AWS Mini Lab with Proxy, Patch Management, and DB Backup

Objective:

To design and deploy a **secure, automated, and cost-effective hybrid infrastructure on AWS** that includes:

- A **public proxy EC2 instance** for internet access,
- A **private EC2 instance** for internal services,
- **Automated patch management** using scripts or tools (e.g., Ansible),
- And **scheduled backup automation** of critical data to Amazon S3.

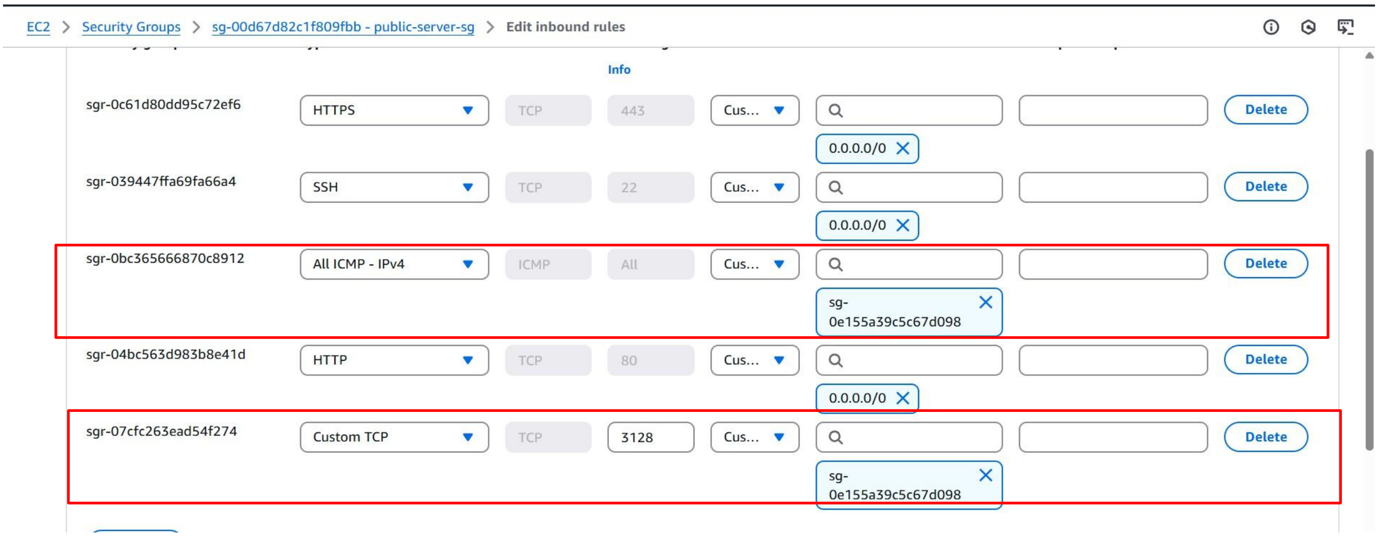
This setup simulates a real-world enterprise environment, focusing on **security, automation, and best practices** in AWS infrastructure management.

Tools & Technologies Used:

Cloud Platform	AWS (EC2, VPC, S3, IAM)
OS & Scripting	Ubuntu, Shell scripting, Cron
Proxy Service	Squid Proxy
Patch Management	Ansible
Database	MySQL
Programming Language	Python
Monitoring/Logging	System logs, cron log, /var/log/

Process:

- ➢ Create vpc for selected region
- ➢ Create subnets 1. Public subnet 2. Private subnet
- ➢ Create a igw and attached public route table
- ➢ Launch ubuntu server 1. Public server 2. Private server
- ➢ Change the security groups
    - o First public server sg

