



Amazon Web Services Hands-On VPC

Table of Contents

Overview	3
Create a VPC.....	3
VPC Object Walkthrough.....	6
Your VPCs	6
Subnets.....	7
Route Tables.....	9
Internet Gateways.....	12
DHCP Options Sets	13
Elastic IPs	14
Network ACLs	14
Security Groups	15
NAT Instance	15
Launching VPC Instances	20
Launch a Private Server	20
Launch a Public Server	26
Terminate Billable Services	33

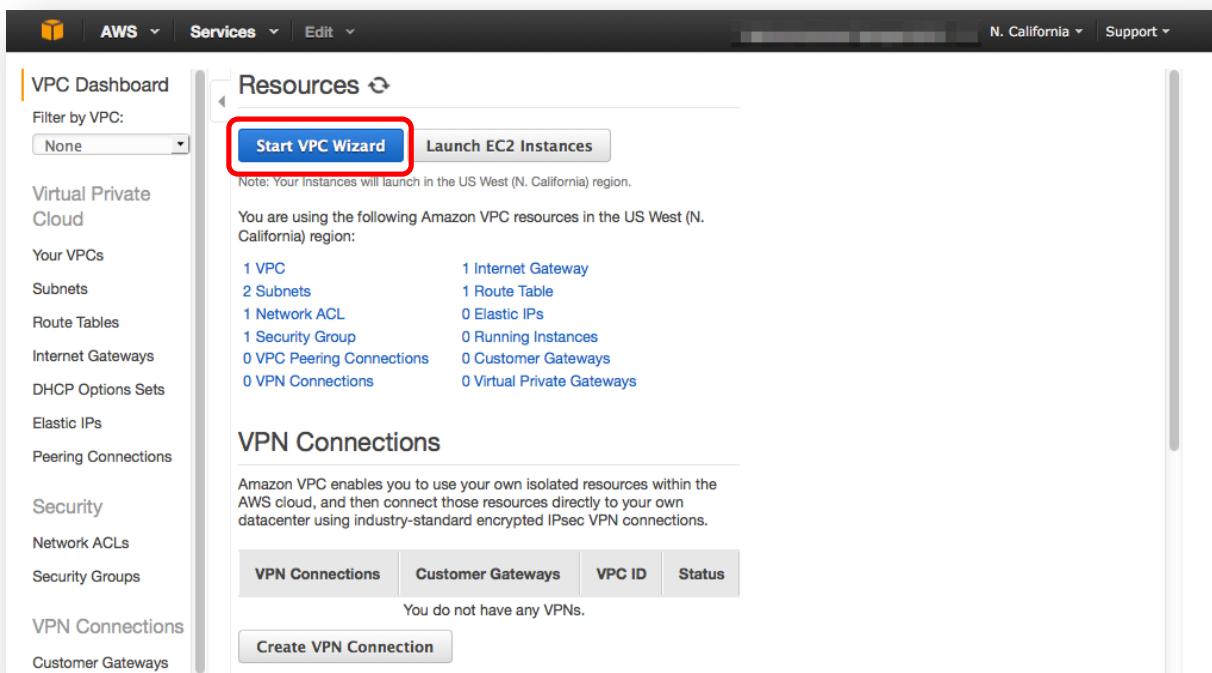
Overview

This lab will walk the user through using the VPC wizard to create a multi-subnet VPC, describe each of the objects created by the wizard, and launch instances into public and private VPC subnets. The following is high-level overview of this lab:

- Create a VPC
- Explore the different VPC objects and what they mean
- Launch EC2 instances into the VPC
- Assign a public IP address (EIP) and test public/private connectivity

Create a VPC

Log into the **AWS Console**, click on **VPC** to go to VPC console and select the **Start VPC Wizard** button to launch the VPC creation wizard.



Select the second option to create a **VPC with Public and Private Subnets** and click **Select**. Note in the picture that the wizard will automatically create and launch an EC2 “NAT” instance to serve as a gateway for your private subnets to make client connections to the Internet. We will discuss this instance in more detail later in this lab.

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via a Network Address Translation (NAT) instance in the public subnet. (Hourly charges for NAT instances apply.)

Select

Internet, S3, DynamoDB, SNS, SQS, etc.

Amazon Virtual Private Cloud

Public Subnet

Private Subnet

NAT

Cancel and Exit

On the summary page, edit the **VPC name** and the default **Public and Private Subnets** as follows, and then click **Create VPC**:

VPC name: <Your Name>

Public Subnet: 10.0.0.0/23

Private Subnet: 10.0.10.0/23

Step 2: VPC with Public and Private Subnets

IP CIDR block:* 10.0.0.0/16 (65531 IP addresses available)

VPC name: <Your Name>

Public subnet:* 10.0.0.0/23 (507 IP addresses available)

Availability Zone:* No Preference

Public subnet name: Public subnet

Private subnet:* 10.0.10.0/23 (507 IP addresses available)

Availability Zone:* No Preference

Private subnet name: Private subnet

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT instance.

Instance type:* m1.small

Key pair name: No key pair

Note: Instance rates apply. [View Rates.](#)

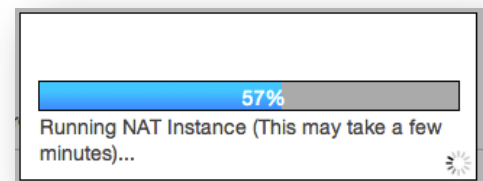
Enable DNS hostnames:* ☒ Yes ☐ No

Hardware tenancy:* Default

[Cancel and Exit](#) [Back](#) [Create VPC](#)

We are modifying the defaults to provide plenty of room to grow within each subnet (507 available IPs) to accommodate the use of additional Amazon Web Services such as ELB or RDS in VPC, as well as providing some room between the “public” and “private” subnet blocks to accommodate expansion to include multiple Availability Zones in the future as well.

The VPC wizard will create your subnet and let you know when it has been successfully created. Behind the scenes, the wizard is creating and launching the NAT instance. Click OK when it’s done



VPC Successfully Created

Your VPC has been successfully created.

You can launch instances into the subnets of your VPC. For more information, see [Launching an Instance into Your Subnet](#).

OK

VPC Object Walkthrough

After your VPC was created, you may notice that several things have been created for you as depicted in the screenshot below. The next set of steps will walk you through the various VPC objects and components that were created for you by the VPC Wizard.

The screenshot shows the AWS VPC Dashboard. On the left is a navigation menu with links like 'Virtual Private Cloud', 'Your VPCs', 'Subnets', 'Route Tables', 'Internet Gateways', 'DHCP Option Sets', 'Elastic IPs', 'Peering Connections', 'Security', 'Network ACLs', 'Security Groups', and 'VPN Connections'. The main area is titled 'Resources' and contains a red-bordered box with the following text:

Note: Your instances will launch in the US West (N. California) region.

You are using the following Amazon VPC resources in the US West (N. California) region:

1 VPCs	1 Internet Gateways
2 Subnets	2 Route Tables
1 Network ACLs	1 Elastic IPs
2 Security Groups	1 Running Instances
0 VPC Peering Connections	0 Customer Gateways
0 VPN Connections	0 Virtual Private Gateways

Below this box is a section for 'VPN Connections' with a brief description. To the right of the 'Resources' section is a 'Service Health' panel showing the status of 'Amazon VPC - US West (N. California)' and 'Amazon EC2 - US West (N. California)', both of which are 'operating normally'. Below the service health panel is an 'Additional Information' section with links to 'VPC Documentation', 'All VPC Resources', 'Forums', and 'Report an Issue'.

