

## 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.

Public IPv4 address 3.236.11.8   <a href="#">open address</a>	Private IPv4 addresses 172.31.66.134
Instance state <span>Running</span>	Public IPv4 DNS ec2-3-236-11-8.compute-1.amazonaws.com   <a href="#">open address</a>

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.

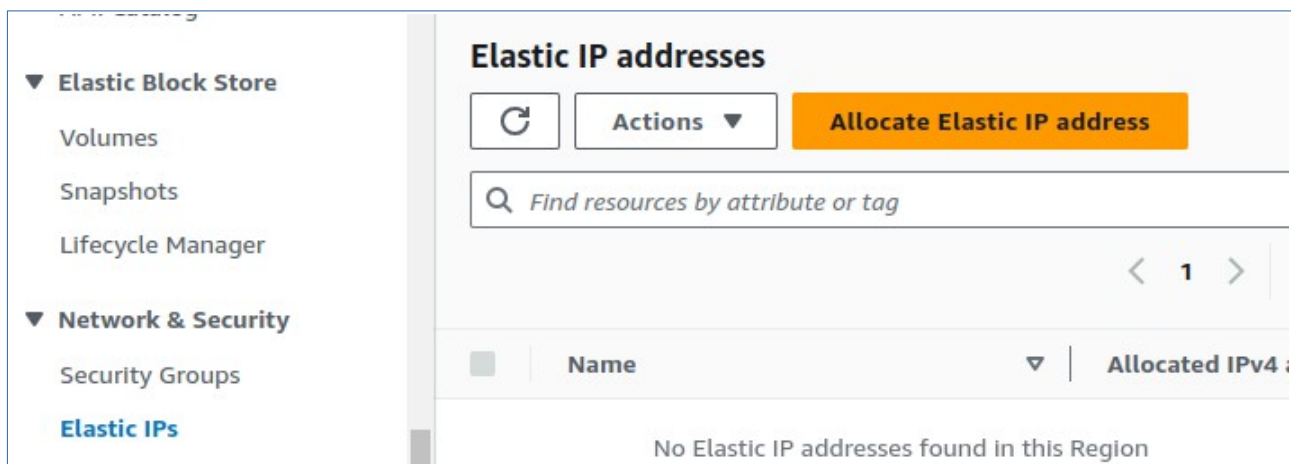
#### 1.1 Elastic IP

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”

## DAW SCREENSHOTS

**Public IPv4 address pool**

- ☒ Amazon's pool of IPv4 addresses
- ☐ Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) [Learn more](#)
- ☐ Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) [Learn more](#)
- ☐ Allocate using an IPv4 IPAM pool (option disabled because no public IPv4 IPAM pools with AWS service as EC2 were found)

**Network border group** [Info](#)

**Global static IP addresses**

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

Create accelerator

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

You can add up to 50 more tag

Cancel

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”

## DAW SCREENSHOTS

**Elastic IP address: 23.23.100.134**

**Resource type**  
Choose the type of resource with which to associate the Elastic IP address.

☒ Instance  
☐ Network interface

**⚠** If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. [Learn more](#)

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

**Instance**

**Private IP address**  
The private IP address with which to associate the Elastic IP address.

**Reassociation**  
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.  
☐ Allow this Elastic IP address to be reassociated

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](http://public_ip:3000)



## DAW SCREENSHOTS

### 1.2 DNS configuration

DNS configuration is made with the Route 53 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.

2

Web Applications Deployment . 2ºDAW. IES Mor de Fuentes 2024-2025

The domain that has been created is dawmor.cloud. The domain is managed by Hostinger. 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”

## DAW SCREENSHOTS

[Route 53](#) > [Hosted zones](#) > Create hosted zone

### Create hosted zone [Info](#)

#### Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

**Domain name** [Info](#)

This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! " # \$ % & ' ( ) \* + , - / : ; < = > ? @ [ \ ] ^ \_ ` { | } . ~

**Description - optional** [Info](#)

This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 0/256

**Type** [Info](#)

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

☒ **Public hosted zone**  
A public hosted zone determines how traffic is routed on the internet.

☐ **Private hosted zone**  
A private hosted zone determines how traffic is routed within an Amazon VPC.

- 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

### Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name

Info

This is the name of the domain that you want to route traffic for.

Otraore.dawmor.cloud

Valid characters: a-z, 0-9, ! " # \$ % & ' ( ) \* + , - / : ; < = > ? @ [ \ ] ^ \_ ` { | } . ~

Description - optional

Info

This value lets you distinguish hosted zones that have the same name.

The hosted zone is used for...

The description can have up to 256 characters. 0/256

Type

Info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

☒ Public hosted zone

A public hosted zone determines how traffic is routed on the internet.

☐ Private hosted zone

A private hosted zone determines how traffic is routed within an Amazon VPC.

### Tags

Info

Apply tags to hosted zones to help organize and identify them.

- 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

<input type="checkbox"/>	otraore.dawmor.cloud	NS	Simple	-	No	ns-1151.awsdns-15.org. ns-120.awsdns-15.com. ns-712.awsdns-25.net. ns-1938.awsdns-50.co.uk.
--------------------------	----------------------	----	--------	---	----	--

- 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

### Create record [Info](#)

#### Quick create record [Switch to wizard](#)

▼ Record 1 [Delete](#)

Record name [Info](#)

webserver001.otraore.dawmor.cloud

Keep blank to create a record for the root domain.

Record type [Info](#)

A – Routes traffic to an IPv4 address and some AWS resources ▼

☐ Alias

Value [Info](#)

100.29.14.239

Enter multiple values on separate lines.

