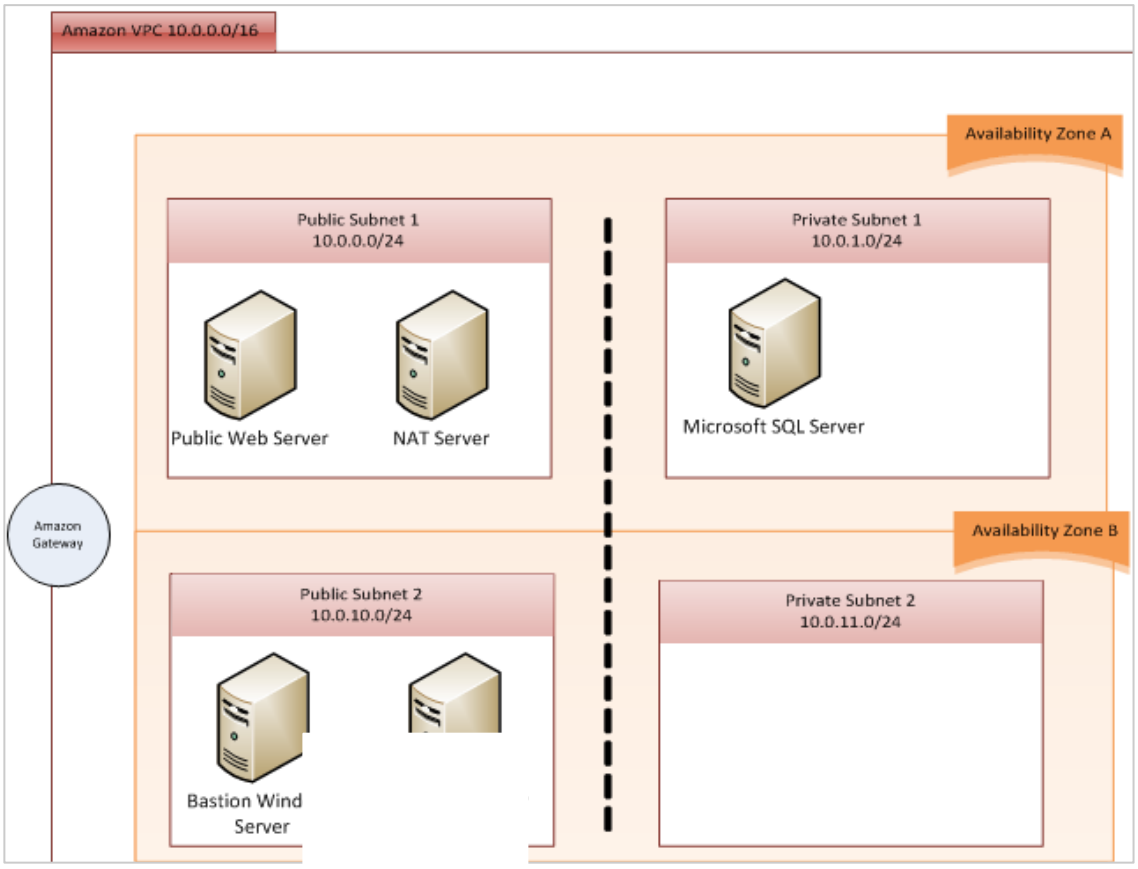
Creating Your First Amazon Virtual Private Cloud (VPC)

Table of Contents

Overview

In this lab session, you create a basic Amazon Virtual Private Cloud (VPC), and then extend it to produce a customized result. You do all of this with the AWS Management Console.

The diagram below is what you will build.



The overall VPC is designed to inc ɜsic features:

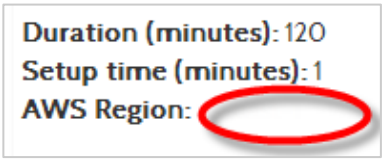
- It spans two Availability Zones (AZs), in order that later you can distribute applications across these zones in order to architect for application durability and availability.
- Within each Availability Zone (AZ) there are two subnets: one “public” subnet is connected directly to the Internet. The other “private” subnet is able to communicate with any other subnet within the VPC; however there is no access to them from the Internet. The dashed line demarcates this isolation.
- You will walk through two alternatives to allowing access to servers that are in the private subnets.

Start your *qwikLAB*™

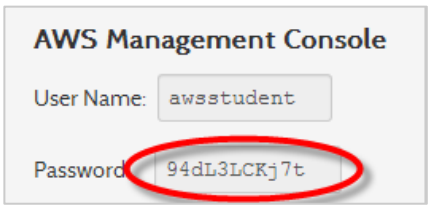
1) To the right of the **Architecting on AWS, Day 1: Your first Virtual Private Cloud** link, click the **Start Lab** button to launch your *qwikLAB*™.
(Hint: If you are prompted for a token, use one you have been given or have purchased.)

You will see the lab creation in progress.

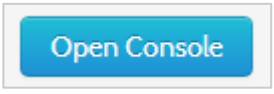
2) On the exercise page, notice the lab properties.
   1) **Duration -** The time the lab will run for before automatically shutting down.
   2) **Setup Time -** The estimated time to setup the lab environment.
   3) **AWS Region** - The AWS Region where the lab resources are being created. (Note: The AWS Region in the image below is intentionally blank. Regions differ depending on the lab setup.)



3) Copy the **Password** provided.
(Hint: Selecting the value shown and pressing CTRL+C works best.)



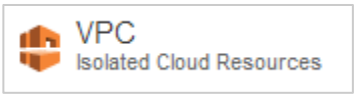
4) Click the **Open Console** button.



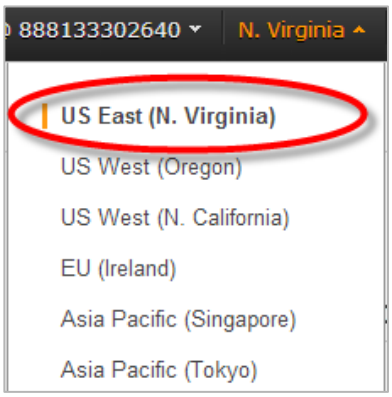
5) Login to the AWS Management Console using the following steps:
   1) In the **User Name** field type **awsstudent**.
   2) Paste the password you copied from the lab details in *qwikLAB*™ into the **Password** field.
   3) Click the **Sign in using our secure server** button.

In this step you logged into the AWS Management Console. Login credentials for the **awsstudent** AWS account are provisioned by *qwikLAB*™ using AWS Identity Access Management.

6) When you are logged into the console, click **VPC**.

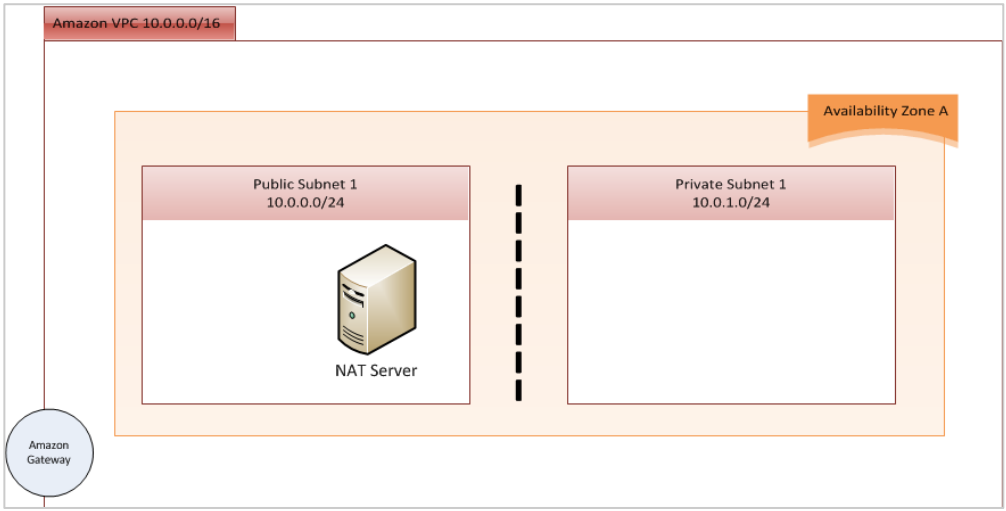


7) Confirm that the same AWS Region shown on the lab page appears in the AWS Management Console toolbar.

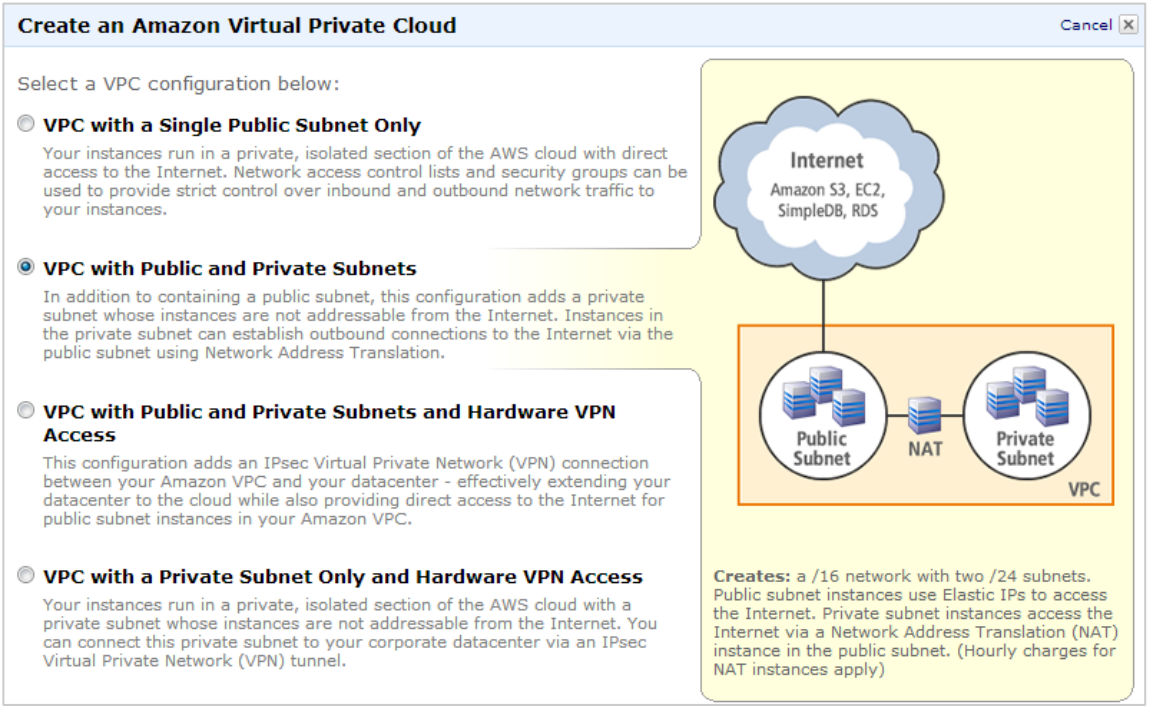


Create the Base VPC

You use a wizard to set up the initial VPC (which is fast and easy), and then you extend the result manually to learn more about VPC configuration options. Initially, you create the following configuration:



1) Click the **Start VPC Wizard** button.
2) Choose the **VPC with Public and Private Subnets** option.



3) Click **Continue**.

