Virtualization

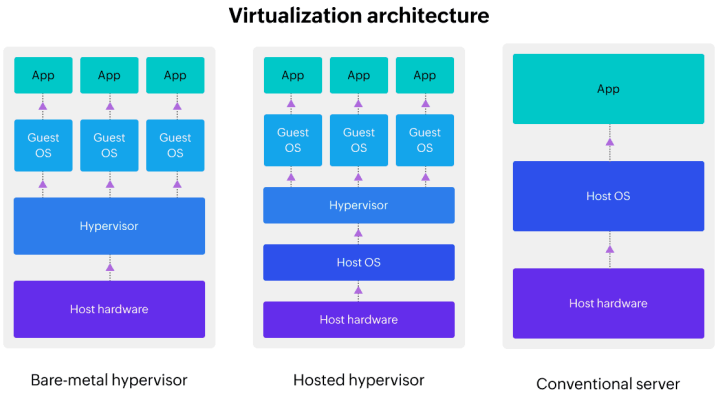
- Virtualization is a technology that allows you to create a virtual or simulated version of a computer, server, operating system, or application.
- Instead of relying on a physical machine, virtualization enables the creation of multiple virtual instances on the same physical hardware.
- It helps to optimize hardware usage and simplify management.

There are several reasons for the widespread adoption of virtualization,

Why Virtualization?

- Resource Optimization
- Cost Savings
- Isolation
- Flexibility and Scalability
- Energy Efficiency

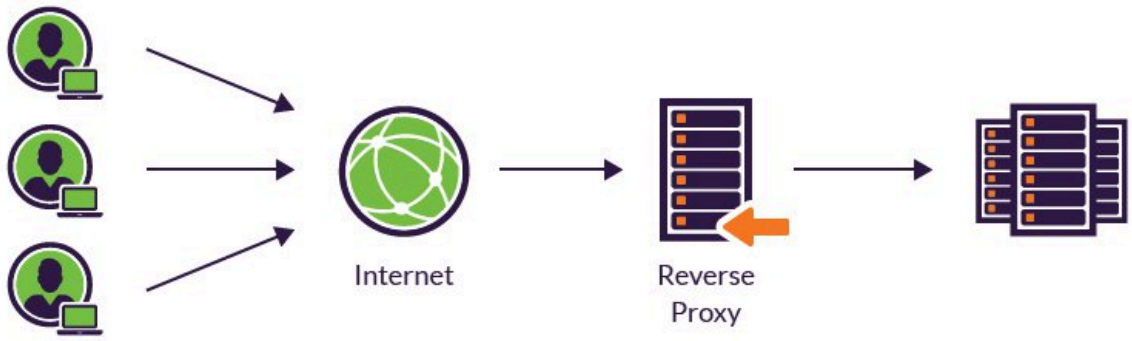
Types of Virtualization:



Amazon EC2:

Nginx:

Reverse proxy with caching: Nginx can act as a reverse proxy server, handling requests on behalf of other servers and caching the results for faster response times



Nginx has built-in support for IPv6, allowing it to serve websites with the latest internet protocol.

Web Server: Nginx can be used as a standalone web server to serve static content (HTML, CSS, images, etc.) and dynamic content (using FastCGI, SCGI, or other backends). It is known for its high performance and low resource usage, making it suitable for handling a large number of concurrent connections.

Reverse Proxy: Nginx is often used as a reverse proxy server to distribute incoming web traffic across multiple backend servers. This helps distribute the load, improves performance, and provides

redundancy. It also allows for easy scaling by adding or removing backend servers without affecting clients.

Load Balancer: Nginx can act as a load balancer, distributing incoming traffic across multiple backend servers to ensure even load distribution and prevent any single server from being overwhelmed. This is particularly useful for applications with high traffic or resource-intensive tasks.

SSL/TLS Termination: Nginx can handle SSL/TLS termination, offloading the SSL/TLS encryption and decryption process from backend servers. This helps improve the overall performance of web applications.

Caching: Nginx can be configured to serve as a caching proxy, storing static content in memory to reduce the load on backend servers and improve response times for frequently requested content. This is especially beneficial for websites with a large number of visitors.

Application Firewall: Nginx can be used as a reverse proxy with additional modules to act as a web application firewall (WAF), protecting web applications from common security threats such as SQL injection, cross-site scripting (XSS), and more.

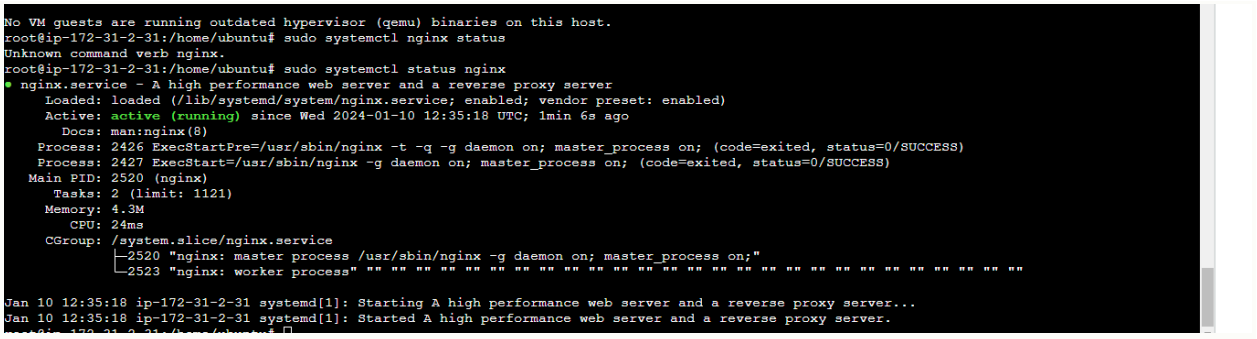
Media Streaming: Nginx can be employed as a media streaming server for delivering audio and video content efficiently. It supports various streaming protocols, including HTTP Live Streaming (HLS) and Dynamic Adaptive Streaming over HTTP (DASH).

