**Create a Custom Amazon VPC with Public and Private Subnets**

Created by[Neal Davis | AWS Certified Solutions Architect & Developer](https://deloittedevelopment.udemy.com/user/63f4a578-c67a-456b-916c-ddadf73e9a26/)

**AWS Workspace**

**60-90 minutes**

In this lab, you will be a Cloud Practitioner at a company that is preparing to launch applications in the AWS Cloud. Your boss has asked you to set up a custom Amazon Virtual Private Cloud (VPC) with public and private subnets into which applications running on Amazon EC2 instances will be launched.

The IT Manager at our company has given approval to start configuring the AWS Cloud in preparation for the migration of workloads from our on-premises environment. Id like you to set up a custom Amazon VPC with public and private subnets. This will provide a platform for testing some applications in the cloud.

To deploy this solution you will need to create an Amazon VPC including IPv4 CIDR block, public and private subnets, Internet Gateway, and route tables. You will need to verify network connectivity is functioning correctly within the VPC using Amazon EC2 instances in public and private subnets.

**How you'll work**

Your project has been broken into a set of tasks. To complete these tasks, use the provided workspace. You can launch your workspace by clicking below or using the button in the top right of the screen.

Tasks

1-Create VPC and subnets

2-Configure VPC subnets

3-Attach an internet gateway and configure route table

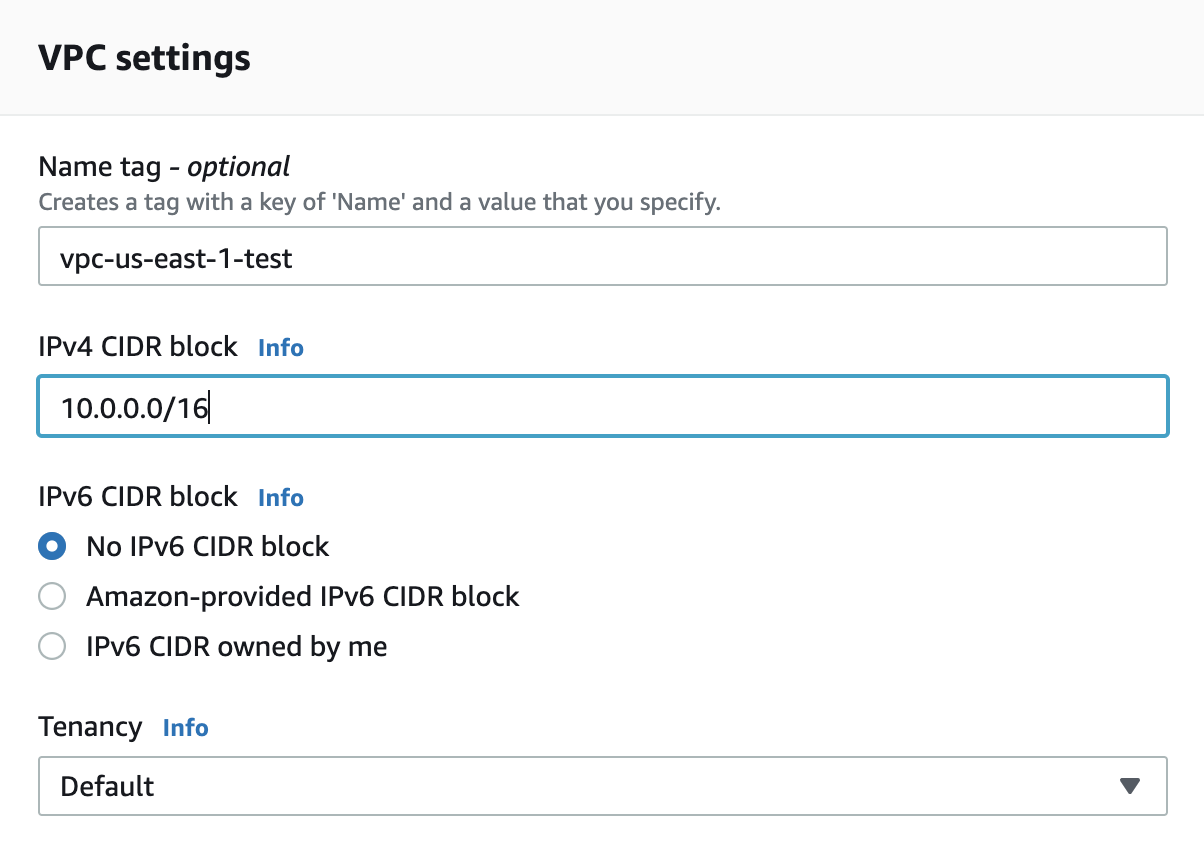
4-Deploy EC2 and test connectivity

5-Clean up

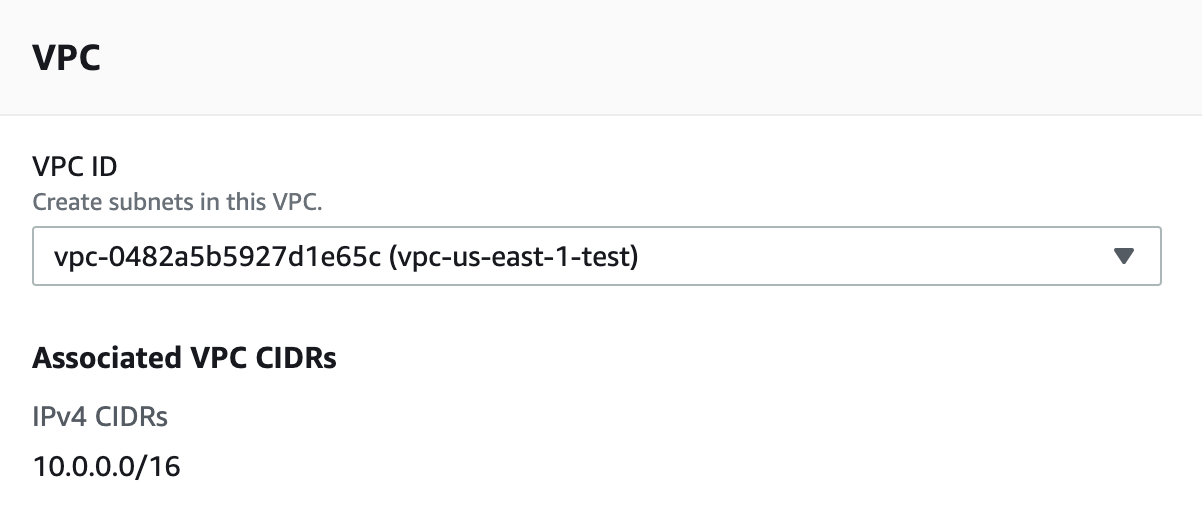
Create VPC and subnets

You will need to create an Amazon VPC with two public subnets and two private subnets across multiple Availability Zones. Use the 10.0.0.0/16 CIDR block for the VPC and use /24 blocks for the subnets. Use the name tags to make it easy to identify the type of subnet and Availability Zone.

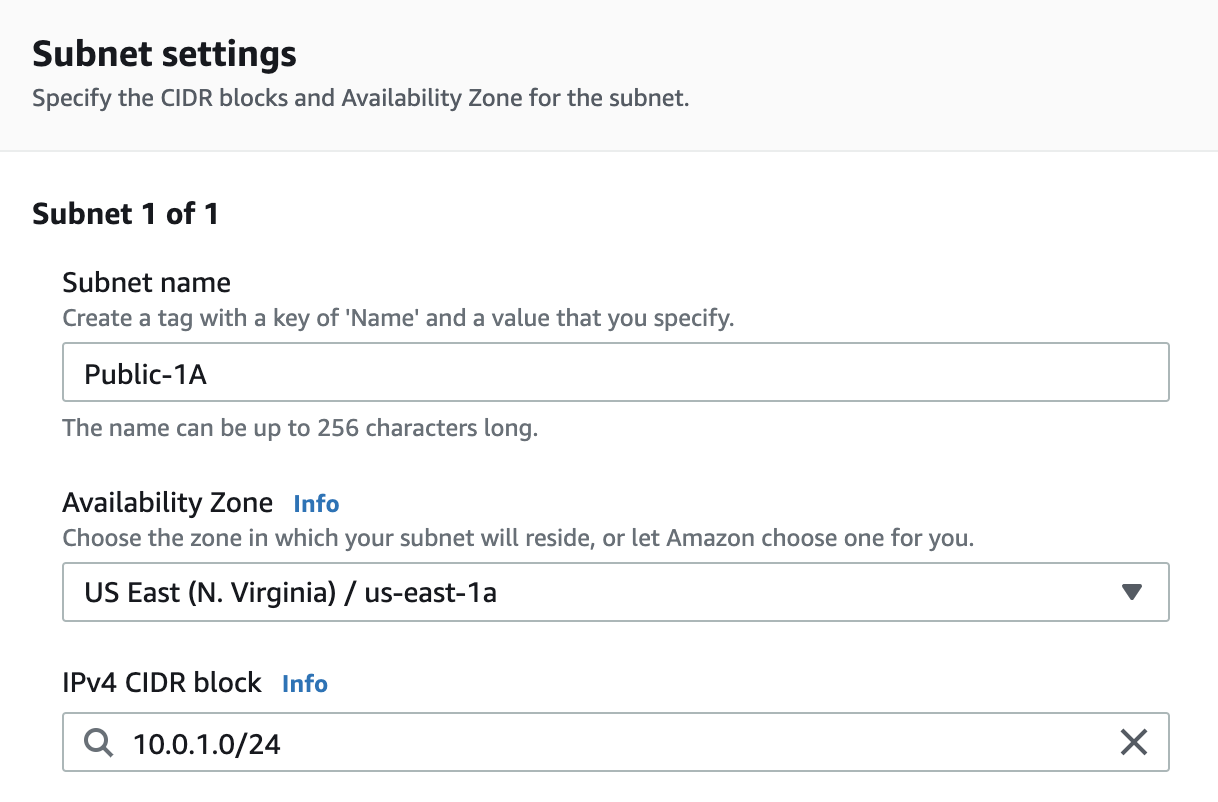
1. In the **AWS Management Console**, go to **Services** and click **VPC.**
2. Select **Your VPCs** in the left-hand navigation and click **Create VPC.**
3. For the name tag enter vpc-us-east-1-test. This name includes the AWS Region code to easily identify the Region the VPC is located in (change as appropriate.)
4. For the IPv4 CIDR block enter 10.0.0.0/16.
5. The VPC settings should now appear similar to the image below:



1. Click **Create VPC** to create the custom VPC.
2. Select **Subnets** in the left-hand navigation and click **Create Subnet.**
3. Use the drop-down selector to select the custom VPC. The VPC should be selected and appear similar to the image below:



1. For **Subnet settings** open the custom-vpc.txt file and fill in the details for the first public subnet. The first public subnet should be configured as in the image below:



1. At the bottom of the page on the left-hand side click **Add new subnet** and fill in the details for the second public subnet and then repeat the same steps for the two private subnets.
2. Once all four subnets have been configured click **Create subnet** to create all four subnets.

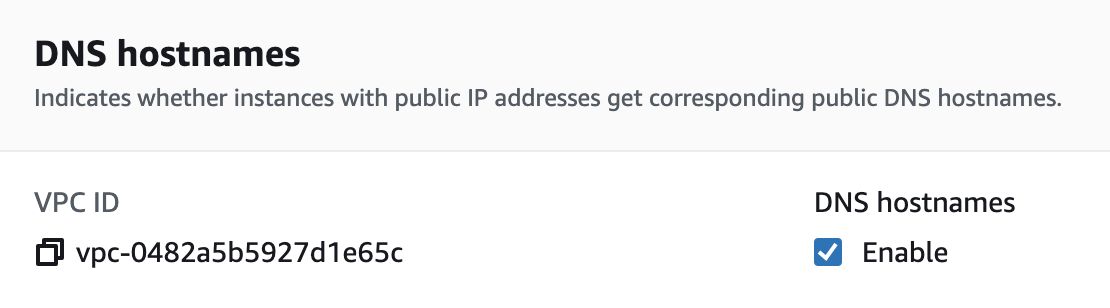
Ref custom-vpc.txt

<https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html>

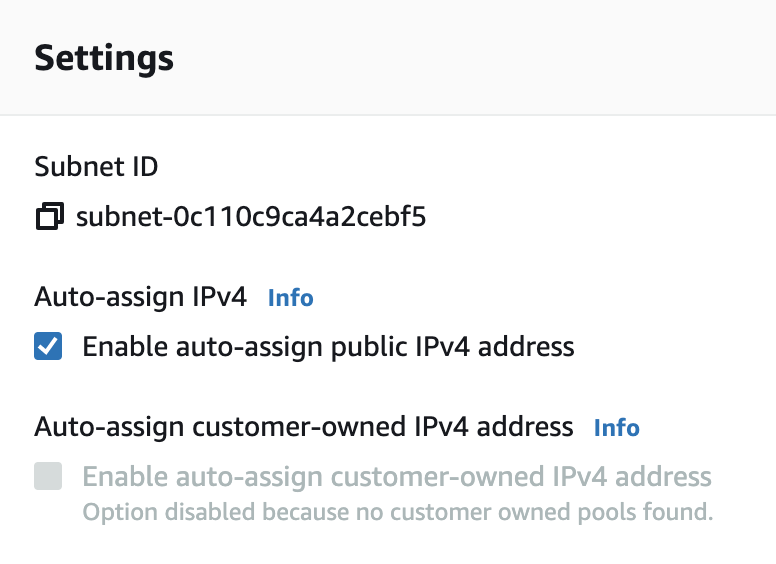
Configure VPC subnets

You need to update the configuration for the custom VPC so that DNS hostnames are assigned to Amazon EC2 instances. You also need to configure the public subnets to automatically assign public IPv4 addresses to EC2 instances. This will ensure that Amazon EC2 instances that are launched in public subnets have both public IPv4 IP addresses and DNS hostnames that are resolvable from the Internet.

