Guided Lab: Setting up a Web server on an EC2 instance

Description

One of the common use cases of Amazon EC2 (Elastic Compute Cloud) is for hosting web servers. EC2 offers a range of instance types to match different needs. Some instances are optimized for compute, others for memory, storage, or other specific tasks. The `t2.micro` instance type, which we'll use in this hands-on lab, is typically chosen for low to moderate-traffic web servers and development environments due to its balance of cost and performance.

Objectives

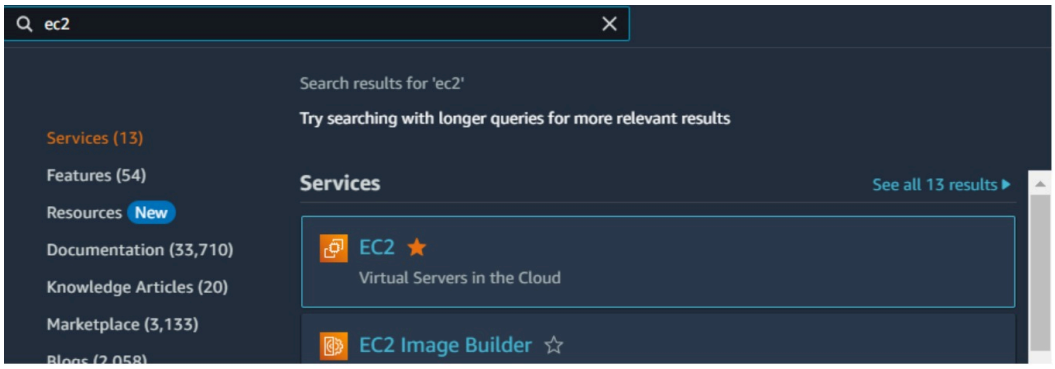
In this lab, you will:

1. Create a t2.micro EC2 instance and configure security groups for HTTP traffic.
2. Install and initialize the Nginx web server on the EC2 instance.
3. Deploy a basic HTML page and make it accessible through the EC2 instance's public IP.

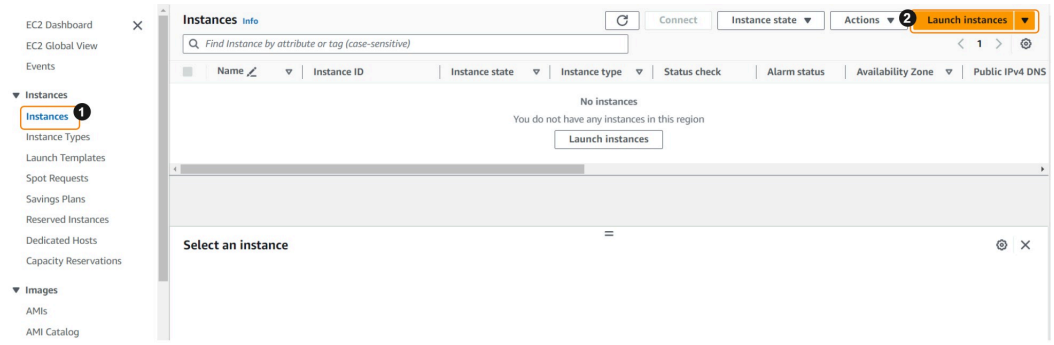
Lab Steps

Creating the EC2 instance

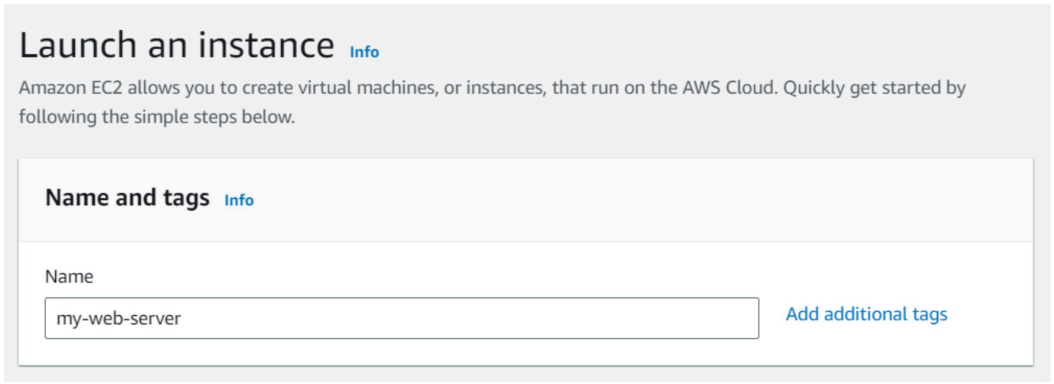
1. Search '*ec2*' in the AWS Management Console search bar. Click **EC2** on the search results.



