Creating a VPC and Launching a Web Application in an Amazon EC2 Instance

TODO

- Create a new Amazon VPC with two public subnets
- Create an Internet Gateway
- Create a Route Table with a Public Route to the Internet
- Create a Security Group
- Launch an Amazon Elastic Compute Cloud (Amazon EC2) instance
- Configure an EC2 instance to host a web application using a user data script

Objective

Create the underlying network architecture needed to run a web application on an Amazon EC2 instance. After building the network, launch the EC2 Instance used to host your web application and conduct a simple test to verify you can access it in a web browser.

Task 1: Create a Virtual Private Cloud

TASK 1.1: CREATE A VPC, PUBLIC SUBNETS, ROUTE TABLE, AND INTERNET GATEWAY

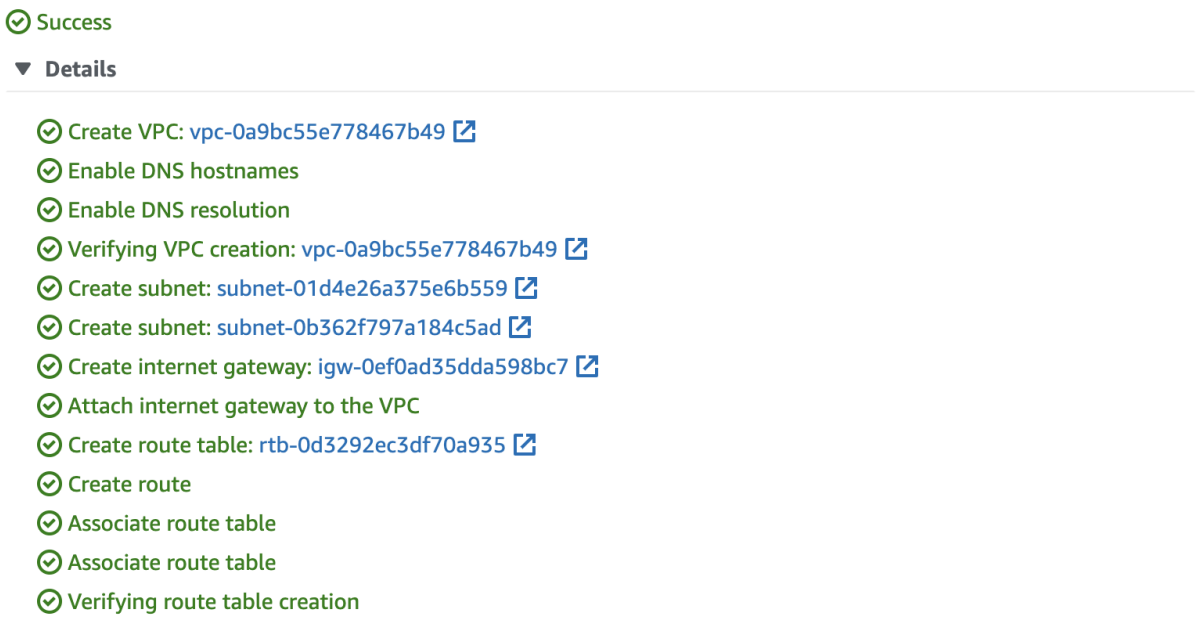
- At the top of the AWS Management Console, in the search bar, search for and choose `VPC`.
- Choose **Create VPC**.
- On the left side of the Create VPC console, locate the VPC settings section, and select VPC and more in the Resources to create section.
- In the Name tag auto-generation section, select Auto-generate, if not already selected.
- In the text box under the Auto-generate option, enter the value `lab-2`.
- In the IPv4 CIDR block section, enter `10.0.0.0/16`.
- In the IPv6 CIDR block section, select the No IPv6 CIDR block option, if not already selected.
- Leave the Tenancy section set to Default.
- In the Number of Availability Zones (AZs) section, select `2`.