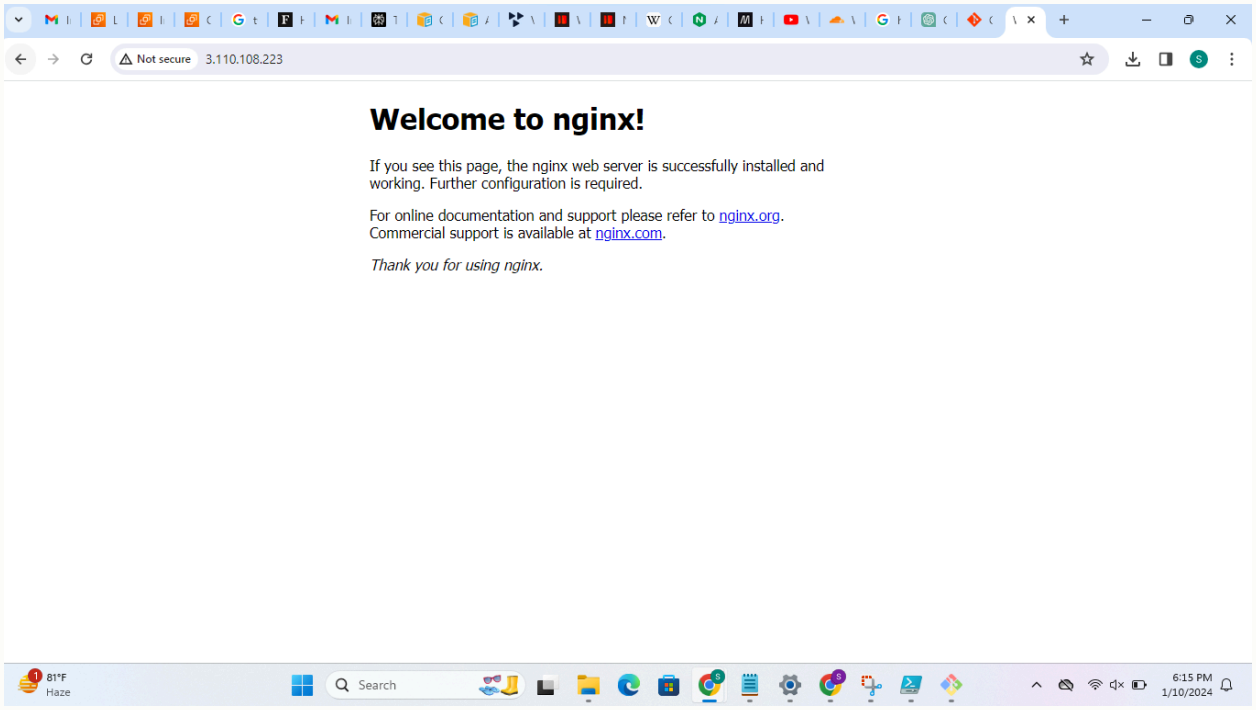
Container Orchestration: In containerized environments, Nginx is often used as an ingress controller for container orchestration platforms like Kubernetes. It helps manage incoming traffic to containerized applications and provides SSL termination, load balancing, and routing.

API Gateway: Nginx can serve as an API gateway, managing and securing access to backend APIs, handling authentication, and rate limiting requests. This is particularly useful in microservices architectures.

Static File Server: Nginx is excellent at serving static files efficiently. It can be used to serve static assets for web applications or act as a file server for distributing files.

In summary, Nginx is a versatile server that can be used in various scenarios to improve performance, scalability, and security for web applications and services.



