ICT1711 ASSIGNMENT 2

Cloud Server Project

Student Name: Lovevish Ramchurun
Student Number: 35501701
Project Name: DubaiCareerHub
Hosting Provider: Amazon EC2
Server IP / Domain Name: https://dubaicareerhub.online
GitHub Repository: https://github.com/lovern13/dxbcareerhub

Contents

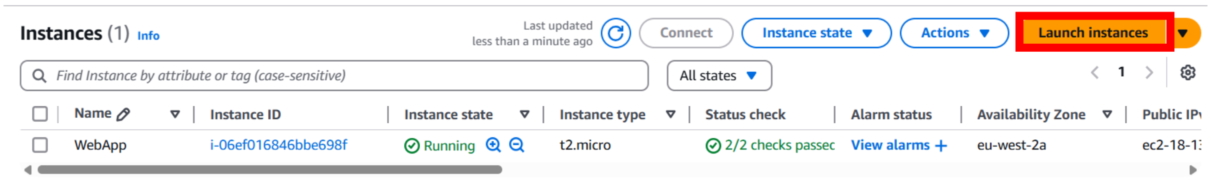
Project Overview

DubaiCareerHub is a static job listing website developed as part of the ICT171 Cloud Project. The goal was to create a visually engaging platform that showcases job opportunities in Dubai, deploy it on a Linux server manually using IaaS (Amazon EC2), and ensure accessibility via the internet.

Website Features

- Custom built homepage with job listings
- Responsive design (mobile-friendly)
- Modal pop-ups for login, register, and apply
- Interactive search bar
- License and footer with social links
- Clean UI using only HTML/CSS and JavaScript
- No database – jobs are static in the HTML code

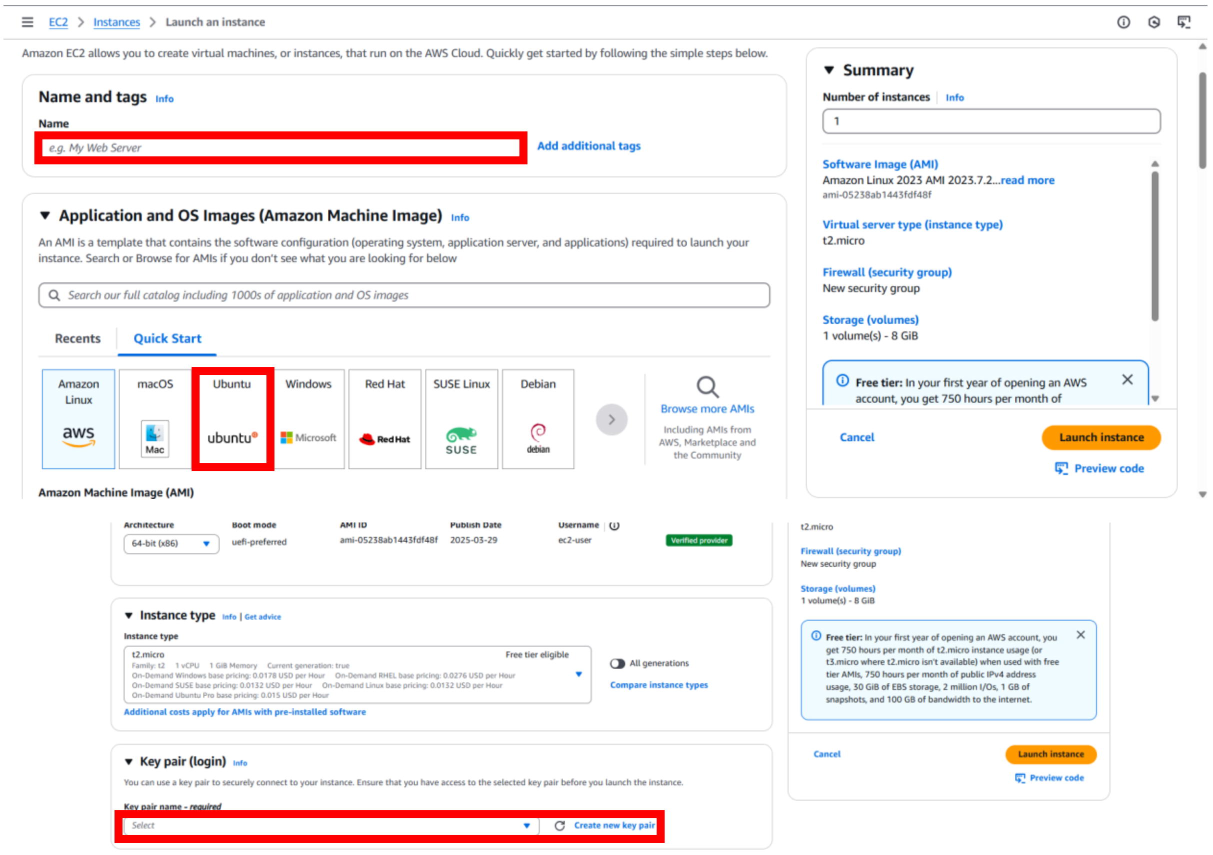
Setting up cloud server



Instances (1) Info

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
WebApp	i-06ef016846bbe698f	Running	t2.micro	2/2 checks passec	View alarms +	eu-west-2a	ec2-18-1



EC2 > Instances > Launch an instance

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: e.g. My Web Server

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
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Architecture	Boot mode	AMI ID	Publish Date	Username	
64-bit (x86)	uefi-preferred	ami-05238ab1443fdf48f	2025-03-29	ec2-user	Verified provider