Your VPCs

The **Your VPCs** link provides a list of your VPCs and is a good location to obtain the VPC ID for your VPCs. If you create multiple VPCs, they will be listed here. Clicking on the VPC that was just created will bring up details about the VPC like the IP address block (CIDR), DHCP Options Set, Route Table, Network ACL, Hardware Tenancy (whether VPC physical hardware will be shared [default] or dedicated to you) and DNS configuration information.

Also note the presence of a Default VPC listed in the **Your VPCs** display. As of December 4th, 2013, we create a default VPC for you in each region. The default VPC includes a subnet per availability zone, a default security group, an Internet gateway, and other networking elements. For the purposes of this lab, we will ignore the Default VPC and focus on the VPC's created as part of the lab exercise.

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with navigation links: Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, DHCP Options Sets, Elastic IPs, Peering Connections, and Security. The main area displays a table of VPCs. Two VPCs are listed: one named '<Your Name>' with ID vpc-85619ae0 and another named 'Default' with ID vpc-9a49baff. Below the table, the details for the selected VPC are shown in a 'Summary' tab. A red box highlights the following information:

- VPC ID: vpc-85619ae0 | <Your Name>
- State: available
- VPC CIDR: 10.0.0.0/16
- DHCP options set: dopt-d2f2e0b0
- Route table: rtb-9b5e99fe
- Network ACL: acl-db39fdb
- Tenancy: Default
- DNS resolution: yes
- DNS hostnames: yes

Subnets

The **Subnets** link lists all of your VPC subnets and allows you to create additional subnets within your VPC with the **Create Subnets** button. Notice that two subnets were created because we asked the VPC Wizard to create both a public and private subnet. Clicking on a subnet will bring up subnet details including its subnet address range (CIDR), availability zone, and associated route table and network ACLs. Clicking on tabs underneath brings up relevant info about the subnet. For example, clicking on Route Table tab brings up routing information.

The screenshot shows the AWS VPC Dashboard with the 'Subnets' link selected in the sidebar. The main area displays a table of subnets. Two subnets are listed: 'Public subnet' with ID subnet-da03d1bf and 'Private subnet' with ID subnet-d803d1bd. Below the table, the details for the selected 'Public subnet' are shown in a 'Summary' tab. A red box highlights the following information:

- Subnet ID: subnet-da03d1bf | Public subnet
- CIDR: 10.0.0.0/23
- State: available
- VPC: vpc-85619ae0 (10.0.0.0/16) | <Your Name>
- Available IPs: 506
- Availability Zone: us-west-1b
- Route table: rtb-9a5e99ff
- Network ACL: acl-db39fdb
- Default subnet: no
- Auto-assign Public IP: no

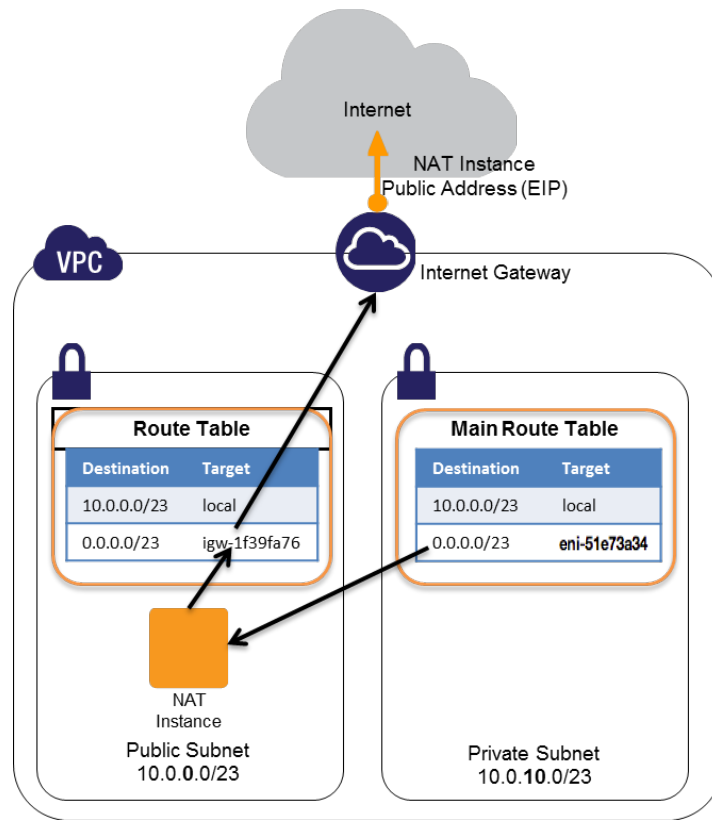
Click on Route Table tab and notice that this subnet's default route (0.0.0.0) is the Internet Gateway (described below in the Internet Gateway section). Internet Gateways can be identified by "igw" prefix in its ID. This route makes this subnet your "public" subnet because it is publically routable through the Internet Gateway.

The screenshot shows the AWS VPC console interface. On the left is a navigation menu with options like VPC Dashboard, Subnets, Route Tables, Internet Gateways, etc. The main area displays a list of subnets. The first subnet, 'Public subnet' (subnet-da03d1bf), is selected. Below the list, the 'Route Table' tab is active for 'subnet-da03d1bf (10.0.0.0/23) | Public subnet'. A red box highlights the 'Route Table' section, which shows a table with two routes:

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-83db33e6

If you click on the second subnet you will notice a different routing table.

This subnet's default route (0.0.0.0) is another EC2 Instance's Elastic Network Interface (ENI). This EC2 instance is the NAT instance and is described in detail below. The network interface attached to the NAT instance can be identified by the "eni-" prefix in its ID. This route makes this subnet your "private" subnet because it is not publically routable through the Internet Gateway. Instead, all client connections to the Internet are directed to, and proxied by, your NAT instance in the "public" subnet. The following diagram depicts these subnets that were created for you:



Route Tables

The **Route Tables** link lists all of your VPC route tables, allows you to modify and associate the route tables to subnets, and allows you to create additional route tables within your VPC with the **Create Route Table** button. Notice that two route tables were created by the VPC Wizard, and these are the same route tables that were displayed in the subnet details in the previous section. Notice the **Main** and **Associated With** columns. The subnet designated as the “Main” subnet (Main = Yes) is the default route table for the listed VPC. This means that all subnets that are not explicitly associated with a more specific route table will use this route table by default. The Associated With column displays number of subnets explicitly associated with the route table.

VPC Hands-On Lab

VPC Dashboard

Filter by VPC: None

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

DHCP Options Sets

Elastic IPs

Peering Connections

Security

Create Route Table Delete Route Table Set As Main Table

Search Route Tables and th X

<< 1 to 3 of 3 Route Tables >>

	Name	Route Table ID	Associated With	Main	VPC
<input type="checkbox"/>		rtb-879f60e2	0 Subnets	Yes	vpc-9a49baff (172.31.0.0/16) Default
<input checked="" type="checkbox"/>		rtb-9a5e99ff	1 Subnet	No	vpc-85619ae0 (10.0.0.0/16) <Your Nam...
<input type="checkbox"/>		rtb-9b5e99fe	0 Subnets	Yes	vpc-85619ae0 (10.0.0.0/16) <Your Nam...

rtb-9a5e99ff

Summary Routes Subnet Associations Route Propagation Tags

Route Table ID: rtb-9a5e99ff

Associated With: 1 Subnet

Main: no

VPC: vpc-85619ae0 (10.0.0.0/16) | <Your Name>

Notice that only 1 of the 2 subnets created with the VPC is associated with a route table. The second subnet is not explicitly associated with a route table and is therefore using the “Main” route table (rtb-9b5e99fe).

