

1. Creating an AWS (free) Account

1. Go to <https://aws.amazon.com/>
2. Click on **[Create an AWS Account]** (top right of the page).
3. (if it takes you to a sign-in page, click on the **[Create an AWS Account]** button at the bottom of that page).

4. Fill in your email address. Use your **@utsa** address.
5. Name the account utsa.
6. Click **[Verify email address]**
7. Check your email and copy the verification code

Sign up for AWS

Root user email address

Used for account recovery and some administrative functions

AWS account name

Choose a name for your account. You can change this name in your account settings after you sign up.

6. Click **[Verify email address]**

7. Check your email and copy the verification code



Verification code



(This code will expire 10 minutes after it was sent.)

8. Create a strong password. Enter it twice. Save it somewhere. Never reuse passwords across services, especially critical ones (bank, LMS, SIS, etc.)

Create your password

It's you! Your email address has been successfully verified.

Your password provides you with sign in access to AWS, so it's important we get it right.

Root user password

Confirm root user password

Continue (step 1 of 5)

9. Fill in the contact information, (The form is longer than what's shown below.) Once completed, tick to accept the agreement, then click **[continue]**

Contact Information

How do you plan to use AWS?

- Business - for your work, school, or organization
- Personal - for your own projects

Who should we contact about this account?

Full Name

Ziad Najem

Organization name

UTSA

...

- I have read and agree to the terms of the [AWS Customer Agreement](#).

Continue (step 2 of 5)

10. Fill in the Billing information (again longer than what's shown here)

Secure verification

ⓘ We will not charge you for usage below AWS Free Tier limits. We may temporarily hold up to \$1 USD (or an equivalent amount in local currency) as a pending transaction for 3-5 days to verify your identity.

Billing Information

Billing country

Your billing country determines the payment methods available to you to pay for AWS services.

United States ▾

Credit or Debit card number

11. Click **[verify and continue]**

Verify and continue (step 3 of 5)

You might be redirected to your bank's website to authorize the verification charge.

12. Now verify your phone number. Follow the steps below.

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

- Text message (SMS)
 Voice call

Country or region code

United States (+1) ▾

Mobile phone number

⚠ A phone number is required.

Send SMS (step 4 of 5)

Security Verification



Type the characters as shown above

Reset **Submit**

Confirm your identity

Verify code

2583

Continue (step 4 of 5)

13. Finally, make sure you choose the FREE support.

- Basic support - Free
- Recommended for new users just getting started with AWS
 - 24x7 self-service access to AWS resources
 - For account and billing issues only
 - Access to Personal Health Dashboard & Trusted Advisor



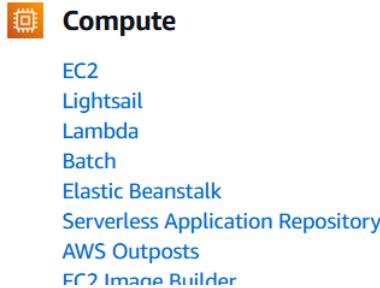
14. Congratulations. You're done. You may now go to the console screen.

Go to the AWS Management Console

2. Creating your server (EC2 instance)

1. Once on the management console page, click on **[view all services]**. You can find it under the burger menu on the top-left corner of the console page.
2. Now select **[EC2]** from **[compute]**

Services by category



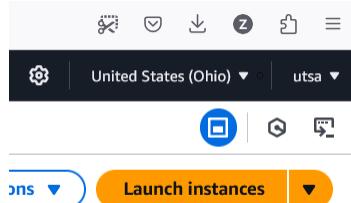
3. Click on the **[instances (running)]** link.

Resources

You are using the following Amazon EC2 resource

Instances (running) 0

4. Make sure you're on the Ohio cloud (top-right corner of the page).



5. Now click the orange button **[Launch instances]**

- a. Name your instance

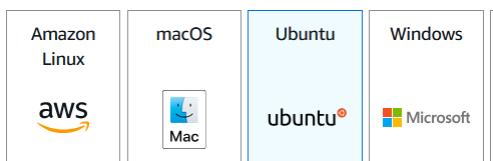
Name and tags Info

Name

cs4413

- b. Choose Ubuntu and don't change anything in the configuration.

