VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

NAME: POLI VARDHINI REDDY

REGISTER NUMBER: 21BIT0382

COURSE: CLOUD COMPUTING

FACULTY: SUDHA.M MAM

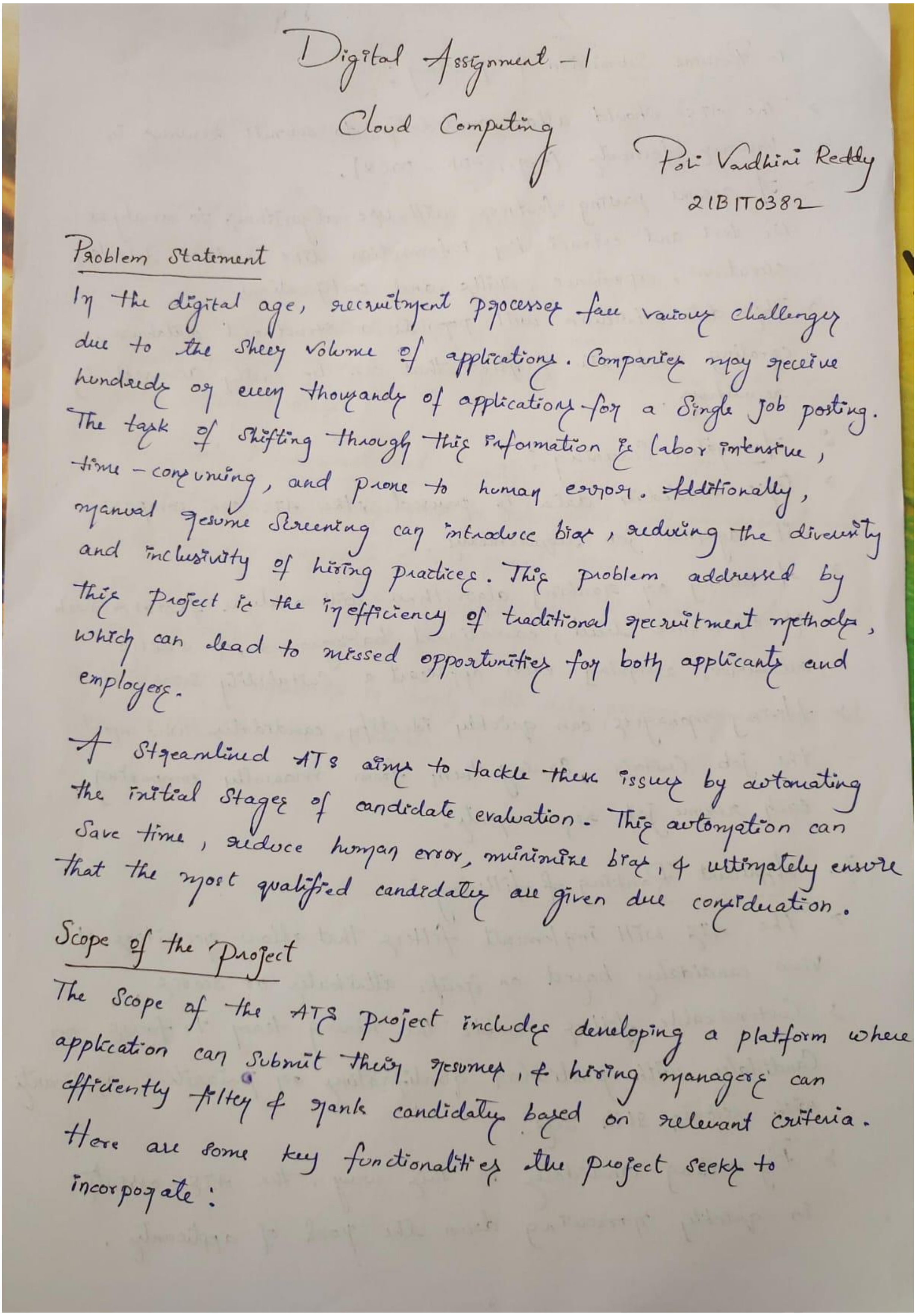
DIGITAL ASSIGNMENT -2

Question :

- Host a website on a cloud infrastructure platform such as AWS, GCP, Azzure.
- Demonstrate step by step implementation of the same.

INDEX:

Problem Statement:



Digital Assignment - 1

Cloud Computing

Poli Vardhini Reddy
21BIT0382

Problem Statement

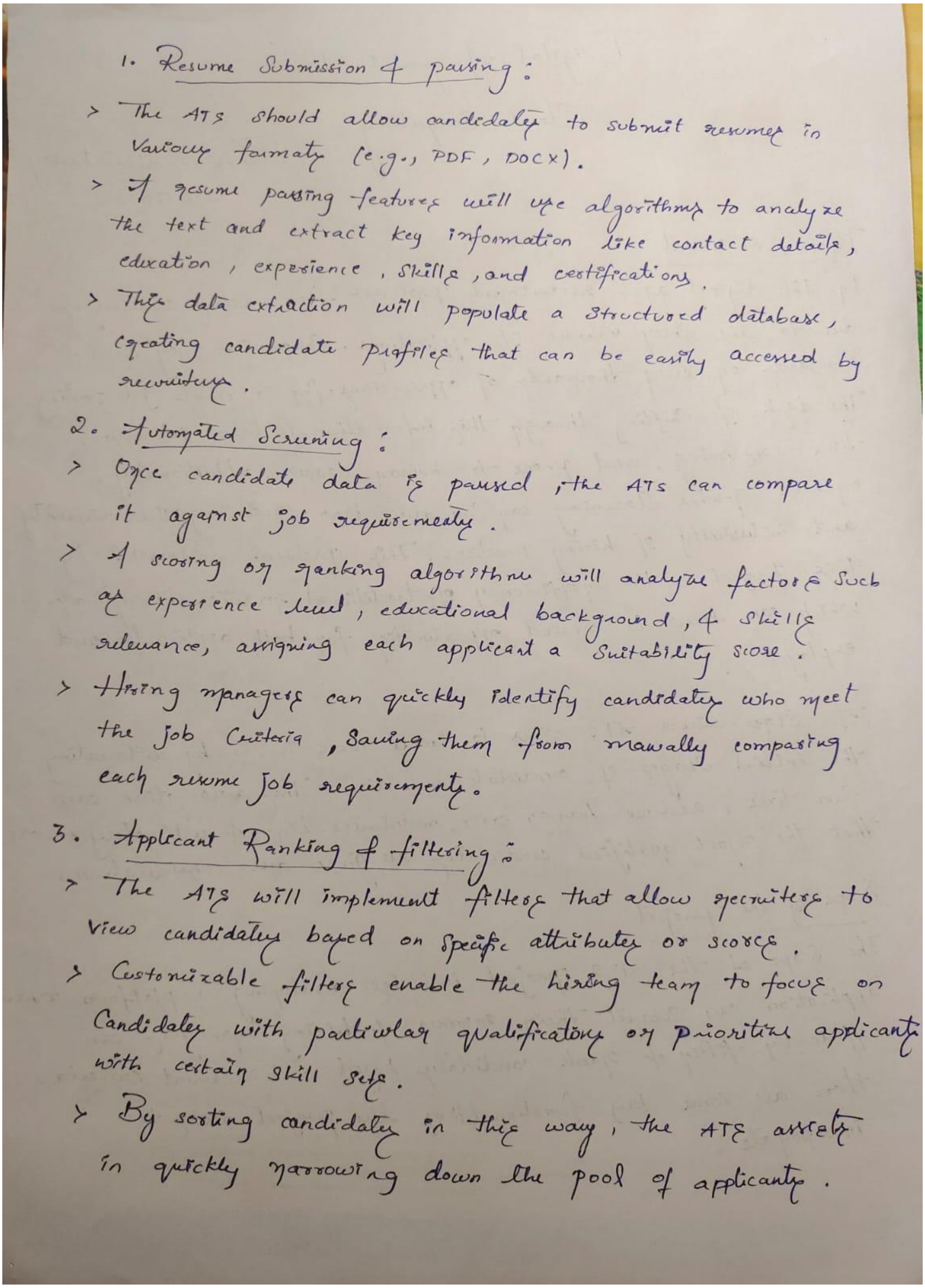
In the digital age, recruitment processes face various challenges due to the sheer volume of applications. Companies may receive hundreds or even thousands of applications for a Single job posting. The task of shifting through this information is labor intensive, time-consuming, and prone to human error. Additionally, manual resume Screening can introduce bias, reducing the diversity and inclusivity of hiring practices. This problem addressed by this Project is the inefficiency of traditional recruitment methods, which can lead to missed opportunities for both applicants and employers.

A Streamlined ATS aims to tackle these issues by automating the initial Stages of candidate evaluation. This automation can Save time, reduce human error, minimize bias, & ultimately ensure that the most qualified candidates are given due consideration.

Scope of the Project

The Scope of the ATS project includes developing a platform where application can Submit their resumes, & hiring managers can efficiently filter & rank candidates based on relevant criteria. Here are some key functionalities the project seeks to incorporate:

Scope of the Project :



1. Resume Submission & parsing :

- The ATS should allow candidates to submit resumes in various formats (e.g., PDF, DOCX).
- A resume parsing features will use algorithms to analyze the text and extract key information like contact details, education, experience, skills, and certifications.
- This data extraction will populate a structured database, creating candidate profiles that can be easily accessed by recruiters.

2. Automated Screening :

- Once candidate data is parsed, the ATS can compare it against job requirements.
- A scoring or ranking algorithm will analyze factors such as experience level, educational background, & skills relevance, assigning each applicant a suitability score.
- Hiring managers can quickly identify candidates who meet the job criteria, saving them from manually comparing each resume job requirements.

3. Applicant Ranking & filtering :

- The ATS will implement filters that allow recruiters to view candidates based on specific attributes or scores.
- Customizable filters enable the hiring team to focus on candidates with particular qualifications or prioritize applicants with certain skill sets.
- By sorting candidates in this way, the ATS assists in quickly narrowing down the pool of applicants.

4. User interface (UI):

> A clean & intutive dashboard will be created using streamlit, an open source Python library that simplifies building web applications.

> Recruiters can view candidate profiles, apply filters, and sort candidates by their scores, making it easy to identify potential hires.

> The UI will be optimized for eficiency, ensuring that users can easily navigate through candidates and perform necessary actions with minimal clicks.