- In the Number of public subnets section, select `2`.
- In the Number of private subnets section, select `0`.
- Expand the Customize subnets CIDR blocks option.
- In the first text box, enter the following: `10.0.0.0/24`. This CIDR range contains all IP addresses starting with 10.0.0.x.
- In the second text box, enter the following: `10.0.1.0/24`. This CIDR range contains all IP addresses starting with 10.0.1.x.
- In the NAT gateways section, select `None`.
- In the VPC endpoints section, select `None`.
- In the DNS options section, select both Enable DNS hostnames and Enable DNS resolution.

Note: The DNS hostnames attribute determines whether instances launched in the VPC receive public DNS hostnames that correspond to their public IP addresses. The DNS resolution attribute determines whether DNS resolution through the Amazon DNS server is supported for the VPC.

- Review the Preview section on the right side of the screen. This preview should include the name of your VPC, the two subnets you created, a public route table, and an Internet gateway.
- If the Preview section presents the correct information, select **Create VPC**.

The Create VPC workflow screen appears. Wait for the workflow to finish. You should receive a message indicating everything created successfully.

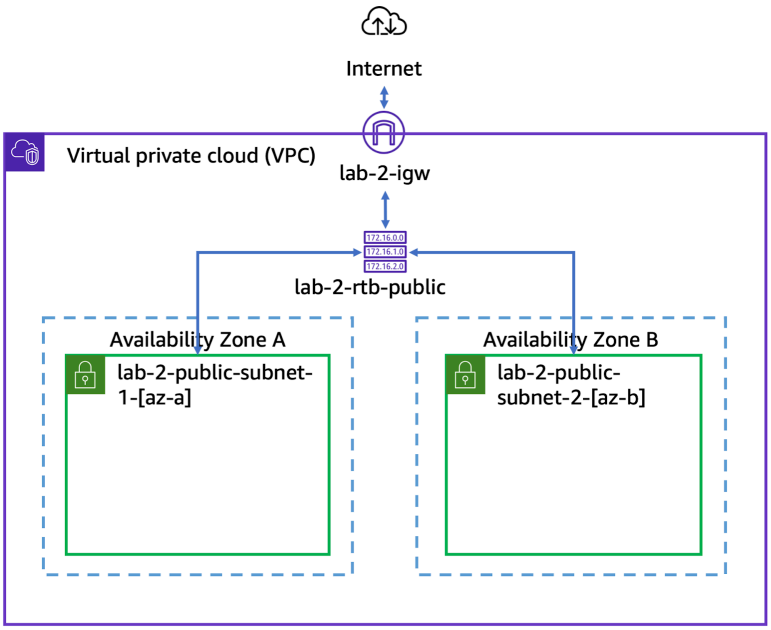


Select **View VPC**.

TASK 1.2: ADD A PUBLIC ROUTE TO THE ROUTE TABLE

- Choose Route tables. Refresh the selection. Select the Main route table for lab-2-vpc.
- There is currently only one route used for all internal traffic within the VPC. To access the Internet, you must first add a route to the Internet gateway.
- Select **Edit routes**
- Select **Add route**
- In the Destination column, locate the text box and enter the following: `0.0.0.0/0`.
- In the Target column, locate the text box and select it. A list of options should present themselves. Select Internet Gateway. The `id` and `name` of your Internet Gateway should populate. Select the Internet Gateway ID named `lab-2-igw`.
- Select **Save changes**.
- You should receive an output message similar to the following:
- Updated routes for rtb-xxxxxxxxxxxxxxx successfully.
- Navigate to the bottom of the VPC console and select the Routes tab, if not already selected.
- Note the two routes:
  - An internal route for VPC traffic
  - A public route to the Internet by way of the Internet Gateway

The diagram below represents the new custom VPC.



Task 2: Create a VPC Security Group

In this task, create a VPC security group. A security group controls the traffic that is allowed to reach and leave the resources that it is associated with. For example, after you associate a security group with an EC2 instance, it controls the inbound and outbound traffic for the instance, similar to a firewall.

- In the left navigation pane, locate the Security section, and choose Security groups.
- In the top-right area of your window, choose **Create security group**.
- In the Basic details section, locate the Security group name text box and enter the value `Web Security Group`.
- In the Basic details section, locate the Description text box and enter the value `Enable HTTP access from anywhere`.
- In the Basic details section, locate the VPC text box, select the X to remove the default VPC, and choose the VPC with name `lab-2-vpc`. Once select, the `lab-2-vpc` name will disappear and only the VPC ID remains. This is normal.