1. In the **VPC Management Console** select **Your VPCs** in the left-hand navigation and then select the custom VPC created previously from the list of VPCs.
2. In the **Actions** menu select **Edit DNS hostnames** and then tick the check box under **DNS hostnames**. The window should appear similar to the image below (your VPC ID will be different):



1. Click **Save changes.**
2. In the left-hand navigation select **Subnets** and then select the **Public-1A**subnet created previously.
3. In the **Actions** menu select **Modify auto-assign IP settings** and tick the check box next to **Enable auto-assign public IPv4 address**. The window should appear similar to the image below (your Subnet ID will be different):



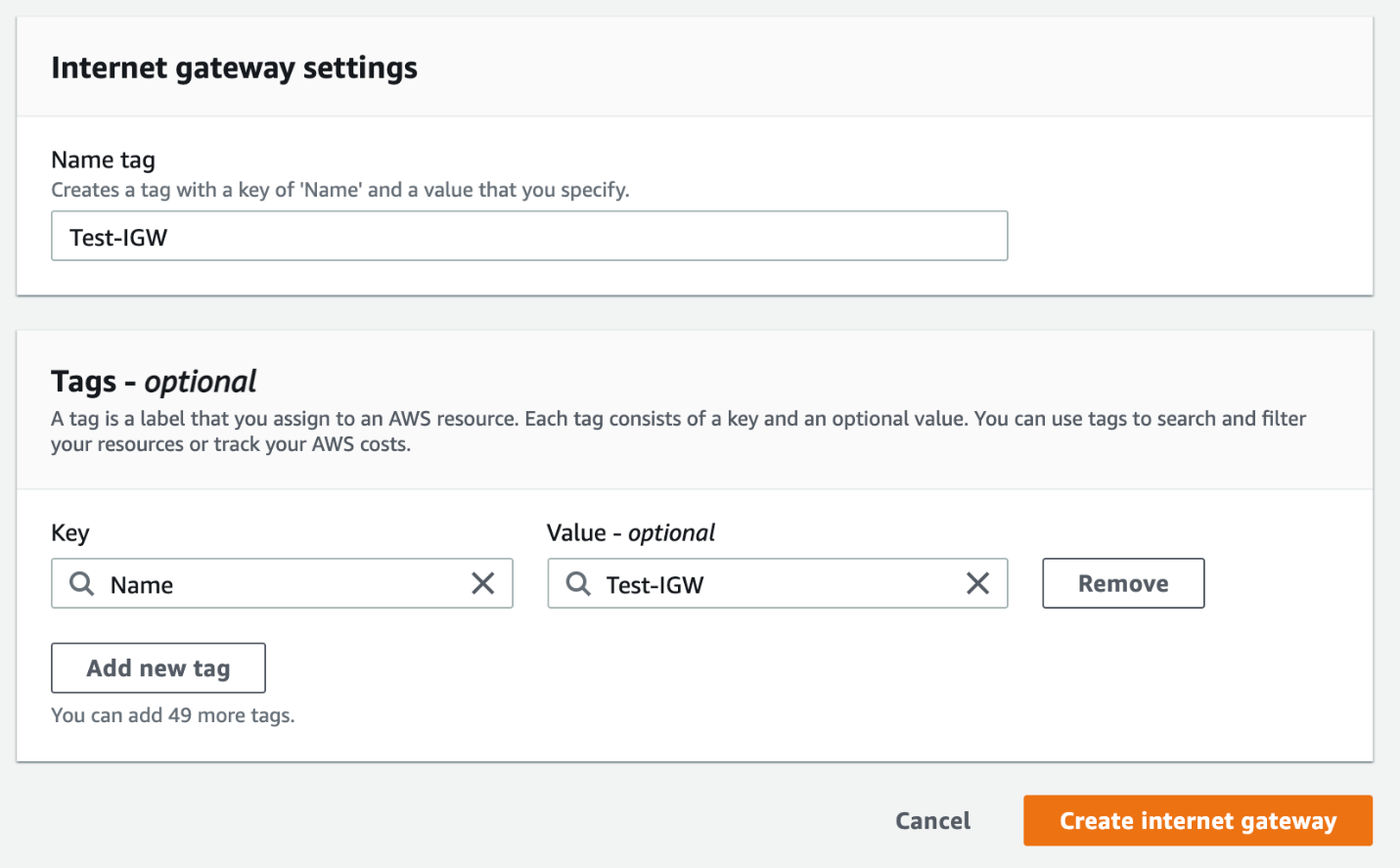
1. Click **Save** and then repeat the above step for the **Public-1B** subnet.

