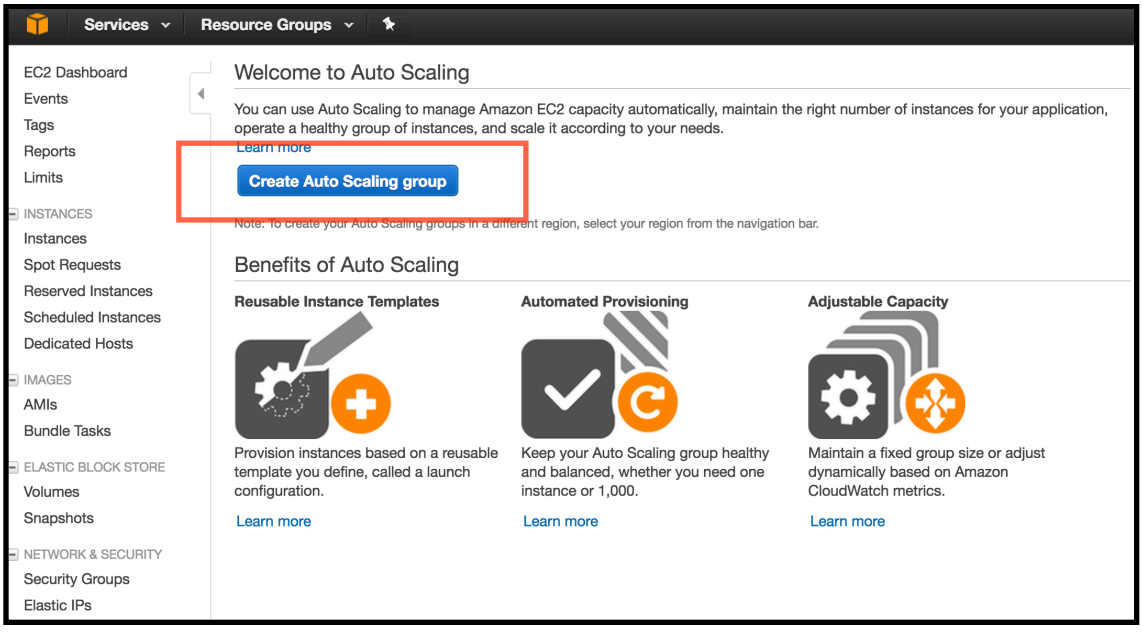
Step 9: Create A Launch Configuration for our Web Servers

