

Day 08 – Cloud Server Setup: Docker, Nginx & Web Deployment

Objective:

Deploy a real web server on a cloud VM and practice essential DevOps tasks such as:

Cloud provisioning, SSH access, Installing and managing services,

Configuring security rules, Extracting and backing up logs.

Cloud Environment:

Cloud Provider: AWS EC2

OS: Ubuntu

Instance Type: t2.micro

Region: ap-southeast-2

Web Server: Nginx

Container Engine: Docker

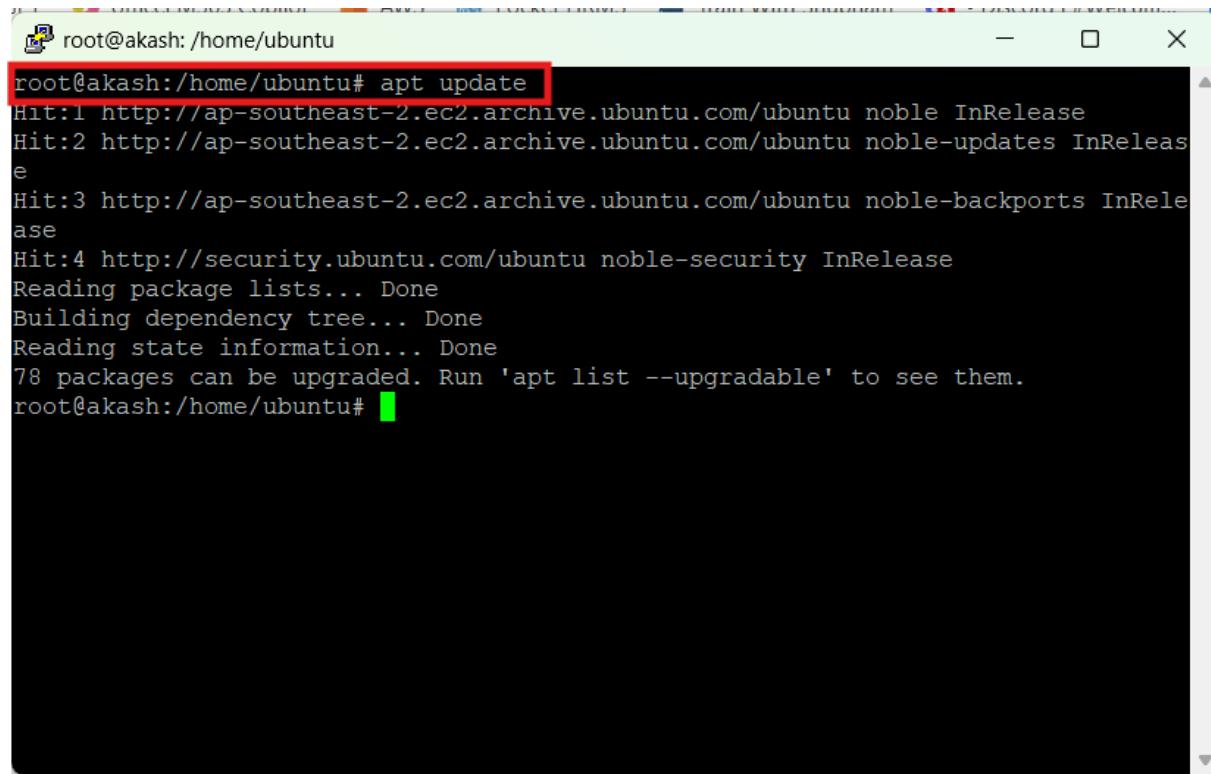
Lab Tasks Performed:

1- Launch EC2 Instance & SSH Access

Created EC2 instance and connected using SSH.

