Module 6 - Guided Lab: Creating a Virtual Private Cloud

Lab overview and objectives

Traditional networking is difficult. It involves equipment, cabling, complex configurations, and specialist skills. Amazon Virtual Private Cloud (Amazon VPC) hides the complexity, and simplifies the deployment of secure private networks.

This lab shows you how to build your own virtual private cloud (VPC), deploy resources, and create private peering connections between VPCs.

After completing this lab, you should be able to:

- Deploy a VPC
- Create an internet gateway and attach it to the VPC
- Create a public subnet
- Create private subnet
- Create an application server to test the VPC

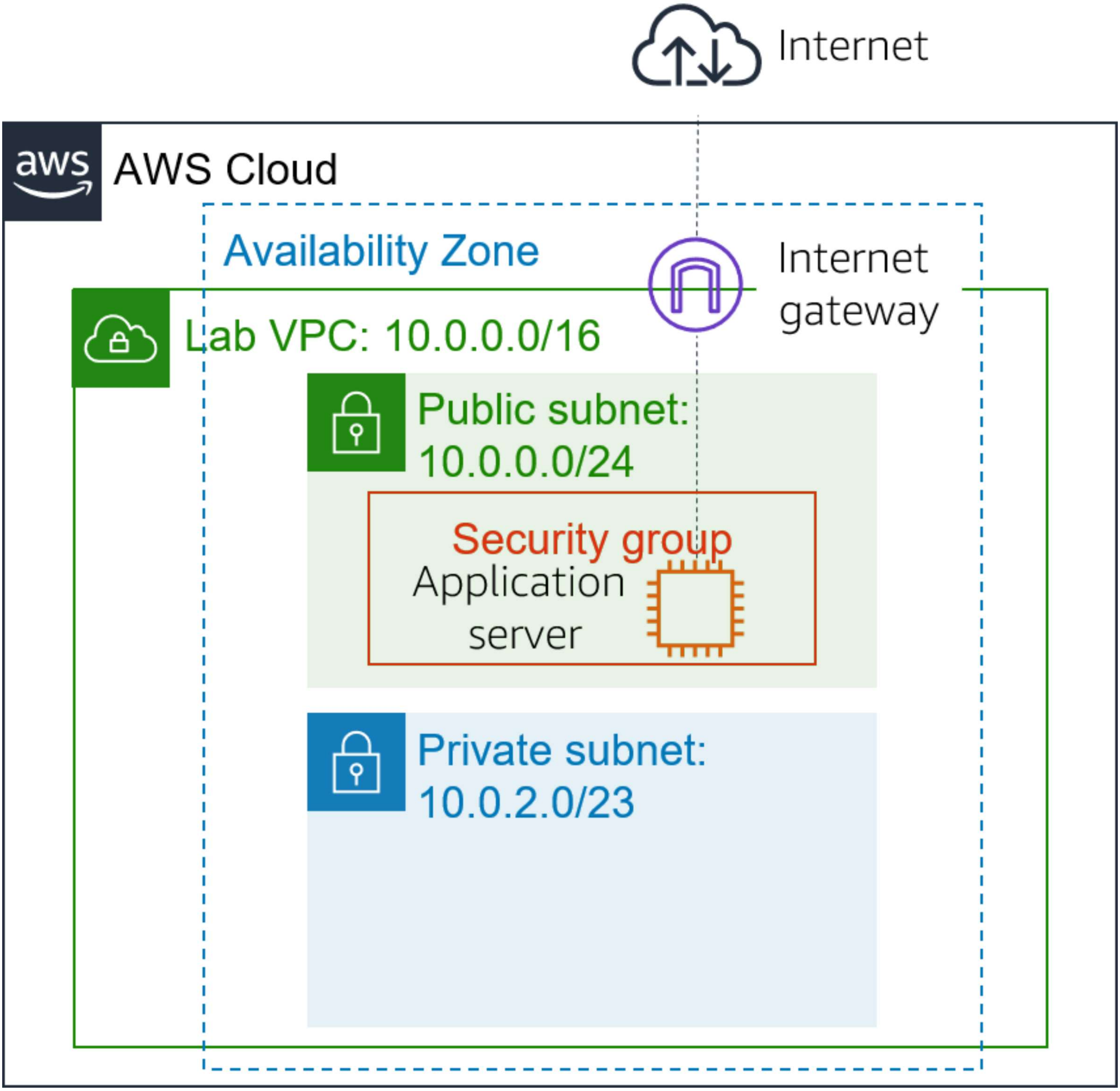
Duration

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

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

At the **end** of this lab, your architecture will look like the following example:



Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.