Quick Start



- c. Double check that Instance type is **[t2.micro]** which is the default. That's the free instance.

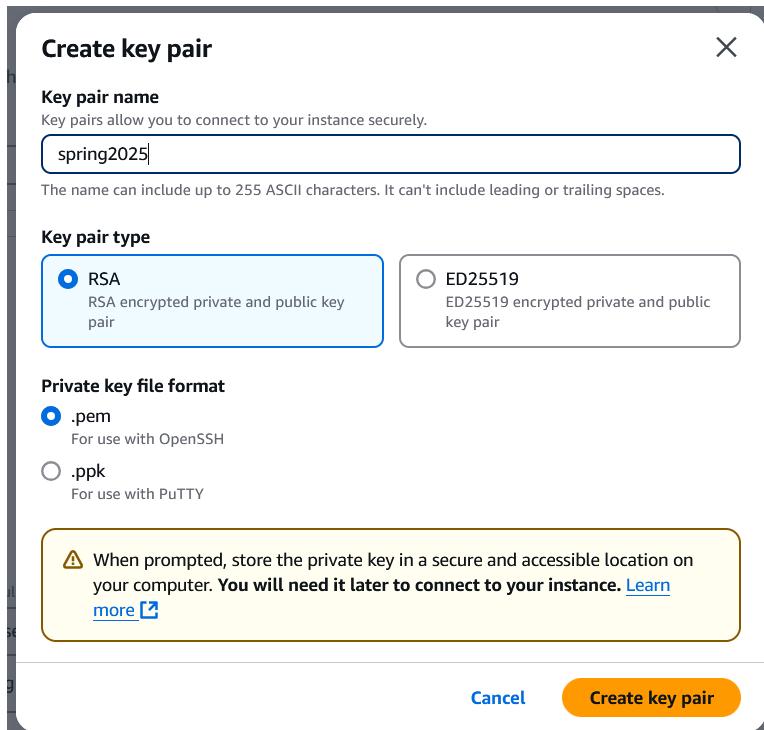
▼ **Instance type** [Info](#) | [Get advice](#)

Instance type

t2.micro	Free tier eligible
Family: t2	1 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour	On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour	On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour	

- d. Create a Key Pair for secured login. Click **[Create key pair]**.

- e. Give the key a name and keep everything else as the default is. Double check with the figure below:



- f. Make sure you save and never lose the **.pem** file that gets downloaded. If you lose it, you'll need to login using the traditional user/password and then regenerate/reconfigure a new pair.

- g. Under network settings, make sure you enable ssh from anywhere, https, and http.

<input checked="" type="checkbox"/> Allow SSH traffic from Helps you connect to your instance	Anywhere 0.0.0.0/0
<input checked="" type="checkbox"/> Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server	
<input checked="" type="checkbox"/> Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server	

- h. Under [configure storage], claim the whole 30 GiB for the root volume,

A screenshot of the 'Configure storage' section. It shows a dropdown menu with '1x' selected, followed by a numeric input field containing '30' with up and down arrows, and a unit selector 'GiB'. To the right is a dropdown menu set to 'gp3'. Below this, there is a 'Launch instance' button.

- i. Finally, click **Launch instance** on the right side of the page.

6. Choose **[dashboard]** from the burger menu. Notice that you have one instance running .
7. Click on the **[running instances]** link and then verify that your instance is indeed running.

<input type="checkbox"/>	Name	Instance ID	Instance state	Inst:
<input type="checkbox"/>	cs4413	i-0d178c5ee5ed1dd40	Running	t2.nr

8. Now click on the instance ID of your server. This should give you interesting details about your server and how to access it.

