

Creating a Virtual Private Cloud

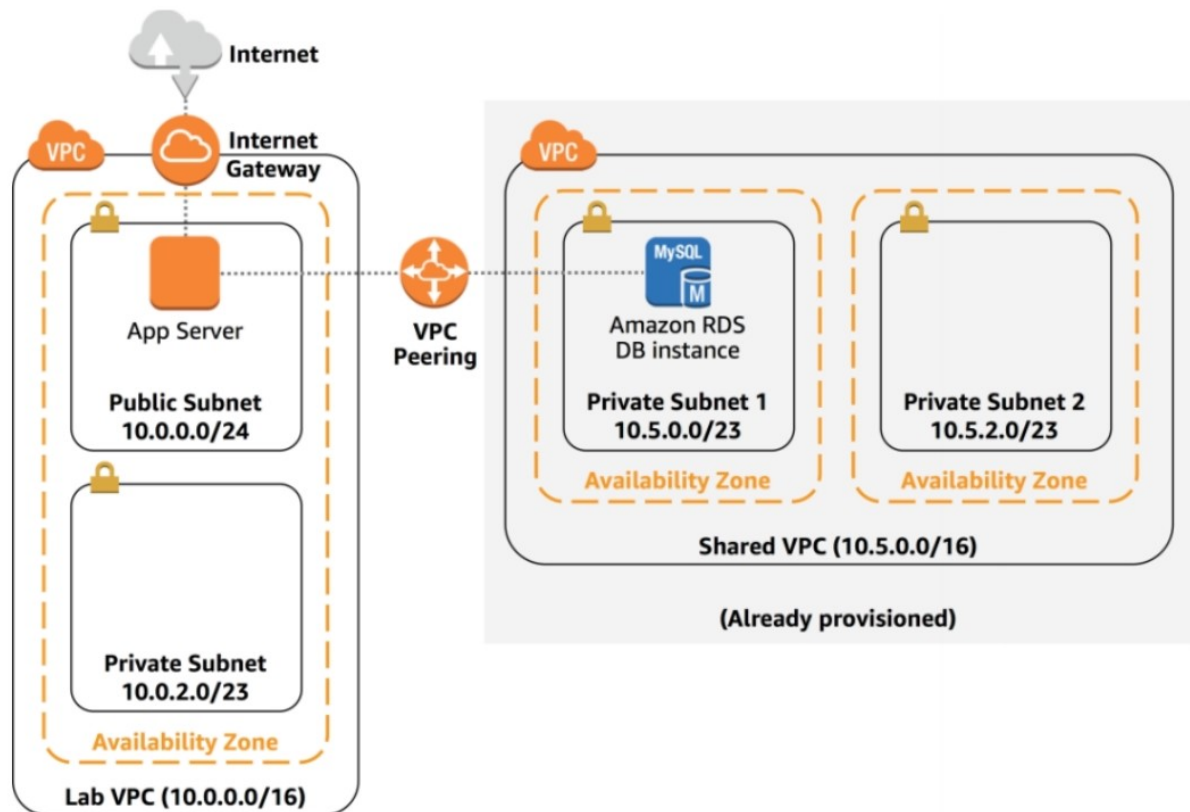
Traditional networking is hard — it involves equipment, cabling, complex configurations and specialist skills. Fortunately, Amazon VPC hides the complexity while making it easy to deploy secure private networks.

This lab shows you how to build your own Virtual Private Network, deploy resources and create private peering connections between networks.

You will deploy an Amazon VPC with:

- An Internet Gateway
- A Public Subnet
- A Private Subnet
- An application server to test the VPC

The final architecture will be:




If you have time, an **optional Challenge section** will then have you create a **VPC Peering Connection** to a *Shared Services VPC*. You will use an application and database to test connectivity between the VPCs.


Duration

This lab will require approximately **30 minutes** to complete.

