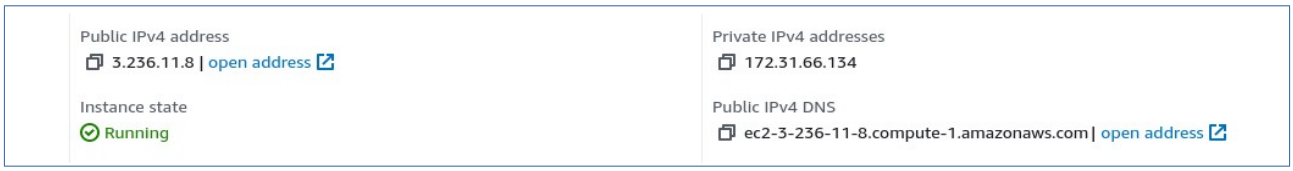
DAW SCREENSHOTS

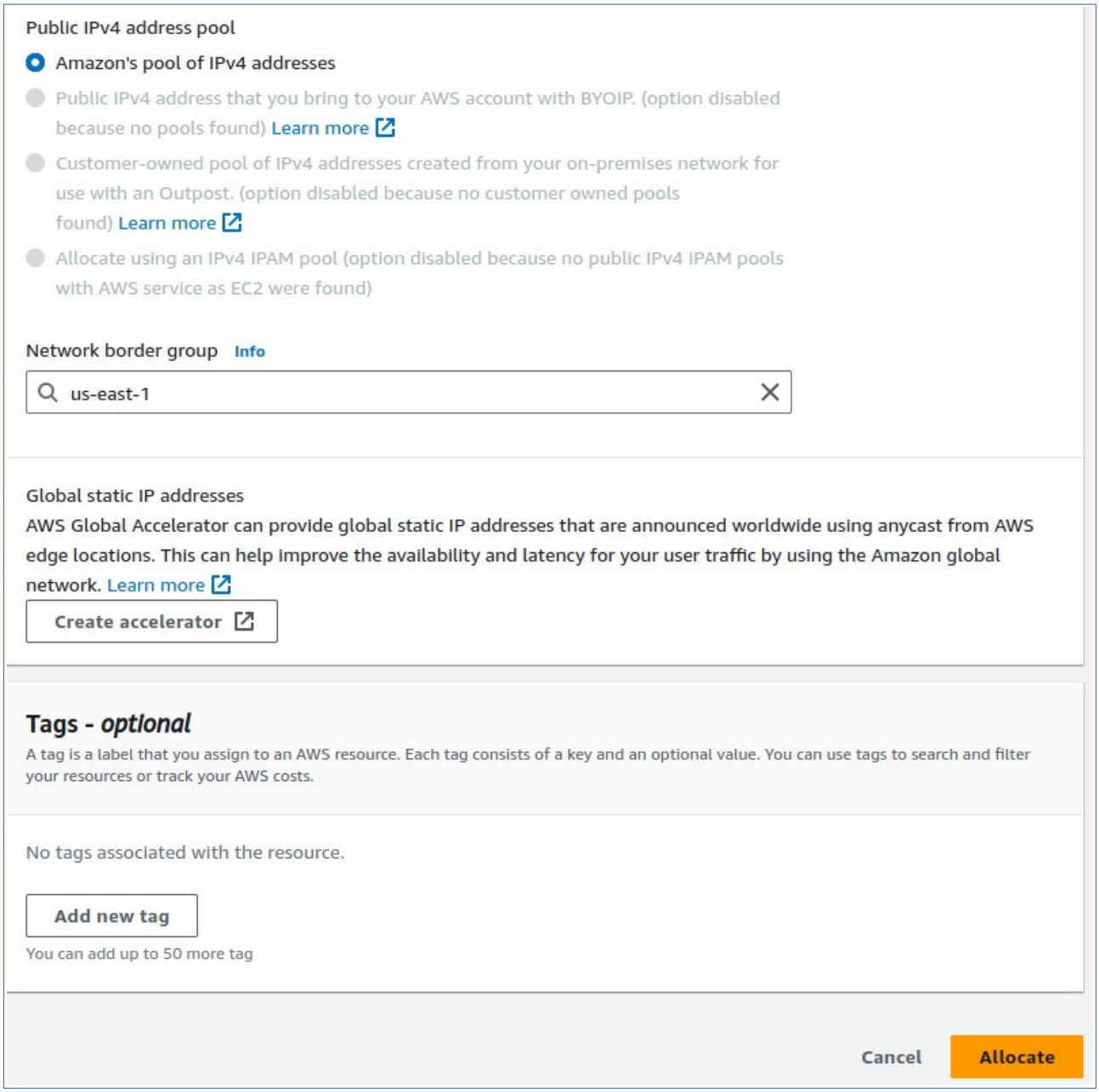
1 Basic configuration with AWS
In this first task, we are going to work with some network services and we will see how to configure them with AWS.

The EC2 instances that we created in unit 1 had a public IP address and a public DNS.
Include below a screenshot of the public IP and the DNS assigned to your web server.