1. Navigate to **Services —> EC2**

2. In the left menu, scroll down and select **Launch Configurations**:



3. Select the blue **Create Auto Scaling group** button:



4. At the bottom right of the screen, select the blue **Create launch configuration** button

5. Click the blue **Select** button for the **Amazon Linux AMI 2016.09.1**

6. For instance type, keep **t2.micro** selected and click **Next: Configure details**

7. On the Create Launch Configuration Page, enter the following:

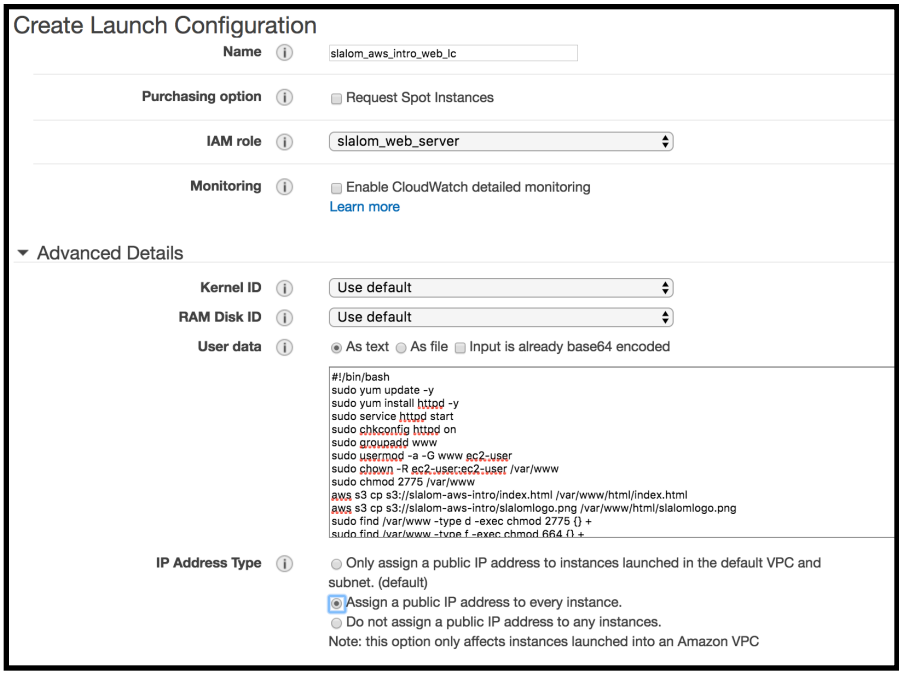
Name: slalom_aws_intro_web_lc
IAM Role: slalom_web_server

7. Select the **Advanced Details** option and copy and paste the following script in to the white box using the **User data** (as text) option:

```
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo service httpd start
sudo chkconfig httpd on
sudo groupadd www
sudo usermod -a -G www ec2-user
sudo chown -R ec2-user:ec2-user /var/www
sudo chmod 2775 /var/www
aws s3 cp s3://slalom-aws-intro/index.html /var/www/html/index.html
aws s3 cp s3://slalom-aws-intro/slalomlogo.png /var/www/html/slalomlogo.png
sudo find /var/www -type d -exec chmod 2775 {} +
sudo find /var/www -type f -exec chmod 664 {} +
```

IP Address Type - Assign a public IP address to every instance

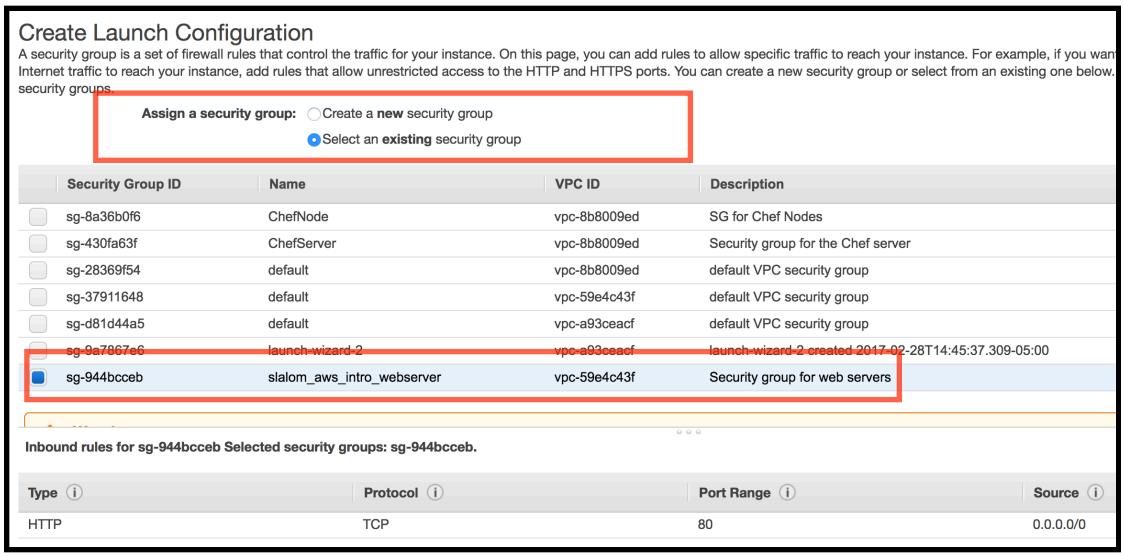
This should leave you with the following configurations:



8. Once you have the above configuration, click **Next: Add Storage**

9. Leave the configured storage settings and click **Next: Configure Security Group**

10. Choose the option to **Select an existing security group** and then select the Security Group called **slalom_aws_intro_webserver** that was created earlier:

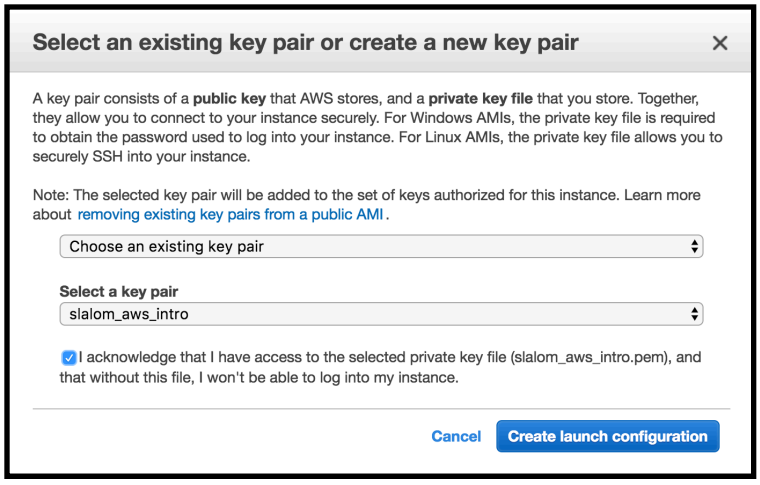


11. Click the blue **Review** button at the bottom right

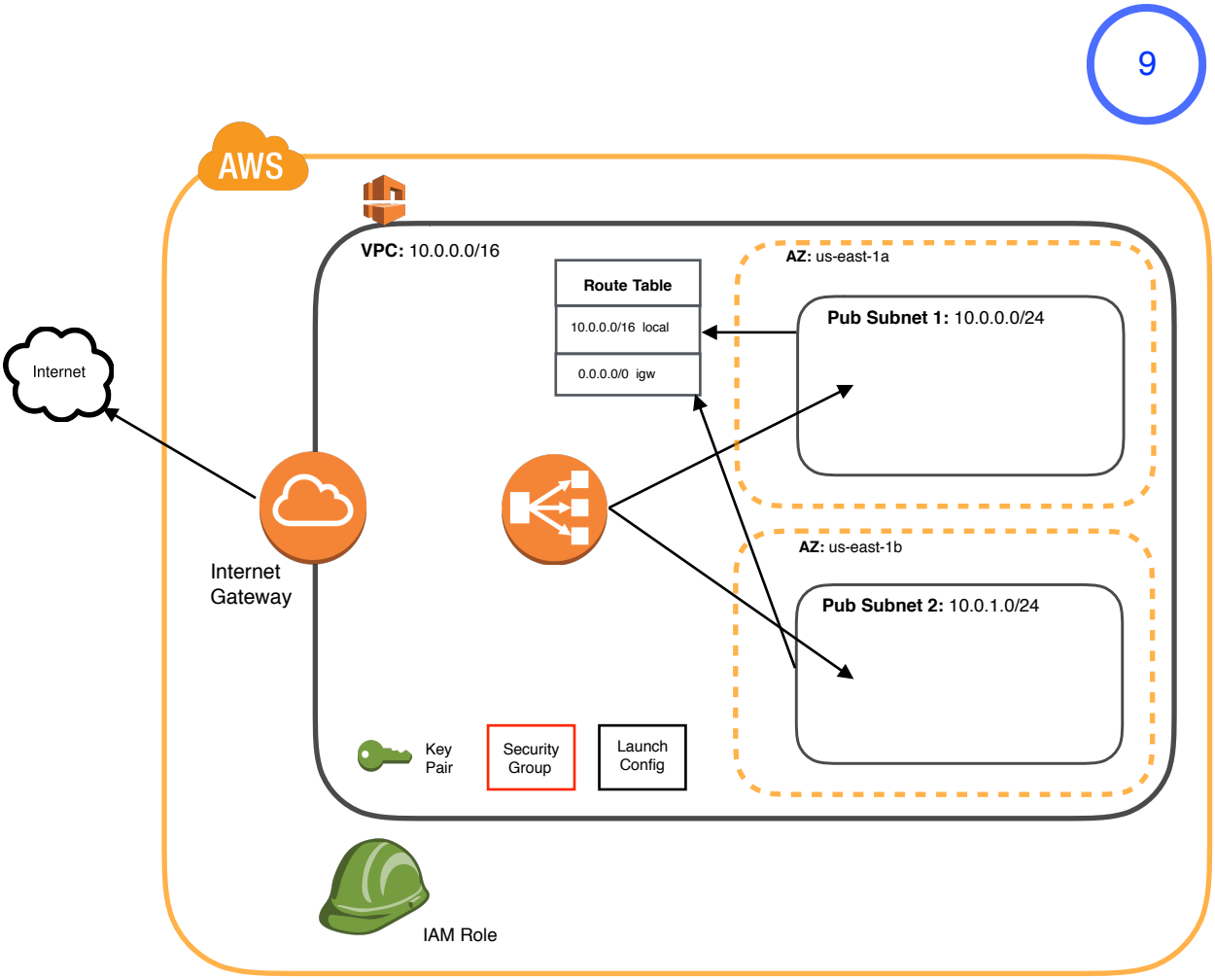
12. A warning message will appear telling you that you won't be able to access the server. This is ok for our purposes, so click **Continue** at the bottom right

