PROJECT: AMAZON AWS EC2 INSTANCE WEBSITE HOSTING

Author: Nina Takkunen 2025

CONTENTS

Goal

- Explore different cloud vendors.
- Set up a cloud server.
- Host a simple website.
- Gain experience and understanding of cloud computing.

Tools

- Amazon Web Services (AWS EC2)
- Amazon Linux 2023
- PuTTYGEN (key generator)
- PuTTY (SSH connection)
- SCP (Secure Copy for file transfer)
- VSC (HTML + CSS)
- Windows OS

MICROSOFT AZURE, GCP

AKA GOOGLE CLOUD PLATFORM

OR

AWS

AKA AMAZON WEB SERVICES

?

OPTIONS

Microsoft Azure
1 month trial 200 credits

GCP
3 month trial 300 credits

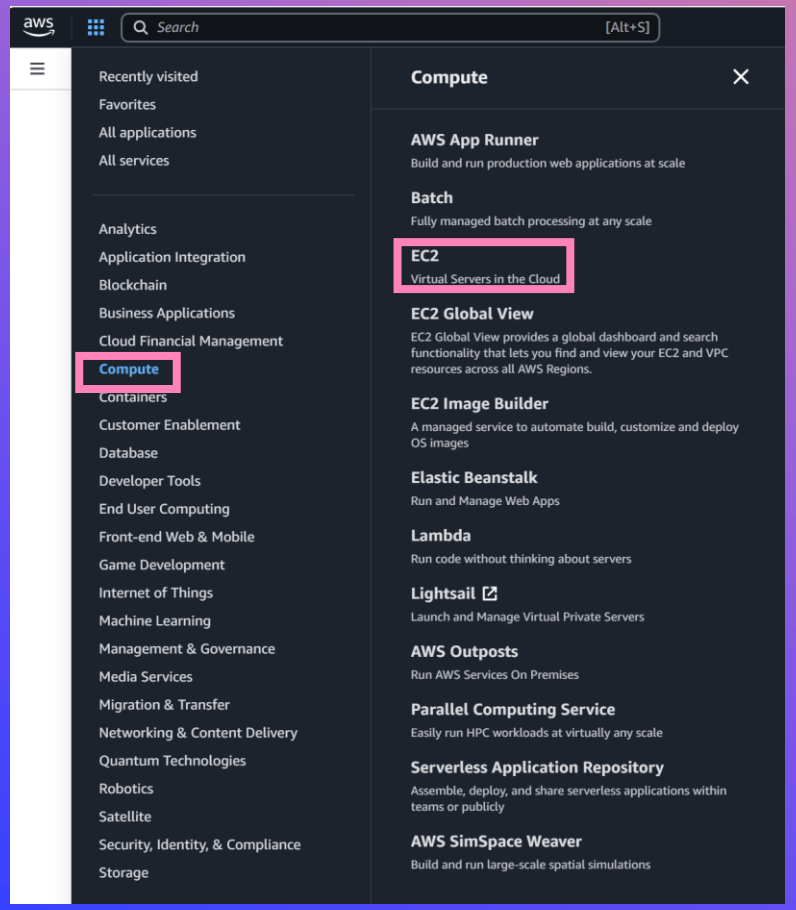
Amazon AWS
12 month free tier

SIGNING UP & SETTING UP

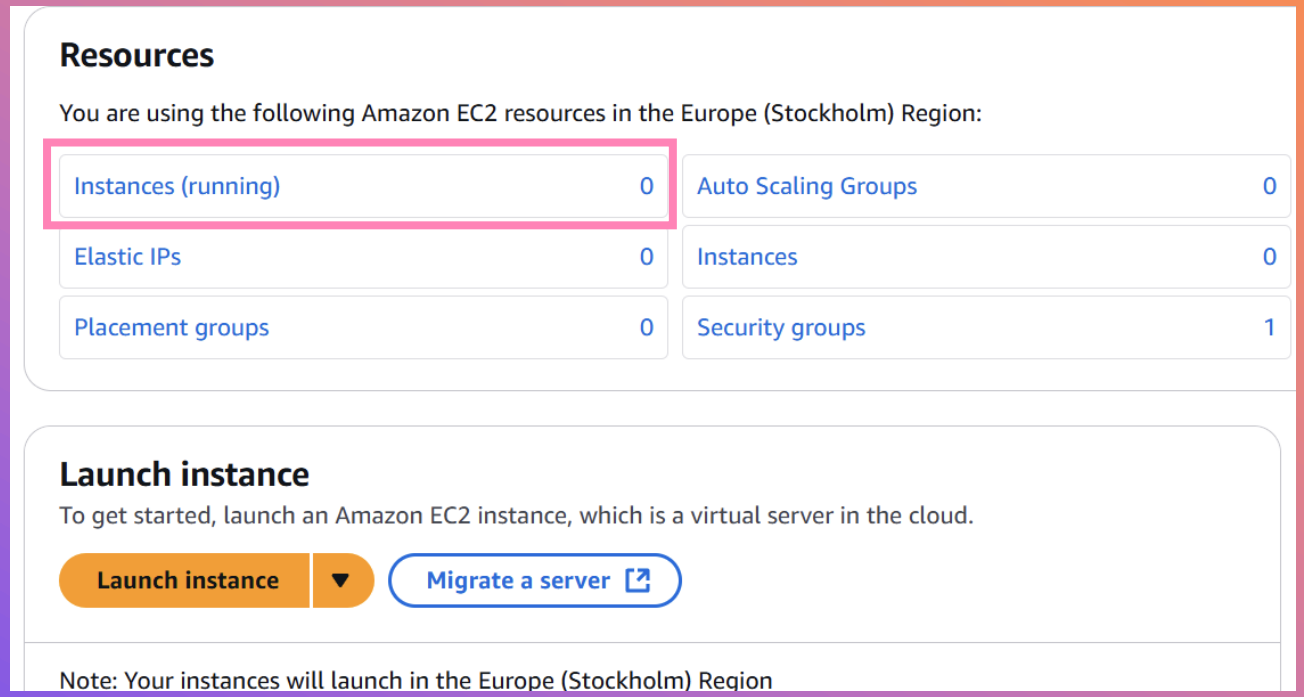
- I signed in and created an AWS account.
- I browsed documentaries, demos and tutorials and watched several Youtube videos!

NOW LET'S CREATE AN EC2 INSTANCE!

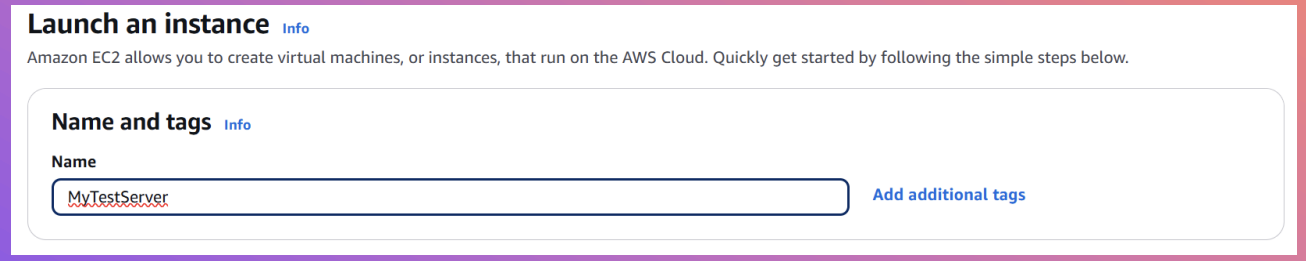
Select Console ->
Compute -> EC2



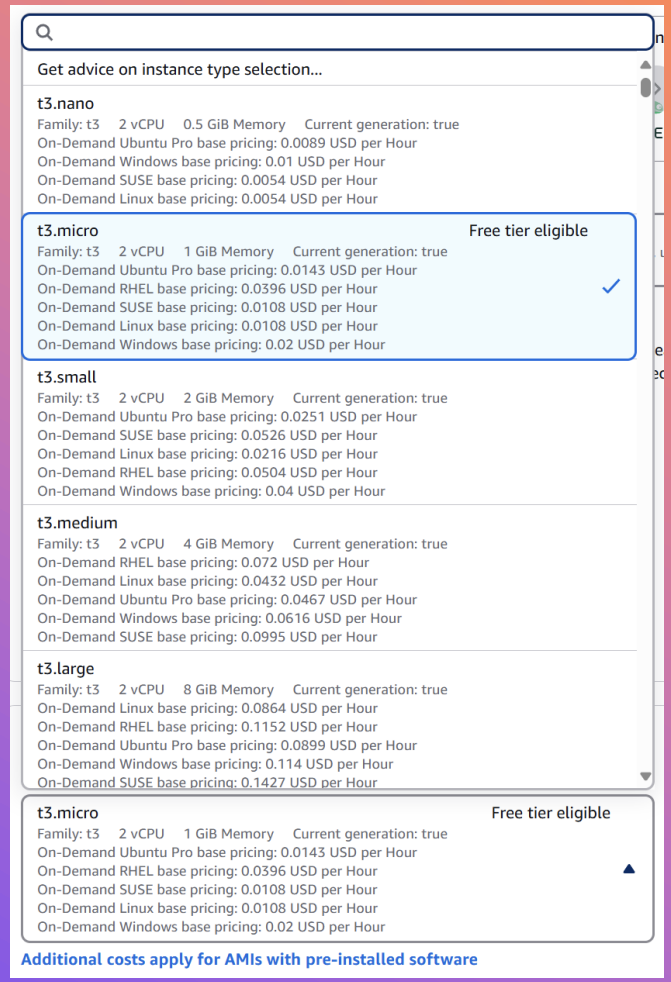
There are no instances running,
so let's create and launch one