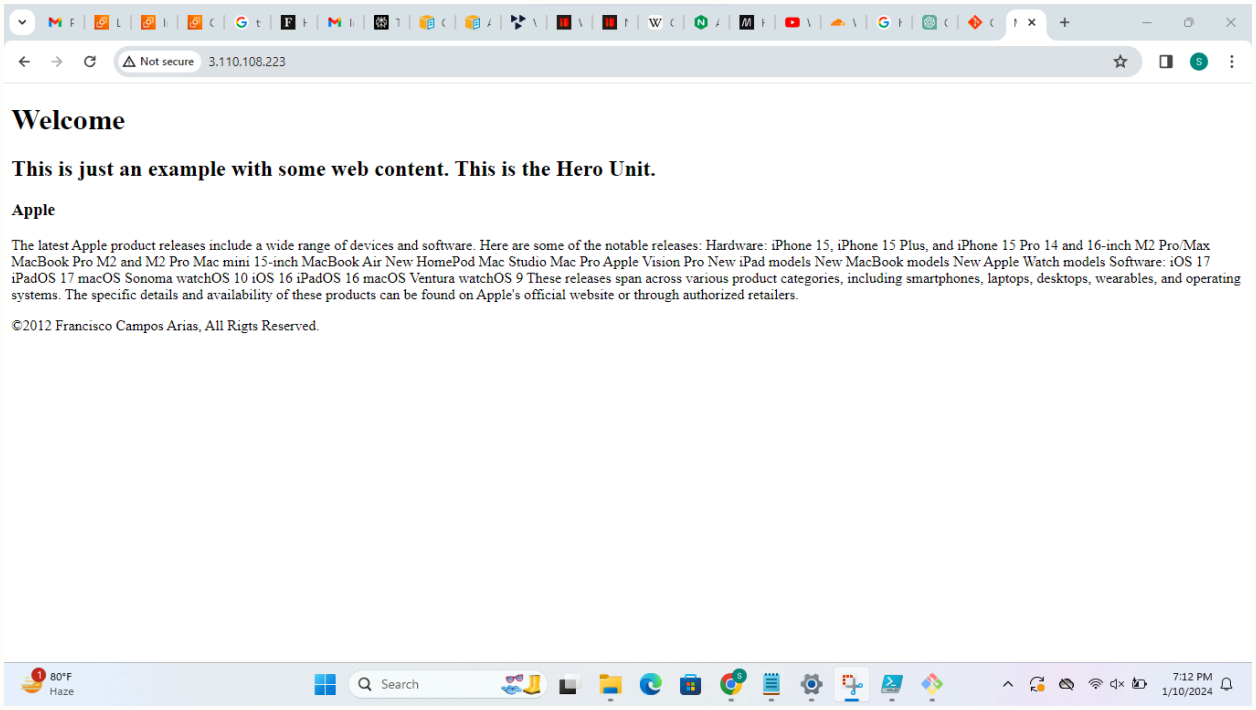


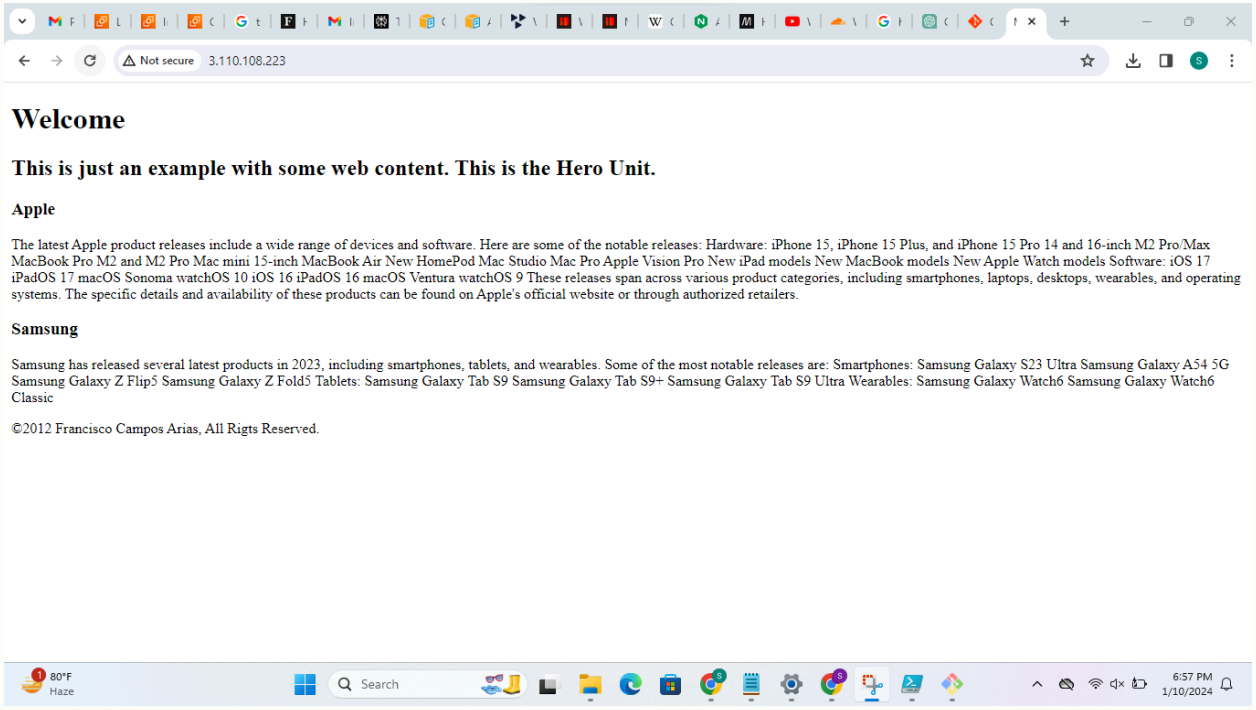
Nginx default file path : /usr/share/nginx/html

Root - /var/www/html

Check Nginx configuration: Verify that Nginx is configured to serve the index.html file from the correct location. **The Nginx configuration file, typically located at /etc/nginx/nginx.conf, or the site-specific configuration file in /etc/nginx/sites-available/, should specify the root directory where Nginx looks for files to serve.**

```
root@ip-172-31-2-31:/etc/nginx/sites-available# ls
default
root@ip-172-31-2-31:/etc/nginx/sites-available#
```


Not secure
3.110.108.223
Welcome
This is just an example with some web content. This is the Hero Unit.
Apple
The latest Apple product releases include a wide range of devices and software. Here are some of the notable releases: Hardware: iPhone 15, iPhone 15 Plus, and iPhone 15 Pro 14 and 16-inch M2 Pro/Max MacBook Pro M2 and M2 Pro Mac mini 15-inch MacBook Air New HomePod Mac Studio Mac Pro Apple Vision Pro New iPad models New MacBook models New Apple Watch models Software: iOS 17 iPadOS 17 macOS Sonoma watchOS 10 iOS 16 iPadOS 16 macOS Ventura watchOS 9 These releases span across various product categories, including smartphones, laptops, desktops, wearables, and operating systems. The specific details and availability of these products can be found on Apple's official website or through authorized retailers.
©2012 Francisco Campos Arias, All Rigts Reserved.
80°F
Haze
Search
7:12 PM
1/10/2024


Not secure
3.110.108.223
Welcome
This is just an example with some web content. This is the Hero Unit.
Apple
The latest Apple product releases include a wide range of devices and software. Here are some of the notable releases: Hardware: iPhone 15, iPhone 15 Plus, and iPhone 15 Pro 14 and 16-inch M2 Pro/Max MacBook Pro M2 and M2 Pro Mac mini 15-inch MacBook Air New HomePod Mac Studio Mac Pro Apple Vision Pro New iPad models New MacBook models New Apple Watch models Software: iOS 17 iPadOS 17 macOS Sonoma watchOS 10 iOS 16 iPadOS 16 macOS Ventura watchOS 9 These releases span across various product categories, including smartphones, laptops, desktops, wearables, and operating systems. The specific details and availability of these products can be found on Apple's official website or through authorized retailers.
Samsung
Samsung has released several latest products in 2023, including smartphones, tablets, and wearables. Some of the most notable releases are: Smartphones: Samsung Galaxy S23 Ultra Samsung Galaxy A54 5G Samsung Galaxy Z Flip5 Samsung Galaxy Z Fold5 Tablets: Samsung Galaxy Tab S9 Samsung Galaxy Tab S9+ Samsung Galaxy Tab S9 Ultra Wearables: Samsung Galaxy Watch6 Samsung Galaxy Watch6 Classic
©2012 Francisco Campos Arias, All Rigts Reserved.
80°F
Haze
Search
6:57 PM
1/10/2024

Apache:

Commands Used :

```
apt-get update
sudo apt install apache2
systemctl status apache2
systemctl enable apache2
```

Apache HTTP Server, commonly referred to as Apache, is another widely used and versatile web server. Similar to Nginx, Apache is a robust and feature-rich server that can be employed in various scenarios. Here are some common use cases for Apache HTTP Server (often referred to as Apache2 in its package names on Linux distributions):

1. **Web Server**: Apache can serve as a standalone web server to deliver static content, dynamic content using server-side scripting languages (like PHP, Python, or Perl), and more. It supports a wide range of modules and configurations for web hosting.

2. **Content Management Systems (CMS)**: Many popular content management systems, such as WordPress and Drupal, can be hosted on Apache. Apache's compatibility with PHP makes it a common choice for PHP-based applications.

3. **Dynamic Content Generation**: Apache supports various server-side scripting languages, making it suitable for generating dynamic content. This includes executing scripts written in PHP, Python, Perl, and other languages.

4. **SSL/TLS Support**: Apache has robust support for SSL/TLS encryption, making it suitable for serving secure websites. It can handle SSL/TLS termination and provide secure communication between clients and servers.

5. **Reverse Proxy**: Apache can function as a reverse proxy, distributing incoming requests to multiple backend servers. This helps distribute the load, improve performance, and add redundancy to the infrastructure.

6. **Load Balancer**: Apache can be configured as a load balancer to distribute incoming traffic across multiple backend servers, ensuring even load distribution and improving overall system reliability.

7. **Virtual Hosting**: Apache supports virtual hosting, allowing a single server to host multiple websites on the same IP address. This is useful for hosting multiple domains on a single server.

8. **Authentication and Authorization**: Apache provides robust authentication and authorization mechanisms. It supports various authentication methods, including basic authentication, digest authentication, and integration with external authentication systems.