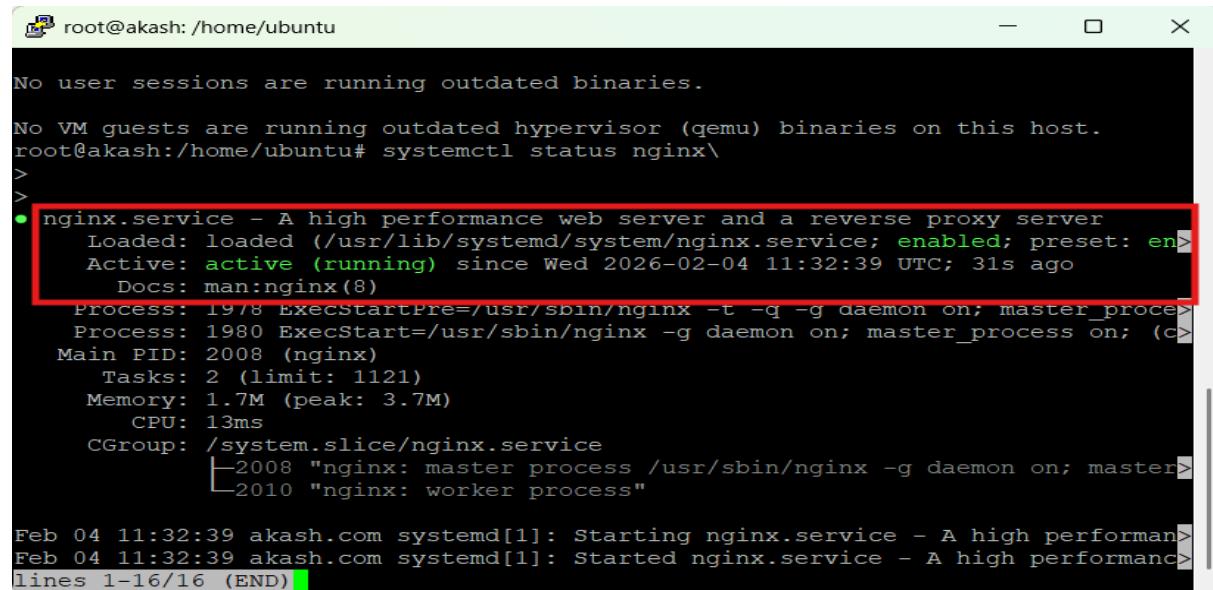
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, AWS Global View, Events, Instances (selected), Images, and Elastic Block Store. The main area displays a table of instances. A red box highlights the second row, which contains the instance 'Day8Practice-machine2'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, View alarms, Availability Zone, Public IPv4 DNS, and a Actions dropdown. Below the table, a detailed view for 'Day8Practice-machine2' is shown. It includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under Details, there's an 'Instance summary' section with fields for Instance ID (i-0a296df699769cd00), Public IPv4 address (13.211.64.3), Private IP4 addresses (172.31.4.9), Instance state (Running), and Public DNS (ec2-13-211-64-3.ap-southeast-2.compute.amazonaws.com).

2- connected using SSH and system update



```
root@akash:/home/ubuntu# apt update
Hit:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
78 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@akash:/home/ubuntu#
```

3-Install Nginx



```
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@akash:/home/ubuntu# systemctl status nginx\
>
>
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en>
  Active: active (running) since Wed 2026-02-04 11:32:39 UTC; 31s ago
    Docs: man:nginx(8)
   Process: 1978 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce>
   Process: 1980 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
 Main PID: 2008 (nginx)
   Tasks: 2 (limit: 1121)
  Memory: 1.7M (peak: 3.7M)
    CPU: 13ms
   CGroup: /system.slice/nginx.service
           └─2008 "nginx: master process /usr/sbin/nginx -g daemon on; master>
             ├─2010 "nginx: worker process"

Feb 04 11:32:39 akash.com systemd[1]: Starting nginx.service - A high performan>
Feb 04 11:32:39 akash.com systemd[1]: Started nginx.service - A high performanc>
lines 1-16/16 (END)
```

```
#sudo apt update
#sudo apt install nginx -y
#sudo systemctl start nginx
#sudo systemctl enable nginx
#sudo systemctl status nginx
```

4-Install Docker

```
root@akash:/home/ubuntu
root@akash:/home/ubuntu# systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: e>
  Active: active (running) since Wed 2026-02-04 11:37:17 UTC; 18s ago
TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
      Main PID: 2522 (dockerd)
        Tasks: 8
       Memory: 94.2M (peak: 94.5M)
         CPU: 317ms
        CGroup: /system.slice/docker.service
                  └─2522 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Feb 04 11:37:16 akash.com dockerd[2522]: time="2026-02-04T11:37:16.844616333Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.219020977Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.257172257Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.257443287Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.276238295Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.276368045Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.308437780Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.318676736Z" >
Feb 04 11:37:17 akash.com dockerd[2522]: time="2026-02-04T11:37:17.318773888Z" >
Feb 04 11:37:17 akash.com systemd[1]: Started docker.service - Docker Application >
lines 1-22/22 (END)
```

```
#sudo apt install docker.io -y
#sudo systemctl start docker
#sudo systemctl enable docker
#sudo systemctl status docker
```

