

EC2 Instance – System Information Commands

When you launch an **EC2 instance (Linux)**, you often need to check **system information** like OS details, CPU, memory, disk, network, uptime, etc.

These commands help in **monitoring, troubleshooting, and system analysis**.

Check Operating System Information

Command:

```
cat /etc/os-release
```

Purpose:

- Shows **Linux distribution name**
- Version, ID, and description

Example Output:

```
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
```

Check Kernel Version

Command:

```
uname -r
```

Purpose:

- Displays **Linux kernel version**

Check Full System Information

Command:

```
uname -a
```

Shows:

- Kernel name
- Kernel version
- Architecture
- Hostname

Check CPU Information

Command:

```
lscpu
```

Purpose:

- CPU architecture
- Number of CPUs
- Core and thread details

Alternative CPU Command:

```
cat /proc/cpuinfo
```

- Detailed CPU information
- Useful for advanced analysis

Check Memory (RAM) Usage

Command:

```
free -h
```

Purpose:

- Shows **total, used, and free RAM**
- **-h** means **human readable**

Example:

```
Mem: 1.0G 300M 700M
```

Check Disk Usage

Command:

```
df -h
```

Purpose:

- Shows disk space usage of mounted file systems
- **-h** makes size readable (GB, MB)

Check Disk Details (Block Devices)

Command:

```
lsblk
```

Purpose:

- Displays attached disks and partitions
- Useful for EBS volume checks

Check System Uptime

Command:

```
uptime
```

Shows:

- How long the EC2 instance has been running
- Load average

Check Hostname

Command:

```
hostname
```

Purpose:

- Displays EC2 instance hostname

Check IP Address Information

Command:

```
ip a
```

Purpose:

- Shows **private IP**
- Network interfaces

Check EC2 Instance Metadata (Important)

Command:

```
curl http://169.254.169.254/latest/meta-data/
```

Purpose:

- EC2-specific system info
- Instance ID, public IP, AMI ID, etc.

Example:

```
curl http://169.254.169.254/latest/meta-data/instance-id
```

Check Logged-in Users

Command:

```
who
```

or

```
w
```

Check Running Processes

Command:

```
top
```

Purpose:

- Real-time CPU and memory usage
- Running processes

Check System Architecture

Command:

```
arch
```

Output:

- x86_64 or arm64 (important for EC2 instance type)

Check AWS EC2 Instance Type

Command:

```
curl http://169.254.169.254/latest/meta-data/instance-type
```

What is NGINX?

NGINX (Engine-X) is a **web server** used to:

- Host websites
- Serve HTML, CSS, JavaScript files
- Handle many users at the same time
- Act as a **reverse proxy & load balancer**

Simple Meaning:

NGINX takes a request from a browser and sends back a web page.

Why Use NGINX on EC2?

- Fast and lightweight
- Easy to install on Ubuntu
- Very popular in **AWS Cloud**
- Used in real production servers

How to Check if NGINX is Installed on EC2

```
nginx -v
```

Result:

- Version shown → NGINX installed
- command not found → Not installed

How to Check NGINX Service Status

```
sudo systemctl status nginx
```

- **active (running)** → working
- **inactive / not found** → not installed or stopped

How to Login as Root User (Ubuntu EC2)

Direct root login is disabled for security.

Switch to root:

```
sudo su -
```

OR

```
sudo -i
```

Verify:

```
whoami
```

Output:

```
root
```

Update Ubuntu System (Important Step)

```
sudo apt update -y
```

Install NGINX on Ubuntu EC2

```
sudo apt install nginx -y
```

Start & Enable NGINX

```
sudo systemctl start nginx
sudo systemctl enable nginx
```

Check NGINX Status

```
sudo systemctl status nginx
```

You should see:

```
active (running)
```

Allow HTTP in EC2 Security Group

Open AWS EC2 → Security Group → Inbound Rules

Add rule:

Type	Protocol	Port	Source
HTTP	TCP	80	0.0.0.0/0

Test Default NGINX Page

Open browser:

```
http://<EC2-Public-IP>
```

You should see:

Welcome to nginx!

Host a Simple HTML Website on NGINX

Step 1: Go to Web Root Directory

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

Step 2: Create an HTML File

```
sudo nano index.html
```

Step 3: Add Simple HTML Code

```
<!DOCTYPE html>
<html>
<head>
    <title>My First EC2 Website</title>
</head>
<body>
    <h1>Welcome to My NGINX Server</h1>
    <p>This website is hosted on AWS EC2 using NGINX.</p>
</body>
</html>
```

Save & exit:

- CTRL + O → Enter
- CTRL + X

Step 4: Restart NGINX

```
sudo systemctl restart nginx
```

Access Your Website

Open browser:

```
http://<EC2-Public-IP>
```

Your HTML website is live!

Important NGINX Directories (Ubuntu)

Purpose	Path
Website Files	/var/www/html/
Main Config	/etc/nginx/nginx.conf
Site Config	/etc/nginx/sites-available/
Enabled Sites	/etc/nginx/sites-enabled/
Logs	/var/log/nginx/

Useful NGINX Commands

```
sudo systemctl stop nginx
sudo systemctl restart nginx
sudo systemctl reload nginx
```

Exam-Friendly Summary

- NGINX is a fast web server
- Installed using apt on Ubuntu
- Runs on **port 80**
- Website files stored in /var/www/html
- EC2 Security Group must allow HTTP

Quick Command Table

Task	Command
Check NGINX	nginx -v
Install	sudo apt install nginx -y
Start	sudo systemctl start nginx
Root Login	sudo su -
Create Website	nano index.html
Restart	sudo systemctl restart nginx