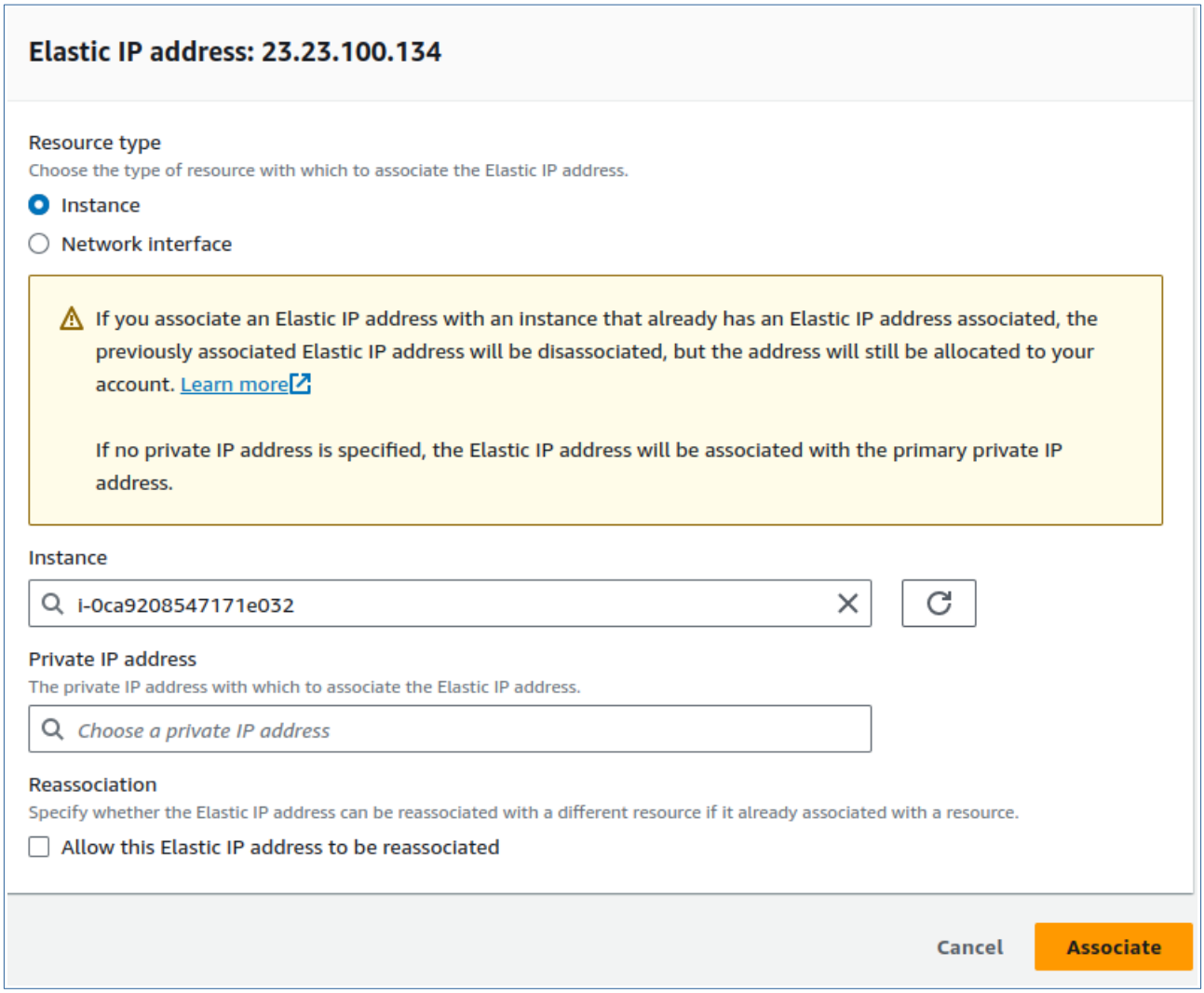


Nevertheless, this public IP and this DNS changed each time that we launched the learning lab. This could
not be used in a real environment, where we want to access a web application through its URL. In order to
assign an URL to a web service, we will assign it a DNS entry but even before that, we must be sure that
an IP has been assigned to the instance.
We are going to use two AWS services that will keep fixed both the IP and the DNS.
1.1 Elastic IP
Unlike the dynamic public IP assigned by AWS when you launch an EC2 instance, Elastic IPs do not change,
allowing you to maintain a consistent endpoint. We are going to assign an elastic IP address to our web
server (the first EC instance we created in unit one). Follow the steps:
• Go to “Elastic IPs” in the leftbar menu.
• “Allocate Elastic IP address”

• Select default option “Amazon’s pool of IPv4 addresses”
• Press “Allocate”



• Once it has been created (you have been assigned the public IP), you have to associate it to your EC2 instance. Select the IP and then press “Actions”, “Associate Elastic IP address”
• In the next screen, select “Resource type: Instance” and then choose the instance related to the web server (you should have the two instances you have created so far, select “MyServer001”
• Press “Associate”



Now it is time for you to check that the public IP has been really assigned to your instance.
• Connect to your instance with “EC2 Instance Connect” (default option)
• Once connected, launch the REST server with node expressRESTserver.js & (if you get an error message, check that you are in the right folder at the EC2 instance)

```
ubuntu@ip-172-31-66-134:~/DAW$ node expressRESTserver.js &
[1] 1945
ubuntu@ip-172-31-66-134:~/DAW$ Server is listening at http://localhost:3000
```