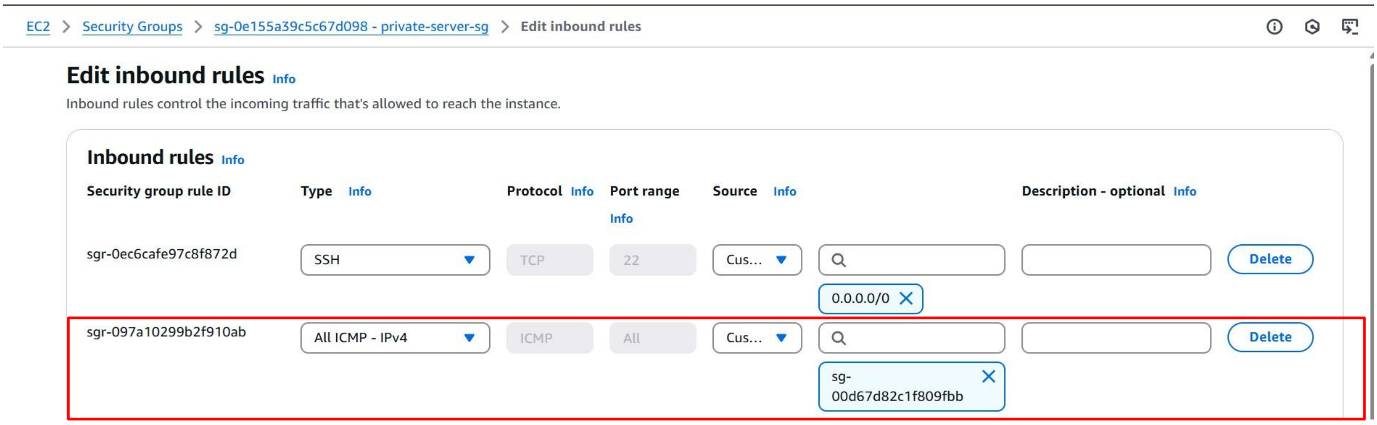


EC2 > Security Groups > sg-00d67d82c1f809fbb - public-server-sg > Edit inbound rules

			Info				
sgr-0c61d80dd95c72ef6	HTTPS	TCP	443	Cus...	0.0.0.0/0		Delete
sgr-039447ffa69fa66a4	SSH	TCP	22	Cus...	0.0.0.0/0		Delete
sgr-0bc365666870c8912	All ICMP - IPv4	ICMP	All	Cus...	sg-0e155a39c5c67d098		Delete
sgr-04bc563d983b8e41d	HTTP	TCP	80	Cus...	0.0.0.0/0		Delete
sgr-07cfc263ead54f274	Custom TCP	TCP	3128	Cus...	sg-0e155a39c5c67d098		Delete

Give here custom private sg id

- ➢ Change the security groups:
  - ○ Private sg



EC2 > Security Groups > sg-0e155a39c5c67d098 - private-server-sg > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0ec6cafe97c8f872d	SSH	TCP	22	Cus...	0.0.0.0/0		Delete
sgr-097a10299b2f910ab	All ICMP - IPv4	ICMP	All	Cus...	sg-00d67d82c1f809fbb		Delete

Give here public sg id.

- ➢ Connect to public server
- ➢ Executive this cmds:
  - ○ sudo apt update
  - ○ sudo apt install squid -y
  - ○ sudo nano /etc/squid/squid.conf (alt + /) paste this one

```
acl allowed_ip src <private server private ip>

http_access allow allowed_ip
```

Check the line number 1625----- http_access allow all

(Alt + Shift + #) # This will toggle **line numbers** ON or OFF.

Check the line number 2175 ---- http_port 3128

  - ctrl+x
  - sudo systemctl restart squid
  - sudo systemctl enable squid
- Connect to private server
- Attach a role ec2-s3
- Execute this cmds:
  - vi /etc/environment

    Paste --- export http_proxy=http://<Public-server-pri-ip>:3128
    export https_proxy=http://<Public-server-pri-ip> :3128
    export no_proxy="169.254.169.254,localhost,127.0.0.1"

  - source /etc/environment
  - curl -h google.com #just test only
  - Ping publicserver-pri-ip
  - sudo apt install python3 python3-pip -y
  - python3 --version
  - pip3 --version
  - sudo apt install mysql-server -y
  - sudo mysql_secure_installation
  - sudo systemctl status mysql
  - sudo systemctl start mysql
  - sudo systemctl enable mysql
  - sudo mysql #insert the data in database
  - create bucket with disable block public access
  - aws s3 ls s3://ansible-proj
  - curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  - apt install unzip
  - unzip awscliv2.zip
  - sudo ./aws/install
  - aws –version
  - aws configure

  -
    ```
    TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" \
      -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
    ```

  -
    ```
    curl -H "X-aws-ec2-metadata-token: $TOKEN" \
      http://169.254.169.254/latest/meta-data/iam/security-credentials/
    ```

  -
    ```
    aws s3 ls s3://ansible-proj
    ```

- write a code in python

  -
    ```
    vi backup_to_s3.py
    ```

```
import boto3
import os
import datetime

#Define bucket name here
bucket_name = 'ansible-proj' #here replace bucket name

timestamp = datetime.datetime.now().strftime('%Y-%m-%d_%H-%M')
backup_file = f"/tmp/backup_{timestamp}.sql"

#Perform MySQL dump
os.system(f"mysqldump -u root sampledb > {backup_file}")

#Upload to S3
s3 = boto3.client('s3')
s3.upload_file(backup_file, bucket_name, f"backup_{timestamp}.sql")
print("Backup completed and uploaded to S3")
```