Clicking on a route table will bring up details about the route. Clicking on Routes tab underneath will bring up routing info as well as the ability to modify the route table’s routes by clicking on **Edit** button. Similarly you can view or modify Subnet Associations, Route Propagation and Tag information pertaining to the selected route.

Search Route Tables and th X

<< 1 to 3 of 3 Route Tables >>

	Name	Route Table ID	Associated With	Main	VPC
<input type="checkbox"/>		rtb-879f60e2	0 Subnets	Yes	vpc-9a49baff (172.31.0.0/16) Default
<input checked="" type="checkbox"/>		rtb-9a5e99ff	1 Subnet	No	vpc-85619ae0 (10.0.0.0/16) <Yo...
<input type="checkbox"/>		rtb-9b5e99fe	0 Subnets	Yes	vpc-85619ae0 (10.0.0.0/16) <Yo...

rtb-9a5e99ff

Summary Routes Subnet Associations Route Propagation Tags

Edit

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-83db33e6	Active	No

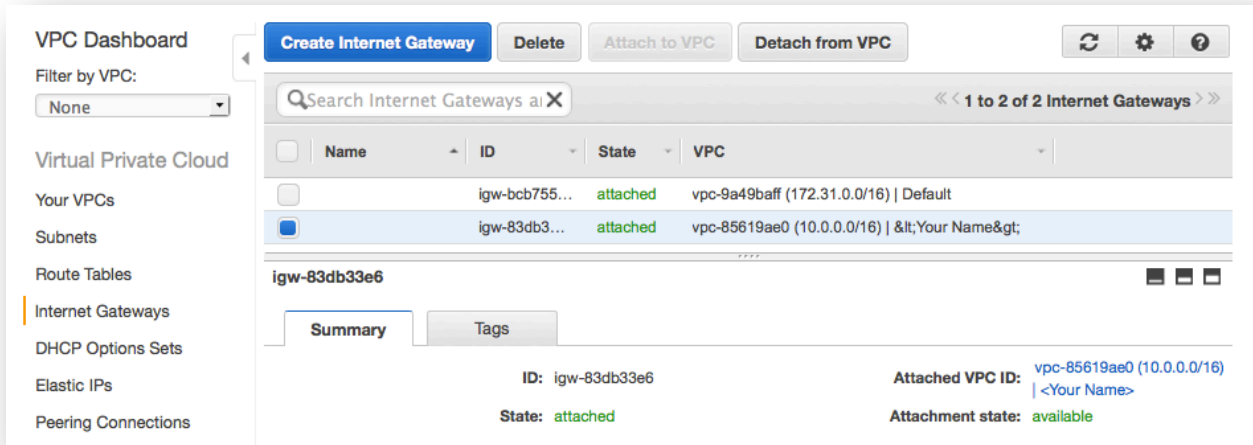
Notice that the selected route table (rtb-9a5e99ff) is NOT the Main route table (Main = No) and its default route (0.0.0.0) is the Internet Gateway (igw-83db33e6). This means your “public” subnet is explicitly associated with this route table (click on the Subnet Associations tab to verify this). If you select the second route table, you will see the default route (0.0.0.0) is your NAT instance.

So what does all this mean? By default, the VPC Wizard created two subnets and two route tables. The “public” subnet is associated with a route table that directs traffic by default out to the Internet. The “private” subnet is not associated with a specific route table and therefore inherits the Main route table rules which directs traffic by default to the NAT instance in the “Public” subnet.

One more thing to note: The rules in the Main route table determine how subnets will be treated by default. Since the Main route table is a “private” route table (it does not route any traffic to the Internet Gateway), all new subnets created in this VPC will be “private” subnets by default. They will remain “private” until they are explicitly associated with a “public” route table (e.g. one that routes traffic directly to the Internet Gateway).

Internet Gateways

An Internet Gateway provides 1-to-1 static network address translation (NAT) mapping for your VPC instance internal IP addresses to publically routable Elastic IP addresses that you must explicitly associate with your “public” VPC instances. For the purposes of this lab, the VPC Wizard created an Internet Gateway and associated it with your VPC.



The screenshot displays the AWS VPC Dashboard's Internet Gateways section. On the left, a sidebar lists navigation options: VPC Dashboard, Filter by VPC (set to None), Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways (selected), DHCP Options Sets, Elastic IPs, and Peering Connections. The main panel shows a table of Internet Gateways with columns for Name, ID, State, and VPC. Two gateways are listed: 'igw-bcb755...' (Default) and 'igw-83db33e6' (Attached to vpc-85619ae0). The 'igw-83db33e6' gateway is selected, and its details are shown in the 'Summary' tab. The details include the ID 'igw-83db33e6', State 'attached', and Attached VPC ID 'vpc-85619ae0 (10.0.0.0/16) | <Your Name>'. The Attachment state is 'available'.

Name	ID	State	VPC
igw-bcb755...	igw-bcb755...	attached	vpc-9a49baff (172.31.0.0/16) Default
igw-83db33e6	igw-83db33e6	attached	vpc-85619ae0 (10.0.0.0/16) <Your Name>

igw-83db33e6

Summary | Tags

ID: igw-83db33e6

State: attached

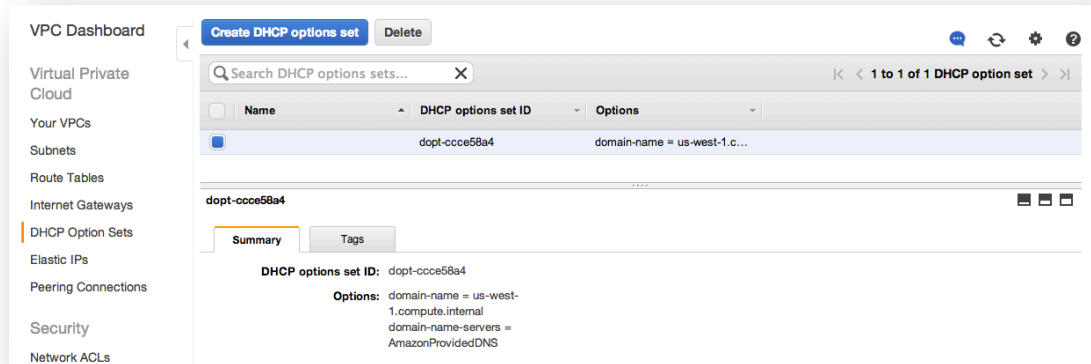
Attached VPC ID: vpc-85619ae0 (10.0.0.0/16) | <Your Name>

Attachment state: available

You do not need to do anything specifically with the Internet Gateway in this lab. We point it out here to explain the Internet Gateway that was created for you, and to point out that Internet Gateways can be independently created, attached and detached to VPCs. This allows you to add or remove the Internet Gateway capabilities to your VPCs after the VPC has been created.

DHCP Options Sets

The **DHCP Options Sets** link allows you to control some DHCP options that the VPC provided DHCP service will present to your instances when they boot. By default the VPC Wizard created a DHCP Options set that tells your VPC instances to use the AWS provided DNS service for domain name resolution.



VPC allows you to create and attach new DHCP Options to your VPCs including setting your domain name, domain name (DNS) servers, time (NTP) servers, and Microsoft Windows NetBIOS name servers and node type. The following screenshot depicts how these options can be configured when creating a new DHCP Options Set.