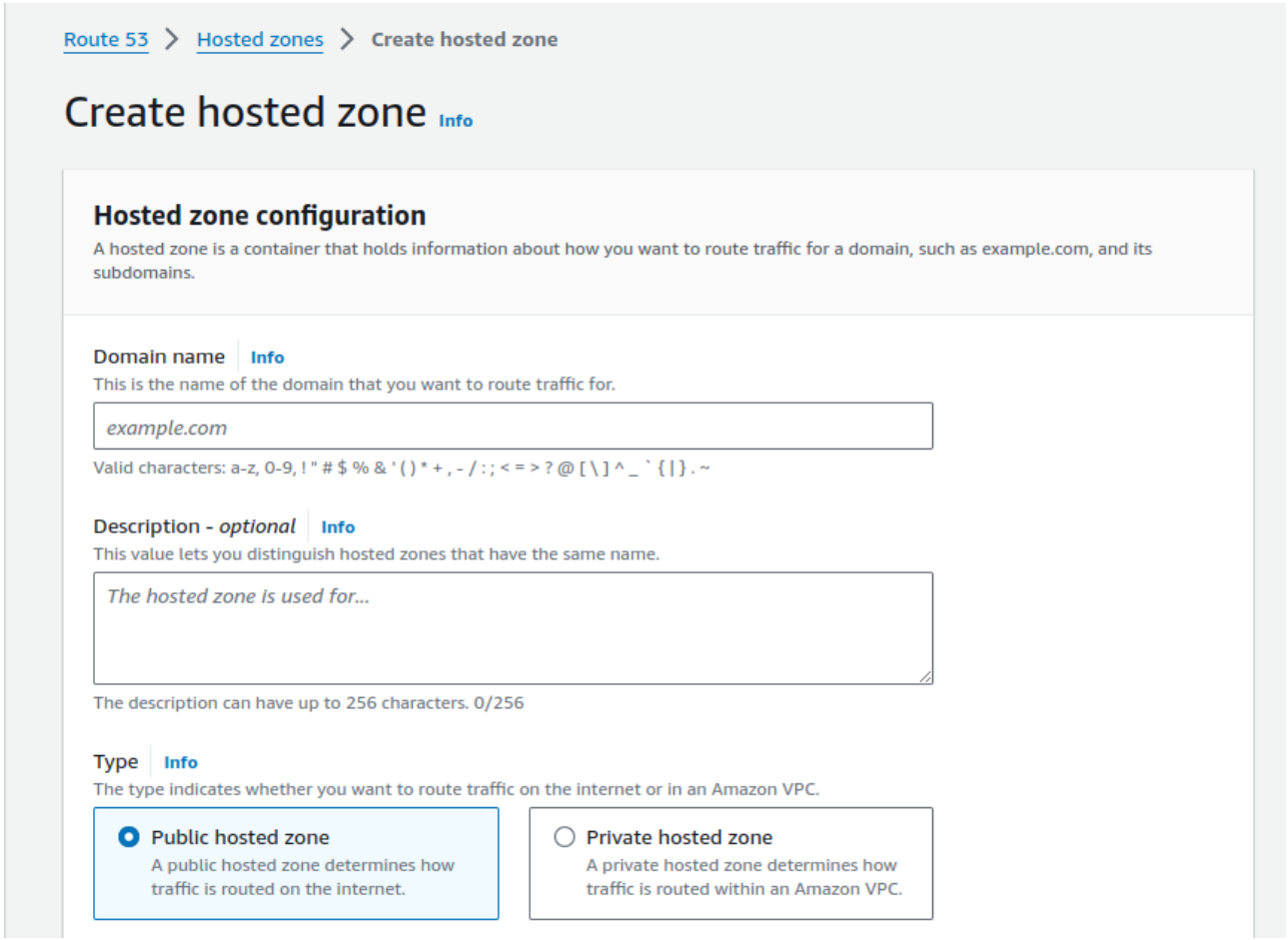
• Connect from your local machine, with your web browser, to the REST server http://public_ip:3000

DAW SCREENSHOTS

1.2 DNS configuration
DNS configuration is made with the Route S3 service. We are going to assign an url to the Webserver001
machine.
The url is going to be a subdomain of the dawmor.cloud domain. The url will have the following format
webserver001.username.dawmor.cloud where username is the name you have before iesmordefuentes.com in your email address (e.g. username:msierra for msierra@iesmordefuentes.com)
DNS service in AWS is called ROUTE 53. We are going to create a domain for our web pages. The domain
will be a subdomain of an existing domain managed by the module teachers.

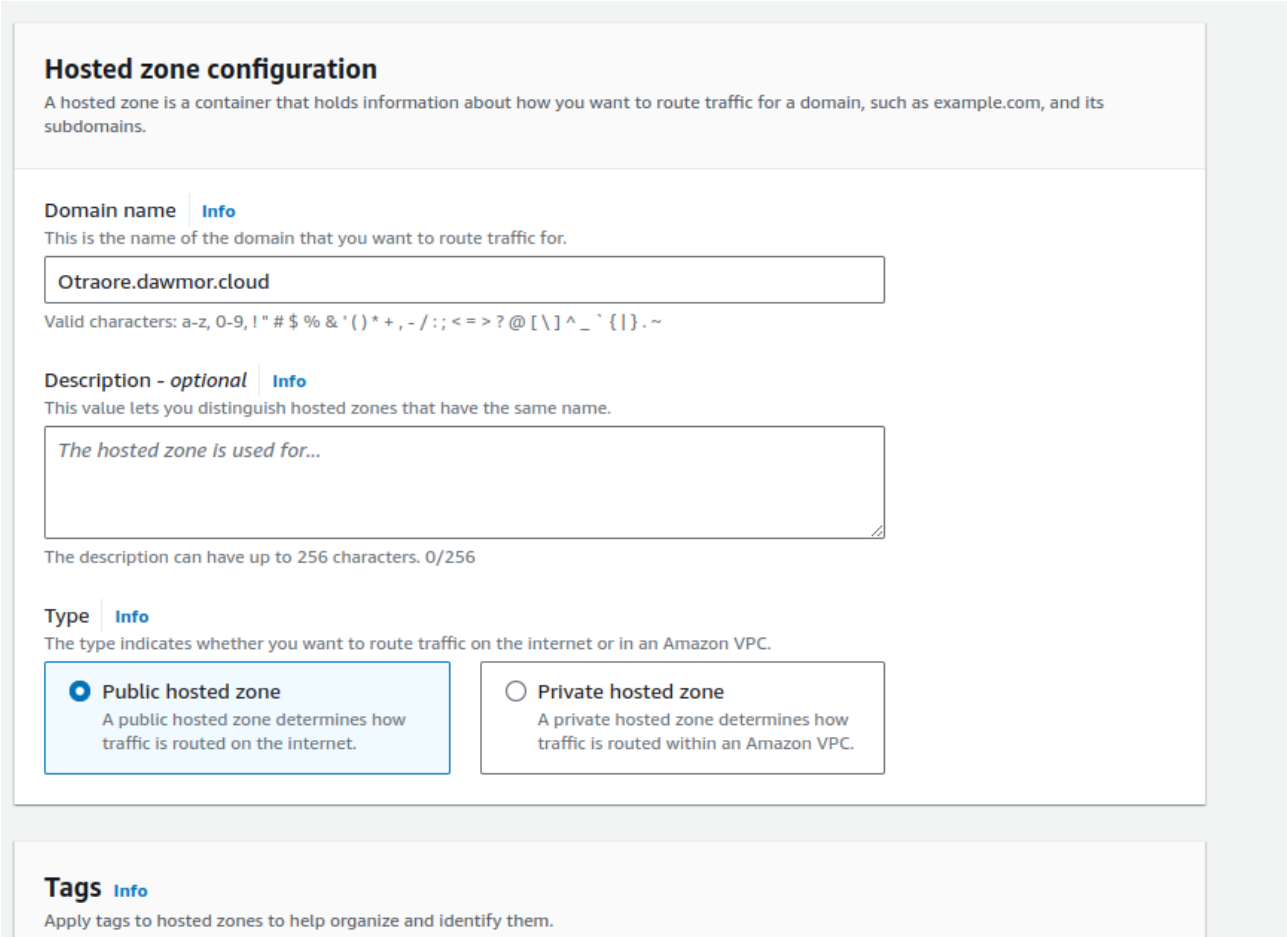
The domain that has been created is dawmor.cloud. The domain is managed by Hostinguer. The steps
that have been followed by the teachers are:
• We bought the domain (around 1 euro first year, a few more the upcoming ones…)
• We created the NS record dawmor.cloud at Route 53 (AWS). This record gave us the value of the servers for this subdomain.
• We took the nameservers and we included them in the Hostinger entry, saying that the name is managed by them, but it is resolved by the nameserver in the list (the AWS ones). So, from now on, the record we created is going to handle the subdomain dawmor.cloud