   A **Start Lab** panel opens, and it displays the lab status.

   Tip: If you need more time to complete the lab, restart the timer for the environment by choosing the Start Lab button again.

2. Wait until the **Start Lab** panel displays the message *Lab status: ready*, then close the panel by choosing the **X**.

3. At the top of these instructions, choose AWS.

This action opens the AWS Management Console in a new browser tab. The system automatically logs you in.

⚠ **Tip**: If a new browser tab does not open, a banner or icon is usually at the top of your browser with the message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and then choose **Allow pop-ups**.

4. Arrange the **AWS Management Console** tab so that it displays alongside these instructions. Ideally, you will have both browser tabs open at the same time so that you can follow the lab steps more easily.

   ⚠ **Do not change the Region unless specifically instructed to do so**.

Task 1: Creating a VPC

You will begin by using Amazon VPC to create a new **virtual private cloud, or VPC**.

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

5. In the search box to the right of ▦ **Services**, search for and choose **VPC** to open the VPC console.

   The VPC console provides a wizard that can automatically create several VPC architectures. However, in this lab, you will create the VPC components manually.

6. In the left navigation pane, choose **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 Classless Inter-Domain Routing (CIDR) range of **10.0.0.0/16**, which includes all IP address that start with **10.0.x.x**. It contains over 65,000 addresses. You will later divide the addresses into separate *subnets*.

7. Choose Create VPC and configure these settings:

   - **Name tag:** `Lab VPC`
   - **IPv4 CIDR block:** `10.0.0.0/16`
   - Choose Create VPC

     A message that you successfully created the VPC appears.

8. In the lower half of the page, choose the **Tags** tab.

   Tags are useful for identifying resources. For example, you can use a tag to identify cost centers or different environments (such as development, test, or production).

9. Choose Actions⌄ and select **Edit VPC settings**.

   This option assigns a *friendly* Domain Name System (DNS) name to EC2 instances in the VPC, such as:

   ec2-52-42-133-255.us-west-2.compute.amazonaws.com

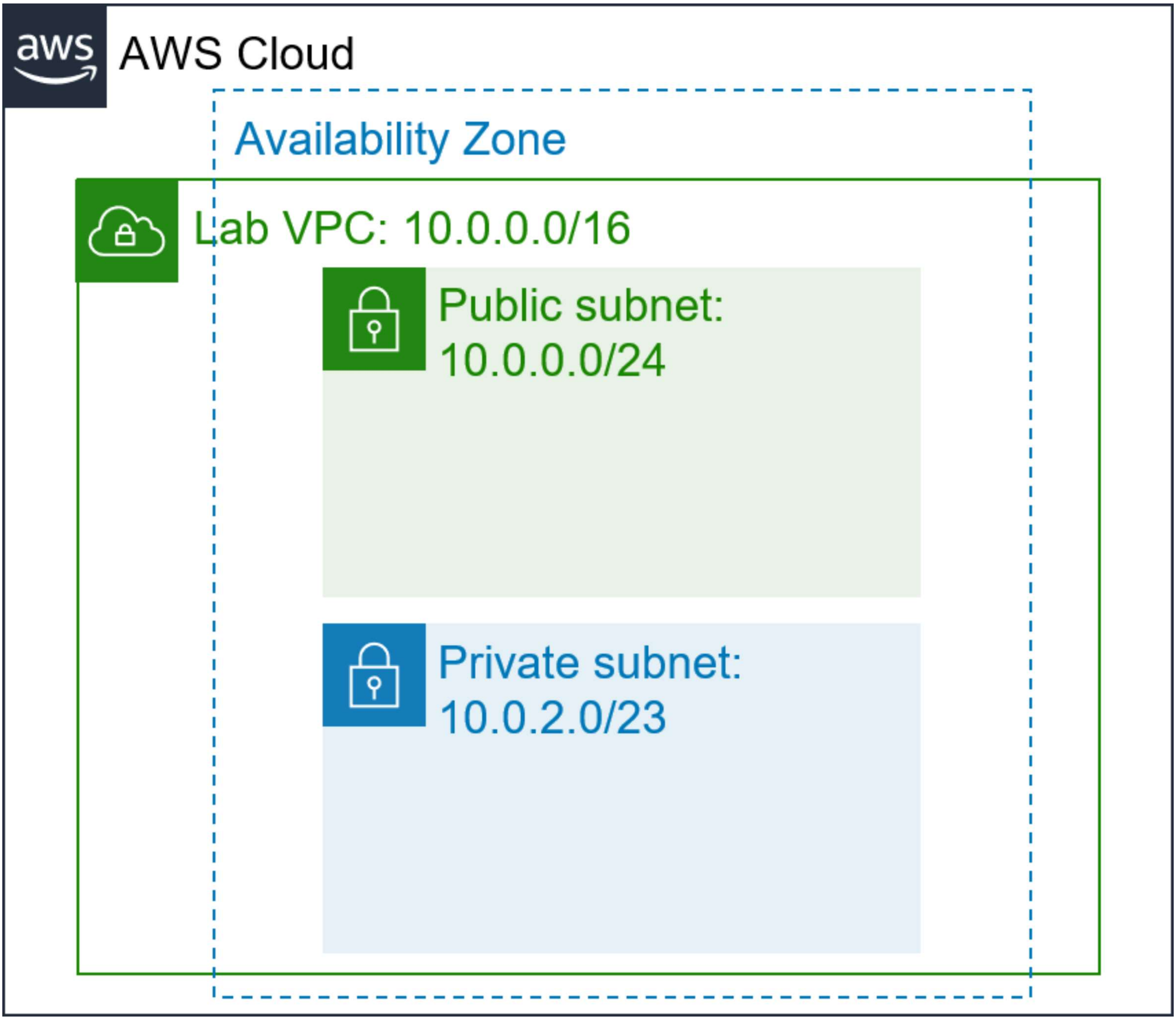
10. Select ☑**Enable DNS hostname** and then choose Save