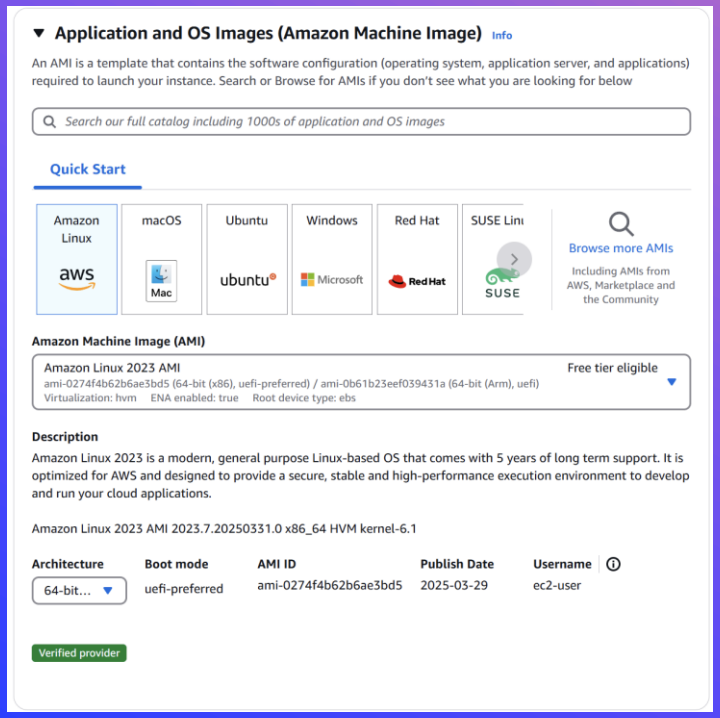
First I named my server



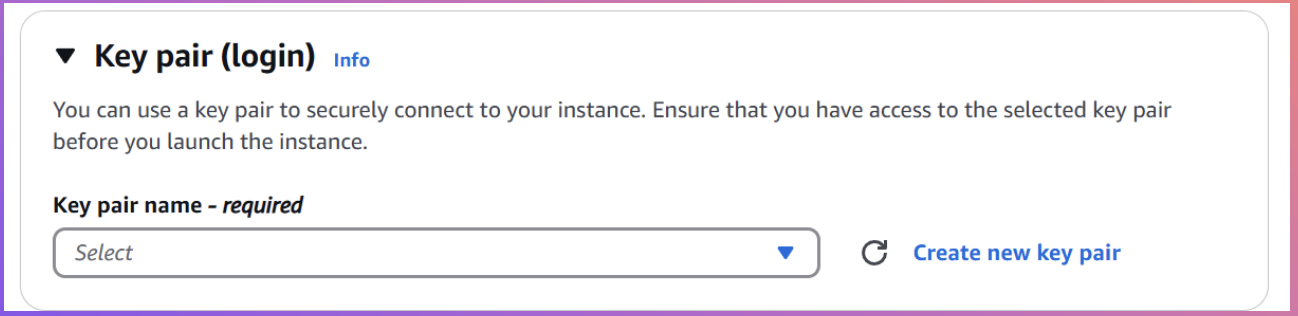
As an instance type I went with free tier eligible t3.micro