<https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html>

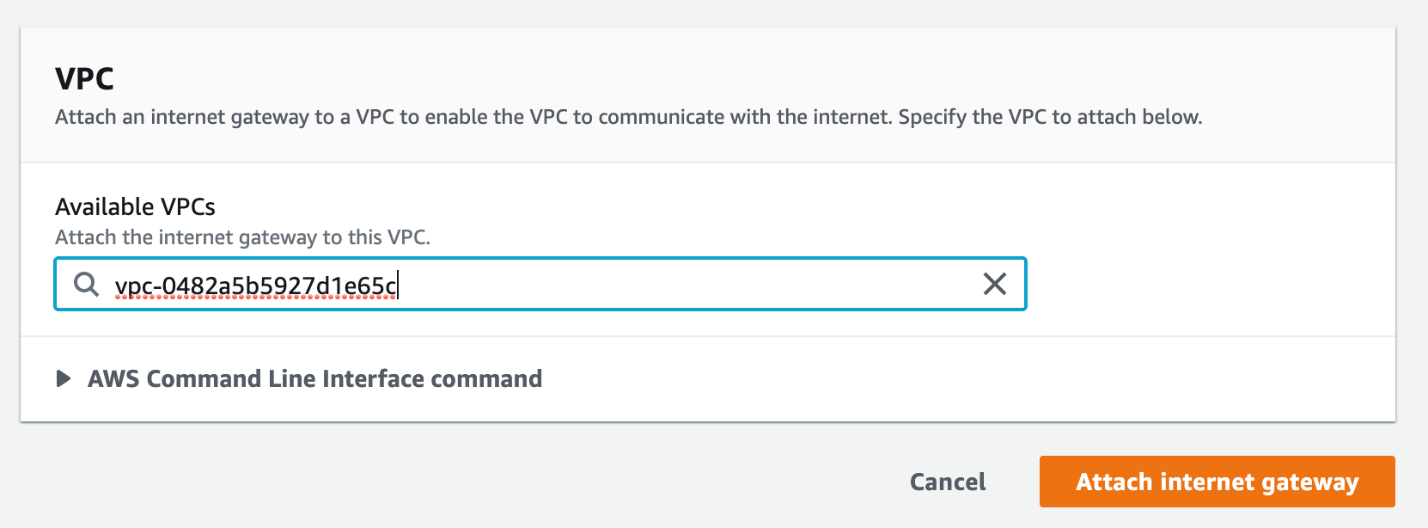
Attach an internet gateway and configure route table

You will now create an Internet Gateway and attach it to the custom VPC. This is required to enable Internet connectivity for the instances in the public subnets. You will also need to configure a route in the VPC route table pointing to the internet gateway.

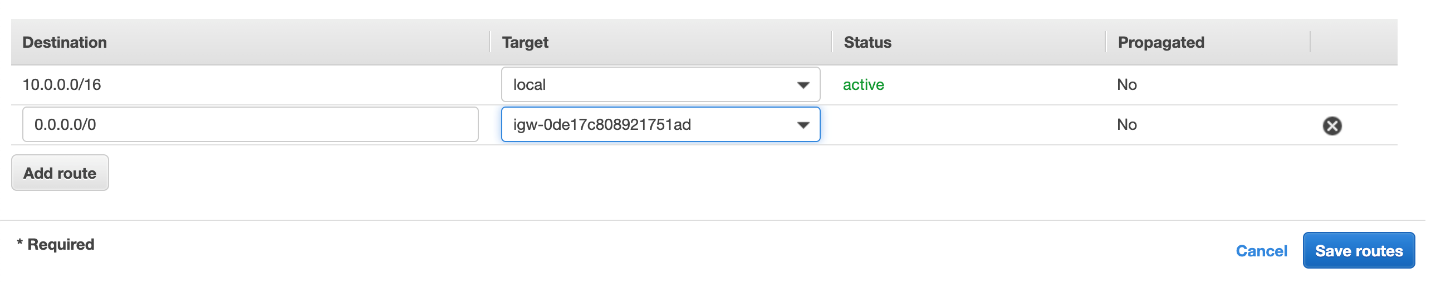
1. In the **VPC Management Console** select **Internet gateways** in the left-hand navigation and then click **Create internet gateway.**
2. For the **Name tag** enter Test-IGW and click **Create internet gateway**