13. Click the **Create launch configuration** button

14. In the key pair box that appears, select the **slalom_aws_intro** key that you created earlier and **also** select the checkbox acknowledging that you possess the key pair before clicking the blue **Create launch configuration** button



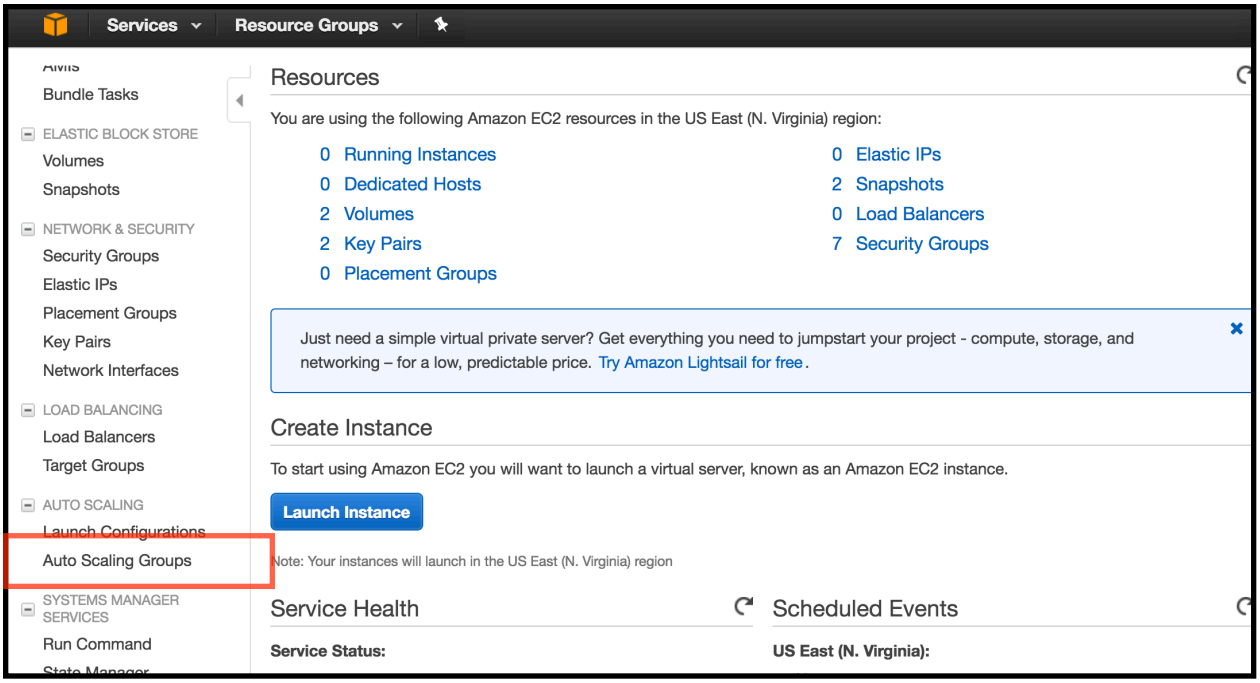
System Architecture: Step 9



Step 10: Create an AutoScaling Group

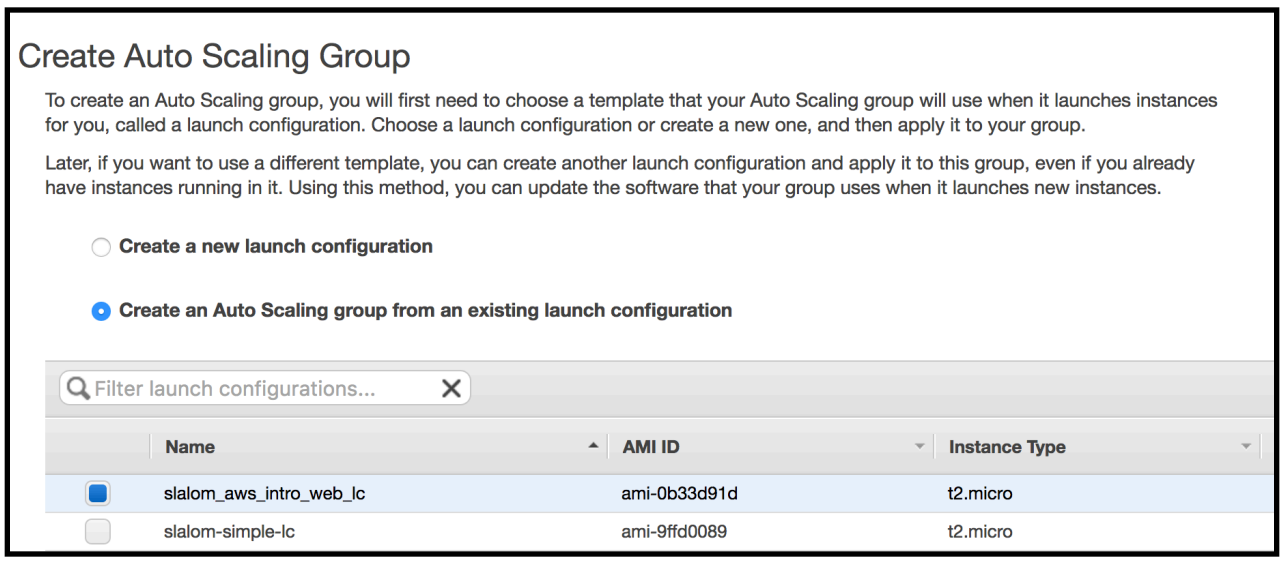
1. Navigate to **Services —> EC2**

2. In the left menu, scroll down and select **Auto Scaling Groups**:



3. Click the blue **Create Auto Scaling group** button at the bottom right

4. Using the **Create an Auto Scaling group from an existing launch configuration** option, select the Launch Configuration **slalom_aws_intro_web_lc** that you just created:



5. In the Create Autoscaling Group box, enter the following options:

Group name: slalom_aws_intro_web_asg

Group Size: 2 instances

Network: slalom-aws-intro

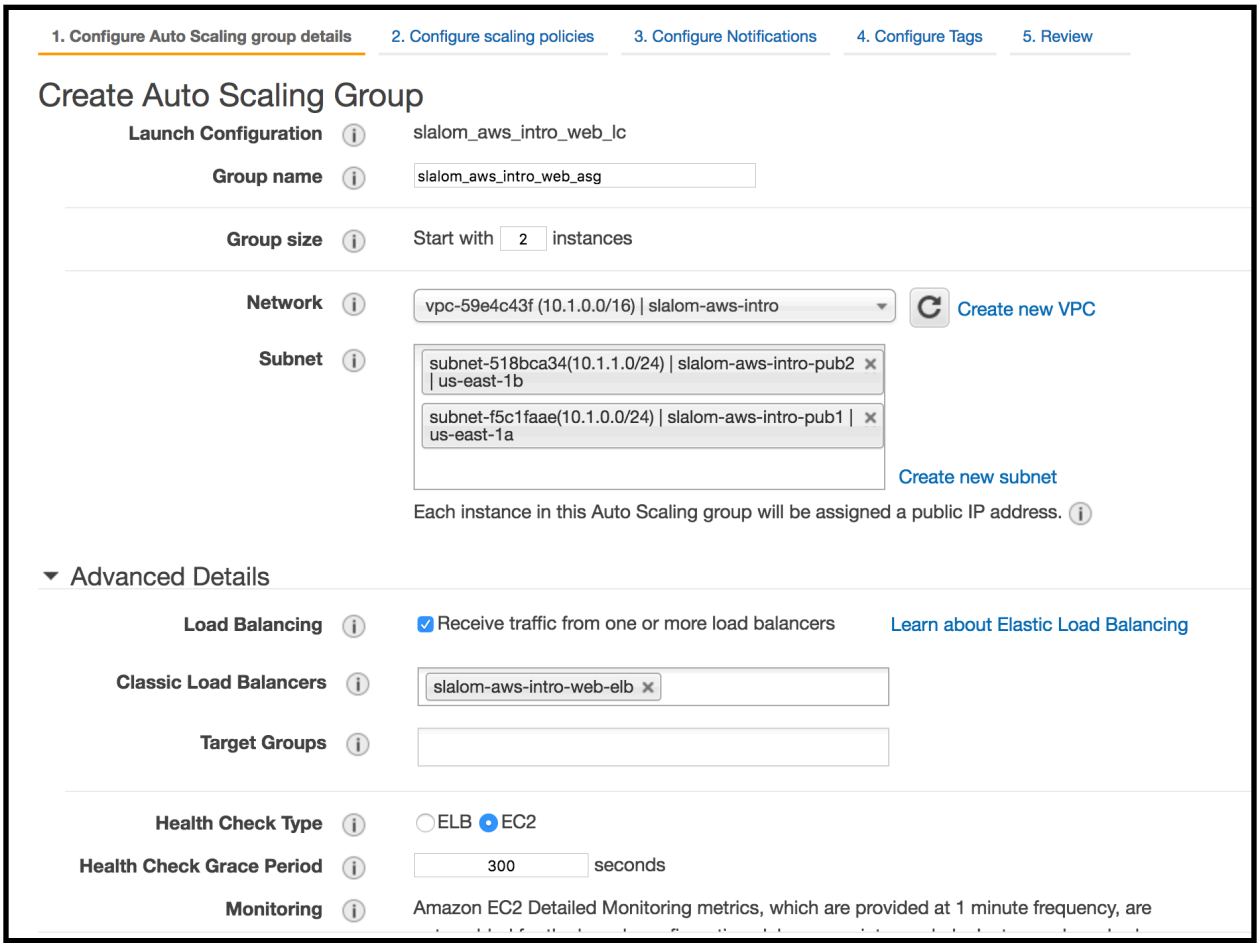
Subnet: Click in the box and add both of the public subnets, **slalom-aws-intro-pub1** and **slalom-aws-intro-pub2**