The “VPC with Public and Private Subnets” panel contains a lot of parameters. Depending on your professional background, the notation may appear different than what you are used to. This notation is

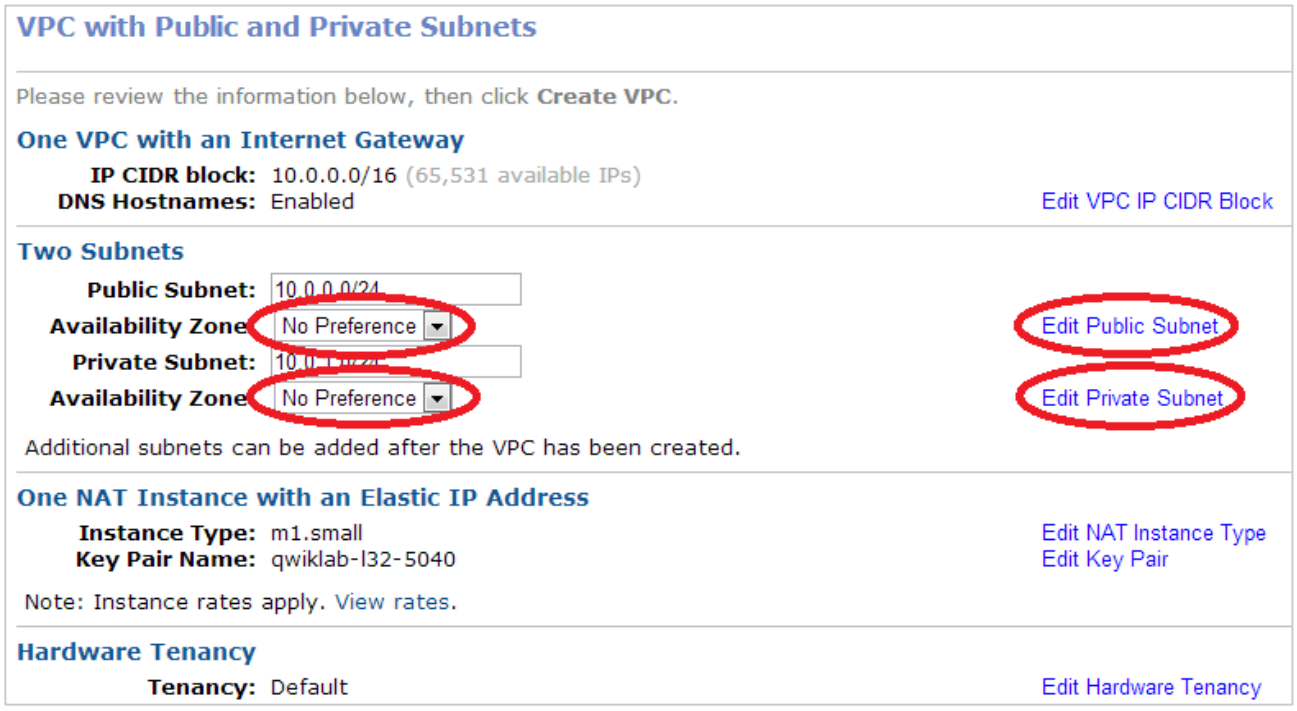
commonly known as CIDR block notation, so, for example, 10.0.1.0/24 can also be expressed as 10.0.1.0 with a subnet mask of 255.255.255.0.

The VPC itself is a Class B network in the 10.0.0.0 space. If you are familiar with the IPv4 address space, this will be familiar as one of the non-routable address blocks. The overall address space uses an IP CIDR block of 10.0.0.0/16, which is the equivalent of a subnet mask of 255.255.0.0 (a full Class B network).

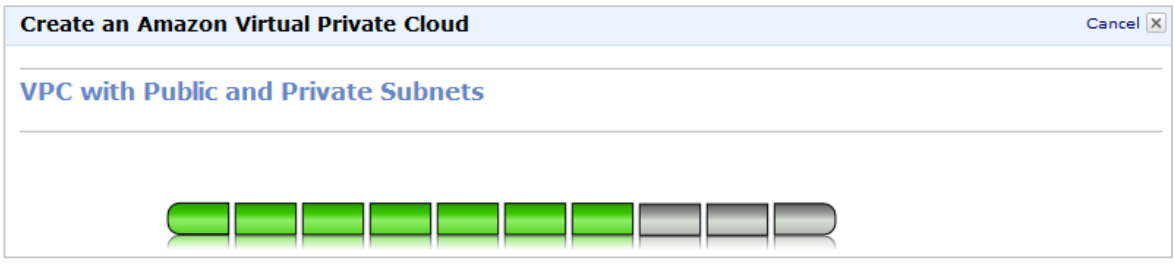
You are going to keep most of the default values, with two exceptions.

4) Click **Edit Public Subnet** and choose an Amazon EC2 Availability Zone (for example – us-east-1a).
5) Click **Edit Private Subnet** and choose the same Availability Zone you selected for the Public Subnet. Also, notice the NAT properties listed in the wizard.

Important: Be certain that the subnets are both in the same Amazon EC2 Availability Zone!



6) After you have selected the same Availability Zone for both the Public and Private Subnet, create the VPC by clicking the **Create VPC** button. This will open a dialog box indicating the progress of creating your VPC.



7) Once the VPC is created, click **Close**.

8) In the VPC Dashboard, the VPC, two subnets, and several other features such as network ACLs and route tables, and so on are displayed. For the moment all that matters is that the network environment is ready to use.

Your VPC does, however, have an important specific: everything is in a single Availability Zone. In order to optimize application availability you need to distribute assets across zones, which means that you need to add another pair of subnets. You will do that later in this lab exercise.

NAT Servers are for Outbound Requests

Note that there is already a running instance, which is the NAT server that the wizard created. The NAT server is an appliance in the sense that its only purpose is to allow servers in the Private subnet to communicate with the Internet in order to get updates, software packages, and so forth. It does not allow Internet clients to make connections to servers in the private subnet. Also note that it is assigned an *Elastic IP*, or NAT (Network Address Translation), address in order to facilitate Internet communication.

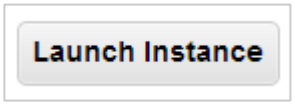
By default the instance type is an m1.small and the EC2 Key Pair Name associated with it is one that was generated for you by *qwikLAB*™. Note: The screen capture below is taken from the VPC wizard above.

Launch a Web Server

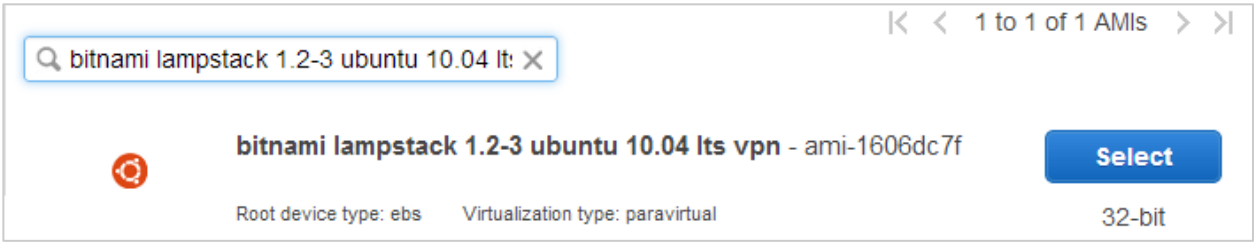
In this lab you launch a BitNami web server as the front-end of your environment. The advantages of this particular AMI are that (a) it was created by a trusted partner, and (b) the Web server will respond to requests using the default configuration.

1) Switch to the EC2 Management Console by clicking **Launch EC2 Instances**.

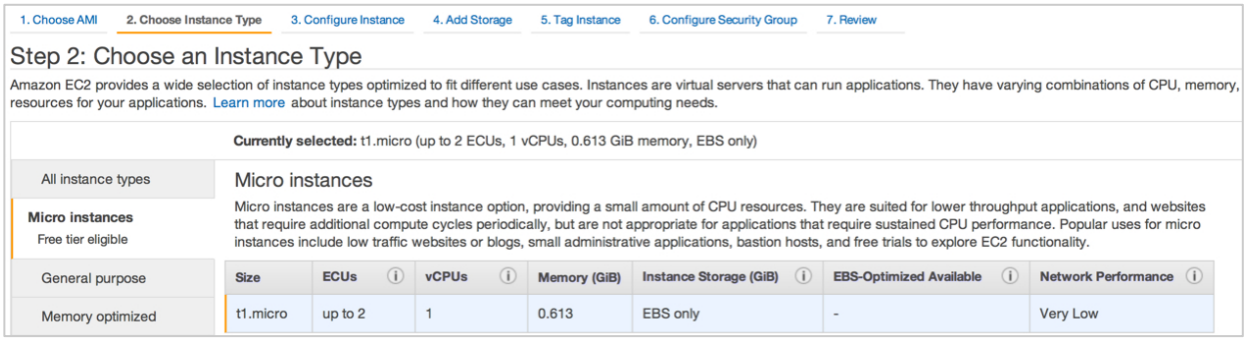
2) Click **Launch Instance**.



3) For Step 1: Choose an Amazon Machine Image, click Community AMIs.
4) In the search window, type **bitnami lampstack 1.2-3 ubuntu 10.04 lts vpn** and press **Enter**.
5) When the AMI appears, click **Select**.

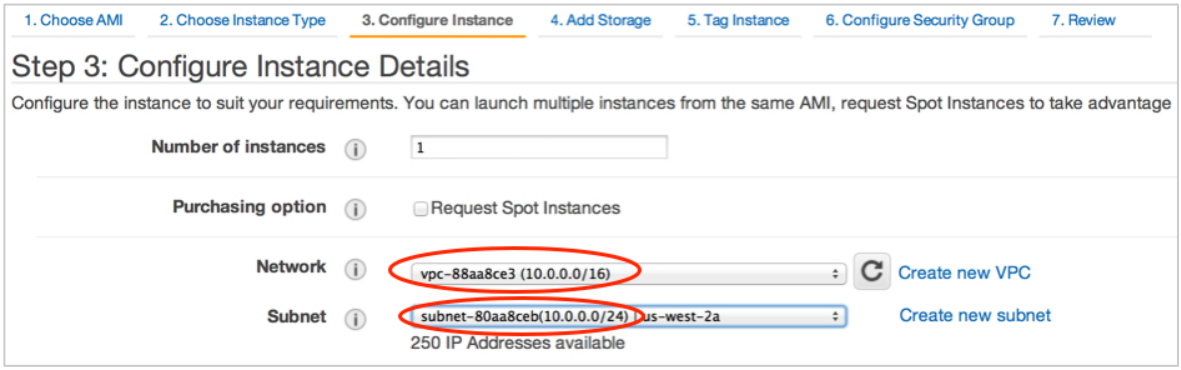


5) At the **Micro Instances** panel, click **Next: Configure Instance Details**. In this lab you use a micro instance type, which is preselected.



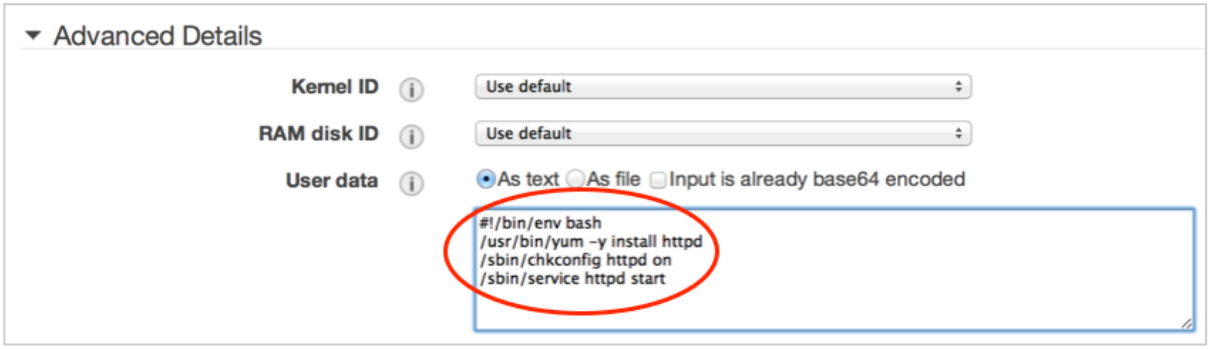
6) At the **Configure Instance Details** panel:

   1) For **Network**, choose the vpc you created (**10.0.0.0/16**).
   2) For **Subnet**, choose the public subnet (**10.0.0.0/24**).



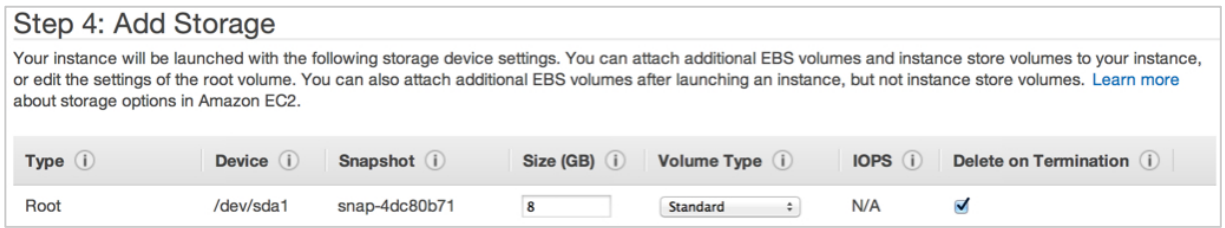
   3) Expand the **Advanced Details** section and type the following text in the **User Data** field. For your convenience, a command reference text file is attached to the qwikLAB page for this lab. You may edit the commands in the file and copy/paste them into your SSH client rather than typing them manually.

```
#!/bin//bash
/usr/bin/yum -y install httpd
/sbin/chkconfig httpd on
/sbin/service httpd start
```

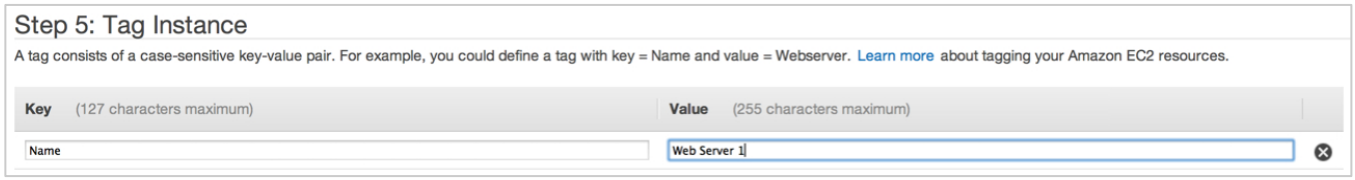


   4) Click **Next: Add Storage**.