Instance type Info | Get advice

Instance type: t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0178 USD per Hour On-Demand RHEL base pricing: 0.0276 USD per Hour
On-Demand SUSE base pricing: 0.0132 USD per Hour On-Demand Linux base pricing: 0.0132 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.015 USD per Hour

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

Summary

Number of instances Info: 1

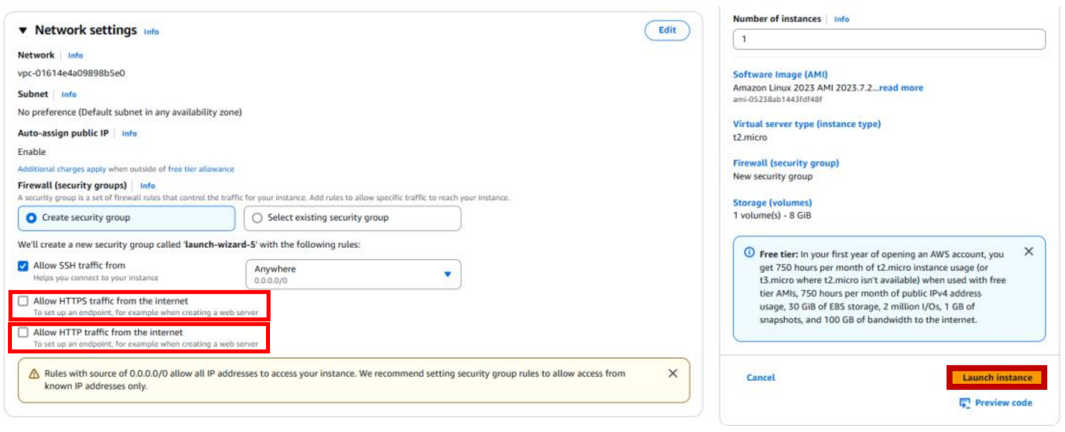
Software Image (AMI): Amazon Linux 2023 AMI 2023.7.2...read more
ami-05238ab1443fdf48f

Virtual server type (instance type): t2.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.



Network settings Info

Edit

Network Info

vpc-01614e4a09898b5e0

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-5**' 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.

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-05238ab1443fdf48f

Virtual server type (instance type)
t2.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.

Connect via SSH

```
ssh -i "key1.pem" ubuntu@ec2-35-177-110-212.eu-west-2.compute.amazonaws.com
```

Web Server Confinguration

Installing Apache Web Server

```
sudo apt update
```

```
sudo apt install apache2 -y
```

```
sudo systemctl enable apache2
```

```
sudo systemctl start apache2
```

Upload to github

To gain root access (no need to use sudo)

```
sudo chown -R $USER:$USER /var/www/html
```

Navigate through the directory

```
cd /var/www/html
```

Initialising the Git repository

```
git init
```

Adding a remote GitHub repository

```
git remote add origin https://github.com/lovern13/dxbcareerhub.git
```

Add all files

```
git add .
```

Commit the files

```
git commit -m "Initial commit"
```

Push the files

```
git branch -M main
```

```
git push -u origin main
```

DNS Configuration (GoDaddy)

- Logged into GoDaddy → My Products → DNS Management.
- Added an A Record:
  - Type: A
  - Host: @ (for root domain)
  - Points to: EC2 Public IP adress
  - TTL: 1 Hour
  - Added www CNAME

Enabling HTTPS (SSL/TPS)

Installing Certbot

```
sudo apt install certbot python3-certbot-apache -y
```

Command line to obtain SSL certificate

```
sudo certbot --apache -d dubaicareerhub.com -d www.dubaicareerhub.com
```

Command line to verify SSL certificate

```
sudo certbot certificates
```

Auto renewal:

```
sudo certbot renew --dry-run
```

DNS Check

```
nslookup dubaicareerhub.online
```

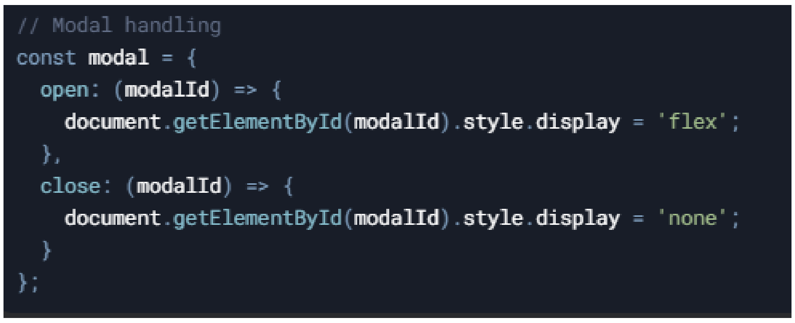
Website documentation

- Index.html
- Styles.css
- Script.js
- README.md
- Setup.sh

Key scripts

Script.js handles:

- User login/registration modals
- Job application form with file uploads
- Search functionality



```
// Modal handling
const modal = {
  open: (modalId) => {
    document.getElementById(modalId).style.display = 'flex';
  },
  close: (modalId) => {
    document.getElementById(modalId).style.display = 'none';
  }
};
```

Conclusion

In this project, I successfully deployed a fully functional static website DubaiCareerHub on an Amazon EC2 instance using the Infrastructure as a Service (IaaS) model. I manually configured the Ubuntu server, installed and managed Apache, linked the deployment with a custom domain via GoDaddy, and secured the site using Let’s Encrypt SSL certificates.

This assignment helped me develop valuable skills in cloud server setup, Linux command-line operations, web hosting, DNS configuration, and SSL implementation. I also became more comfortable with documentation and version control using GitHub.

In the future, this project could be enhanced by integrating a backend (e.g., PHP and MySQL) to allow dynamic job posting and user accounts. For now, it demonstrates a solid understanding of cloud infrastructure and static web hosting.