TTL (seconds) [Info](#)

60

1m1h1d

Recommended values: 60 to 172800 (two days)

Routing policy [Info](#)

Simple routing ▼

[Add another record](#)

<input type="checkbox"/>	webserver001.otraore.dawmor.cloud	A	Simple	-	No	23.23.100.134	60
<input type="checkbox"/>	webserver002.otraore.dawmor.cloud	A	Simple	-	No	100.29.14.239	60

### 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

### Create image [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID

 **I-0ca9208547171e032** (MyServer001)

Image name

ServerImageForDAW

Maximum 127 characters. Can't be modified after creation.

Image description - *optional*

Image description

Maximum 255 characters

☒ **Reboot instance**

When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

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

Amazon Machine Images (AMIs) (1) [Info](#)

Owned by me ▾

Find AMI by attribute or tag

↻

Recycle Bin

EC2 Image Builder

Actions ▾

Launch instance from AMI

< 1 >

⚙

<input type="checkbox"/>	Name <a href="#">↗</a>	AMI name	AMI ID	Source	Owner	Visibility	Status	Created
<input type="checkbox"/>	ServerImageForDAW		<a href="#">ami-09850a059a059488e</a>	893671499391/ServerImageForDAW	893671499391	Private	<div><div>⌚</div><div>Pending</div><div><div>🔍</div><div>🔍</div></div></div>	2024/

- 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.



## DAW SCREENSHOTS

### 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

[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

[AMI from catalog](#)[Recents](#)[My AMIs](#)[Quick Start](#)

Name

ServerImageForDAW

Description

Image ID

ami-09850a059a059488e

Username ⓘ

root

Catalog

My AMIs

Published

2024-10-03T12:15:43.000Z

Architecture

x86\_64

Virtualization

hvm

Root device

type  
ebs

ENA Enabled

Yes

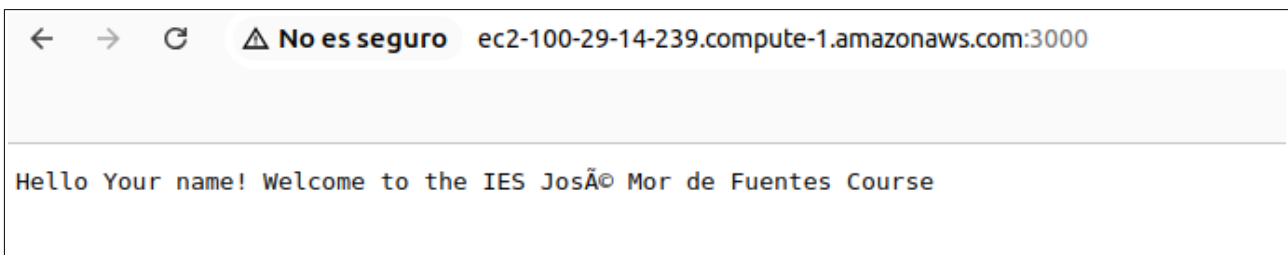
[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

- 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 instructions 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"`

```
desdoble@UDAW2-xx:~$ 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
```

```
desdoble@UDAW2-xx:~$ chmod 400 myServer001keypair.pem
desdoble@UDAW2-xx:~$ ssh -i "myServer001keypair.pem" ubuntu@ec2-54-198-254-205.c
ompute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Oct 10 11:48:38 UTC 2024

System load:  0.0               Processes:            106
Usage of /:   10.9% of 29.95GB   Users logged in:     1
Memory usage: 24%              IPv4 address for enX0: 172.31.34.150
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro
```

NOTE: This option is possible 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.

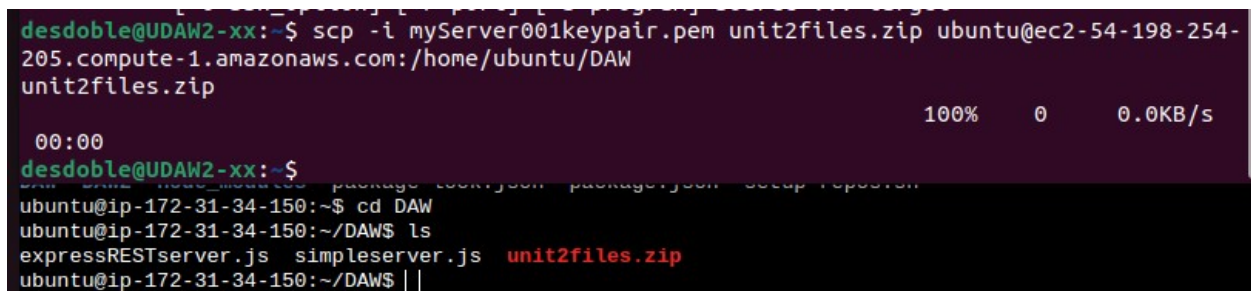
You are going to copy a .zip file that you will need for the next task. The name of the file is unit2files.zip.

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.



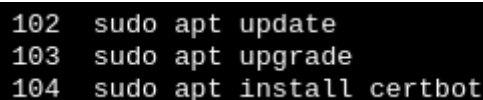
```
desdoble@UDAW2-xx:~$ scp -i myServer001keypair.pem unit2files.zip ubuntu@ec2-54-198-254-205.compute-1.amazonaws.com:/home/ubuntu/DAW
unit2files.zip
100% 0 0.0KB/s
00:00
desdoble@UDAW2-xx:~$
ubuntu@ip-172-31-34-150:~$ cd DAW
ubuntu@ip-172-31-34-150:~/DAW$ ls
expressRESTserver.js simpleserver.js unit2files.zip
ubuntu@ip-172-31-34-150:~/DAW$ |
```

### 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
```



```
102 sudo apt update
103 sudo apt upgrade
104 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 change 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. You
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 http 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)