Next, add a rule to the security group to allow inbound web requests from anywhere.

- In the Inbound rules section, choose **Add rule**.
- In the Type column, select the dropdown and choose `HTTP`.
- In the Source column, select the dropdown and choose `Anywhere-IPv4`.
- In the Description column, locate the text box and enter `Allow web requests from anywhere`.
- Navigate to the bottom of the Create security group window, and choose **Create security group**.

You will use this security group in the next task when you launch an Amazon EC2 instance. Without this inbound rule, the EC2 instance would not receive inbound web requests.

Task 3: Launch Your Amazon EC2 Instance

In this task, create an EC2 instance and provide a bootstrap script to install and configure the requirements for your web application. You also enable SSH (Secure Shell) access to the instance.

- At the top of the AWS Management Console, in the search bar, search for and choose `EC2`.
- In the left navigation pane, in the Instances section, choose Instances.
- Choose **Launch instances**.
- In the Name and tags section, locate the Name text box, and enter the value `Web Application`.
- In the Application and OS images section, locate the Quick start window, and select Amazon Linux 2023.

Note: An Amazon Machine Image (AMI) is a supported and maintained image provided by AWS that provides the information required to launch an instance.

- Under the Quick start window, locate the Amazon Machine Image (AMI) dropdown and select the AMI named `Amazon Linux 2023 AMI`.
- Under the Description section, locate the Architecture dropdown and select `64-bit (x86)`.
- In the Instance type section, locate the Instance type dropdown and select `t3.micro`.
- In the Key pair (login) section, locate the key pair name dropdown and select Proceed without a key pair (Not recommended).

Note: You do not need SSH to access this EC2 instance during this lab.

- In the Network settings section, select **Edit**.
- Locate the VPC dropdown and select `lab-2-vpc`.
- Locate the Subnet dropdown and select `lab-2-subnet-public-1-[az-a]` where `[az-a]` is the first availability zone in your region.
- Locate the Auto-assign public IP dropdown and select `Enable`.

Note: This setting assigns a public IP address to the instance so you can access the application in your browser.

- In the Firewall (security groups) section, choose `Select an existing security group`.
- Locate the Common security groups dropdown and select the security group named `Web Security Group`.
- You can leave the Configure storage section set to the default values.
- Scroll down to the Advanced Details section, expand the dropdown, and locate the User data text box.

Note: When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.

Your instance uses an Amazon Linux 2023 operating system; therefore, you need to provide a *shell script* that will run when the instance starts. This script installs the required application dependencies and launches the application. Without this user data script, you would need to log in to the EC2 Instance and execute all of the commands yourself.

```
#!/bin/bash -ex

# Update yum
yum -y update

#Install nodejs
yum -y install nodejs

# Create a dedicated directory for the application
mkdir -p /var/app

# Get the app from S3
wget
https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-TECESS-5
/app/app.zip

# Extract it to the desired folder
unzip app.zip -d /var/app/
cd /var/app/

# Install dependencies
npm install

# Start the app
npm start
```

This script performs the following tasks:

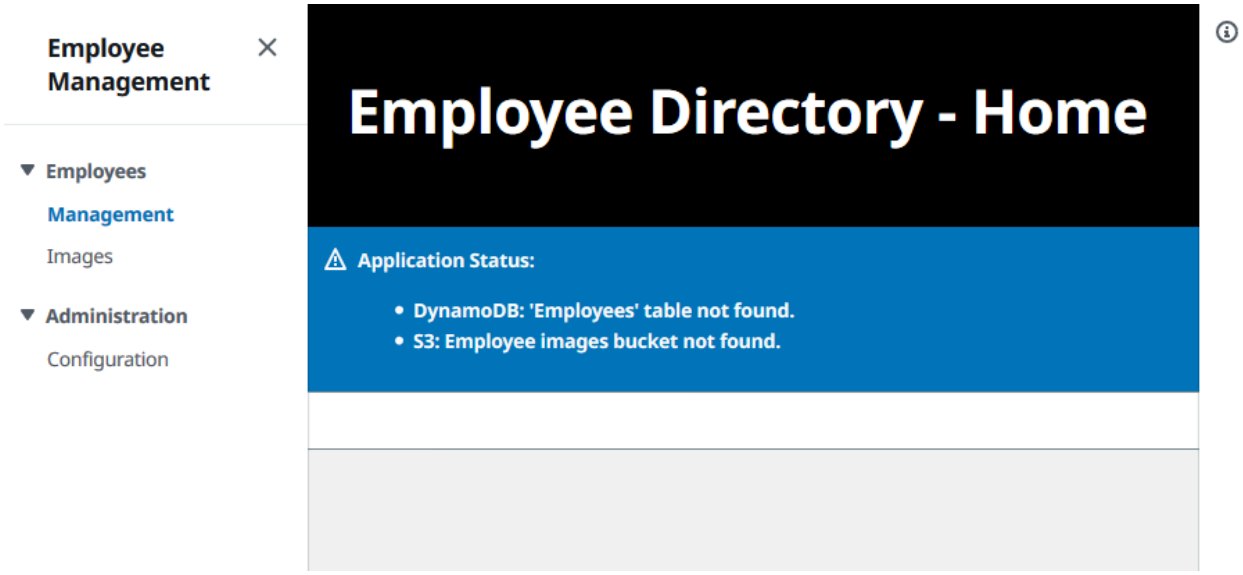
- Install system updates
- Install a source repository to download the Node.js installer
- Install Node.js

- ❖ Download the application code
- ❖ Create a dedicated directory for the web application
- ❖ Download and extract the application into the specified directory
- ❖ Install the application dependencies
- ❖ Set the listener port for the application
- ❖ Start the web application

- In Summary section, for Number of instances text box, enter the value `1`, if not already done.
- Select **Launch instance**.
- Wait for the instance to launch. You should receive an message indicating the launch completed. See the message below:

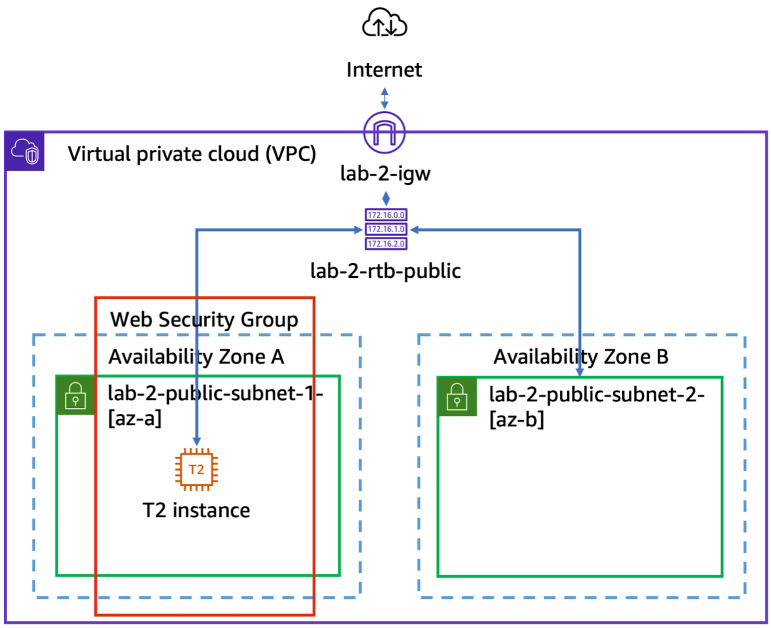
Successfully initiated launch of instance (instance-id)

- Choose **View all instances**.
- In the Details tab, locate the Public IPv4 address section, and copy the IP address.
- Note: If you refreshed the Instances window, you may need to choose your Web Application instance again.
- Open a new browser tab and paste the IP address you copied in the previous step. You should see the following application screen.



Note: If you use the open address link, your browser might try to browse to the application using `https://`, which won't work. The application can only be accessed using `http://` on port `80`.

The follow diagram represents the completed lab environment:



Once you access the web application successfully, you can close the browser tab.