Accessing the AWS Management Console

 **Windows Users:** Please use Chrome or Firefox as your web browser for this lab. The lab instructions are **not compatible with Internet Explorer** due to a difference in the Amazon RDS console.

Sign to the AWS Management Console using your credentials.

 Please do not change the Region during this lab.

Task 1: Create a VPC

You will begin by creating a new **Amazon Virtual Private Cloud (VPC)**.

A VPC is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch AWS resources, such as Amazon EC2 instances, into the VPC. You can configure the VPC by modifying its IP address range, create subnets, and configure route tables, network gateways, and security settings.

1. In the **AWS Management Console**, on the **Services** menu, click **VPC**.

The VPC console offers a Wizard that can automatically create several VPC architectures. However, in this lab you will be creating the VPC components manually.

2. In the left navigation pane, click **Your VPCs**.

A default VPC is provided so that you can launch resources as soon as you start using AWS. There is also a **Shared VPC** that you will use later in the lab. However, you will now create your own Lab VPC.

The VPC will have a CIDR range of **10.0.0.0/16**, which includes all IP address that start with **10.0.x.x** — containing over 65,000 addresses. You will later divide the addresses into separate subnets.

3. Click **Create VPC** and configure:

- **Name tag:** Lab VPC
- **IPv4 CIDR block:** 10.0.0.0/16
- Click **Create** then click **Close**

If these options do not appear, cancel and make sure you clicked **Your VPCs** in the left navigation pane. Then **Create VPC** again.

4. Select ☒ **Lab VPC**, ensuring that it is the only VPC selected.

5. Click the **Tags tab** in the lower half of the page.

Tags are useful for identifying resources. For example, a tag can be used to identify Dev/Test/Production environments or cost centers.

6. Click **Actions** and select **Edit DNS hostnames**. This option assigns a friendly DNS name to Amazon EC2 instances in the VPC, such as: *ec2-52-42-133-255.us-west-2.compute.amazonaws.com*

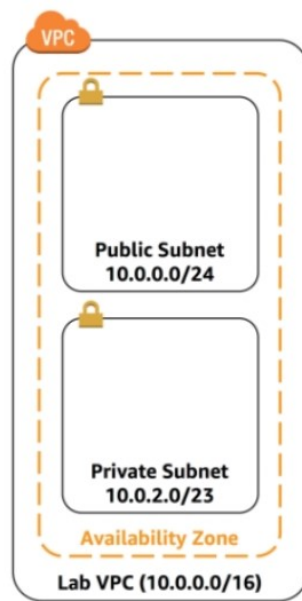
7. Select ☒ **enable**, then click **Save** and then click **Close**

Any Amazon EC2 instances launched into the VPC will now automatically receive a DNS hostname. You can also add a more meaningful DNS Name (eg app.company.com) later by using Amazon Route 53.

Task 2: Create Subnets

A subnet is a sub-range of IP addresses within the VPC. AWS resources can be launched into a specified subnet. Use a public subnet for resources that must be connected to the internet, and use a private subnet for resources that are to remain isolated from the internet.

In this task, you will create a public subnet and a private subnet:



Create a Public Subnet

The public subnet will be used for internet-facing resources.

8. In the left navigation pane, click **Subnets**.

9. Click **Create subnet** and configure:

- **Name tag:** Public Subnet
- **VPC:** Lab VPC
- **Availability Zone:** Select the first AZ in the list
- **IPv4 CIDR block:** 10.0.0.0/24
- Click **Create** then click **Close**

The VPC has a CIDR of **10.0.0.0/16**, which includes all **10.0.x.x** IP addresses. The subnet you just created has a CIDR of **10.0.0.0/24**, which includes all **10.0.0.x** IP addresses. They might look similar, but the subnet is smaller than the VPC due to the **/24** in the CIDR range.

You will now configure the subnet to automatically assign a public IP address for all instances launched within it.