A screenshot of the 'Instance summary' page for instance 'i-0d178c5ee5ed1dd40 (cs4413)'. The page displays various details about the instance, including its state, type, and network configuration. Key information includes:

- Public IPv4 address:** 18.191.66.23 | [open address](#)
- Private IPv4 addresses:** 172.31.10.2
- Public IPv4 DNS:** ec2-18-191-66-23.us-east-2.compute.amazonaws.com | [open address](#)
- Instance state:** Running
- Instance type:** t2.micro
- VPC ID:** vpc-042c52f907035a7ds
- Subnet ID:** subnet-0d47429e0b9a57552
- Instance ARN:** arn:aws:ec2:us-east-2:148761657770:instance/i-0d178c5ee5ed1dd40
- Auto Scaling Group name:** -
- Managed:** false

Never stop the instance!

You can reboot it, but never stop it. If stopped, you'll lose the IP address given to you, which will require you redo many steps we'll be taking to configure the webserver based on the current IP.

9. Now click the **[connect]** button on the top-right of the page. This will take you to a screen with four methods of connecting to the server.

Connect to instance Info

Connect to your instance i-0d178c5ee5ed1dd40 (cs4413) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID
 i-0d178c5ee5ed1dd40 (cs4413)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IPv4 address
 18.191.66.23

IPv6 address

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default `ubuntu`.

ubuntu X

Note: In most cases, the default username, `ubuntu`, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

[Cancel](#)

Connect

10. The **[EC2 Instance Connect]** is your safe way to connect to the server through the browser. Understandably it will be slow, but it could be useful as a last resort. Just make sure you can get to your aws dashboard.

3. Connecting to your server

In normal operation, you'll connect to your AWS server using using SSH. Follow the instructions in the previous section to go to the connections page, then continue with the following steps:

1. Click on the **[SSH client]** tab.

The screenshot shows the AWS EC2 Instance Connect interface. The top navigation bar has four tabs: EC2 Instance Connect, Session Manager, **SSH client**, and EC2 serial console. The SSH client tab is selected. Below the tabs, there is a section for "Instance ID" which shows "i-0d178c5ee5ed1dd40 (cs4413)". Underneath this, a numbered list of steps for connecting via SSH is provided:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `spring2025.pem`
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "spring2025.pem"`
4. Connect to your instance using its Public DNS:
`ssh -i "spring2025.pem" ubuntu@ec2-18-191-66-23.us-east-2.compute.amazonaws.com`

A section titled "Example:" contains the command:`ssh -i "spring2025.pem" ubuntu@ec2-18-191-66-23.us-east-2.compute.amazonaws.com`Below this, a callout box contains the note:

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

[Cancel](#)

2. Copy the ssh command that's shown in the Example. Notice that the command already includes the DNS name for your AWS server, and the file name for your .pem key.
3. Open the command-line (CLI) on your local machine (Terminal, PowerShell, etc.) You can also use any of the dedicated SSH apps.
4. Paste the ssh command you copied from the browser (step 2) into the CLI.
5. It is normal for ssh to ask you for confirmation the first time you try to connect to a new machine (something about fingerprint). Answer [yes].
6. The command will fail complaining that your .pem file is not found.
7. Copy the .pem (from wherever it got downloaded) into whichever local directory your CLI is in. Normally that's your home directory
8. Now redo step 4.
9. Are you on your server? If so, go to the next section. If not, call for help!

4. Updating / Upgrading your Ubuntu server

It's a good practice to always check if you need to update/upgrade your server... especially upgrades related to security. Updating packages, however, is a critical process since packages are usually inter dependent on each other, and updating one might break some dependency for the another.

Bottom line!

we'll only update our system once right after creating the instance, but never again during this semester. But do keep in mind that this is not the right approach if you're a system admin.

To update/upgrade your Ubuntu, issue the following two commands: (don't copy the \$ sign. That's just an indication that the line is a UNIX shell command)

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

5. Creating a self-signed certificate

Normally, if you're planning to offer https services through your webserver, you'd get a certificate from a 3rd party (an independent entity that can verify the authenticity of your webserver to the user.) But for this course, we'll issue our own.