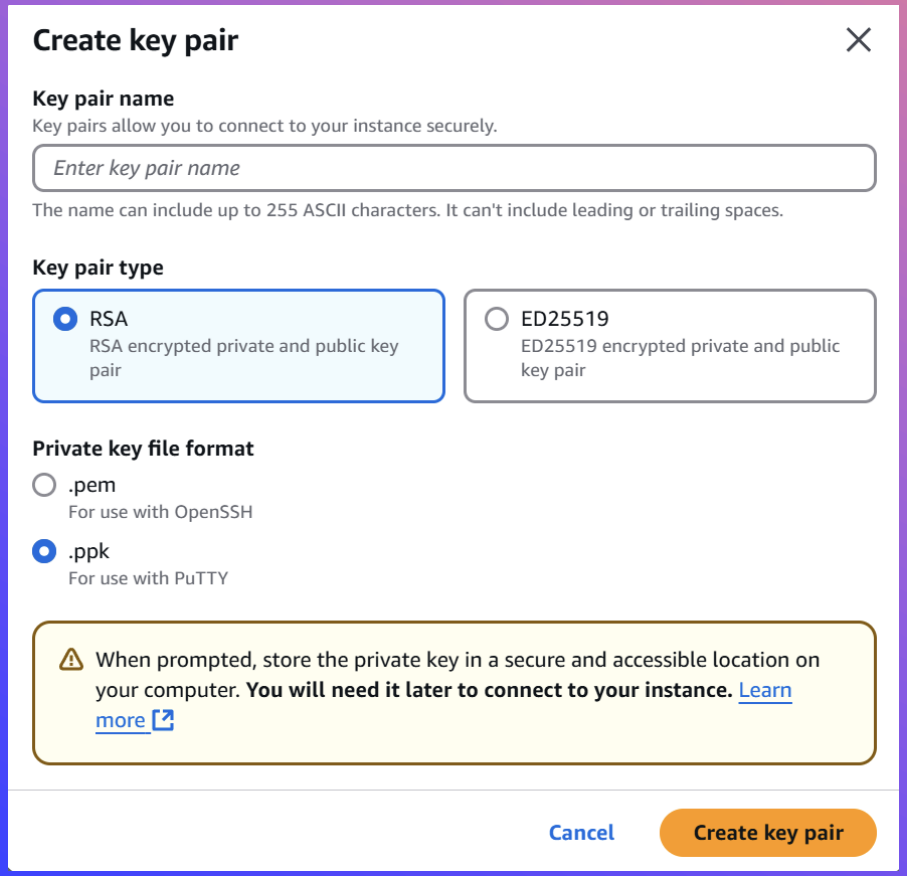


Then chose desired operating system. I chose Amazon Linux, since it is free tier eligible