10. Select ☒ **Public Subnet**.

11. Click **Actions** and select **Modify auto-assign IP settings**, then:

- Select ☒ **Auto-assign IPv4**
- Click **Save**

Even though this subnet has been named Public Subnet, it is not yet public. A public subnet must have an internet gateway, which you will attach in the next task.

Create a Private Subnet

The private subnet will be used for resources that are to remain isolated from the internet. 12. Use what you have just learned to create another Subnet with:

- **Name tag:** Private Subnet
- **VPC:** Lab VPC
- **Availability Zone:** Select the first AZ in the list
- **IPv4 CIDR block:** 10.0.2.0/23

The CIDR block of **10.0.2.0/23** includes all IP addresses that start with **10.0.2.x** and **10.0.3.x**. This is twice as large as the public subnet because most resources should be kept private, unless they specifically need to be accessible from the internet.

Your VPC now has two subnets. However, it is totally isolated and cannot communicate with resources outside the VPC. You will next configure the Public Subnet to connect to the internet via an Internet Gateway.

Task 3: Create an Internet Gateway

An **internet gateway** is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in a VPC and the internet. It imposes no availability risks or bandwidth constraints on network traffic.

An internet gateway serves two purposes:

- To provide a target in route tables to connect to the internet
- To perform network address translation (NAT) for instances that have been assigned public IPv4 addresses

In this task, you will create an internet gateway so that internet traffic can access the public subnet.

13. In the left navigation pane, click **Internet Gateways**.

14. Click **Create internet gateway** and configure:

- **Name tag:** Lab IGW
- Click **Create** then click **Close**

You can now attach the internet gateway to your Lab VPC.

15. Select ☒ **Lab IGW**, ensuring that it is the only gateway selected.

16. Click **Actions** then **Attach to VPC** and configure:

- **VPC:** Select Lab VPC from the list
- Click **Attach**

This will attach the internet gateway to your Lab VPC. Even though you have created an internet gateway and attached it to your VPC, you must also configure the public subnet route table to use the internet gateway.

Task 4: Configure Route Tables

A **route table** contains a set of rules, called routes, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

To use an internet gateway, a subnet's route table must contain a route that directs internet-bound traffic to the internet gateway. If a subnet is associated with a route table that has a route to an internet gateway, it is known as a public subnet.

In this task, you will:

- Create a public route table for internet-bound traffic
- Add a route to the route table to direct internet-bound traffic to the internet gateway
- Associate the public subnet with the new route table

17. In the left navigation pane, click Route Tables.

Several route tables will be displayed, but there is only one route table associated with Lab VPC. This route table routes traffic locally, so it is called a Private Route Table.

18. Select ☒ the route table that shows Lab VPC in the VPC ID column.

19. Click in the Name column and enter a name of: Private Route Table then click .

20. Click the Routes tab in the lower half of the page.

There is only one route. It shows that all traffic destined for 10.0.0.0/16 (which is the range of the Lab VPC) will be routed locally. This allows all subnets within a VPC to communicate with each other.

You will now create a new Public Route Table to send public traffic to the internet gateway.

21. Click Create Route Table and configure:

- **Name tag:** Public route table
- **VPC:** Lab VPC
- Click **Create** then click **Close**

22. Select ☒ **Public Route Table**, ensuring that it is the only route table selected.
23. In the **Routes** tab, click **Edit routes**. You will now add a route to direct internet-bound traffic (0.0.0.0/0) to the internet gateway.
24. Click **Add route** then configure:
 - Click **Save routes** then click **Close**
 - **Destination:** 0.0.0.0/0
 - **Target:** Select Internet Gateway then select Lab IGW from the list
 - Click **Save routes** then click **Close**

The last step is to associate this new Route Table with the Public Subnet.

25. Click the **Subnet Associations** tab.
26. Click **Edit subnet associations**
27. Select ☒ the row with **Public Subnet**.
28. Click **Save**

The Public Subnet is now public because it has a route table entry that sends traffic to the internet via the internet gateway.