The screenshot shows the 'Create DHCP options set' dialog box. It includes a description of DHCP and fields for configuring various parameters. The 'Yes, Create' button is highlighted.

Create DHCP options set

Dynamic Host Configuration Protocol (DHCP) provides a standard for passing configuration information to hosts on a TCP/IP network. The options field of a DHCP message contains configuration parameters.

Name tag:

Specify at least one of the following configuration parameters

Domain Name:

Domain Name Servers:

NTP Servers:

NetBIOS Name Servers:

NetBIOS Node Type:

Cancel **Yes, Create**

Elastic IPs

VPC Elastic IPs are static, publically routable IP addresses that you can associate with your VPC Instances. Although you have not launched any VPC instances yet, the VPC Wizard launched a VPC NAT instance, created a public Elastic IP address, and associated the EIP with the NAT instance on your behalf. You can see this EIP and association by clicking on the **Elastic IPs** link and selecting the Address.

The screenshot shows the AWS VPC Dashboard with the 'Elastic IPs' link selected in the left-hand navigation menu. The main panel displays a table of Elastic IP addresses. One address is listed: 54.183.17.243, associated with instance ID i-a2d087fe and network interface ID eni-51e73a34. Below the table, the 'Summary' tab is active, showing details for the selected address: Address: 54.183.17.243, Instance ID: i-a2d087fe, Network Interface ID: eni-51e73a34, Network Interface Owner: 896501016854, Scope: vpc, and Allocation ID: eipalloc-4a948228.

Network ACLs

Network Access Control Lists (NACLs) act as a subnet *stateless* firewall, controlling ingress and egress for an entire subnet (as a second layer of defense on top of security groups). If you click on the **Network ACLs** link you will see that the VPC Wizard created a single “default” NACL for your VPC with a default Allow ALL rule. Since NACLs are stateless, we recommend using NACLs only when you want to explicitly deny traffic. For example, we never want to use TFTP or “this” subnet should never be able to talk to “that” subnet.

The screenshot shows the AWS VPC Dashboard with the 'Network ACLs' link selected in the left-hand navigation menu. The main panel displays a table of Network ACLs. Two are listed: acl-9e817dfb and acldb39fdbe. The second one is selected. Below the table, the 'Summary' tab is active, showing details for the selected ACL: Network ACL ID: acldb39fdbe, Default: yes, Associated with: 2 Subnets, and VPC: vpc-85619ae0 (10.0.0.0/16). The 'Inbound Rules' and 'Outbound Rules' tabs are also visible.

Security Groups

At this point you should already be familiar with EC2 Security Groups and understand the difference between [EC2 and VPC Security Groups](#). The **Security Groups** link allows you to see your VPC Security Groups. Notice that the VPC Wizard created Security Group for you called “default”.

The screenshot shows the AWS VPC Dashboard. On the left is a sidebar with navigation links: VPC Dashboard, Filter by VPC: (None), Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, DHCP Options Sets, Elastic IPs, Peering Connections, Security (highlighted), Network ACLs, and Security Groups. The main content area has buttons for 'Create Security Group' and 'Delete Security Group'. Below these is a filter section with 'All security groups' and a search bar. A table lists security groups:

Name tag	Group ID	Group Name	VPC	Description
	sg-b412d8d1	default	vpc-9a49baff (172.31.0.0/16...)	default VPC security group
<input checked="" type="checkbox"/>	sg-a606d8c3	default	vpc-85619ae0 (10.0.0.0/16...)	default VPC security group

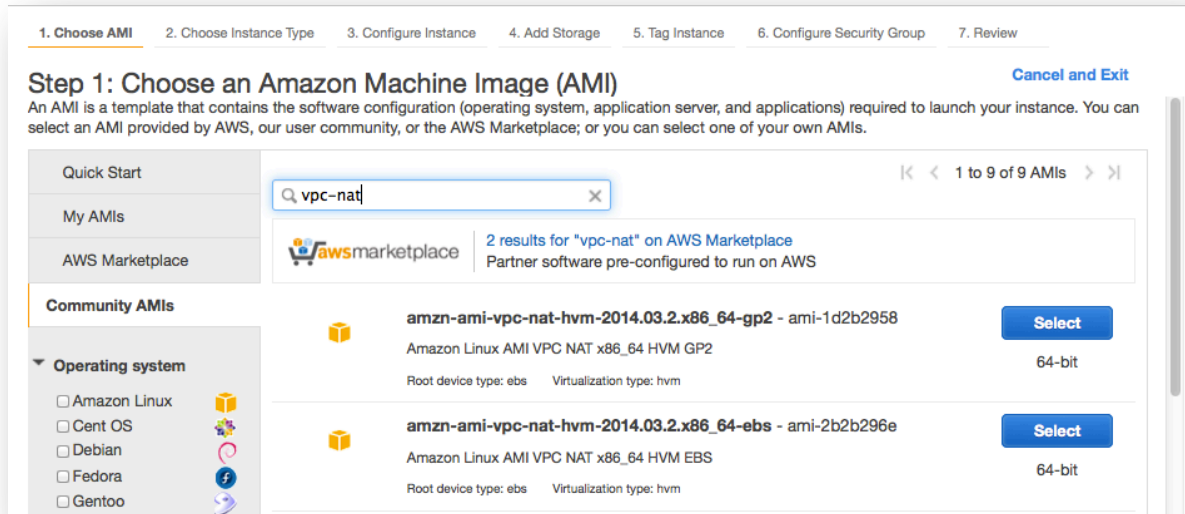
Below the table, the details for the selected security group 'sg-a606d8c3' are shown. It has tabs for Summary, Inbound Rules, Outbound Rules, and Tags. The Summary tab is active, showing:

- Group name: default
- Group ID: sg-a606d8c3
- VPC: vpc-85619ae0 (10.0.0.0/16) | <Your Name>
- Group description: default VPC security group

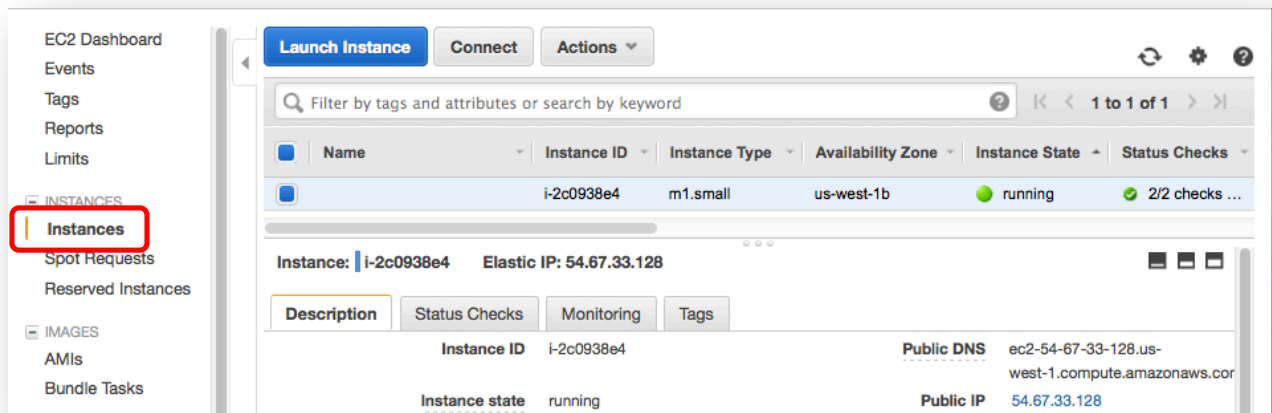
NAT Instance

So far we have mentioned a NAT instance several times during this walkthrough. In any private network (such as a corporate or home network), Internet clients must pass through a device that will translate “private” (internal, non-Internet routable) addresses to a “public” (external, Internet routable) address for routing over the Internet. At home this is typically some sort of wireless router and at work it is typically a corporate firewall or web monitoring proxy server. At its most basic level, this device or service provides a 1-to-many mapping where many private clients masquerade as a single public address. A NAT instance provides this capability within a VPC, allowing multiple private instances to indirectly make outbound client connections to the Internet without requiring individual public IPs. There is nothing overly special about the VPC Wizard provided NAT instance. Technically it is simply an Amazon Linux instance with IP masquerading enabled using iptables. You can find the most recent AMIs

by searching AWS public AMIs for “VPC-nat”.