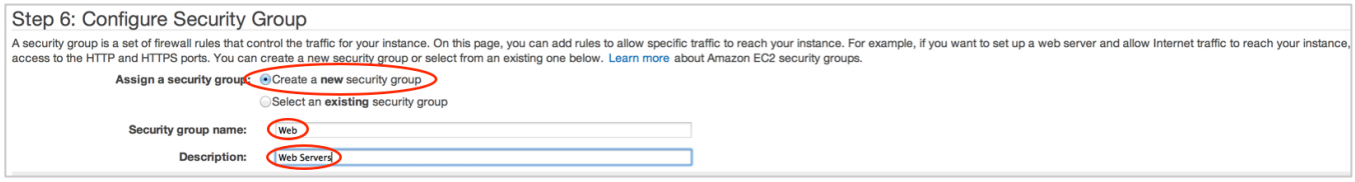
7) There are no modifications needed in the Add Storage panel. Accept the default values and click **Next: Tag Instance**.



8) At the **Tag Instance** panel, for **Name**, in the **Value** column type **Web Server 1**.

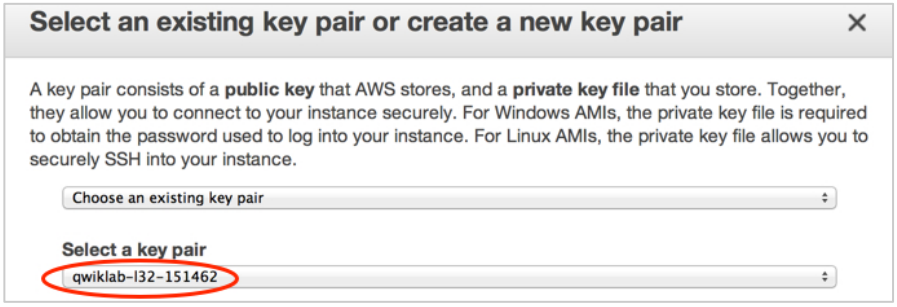


9) Click **Next: Configure Security Group**.
10) AWS recommends that you create a custom security group, based on role, instead of selecting an existing Security Group. At the Configure Security Group panel:
    1) Leave **Create a new security group** selected.
    2) In the **Security group name** field, type **Web** and (optionally) add a **Description**.

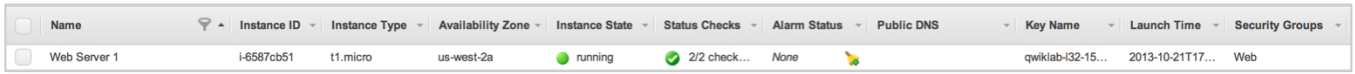


    3) Click **Add Rule**.
    4) For **Protocol**, choose **SSH**.
    5) For **Source**, accept the default values (**Anywhere** and **0.0.0.0/0**).
    6) Repeat the previous stepd to add rules for **HTTP** port **80** and **HTTPS** port **443**.

    7) Click **Review and Launch**.
11) At the **Review Instance Launch** panel, examine your options and click **Launch**.
12) You are presented with the **Select an existing key pair or create a new key pair** dialog.
13) Verify that the **Select a key pair** field is set to the **qwikLAB** key pair created for you, check the acknowledgement box and click **Launch Instances**.



14) Click **View Instances**. The instance is in the 'pending' state initially, followed by the 'running' state.

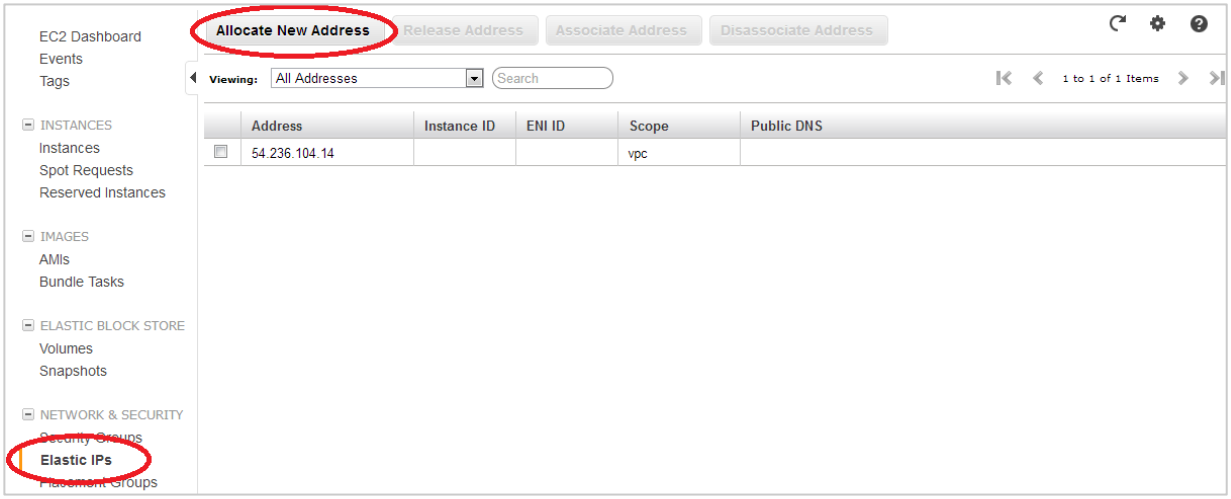


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Key Name	Launch Time	Security Groups
Web Server 1	i-6587cb51	t1.micro	us-west-2a	running	2/2 check...	None		qwiklab-l32-15...	2013-10-21T17...	Web

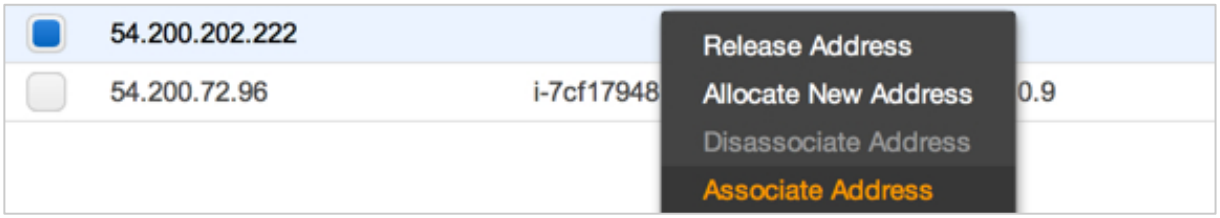
Create and Assign an Elastic IP Address

By default instances in the VPC do not have a public IP address. Because this Web server is meant to be public, you must allocate an Elastic IP address (EIP) and associate it with the server.

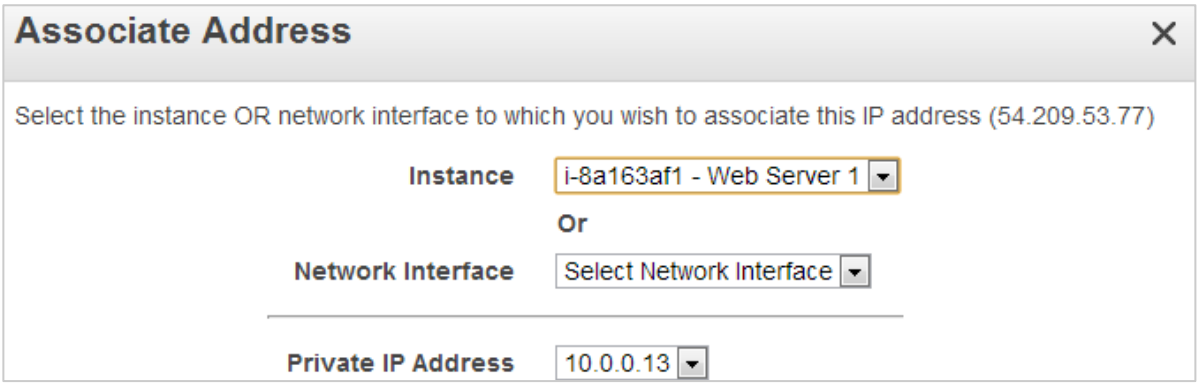
1) In the **EC2 Dashboard**, in the **Network & Security** section, click **Elastic IPs**.
2) Click **Allocate New Address**.



3) Click **Yes, Allocate**.
4) Right-click the new address and choose **Associate Address**.



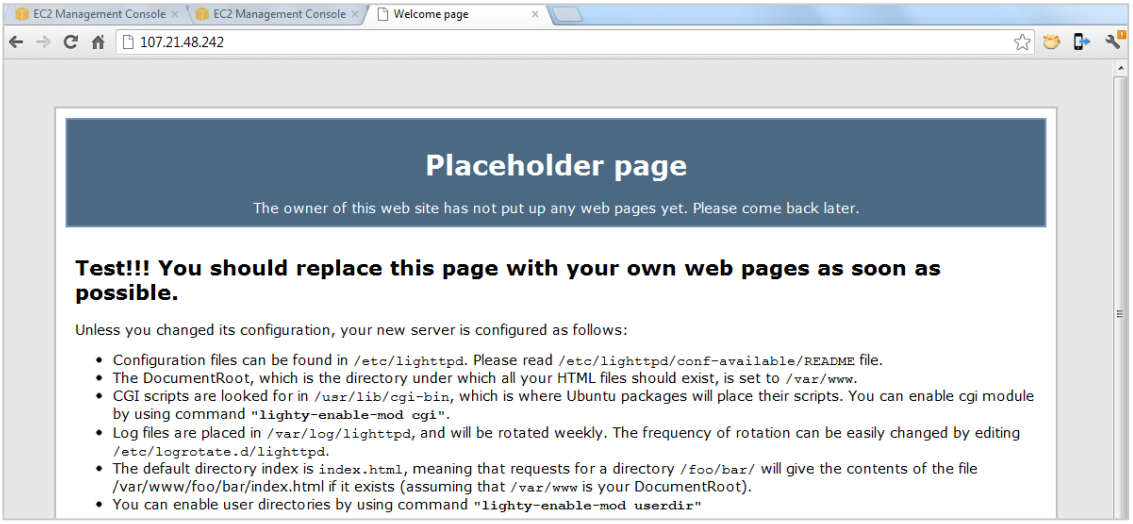
5) In the **Associate Address** dialog, for **Instance**, select your **Web Server 1** Instance. Be certain that you select the appropriate server, because the following options are unlikely to retain your context.



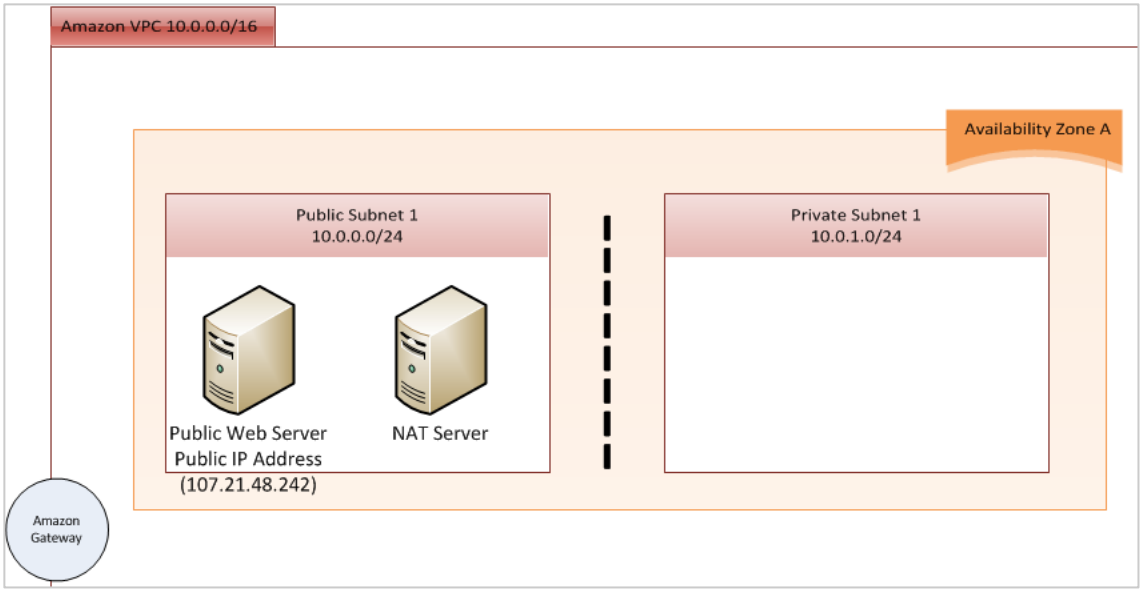
6) Accept the remaining default values and click **Associate**.