Instructions to create your own subdomain in “dawmor.cloud”
• Go to the Route 53 Service and press “DNS Management”, “Create hosted zone”

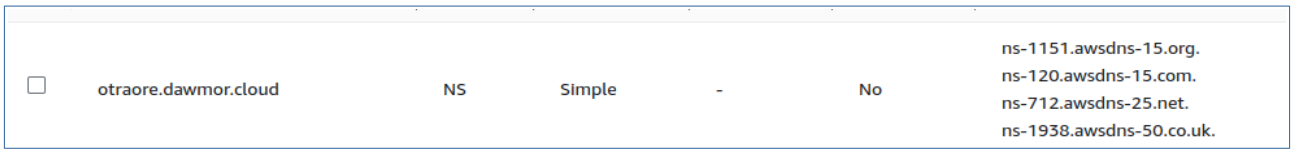


• In the domain name, enter the domain name. You must select username.dawmor.cloud (e.g. jhendrix.dawmor.cloud), it will be a public hosted zone, “Create hosted zone”

DAW SCREENSHOTS



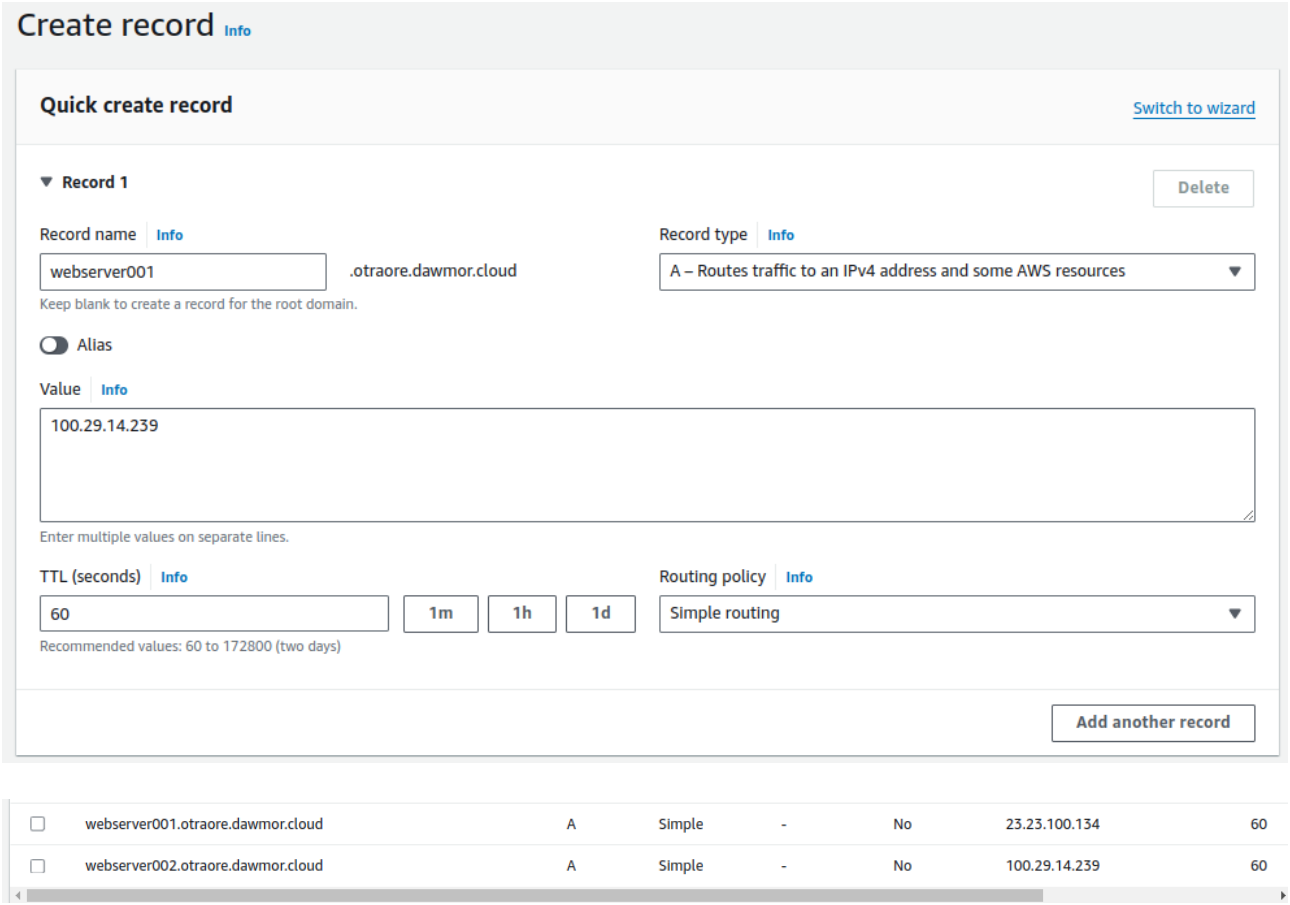
• Check the record “NS” NameServer / Name Servers for a hosted zone and copy the name of the servers in a .txt file. You must also include the domain name in that file and you must send it to the teacher course. The teacher will add a new NS record in the server domain in order to register that subdomain