Any EC2 instances that are launched into the VPC will now automatically receive a DNS hostname. You can also add a more meaningful DNS name (such as *app.example.com*) later by using Amazon Route 53.

Task 2: Creating subnets

A subnet is a subrange of IP addresses in 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 must remain isolated from the internet.

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



Creating a public subnet

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

11. In the left navigation pane, choose **Subnets**.

12. Choose Create subnet and configure these settings:
    - **VPC ID:** *Lab VPC*

  - **Subnet name:** `Public Subnet`
  - **Availability Zone:** Select the *first* Availability Zone in the list (do *not* keep the No Preference default)
  - **IPv4 CIDR block:** `10.0.0.0/24`
  - Choose **Create subnet**

    💬 The VPC has a CIDR block of *10.0.0.0/16*, which includes all *10.0.x.x* IP addresses. The subnet you just created has a CIDR block 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 because of the */24* in the CIDR range.

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

13. Select ☑ **Public Subnet**.

14. Choose **Actions⌄** and select **Edit subnet settings**, then:
    - Select ☑**Enable auto-assign public IPv4 address**
    - Choose **Save**

      💬Though this subnet is named *Public Subnet*, it is not yet public. A public subnet must have an internet gateway, which you attach in the next task.

Creating a private subnet

The private subnet will be used for resources that must remain isolated from the internet.

15. Use what you just learned to create another subnet with these settings:
    - **VPC ID:** *Lab VPC*
    - **Subnet name:** `Private Subnet`
    - **Availability Zone:** Select the *first* Availability Zone in the list (do *not* keep the No Preference default)
    - **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 must be accessible from the internet.

      Your VPC now has two subnets. However, the public subnet 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: Creating an internet gateway

An *internet gateway* is a horizontally scaled, redundant, and highly available VPC component. It allows communication between the 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 that connects to the internet
- To perform network address translation (NAT) for instances that were assigned public IPv4 addresses

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

16. In the left navigation pane, choose **Internet Gateways**.

17. Choose Create internet gateway and configure these settings:

    - **Name tag:** `Lab IGW`
    - Choose Create internet gateway

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

18. Choose Actions⌄ then **Attach to VPC**, and configure these settings:

    - **Available VPCs:** Place you cursor in the search box, then select *Lab VPC*
    - Choose Attach internet gateway

      This action will attach the internet gateway to your *Lab VPC*. Though you created an internet gateway and attached it to your VPC, you must also configure the public subnet *route table* so it uses the internet gateway.

Task 4: Configuring 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 because 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

19. In the left navigation pane, choose **Route Tables**.

    Several route tables are 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*.

20. Scroll to the right so that you can see the **VPC** column, then expand the width of the column so that you can see which one is used by **Lab VPC**.

21. Scroll back to the left and select ☑ the route table that shows **Lab VPC**.

22. In the **Name** column, choose ✎ then enter the name `Private Route Table` and choose ✔.

23. In the lower half of the page, choose the **Routes** tab.

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

    You will now create a new public route table to send public traffic to the internet gateway.

24. Choose Create route table and configure these settings:

    - **Name:** `Public Route Table`

- **VPC:** *Lab VPC*
- Choose **Create route table**

25. In the **Routes** tab, choose **Edit routes**

    You will now add a route to direct internet-bound traffic (*0.0.0.0/0*) to the internet gateway.