6. On the same page, click the arrow beside the *Advanced Details* option and enter the following:

Load Balancing: Select the checkbox to Receive traffic from one or more load balancers

Classic Load Balancers: Click in the box and select **slalom-aws-intro-web-elb**

This should result in the following configurations:



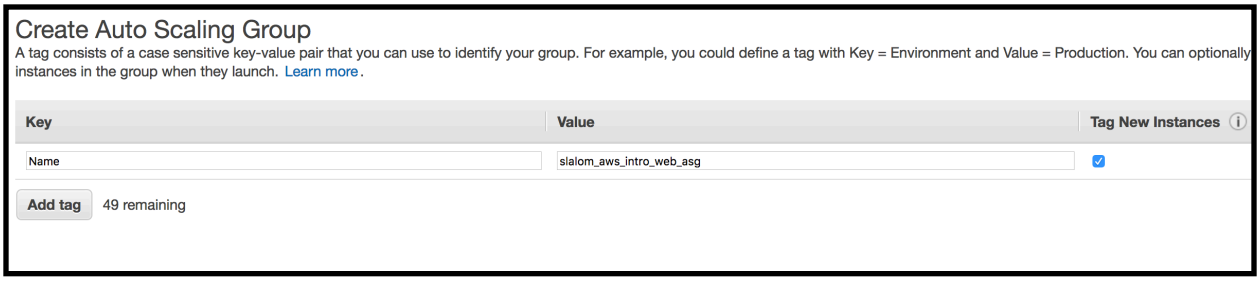
7. Once you have the correct configuration, click **Next: Configure scaling policies**