Note the options on the Elastic IPs page. Later in the lab exercise you are asked to associate another IP address, but next time, screen captures are not provided.

7) Connect to the Web server by typing the elastic IP address in your browser's address bar. You should see a page similar to the one below. Since the focus of this exercise is networking, you will not modify the Web site.



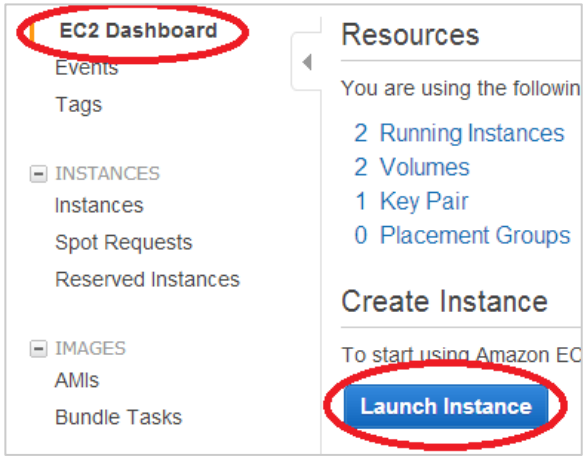
The diagram below shows what you have configured in the exercise thus far:



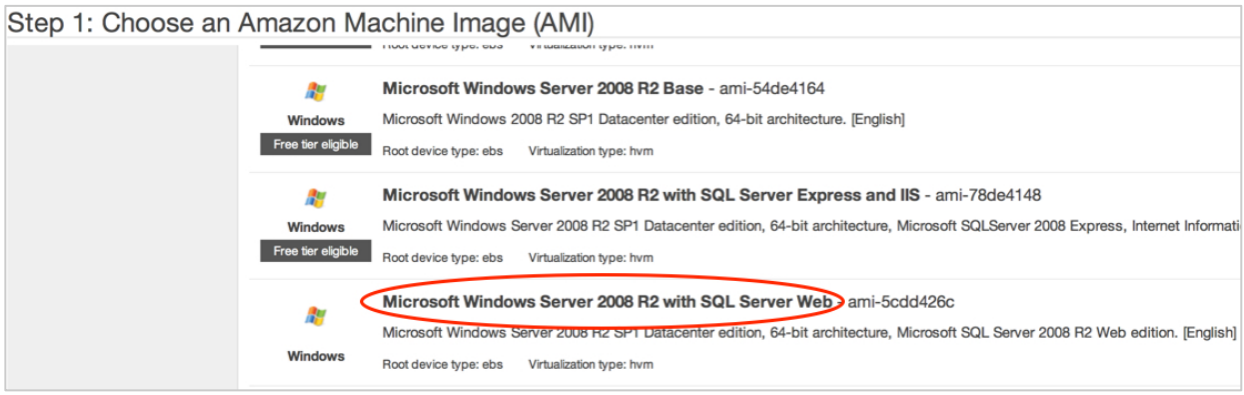
Launch a Back-End Microsoft SQL Server

Database security is a serious subject. You will place your database in a private subnet, away from Internet traffic. You do not use the database in this lab. Rather, the objective is to create a "pot of gold" where the server is reachable via RDP under a limited set of conditions.

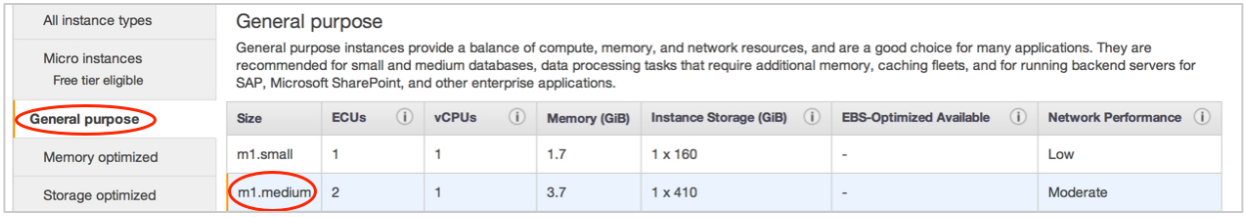
1) Click **EC2 Dashboard**, click **Launch Instance**.



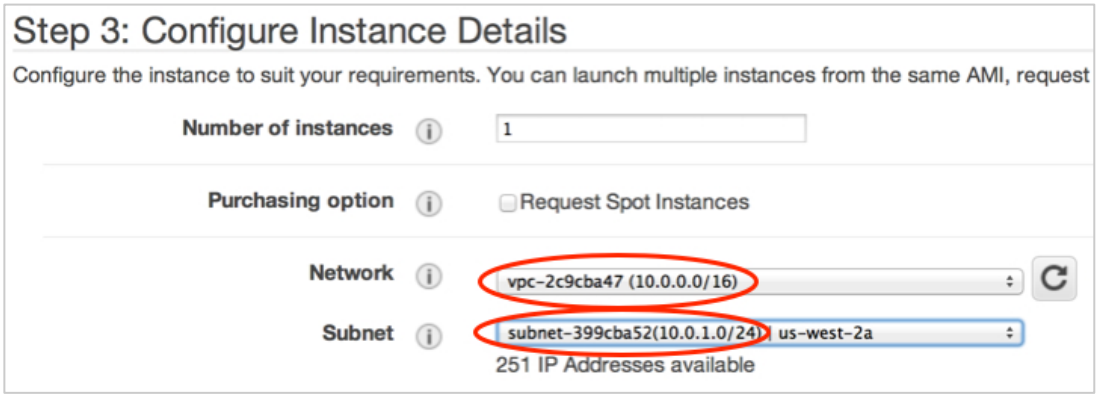
2) Choose the **Quick Start** wizard.
3) Click **Select** to choose the **Microsoft Windows Server 2008 R2 with SQL Server Web** AMI.



4) At the Micro Instances panel, click **General Purpose > m1.medium**.

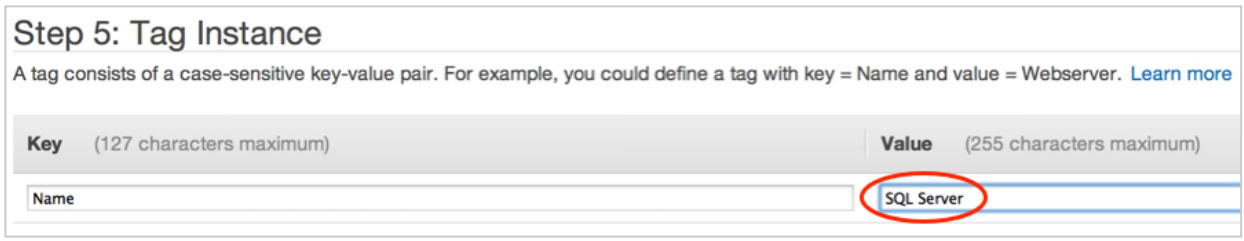


5) Click **Next: Configure Instance Details**.
6) At the **Configure Instance Details** panel:
   1) For **Network**, choose the vpc **10.0.0.0/16**.
   2) For **Subnet**, choose the private subnet, **10.0.1.0/24**.



3) Expand **Network interfaces**, and in the **Primary IP** field for the **eth0** device, type the following IP address: **10.0.1.99**.

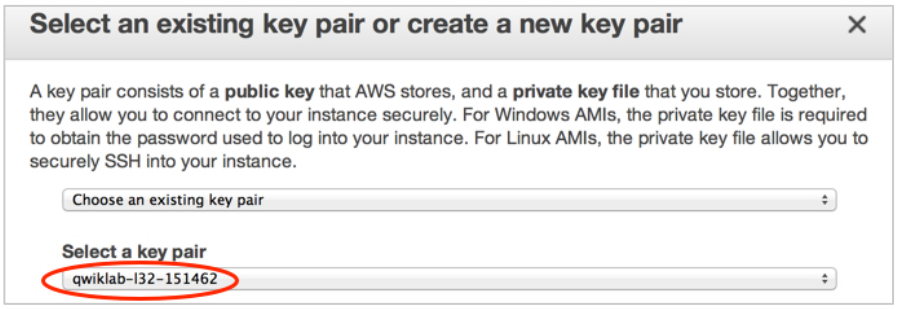
7) Click **Next: Add Storage**.
8) At the **Add Storage** panel, accept the default values and click **Next: Tag Instance**.
9) At the **Tag Instance** panel, for **Name**, type **SQL Server** in the **Value** column.



10) Click **Next: Configure Security Group**.
11) At the **Configure Security Group** panel:
    1) Click **Create a new security group**.
    2) For **Security group name**, type **SQL Server** and (optionall) add a **Description**.
    3) Verify there are existing rules for ports **3389** and **1433**.

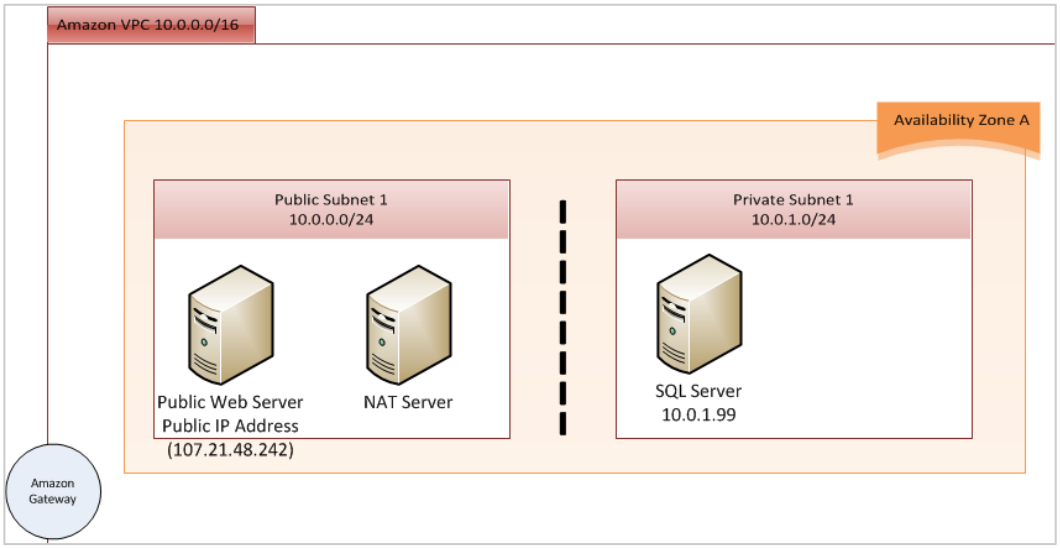
    Note: The source IP address range for the rules is set to 0.0.0.0/0, meaning "from anywhere". In fact, the routing restrictions translate this meaning into "from any host, as long as it is on one of the VPC subnets." You tighten this rule when the bastion instance is created.

12) Click **Review and Launch**.
13) Click **Launch** and in the **Select an existing key pair or create a new key pair** dialog:
    1) Verify that the **Select a key pair** field is set to the **qwikLAB** key pair created for you.
    2) Check the acknowledgement box.
    3) Click **Launch Instances**.



14) Click **View Instances**. The instance is in the 'pending' state initially, followed by the 'running' state.

Your network should look like the following diagram. It is not production ready because the database server is not set up to serve the Web server, and a secure way to connect to and administer the SQL Server is needed. The NAT will act as a router that allows the SQL Server to make outbound calls to the Internet in order to download Windows Updates, and so on.



There is one other very important item missing from the environment: a second Availability Zone with another Web server and a second database server. AWS provides you access to multiple Availability Zones at no

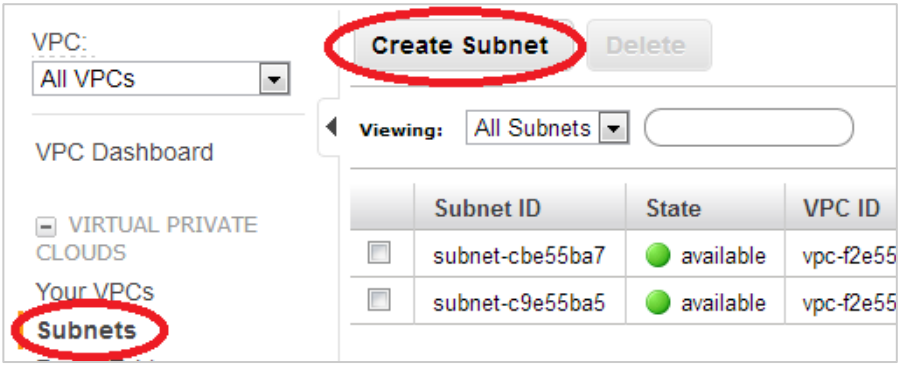
additional cost to you. A best practice is to mirror servers across two zones, and then use load balancing and other techniques in order to distribute traffic between them.