• Now you have the “control” of the subdomain, you must create a new entry in the hosted zone. Type A, including the subdomain myserver001.username.dawmor.cloud (select a TTL of 60)

• Once the change has been made, you should be able to access the server through its DNS (test the expressRESTserver service.

DAW SCREENSHOTS

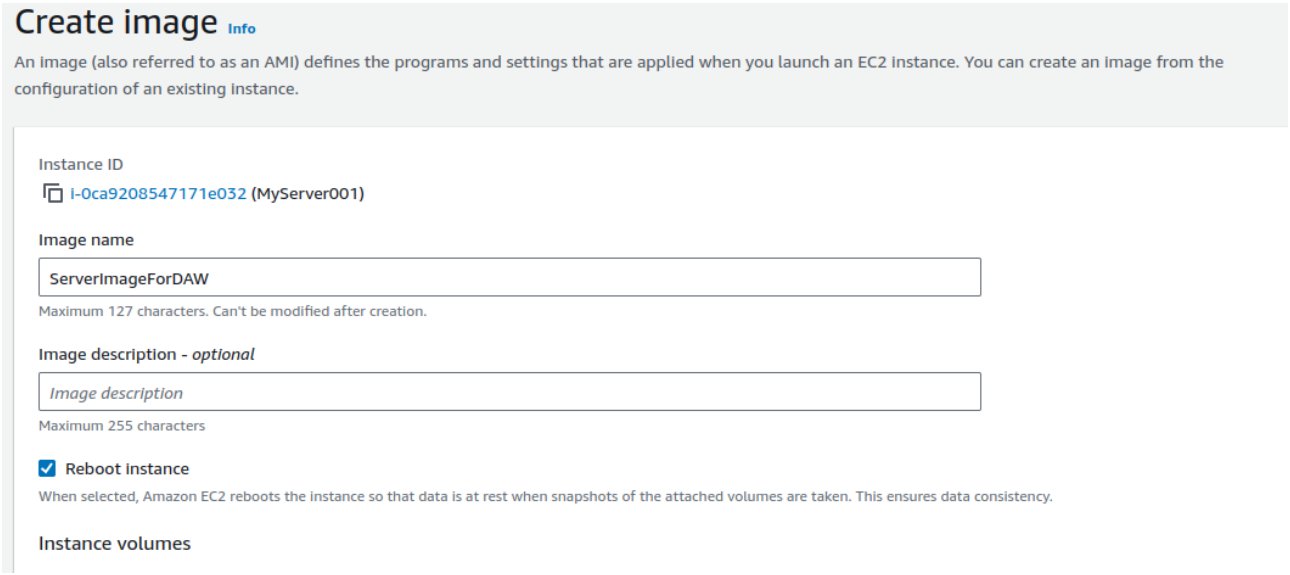


1.3 Creation of AMIs and DNS configuration
An Amazon Machine Image (AMI) in the context of AWS (Amazon Web Services) is a pre-configured virtual machine image that provides the information required to launch an instance (a virtual server) in the Amazon Elastic Compute Cloud (EC2). An AMI includes everything needed for an EC2 instance to run.

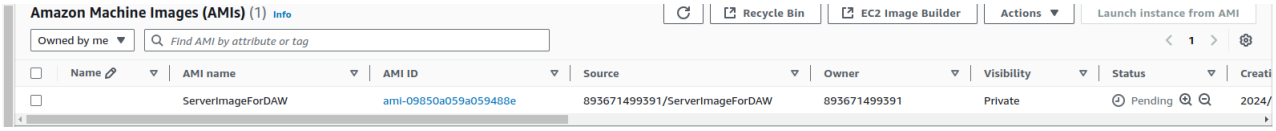
In this task, we are going to create an AMI from our server, we are going to create an EC2 instance from that AMI, we will launch it and we will assign it a new entry in our DNS subdomain (webserver002.username.dawmor.cloud)

• At the images list, select the MyServer001 instance, press “Actions” and select “Create Image”. You can call it ServerImageForDAW. “Create image”