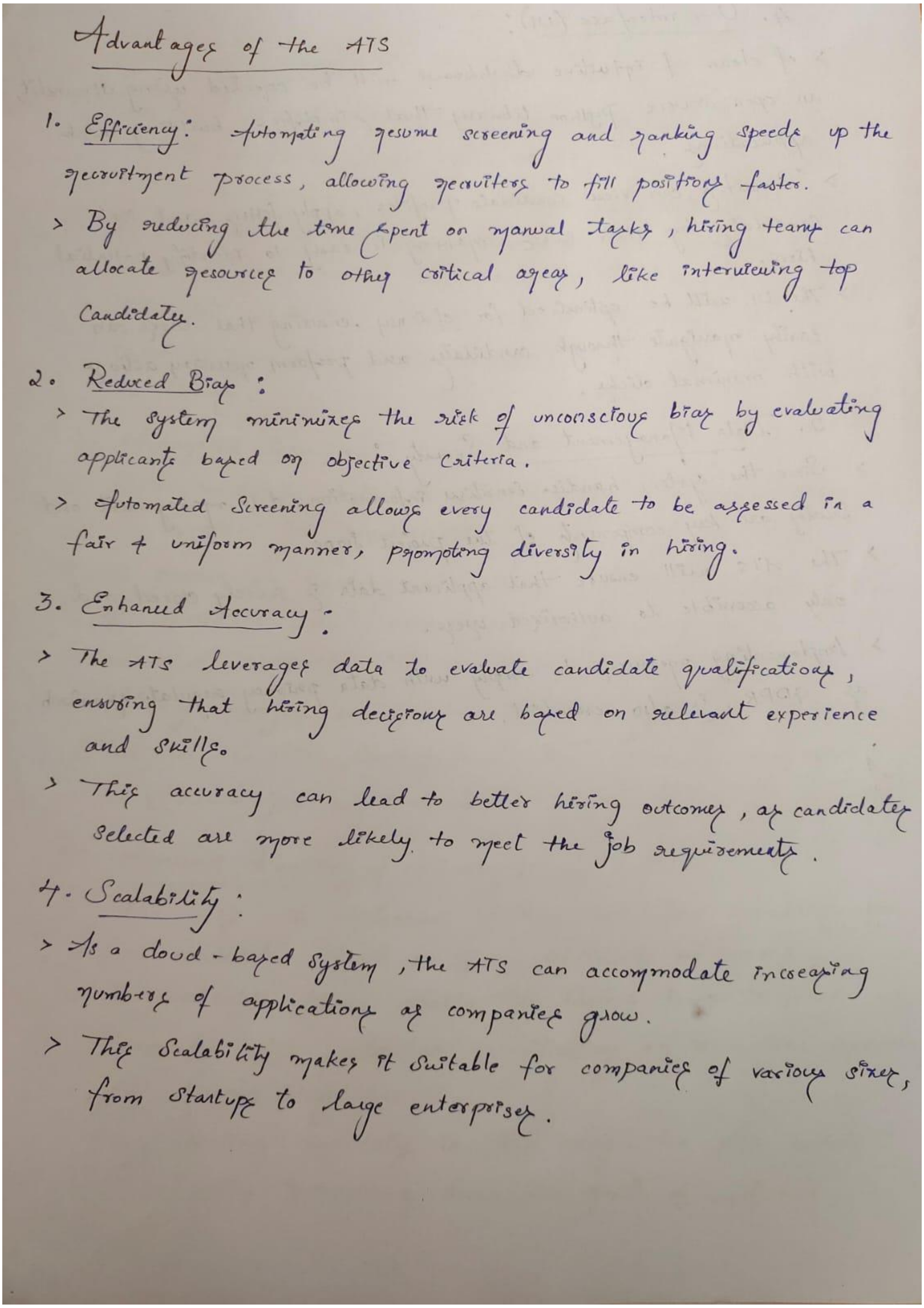
5. Data Management and Security :

> Since the system handles sensitive information, data management and Security are key components of the project scope.

> The ATS will ensure that applicant data is securely stored and only accessible to authorized users.

> Implementing measures to comply with data privacy regulations, such as GDPR, is also essential.

ADVANTAGES:



Advantages of the ATS

1. Efficiency: Automating resume screening and ranking speeds up the recruitment process, allowing recruiters to fill positions faster.

> By reducing the time spent on manual tasks, hiring teams can allocate resources to other critical areas, like interviewing top candidates.

2. Reduced Bias:

> The system minimizes the risk of unconscious bias by evaluating applicants based on objective criteria.

> Automated Screening allows every candidate to be assessed in a fair + uniform manner, promoting diversity in hiring.

3. Enhanced Accuracy:

> The ATS leverages data to evaluate candidate qualifications, ensuring that hiring decisions are based on relevant experience and skills.

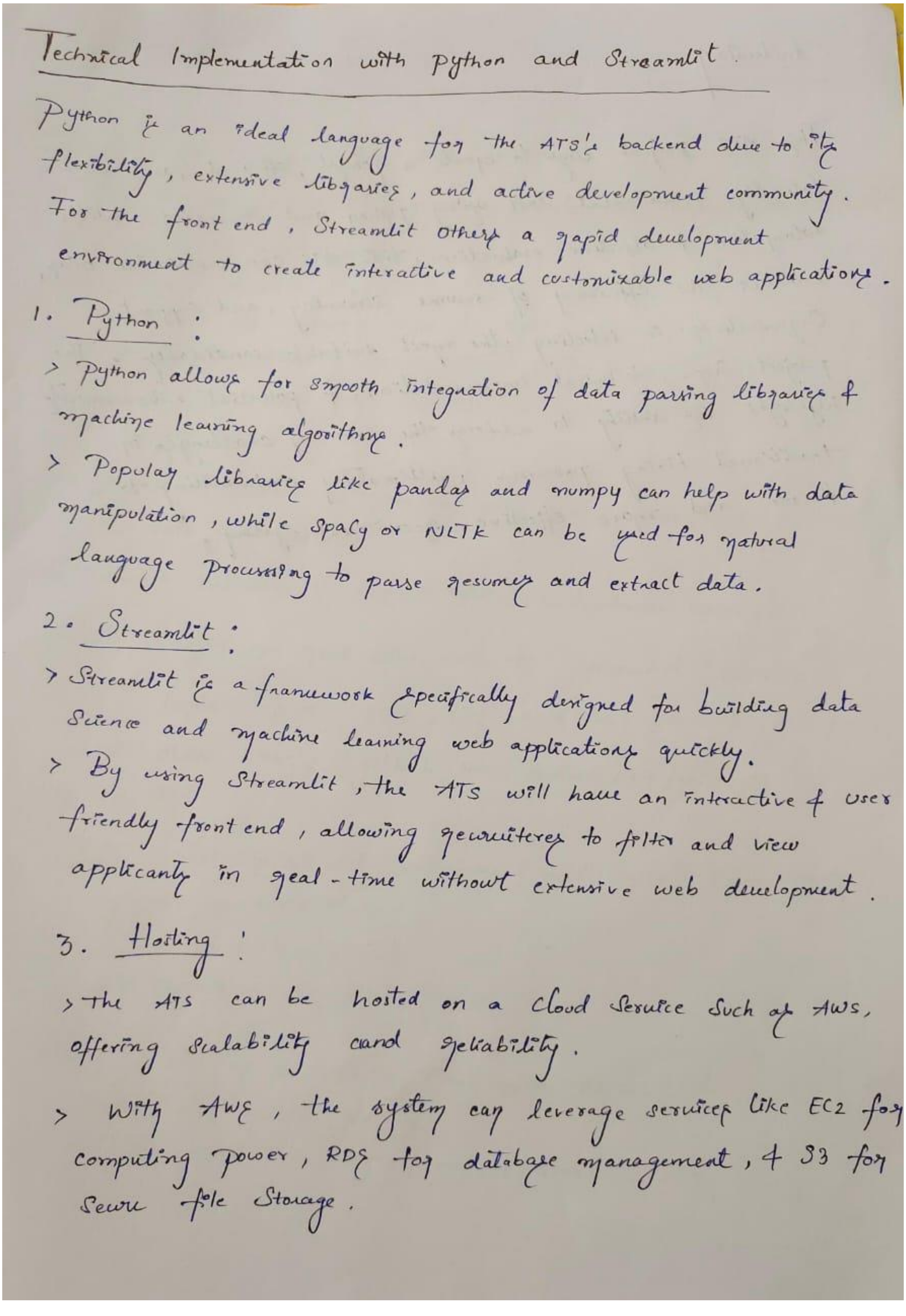
> This accuracy can lead to better hiring outcomes, as candidates selected are more likely to meet the job requirements.

4. Scalability:

> As a cloud-based system, the ATS can accommodate increasing numbers of applications as companies grow.

> This Scalability makes it suitable for companies of various sizes, from Startups to large enterprises.

TECHNICAL IMPLEMENTATION



Technical Implementation with python and Streamlit .

Python is an ideal language for the ATS's backend due to its flexibility, extensive libraries, and active development community. For the front end, Streamlit others a rapid development environment to create interactive and customizable web applications.

1. Python :

> Python allows for smooth integration of data parsing libraries & machine learning algorithms.

> Popular libraries like pandas and numpy can help with data manipulation, while spaCy or NLTK can be used for natural language processing to parse resumes and extract data.

2. Streamlit :

> Streamlit is a framework specifically designed for building data science and machine learning web applications quickly.

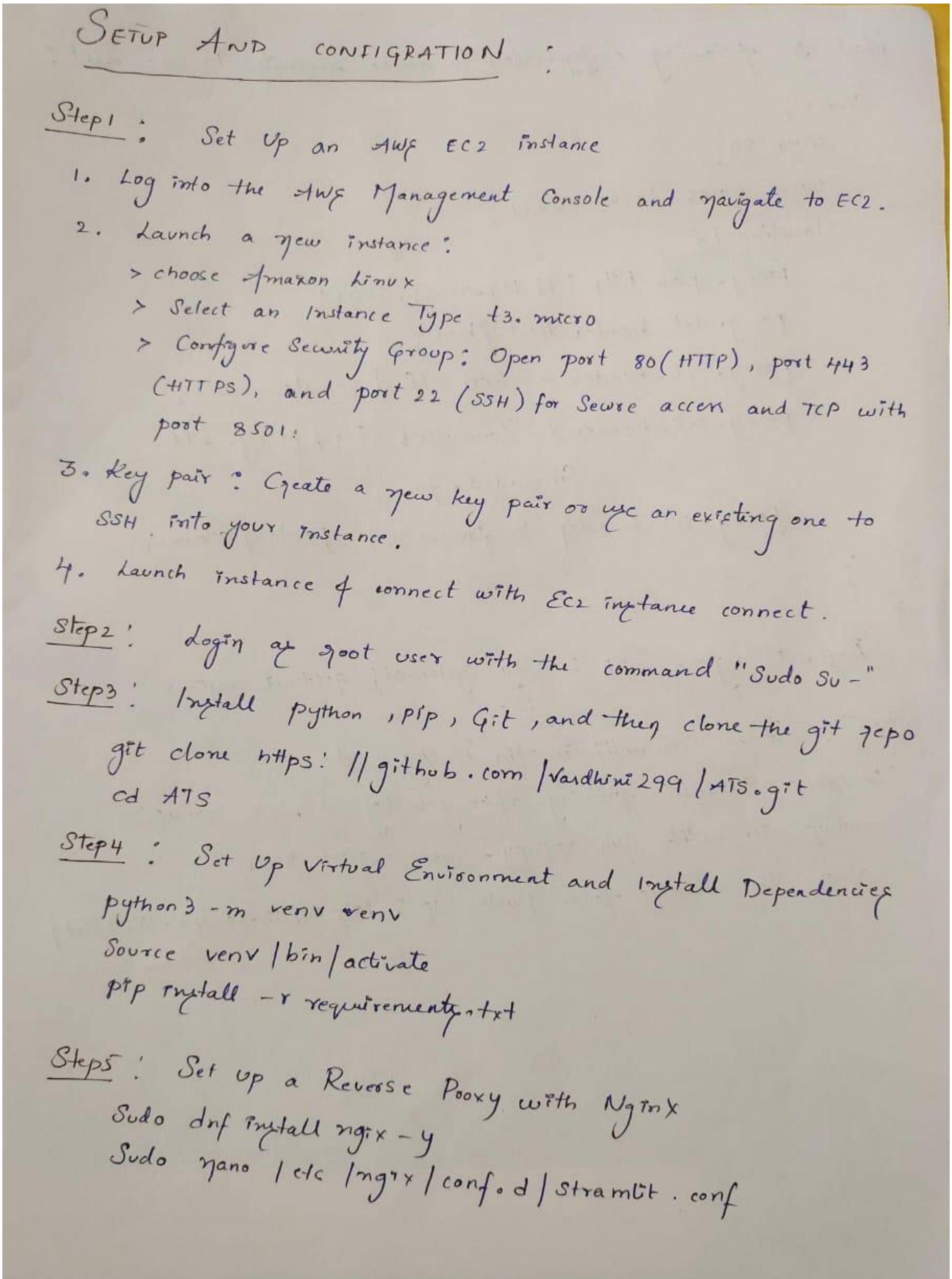
> By using Streamlit, the ATS will have an interactive & user friendly front end, allowing recruiters to filter and view applicants in real-time without extensive web development.