AWS considers multi-AZ deployments essential to your welfare. Our data centers are more reliable than typical Enterprise data centers, but outages can happen. If your environment is in a single AZ, you have no SLA protection. **The EC2 SLA is activated only if two or more Availbility Zones in an AWS Region go offline at the same time.**

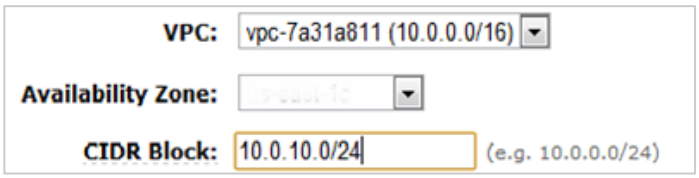
Manually Create Two More Subnets

You need to create a public subnet, and also a private subnet in another Availability Zone. Unlike the previous subnets, you create these without the assistance of a wizard. Along the way you learn more about how they operate. These will be in the same Availability Zone as each other but in a different Availability Zone from the first two you created. To review, the original subnets were 10.0.0.0/24 (public), and 10.0.1.0/24 (private). Both were in the same Availability Zone.

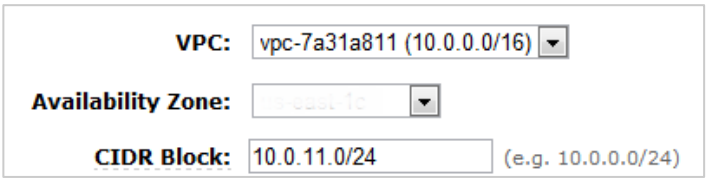
1) From the toolbar, choose **Services > All AWS Services > VPC** to open the **VPC Dashboard**.
2) In the **Virtual Private Clouds** section, click **Subnets** and then click **Create Subnet**.



3) Using your knowledge from previous steps:
   1) Create a new public subnet with **CIDR Block 10.0.10.0/24**.
   2) For **Availability Zone**, choose a different zone than the one used for the previous public subnet. **Note**: The screen capture below is left intentionally blank as your AZ may differ.



4) Click **Yes, Create**.
5) Repeat the steps to create a new private subnet using **CIDR Block 10.0.11.0/24**. Be sure to place it in the same Availability Zone as the public subnet. Note: The screen capture following is left intentionally blank as your AZ may differ.



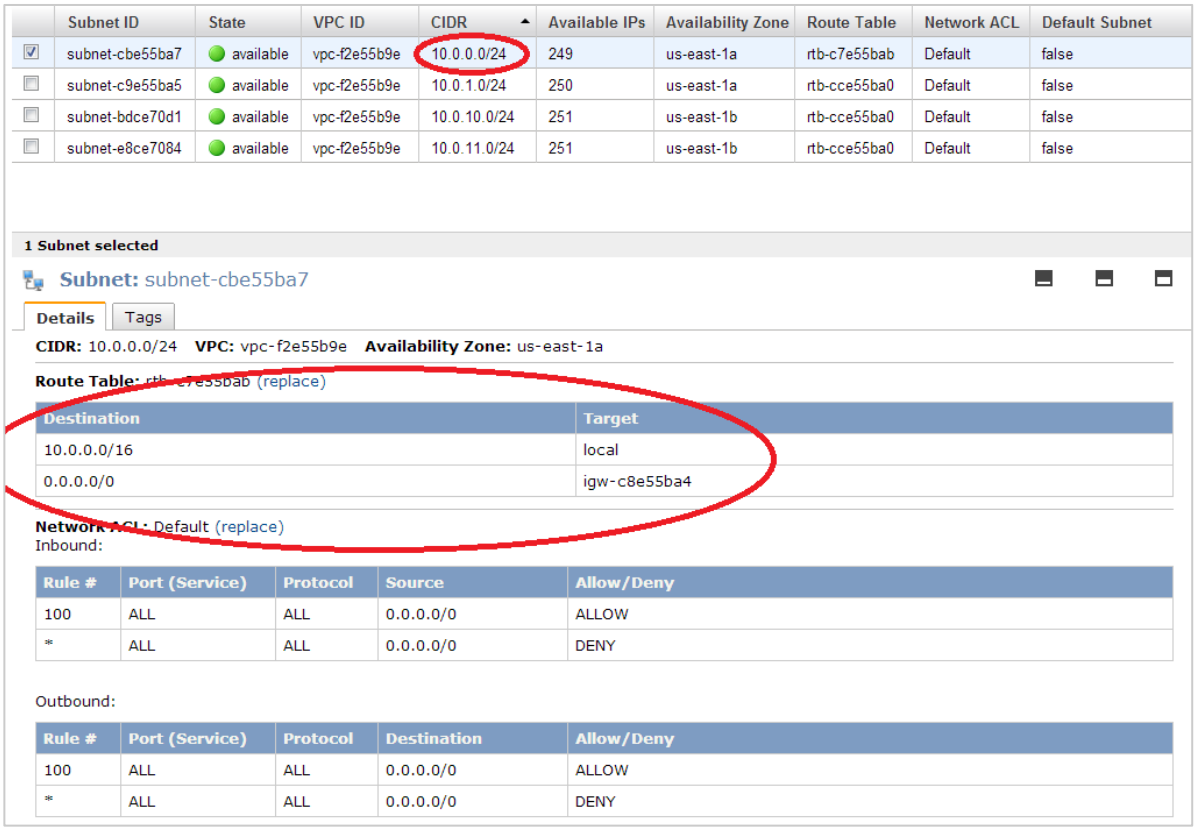
What Determines Whether a Subnet is Public or Private?

Now we have two more subnets, but what makes them private or public? It is the routing rules.

1) Select the subnet using CIDR **10.0.0.0/24**, and note that there are two routing rules in the Route Table (on the **Details** tab below):

- Any machine in this subnet can communicate with any other machine in 10.0.0.0/16, which is the entire VPC. In other words, communication between all subnets is wide open. Later in this lab you examine security groups as a mechanism to restrict traffic.
- Any traffic to/from the Internet (0.0.0.0/0) will be routed thru the Internet Gateway device. You have not looked at that device so far, but think of it as a router on the edge of the VPC. In fact, that is how it is depicted in the network diagrams.

2) Scroll down and you will see some Network ACLs, which in theory could also control traffic. However, the VPC supports a limited number of rules so we will use alternate controls that are even more granular.



Subnet ID	State	VPC ID	CIDR	Available IPs	Availability Zone	Route Table	Network ACL	Default Subnet
subnet-cbe55ba7	available	vpc-f2e55b9e	10.0.0.0/24	249	us-east-1a	rtb-c7e55bab	Default	false
subnet-c9e55ba5	available	vpc-f2e55b9e	10.0.1.0/24	250	us-east-1a	rtb-cce55ba0	Default	false
subnet-bdce70d1	available	vpc-f2e55b9e	10.0.10.0/24	251	us-east-1b	rtb-cce55ba0	Default	false
subnet-e8ce7084	available	vpc-f2e55b9e	10.0.11.0/24	251	us-east-1b	rtb-cce55ba0	Default	false

1 Subnet selected

Subnet: subnet-cbe55ba7

Details | Tags

CIDR: 10.0.0.0/24 **VPC:** vpc-f2e55b9e **Availability Zone:** us-east-1a

Route Table: rtb-c7e55bab (replace)

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-c8e55ba4

Network ACL: Default (replace)

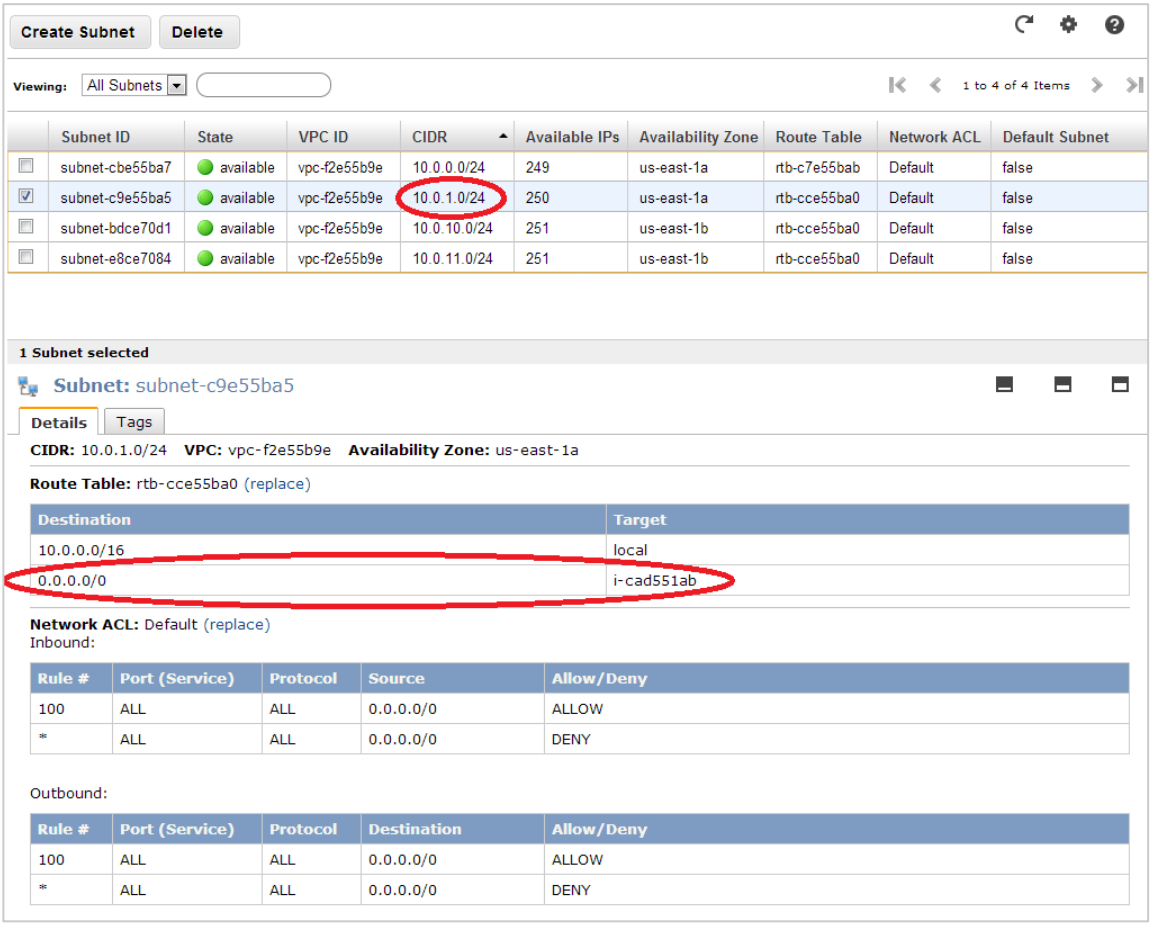
Inbound:

Rule #	Port (Service)	Protocol	Source	Allow/Deny
100	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL	ALL	0.0.0.0/0	DENY

Outbound:

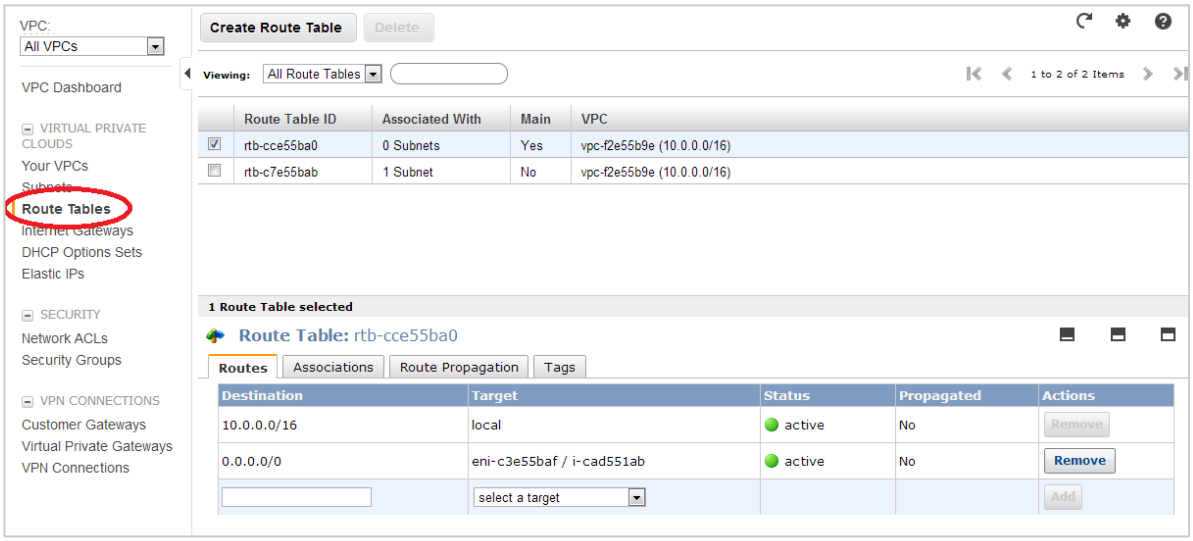
Rule #	Port (Service)	Protocol	Destination	Allow/Deny
100	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL	ALL	0.0.0.0/0	DENY

3) Select the **10.0.1.0/24** subnet. Similarly, 10.0.1.0/24 has routing rules:
   - Traffic bound for any other subnet in the VPC (10.0.0.0/16) is unrestricted.
   - Traffic destined for the Internet will flow to the EC2 instance, which is the NAT instance. Note that the NAT will not route random requests from the Internet back into this subnet. It will only route replies made in response to outbound requests from inside this subnet.