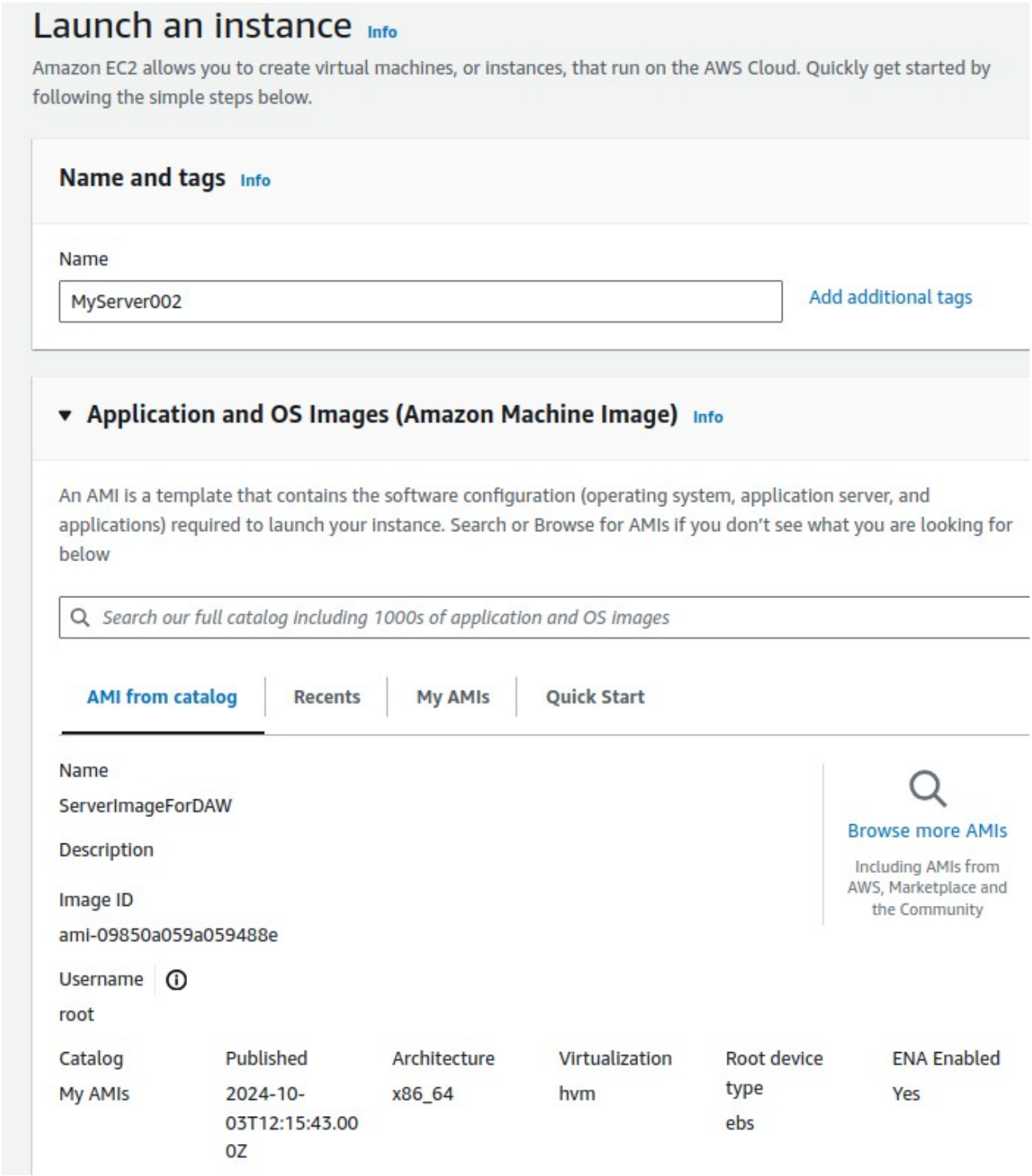
DAW SCREENSHOTS



• At the “Images/AMIs” menu, you can check whether the AMI creation has finished or not



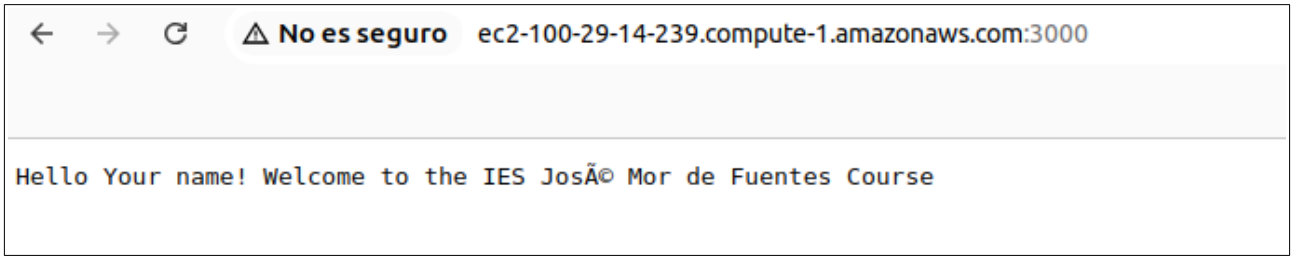
• Create one instance by selecting the image you have just created. It will be myserver002. You can select the same keys and security group as for the first server.



• Modify the corresponding simpleserver.js files for each server so they show the right name for each of them (instead of “Welcome to the IES José Mor de Fuentes” put something like “I am myserver001 / myserver002”

DAW SCREENSHOTS

```
 GNU nano 7.2                                  simpleserver.js
// DAW2
// Simple server.js
const http = require('http');
const hostname = '0.0.0.0';
const port = 3000;
const server = http.createServer((req, res) => {
res.statusCode = 200;
res.setHeader('Content-Type', 'text/plain');
res.end('Hello Your name! Welcome to the IES José Mor de Fuentes Course');
});
server.listen(port, hostname, () => {
console.log(`Server running at http://${hostname}:${port}/`);
});
```



1.4 SSH connection to the EC2 instance
We are connecting with the instance thanks to the “EC2 Instance Connect” application from AWS. But we can also connect from our terminals by using SSH. In order to do that, you must follow the instructions that appear in the “SSH client” tab after you press “Connect” in the EC2 instance list. These instrucions have been copied below for the teacher’s instance (you must follow the steps for your own machine, do not copy-paste these ones….)