3. Hosting :

> The ATS can be hosted on a cloud service such as AWS, offering scalability cand reliability.

> With AWS, the system can leverage services like EC2 for computing power, RDS for database management, & S3 for secure file storage.

SETUP AND CONFIGRATION :



Setup And configration :

Step1 : Set Up an AWS EC2 instance

1. Log into the AWS Management Console and navigate to EC2.
2. Launch a new instance :
   > choose Amazon Linux
   > Select an Instance Type t3. micro
   > Configure Security Group: Open port 80(HTTP), port 443 (HTTPS), and port 22 (SSH) for Secure access and TCP with port 8501:
3. Key pair : Create a new key pair or use an existing one to SSH into your instance.
4. Launch instance & connect with Ec2 instance connect.

Step2 : login as root user with the command "Sudo Su -"

Step3 : Install python, pip, Git, and then clone the git repo

```
git clone https://github.com/Vardhini299/ATS.git
cd ATS
```

Step4 : Set Up Virtual Environment and Install Dependencies

```
python3 -m venv venv
Source venv/bin/activate
pip install -r requirements.txt
```

Step5 : Set up a Reverse Proxy with Nginx

```
Sudo dnf install nginx -y
Sudo nano /etc/nginx/conf.d/streamlit.conf
```

Add the following configuration to route requests to Streamlit:

```
Server {
   listen 80;
   server_name  13.60.44.70;
   location / {
      proxy_pass http://127.0.0.1:8501;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $proxy_add_x_
                      forwarded_for;
      proxy_set_header X-forwarded-proto $schema;
   }
}
```

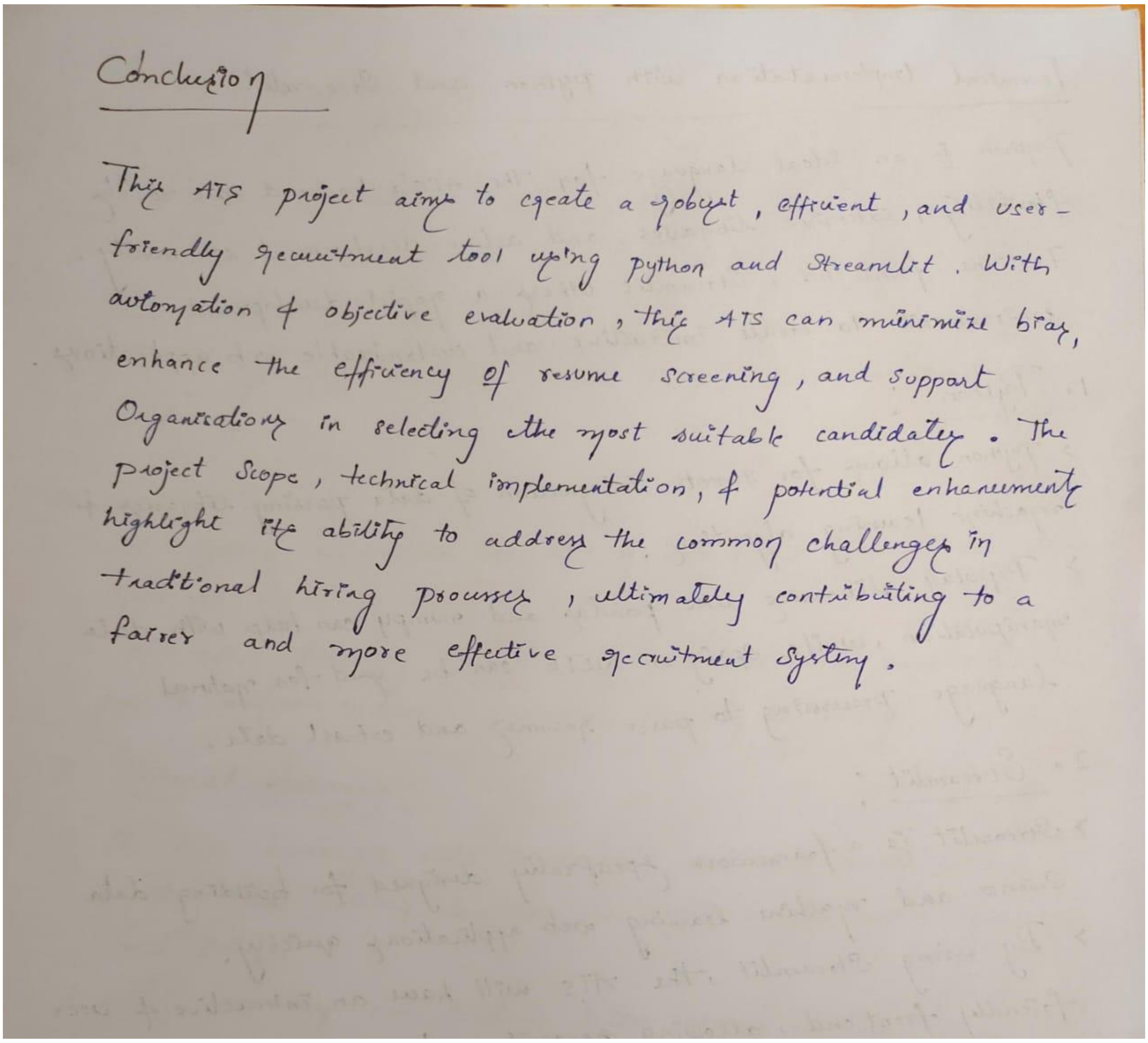
Restart nginx with : Sudo systemctl restart nginx

Step6 : Run Streamlit in the Background

To keep Streamlit running after you log out, use nohup:

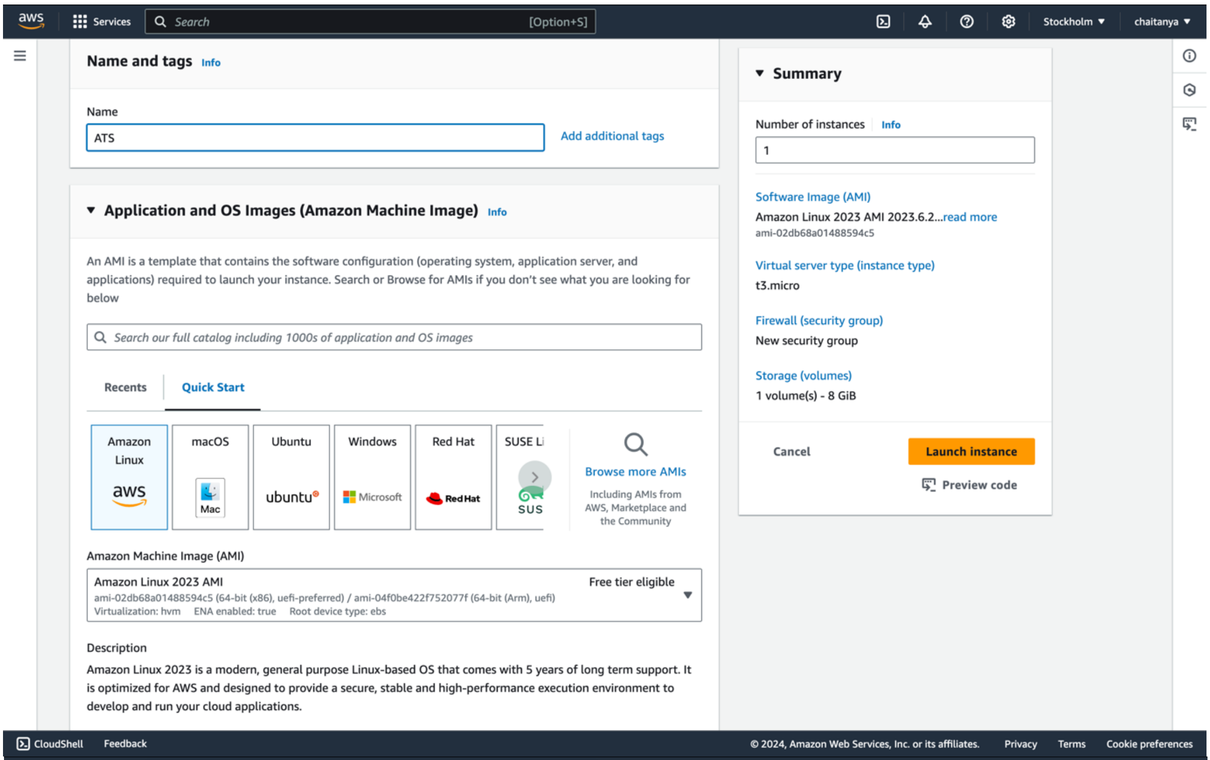
```
nohup streamlit run app.py --server.port 8501
```

access the website from public ip: http://13.60.44.70:8501/



Conclusion

This ATS project aims to create a robust, efficient, and user-friendly recruitment tool using python and Streamlit. With automation & objective evaluation, this ATS can minimize bias, enhance the efficiency of resume screening, and support Organizations in selecting the most suitable candidates. The project Scope, technical implementation, & potential enhancements highlight its ability to address the common challenges in traditional hiring processes, ultimately contributing to a fairer and more effective recruitment system.



Name and tags Info

Name

ATS

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

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-02db68a01488594c5 (64-bit (x86), uefi-preferred) / ami-04f0be422f752077f (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.