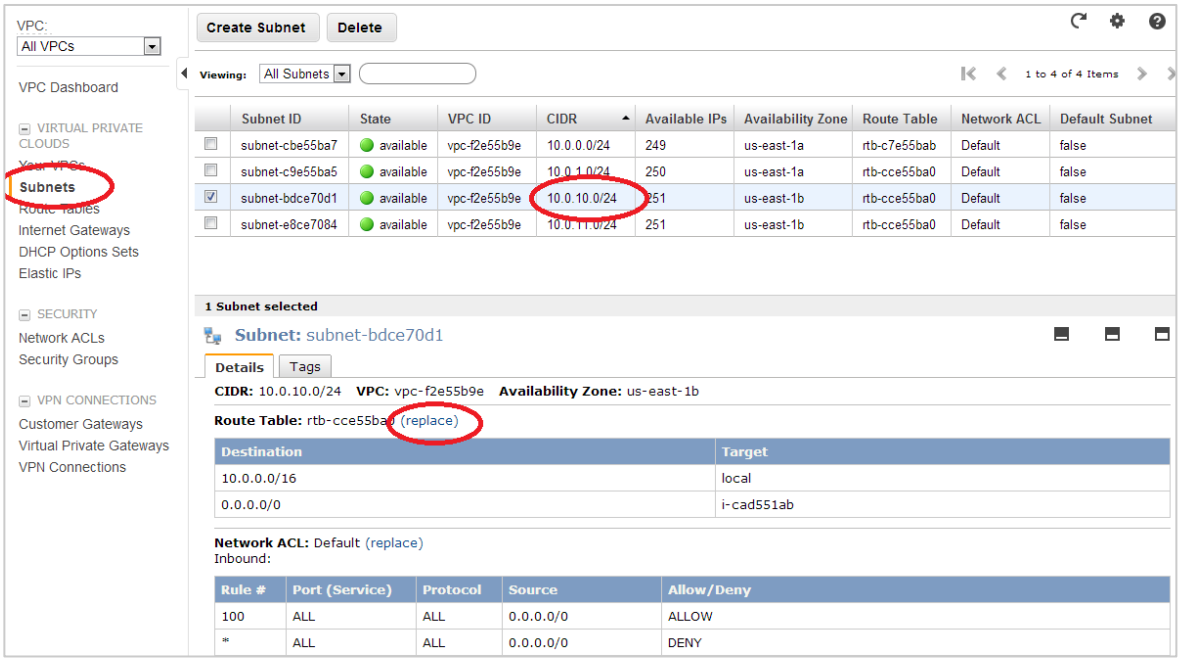


4) In the **Virtual Private Clouds** section, click **Route Tables** and look at this from the other side. According to this view, only 1 subnet is associated with any routing rule, but there a total of 4 subnets! Why?

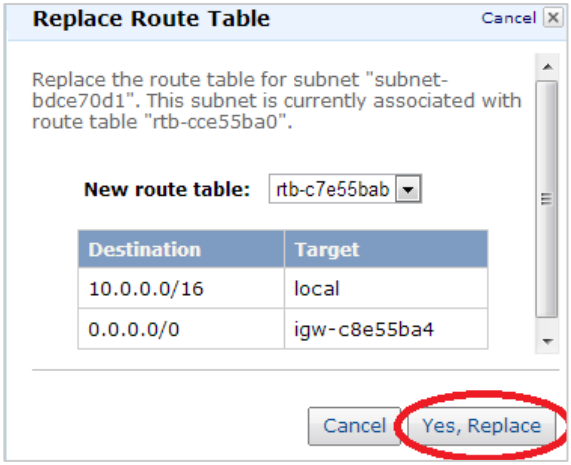
The Amazon VPC operates on a “safety first” principle. Note that one of the rule sets is marked “main.” If a subnet is not explicitly associated with a routing ruleset, it uses the Main ruleset, which happens to be the ruleset that does not talk to the Internet. **So by default, no subnet is able to communicate with the Internet** (unless you change the default behavior).



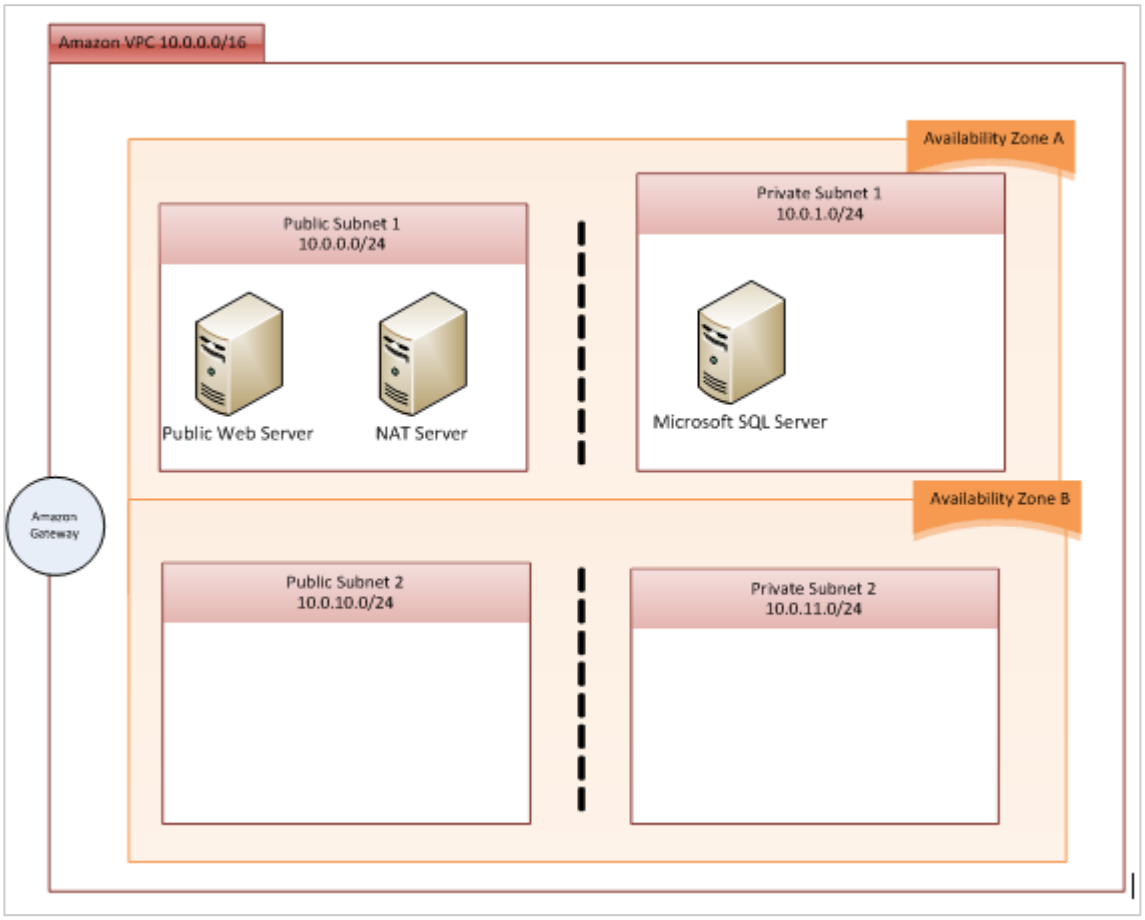
5) You need to associate the new public subnet (10.0.10.0/24) with the routing ruleset that routes bi-directionally to the Internet. Click **Subnets** and Select the **10.0.10.0/24** subnet.
6) On the **Details** tab, to the right of **Route Table**, click **replace** to replace the ruleset.



7) There is only one choice in the drop-down list because the console is smart enough to know that you cannot replace the current routing rules with…..the current routing rules. Note the new value in the **Target** column for **0.0.0.0/0** and click **Yes, Replace**.



Your VPC should now look like the following:

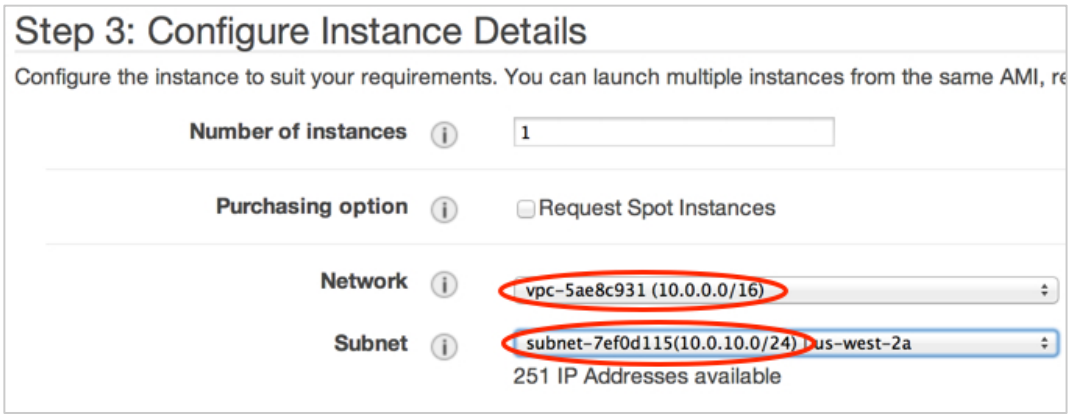


Launch a Bastion Windows Host

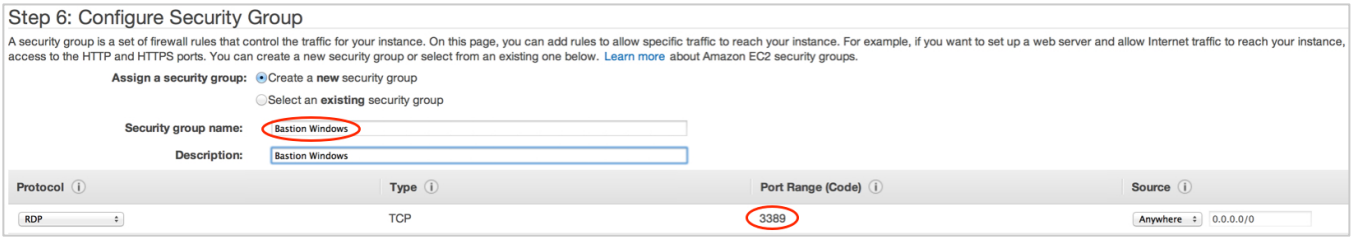
A bastion host is a computer that is configured to prevent unauthorized network access. The bastion host is typically in front of a firewall or in a corporate DMZ. The bastion host usually runs a very limited set of services (such as a proxy server) so there are fewer network entry points that can be exploited.

You will create your bastion host in your new public subnet, though the original public subnet would also work.

1) Click **VPC Dashboard** and then click the **Launch EC2 Instances** button.
2) Click **Launch Instance** to open the new instance wizard and choose **Quick Start**.
3) Choose the **Windows Server 2008 R2 Base** AMI.
4) At the **Micro Instances** panel, click **General Purpose**, select **m1.small** then click **Next**.
5) At the **Configure Instance Details** panel, for Network choose **10.0.0.0/16**, and for Subnet, choose **10.0.10.0/24**.