•Open an SSH client.

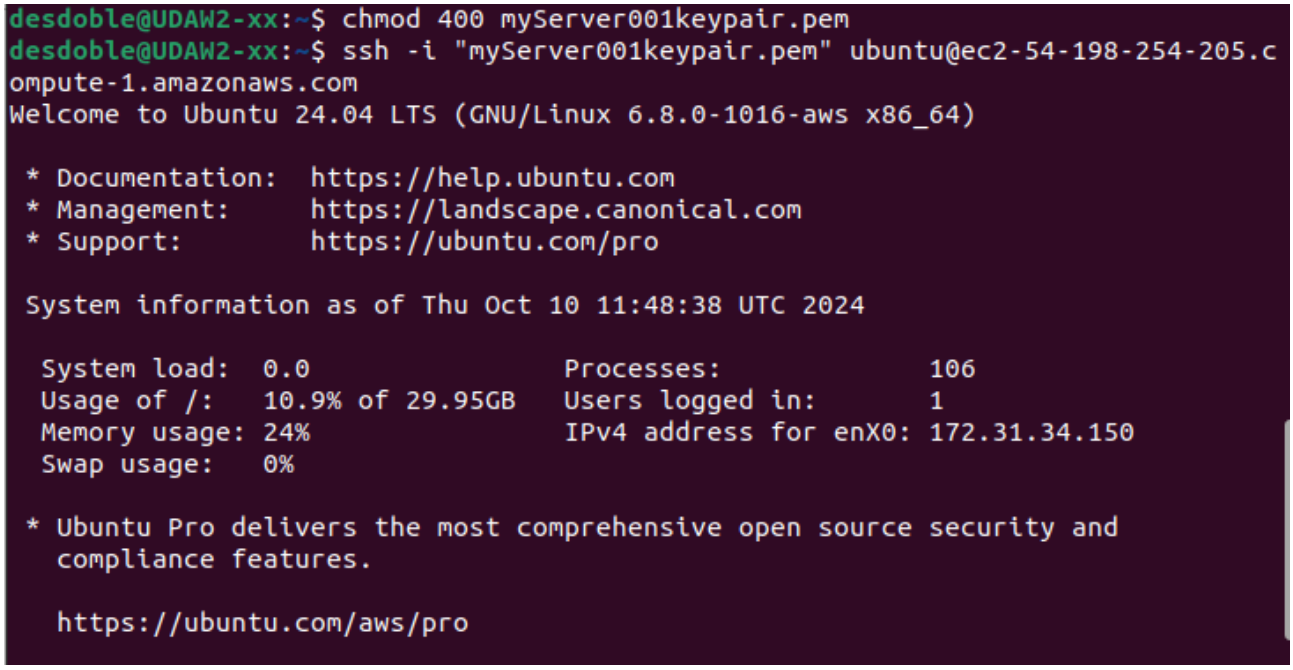
```
desdoble@UDAW2-xx:~$ ssh -i "myServer001keypair.pem" ubuntu@ec2-54-198-254-205.c
ompute-1.amazonaws.com
The authenticity of host 'ec2-54-198-254-205.compute-1.amazonaws.com (54.198.254
.205)' can't be established.
ED25519 key fingerprint is SHA256:jaNRQZXsTfenojNbh0IfXP5vvE2suJz+DB4j/E4QjPo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-198-254-205.compute-1.amazonaws.com' (ED25519
) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'myServer001keypair.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 "myServer001keypair.pem": bad permissions
ubuntu@ec2-54-198-254-205.compute-1.amazonaws.com: Permission denied (publickey)
```

DAW SCREENSHOTS

•Locate your private key file. The key used to launch this instance is myserver001keypair.pem
•Run this command, if necessary, to ensure your key is not publicly viewable.
- chmod 400 "myserver001keypair.pem"

•Connect to your instance using its Public DNS: ec2-52-73-245-85.compute-1.amazonaws.com
- Example:

ssh -i "myserver001keypair.pem" ubuntu@ec2-52-73-245-85.compute-1.amazonaws.com



NOTE: This option is posible because when you created the EC2 instance, you left one specific port open. Which one? Close the port and try the operation again. Is it possible? Is it possible to connect from the EC2 Instance Connect? Open it again and check again the connection.

DAW SCREENSHOTS

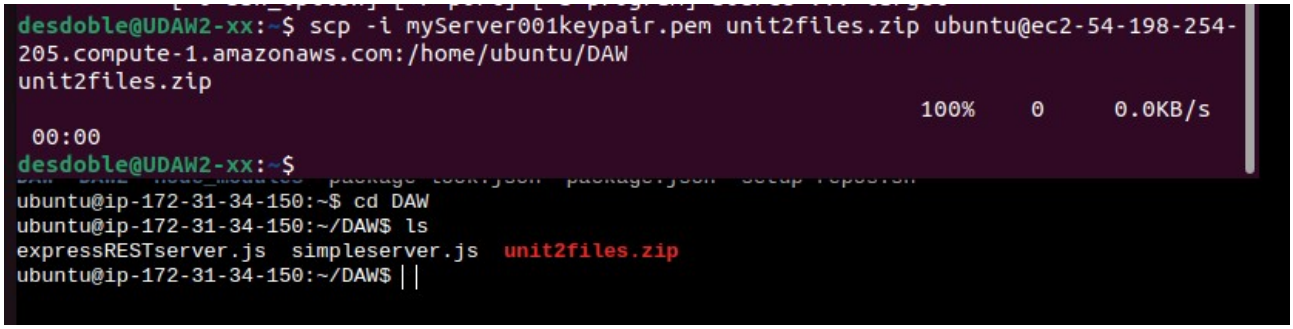
1.5 Copying files to the EC2 instance with SCP
SCP stands for Secure Copy Protocol, a network protocol used to securely transfer files between a local host and a remote host or between two remote hosts. SCP is based on SSH (Secure Shell), which provides encrypted communication to ensure that the data is transmitted securely.