To summarize, you can create a public subnet as follows:

- Create an internet gateway
- Create a route table
- Add a route to the route table that directs 0.0.0.0/0 traffic to the internet gateway
- Associate the route table with a subnet, which therefore becomes a public subnet

Task 5: Create a Security Group for the App Server

A security group acts as a virtual firewall for instances to control inbound and outbound traffic. Security groups operate at the instance network interface level, not the subnet level. Therefore, each instance can have its own firewall that controls traffic. If you do not specify a particular security group at launch time, the instance is automatically assigned to the default security group for the VPC.

In this task, you will create a security group that allows users to access your app server via HTTP.

29. In the left navigation pane, click Security Groups.
30. Click Create security group and configure:

- Security group name: App-SG
- Description: Allow HTTP traffic
- VPC: Lab VPC

- Click Create then click Close

31. Select ☒ App-SG.

32. Click the Inbound Rules tab.

The Inbound Rules determine what traffic is permitted to reach the instance. You will configure it to permit HTTP (port 80) traffic coming from anywhere on the internet (0.0.0.0/0).

33. Click Edit rules

34. Click Add Rule then configure:

- Type: HTTP
- Source: Anywhere
- Description: Allow web access
- Click Save rules then click Close You will use this App-SG in the next task.

Task 6: Launch an App Server in the Public Subnet

To test that your VPC is correctly configured, you will now launch an Amazon EC2 instance into the Public Subnet and confirm that it is accessible from the internet.

35. On the Services menu, click EC2.

36. Click Launch Instance and configure:

- Step 1 (Choose AMI): o AMI: Amazon Linux 2
- Step 2 (Choose Instance Type):
 - o **Instance Type:** t2.micro
- Step 3 (Configure Instance Details):
 - o **Network:** Lab VPC
 - o **Subnet:** Public Subnet
 - o **IAM role:** Inventory-App-Role
 - o **User Data (Under ☒ Advanced Details):**

```
#!/bin/bash
# Install Apache Web Server and PHP
yum install -y httpd mysql
amazon-linux-extras install -y php7.2
# Download Lab files
wget https://us-west-2-tcprod.s3.amazonaws.com/courses/ILT-TF-100-ARCHIT/v6.4.0/lab-2-webapp/scripts/inventory-app.zip
unzip inventory-app.zip -d /var/www/html/
# Download and install the AWS SDK for PHP
```

```
wget https://github.com/aws/aws-sdk-php/releases/download/3.62.3/aws.zip
unzip aws -d /var/www/html
# Turn on web server
chkconfig httpd on
service httpd start
```

- Step 4 (Add Storage):
 - Use default settings (no changes)
- Step 5 (Add Tags):
 - Click Add Tag
 - Key: Name
 - Value: App Server
- Step 6 (Configure Security Group)
 - **Select an existing security group:** App-SG

You will receive a warning that You will not be able to connect to the instance. This is acceptable because you will not be connecting to the instance. All configuration is done via the User Data script.

- Step 7 (Review):
 - **Launch**

37. In the Select an existing key pair or create a new key pair window:

- Click ☒ I acknowledge that...
- Click Launch Instances A status page notifies you that your instances are launching.

38. Click View Instances

39. Wait for the App Server to fully launch. It should display the following:

- **Instance State:** ● **running**


You can click refresh occasionally to update the display.

40. Select ☒ **App Server**.

41. Copy the **IPv4 Public IP** address shown in the **Description** tab.

42. Open a new web browser tab with that IP address.

If your VPC was configured correctly, you should see the Inventory application and the message "Please configure settings to connect to database". No database settings have been configured yet, but the appearance of the Inventory application proves that the Public Subnet has been correctly configured.

 If the Inventory application does not appear, wait 60 seconds and refresh the page to try again. It can take a couple of minutes for the EC2 instance to boot and run the script that installs software.

Lab Complete

Congratulations! You have completed the lab. Click End Lab at the top of this page to clean up your lab environment.