To view the NAT instance, you must unfortunately leave the VPC tab, go to the **EC2 Tab** in the AWS Management Console and click on the **Instances** link.



Locate the NAT instance. The most effective way to locate the NAT instance is to search by its instance ID. We discovered this ID previously in the

Subnets and **Route Tables** sections above. However it is also easy to locate in a new AWS account because it will be the only instance running (or the only instance that has not been given a tag yet). Clicking on the instance will bring up its details, including the VPC and Subnet where the instance is running.

VPC Hands-On Lab

The screenshot displays the AWS Management Console for an EC2 instance. The left sidebar shows the navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area shows the instance details for 'i-2c0938e4'. The instance is in a 'running' state, using the 'm1.small' instance type in the 'us-west-1b' availability zone. The 'Description' tab is active, showing various attributes. A red box highlights the instance ID 'i-2c0938e4' in the table and the 'VPC ID' 'vpc-85619ae0' and 'Subnet ID' 'subnet-da03d1bf' in the details section.

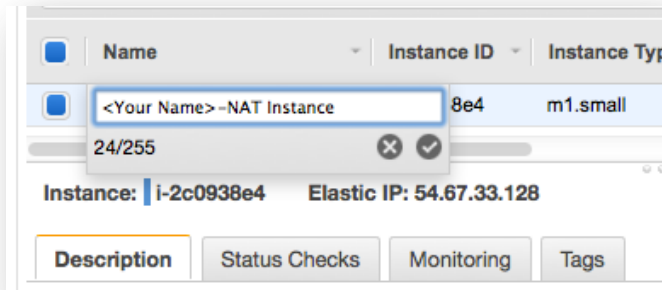
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-2c0938e4	m1.small	us-west-1b	running	2/2 checks ...

Instance: i-2c0938e4 Elastic IP: 54.67.33.128

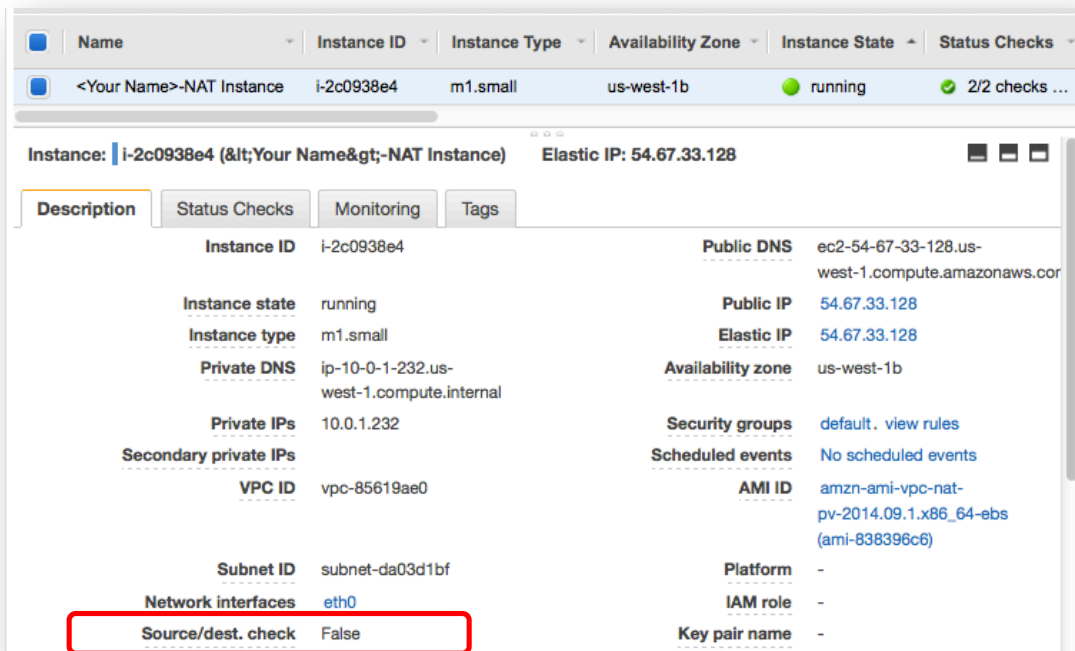
Description Status Checks Monitoring Tags

Instance ID	i-2c0938e4	Public DNS	ec2-54-67-33-128.us-west-1.compute.amazonaws.com
Instance state	running	Public IP	54.67.33.128
Instance type	m1.small	Elastic IP	54.67.33.128
Private DNS	ip-10-0-1-232.us-west-1.compute.internal	Availability zone	us-west-1b
Private IPs	10.0.1.232	Security groups	default, view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-85619ae0	AMI ID	amzn-ami-vpc-nat-pv-2014.09.1.x86_64-ebs (ami-838396c6)
Subnet ID	subnet-da03d1bf	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	False	Key pair name	-

We recommend that you give this instance a Name like **<Your Name>-NAT Instance** to make it easier to identify in the future. You can do this either by clicking on the Tags tab or by putting your mouse over the empty name column, clicking on the pen icon, and entering the name directly.



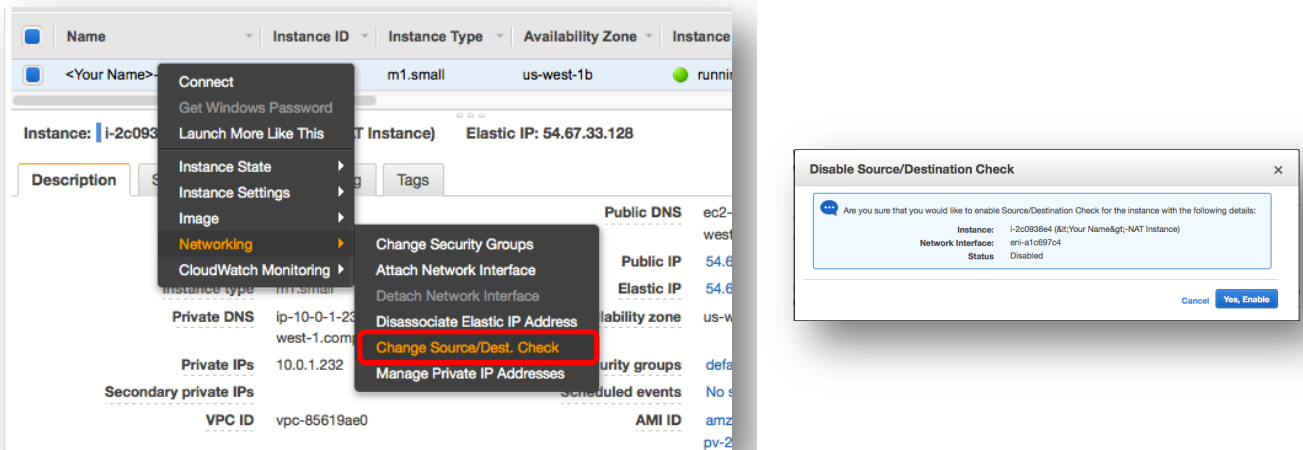
One last thing to note about this instance – in the instance details, notice that Source/Dest. Check is listed as **False**.