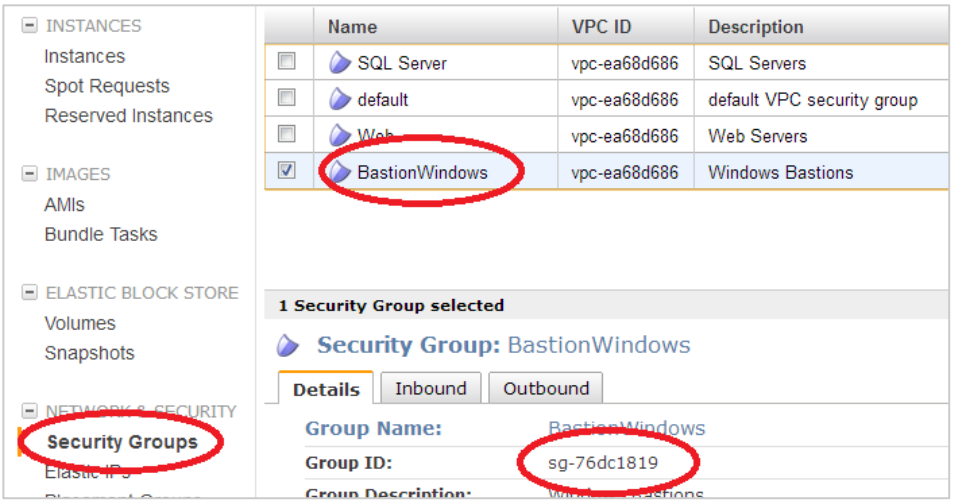
6) Click **Next** twice.
7) At the **Tag Instance** panel, for **Name**, type **Bastion Windows Host** in the **Value** column.
8) Click **Next**.
9) At the **Configure Security Group** panel, create a new security group, named **Bastion Windows**. We are only allowing access to port 3389, which is the Windows Remote Desktop Protocol (RDP). For this lab we are allowing access from any IP address on the Internet. In real life you will want to restrict access to the address ranges required for administration.



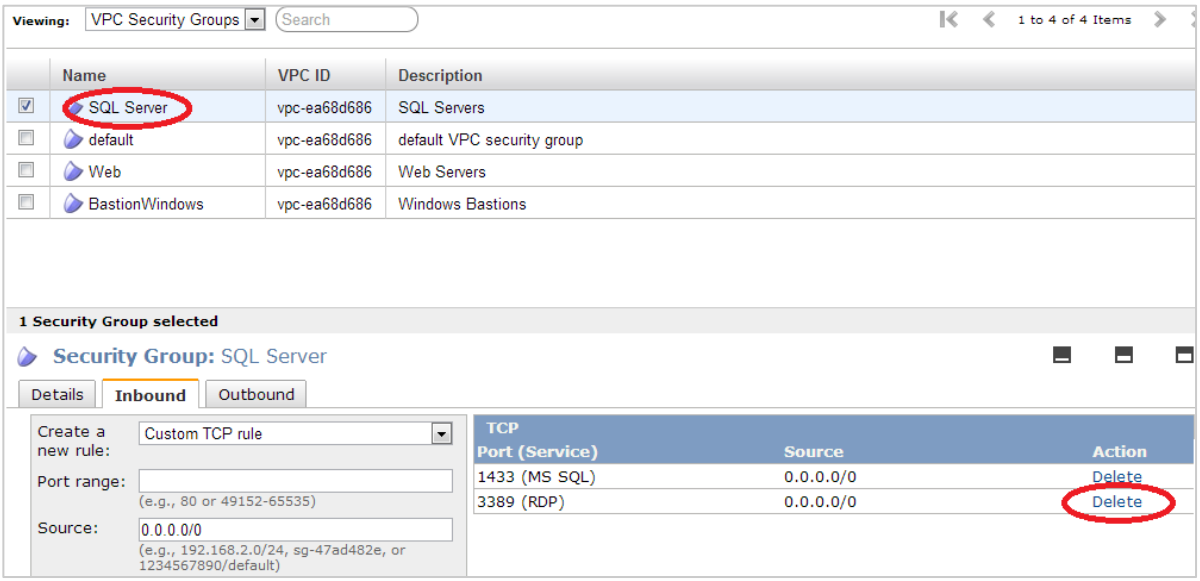
10) Click **Review and Launch**.
11) Click **Launch**.
12) Choose the **qwikLAB** key pair, check the acknowledgement and click **Launch Instances**.
13) Click **View Instances**.

Now that you have a security group for the bastion server, change the rules for the database server so the only traffic it accepts is from the bastion security group.

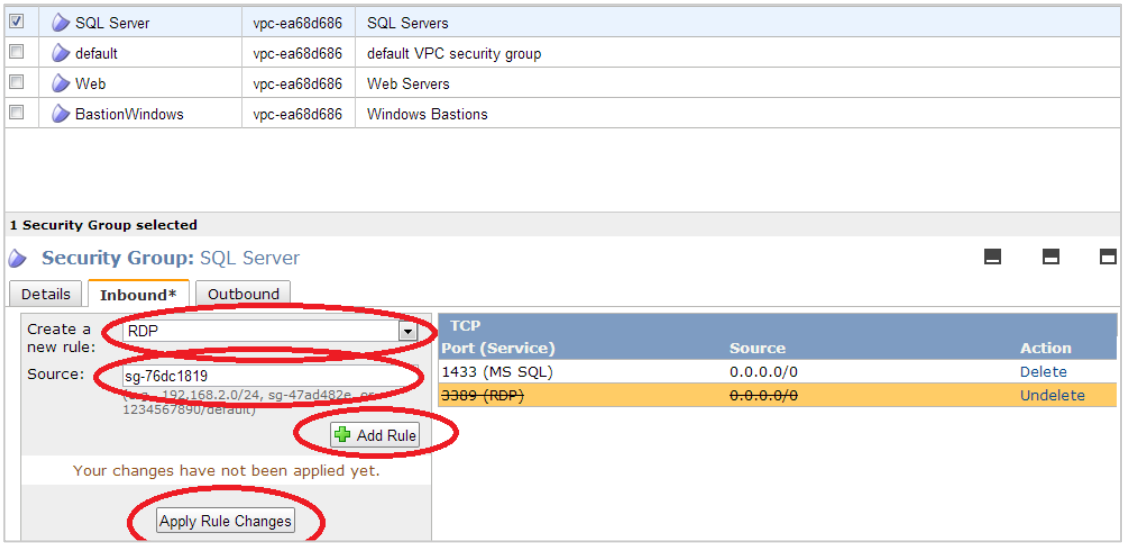
1) In the **EC2 Dashboard**, in the **Network & Security** section, click **Security Groups**.
2) Select the **Bastion Windows** security group and view the **Details** tab below.
3) Make a note of the Bastion Windows security group's **Group ID** (you will need it in a moment). You can paste it into a notepad document or copy it to the clipboard.



4) Select the **SQL Server** security group, switch to the **Inbound** tab below, and click **Delete** to remove the existing rule for port **3389 (RDP)**.



5) Create a new **3389 (RDP)** rule that is restricted to the **BastionWindows** security group. Use the Security Group ID you pasted into notepad or copied to the clipboard as the **Source** then click **Add Rule**. This rule illustrates another, powerful, way to use security groups.
6) Apply your changes by clicking **Apply Rule Changes**.

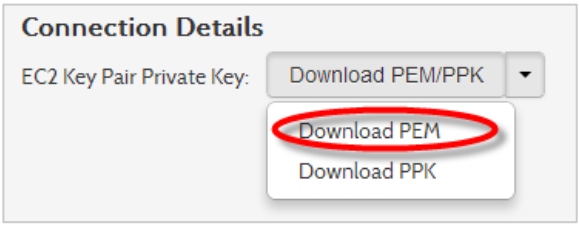


7) In order to use the bastion server, you will need a public IP address. Once assigned, the address will appear as part of the details for the bastion host. In the **Network & Security** section, click **Elastic IPs**.
8) Allocate a new IP address.
9) Associate the IP address with the **Bastion Windows Host** instance.

Note the public IP address that you allocated and associated is the one you will use to connect to the bastion host via RDP in the next section.

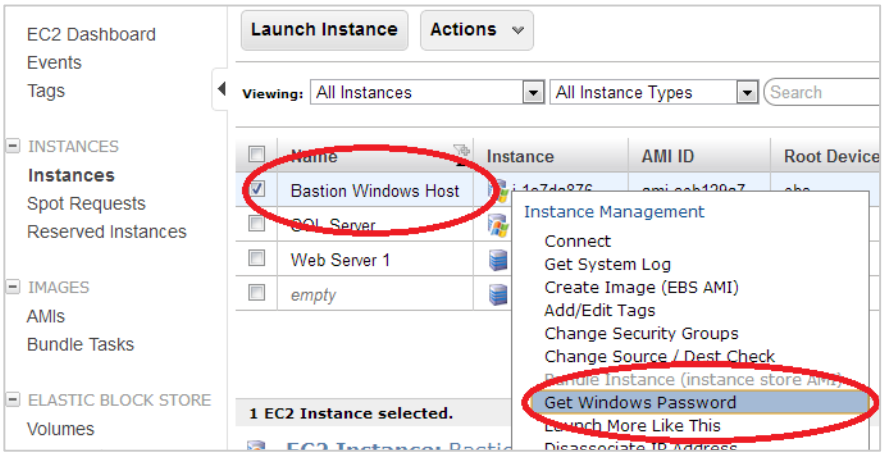
Retrieve your windows Password

1) In your browser, switch to the ***qwikLAB*** tab.
2) For **EC2 Key Pair Private Key**, choose **Download PEM** from the drop-down list. This downloads the *qwikLAB*™ provided EC2 Key Pair private key file in PEM format.

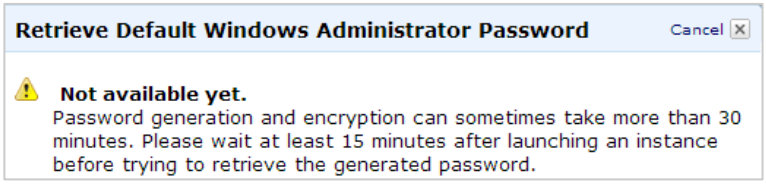


3) Save the file to your computer's **`\Downloads`** folder (the default) or move it to the folder or directory of your choice.
4) Switch to the **EC2 Management Console** tab.
5) In the **Instances** section, click the **Instances** link.

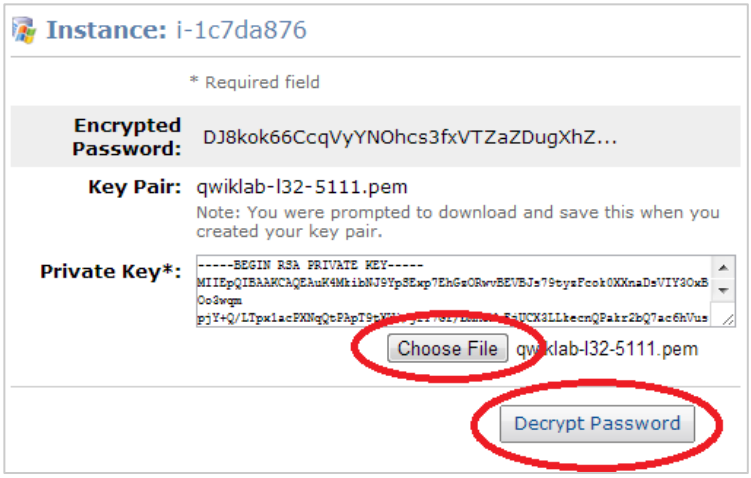
6) Right-click the **Bastion Windows Host** instance and choose **Get Windows Password**.



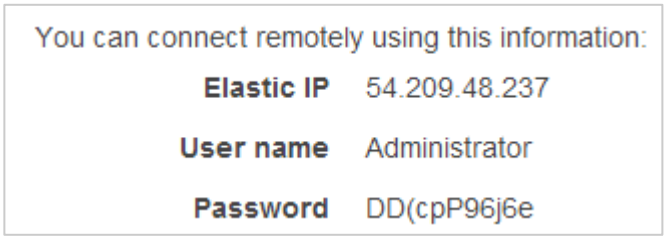
7) If the instance is still starting, you may receive a "not available yet" message. Click **Close**, wait a moment and choose **Get Windows Password** again.



8) Click **Choose File** and navigate to your **`\Downloads`** folder (or your download directory/folder) and select the EC2 Key Pair private key (`.pem`) file that you downloaded from *qwikLAB*.
9) Click **Decrypt Password**.



10) Make a note of the **Elastic IP**, **User name**, and **Password**. Since you need these items later, consider pasting them into a text file. (Your information will differ from the screen capture below).