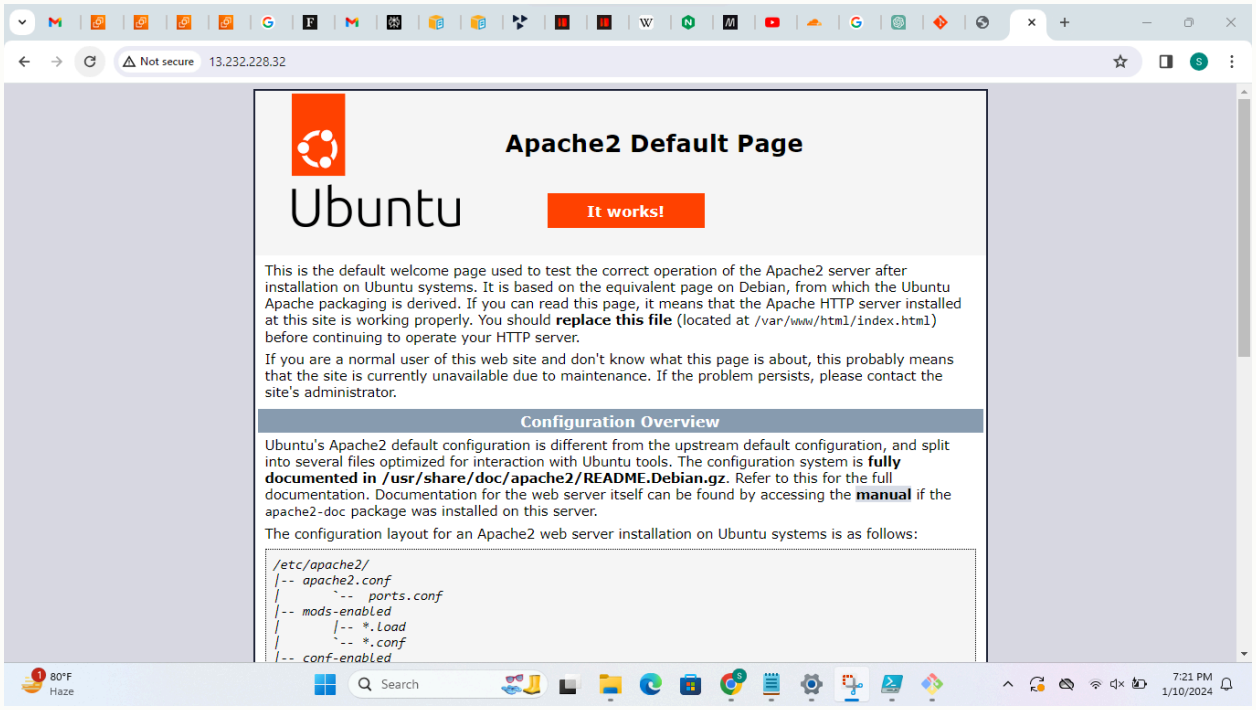
9. **Logging and Monitoring**: Apache offers extensive logging capabilities, allowing administrators to monitor and analyze web server activity. Access logs, error logs, and other log formats provide valuable insights into server performance and potential issues.

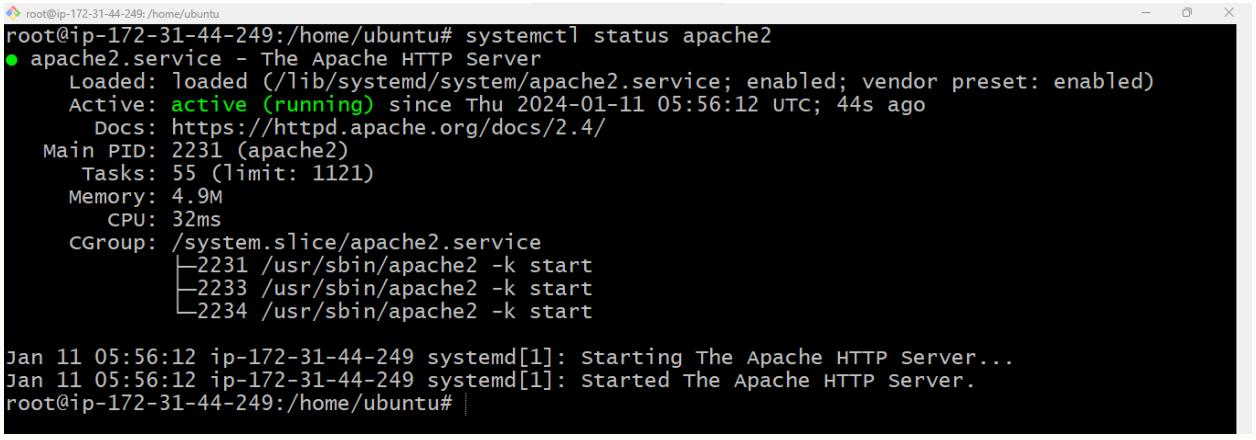
10. **.htaccess Support**: Apache supports the use of `.htaccess` files, allowing users to override server configuration settings at the directory

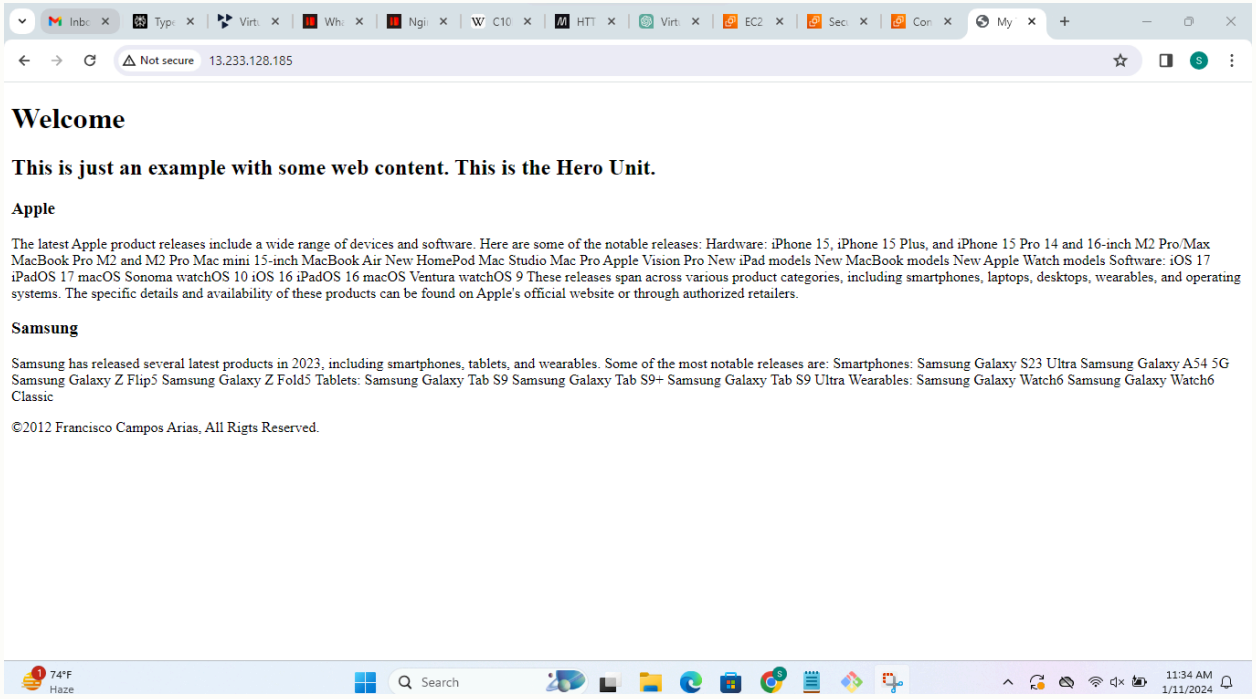
level. This is useful for website owners who may not have access to the main server configuration.