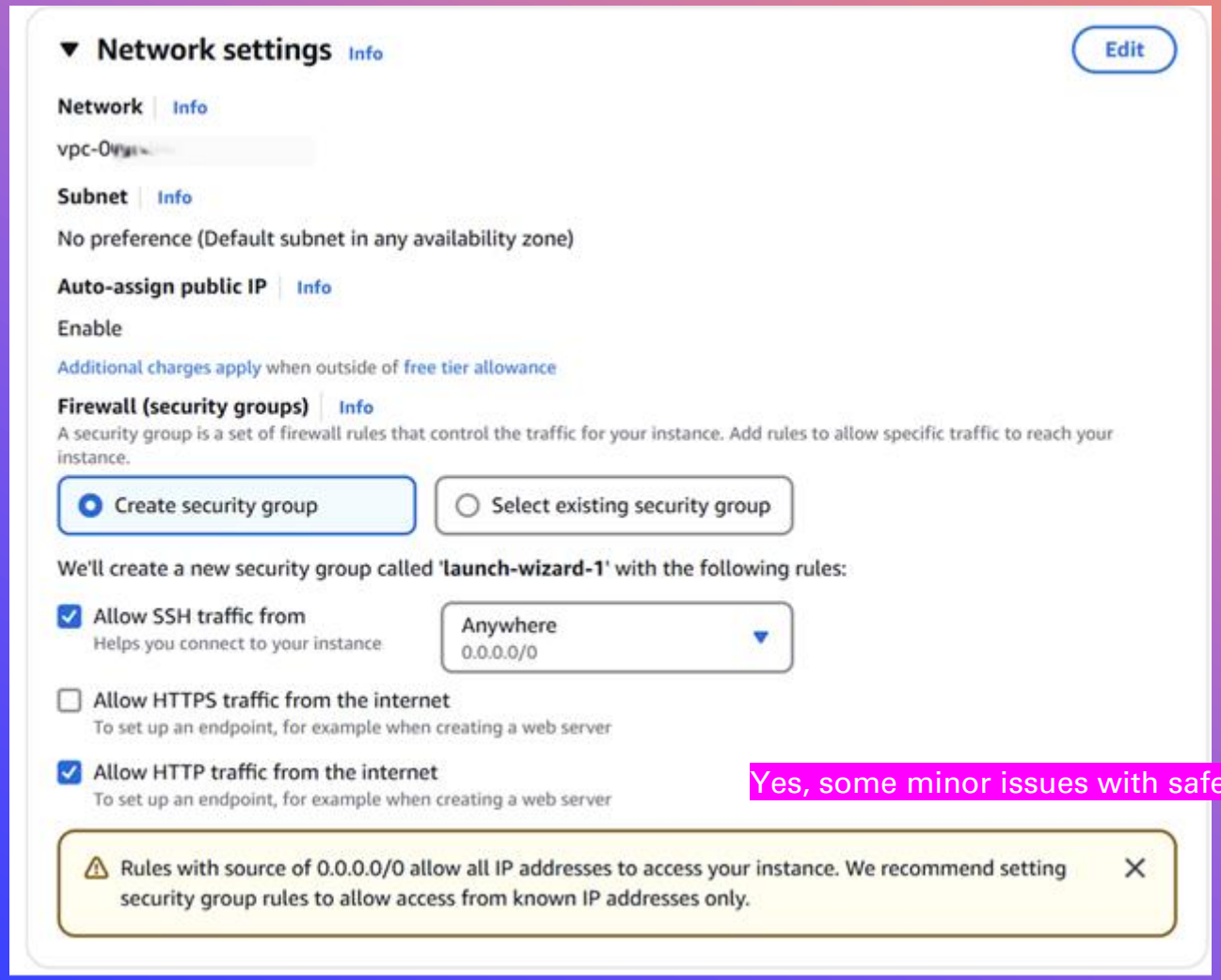
Now I needed to create key pair for login





Since I’m using Windows, I chose .ppk format to use with PuTTY

Let’s define security rules



I wanted to host a website from this server, so I had to allow some HHTP traffic

Yes, some minor issues with safety here..!

Summary

Number of instances | Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0274f4b62b6ae3bd5

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance | Preview code

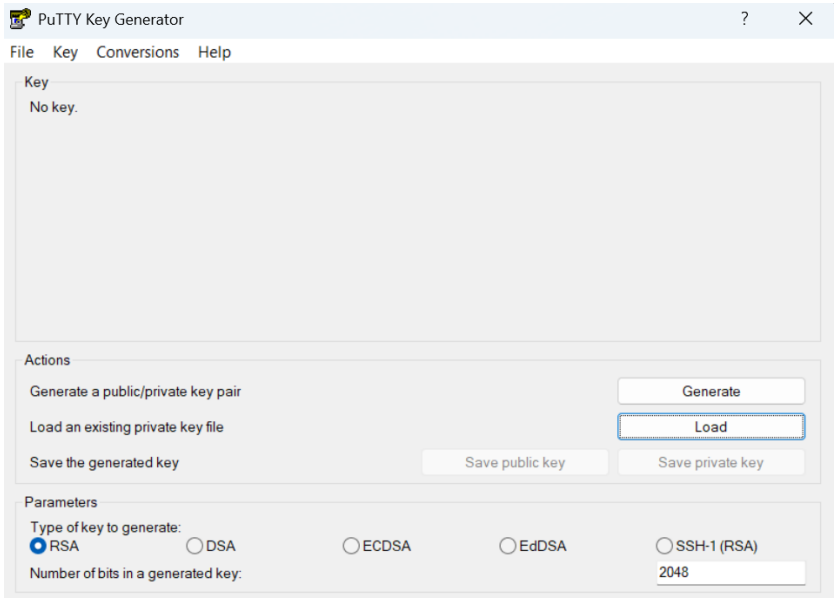
Ready to board and launch!

Success
Successfully initiated launch of instance (i-09f374bc9519e0a9b)

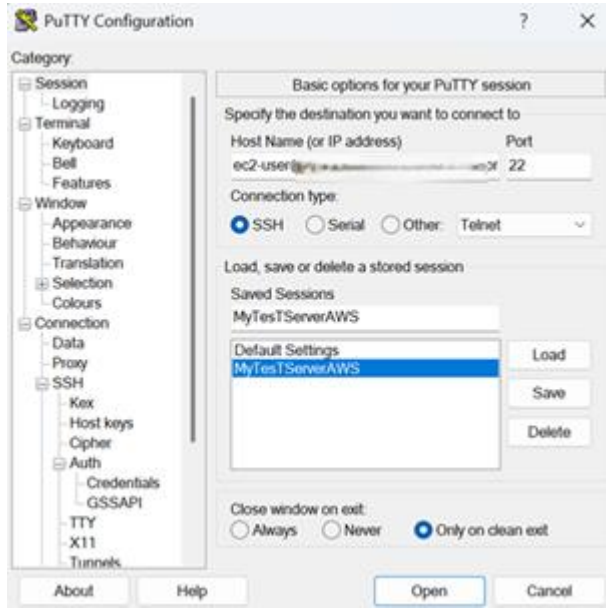
Name	Instance ID	Instance state	Instance type	Status check
MyTestServer	i-09f374bc9519e0a9b	Running	t3.micro	Initializing

Name	Instance ID	Instance state	Instance type	Status check
MyTestServer	i-09f374bc9519e0a9b	Running	t3.micro	3/3 checks passec

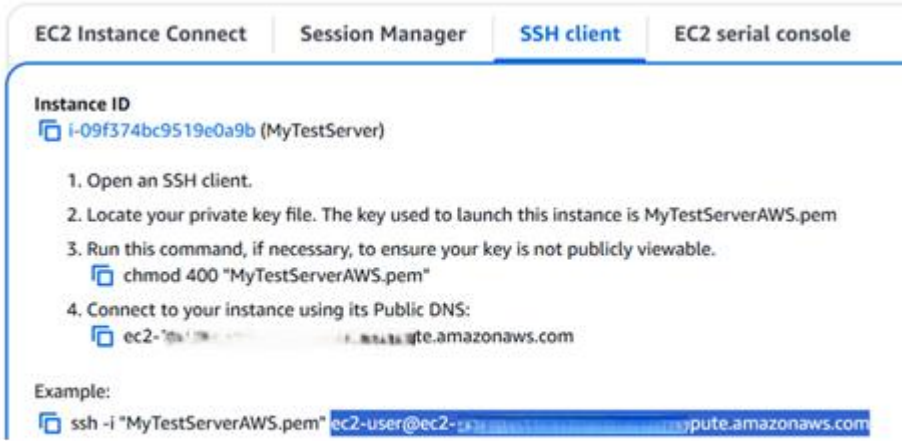
EC2 instance is ready to go, let’s SSH our way in



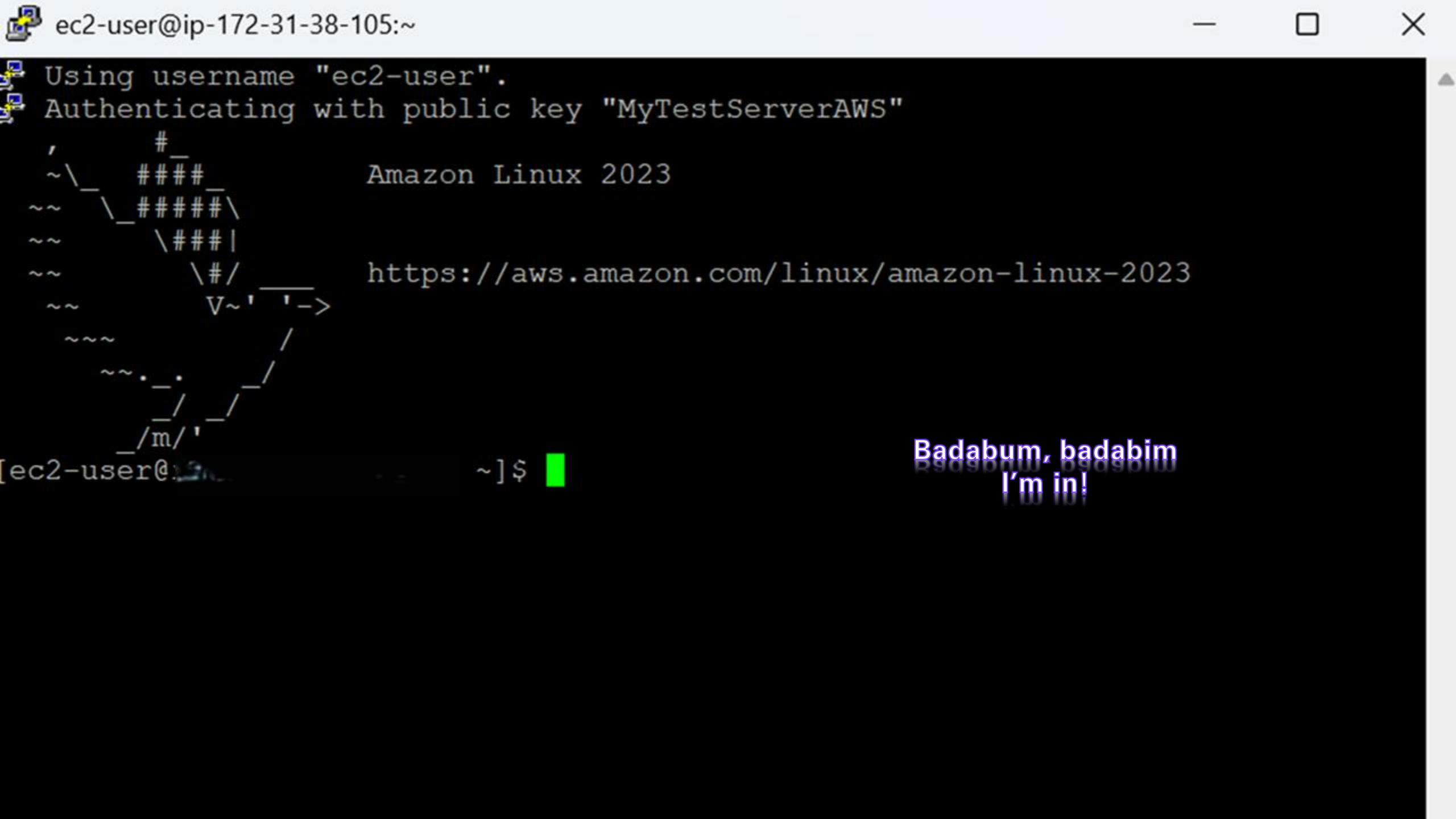
I needed to create key with PuTTy Key Generator.
Checked RSA and clicked Load.
I chose .ppk file I created earlier.



Then I could leave PuTTYGen and go to PuTTY to SSH in the server. I got my host name from the EC2 instance.



Paste host name, add the key and save the session and click open…


ec2-user@ip-172-31-38-105:~
Using username "ec2-user".
Authenticating with public key "MyTestServerAWS"
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@
~]$
Badabum, badabim
I'm in!

```
[ec2-user@i          ~]$ sudo su
[root@ip-17          user]# yum update
Amazon Linux 2023 Kernel Livepatch repository                145 kB/s |  15 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@ip          ec2-user]# yum update -y
Last metadata expiration check: 0:02:20 ago on Tue Apr 15 10:54:10 2025.
Dependencies resolved.
Nothing to do.
Complete!
[roo          -user]#
```

➔ sudo su
➔ yum update

```
Installed:
  apr-1.7.5-1.amzn2023.0.4.x86_64
  apr-util-1.6.3-1.amzn2023.0.1.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.62-1.amzn2023.x86_64
  httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch
  httpd-tools-2.4.62-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-2.0.27-1.amzn2023.0.3.x86_64
  mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[root@i          ec2-user]#
```

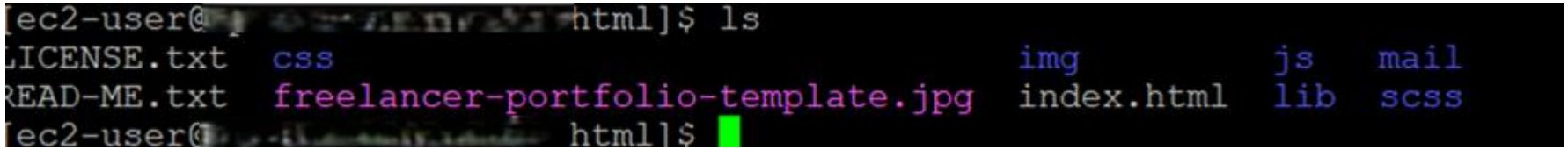
➔ yum install –y httpd

```
[root@          ;2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@          ec2-user]#
```

➔ yum systemctl status httpd

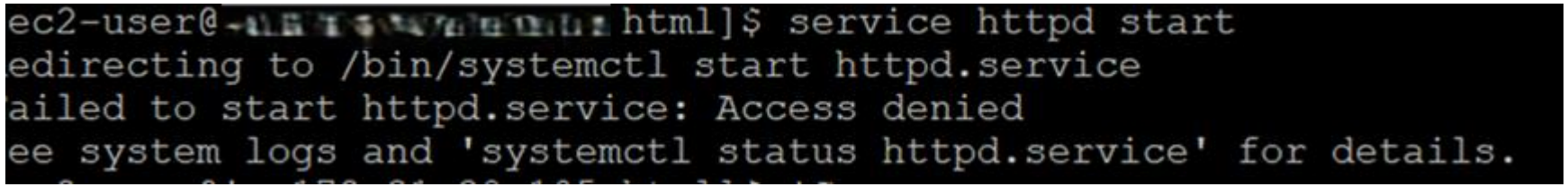
Then I needed to copy the code for my websites to server's directory with WinSCP. There seemed to be a problem with permissions, so I needed to run a command to give the permission to change files. (This is not safe setting for real usage!)

Let’s see if everyone’s made it to the new home…



All good! It is time to start the server!

➔ service httpd start



Whoopsie, *someone* kicked the root user out, let’s try again.

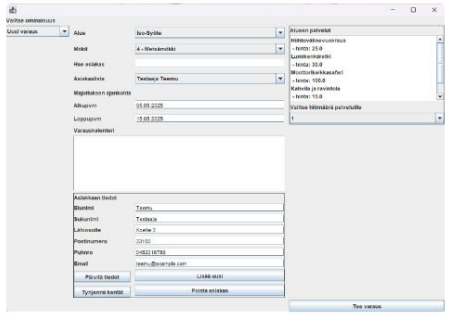
Main language used at UEF.

Completed a course in Information Management and SQL, focusing on relational databases, data modeling, and SQL queries.

Currently studying a web course on R.

PROJECTS

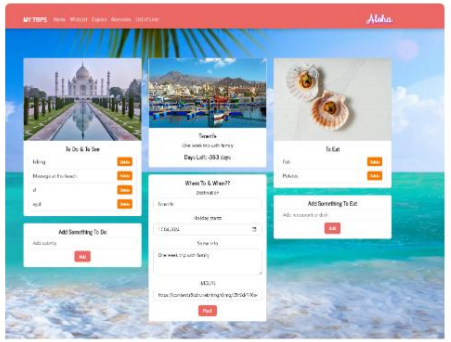
MY PORTFOLIO



Cottage Rental Application

Java + MySQL

ArcadeGame on Android
Java + AndroidStudio



TravelPlanner
JavaScript + React + HTML

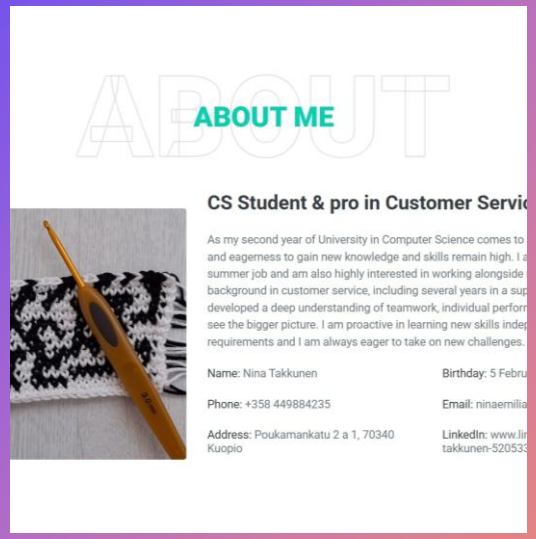
dress

| open address

My project:
studying Linux basics

My project:
experimenting with Amazon AWS

The website is up and running!



What i learned?

- How to launch a cloud-based virtual server
- Basics of Linux management
- How to install and manage HTTP server
- How to deploy and serve static websites from EC2
- Troubleshooting, as in every project ever!
- There are still major safety issues to be solved

I HOPE YOU ENJOYED MY LITTLE EXPERIMENT!

Nina Takkunen

LinkedIn : www.linkedin.com/in/nina-takkunen-5205332ab