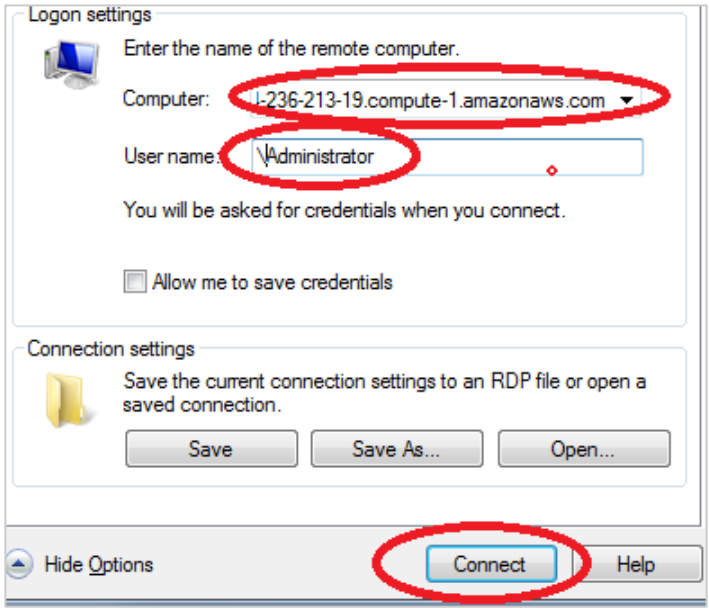


Connect to the Bastion Server (Windows)

Important: Proceed to one of the later sections entitled "Connect to the bastion server (OS X)" or "Connect to the bastion server (Linux)" if you are using a Linux-based laptop to perform this exercise. This section is *only* for Microsoft Windows users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

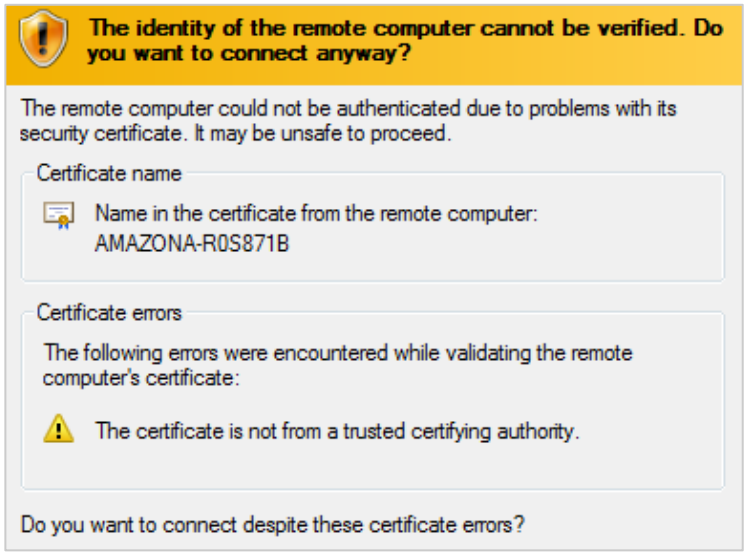
1) On your local computer click **Start > Run**, type **MSTSC** and click **OK** to start the local RDP client.
2) Click **Show Options**, type or paste the **Computer** name and **User name** you noted, and then click **Connect**.

Note: You are signing in as another user - **Administrator**, and may need to specify the user name as "\Administrator" (with a leading backslash) in order to differentiate from the Administrator user on your local computer.



3) When prompted, type the **Password** you noted.

4) Click **Yes** if you see a certificate verification message similar to “the identity of the remote computer cannot be verified:”

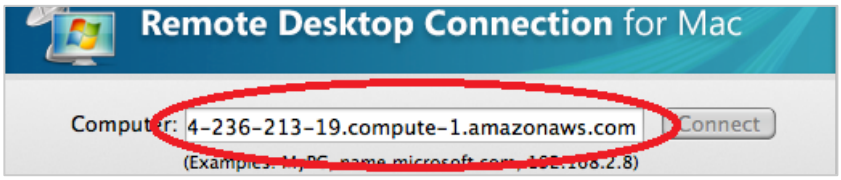


5) Proceed to the section entitled “Log in to the database server.”

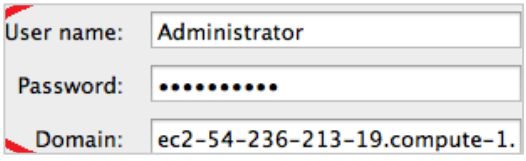
Connect to the Bastion Server (OS X)

Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled “Log in to the database server.” This section is for OS X users *only*. If you are using a different Linux-based operating system, please proceed to the section entitled “Connect to the bastion server (Linux).” There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

1) Open the Remote Desktop Connection for Mac application.
2) Type the bastion computer’s host name you noted in the **Computer** field and click **Connect**.



3) When prompted, type the **User name** and **Password** you noted. The Domain will auto-populate with the EC2 Instance DNS and you can ignore it.

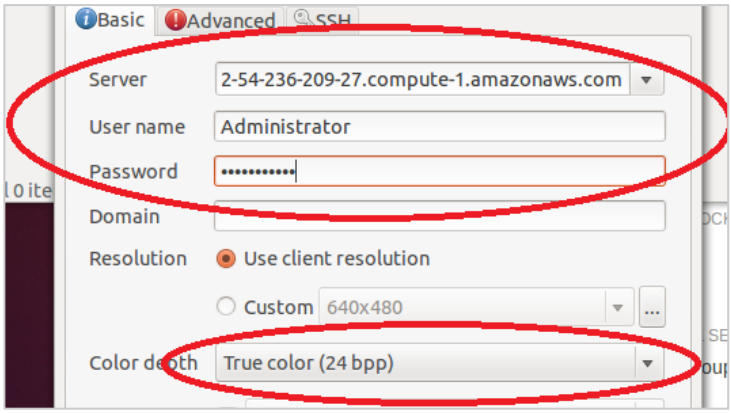


4) Click **OK**.
5) Click **Connect** if you see a verification message similar to “the server name is incorrect.”
6) Proceed to the section entitled “Log in to the database server.”

Connect to the Bastion Server (Linux)

Important: Only complete this section if you are using a non-Macintosh, Linux-based laptop to perform this exercise. This section is *only* for Linux users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

1) Open the Remmina Remote Desktop Client.
2) Type the bastion host computer's host name you noted in the **Server** field, and then type the **User name** and **Password**.
3) Optionally, choose a **Color depth** that your bandwidth supports (in this example 'True color (24 bpp)') for a nicer remote desktop environment.
4) Click **Connect**.

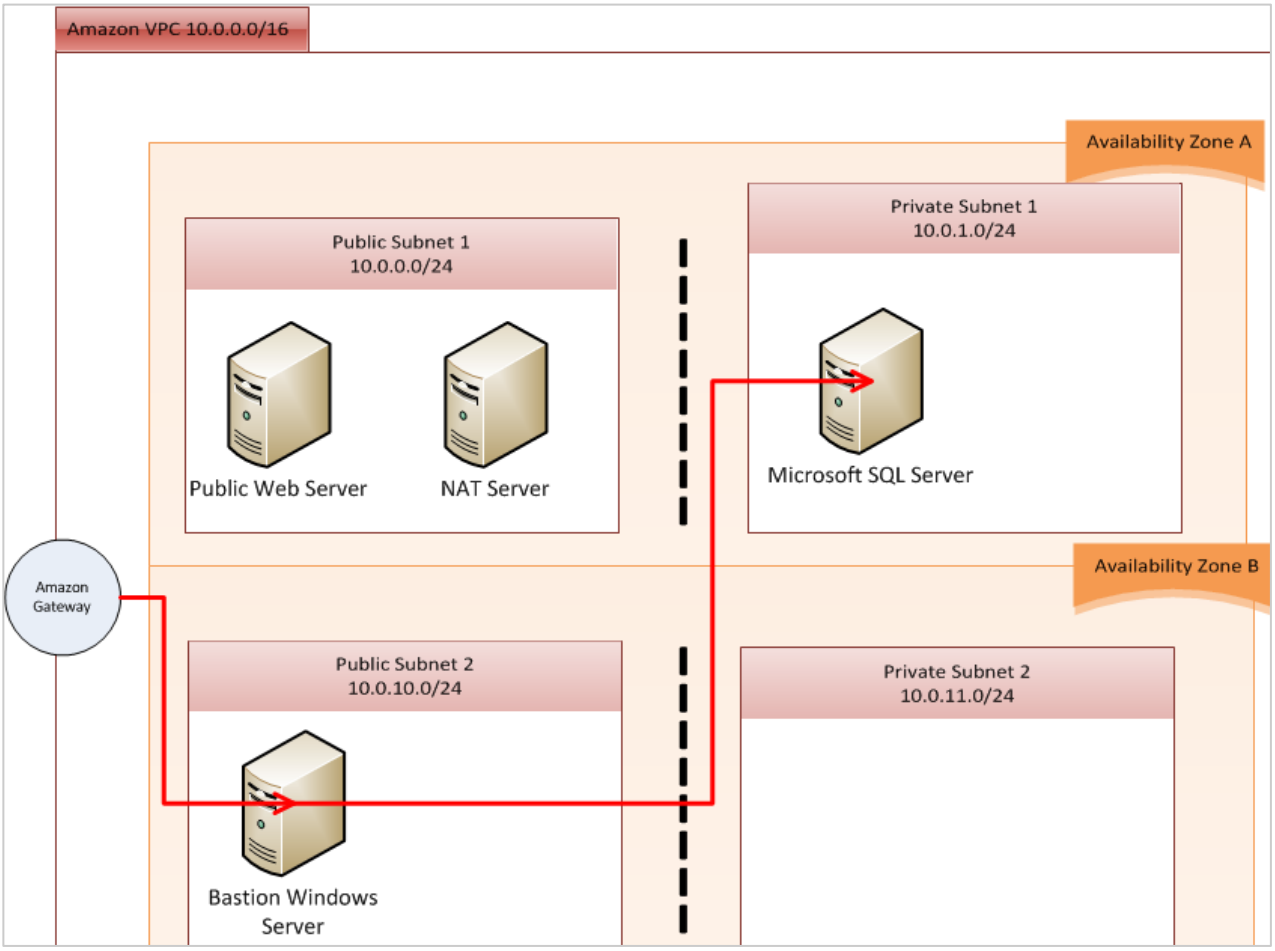


5) Click **OK** when prompted to accept the remote certificate.

Log in to the Database Server

1) Using previous steps as a guide, retrieve the password for the SQL Server instance.
2) Switch to your Remote Desktop Client session.
3) Since you are logged into the Windows bastion host, repeat the previous Windows connection steps to log into the SQL Server instance from inside the bastion host session. Use the "Connect to the bastion server (Windows)" steps even if you are running OS X or Linux locally. You are running the RDP client from inside the remote Windows bastion host.
4) In the Remote Desktop Client, in the **Computer** field, type **10.0.1.99**.

Your completed environment should look like the following. The line from the gateway device to the SQL Server illustrates traffic flow from the edge of the VPC network, through the bastion host, and to the SQL Server. You might wonder why you created the second private subnet (labeled Private Subnet 2 (10.0.11/0/24) subnet in the diagram). This subnet acts as a slave, replica SQL Server instance for the SQL Server in Private Subnet 1 (10.0.1.0/24).



Conclusion

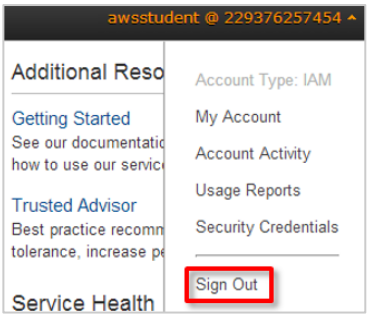
Amazon networking is secure by default, and as you just learned there are multiple ways to safely connect to servers that are kept in private subnets. In order to ensure that your network is secure, pay attention to the subnets containing your servers.

Bastion hosts and VPN tunnels each have an advantage. Bastion hosts allow you a secure method of logging in to manage servers, especially if only a few people need to perform this activity. If you want the VPC to act as a virtual extension to your corporate network, then a VPN may make more sense.

Finally, you learned that security group rules can be either very precise or quite loose. Make certain that your security groups are as restrictive as possible but not so restrictive that there are unintended side effects.

Ending the Lab

1. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).



2. Close any active SSH client sessions or remote desktop sessions.
3. Click the **End Lab** button on the *qwikLAB*™ lab details page.

4. When prompted for confirmation, click **OK**.
5. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following: 1 star = very dissatisfied, 2 stars = dissatisfied, 3 stars = neutral, 4 stars = satisfied, and 5 stars = very satisfied. Also, you may close the dialog if you do not wish to provide feedback.