11. **Modular Architecture**: Apache's modular architecture allows users to enable or disable modules based on specific requirements. This flexibility makes it easy to tailor the server to the needs of different applications.

In summary, Apache is a versatile and widely used web server with a long history. It is well-suited for hosting a variety of web applications and services, offering flexibility, reliability, and extensive configuration options. The choice between Apache and Nginx often depends on specific use cases, preferences, and system requirements.


Not secure
13.232.228.32
Apache2 Default Page
Ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
80°F
Haze
Search
7:21 PM
1/10/2024


root@ip-172-31-44-249: /home/ubuntu
root@ip-172-31-44-249:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2024-01-11 05:56:12 UTC; 44s ago
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 2231 (apache2)
Tasks: 55 (limit: 1121)
Memory: 4.9M
CPU: 32ms
CGroup: /system.slice/apache2.service
├─2231 /usr/sbin/apache2 -k start
├─2233 /usr/sbin/apache2 -k start
└─2234 /usr/sbin/apache2 -k start
Jan 11 05:56:12 ip-172-31-44-249 systemd[1]: Starting The Apache HTTP Server...
Jan 11 05:56:12 ip-172-31-44-249 systemd[1]: Started The Apache HTTP Server.
root@ip-172-31-44-249:/home/ubuntu#



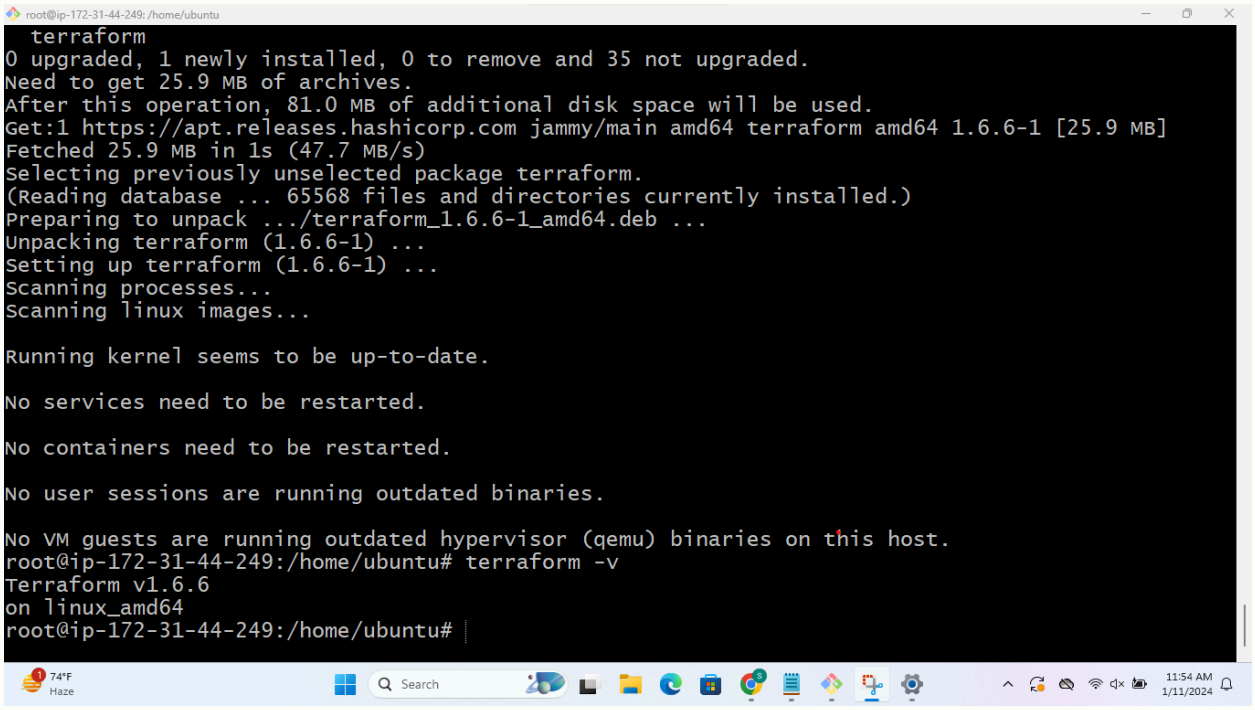
Terraform:

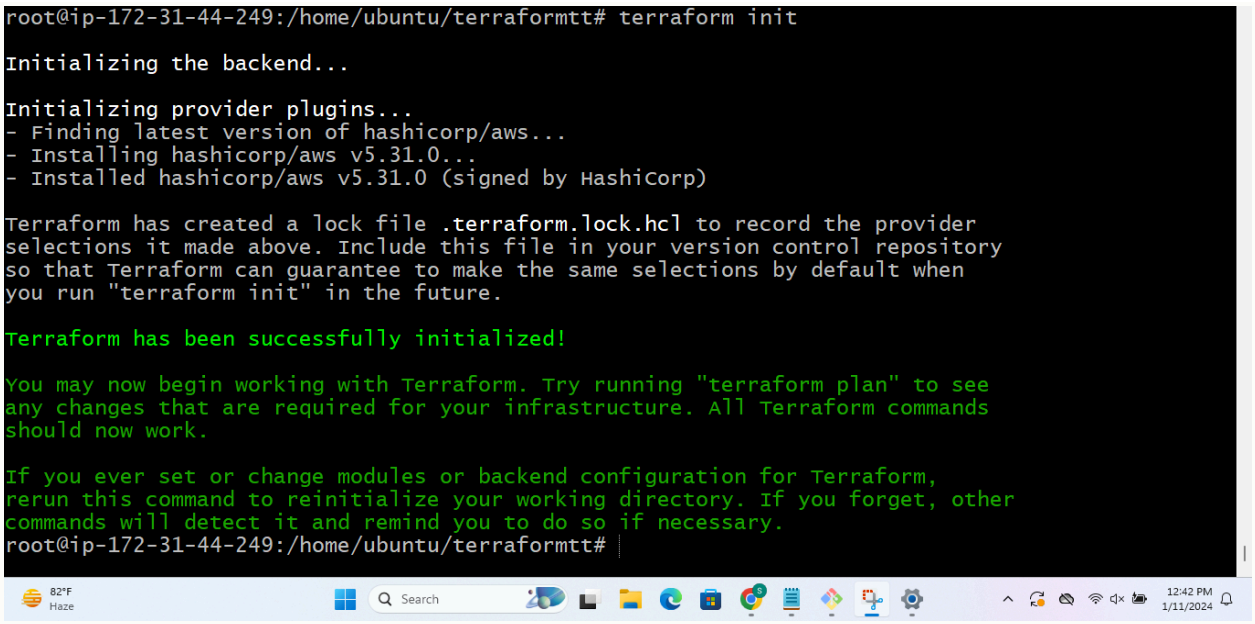
Terraform script used:

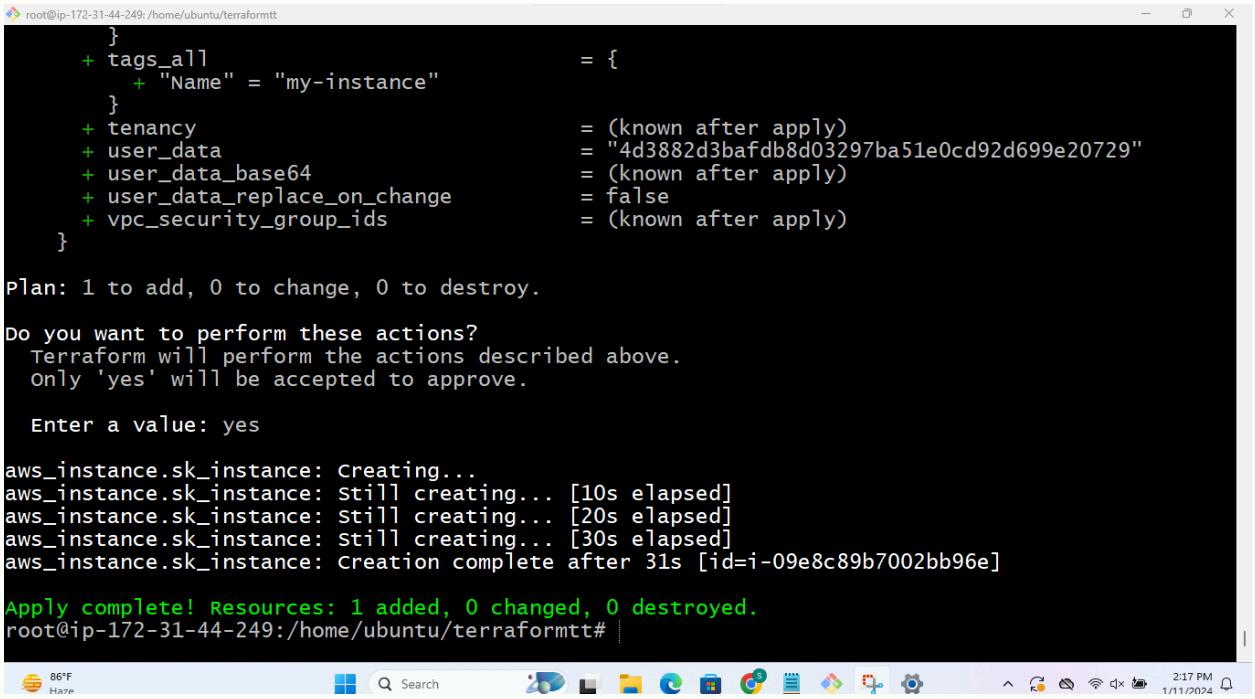
```
provider "aws" {
 region = "ap-south-1"  # Specify the desired AWS region
 shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_instance" "sk_instance" {
 ami           = "ami-03f4878755434977f"  # Specify the AMI ID for your desired Amazon Machine>
 instance_type = "t2.micro"  # Specify the instance type

 user_data = <<-EOF
  #!/bin/bash
  apt update
  apt install -y nginx
  cat <<HTML > /var/www/html/index.html
  <!DOCTYPE html>
```

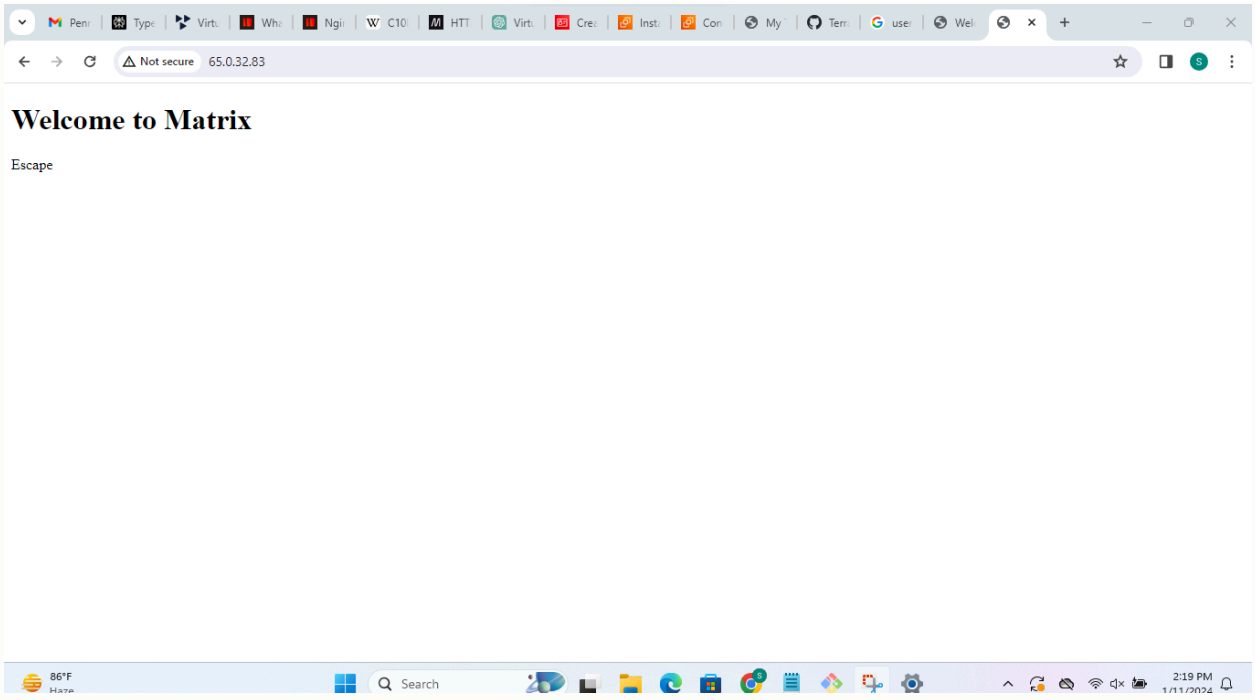
```
  <html>
  <head>
     <title>My Test Website</title>
  </head>
  <body>
     <h1>Welcome to Matrix</h1>
     <p>Escape</p>
  </body>
  </html>
  HTML
  systemctl start nginx
 EOF

 tags = {
  Name = "my-instance"
 }

 key_name = "sk"  # Specify your key pair name for SSH access
}
```