▼ Summary

Number of instances Info

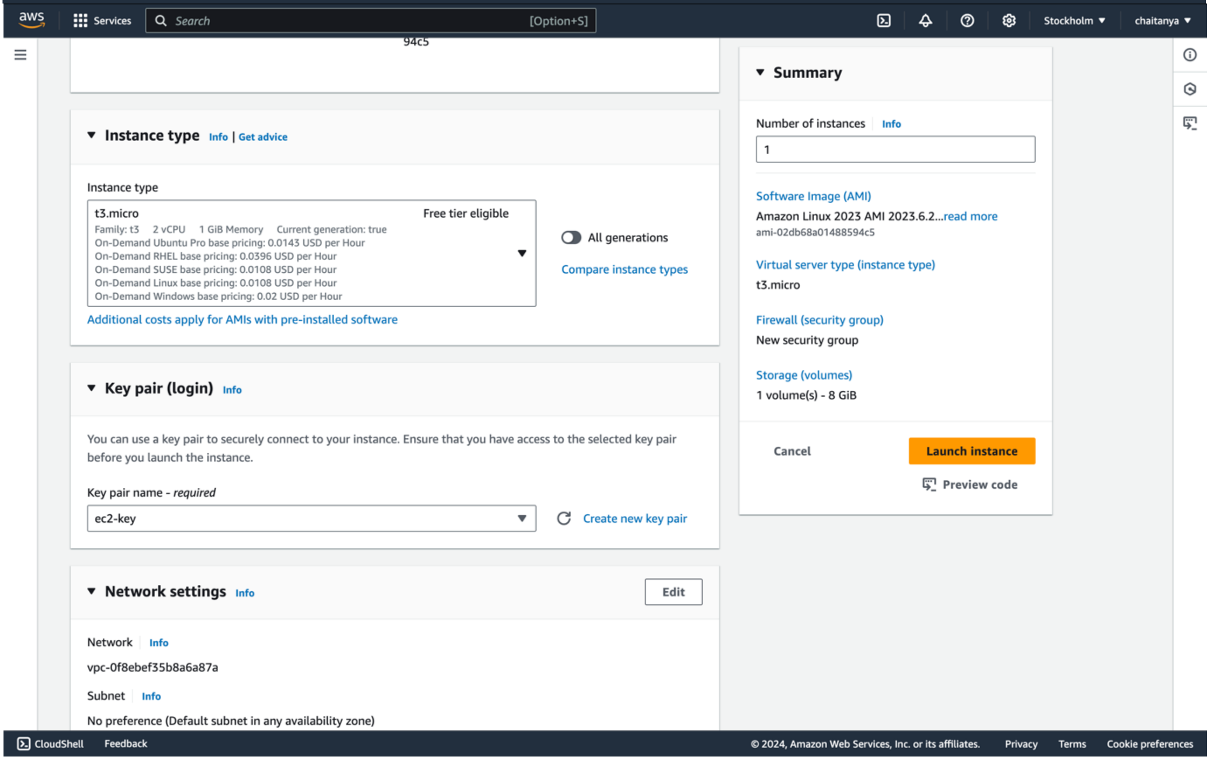
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-02db68a01488594c5

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

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



94c5

▼ Instance type Info | Get advice

Instance type

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

Additional costs apply for AMIs with pre-installed software

▼ 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*

ec2-key

▼ Network settings Info

Network Info

vpc-0f8ebef35b8a6a87a

Subnet Info

No preference (Default subnet in any availability zone)

▼ Summary

Number of instances Info

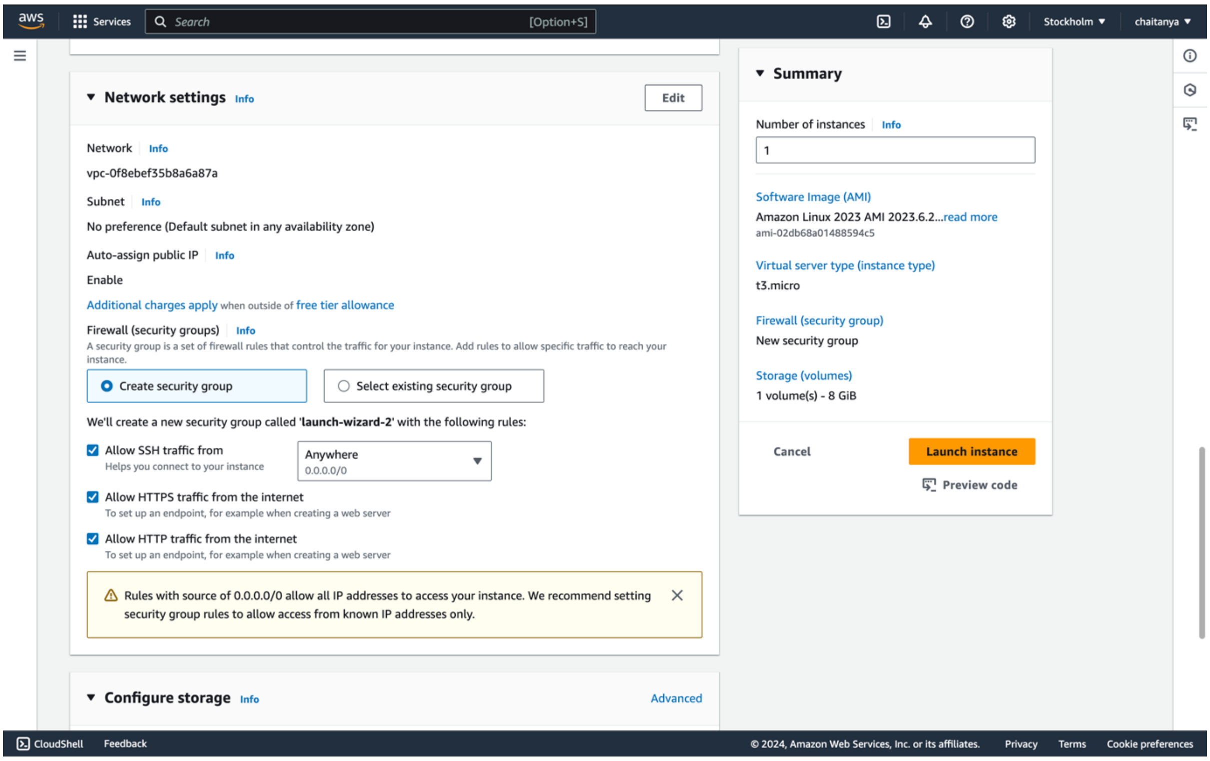
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-02db68a01488594c5

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

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



▼ **Network settings** Info

Edit

Network Info
vpc-0f8ebef35b8a6a87a

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-2**' with the following rules:

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

☑ 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.

▼ **Configure storage** Info
Advanced

▼ **Summary**

Number of instances Info
1

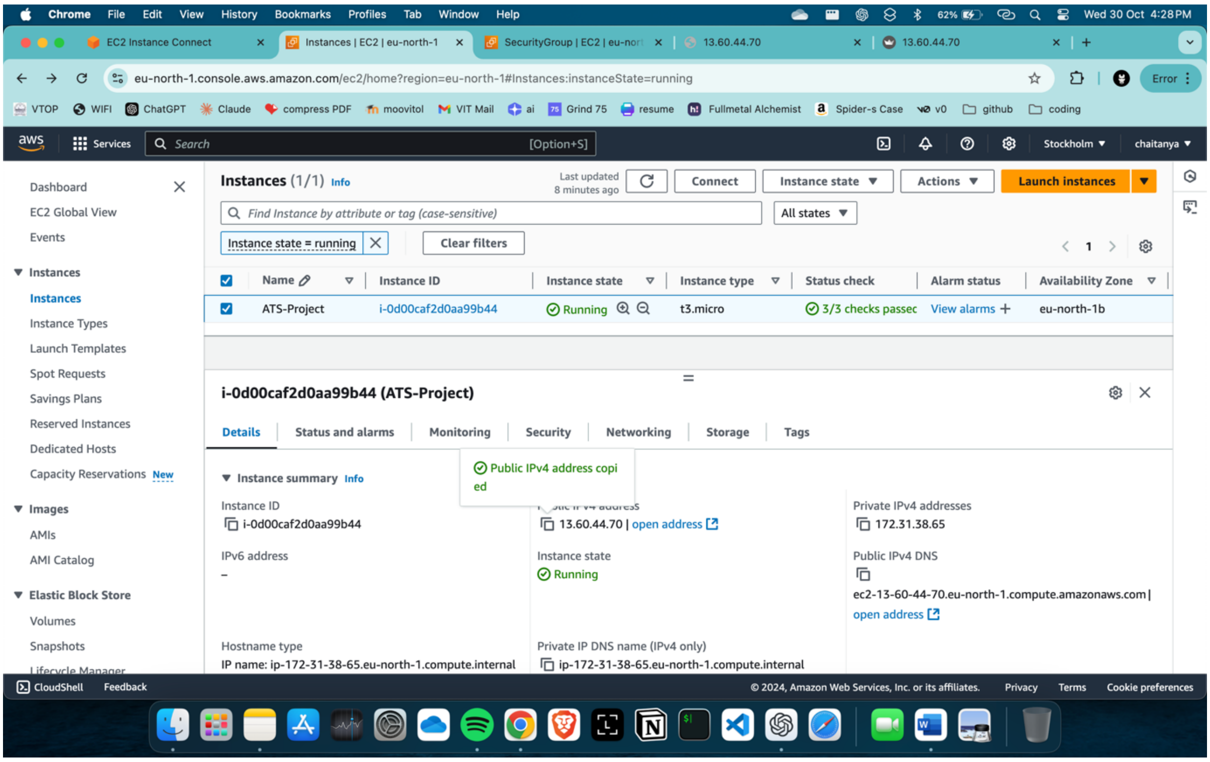
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-02db68a01488594c5

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

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

After launching instance:


Chrome File Edit View History Bookmarks Profiles Tab Window Help
62%
Wed 30 Oct 4:28 PM
EC2 Instance Connect
Instances | EC2 | eu-north-1
SecurityGroup | EC2 | eu-nort
13.60.44.70
13.60.44.70
eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Instances:instanceState=running
Error
VTOP WIFI ChatGPT Claude compress PDF moovitol VIT Mail ai Grind 75 resume Fullmetal Alchemist Spider-s Case v0 github coding
aws
Services
Search
[Option+S]
Stockholm ▼
chaitanya ▼
Dashboard
EC2 Global View
Events
▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
▼ Images
AMIs
AMI Catalog
▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager

Instances (1/1) Info

Last updated 8 minutes ago

Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find Instance by attribute or tag (case-sensitive)
All states ▼
Instance state = running ✕
Clear filters

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	ATS-Project	i-0d00caf2d0aa99b44	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	eu-north-1b

i-0d00caf2d0aa99b44 (ATS-Project)