By default, AWS assumes that your VPC instances are network hosts and not routers – meaning that they should only send and receive network traffic addressed to its private IP. This results in AWS performing Source & Destination IP address checking (enabled) to ensure your instances can only send and receive traffic addressed to/from their address. Since a NAT instance's job is to route traffic

between internal clients and the Internet, it will require this option to be False (which the VPC Wizard already did for you).

To modify this setting in the future (e.g. if you want to allow other instances to similarly route traffic for compliance or security reasons), right-click on an instance, select **Networking** -> **Change Source/Dest Check**, and click the option to “Yes, Enable” or “Yes, Disable” depending on whether the current setting is disabled or enabled.



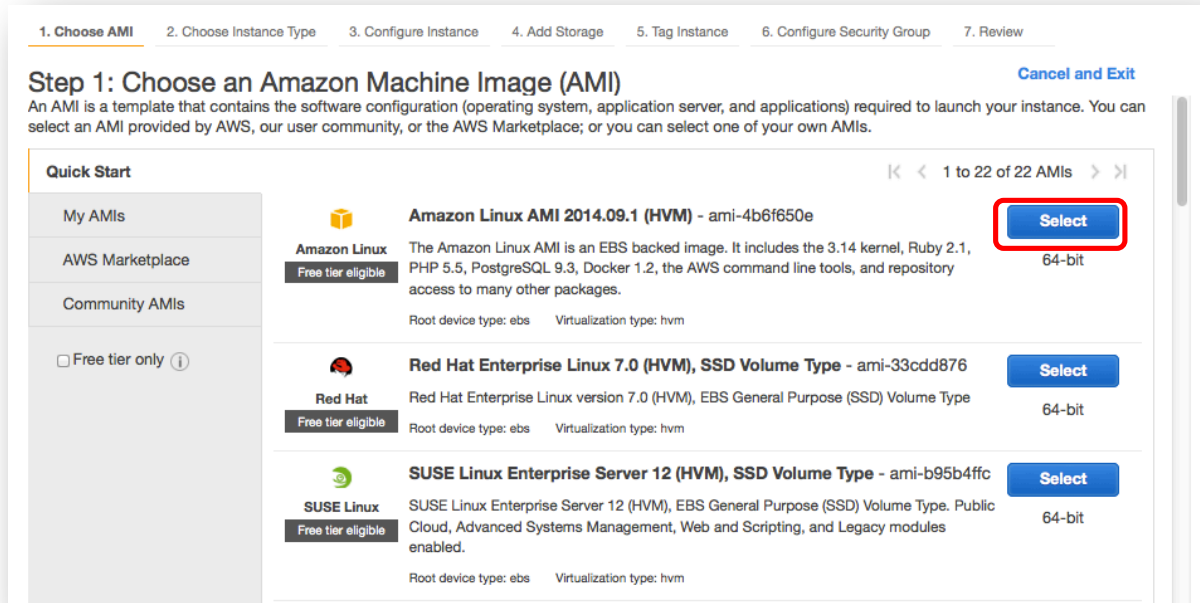
Launching VPC Instances

Walk through launching an instance in the private subnet. Create a security group and allow ICMP requests from the VPC CIDR. Notice how there is no public way to route to the instance (e.g. you can't ping it)?

Now launch an instance in the public subnet. Create a new security group and allow ICMP requests from the world. Note how you still can't ping it? Add an EIP. Note how you can now ping the public instance but not the private one. Connect to public instance and ping the private one.

Launch a Private Server

In the AWS Management Console, EC2 tab, click on the **Launch Instance** button. On **Step 1: Choose an Amazon Machine Image (AMI)** select the latest Amazon Linux AMI.



On **Step 2: Choose an Instance Type**, change the instance type to **t2.micro** and click **Next: Configure Instance Details**

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate

Cancel Previous Review and Launch Next: Configure Instance Details

On **Step 3: Configure Instance Details**, select the **VPC** and **Private Subnet** that was created in previous steps and click **Next: Add Storage**

VPC Hands-On Lab

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances ⓘ 1

Purchasing option ⓘ ☐ Request Spot Instances

Network ⓘ vpc-85619ae0 (10.0.0.0/16) | <Your Name> ⓘ Create new VPC

Subnet ⓘ subnet-d803d1bd (10.0.10.0/23) | Private sub ⓘ Create new subnet
507 IP Addresses available

Auto-assign Public IP ⓘ Use subnet setting (Disable) ⓘ

IAM role ⓘ None ⓘ

Shutdown behavior ⓘ Stop ⓘ

Enable termination protection ⓘ ☐ Protect against accidental termination

Monitoring ⓘ ☐ Enable CloudWatch detailed monitoring
[Additional charges apply.](#)

Tenancy ⓘ Shared tenancy (multi-tenant hardware) ⓘ
[Additional charges will apply for dedicated tenancy.](#)

Network interfaces

Cancel Previous **Review and Launch** Next: Add Storage

Leave defaults on Step 4. On the next screen, Step 5 (Tag Instance), you can provide a name for your private server (e.g. **Private Server**) and click **Next**.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	Private Server ⓘ

Create Tag (Up to 10 tags maximum)

Cancel Previous **Review and Launch** Next: Configure Security Group

On **Step 6: Configure Security Group**, create a new security group. In this example we call it **Private_Servers** and give permission for all instances in the VPC to “ping” these servers.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source
All ICMP	ICMP	0 - 65535	Custom IP 10.0.0.0/16

Add Rule

Warning
You will not be able to connect to this instance as the AMI requires port(s) 22 to be open in order to have access. Your current security group doesn't have port(s) 22 open.

Cancel Previous **Review and Launch**


Review your selected options and **Launch** your instance.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▼ AMI Details [Edit AMI](#)

 **Amazon Linux AMI 2014.09.1 (HVM) - ami-4b6f650e**

The Amazon Linux AMI is an EBS backed image. It includes the 3.14 kernel, Ruby 2.1, PHP 5.5, PostgreSQL 9.3, Docker 1.2, the AWS command line tools, and repository access to many other packages.

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups [Edit security groups](#)

Security group name Private_Servers
Description Lab Private Servers

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
All ICMP	All	N/A	10.0.0.0/16

▼ Instance Details [Edit instance details](#)

[Cancel](#) [Previous](#) [Launch](#)

Use your existing key pair, acknowledge that you have access to the selected private key file (*.pem) and click **Launch Instances**.

Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Choose an existing key pair

Select a key pair

Lab

☒ I acknowledge that I have access to the selected private key file (*.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

You have now launched a private server in your VPC. Find the new instance in your list of EC2 instances and select it. In the instance description, note that the instance has a private IP address (10.0.10.177 in the screenshot below), but does not have any associated public information for connecting to this instance (e.g. no EIP or Public DNS information). This instance is only locally accessible from within your VPC (theoretically it could also be locally accessible from inside a corporate network if we had established a hardware VPN connection to the VPC from our corporate network).