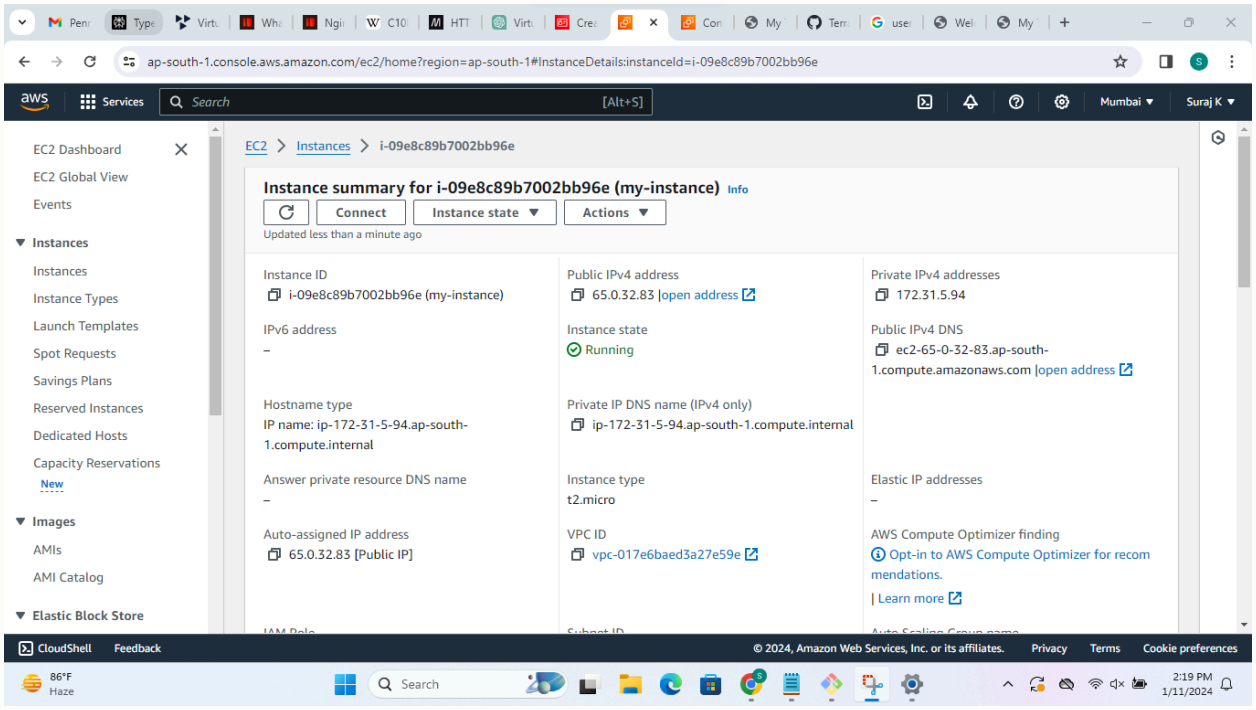
```
root@ip-172-31-44-249:/home/ubuntu# curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt
-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@ip-172-31-44-249:/home/ubuntu# sudo apt-add-repository "deb [arch=amd64] https://apt.releas
es.hashicorp.com $(lsb_release -cs) main"
Repository: 'deb [arch=amd64] https://apt.releases.hashicorp.com jammy main'
Description:
Archive for codename: jammy components: main
More info: https://apt.releases.hashicorp.com
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-jammy.l
ist
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicor
p_com-jammy.list
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 https://apt.releases.hashicorp.com jammy/main amd64 Packages [113 kB]
Fetched 236 kB in 1s (206 kB/s)
Reading package lists... Done
W: https://apt.releases.hashicorp.com/dists/jammy/InRelease: Key is stored in legacy trusted.gpg
 keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@ip-172-31-44-249:/home/ubuntu#
```

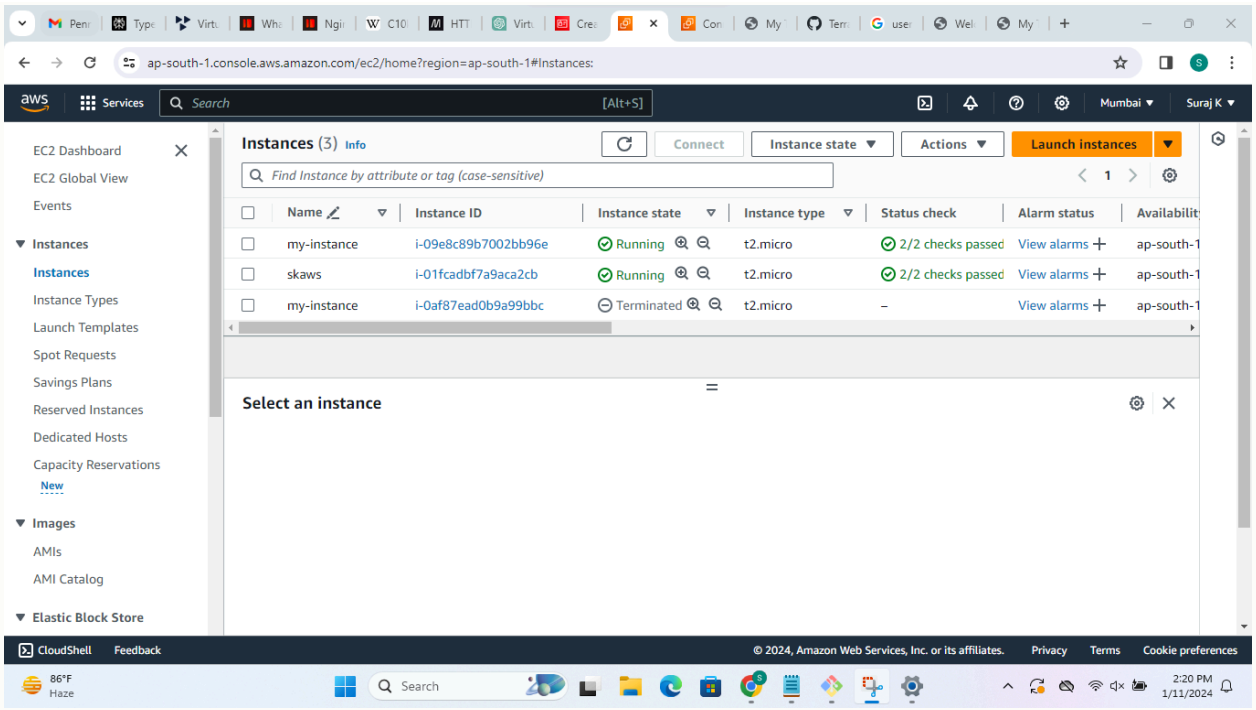

root@ip-172-31-44-249: /home/ubuntu
terraform
0 upgraded, 1 newly installed, 0 to remove and 35 not upgraded.
Need to get 25.9 MB of archives.
After this operation, 81.0 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.6.6-1 [25.9 MB]
Fetched 25.9 MB in 1s (47.7 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 65568 files and directories currently installed.)
Preparing to unpack .../terraform_1.6.6-1_amd64.deb ...
Unpacking terraform (1.6.6-1) ...
Setting up terraform (1.6.6-1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-44-249:/home/ubuntu# terraform -v
Terraform v1.6.6
on linux_amd64
root@ip-172-31-44-249:/home/ubuntu#
74°F
Haze
Search
11:54 AM
1/11/2024

