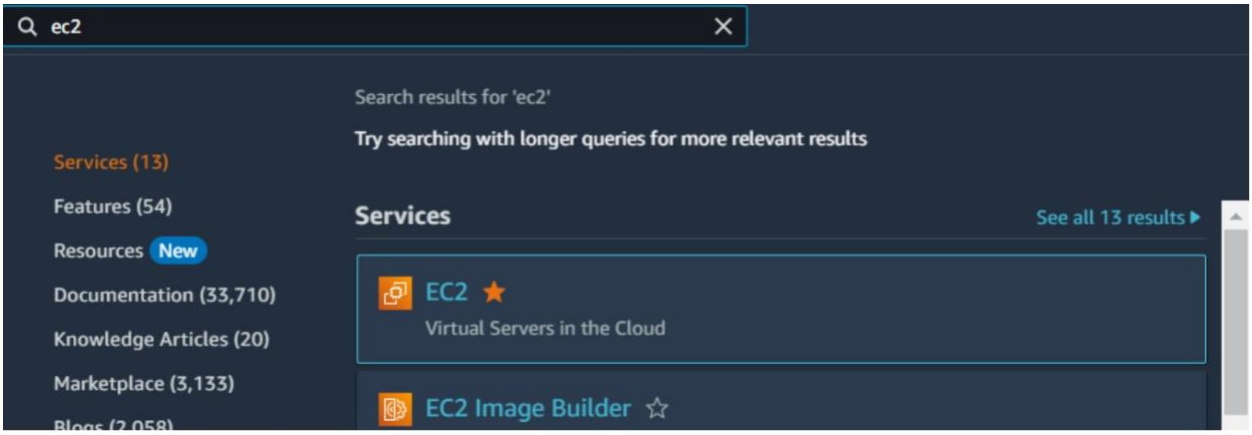
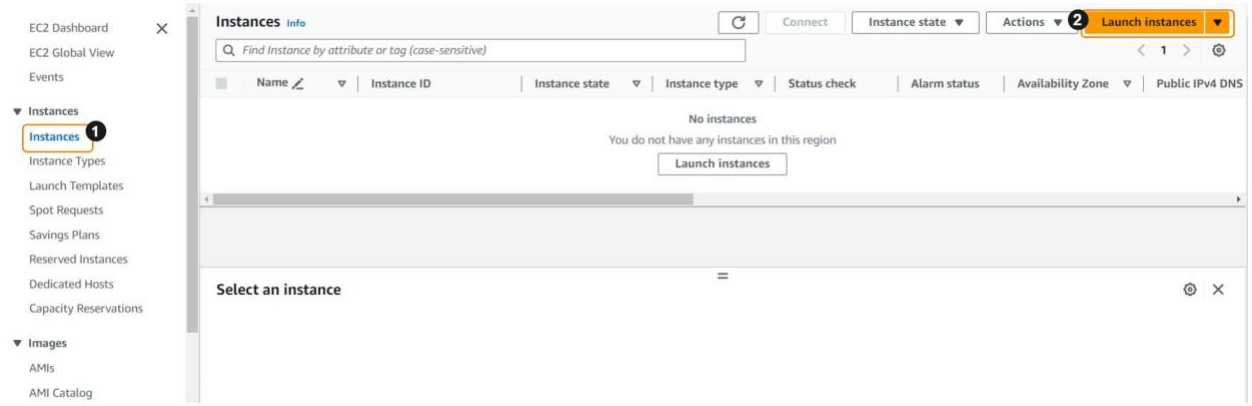
Setting up a Web server on an EC2 instance

Objectives:

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.

Creating the EC2 instance





Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

my-web-server

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible

ami-0dbc3d7bc646e8516 (64-bit (x86)) / ami-055859c8e0f361065 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.2.20231018.2 x86_64 HVM kernel-6.1

Instance type Info

Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

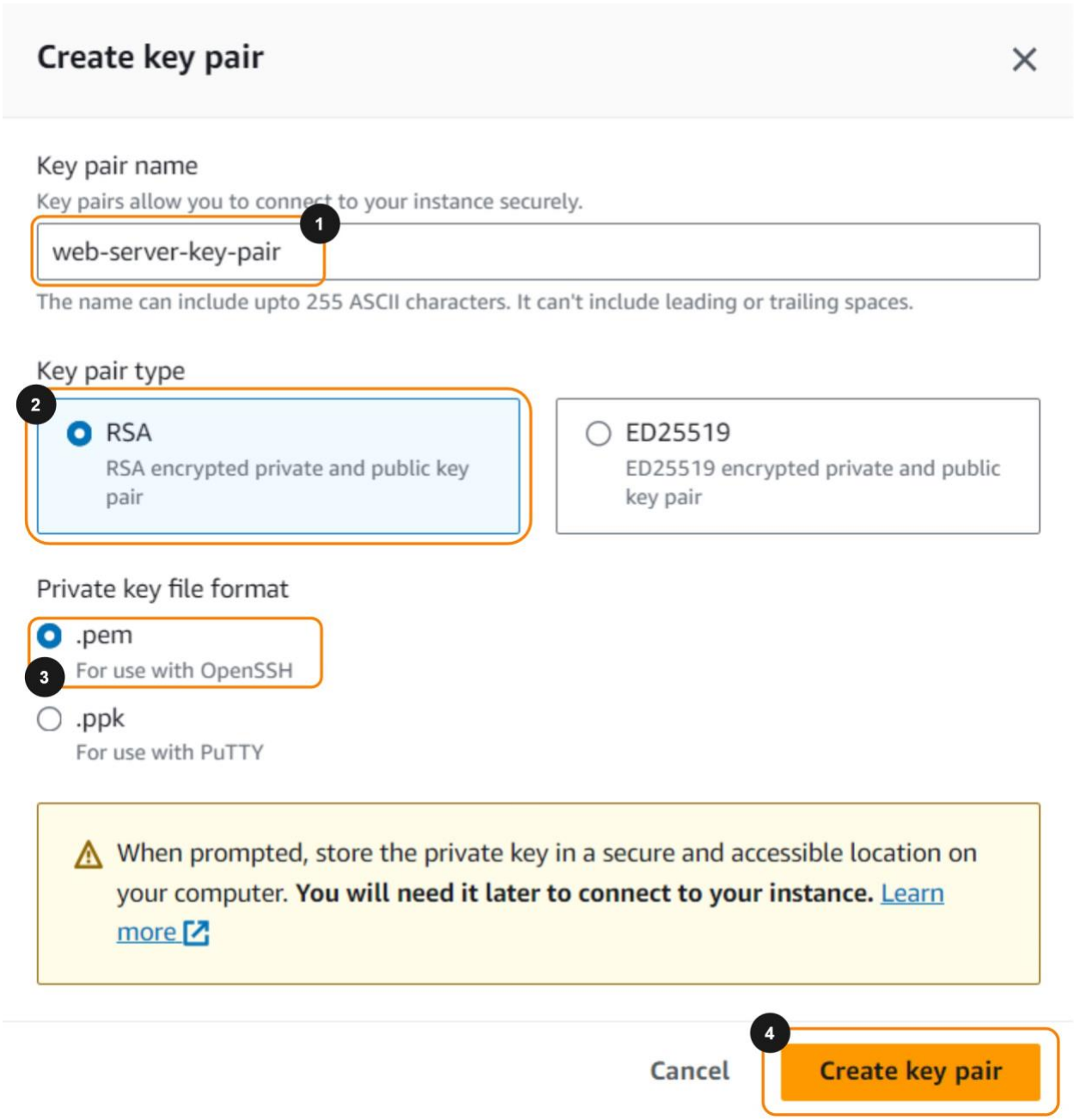
Key pair (login) Info

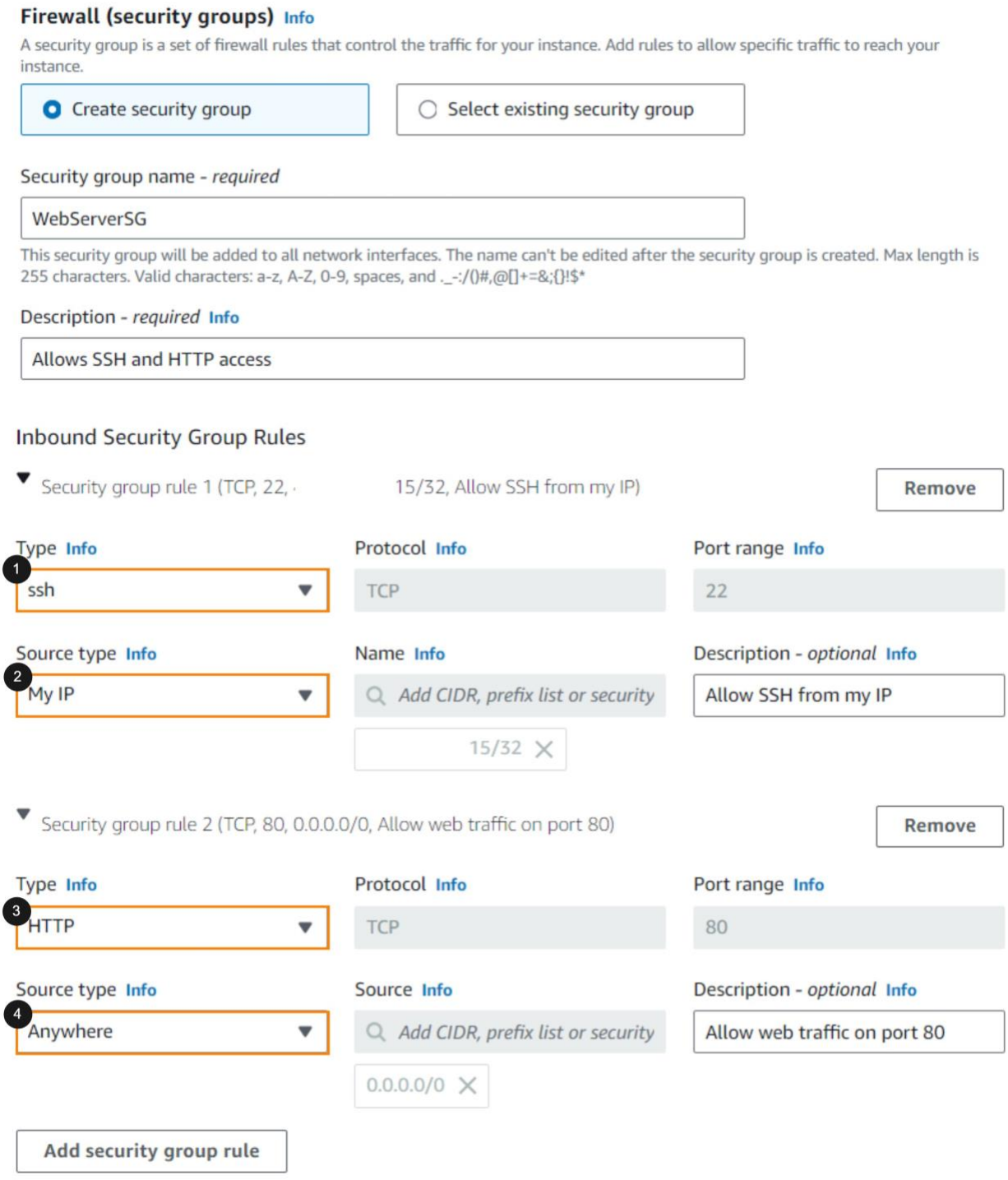
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select

Create new key pair

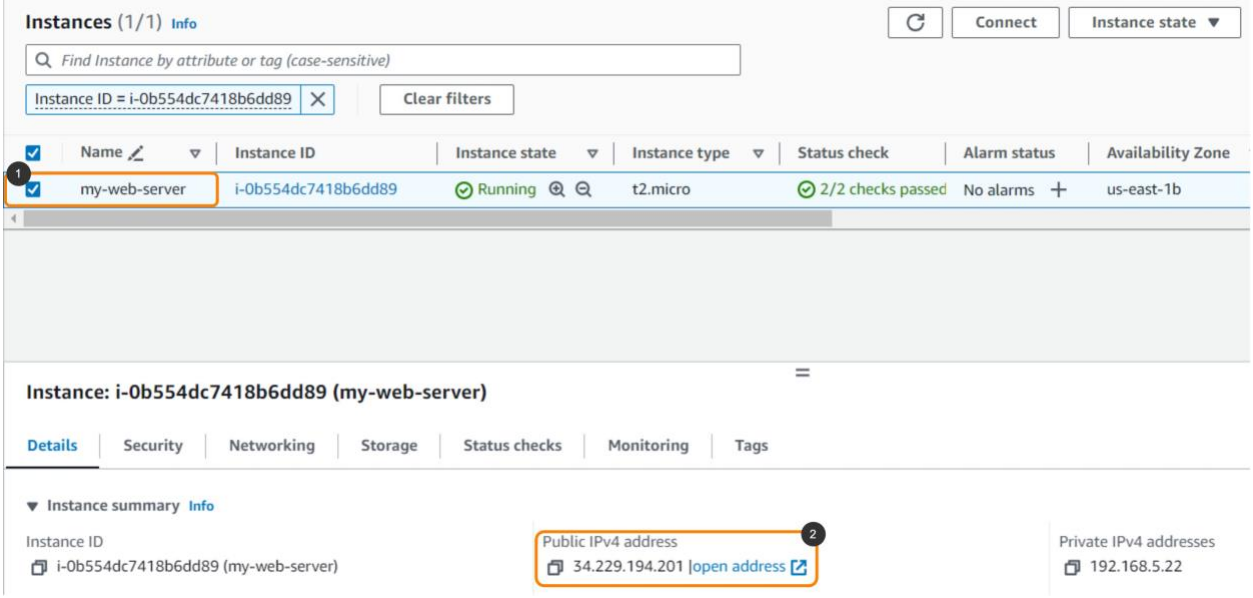

Create key pair
Key pair name
Key pairs allow you to connect to your instance securely.
1
web-server-key-pair
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type
2
RSA
RSA encrypted private and public key pair
ED25519
ED25519 encrypted private and public key pair
Private key file format
.pem
For use with OpenSSH
3
.ppk
For use with PuTTY
When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more
4
Cancel
Create key pair
Network settings Info
Edit



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

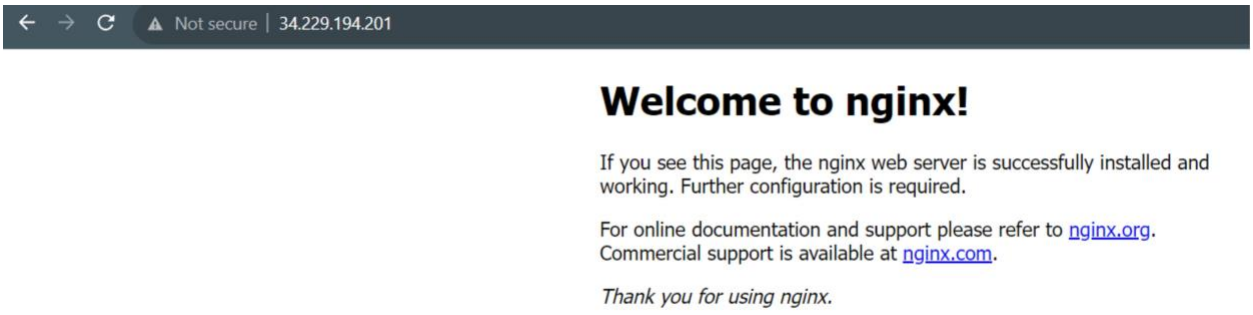
16. Once the update is completed, install Nginx.

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.

successfully set up a web server on an Amazon EC2 instance using Nginx