The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there is a search bar and a table of instances. The instance 'Private Server' (ID: i-5e6e5e96) is selected. Below the table, the instance details are displayed in a tabbed view, with the 'Description' tab active. The details show that the instance is running in the us-west-1b availability zone, has a private IP of 10.0.10.177, and is associated with the VPC ID vpc-85619ae0 and Subnet ID subnet-d803d1bd. The instance type is t2.micro and the AMI ID is amzn-ami-hvm-2014.09.1.x86_64-ebs (ami-4b6f650e).

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
<input checked="" type="checkbox"/> Private Server	i-5e6e5e96	t2.micro	us-west-1b	running	Initializing
<input type="checkbox"/> <Your Name>-NAT Instance	i-2c0938e4	m1.small	us-west-1b	running	2/2 checks ..

Instance: i-5e6e5e96 (Private Server) Private IP: 10.0.10.177

Description | Status Checks | Monitoring | Tags

Instance ID	i-5e6e5e96	Public DNS	-
Instance state	running	Public IP	-
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-10-0-10-177.us-west-1.compute.internal	Availability zone	us-west-1b
Private IPs	10.0.10.177	Security groups	Private_Servers. view rules
Secondary private IPs	-	Scheduled events	No scheduled events
VPC ID	vpc-85619ae0	AMI ID	amzn-ami-hvm-2014.09.1.x86_64-ebs (ami-4b6f650e)
Subnet ID	subnet-d803d1bd	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	Lab
		Owner	518372419593

Launch a Public Server

Now that you have a private server, we will launch a public server and differentiate between the two. In the AWS Management Console, EC2 tab, click on **Launch Instance** button. On **Step 1: Choose an Amazon Machine Image (AMI)** select the 64-bit Amazon Linux AMI. Change the instance type to **t2.micro** on Step 2. On Step 3, select the **VPC** and select the Public subnet (**10.0.0.0/23**).

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances ⓘ 1

Purchasing option ⓘ ☐ Request Spot Instances

Network ⓘ vpc-85619ae0 (10.0.0.0/16) | <Your Name> ⓘ Create new VPC

Subnet ⓘ subnet-da03d1bf(10.0.0.0/23) | Public subnet ⓘ Create new subnet
506 IP Addresses available

Auto-assign Public IP ⓘ Use subnet setting (Disable) ⓘ

IAM role ⓘ None ⓘ

Shutdown behavior ⓘ Stop ⓘ

Enable termination protection ⓘ ☐ Protect against accidental termination

Monitoring ⓘ ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy ⓘ Shared tenancy (multi-tenant hardware) ⓘ
Additional charges will apply for dedicated tenancy.

Network interfaces

Cancel Previous **Review and Launch** **Next: Add Storage**

Leave the defaults on Step 4, provide a name for your private server (e.g. **Public Server**) and click **Next**.

VPC Hands-On Lab

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Tag Instance

6. Configure Security Group

7. Review

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	Public Server

Create Tag (Up to 10 tags maximum)

[Cancel](#)[Previous](#)[Review and Launch](#)[Next: Configure Security Group](#)

Create a new security group for your public servers. In this example we create a security group called **Public_Servers**, with rules to allow anyone to “ping” and SSH into the instance.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere (0.0.0.0/0)
All ICMP	ICMP	0 - 65535	Anywhere (0.0.0.0/0)

Warning
 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

Finally, review your settings, click **Launch** and use your existing key pair, acknowledge that you have access to the selected private key file (*.pem) and click **Launch Instances**.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Choose an existing key pair:

Select a key pair:

☒ I acknowledge that I have access to the selected private key file (*.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

You have now launched a server in your public subnet; however it is still not publicly accessible. Find the new instance in your list of EC2 instances and select it. In the instance description, note that the instance has a private IP address (10.0.1.79 in the screenshot below), but does not have any associated public information for connecting to this instance (e.g. no EIP or Public DNS information) – just like your private instance.

The screenshot shows the AWS Management Console interface. At the top, there is a search bar and a filter dropdown. Below this is a table of EC2 instances. The first instance, 'Public Server' with ID 'i-e26d5d2a', is selected and highlighted with a red box. Below the table, the details for this instance are shown. The 'Description' tab is active, displaying various attributes. A red box highlights the 'Private DNS' field, which shows 'ip-10-0-0-74.us-west-1.compute.internal'. Other fields include 'Instance ID', 'Instance state', 'Instance type', 'Private IPs', 'Secondary private IPs', 'VPC ID', 'Subnet ID', 'Network interfaces', 'Public DNS', 'Public IP', 'Elastic IP', 'Availability zone', 'Security groups', 'Scheduled events', 'AMI ID', 'Platform', and 'IAM role'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm St
Public Server	i-e26d5d2a	t2.micro	us-west-1b	running	Initializing	None
Private Server	i-5e6e5e96	t2.micro	us-west-1b	running	2/2 checks ...	None
<Your Name>-NAT Instance	i-2c0938e4	m1.small	us-west-1b	running	2/2 checks ...	None

Instance: **i-e26d5d2a (Public Server)** Private IP: 10.0.0.74

Description | Status Checks | Monitoring | Tags

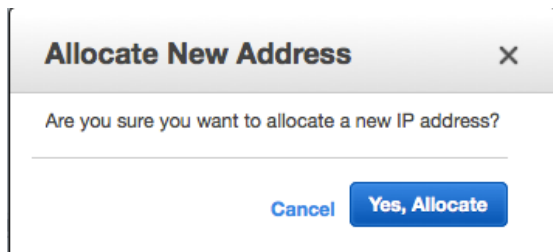
Instance ID	i-e26d5d2a	Public DNS	-
Instance state	running	Public IP	-
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-10-0-0-74.us-west-1.compute.internal	Availability zone	us-west-1b
Private IPs	10.0.0.74	Security groups	Public_Server. view rules
Secondary private IPs	-	Scheduled events	No scheduled events
VPC ID	vpc-85619ae0	AMI ID	amzn-ami-hvm-2014.09.1.x86_64-ebs (ami-4b6f650e)
Subnet ID	subnet-da03d1bf	Platform	-
Network interfaces	eth0	IAM role	-

To make this instance publicly accessible, we need to assign the server a public Elastic IP address. In the **EC2** console, click on the **Elastic IPs** link. You will see that the VPC Wizard already created an EIP and assigned it to your NAT Instance. Click on the **Allocate New Address** button.

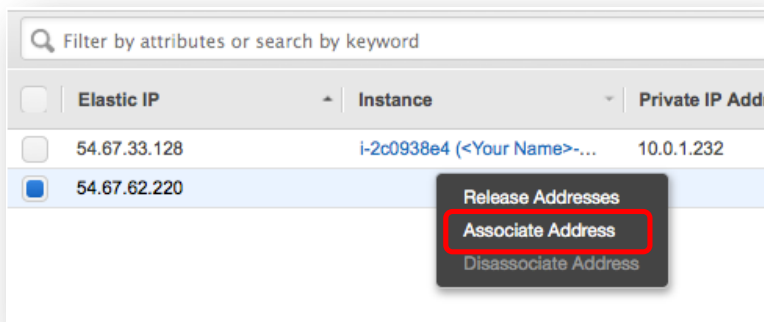
VPC Hands-On Lab

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The 'Elastic IPs' option under NETWORK & SECURITY is selected. The main content area has a header with buttons: 'Allocate New Address' (highlighted with a red box), 'Release Addresses', 'Associate Address', and 'Disassociate Address'. Below the header is a table with columns: Address, Instance, Private IP Address, Scope, and Public DNS. One address is listed: 54.183.17.243, associated with instance i-a2d087fe (NAT Instance), private IP 10.0.0.12, scope vpc, and public DNS ec2-54-183-17-243.us-west-1.compute.amazonaws.com. Below the table, details for the selected address are shown, including Public IP, Instance, Scope, Public DNS, Network interface ID, Private IP address, Network interface owner, and Allocation ID.