▼ Instance summary Info

✓ Public IPv4 address copied

Instance ID
i-0d00caf2d0aa99b44

Public IPv4 address
13.60.44.70 | open address

Private IPv4 addresses
172.31.38.65

IPv6 address
–

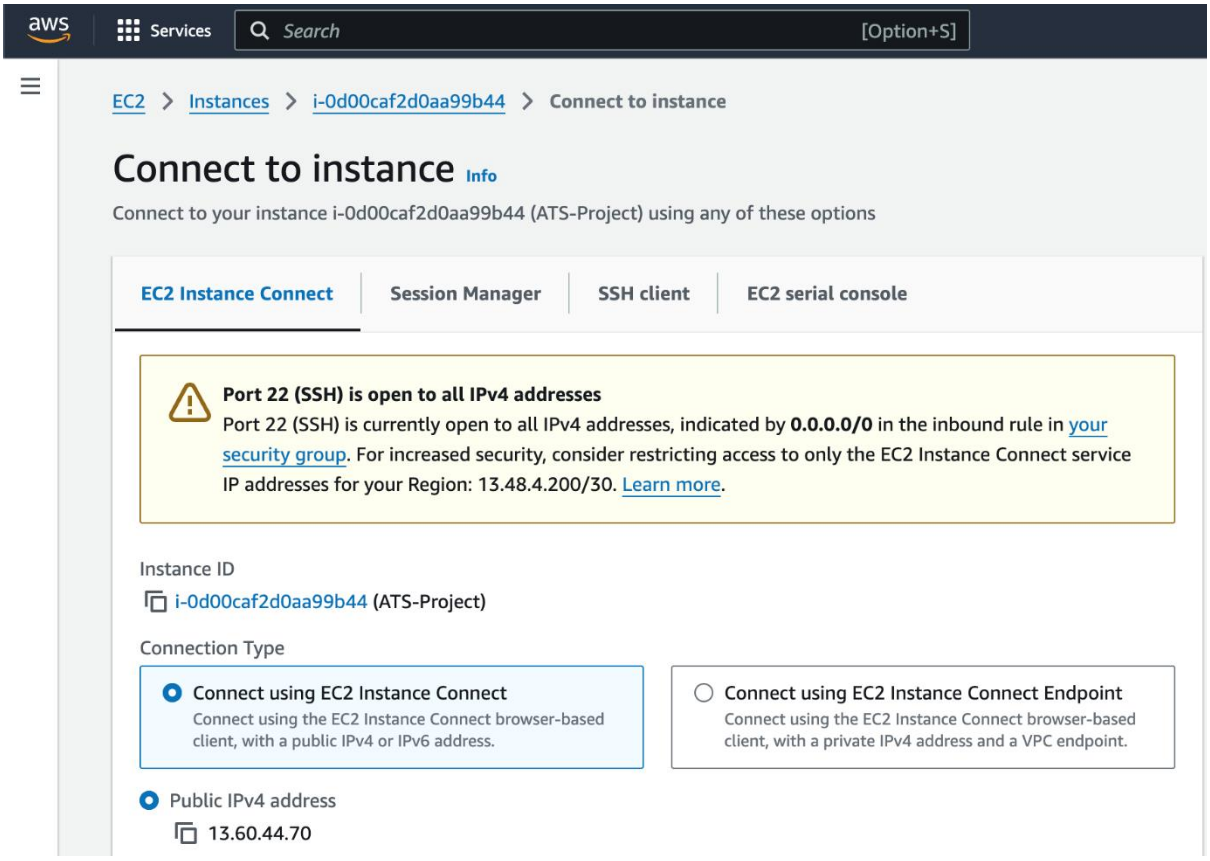
Instance state
✓ Running

Public IPv4 DNS
ec2-13-60-44-70.eu-north-1.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-38-65.eu-north-1.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-38-65.eu-north-1.compute.internal

launch instance and connect with EC2 instance connect :



Connect to instance Info

Connect to your instance i-0d00caf2d0aa99b44 (ATS-Project) using any of these options

Port 22 (SSH) is open to all IPv4 addresses

Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in your security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.48.4.200/30. Learn more.

Instance ID

i-0d00caf2d0aa99b44 (ATS-Project)