26. Choose **Add route** then configure these settings:

    - **Destination:** `0.0.0.0/0`
    - **Target:** Select *Internet Gateway* and then, from the list, select *Lab IGW*
    - Choose **Save changes**

    The last step is to *associate* this new route table with the public subnet.

27. Choose the **Subnet associations** tab.

28. Choose **Edit subnet associations**

29. Select ☑ the row with **Public Subnet**.

30. Choose **Save associations**

    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 by following these steps:

    - 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 thus becomes a *public subnet*

Task 5: Creating a security group for the application server

A *security group* acts as a virtual firewall for instances to control inbound and outbound traffic. Security groups operate at the level of the *elastic network interface for the instance*. Security groups do not operate at the *subnet* level. Thus, 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 application server via HTTP.

31. In the left navigation pane, choose **Security Groups**.

32. Choose **Create security group** and configure these settings:

    - **Security group name:** `App-SG`
    - **Description:** `Allow HTTP traffic`
    - **VPC:** select the X to clear the default selection, then choose *Lab VPC*
    - Scroll to the bottom and choose **Create security group**

33. Verify the **Inbound rules** tab is selected below.

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

34. Choose **Edit inbound rules**

35. Choose **Add rule** and then configure these settings:

    - **Type:** *HTTP*
    - **Source type:** *Anywhere-IPv4*
    - **Description:** `Allow web access`
    - Choose **Save rules**

    You use this *App-SG* in the next task.

Task 6: Launching an application server in the public subnet

To test that your VPC is correctly configured, you will now launch an EC2 instance into the public subnet. You will also confirm that you can access the EC2 instance from the internet.

36. In the search box to the right of ▦ **Services**, search for and choose **EC2** to open the EC2 console.

37. From the **Launch instance** menu, choose **Launch Instance**. Configure these options:

    - **Name**: `App Server`
    - In the list of available *Quick Start* AMIs, keep the default **Amazon Linux** selected. Also keep the specific default **Amazon Linux 2023 AMI** selected.
    - In the *Instance type* panel, keep the default **t2.micro** selected.
    - From the **Key pair name** menu, select **vockey**.
    - Next to Network settings, choose **Edit**, then configure:
      - **Network:** *Lab VPC*
      - **Subnet:** *Public Subnet*
    - Under Firewall (security groups), choose ● **Select an existing security group**.
      - For **Common security groups**, select **App-SG**.
    - In the *Configure storage* section, keep the default settings.
    - Expand the **Advanced details** panel.
    - **IAM instance profile:** *Inventory-App-Role*
    - For the **Metadata version** set to **V1 and V2 (token optional)**.
    - Scroll to the bottom of the page and then copy and paste the code shown below into the **User data** box:

```
#!/bin/bash
# Install Apache Web Server and PHP
dnf install -y httpd wget php-fpm php-mysqli php-json php php-devel
dnf install -y mariadb105-server
# Download Lab files
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod6-
guided/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/3.298.5/aws.zip
unzip -o aws.zip
unzip -o aws.zip -d /var/www/html/
# Turn on web server
systemctl enable httpd
systemctl start httpd
```

- At the bottom of the **Summary** panel on the right side of the screen choose Launch instance

  You will see a Success message.

38. Choose View all instances

39. Wait until the App Server instance shows *2/2 checks passed* in the **Status check** column.

    💬 This may take a few minutes. Choose the refresh ⟳ icon at the top of the page every 30 seconds or so to more quickly become aware of the latest status of the instance.

40. Select ☑ **App Server**.

41. Copy the **Public IPv4 DNS** value shown in the **Details** tab at the bottom of the page.

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

    If you configured the VPC correctly, the Inventory application and this message should appear: *Please configure Settings to connect to database*. You have not configured any database settings yet, but the appearance of the Inventory application demonstrates that the public subnet was 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 the software.

Submitting your work

43. At the top of these instructions, choose Submit to record your progress and when prompted, choose Yes .

44. If the results don't display after a minute, return to the top of these instructions and choose Grades

    Tip: You can submit your work multiple times. After you change your work, choose **Submit** again. Your last submission is what will be recorded for this lab.

45. To find detailed feedback on your work, choose Details followed by ▸ **View Submission Report**.

Lab complete 🎓

🏁 Congratulations! You have completed the lab.

46. Choose End Lab at the top of this page, and then select Yes to confirm that you want to end the lab.

    A panel indicates that *DELETE has been initiated... You may close this message box now.*

47. Select the **X** in the top right corner to close the panel.