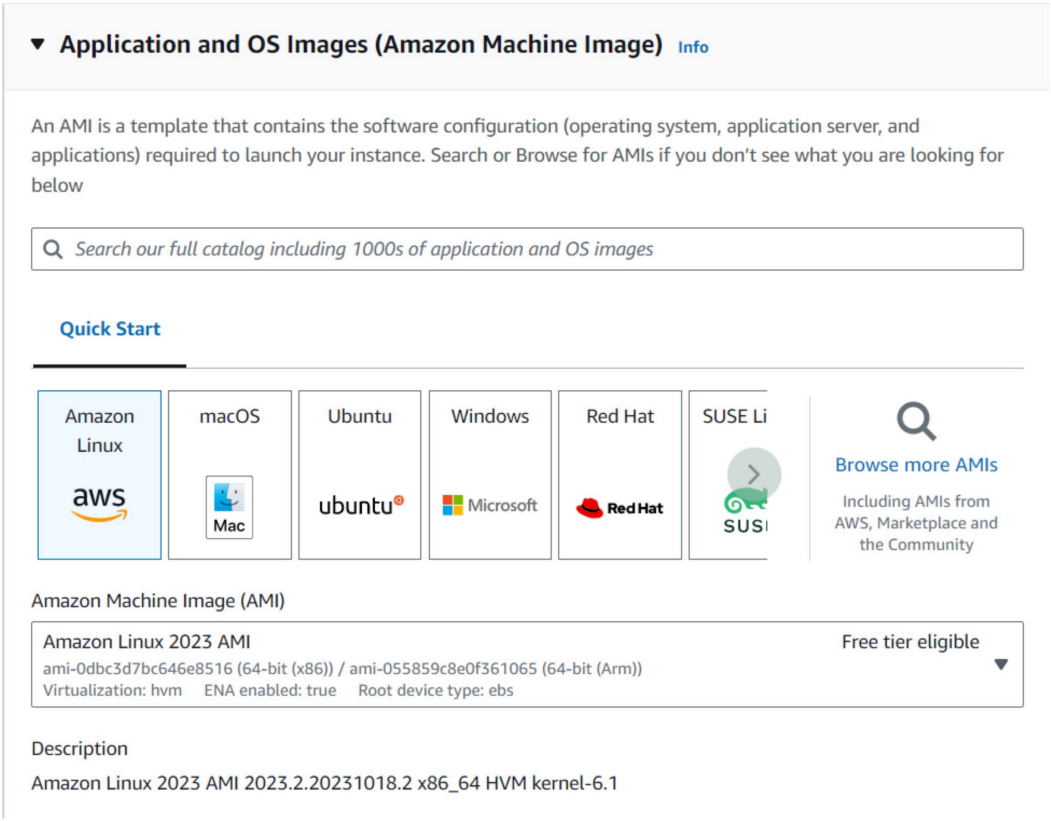
2. In the left window pane, select **Instances,** then click the **Launch instances** option.



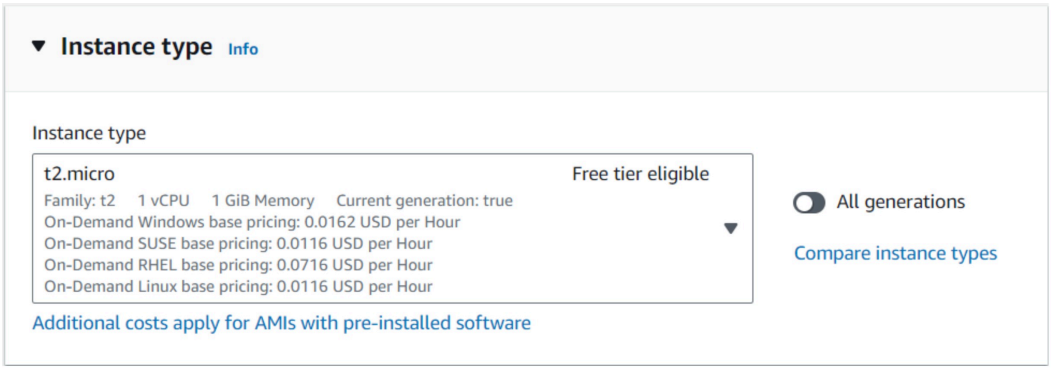
3. Name the instance '***my-web-server***' or any name that you prefer.



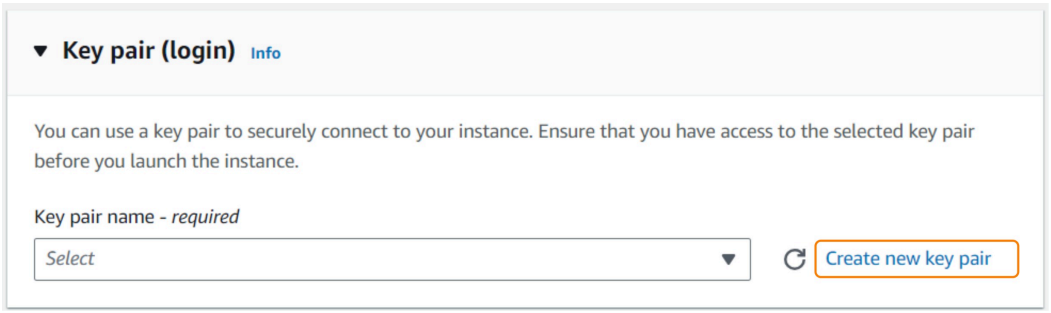
4. Under the **Application and OS Images** section, click the default **Amazon Linux** AMI.



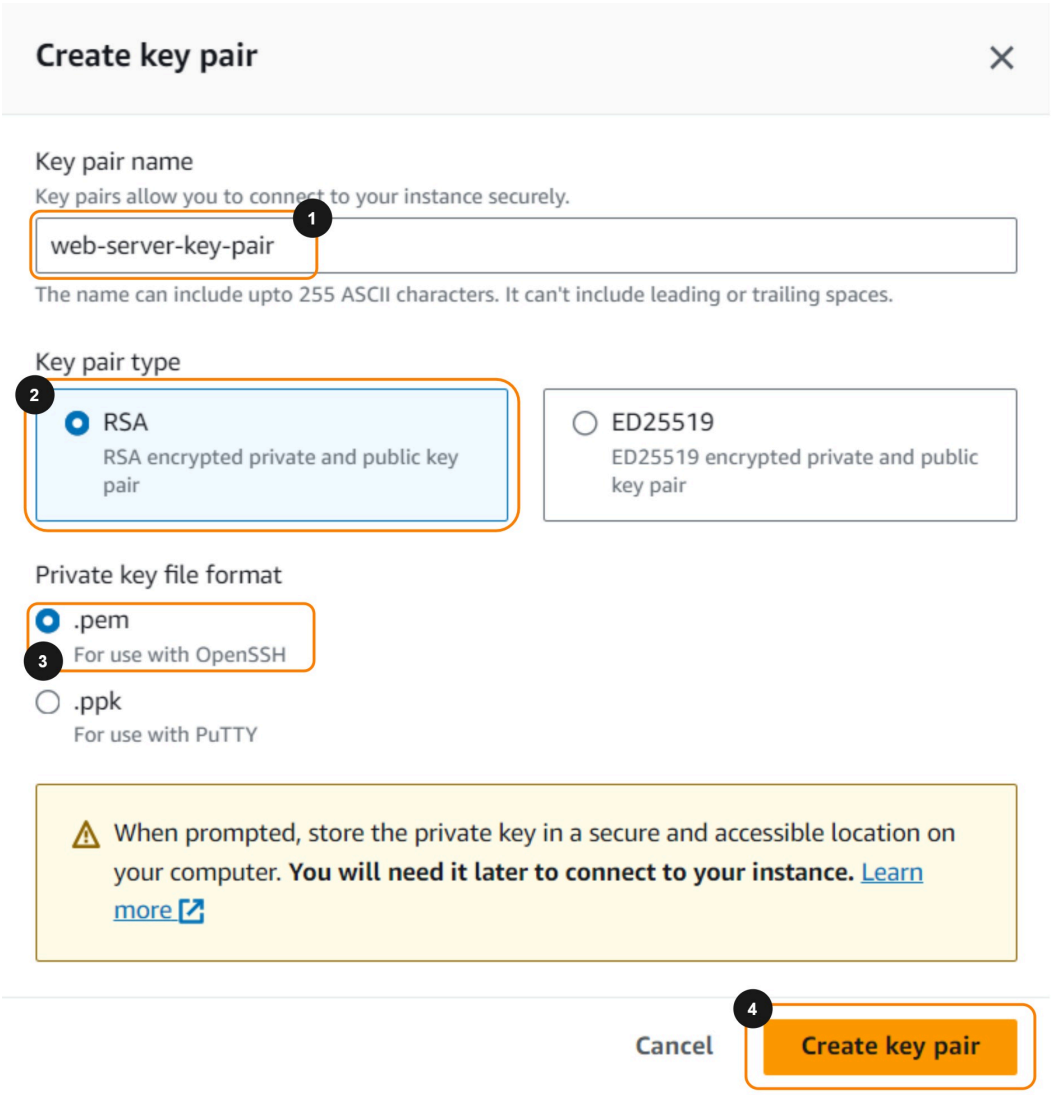
5. Under the **Instance Type** section, select **t2.micro**.



6. Under the **Key Pair** section, click **Create new key pair**.

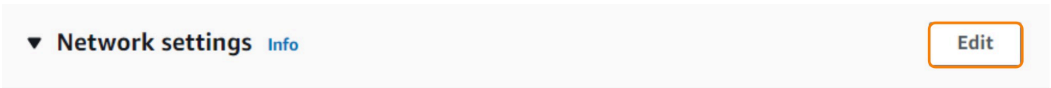


7. Enter a key pair name and follow the configurations below. Then, click **Create key pair**.



After creating a key pair, the private key will be downloaded to your computer. Remember to note the location of this file, as you'll need it later to SSH to your EC2 instance.

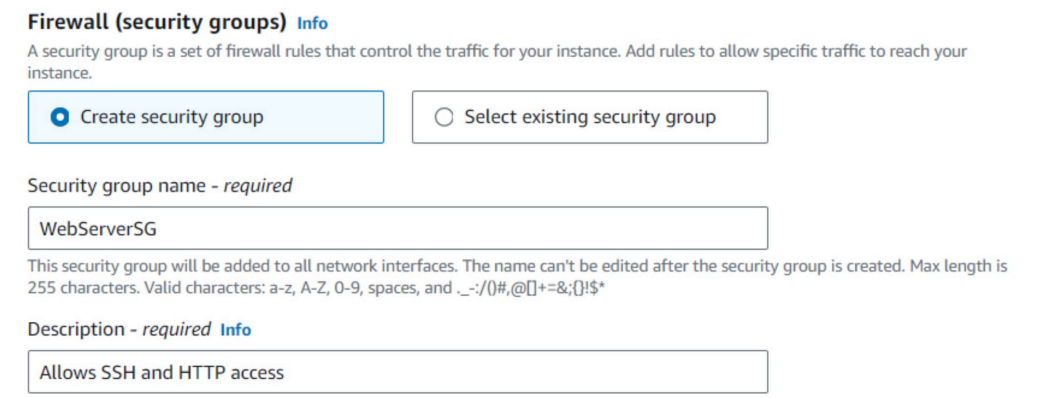
8. Under the **Network settings** section, click **Edit.**