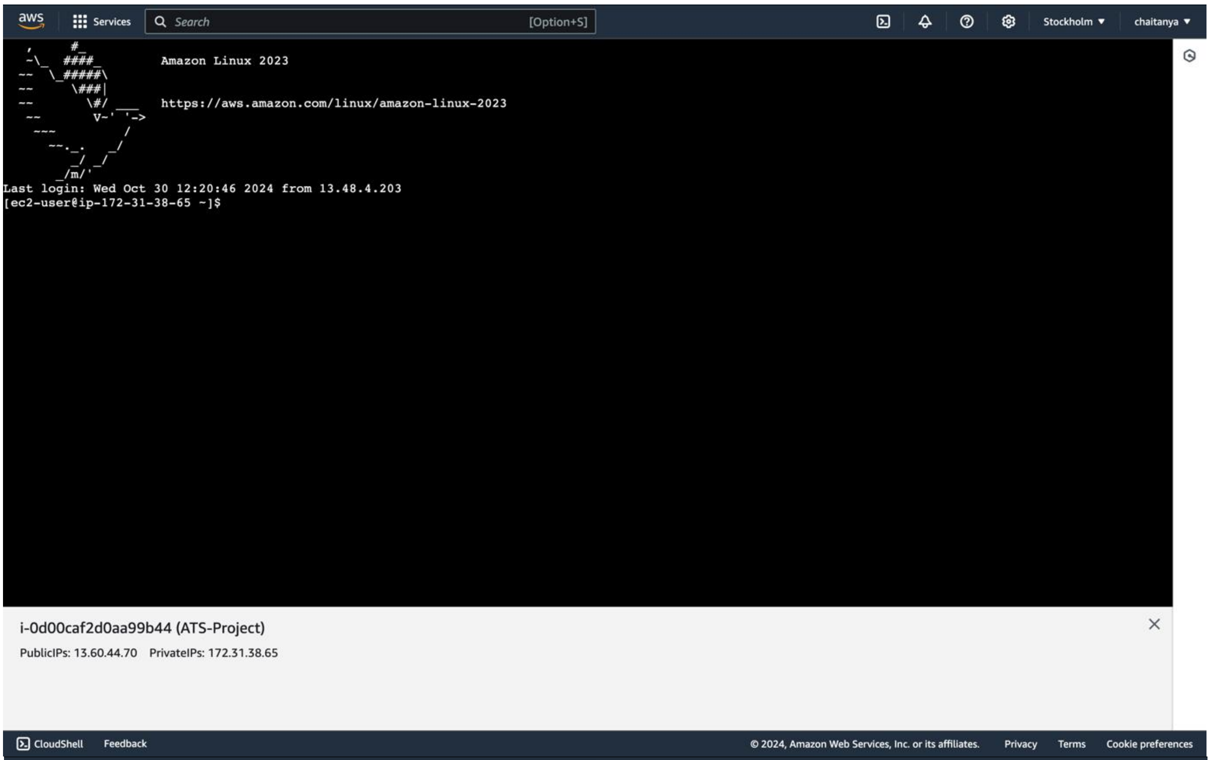
Connection Type

● Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

○ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

● Public IPv4 address

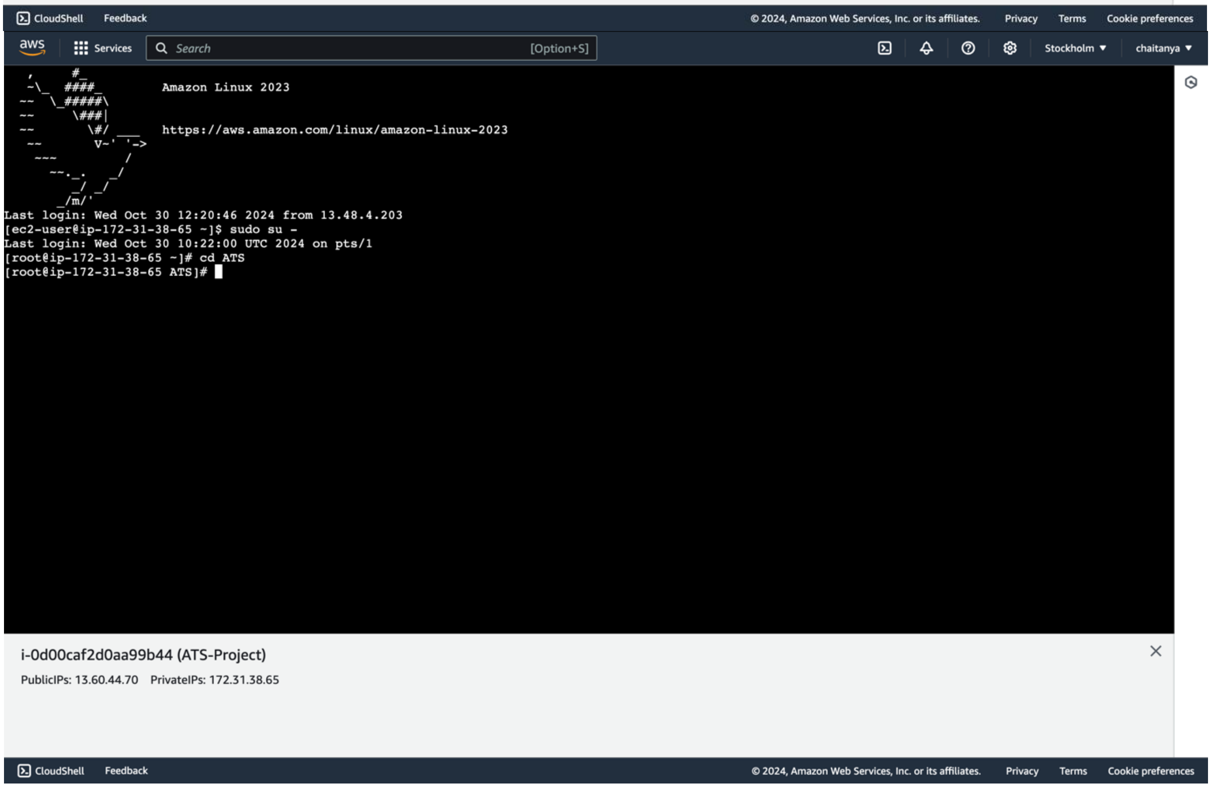
13.60.44.70



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Oct 30 12:20:46 2024 from 13.48.4.203
[ec2-user@ip-172-31-38-65 ~]$
```

i-0d00caf2d0aa99b44 (ATS-Project)

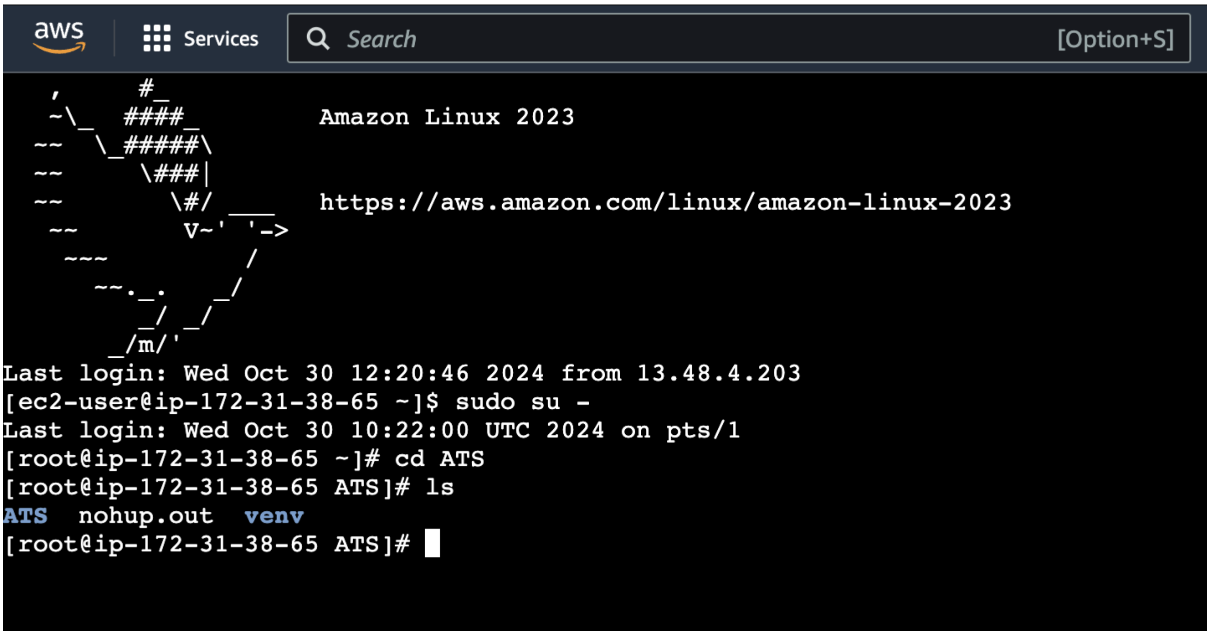
PublicIPs: 13.60.44.70   PrivateIPs: 172.31.38.65



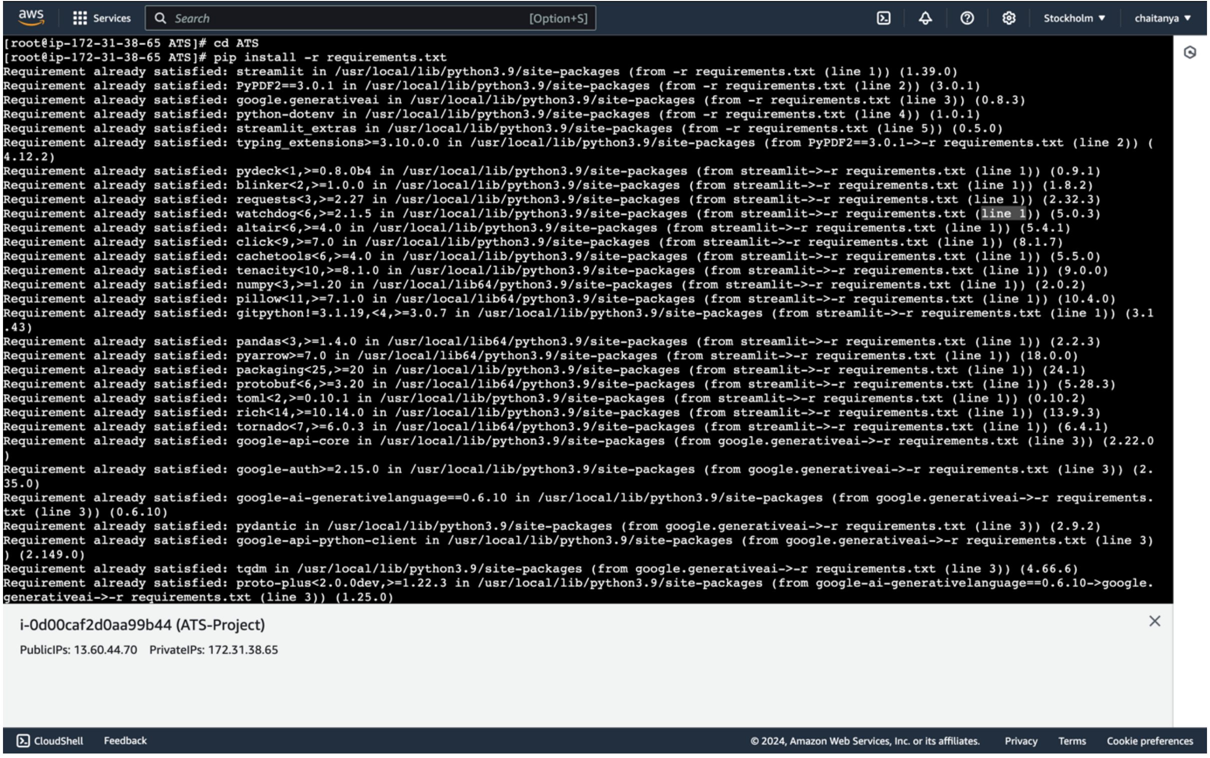
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Oct 30 12:20:46 2024 from 13.48.4.203
[ec2-user@ip-172-31-38-65 ~]$ sudo su -
Last login: Wed Oct 30 10:22:00 UTC 2024 on pts/1
[root@ip-172-31-38-65 ~]# cd ATS
[root@ip-172-31-38-65 ATS]#
```

i-0d00caf2d0aa99b44 (ATS-Project)