INSTALL ANSIBLE IN PUBLIC SERVER

```
sudo apt update

sudo apt upgrade -y

sudo apt install ansible -y

ansible --version

sudo nano /etc/ansible/hosts
```

- here menstion the group inventory name and paste private ip

TEST CONNECTION:

- create one playbook file vi apache.yml

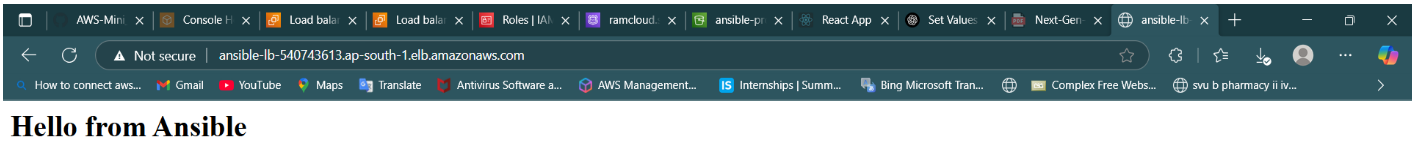
```
---
- name: Simple Apache Web Server Setup
  hosts: webservers
  become: yes

  tasks:
    - name: Install Apache
      yum:
        name: httpd
        state: present

    - name: Start Apache
      service:
        name: httpd
        state: started
        enabled: yes

    - name: Create a simple index.html
      copy:
        dest: /var/www/html/index.html
        content: "<h1>Hello from Ansible</h1>"
```

- ansible-playbook -i /etc/ansible/hosts apache.yml
- The ansible successfully execute the task in the private server using the squid proxy
- create an ALB, AUTOSCALING GROUP and Route53 for access the application securely, high available, scalable

(or)

- For internet testing purpose we can use this also:
- ansible all -i hosts -m ping
- crontab -e
  - select 1
  - paste------> ***/5 * * * * /root/bin/python3 /root/backup_to_s3.py >> /root/db_backup.log 2>&1**

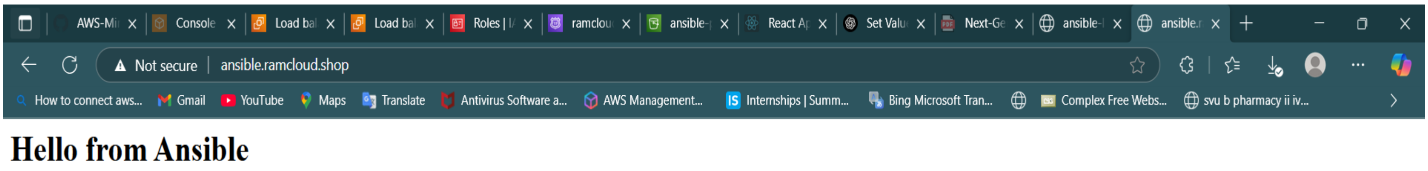
- cat /root/db_backup.log #check the logs
- aws s3 ls s3://ansible-proj