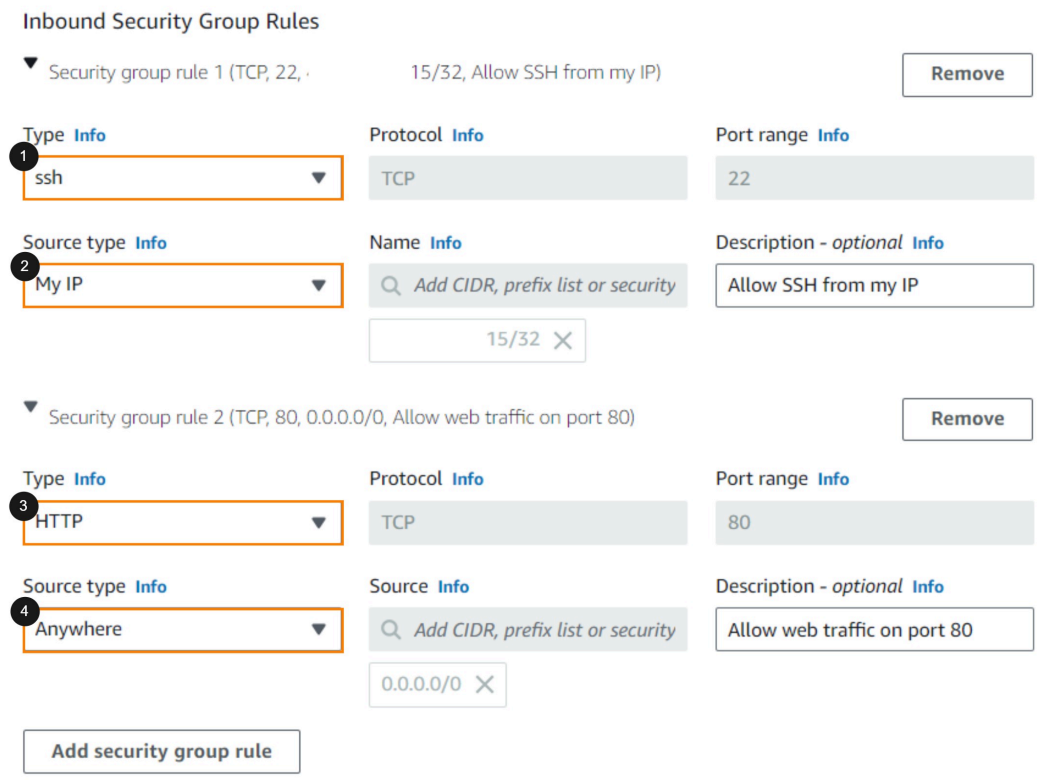
9. Scroll down the **Firewall (Security Groups)** option.

a. Enter '***WebServerSG***' for the security group name.

b. For **Description**, enter '***Allows SSH and HTTP access***'.



10. Add two inbound security group rules with the following configuration.



Inbound rule 1

Type	Source Type
SSH	My IP

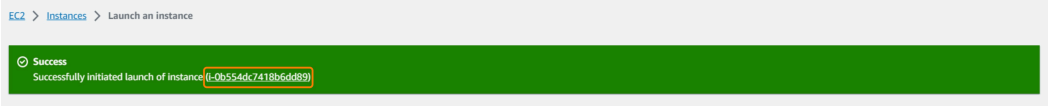
Inbound rule 2

Type	Source Type
HTTP	Anywhere (0.0.0.0/0)

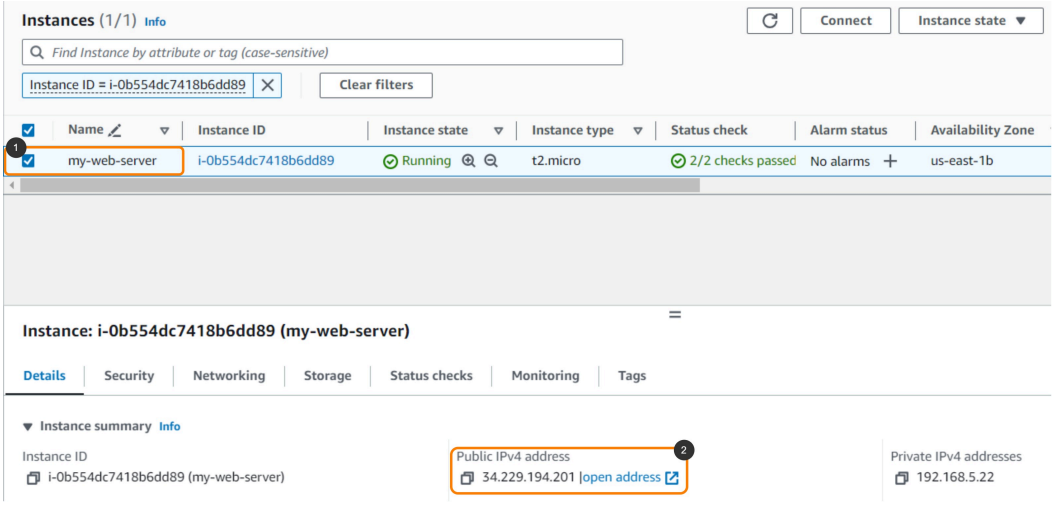
11. In the right window pane, at the bottom section, click **Launch instance**.