You are going to copy a .zip file that you will need for the next task. The name of the file is unit2files.zip.
Be sure that folder /home/ubuntu/DAW2 is created.

The command to do the copy is scp -i myserver001keypair.pem unit2files.zip

ubuntu@PUBLIC_IP:FOR_YOUR_MACHINE:/home/ubuntu/DAW

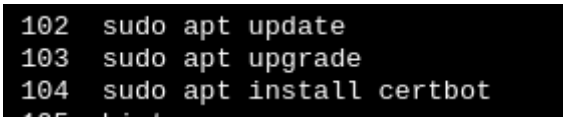
where you have to specify your own .pem file, and the PUBLIC_IP or DNS of your server001 machine.



1.6 HTTPS server
Certbot from Let’s Encrypt is the most popular automatic tool to obtain free certificates from Let’s Encrypt. We are going to create a certificate and use it to make our own simpleserver secure.

Install Certbot

sudo apt update
sudo apt upgrade
sudo apt install certbot



Generate a certificate with the standalone Certbot mode (no need for a webserver as Nginx or Apache to run). Change yourdomain.com with the name of your subdomain. You will be asked for your email and you will also have to accept the terms of use.

sudo certbot certonly --standalone -d yourdomain,
the ip can changue then you must put the new

DAW SCREENSHOTS

```
ubuntu@ip-172-31-34-150:~$ sudo certbot certonly --standalone -d webserver002.otraore.dawmor.cloud
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for webserver002.otraore.dawmor.cloud

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/webserver002.otraore.dawmor.cloud/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/webserver002.otraore.dawmor.cloud/privkey.pem
This certificate expires on 2025-01-13.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.
We were unable to subscribe you the EFF mailing list because your e-mail address appears to be invalid. Yo
u can try again later by visiting https://act.eff.org.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
```

Check the certificates that have been generated at (have a look at them and check they exist):

/etc/letsencrypt/live/yourdomain/fullchain.pem # Certificate
/etc/letsencrypt/live/yourdomain/privkey.pem # Private key

```
ubuntu@ip-172-31-34-150:~$ sudo ls /etc/letsencrypt/live/webserver002.otraore.dawmor.cloud/
README  cert.pem  chain.pem  fullchain.pem  privkey.pem
```

Create a file named simplesecureserver.js with the following code (check the routes for the certificates)

```
  GNU nano 7.2                                simplesecureserver.js
const https = require('https');
const fs = require('fs');
const hostname = '0.0.0.0';
const port = 443;
const server = https.createServer(
{
 cert: fs.readFileSync('/etc/letsencrypt/live/webserver002.otraore.dawmor.cloud.com/fullchain.pem'),
 key: fs.readFileSync('/etc/letsencrypt/live/webserver002.otraore.dawmor.cloud.com/privkey.pem')
},
(req, res) => {
 res.statusCode = 200;
 res.setHeader('Content-Type', 'text/plain');
 res.end('Hello DAW2, I am a Secure Server');
});
server.listen(port, hostname, () => {
 console.log(`Server running at https://${hostname}:${port}/`);
});
```

Launch the server and check you can access to it with https.
Look for the information of the certificate with your web browser. Include a screenshot.
Compare the simpleserver.js code and the simplesecureserver.js code.

Explain the differences between them.

In the simpleserver i havent the certificate, then the acces is httm en vez del https del simplesecureserver, then that last is secured because have certificate

DAW SCREENSHOTS

1.7 FTP server
• Install vsftpd

```
ubuntu@ip-172-31-34-150:~/DAW$ sudo apt install vsftpd
```

• Configure vsftpd. Configuration file is at /etc/vsftpd.conf. Check the following configuration options (it is highly recommended to make a copy of the file before this process)
▪ enable local users:
local_enable=YES
▪ allow users to upload files:
write_enable=YES (allow to upload files)
▪ ensure local users are restricted to their own home directory
chroot_local_user=YES
▪ enable pasive mode (new entries in the file)
pasv_min_port=40000
pasv_max_port=50000
pasv_address=your_server_ip
• Create FTP users
◦ sudo adduser ftpuser

```
ubuntu@ip-172-31-34-150:~/DAW$ sudo adduser ftpuser
info: Adding user `ftpuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `ftpuser' (1001) ...
info: Adding new user `ftpuser' (1001) with group `ftpuser (1001)' ...
info: Creating home directory `/home/ftpuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
No password has been supplied.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser
Enter the new value, or press ENTER for the default
        Full Name []: FTPuser
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Set Permissions

```
sudo chown ftpuser:ftpuser /home/ftpuser
sudo chmod a-w /home/ftpuser
sudo mkdir /home/ftpuser/uploads
sudo chown ftpuser:ftpuser /home/ftpuser/uploads
sudo chmod 755 /home/ftpuser/uploads
```

DAW SCREENSHOTS

Restart vsftpd. You can check the status replacing the word restart with status
- sudo systemctl restart vsftpd

Configure ports
- Check that ports 21 and 40000 to 50000 can be used.

Use a ftp client as Filezilla to connect to the server
- Download filezilla at your local ubuntu machine

(esta captura es del examen practico por eso tiene otro ip)