1. Issue the following command (this should all be copied as one line.)

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-  
selfsigned.crt
```

2. Enter the requested information as seen below. Make sure you enter your own email utsa email address and the FQDN of your server (found on the dashboard, see the screenshot that follows)

```
Country Name (2 letter code) [AU]:US  
State or Province Name (full name) [Some-State]:Texas  
Locality Name (eg, city) []:San Antonio  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UTSA  
Organizational Unit Name (eg, section) []:Computer Science  
Common Name (e.g. server FQDN or YOUR name) []:ec2-....amazonaws.com  
Email Address []:your.email@utsa.edu
```

3. We'll use a dhparam file to strengthen the security of TLS key exchanges, making it harder for attackers to decrypt our encrypted communications. Issue the following command:

```
$ sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

6. Installing and configuring your very own nginx HTTP Server

1. Let's install the NGINX (engine x) http/https server:

```
$ sudo apt install nginx
```

2. Configure the self-signed certificate:

```
$ echo
'c3NsX2NlcnRpZmljYXR1IC9ldGMvc3NsL2NlcnRzL25naW54LXN1bGZzaWduZWQuY3J0Owpzc2xfY2VydGlm
aWNhdGVfa2V5IC9ldGMvc3NsL3ByaXhdGUvbmdpbngtc2VsZnNpZ25lZC5rZXk7Cg==' | base64 -d - |
sudo tee /etc/nginx/snippets/self-signed.conf > /dev/null
```

3. Configure the SSL parameters:

```
$ echo
'c3NsX3Byb3RvY29scyBUTFN2MS4yOwpzc2xfcHJ1ZmVyX3NlcnZlc19jaXBoZXJzIG9uOwpzc2xfY21waGVy
cyAiRUVDRERgQUVTR0NNokVESctBRVNHQ006QUVTMjU2K0VFQ0RIOkFFUzI1NtFREgiOwpzc2xfZWNkaF9jd
XJ2ZSBzZWNwMzg0cjE7CnNzbF9zZXNzaW9uX2NhY2h1IHNoYXJ1ZDpTU0w6MTBtOwpzc2xfc2Vzc2lvb190aW
NrZXrZIG9mZjsKc3NsX3N0YXBsaW5nIG9uOwpzc2xfc3RhcgxpbmdfdmVyaWZ5IG9uOwpzyZXNvbHZlciA4Ljg
uOC44IDguOC40LjQgdmFsaWQ9MzAwczsKcmVzb2x2ZXJfdGltZW91dCA1czsKYWRkX2h1YWRlciBTdHJpY3Qt
VHJhbNwb3J0LVN1Y3VyaXR5ICJtYXgtYWdlPTYzMdcyMDAwOyBpbmNsWR1U3ViZG9tYWlucyI7CmFkZF9oZ
WFkZXIgWC1GcmFtZS1PcHRpb25zIERFT1k7CmFkZF9oZWfKZXIgWC1Db250ZW50LVR5cGUTT3B0aW9ucyBub3
NuaWZmOwpzc2xfZGhwYXJhbSAvZXrjl3NzbC9jZXJ0cy9kaHBhcmFtLnBlaTsK' | base64 -d - |
sudo tee /etc/nginx/snippets/ssl-params.conf > /dev/null
```

4. Make a backup of the main configuration file:

```
$ sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/distrib
```