PublicIPs: 13.60.44.70   PrivateIPs: 172.31.38.65



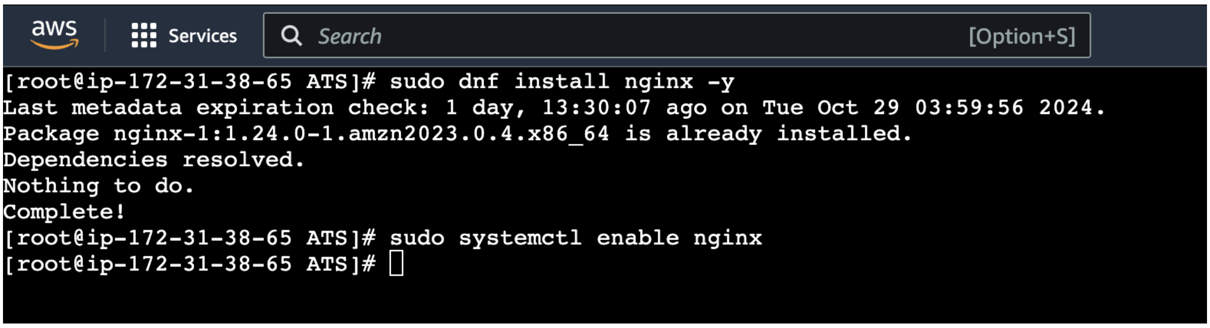
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Oct 30 12:20:46 2024 from 13.48.4.203
[ec2-user@ip-172-31-38-65 ~]$ sudo su -
Last login: Wed Oct 30 10:22:00 UTC 2024 on pts/1
[root@ip-172-31-38-65 ~]# cd ATS
[root@ip-172-31-38-65 ATS]# ls
ATS  nohup.out  venv
[root@ip-172-31-38-65 ATS]# 
```



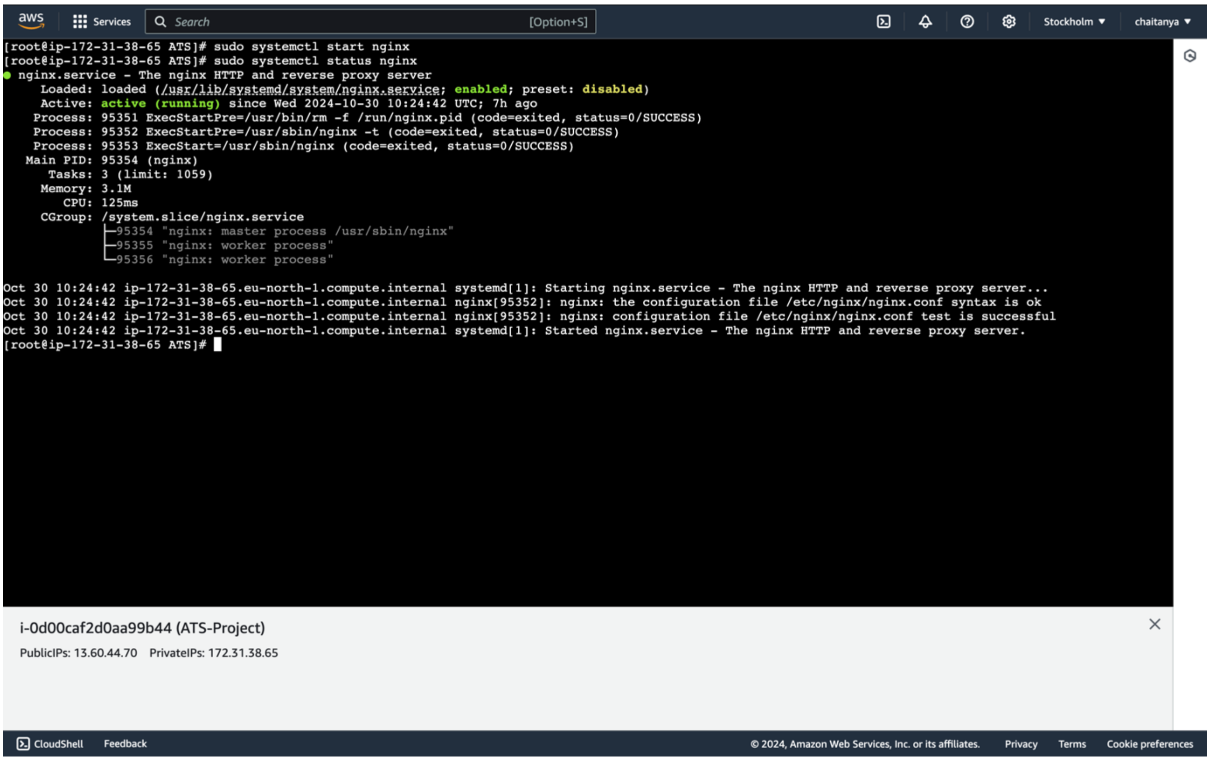
```
[root@ip-172-31-38-65 ATS]# cd ATS
[root@ip-172-31-38-65 ATS]# pip install -r requirements.txt
Requirement already satisfied: streamlit in /usr/local/lib/python3.9/site-packages (from -r requirements.txt (line 1)) (1.39.0)
Requirement already satisfied: PyPDF2==3.0.1 in /usr/local/lib/python3.9/site-packages (from -r requirements.txt (line 2)) (3.0.1)
Requirement already satisfied: google.generativeai in /usr/local/lib/python3.9/site-packages (from -r requirements.txt (line 3)) (0.8.3)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.9/site-packages (from -r requirements.txt (line 4)) (1.0.1)
Requirement already satisfied: streamlit_extras in /usr/local/lib/python3.9/site-packages (from -r requirements.txt (line 5)) (0.5.0)
Requirement already satisfied: typing_extensions>=3.10.0.0 in /usr/local/lib/python3.9/site-packages (from PyPDF2==3.0.1->-r requirements.txt (line 2)) (
4.12.2)
Requirement already satisfied: pydeck<1,>=0.8.0b4 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (0.9.1)
Requirement already satisfied: blinker<2,>=1.0.0 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (1.8.2)
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (2.32.3)
Requirement already satisfied: watchdog<6,>=2.1.5 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (5.0.3)
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (5.4.1)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (8.1.7)
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (5.5.0)
Requirement already satisfied: tenacity<10,>=8.1.0 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (9.0.0)
Requirement already satisfied: numpy<3,>=1.20 in /usr/local/lib64/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (2.0.2)
Requirement already satisfied: pillow<11,>=7.1.0 in /usr/local/lib64/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (10.4.0)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (3.1
.43)
Requirement already satisfied: pandas<3,>=1.4.0 in /usr/local/lib64/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (2.2.3)
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib64/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (18.0.0)
Requirement already satisfied: packaging<25,>=20 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (24.1)
Requirement already satisfied: protobuf<6,>=3.20 in /usr/local/lib64/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (5.28.3)
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (0.10.2)
Requirement already satisfied: rich<14,>=10.14.0 in /usr/local/lib/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (13.9.3)
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib64/python3.9/site-packages (from streamlit->-r requirements.txt (line 1)) (6.4.1)
Requirement already satisfied: google-api-core in /usr/local/lib/python3.9/site-packages (from google.generativeai->-r requirements.txt (line 3)) (2.22.0
)
Requirement already satisfied: google-auth>=2.15.0 in /usr/local/lib/python3.9/site-packages (from google.generativeai->-r requirements.txt (line 3)) (2.
35.0)
Requirement already satisfied: google-ai-generativelanguage==0.6.10 in /usr/local/lib/python3.9/site-packages (from google.generativeai->-r requirements.
txt (line 3)) (0.6.10)
Requirement already satisfied: pydantic in /usr/local/lib/python3.9/site-packages (from google.generativeai->-r requirements.txt (line 3)) (2.9.2)
Requirement already satisfied: google-api-python-client in /usr/local/lib/python3.9/site-packages (from google.generativeai->-r requirements.txt (line 3)
) (2.149.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.9/site-packages (from google.generativeai->-r requirements.txt (line 3)) (4.66.6)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in /usr/local/lib/python3.9/site-packages (from google-ai-generativelanguage==0.6.10->google.
generativeai->-r requirements.txt (line 3)) (1.25.0)
```

i-0d00caf2d0aa99b44 (ATS-Project)

PublicIPs: 13.60.44.70 PrivateIPs: 172.31.38.65



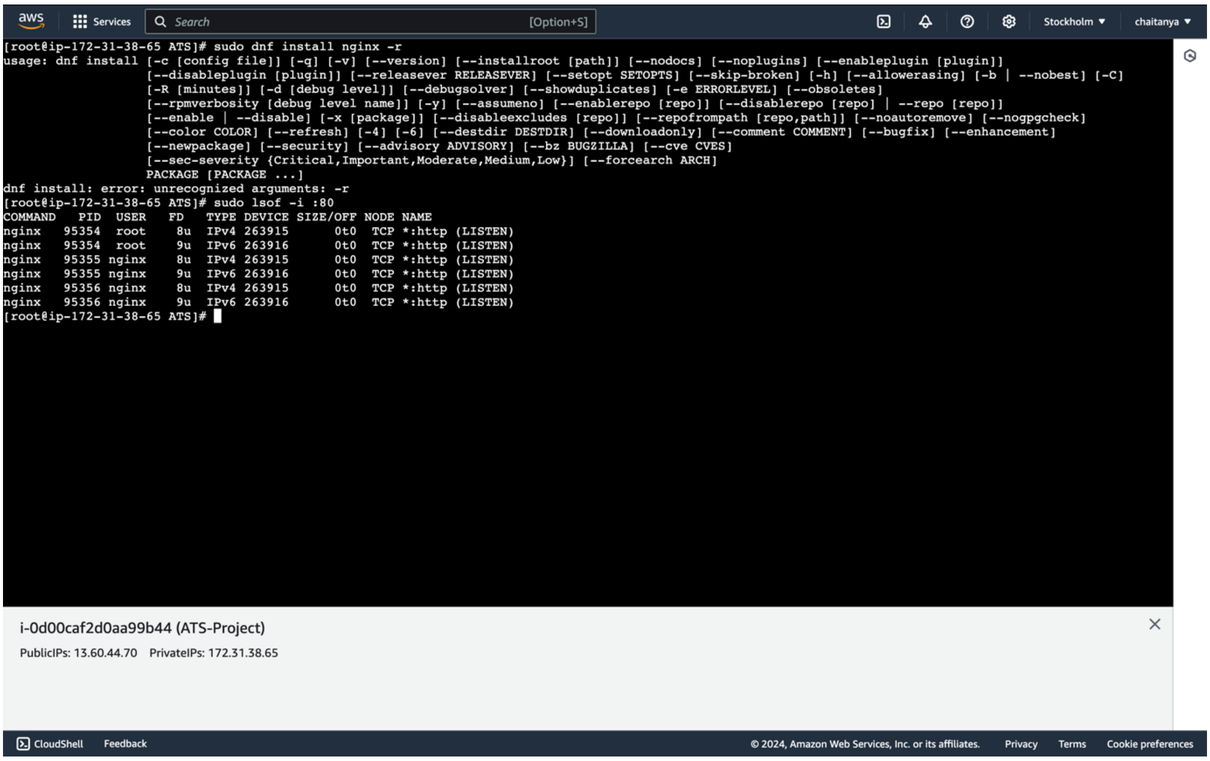
```
[root@ip-172-31-38-65 ATS]# sudo dnf install nginx -y
Last metadata expiration check: 1 day, 13:30:07 ago on Tue Oct 29 03:59:56 2024.
Package nginx-1:1.24.0-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-38-65 ATS]# sudo systemctl enable nginx
[root@ip-172-31-38-65 ATS]# 
```



```
[root@ip-172-31-38-65 ATS]# sudo systemctl start nginx
[root@ip-172-31-38-65 ATS]# sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-10-30 10:24:42 UTC; 7h ago
    Process: 95351 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 95352 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 95353 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 95354 (nginx)
      Tasks: 3 (limit: 1059)
     Memory: 3.1M
        CPU: 125ms
     CGroup: /system.slice/nginx.service
             ├─95354 "nginx: master process /usr/sbin/nginx"
             ├─95355 "nginx: worker process"
             └─95356 "nginx: worker process"

Oct 30 10:24:42 ip-172-31-38-65.eu-north-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Oct 30 10:24:42 ip-172-31-38-65.eu-north-1.compute.internal nginx[95352]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Oct 30 10:24:42 ip-172-31-38-65.eu-north-1.compute.internal nginx[95352]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Oct 30 10:24:42 ip-172-31-38-65.eu-north-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-172-31-38-65 ATS]# 
```

i-0d00caf2d0aa99b44 (ATS-Project)

PublicIPs: 13.60.44.70 PrivateIPs: 172.31.38.65

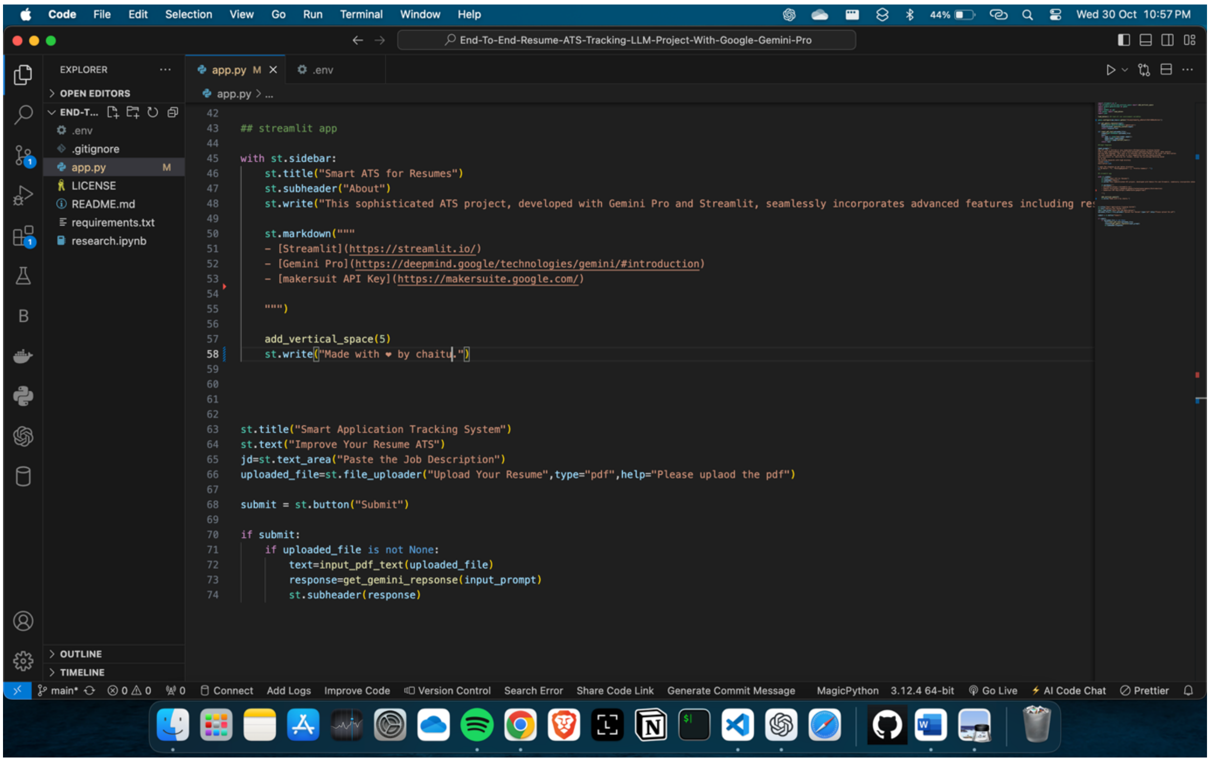