Setting up the web server

12. After the instance is created successfully, click the instance ID.



13. Tick the checkbox next to your instance name. Then, copy the Public IP address of your instance and paste it somewhere you can easily retrieve it later.



In this lab, we'll be using the SSH utility from OpenSSH. It usually comes built-in with Windows 10 and 11, Mac, and most Linux distributions. If your operating system doesn't have it pre-installed, ensure you install it first before proceeding.

14. Open up a terminal, then run the command below to connect to your instance via SSH.

```
ssh -i /path/to/YOUR-KEY.pem ec2-user@YOUR-EC2-PUBLIC-IP
```

Ensure that you reference the correct path to your private key pair and that you use the correct public IP of your EC2 instance.

Once connected, your shell prompt should change to something similar to `ec2-user@ip-192-168-5-22:~$`, confirming that you're now connected to your EC2 instance.

In the next steps, you will configure the necessary settings to set up a web server on the EC2 instance.

15. Run the command below to update the system.

```
sudo yum update -y
```

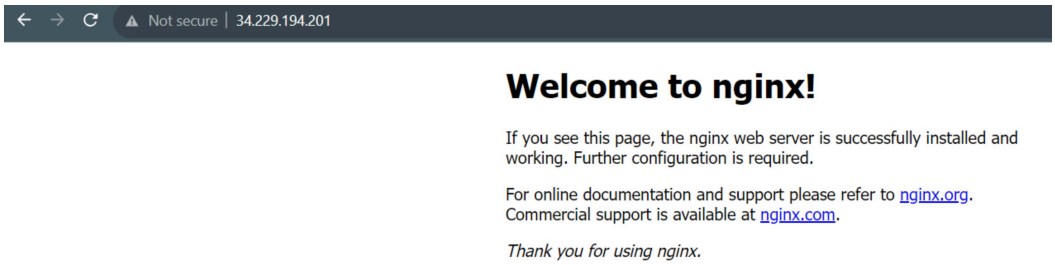
16. Once the update is completed, install Nginx.

```
sudo yum install nginx -y
```

17. Start the Nginx Service.

```
sudo service nginx start
```

18. Enter your EC2 instance's public IP in your browser. The default Nginx welcome page should be displayed.



Now, let's replace the welcome page with a custom one.

19. Go to the `/usr/share/nginx/html/`

```
cd /usr/share/nginx/html/
```

20. Create a custom HTML page.

```
echo '<h1>Welcome to my web page!</h1>' | sudo tee
mypage.html > /dev/null
```

21. Let's override the default Nginx configuration by creating a new configuration file in the `/etc/nginx/conf.d/`

```
sudo vi /etc/nginx/conf.d/server.conf
```

22. Press `i` to enter Insert mode in Vi and paste the following configuration.

```
server {
    listen 80 default_server;
    server_name _;
    root /usr/share/nginx/html;

    location / {
        index mypage.html;
    }
}
```

23. Press the Escape button and enter `:wq!` to exit and save your changes.

24. Reload Nginx for the changes to take effect.

```
sudo nginx -t && sudo service nginx reload
```

25. Reload your browser to see the changes you've made.

Congratulations! You've successfully set up a web server on an Amazon EC2 instance using Nginx. You've also hosted a custom web page, giving you foundational skills in web hosting on the cloud. This is just the beginning. As you continue to explore, you can experiment with different configurations, host more complex web applications, and even integrate databases.