5. Now we can do our changes to the main configuration file.

```
$ echo
'c2VydMViHsKCSMgU1NMIGNvbmZpZ3VyYXRpb24KCWxpc3R1biA0NDMgc3NsIGH0dHAyIGR1ZmF1bHRfc2Vy
dmVyOwoJbGlzdGVuIFs6O106NDQzIHNzbCBodHRwMiBkZWZhdWx0X3NlcnZlcjsKCWluY2x1ZGUgc25pcHB1d
HMvc2VsZi1zaWduZWQuY29uZjsKCWluY2x1ZGUgc25pcHB1dHMvc3NsLXBhcmFtcy5jb25mOwoKCXJvb3QgL3
Zhci93d3cvaHRtbDsKCgkjIEFkZCBpbmRleC5waHAgdG8gdGh1IGxpc3QgaWYgeW91IGFyZSB1c2luZyBQSFA
KCWluZGV4IGluZGV4Lmh0bWwgaW5kZXgucGhwOwoKCXN1cnZlc19uYw11IF87CgoJbG9jYXRpb24gLyB7CgkJ
IyBGaXJzdCBhdHR1bXB0IHRvIHN1cnZlIHJ1cXV1c3QgYXMgZmlsZSwgdGh1bgoJCSMgYXMgZGlyZWN0b3J5L
CB0aGVuIGZhbgwYmFjayB0byBkaXNwbGF5aW5nIGEgNDA0LgoJCXRYeV9maWx1cyAkdXJpICR1cmkvID00MD
Q7Cg19CgojCWxvY2F0aW9uIH4gXC5waHAKiHsKIwkgIHJvb3QgL3Zhci93d3cvaHRtbDsKIwkgIHRyeV9maWx
1cyAkdXJpID00MDQ7CiMJICBmYXN0Y2dpX3NwbG10X3BhdGhfaW5mbyBeKC4rXC5waHApKC8uKykkOwojCSAg
ZmFzdGNnaV9wYXNzIHVuaXg6L3Zhci9ydW4vcGhwL3BocC1mcG0uc29jazsKIwkgIGZhc3RjZ21faW5kZXgga
W5kZXgucGhwOwojCSAgZmFzdGNnaV9wYXJhbSBTQ1JJUFRfRk1MRU5BTUUgJGRvY3VtZW50X3Jvb3QkZmFzdG
NnaV9zY3JpcHRfbmFtZTsKIwkgIGluY2x1ZGUgZmFzdGNnaV9wYXJhbXM7CiMJICBmYXN0Y2dpX3J1YWRfdG1
tZW91dCAzMDA7CiMJfQoKfQoKc2VydmyIhsKCWxpc3R1biA4MCBkZWZhdWx0X3NlcnZlcjsKCWxpc3R1biBb
OjpdOjgwIGR1ZmF1bHRfc2VydmyOwoJIyBjaGFuZ2UgdGh1IG5leHQgbGluZSB0byB5b3VyiHNLcnZlcidzI
GZxZG4KCXN1cnZlc19uYw11IGVjMi5jb21wdXR1LmFtYXpbmF3cy5jb20gOwoJcmV0dXJuIDMwMiBodHRwcZ
ovLyRzZXJ2ZXJfbmFtZSRyZXF1ZXN0X3VyaTsKfQo=' | base64 -d - |
sudo tee /etc/nginx/sites-available/default > /dev/null
```

6. Let's make sure we got all three config files correct:

```
$ echo ce06df2c2a53f169cc48ad3e8f84d89e5c295c1c00886d31dee3606fdb58b92d  
/etc/nginx/sites-available/default | sha256sum -c
```

```
$ echo 2ebe327e944269f402267f0a1e06eff60723d3847329c7d233aa20012bb574f3  
/etc/nginx/snippets/self-signed.conf | sha256sum -c
```

```
echo cad3cb9841d0d8471f98470ae11b5e9501f5ebd563bfd582d6843f5b6d0fd354  
/etc/nginx/snippets/ssl-params.conf | sha256sum -c
```

7. Manually edit **/etc/nginx/sites-available/default** so you can configure it with your server's specific FQDN. (look for the **server_name** directive in the 2nd stanza)

```
$ sudo vi /etc/nginx/sites-available/default
```

8. Test that the configuration is correct.

```
$ sudo nginx -t
```

9. Reload (or restart) the server for the modified configuration to take effect.

```
$ sudo service nginx reload
```

10. Let's get a home page quickly.

```
$ sudo cp /var/www/html/index.nginx-debian.html /var/www/html/index.html
```

11. Point your browser to your server. Try https: first, then see what happens when you try http: