PROJECT: AMAZON AWS EC2 INSTANCE WEBSITE HOSTING

Author: Nina Takkunen 2025

CONTENTS

My goal

Tools

Options

Sign in & Set up

What I learned

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

PuTTY

Visual Studio Code

aws

OPTIONS

Microsoft Azure
1 month trial 200 credits

GCP
3 month trial 300 credits

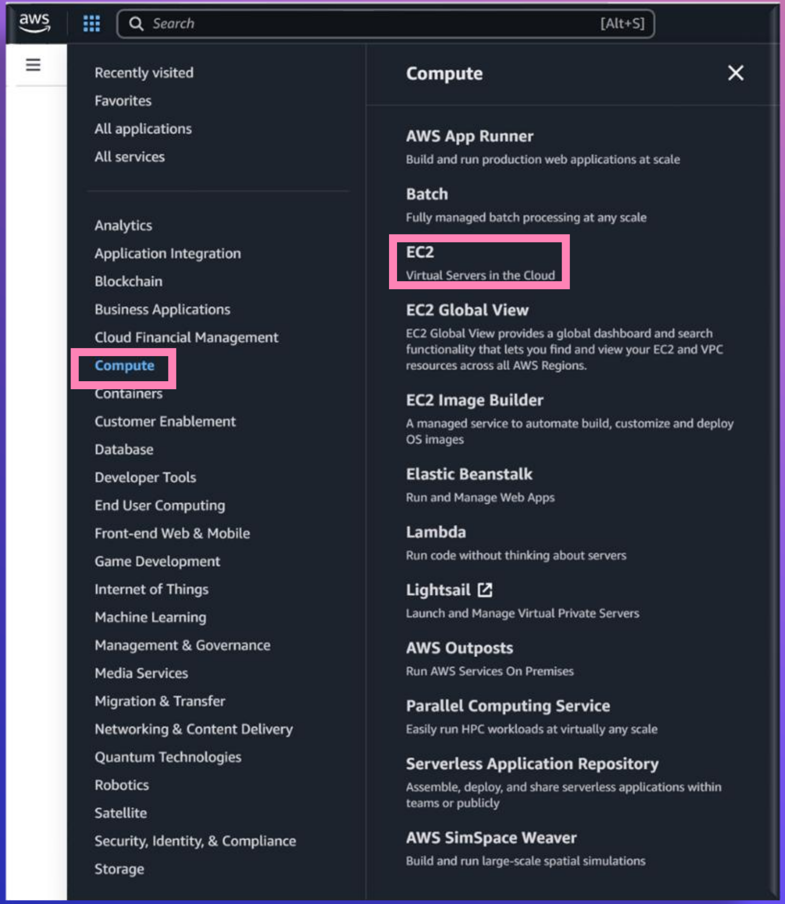
Amazon AWS
12 month free tier

SIGNING UP & SETTING UP

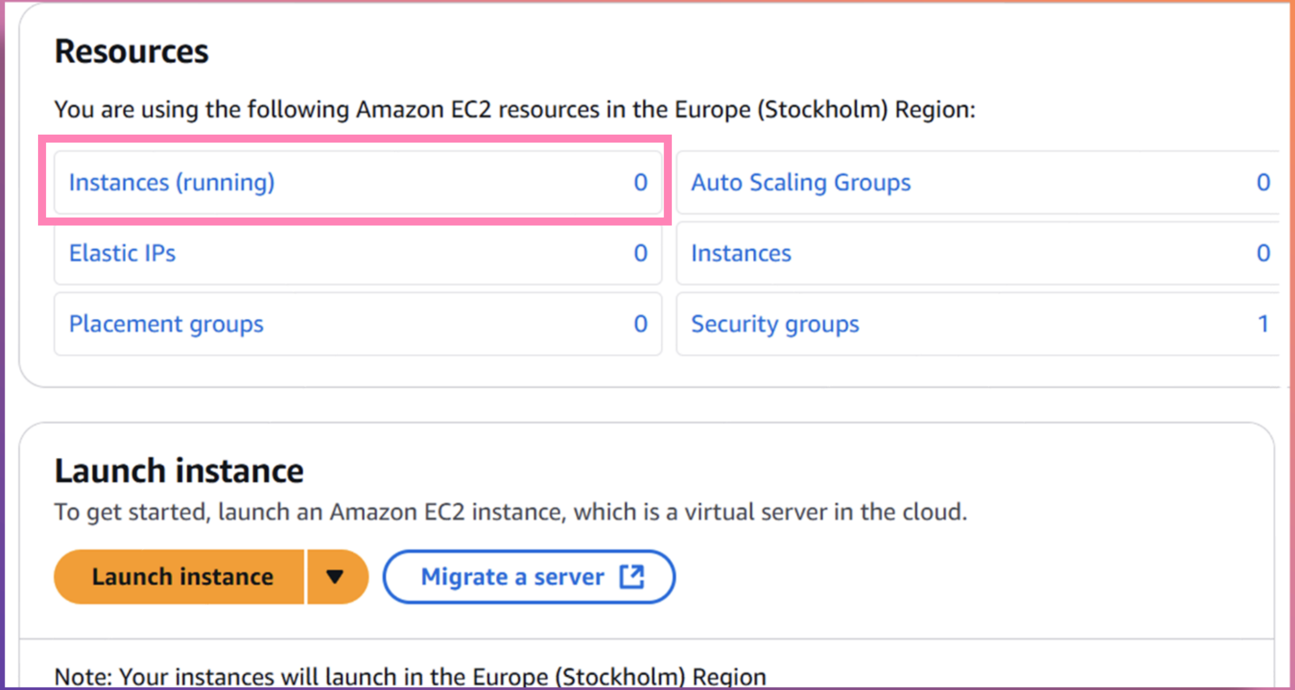
- I signed in and created an AWS account.
- I browsed documentaries, demos and tutorials and watched several Youtube videos!

NOW LET’S CREATE AN EC2 INSTANCE!

Select Console -> Compute -> EC2


aws
Search
[Alt+S]
Recently visited
Favorites
All applications
All services
Analytics
Application Integration
Blockchain
Business Applications
Cloud Financial Management
Compute
Containers
Customer Enablement
Database
Developer Tools
End User Computing
Front-end Web & Mobile
Game Development
Internet of Things
Machine Learning
Management & Governance
Media Services
Migration & Transfer
Networking & Content Delivery
Quantum Technologies
Robotics
Satellite
Security, Identity, & Compliance
Storage
Compute
AWS App Runner
Build and run production web applications at scale
Batch
Fully managed batch processing at any scale
EC2
Virtual Servers in the Cloud
EC2 Global View
EC2 Global View provides a global dashboard and search functionality that lets you find and view your EC2 and VPC resources across all AWS Regions.
EC2 Image Builder
A managed service to automate build, customize and deploy OS images
Elastic Beanstalk
Run and Manage Web Apps
Lambda
Run code without thinking about servers
Lightsail
Launch and Manage Virtual Private Servers
AWS Outposts
Run AWS Services On Premises
Parallel Computing Service
Easily run HPC workloads at virtually any scale
Serverless Application Repository
Assemble, deploy, and share serverless applications within teams or publicly
AWS SimSpace Weaver
Build and run large-scale spatial simulations

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



Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

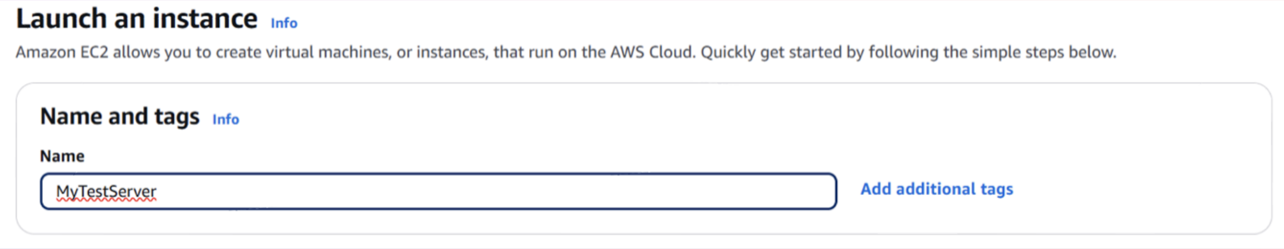
Instances (running)	0	Auto Scaling Groups	0
Elastic IPs	0	Instances	0
Placement groups	0	Security groups	1

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Europe (Stockholm) Region

First I named my server



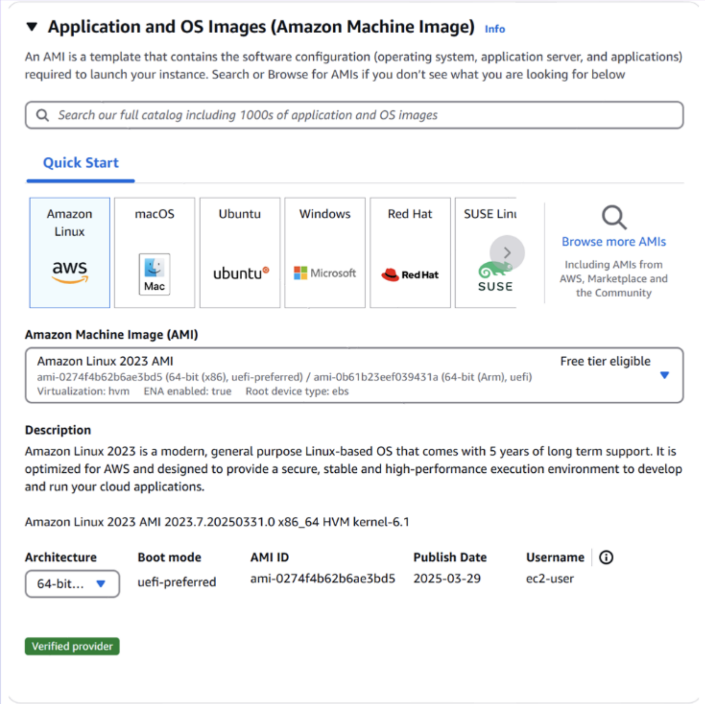
Launch an instance Info

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: MyTestServer

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



▼ 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

Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
SUSE Linu
SUSE
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0274f4b62b6ae3bd5 (64-bit (x86), uefi-preferred) / ami-0b61b23eef039431a (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

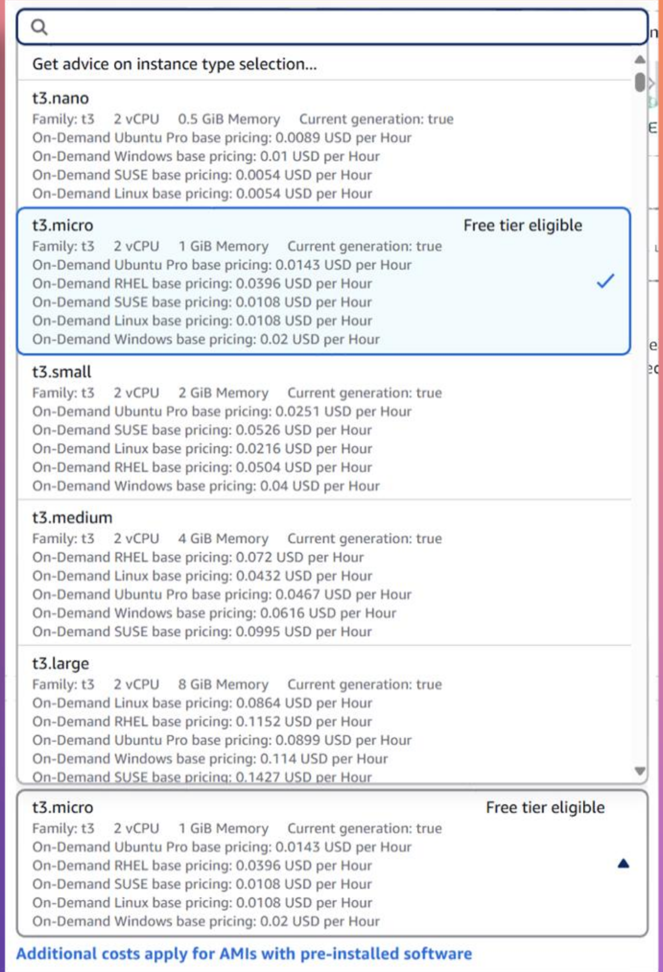
Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250331.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username
64-bit... ▼	uefi-preferred	ami-0274f4b62b6ae3bd5	2025-03-29	ec2-user

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



t3.nano
Family: t3 2 vCPU 0.5 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0089 USD per Hour
On-Demand Windows base pricing: 0.01 USD per Hour
On-Demand SUSE base pricing: 0.0054 USD per Hour
On-Demand Linux base pricing: 0.0054 USD per Hour

t3.micro — Free tier eligible ✓
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0143 USD per Hour
On-Demand RHEL base pricing: 0.0396 USD per Hour
On-Demand SUSE base pricing: 0.0108 USD per Hour
On-Demand Linux base pricing: 0.0108 USD per Hour
On-Demand Windows base pricing: 0.02 USD per Hour

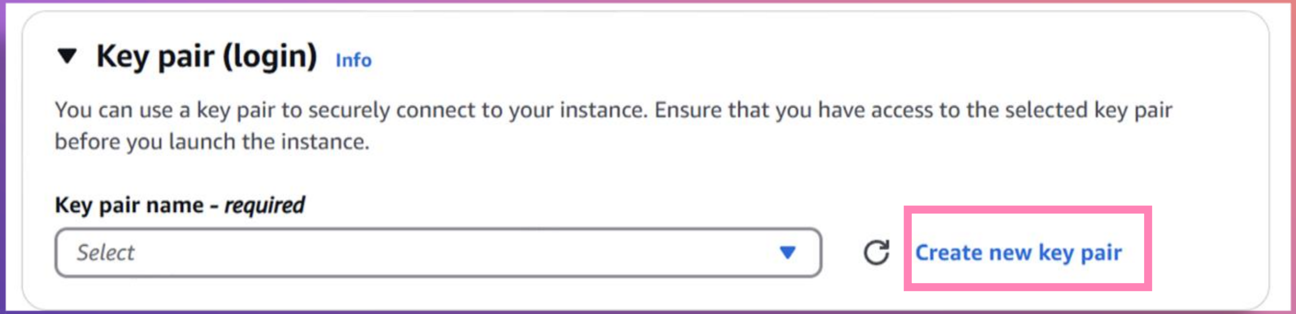
t3.small
Family: t3 2 vCPU 2 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0251 USD per Hour
On-Demand SUSE base pricing: 0.0526 USD per Hour
On-Demand Linux base pricing: 0.0216 USD per Hour
On-Demand RHEL base pricing: 0.0504 USD per Hour
On-Demand Windows base pricing: 0.04 USD per Hour

t3.medium
Family: t3 2 vCPU 4 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.072 USD per Hour
On-Demand Linux base pricing: 0.0432 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0467 USD per Hour
On-Demand Windows base pricing: 0.0616 USD per Hour
On-Demand SUSE base pricing: 0.0995 USD per Hour

t3.large
Family: t3 2 vCPU 8 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0864 USD per Hour
On-Demand RHEL base pricing: 0.1152 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0899 USD per Hour
On-Demand Windows base pricing: 0.114 USD per Hour
On-Demand SUSE base pricing: 0.1427 USD per Hour

t3.micro — Free tier eligible ▲
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0143 USD per Hour
On-Demand RHEL base pricing: 0.0396 USD per Hour
On-Demand SUSE base pricing: 0.0108 USD per Hour
On-Demand Linux base pricing: 0.0108 USD per Hour
On-Demand Windows base pricing: 0.02 USD per Hour

Now I needed to create a key pair for login

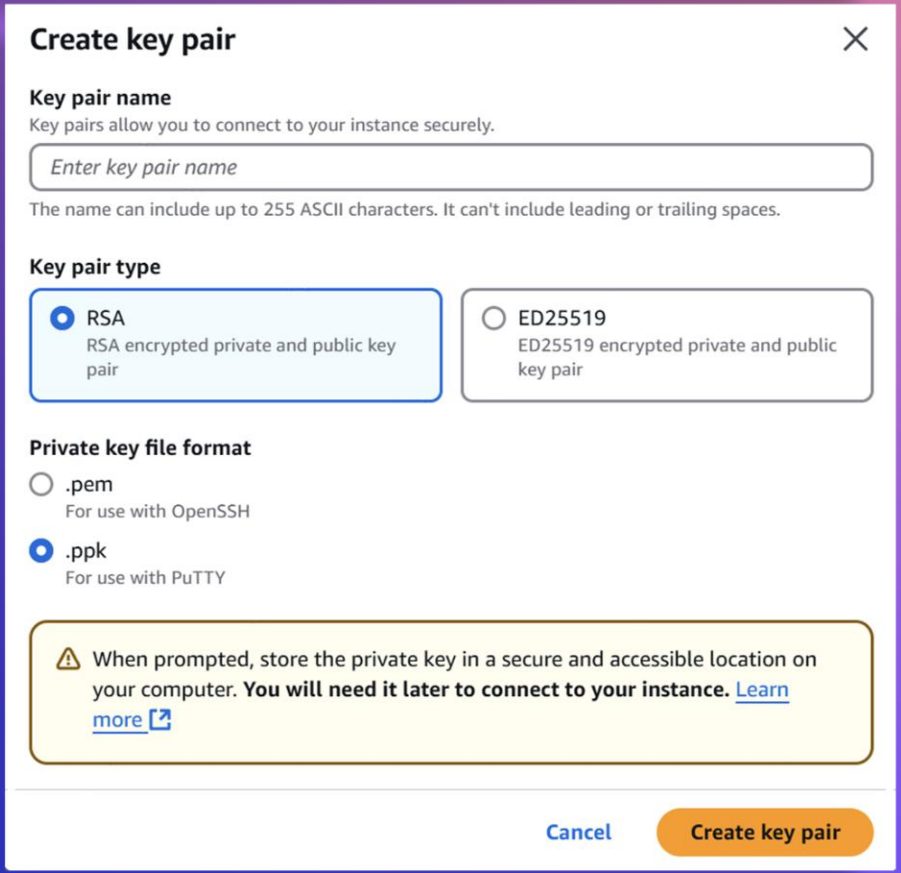


▼ **Key pair (login)** Info

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 key pair

Key pair name

Key pairs allow you to connect to your instance securely.

Enter key pair name

The name can include up to 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

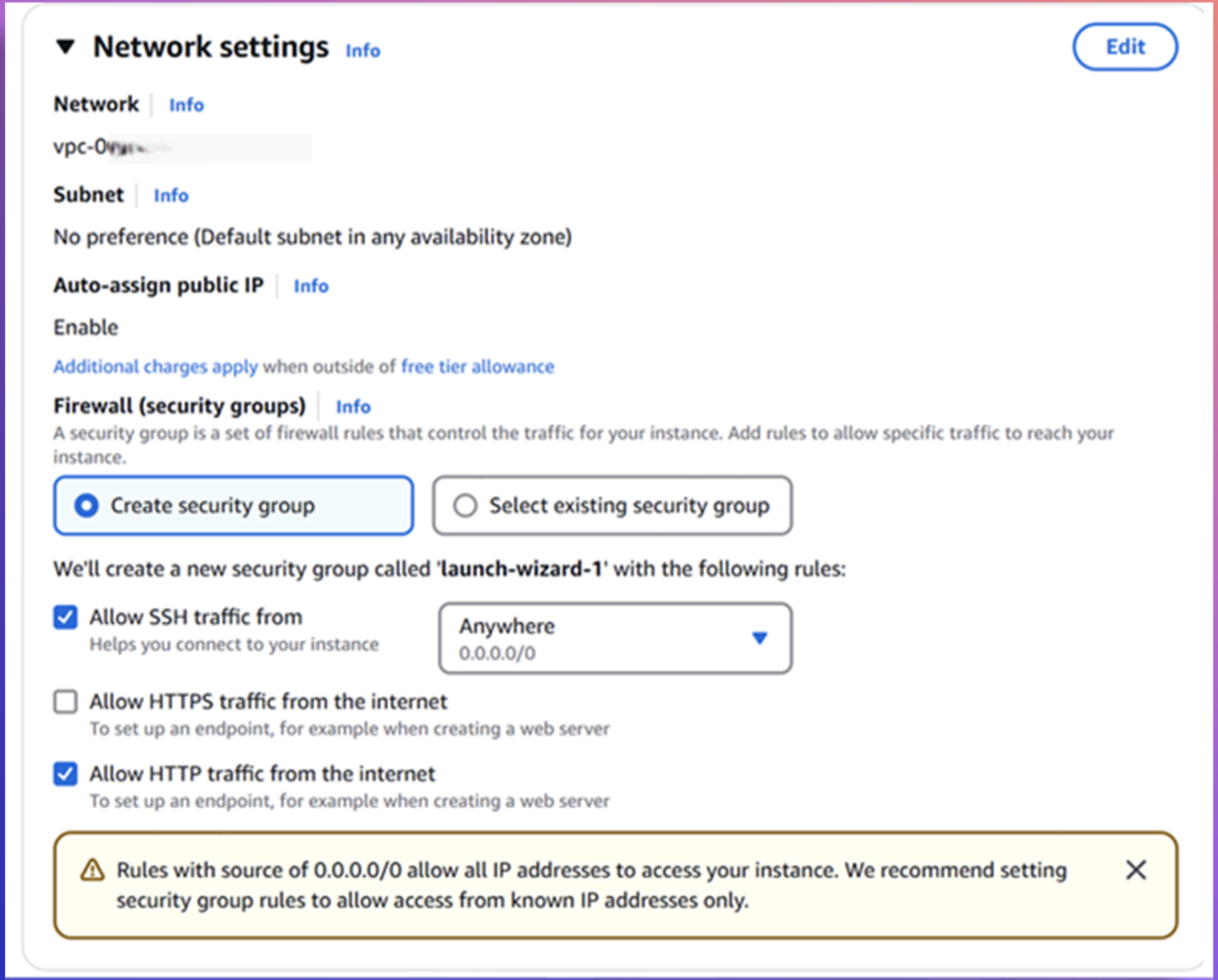
○ .pem
For use with OpenSSH

● .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

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

Let’s define security rules



▼ **Network settings** Info

Edit

Network | Info

vpc-0[redacted]

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

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 reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0 ▼
Helps you connect to your instance

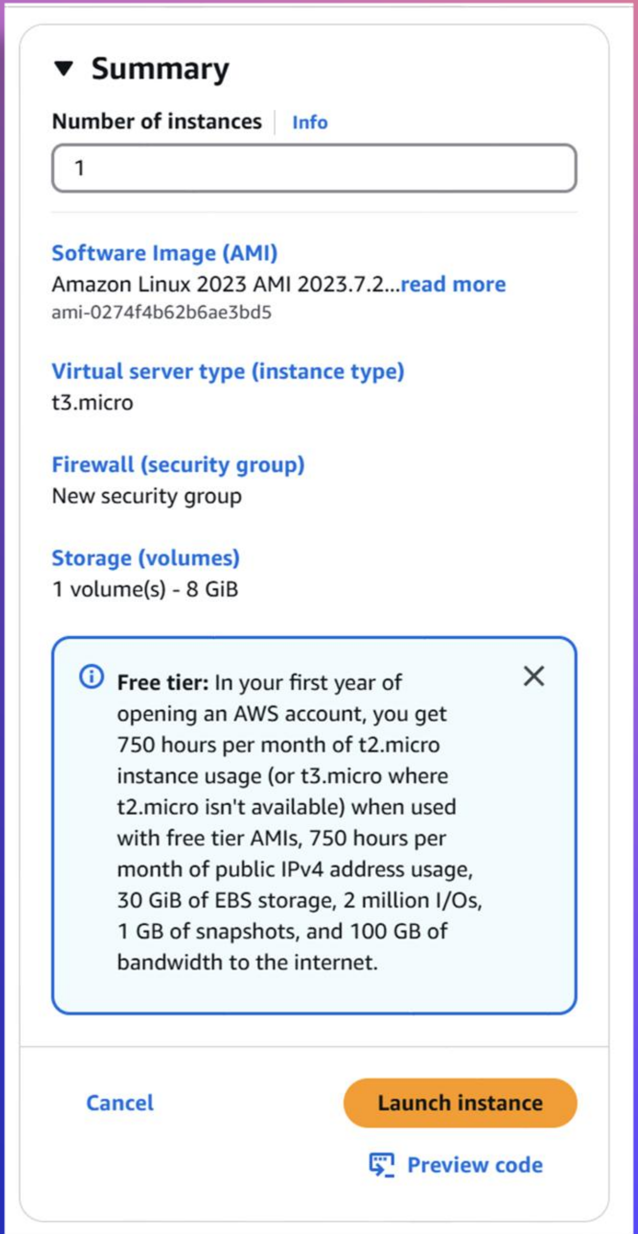
□ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

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

I set the SSH to be allowed only from my IP-address



▼ Summary

Number of instances | Info

1

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

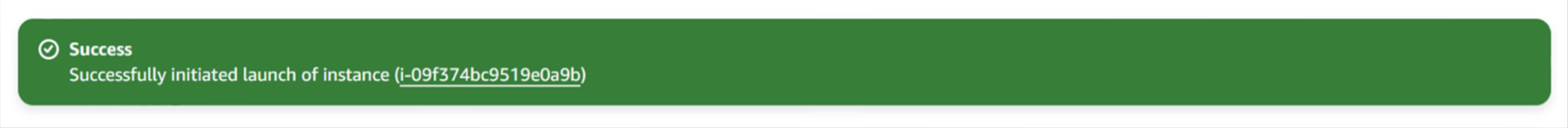
Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

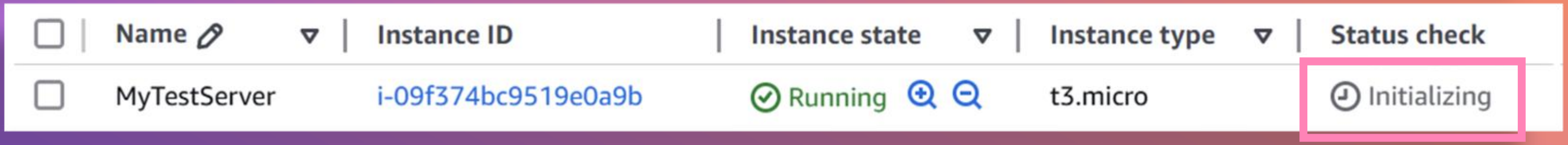
Storage (volumes)
1 volume(s) - 8 GiB

ⓘ **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. ✕

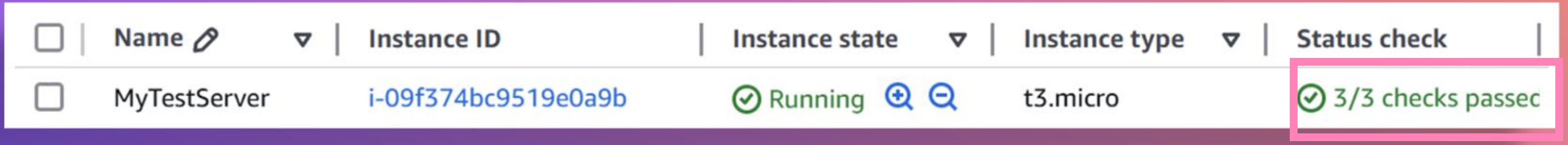
Ready to board and launch!



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

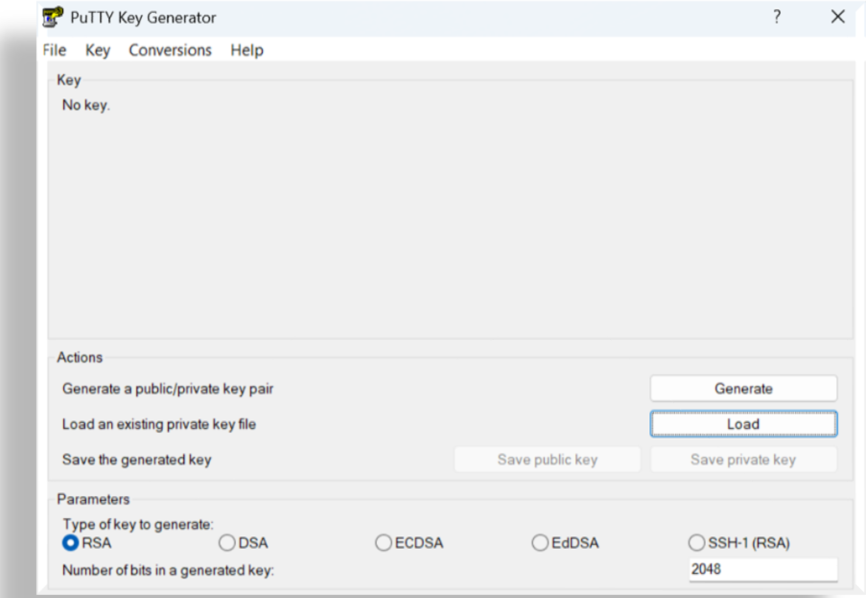


□	Name	Instance ID	Instance state	Instance type	Status check
□	MyTestServer	i-09f374bc9519e0a9b	✓ Running	t3.micro	Initializing



□	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



Key

No key.

Actions

Generate a public/private key pair — Generate

Load an existing private key file — Load

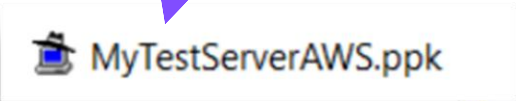
Save the generated key — Save public key — Save private key

Parameters

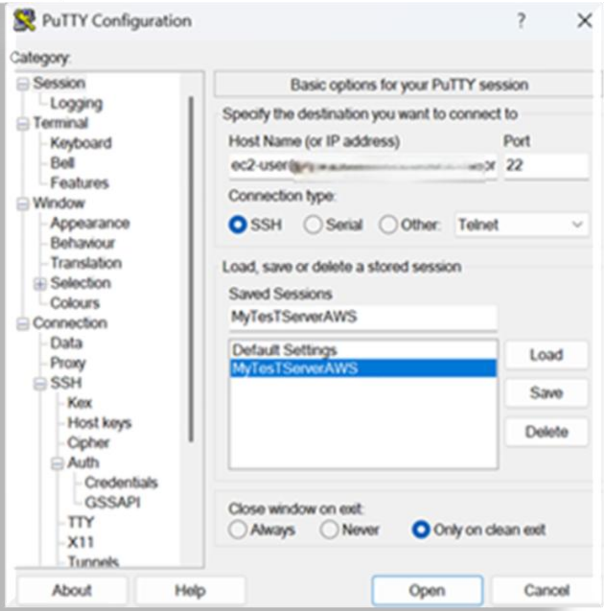
Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ EdDSA ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

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



MyTestServerAWS.ppk



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ec2-user@[redacted] Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

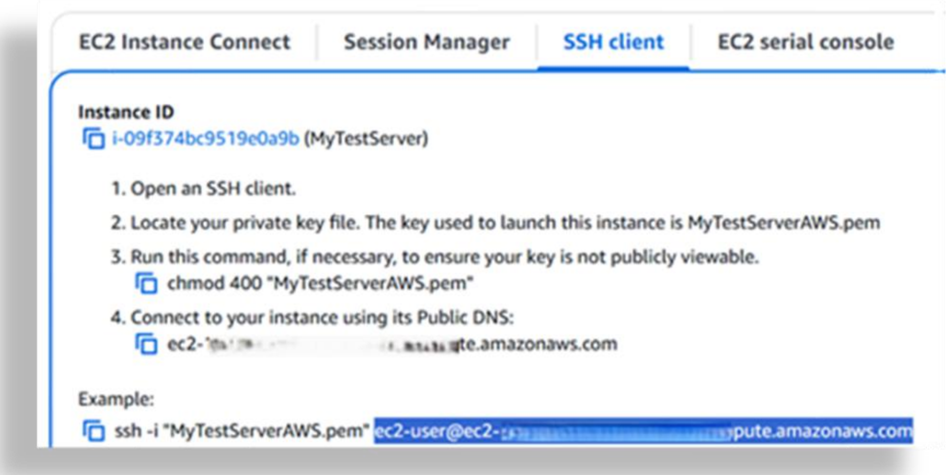
Saved Sessions: MyTesTServerAWS

Default Settings
MyTesTServerAWS

Load / Save / Delete

Close window on exit: ○ Always ○ Never ● Only on clean exit

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



Instance ID

i-09f374bc9519e0a9b (MyTestServer)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is MyTestServerAWS.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

```
chmod 400 "MyTestServerAWS.pem"
```

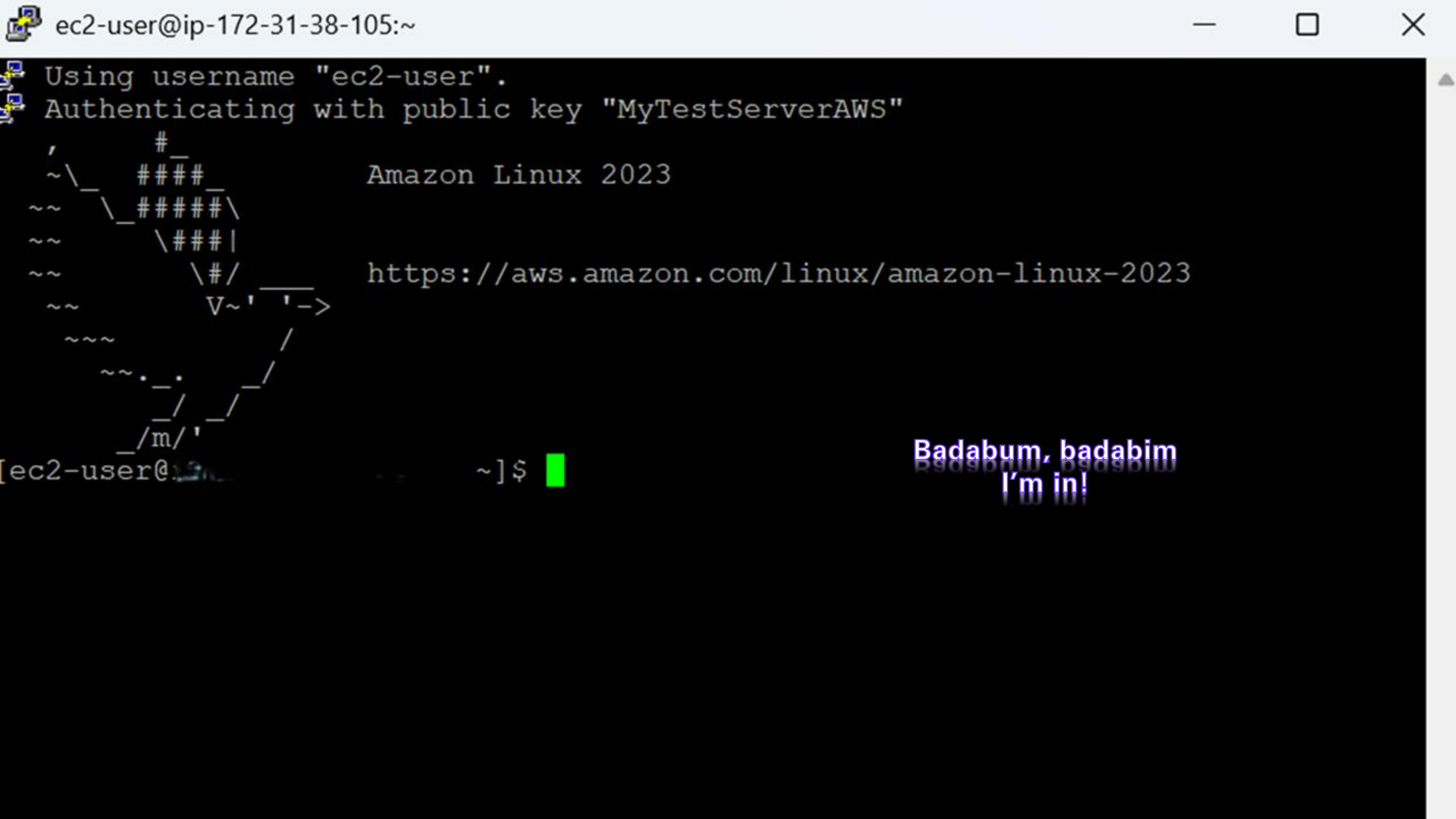
4. Connect to your instance using its Public DNS:

```
ec2-[redacted]te.amazonaws.com
```

Example:

```
ssh -i "MyTestServerAWS.pem" ec2-user@ec2-[redacted]pute.amazonaws.com
```

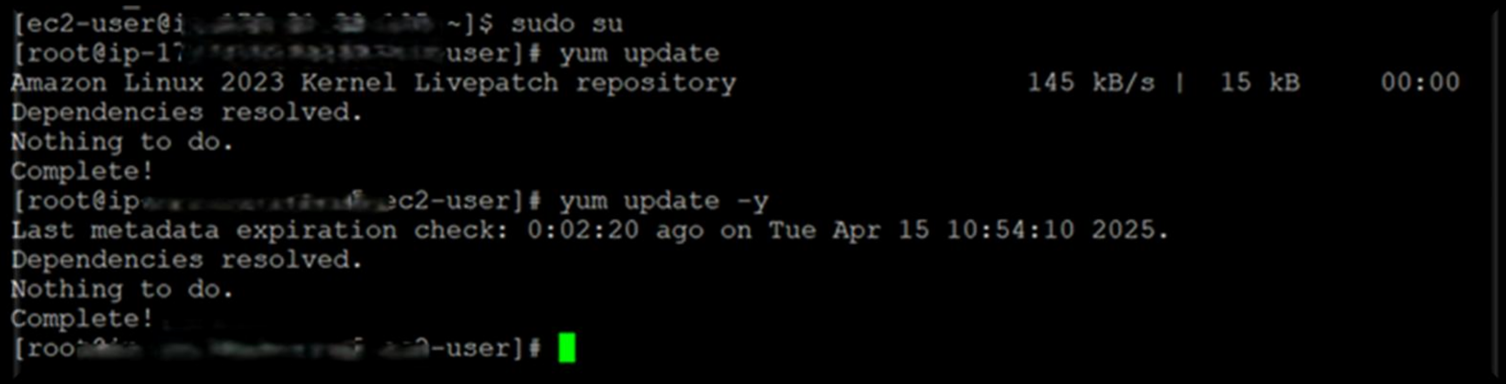
Paste host name, add the key and save the session and click open...



ec2-user@ip-172-31-38-105:~

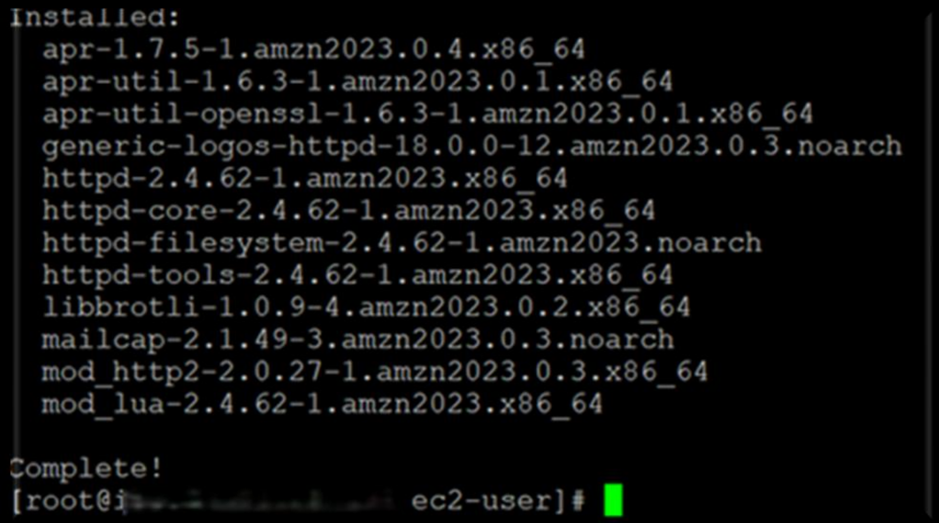
```
Using username "ec2-user".
Authenticating with public key "MyTestServerAWS"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@[redacted] ~]$
```

Badadum, badadim
I’m in!



```
[ec2-user@i[redacted] ~]$ sudo su
[root@ip-17[redacted]user]# yum update
Amazon Linux 2023 Kernel Livepatch repository                145 kB/s |  15 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@ip[redacted]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[redacted]-user]# █
```

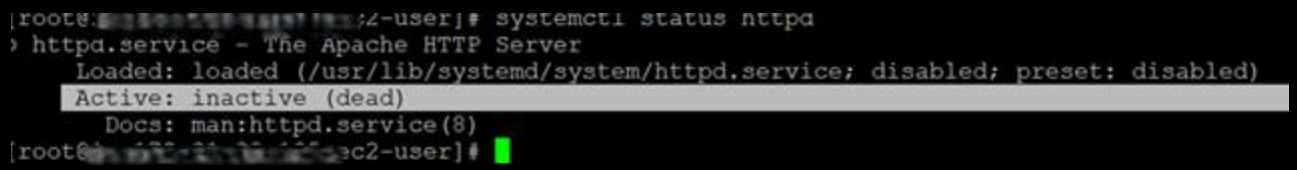
➔ sudo su
➔ yum update



```
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[redacted] ec2-user]# █
```

➔ yum install -y httpd

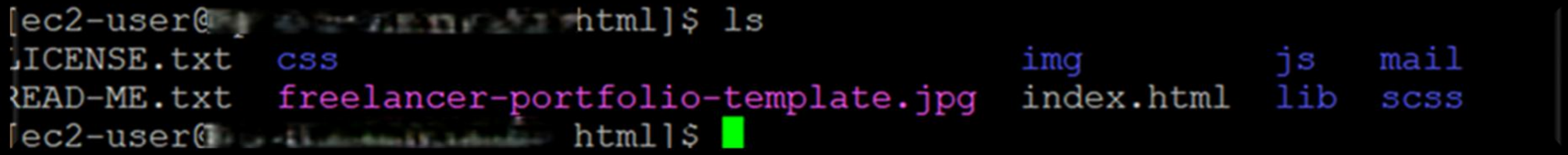


```
[root@[redacted]2-user]# systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@[redacted]ec2-user]# █
```

➔ yum systemctl status httpd

Then I needed to copy the code for my websites to server’s directory with WinSCP, which was easy to use with the both directories side to side

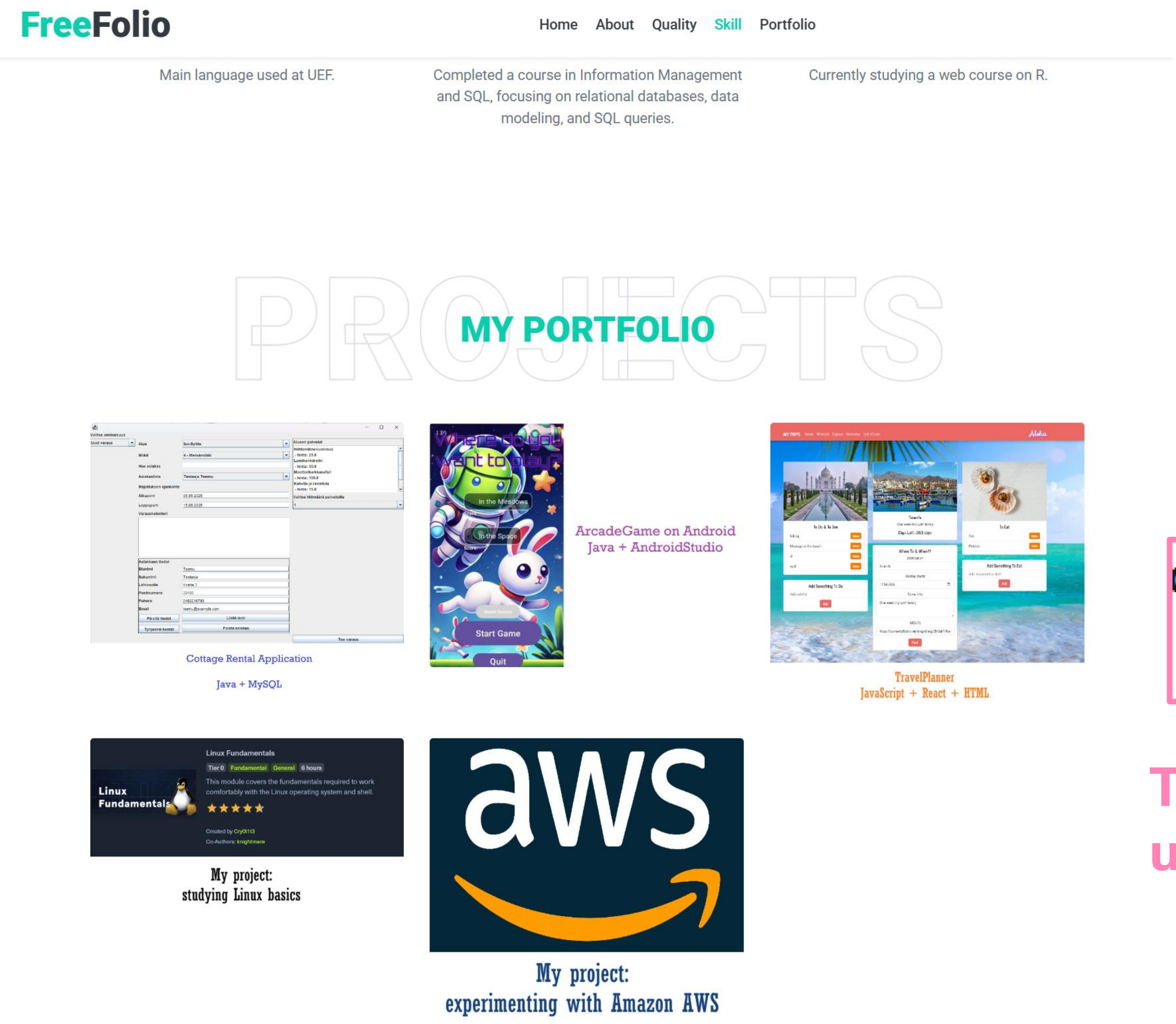
Let’s see if everyone’s made it to the new home…



```
[ec2-user@[redacted] html]$ ls
LICENSE.txt  css                                 img          js   mail
README.txt   freelancer-portfolio-template.jpg  index.html   lib  scss
[ec2-user@[redacted] html]$ 
```

All good! It is time to start the server!

➔ `service httpd start`



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.

MY PORTFOLIO

Cottage Rental Application

Java + MySQL

ArcadeGame on Android
Java + AndroidStudio

TravelPlanner
JavaScript + React + HTML

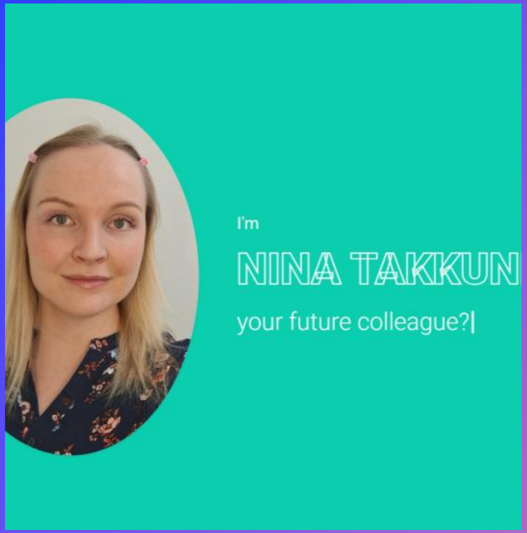
My project:
studying Linux basics

My project:
experimenting with Amazon AWS

The website is up and running!

What i learned?

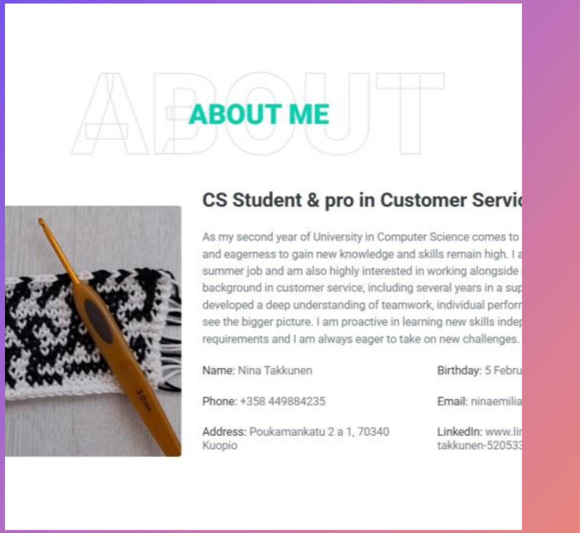
- How to launch a cloud-based virtual server
- Basics of Linux management
- How to install and manage HTTP server
- How to deploy and serve static websites from EC2
- Troubleshooting, as in every project ever!
- There are still major safety issues to be solved



I'm

NINA TAKKUN

your future colleague?|



ABOUT ME

CS Student & pro in Customer Servic

As my second year of University in Computer Science comes to
and eagerness to gain new knowledge and skills remain high. I a
summer job and am also highly interested in working alongside
background in customer service, including several years in a sup
developed a deep understanding of teamwork, individual perforr
see the bigger picture. I am proactive in learning new skills indep
requirements and I am always eager to take on new challenges.

Name: Nina Takkunen

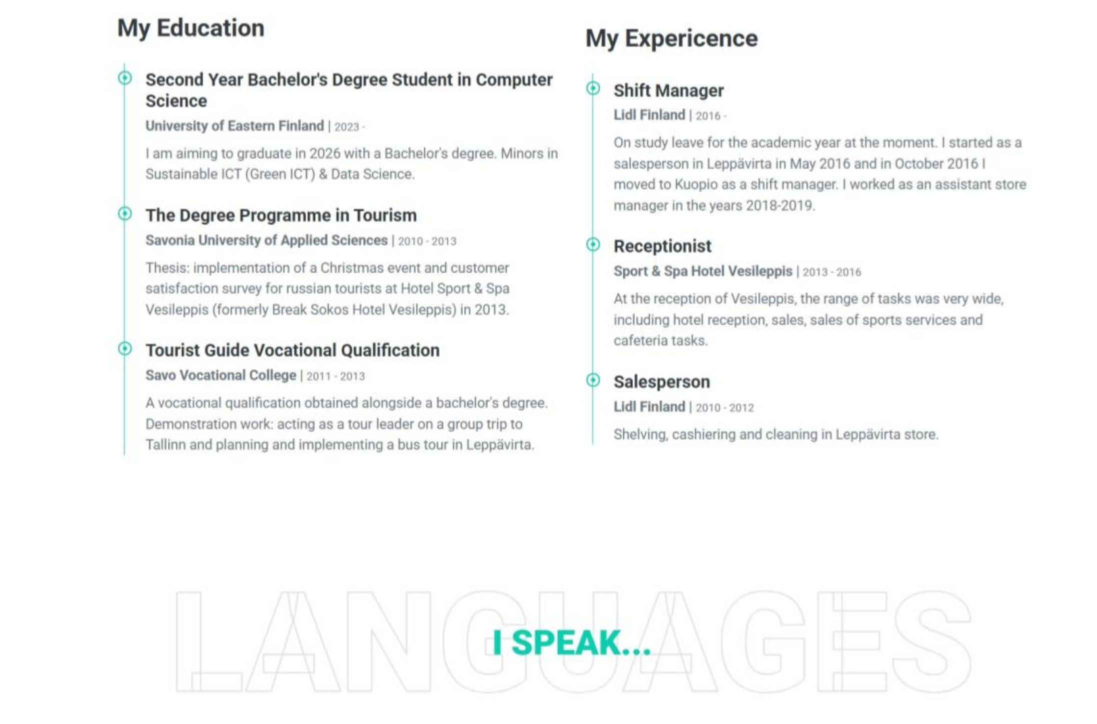
Birthday: 5 Febru

Phone: +358 449884235

Email: ninaemilia

Address: Poukamankatu 2 a 1, 70340
Kuopio

LinkedIn: www.lin
takkunen-520533



My Education

Second Year Bachelor's Degree Student in Computer Science
University of Eastern Finland | 2023 -

I am aiming to graduate in 2026 with a Bachelor's degree. Minors in Sustainable ICT (Green ICT) & Data Science.

The Degree Programme in Tourism
Savonia University of Applied Sciences | 2010 - 2013

Thesis: implementation of a Christmas event and customer satisfaction survey for russian tourists at Hotel Sport & Spa Vesileppis (formerly Break Sokos Hotel Vesileppis) in 2013.

Tourist Guide Vocational Qualification
Savo Vocational College | 2011 - 2013

A vocational qualification obtained alongside a bachelor's degree. Demonstration work: acting as a tour leader on a group trip to Tallinn and planning and implementing a bus tour in Leppävirta.

My Expericence

Shift Manager
Lidl Finland | 2016 -

On study leave for the academic year at the moment. I started as a salesperson in Leppävirta in May 2016 and in October 2016 I moved to Kuopio as a shift manager. I worked as an assistant store manager in the years 2018-2019.

Receptionist
Sport & Spa Hotel Vesileppis | 2013 - 2016

At the reception of Vesileppis, the range of tasks was very wide, including hotel reception, sales, sales of sports services and cafeteria tasks.

Salesperson
Lidl Finland | 2010 - 2012

Shelving, cashiering and cleaning in Leppävirta store.

LANGUAGES

I SPEAK...

I HOPE YOU ENJOYED MY LITTLE EXPERIMENT!

Nina Takkunen

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