

Lab D

Create an EC2 instance running nginx to create a web server.

1. Open the Learner Lab

Choose the 'Learner Lab' course, **not** 'Cloud Foundations Lab'

In the modules click 'Launch AWS Academy Learner Lab'

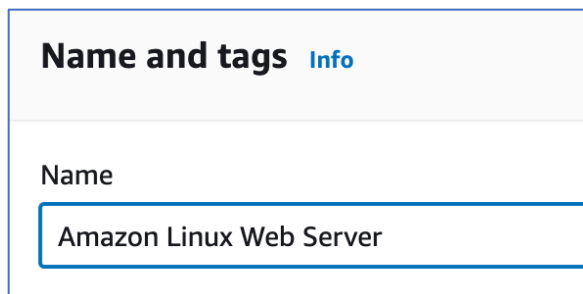
Click Start Lab – then click AWS link when ready (green)

From AWS console. **Change to the region US West Oregon.**

2. Create the instance

Open EC2 service.

Launch Instance – Give it a name



Name and tags [Info](#)

Name

Amazon Linux Web Server

Select the Amazon Linux



Leave architecture at 64-bit (x86) and the instance type of t2.micro.

Click the 'Create new key pair' link. Even if you already have some, we want to create one that is unique for this server.

▼ 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

Give it a name, leave the key pair type as RSA

Everyone use the .pem OpenSSH (For those with windows less than version 10 you would need to use PuTTY—this is how you do SSH on older windows 7 and 8. See step instructor for steps.)

Click create key pair.

Create key pair X

Key pair name
Key pairs allow you to connect to your instance securely.
amazon_linux_web_kp
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ 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

☐ .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](#) ↗

A file will download with your key. Open that up in file management. You will want to place it somewhere other than 'Downloads'. So, move it where you want it. You will navigate to this location later in command line interface.

In the 'Network settings' make sure it's on 'Create security group' and click Edit in the upper right of that Network settings section. Give 'Security group name' a name (replace what was there if anything). Give it a description.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to your instance.

☒ Create security group

☐ Select existing security group

Security group name - *required*

http_ssh_sg

This security group will be added to all network interfaces. The name can't be edited after the security group is created. The name can be up to 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . _ - : / () # , @ [] + = & ; ' ! \$ % * ~

Description - *required* [Info](#)

Security Group for HTTP and SSH

'Type' should be ssh and 'Source type' is Anywhere. Can give it a description of SSH and leave all other defaults.

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0, SSH) [Remove](#)

Type Info	Protocol Info	Port range Info
ssh ▼	TCP	22
Source type Info	Source Info	Description - <i>optional</i> Info
Anywhere ▼	<input type="text" value="0.0.0.0/0"/> Add CIDR, prefix list or security group	SSH

Add another security group rule. This one is HTTP

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0, HTTP) [Remove](#)

Type Info	Protocol Info	Port range Info
HTTP ▼	TCP	80
Source type Info	Source Info	Description - <i>optional</i> Info
Anywhere ▼	<input type="text" value="0.0.0.0/0"/> Add CIDR, prefix list or security group	HTTP

Scroll down to the next section and leave the storage default of 8GB.

Click the expand triangle next to 'Advanced details' and enable 'Termination protection'. This keeps it from being accidentally terminated.

Termination protection

Info

Enable

Click Launch instance.

Wait for the Instance state to go from 'Pending' to 'Running'.

Open the instance we just created. It might be pending in the 'Instance state' and take a little while for it to get started. Click the check mark next to your instance.

Under the 'Details' tab notice the Public IPv4 address available to us we will be viewing it later.

3. Connect to your instance.

Go back to your Instance and check the box next to the instance. Click 'Connect'.

Instances (1/1) Info			Connect	Instance state ▼	Actions ▼
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>					
<input checked="" type="checkbox"/>	Name	▼	Instance ID	Instance state ▼	Instance
<input checked="" type="checkbox"/>	Amazon Linux Web Server		i-09d22341175de57bf	Running	t2.micro

Click on the SSH client tab

Info

Connect to your instance `i-09d22341175de57bf` (Amazon Linux Web Server) using any of these options


EC2 Instance Connect

Session Manager

SSH client

EC2 serial console


Instance ID

 `i-09d22341175de57bf` (Amazon Linux Web Server)


1. Open an SSH client.

2. Locate your private key file. The key used to launch this instance is `amazon_linux_web_kp.pem`


3. Run this command, if necessary, to ensure your key is not publicly viewable.

 `chmod 400 amazon_linux_web_kp.pem`

4. Connect to your instance using its Public DNS:

 `ec2-3-87-154-9.compute-1.amazonaws.com`

Example:

 `ssh -i "amazon_linux_web_kp.pem" ec2-user@ec2-3-87-154-9.compute-1.amazonaws.com`

Copy the Example command line command given near the bottom that will allow you to connect to this instance. We will use it a bit later.

Open up your terminal. Not cmd for windows users, windows users use PowerShell.

Navigate your prompt to the directory where your keys are stored.

Paste in the command we copied from our instance:

```
~/keys$ ssh -i "amazon_linux_web_kp.pem" ec2-user@ec2-34-208-167-1.us-west-2.compute.amazonaws.com
```

Say yes:

```
The authenticity of host 'ec2-34-208-167-1.us-west-2.compute.amazonaws.com (34.208.167.1)' can't be established.  
ED25519 key fingerprint is SHA256:0ukm86aeF6G+bDPNr/XX5jAIkG0dD08Mmufj01Augkk.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Then you should get connected. It will look like this (if you don't see troubleshooting below):

```
Warning: Permanently added 'ec2-54-162-81-82.compute-1.amazonaws.com,54.162.81.8'
2' (ECDSA) to the list of known hosts.
```


Amazon Linux 2023

|
\#/
V~> https://aws.amazon.com/linux/amazon-linux-2023

#####


```
[ec2-user@ip-172-31-37-15 ~]$
```

Troubleshooting only - If you get this:

```
Warning: Permanently added 'ec2-35-161-82-30.us-west-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@          WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'amazon_linux_web_kp.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "amazon_linux_web_kp.pem": bad permissions
ec2-user@ec2-35-161-82-30.us-west-2.compute.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

Then copy and run the chmod command from AWS and then try the ssh command again:

3. Run this command, if necessary, to ensure your key is not publicly viewable.

 `chmod 400 amazon_linux_web_kp.pem`

4. Install nginx

Install nginx with this command:

```
sudo yum install nginx
```

y

Start the engine with this command: (that is systemctl, not a number 1)

```
]$ sudo systemctl start nginx
```

To verify everything is up and running:

```
$ wget http://localhost
```

And we have an index.html file there.

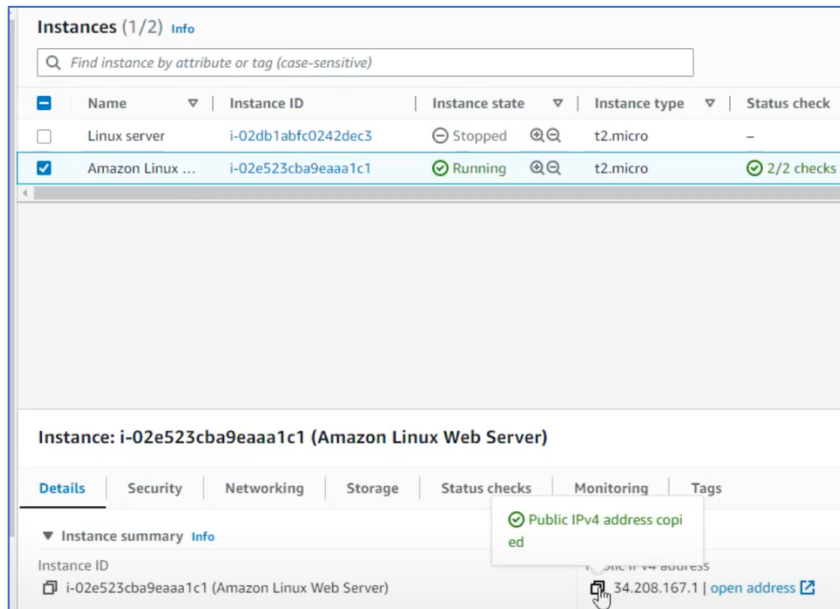
```
2023-02-09 16:28:20 (119 MB/s) - 'index.html' saved [615/615]
```

Go back to our instance get out of the 'Connect to instance' window by clicking cancel

And check the instance again.

5. To view the index.html of our new web server.

Grab the IP address

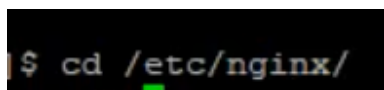


And paste it into a browser tab <http://xx.xxx.xxx.1>

And you should be able to get into your nginx page:

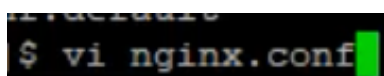


Let's find where that file is. Go to where nginx was installed

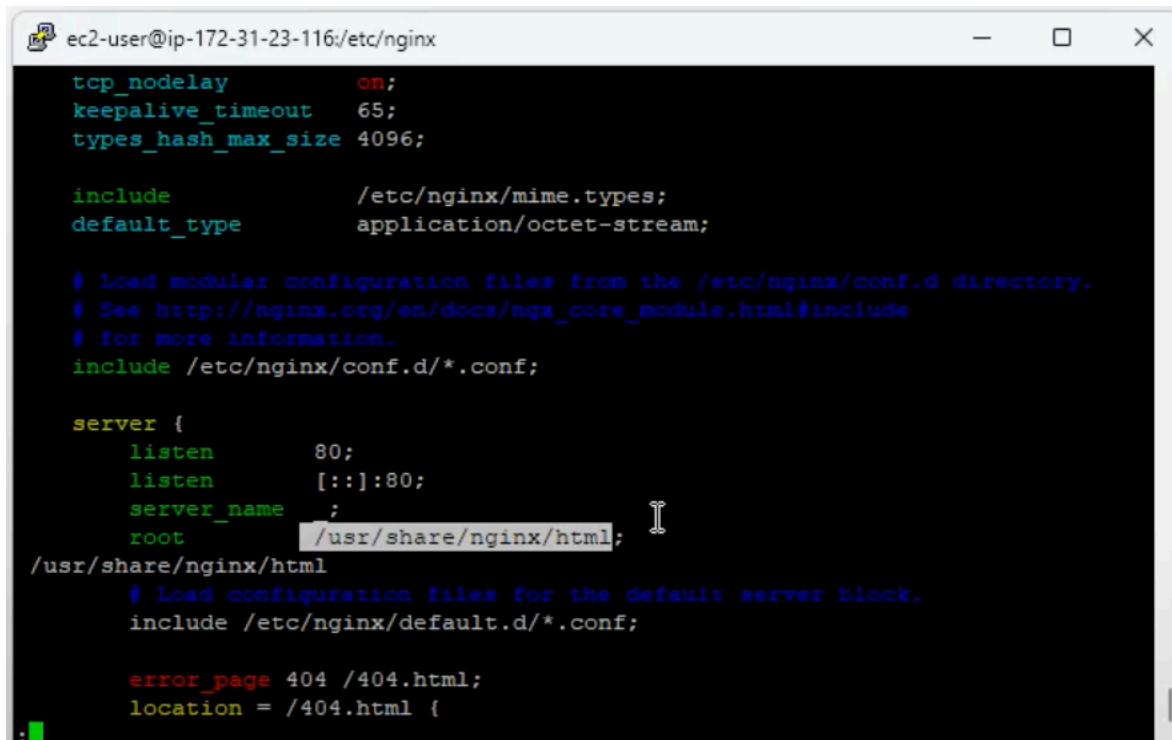


The ls command there will show the files

Edit the configuration file with:



Arrow down to the server root and copy this path:



```
ec2-user@ip-172-31-23-116:/etc/nginx
tcp_nodelay      on;
keepalive_timeout 65;
types_hash_max_size 4096;

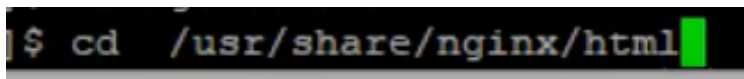
include          /etc/nginx/mime.types;
default_type     application/octet-stream;

# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    listen        80;
    listen        [::]:80;
    server_name    ;
    root          /usr/share/nginx/html;
/usr/share/nginx/html
    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    error_page 404 /404.html;
    location = /404.html {
```

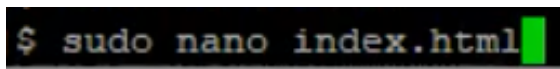
Don't save it and get out of the editor Shift+Q+Z and go to that directory.



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

The 'ls' command will now show the index.html file.

Type this to edit that file:



```
$ sudo nano index.html
```

Use the arrow keys to edit the <h1> for example. *Use your name, not mine.*

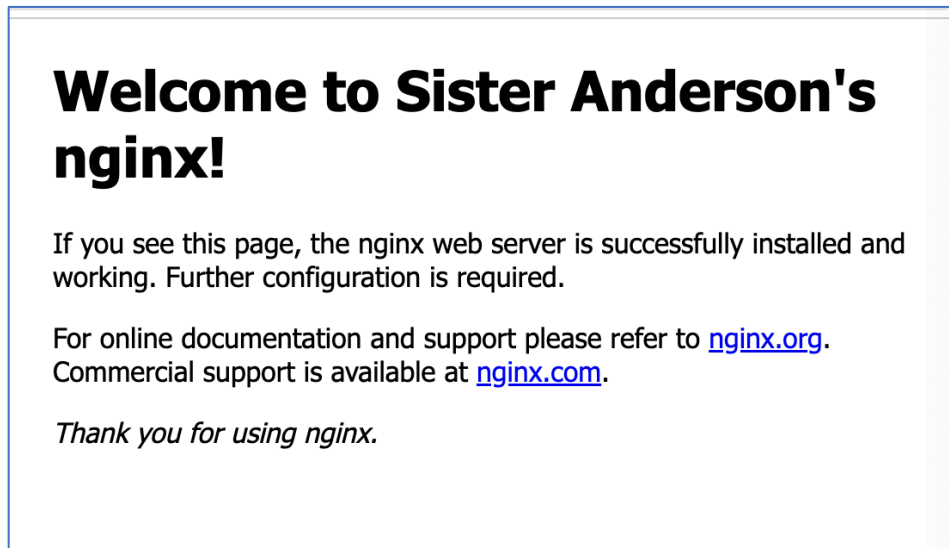

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to Sister Anderson's nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```

Ctrl-O then enter, to write. Should see something like 'Wrote out 23 lines'

Ctrl-X to exit

Refresh your browser page and it should show the changes.



Screenshot this index.html page for submission.

Note: you can stop the instance to save money. If it's running you should see your web page.

To get it going again if you stopped it. You will have to start the instance (note: the instance list may be filtering to only show the running instances so take off the filter if you need to), connect the instance again. In your command line terminal go the path where your key is still stored and paste the ssh command again. Start nginx again with 'sudo systemctl start nginx' and your web page should work again.