1. On the confirmation page you can then attach the **internet gateway** to the VPC using the **Attach to a VPC** button or by going to **Actions** and then selecting **Attach to VPC.**
2. Use the drop-down selector to select the VPC to attach the internet gateway to. There should only be one VPC available and it is listed by the VPC ID. The window should appear similar to the following image (your VPC ID will be different):



1. Click **Attach internet gateway** to attach the internet gateway to the VPC.
2. In the left-hand navigation, select **Route Tables** and select the route table associated with the custom VPC. This can be identified by the name of the VPC in the **VPC ID** column.
3. With the route table selected, click **Routes**, then click **Edit routes.**
4. In the **Edit routes** windows click **Add route** and enter 0.0.0.0/0 for the **Destination.**
5. Use the drop-down selector under **Target**and select **Internet Gateway.**
6. The internet gateway previously created should be listed and can be selected. The windows should appear similar to the following image (your internet gateway ID will be different):



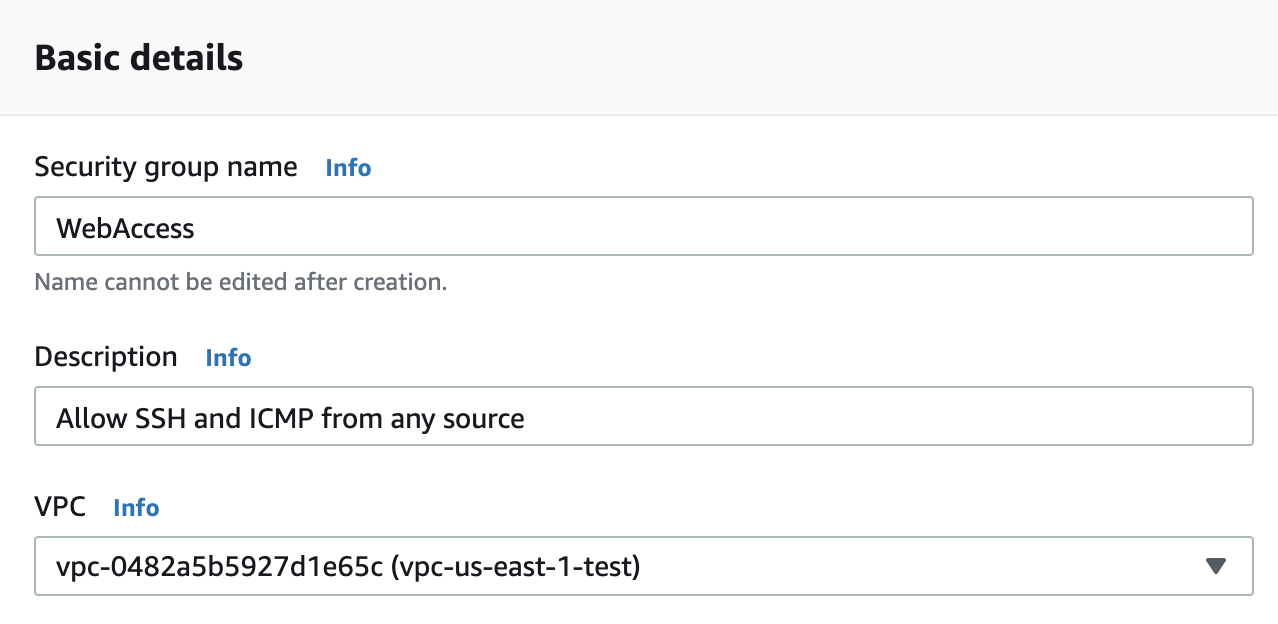