Click **Yes, Allocate**.



Next right-click on the new EIP that was allocated and select **Associate Address**.



Select your **Public Server** from the Instance dropdown and click **Associate**.

Associate Address

Select the instance OR network interface to which you wish to associate this IP address (54.67.62.220)

Instance

Search instance ID or Name tag

I-e26d5d2a (Public Server) (running)

I-5e6e5e96 (Private Server) (running)

I-2c0938e4 (<Your Name>-NAT Instance) (running)

Network Interface

Private IP Address

☐ Reassociation

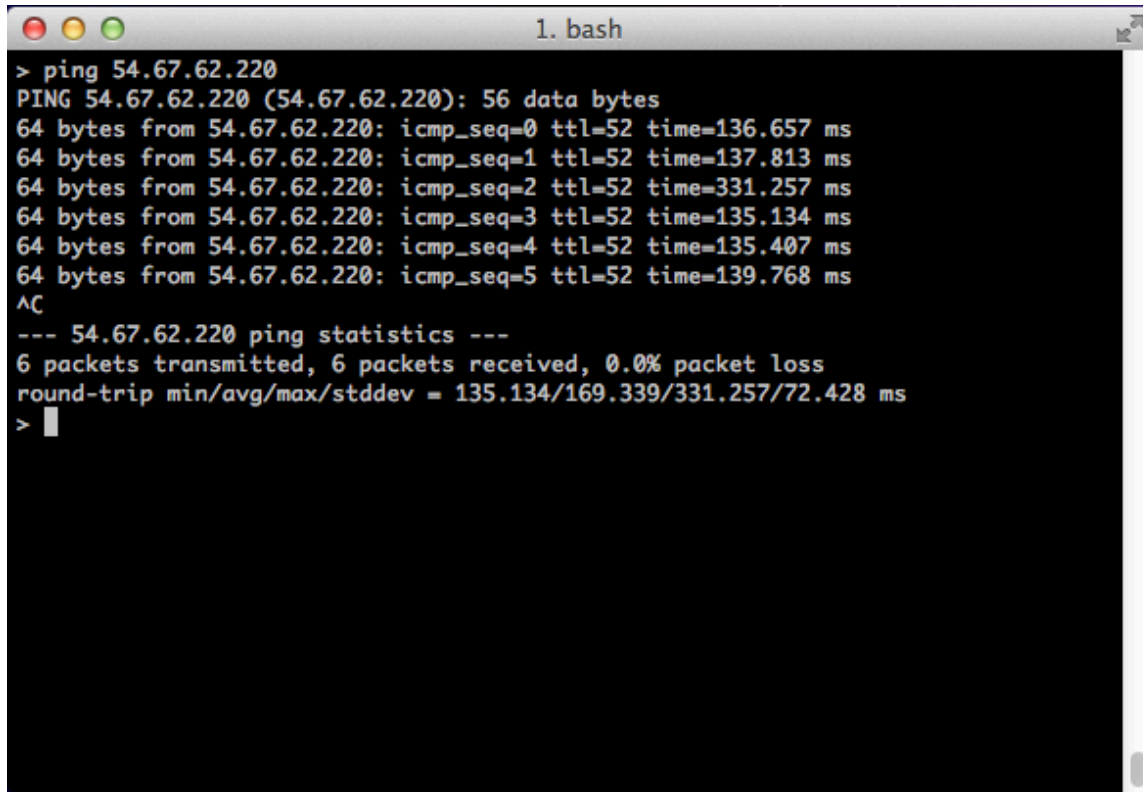
Warning

If you associate an Elastic IP address with your instance, your current public IP address is released. Learn more about [public IP addresses](#).

Cancel

Associate

You should now be able to connect to your public server using its new Elastic IP address. In the example screenshot below, we demonstrate this connectivity by simply “pinging” the server.



```
1. bash
> ping 54.67.62.220
PING 54.67.62.220 (54.67.62.220): 56 data bytes
64 bytes from 54.67.62.220: icmp_seq=0 ttl=52 time=136.657 ms
64 bytes from 54.67.62.220: icmp_seq=1 ttl=52 time=137.813 ms
64 bytes from 54.67.62.220: icmp_seq=2 ttl=52 time=331.257 ms
64 bytes from 54.67.62.220: icmp_seq=3 ttl=52 time=135.134 ms
64 bytes from 54.67.62.220: icmp_seq=4 ttl=52 time=135.407 ms
64 bytes from 54.67.62.220: icmp_seq=5 ttl=52 time=139.768 ms
^C
--- 54.67.62.220 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 135.134/169.339/331.257/72.428 ms
>
```

You have now successfully created public and private servers in a VPC. Feel free to explore the instance details for both instances to see the EIP assignment to your public server and examine the differences between the two instances.

Terminate Billable Services

You will not be able to delete your VPC until all instances using the VPC have been terminated. At this point feel free to terminate the Public and Private Servers that we created in this lab.

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm St
Public Server	i-e26d5d2a	t2.micro		running	2/2 checks ...	None
Private Server	i-5e6e5e96	t2.micro		running	2/2 checks ...	None
<Your Name>-NAT Instance	i-2c0938e4	m1.small		running	2/2 checks ...	None

Instances: i-e26d5d2a (Public Server), i-5e6e5e96 (Private Server)

Description

Status Checks

Monitoring

Tags

i-e26d5d2a: ec2-54-67-62-220.us-west-1.compute.amazonaws.com

i-5e6e5e96:

Connect

Get Windows Password

Launch More Like This

Instance State

Instance Settings

Image

Networking

CloudWatch Monitoring

Start

Stop

Reboot

Terminate

Check the box to Release the EIP along with instance termination so that you don't incur Idle EIP charges and click **Yes, Terminate**.

Terminate Instances

Warning

On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

i-e26d5d2a (Public Server, ec2-54-67-62-220.us-west-1.compute.amazonaws.com)

i-5e6e5e96 (Private Server)

Clean up associated resources

Associated resources may incur costs after these instances are terminated.

Release attached Elastic IPs

Elastic IPs which are not associated with an instance will incur an hourly cost. [Amazon EC2 Pricing](#)

☒ Release Elastic IPs (54.67.62.220)

Note: These Elastic IPs will no longer be associated with your account.

Cancel

Yes, Terminate

Finally, to completely delete the VPC, first terminate the NAT instance and release its EIP, then go to the **VPC** console, Click on Your **VPCs** link, select your VPC, and click on the **Delete** button.

VPC Hands-On Lab

VPC Dashboard

Filter by VPC: None

Virtual Private Cloud

- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- DHCP Options Sets

Create VPC Actions

Search VPC

Delete VPC

Edit DHCP Options Set

Edit DNS Resolution

Edit DNS Hostnames

Name	State	VPC CIDR
<Your Name>	available	10.0.0.0/16
Default	available	172.31.0.0/16

vpc-85619ae0 (10.0.0.0/16) | <Your Name>