8. Select the **Keep this group at its initial size option**. Click **Next: Configure Notifications**

9. We don't need to configure any notifications, so continue to click **Next: Configure Tags**

10. Add a tag with the following values:

Key = Name
Value = slalom_aws_intro_web_asg

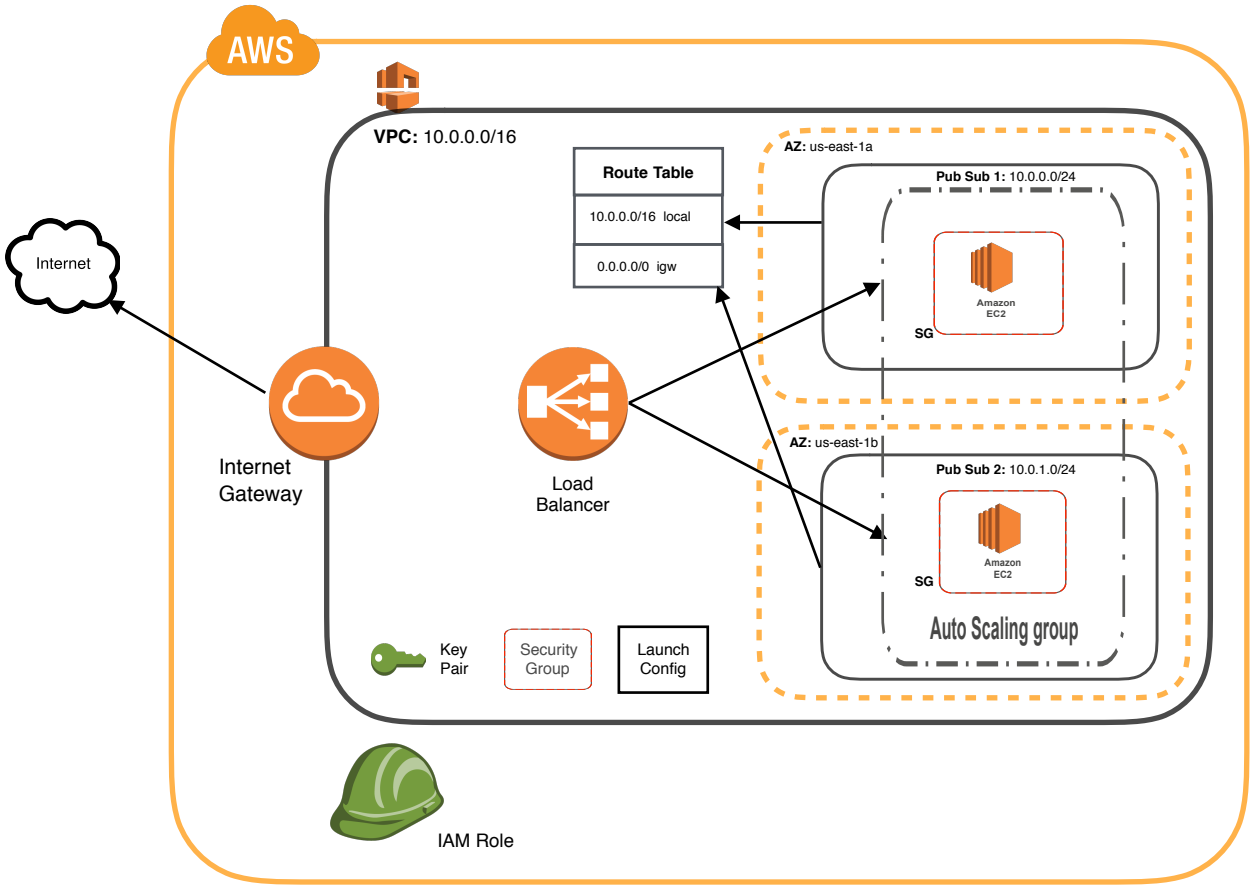


11. Click the blue **Review** button once you have configured the Name Tag

12. Continue to click the blue **Create Auto Scaling group button**

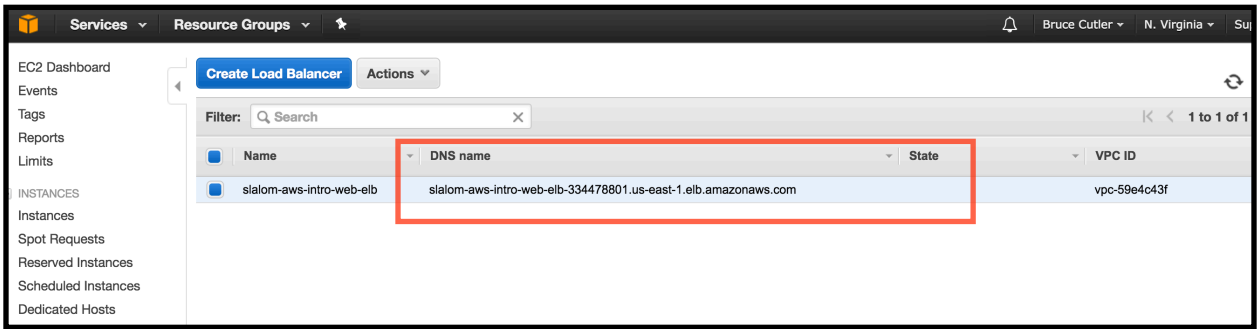
System Architecture: Step 10

10



Step 11: Test it out!

1. Navigate to **Services —> EC2**
2. In the left menu, select **Load Balancers**
3. Navigate to the **slalom-aws-intro-web-elb** load balancer
4. Copy and paste the value in the DNS Name column



5. Navigate to a web browser (Chrome / Firefox etc.), paste the value in and hit enter
6. It may take a few minutes for the EC2 instances to finish creating, so the page may not load immediately within your chosen web browser
7. If all has gone to plan, you should see a Hello message!

System Architecture: Step 11

11