1. Click **Save routes.**

<https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html>

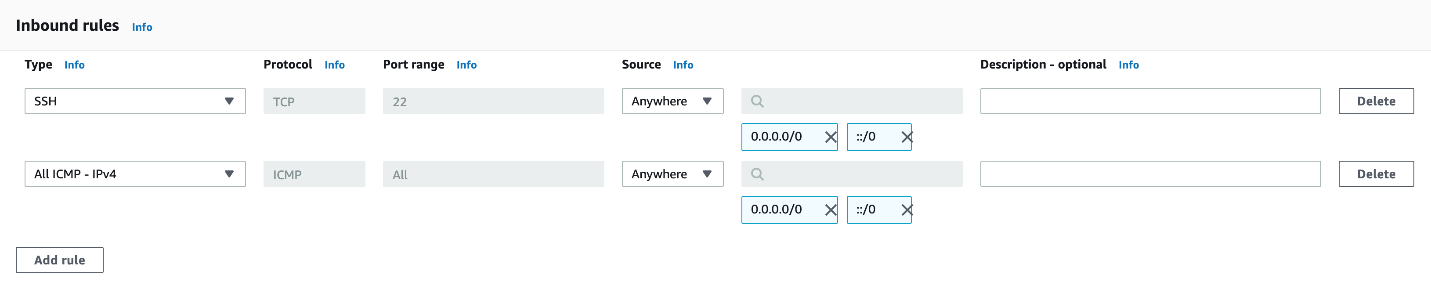
Deploy EC2 and test connectivity

You need to test that connectivity to EC2 instances launched in the custom VPC subnets works from the internet. You also need to check the connectivity within the VPC between public and private subnets. The first task is to create a security group allowing the required protocols and ports. You should then launch two Amazon EC2 instances, one in a public subnet, and one in a private subnet. You can then go ahead and test connectivity by connecting to the public instance and using the ping utility to connect from the public instance to the private instance.

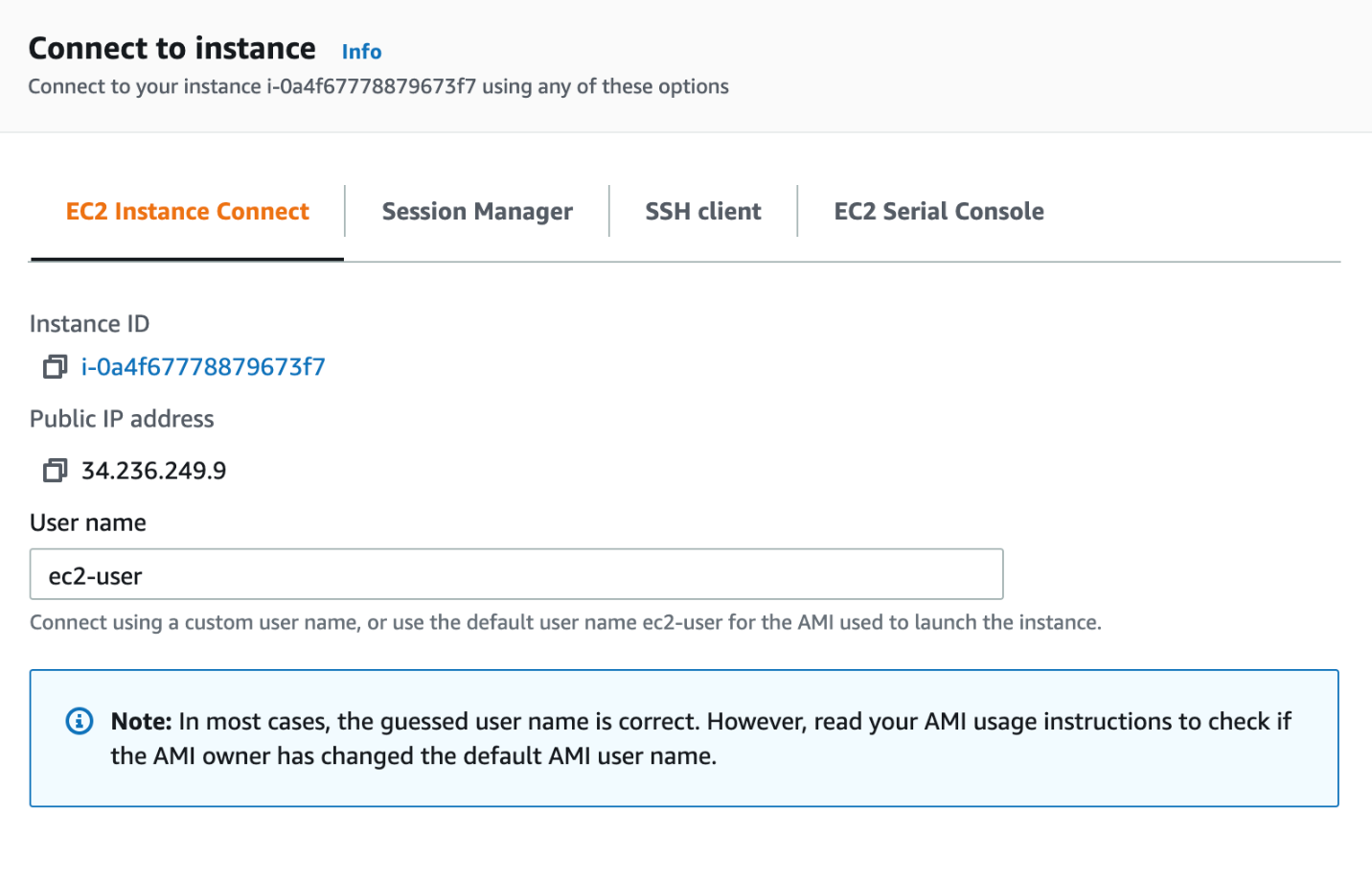
1. In the **AWS Management Console**, go to **Services** and click **EC2.**
2. In the left-hand navigation, select **Security Groups** and click **Create security group.**
3. For the **Security group name** enter WebAccess and for **Description** enter Allow SSH and ICMP from any source.
4. Use the drop-down selector under **VPC** to select the custom VPC. The basic details should now appear similar to the following image:



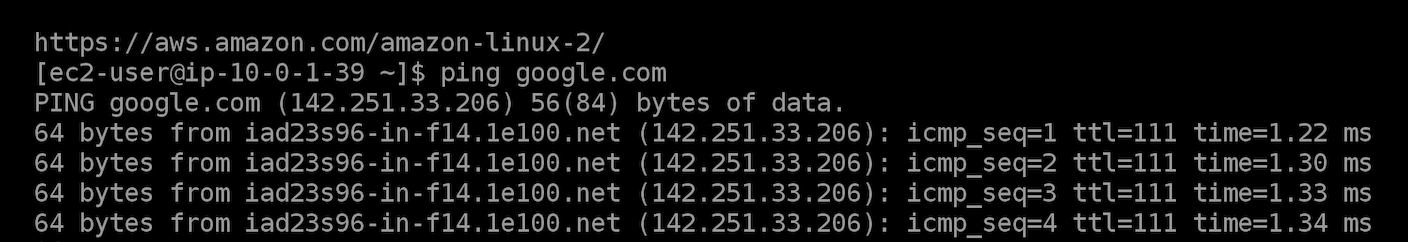
1. Under **Inbound rules** click **Add rule**. Under **Type** select **SSH** and then under **Source** select **Anywhere**.
2. Click **Add rule** again and under **Type** select **All ICMP - IPv4** and then under **Source**select **Anywhere**. The inbound rules should appear similar to the following image:



1. Do not modify the outbound rules.
2. At the bottom of the page click **Create security group.**
3. In the left-hand navigation, click **Instances** and click **Launch instances**
4. Open the instance-configuration.txt file and use the configuration details for launching the public instance.
5. Once the public instance has been launched, restart the **Launch instances** wizard and use the configuration details for launching the private instances.
6. Wait a few minutes for the instances to launch and then in the **Instances** pane, select the public instance (the one with the public IPv4 address) and click **Connect**. You should see a window that looks like this:



1. Click **Connect** to connect to the EC2 instance. A new browser window will open and you will be connected to the command line on the EC2 instance.
2. Using the command prompt type ping google.com. A successful internet connection will result in a response similar to the following:



1. Use a key sequence to stop the ping command from running (control/command + c) but dont close the instance connect browser window.
2. Back in the **Instances** pane select the EC2 instance in the private subnet and on the **Details** pane copy the **Private IPv4 address** to your clipboard.
3. In the instance connect browser window, run the command ping <private-ipv4-address>, and paste the private IP address from your clipboard. A similar response to the previous command should be seen and this indicates a successful connection to the private instance from the public instance.

Ref instance-configuration.txt

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/security-group-rules-reference.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html>

Clean up

Now that you have created a custom VPC with EC2 instances, lets clean up the lab environment.

1. Go to the **EC2** screen and click on the **Instances** menu in the left-hand navigation.
2. Select a running instance by clicking its row. This should check the left-hand checkbox. Click on the second running instances checkbox to select it as well. Go to the **Instance State** menu and select **Terminate Instance**. Press **Terminate** to confirm.
3. Go to the **VPC** screen. Select **Your VPCs** in the left-hand navigation, then select the vpc-us-east-1-test vpc. Click on the **Actions** menu and select **Delete VPC**.
4. The dialog will warn you that the security group, 4 subnets, and internet gateway will be deleted. Confirm the delete by typing in delete and pressing **Delete**.