5-Configure SecurityGroup

Inbound rule added:

HTTP – Port 80
HTTPS– Port 443
SSH – Port 22

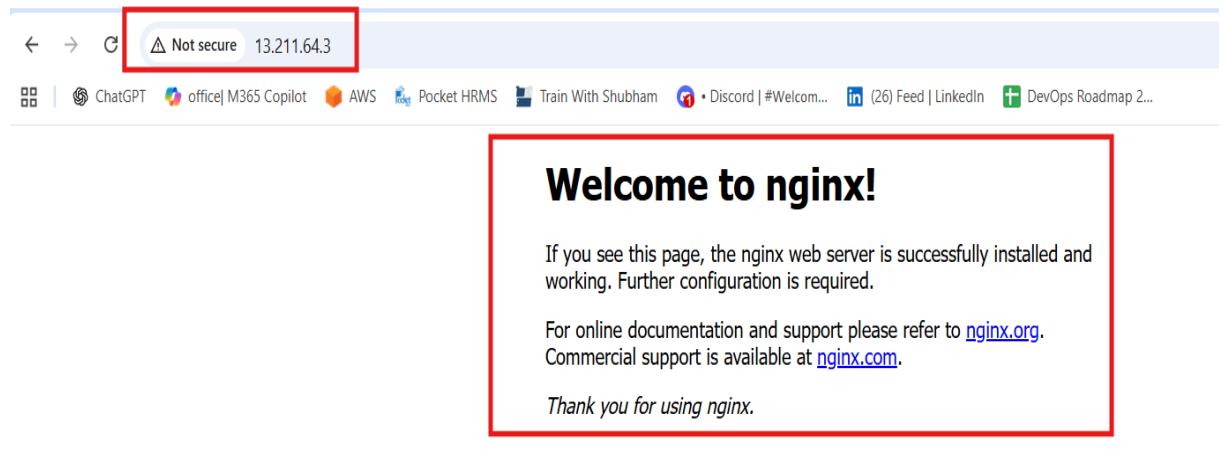
The screenshot shows the AWS EC2 Security Groups console. On the left, there's a navigation sidebar with options like Capacity Manager, Images, Elastic Block Store, Network & Security, and Load Balancing. The main area displays the details for a security group named "sg-022c3504b483ec05d - launch-wizard-1". The "Details" section includes fields for Security group name (set to "launch-wizard-1"), Security group ID (sg-022c3504b483ec05d), Description (set to "launch-wizard-1 created 2026-01-25T04:48:04.846Z"), and VPC ID (vpc-09522ba8f08214f9). The "Inbound rules" tab is selected, showing three rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0ae1bb58db06afc74	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-0d1a54954e4eb01e1	IPv4	HTTPS	TCP	443	0.0.0.0/0
-	sgr-0cb34dcad89fa1a4	IPv4	HTTP	TCP	80	0.0.0.0/0

Verify Web Access

Accessed Nginx using public IP:

<http://13.211.64.3>



Extract Nginx Logs

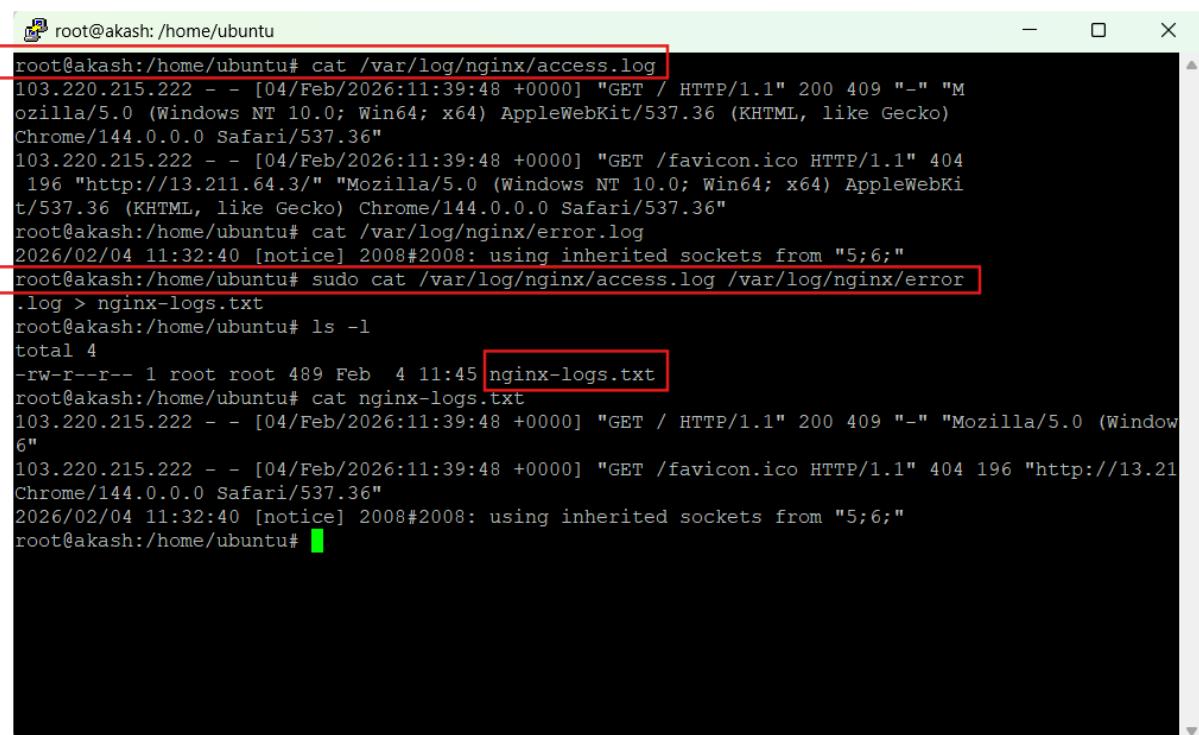
```
#cat /var/log/nginx/access.log
```

```
#cat /var/log/nginx/error.log
```

```
root@akash:/home/ubuntu
root@akash:/home/ubuntu# cat /var/log/nginx/access.log
103.220.215.222 - - [04/Feb/2026:11:39:48 +0000] "GET / HTTP/1.1" 200 409 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36"
103.220.215.222 - - [04/Feb/2026:11:39:48 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://13.211.64.3/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36"
root@akash:/home/ubuntu# cat /var/log/nginx/error.log
2026/02/04 11:32:40 [notice] 2008#2008: using innerited sockets from "5;6;"
```

Save Logs to File

```
#sudo cat /var/log/nginx/access.log /var/log/nginx/error.log > nginx-logs.txt  
#ls -l  
#Scat nginx-logs.txt
```



A terminal window titled 'root@akash:/home/ubuntu' showing the command to merge access and error logs into a single file, followed by a file listing.

```
root@akash:/home/ubuntu# cat /var/log/nginx/access.log  
103.220.215.222 - - [04/Feb/2026:11:39:48 +0000] "GET / HTTP/1.1" 200 409 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36"  
103.220.215.222 - - [04/Feb/2026:11:39:48 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://13.211.64.3/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36"  
root@akash:/home/ubuntu# cat /var/log/nginx/error.log  
2026/02/04 11:32:40 [notice] 2008#2008: using inherited sockets from "5;6;"  
root@akash:/home/ubuntu# sudo cat /var/log/nginx/access.log /var/log/nginx/error.log > nginx-logs.txt  
root@akash:/home/ubuntu# ls -l  
total 4  
-rw-r--r-- 1 root root 489 Feb 4 11:45 nginx-logs.txt  
root@akash:/home/ubuntu# cat nginx-logs.txt  
103.220.215.222 - - [04/Feb/2026:11:39:48 +0000] "GET / HTTP/1.1" 200 409 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36"  
103.220.215.222 - - [04/Feb/2026:11:39:48 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://13.211.64.3/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36"  
2026/02/04 11:32:40 [notice] 2008#2008: using inherited sockets from "5;6;"
```

Commands Used (Summary):

```
sudo apt update
sudo apt install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl status nginx
sudo apt install docker.io -y
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
cat /var/log/nginx/access.log
cat /var/log/nginx/error.log
sudo cat /var/log/nginx/access.log /var/log/nginx/error.log > nginx-logs.txt
```

Challenges Faced:

Website not accessible – Allowed port 80 in AWS Security Group

Permission denied on logs – Used sudo

Service persistence – Enabled services using systemctl enable