Hello from Ansible

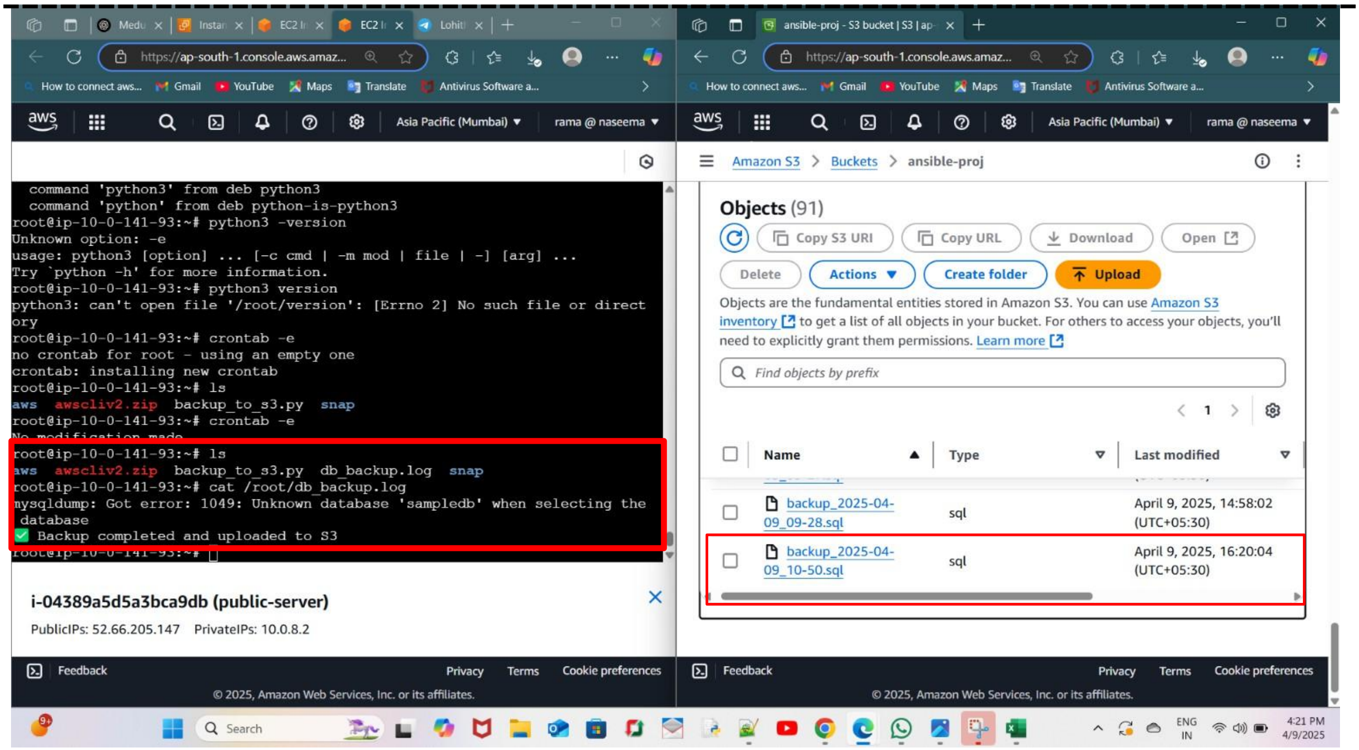
- For internet testing perpose I use ansible tool to install the apache2 web server in private server and I can access the apache2 application through the Load balancer
- For distributing the traffic from public server to private server.

- So finally I implement route53 to access the web application through global. wide.



Hello from Ansible

Please access this DNS onces----> http://ansible.ramcloud.shop (Note: check once access through http)



```
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
root@ip-10-0-141-93:~# python3 -version
Unknown option: -e
usage: python3 [option] ... [-c cmd | -m mod | file | -] [arg] ...
Try `python -h' for more information.
root@ip-10-0-141-93:~# python3 version
python3: can't open file '/root/version': [Errno 2] No such file or direct
ory
root@ip-10-0-141-93:~# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
root@ip-10-0-141-93:~# ls
aws  awscliv2.zip  backup_to_s3.py  snap
root@ip-10-0-141-93:~# crontab -e
No modification made
root@ip-10-0-141-93:~# ls
aws  awscliv2.zip  backup_to_s3.py  db_backup.log  snap
root@ip-10-0-141-93:~# cat /root/db_backup.log
mysqldump: Got error: 1049: Unknown database 'sampledb' when selecting the
 database
✅ Backup completed and uploaded to S3
root@ip-10-0-141-93:~# 
```

i-04389a5d5a3bca9db (public-server)

PublicIPs: 52.66.205.147 PrivateIPs: 10.0.8.2

Amazon S3 > Buckets > ansible-proj

Objects (91)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified
backup_2025-04-09_09-28.sql	sql	April 9, 2025, 14:58:02 (UTC+05:30)
backup_2025-04-09_10-50.sql	sql	April 9, 2025, 16:20:04 (UTC+05:30)

- so finally, the MySQL dB backup and server logs are sent into the s3 bucket using python code and crontab expression.

Benefits:

- Secure Private Networking
    - Keeps sensitive services (like the database) isolated from the internet.
- Centralized Internet Control
    - All traffic from the private subnet goes through the public proxy—enabling filtering, logging, and monitoring.
- Automated Patch Management
    - Ensures systems are always up-to-date with the latest security updates.
- Scheduled DB Backups
    - Regular and automatic backups prevent data loss.
- Data Durability with S3
    - Amazon S3 provides a secure, reliable backup storage solution.
- Limited External Exposure
    - Only the proxy server has internet access, reducing attack surfaces.

Advantages:

- Improved Security Posture
    - Better control over traffic and access through subnet isolation and IAM policies.
- Efficient Resource Management

- Public EC2 acts as a hub for managing updates and internet traffic.

- Reliability and Continuity
  - Even if an instance is lost, backups on S3 can be used to restore data.
- Scalability
  - Easily extendable architecture—more private instances can route traffic through the same proxy.
- Customizability
  - Tools like Ansible allow easy customization for updates and monitoring.
- Cost Efficiency