```
[root@ip-172-31-38-65 ATS]# sudo dnf install nginx -r
usage: dnf install [-c [config file]] [-q] [-v] [--version] [--installroot [path]] [--nodocs] [--noplugins] [--enableplugin [plugin]]
                   [--disableplugin [plugin]] [--releasever RELEASEVER] [--setopt SETOPTS] [--skip-broken] [-h] [--allowerasing] [-b | --nobest] [-C]
                   [-R [minutes]] [-d [debug level]] [--debugsolver] [--showduplicates] [-e ERRORLEVEL] [--obsoletes]
                   [--rpmverbosity [debug level name]] [-y] [--assumeno] [--enablerepo [repo]] [--disablerepo [repo] | --repo [repo]]
                   [--enable | --disable] [-x [package]] [--disableexcludes [repo]] [--repofrompath [repo,path]] [--noautoremove] [--nogpgcheck]
                   [--color COLOR] [--refresh] [-4] [-6] [--destdir DESTDIR] [--downloadonly] [--comment COMMENT] [--bugfix] [--enhancement]
                   [--newpackage] [--security] [--advisory ADVISORY] [--bz BUGZILLA] [--cve CVES]
                   [--sec-severity {Critical,Important,Moderate,Medium,Low}] [--forcearch ARCH]
                   PACKAGE [PACKAGE ...]
dnf install: error: unrecognized arguments: -r
[root@ip-172-31-38-65 ATS]# sudo lsof -i :80
COMMAND   PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
nginx   95354  root    8u  IPv4 263915      0t0  TCP *:http (LISTEN)
nginx   95354  root    9u  IPv6 263916      0t0  TCP *:http (LISTEN)
nginx   95355 nginx    8u  IPv4 263915      0t0  TCP *:http (LISTEN)
nginx   95355 nginx    9u  IPv6 263916      0t0  TCP *:http (LISTEN)
nginx   95356 nginx    8u  IPv4 263915      0t0  TCP *:http (LISTEN)
nginx   95356 nginx    9u  IPv6 263916      0t0  TCP *:http (LISTEN)
[root@ip-172-31-38-65 ATS]# █
```

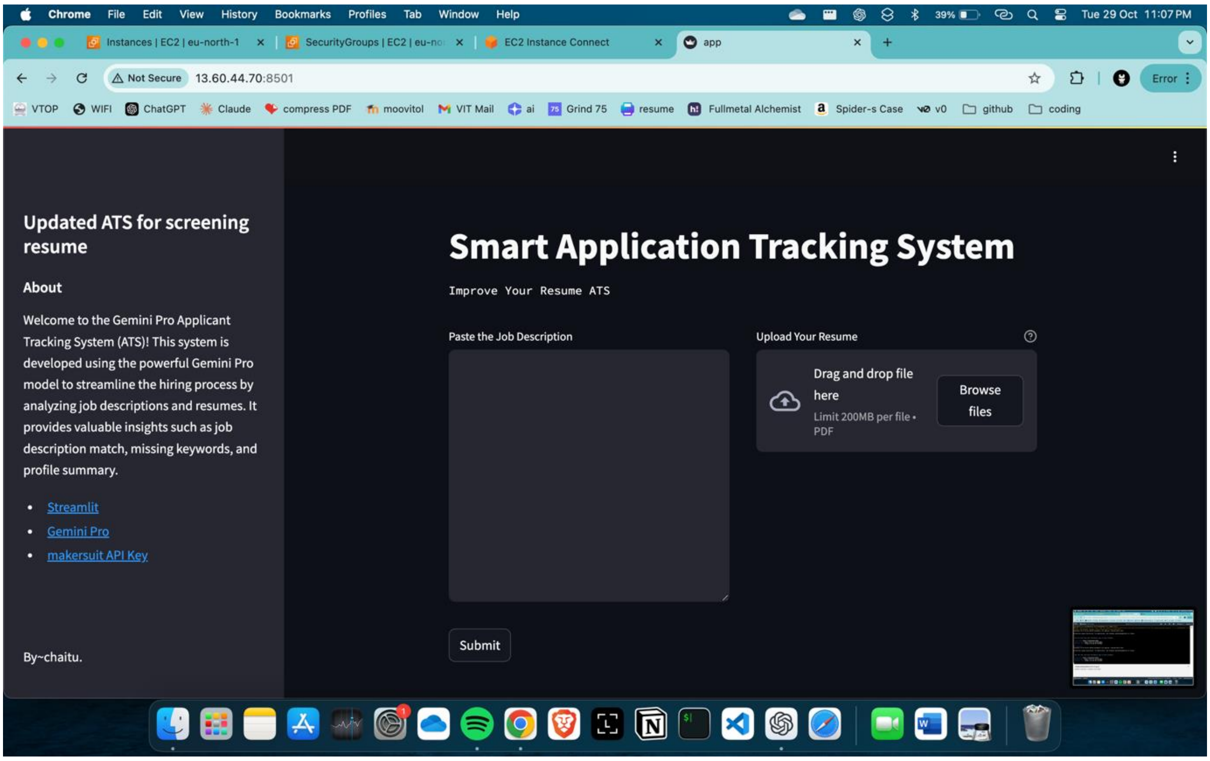
i-0d00caf2d0aa99b44 (ATS-Project)

PublicIPs: 13.60.44.70   PrivateIPs: 172.31.38.65

App.py:



```python

## streamlit app

with st.sidebar:
    st.title("Smart ATS for Resumes")
    st.subheader("About")
    st.write("This sophisticated ATS project, developed with Gemini Pro and Streamlit, seamlessly incorporates advanced features including res

    st.markdown("""
    - [Streamlit](https://streamlit.io/)
    - [Gemini Pro](https://deepmind.google/technologies/gemini/#introduction)
    - [makersuit API Key](https://makersuite.google.com/)

    """)

    add_vertical_space(5)
    st.write("Made with ❤ by chaitu.")




st.title("Smart Application Tracking System")
st.text("Improve Your Resume ATS")
jd=st.text_area("Paste the Job Description")
uploaded_file=st.file_uploader("Upload Your Resume",type="pdf",help="Please uplaod the pdf")

submit = st.button("Submit")

if submit:
    if uploaded_file is not None:
        text=input_pdf_text(uploaded_file)
        response=get_gemini_repsonse(input_prompt)
        st.subheader(response)
```


Chrome File Edit View History Bookmarks Profiles Tab Window Help
39% Tue 29 Oct 11:07 PM
Instances | EC2 | eu-north-1
SecurityGroups | EC2 | eu-no
EC2 Instance Connect
app
Not Secure 13.60.44.70:8501
Error
VTOP WIFI ChatGPT Claude compress PDF moovitol VIT Mail ai Grind 75 resume Fullmetal Alchemist Spider-s Case v0 github coding

Updated ATS for screening resume

About

Welcome to the Gemini Pro Applicant Tracking System (ATS)! This system is developed using the powerful Gemini Pro model to streamline the hiring process by analyzing job descriptions and resumes. It provides valuable insights such as job description match, missing keywords, and profile summary.

- Streamlit
- Gemini Pro
- makersuit API Key

By~chaitu.

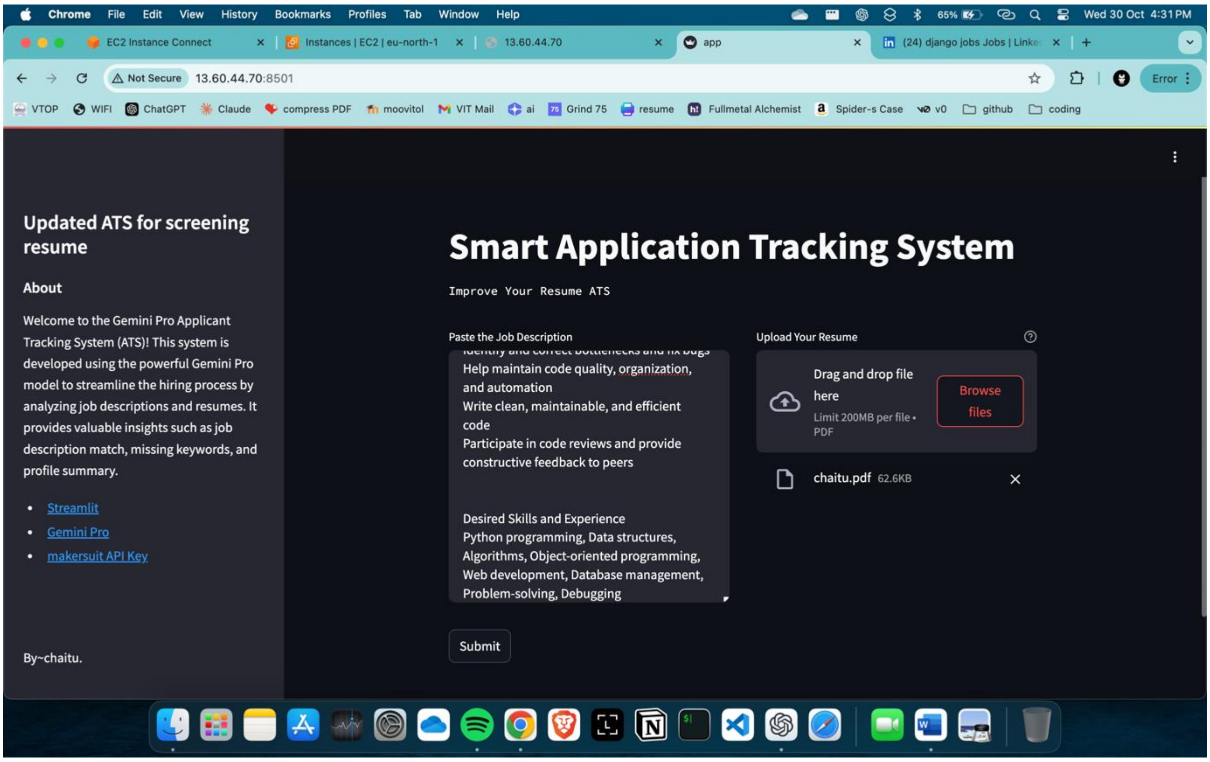
Smart Application Tracking System

Improve Your Resume ATS

Paste the Job Description

Upload Your Resume

Drag and drop file here
Limit 200MB per file • PDF


Chrome File Edit View History Bookmarks Profiles Tab Window Help
65% Wed 30 Oct 4:31 PM
EC2 Instance Connect
Instances | EC2 | eu-north-1
13.60.44.70
app
(24) django jobs Jobs | Linke
Not Secure 13.60.44.70:8501
Error
VTOP WIFI ChatGPT Claude compress PDF moovitol VIT Mail ai Grind 75 resume Fullmetal Alchemist Spider-s Case v0 github coding

Updated ATS for screening resume

About

Welcome to the Gemini Pro Applicant Tracking System (ATS)! This system is developed using the powerful Gemini Pro model to streamline the hiring process by analyzing job descriptions and resumes. It provides valuable insights such as job description match, missing keywords, and profile summary.

- Streamlit
- Gemini Pro
- makersuit API Key

By~chaitu.

Smart Application Tracking System

Improve Your Resume ATS

Paste the Job Description

Identify and correct bottlenecks and fix bugs
Help maintain code quality, organization, and automation
Write clean, maintainable, and efficient code
Participate in code reviews and provide constructive feedback to peers

Desired Skills and Experience
Python programming, Data structures, Algorithms, Object-oriented programming, Web development, Database management, Problem-solving, Debugging

Upload Your Resume

Drag and drop file here
Limit 200MB per file • PDF

chaitu.pdf 62.6KB

Access the website from public IP: http://13.60.44.70:8501/

OUTPUT:
{{
"JD Match": "85%"'

"Missing Keywords": ["Advanced Data Wrangling Techniques", "Machine Learning Frameworks"]

"Profile Summary": "An experienced Data Scientist with a proven track record of developing and implementing data-driven solutions. Demonstrated expertise in utilizing statistical techniques, machine learning algorithms, and big data platforms to extract insights and drive business outcomes. Skilled in data wrangling, feature engineering, model selection, and optimization."
}}