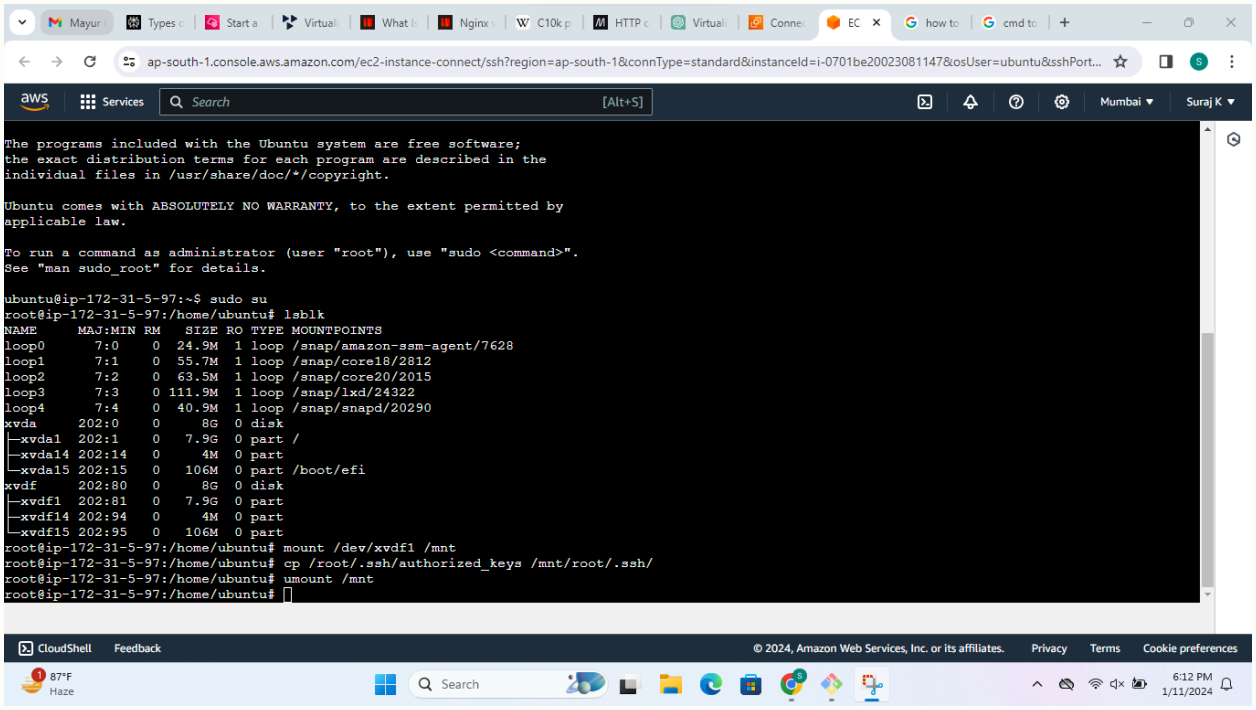

root@ip-172-31-44-249:/home/ubuntu/terraformtt# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-44-249:/home/ubuntu/terraformtt#
82°F
Haze
Search
12:42 PM
1/11/2024


root@ip-172-31-44-249: /home/ubuntu/terraformtt
}
+ tags_all = {
+ "Name" = "my-instance"
}
+ tenancy = (known after apply)
+ user_data = "4d3882d3bafdb8d03297ba51e0cd92d699e20729"
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}
Plan: 1 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
aws_instance.sk_instance: Creating...
aws_instance.sk_instance: Still creating... [10s elapsed]
aws_instance.sk_instance: Still creating... [20s elapsed]
aws_instance.sk_instance: Still creating... [30s elapsed]
aws_instance.sk_instance: Creation complete after 31s [id=i-09e8c89b7002bb96e]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-44-249:/home/ubuntu/terraformtt#
86°F
Haze
Search
2:17 PM
1/11/2024


Not secure
65.0.32.83
Welcome to Matrix
Escape
86°F
Haze
Search
2:19 PM
1/11/2024


ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#InstanceDetails:instanceId=i-09e8c89b7002bb96e
Services
Search
[Alt+S]
Mumbai
Suraj K
EC2 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
EC2 > Instances > i-09e8c89b7002bb96e
Instance summary for i-09e8c89b7002bb96e (my-instance) Info
Connect
Instance state
Actions
Updated less than a minute ago
Instance ID
i-09e8c89b7002bb96e (my-instance)
Public IPv4 address
65.0.32.83 |open address
Private IPv4 addresses
172.31.5.94
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-65-0-32-83.ap-south-1.compute.amazonaws.com |open address
Hostname type
IP name: ip-172-31-5-94.ap-south-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-5-94.ap-south-1.compute.internal
Answer private resource DNS name
–
Instance type
t2.micro
Elastic IP addresses
–
Auto-assigned IP address
65.0.32.83 [Public IP]
VPC ID
vpc-017e6baed3a27e59e
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
86°F
Haze
Search
2:19 PM
1/11/2024


ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:
Services
Search
[Alt+S]
Mumbai
Suraj K
EC2 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
Instances (3) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availabilit
my-instance
i-09e8c89b7002bb96e
Running
t2.micro
2/2 checks passed
View alarms
ap-south-1
skaws
i-01fcadbf7a9aca2cb
Running
t2.micro
2/2 checks passed
View alarms
ap-south-1
my-instance
i-0af87ead0b9a99bbc
Terminated
t2.micro
–
View alarms
ap-south-1
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
86°F
Haze
Search
2:20 PM
1/11/2024



https://stackoverflow.com/questions/52560096/how-to-access-ec2-instance-even-if-pem-file-is-lost

In Linux, there are several types of virtualization, each serving different purposes. Here are some common types:

1. **Full Virtualization (Hardware Virtualization):** Uses a hypervisor to create virtual machines that can run multiple operating systems on the same hardware. Examples include KVM (Kernel-based Virtual Machine) and Xen.

2. **Containerization:** Involves encapsulating applications and their dependencies in containers, such as Docker. Containers share the host OS kernel but operate in isolated user spaces, providing lightweight and efficient virtualization.

3. **Para-Virtualization:** Requires modifying the guest operating system to be aware of the virtualized environment, allowing for improved performance. Xen is an example of a para-virtualization hypervisor.

4. **Hardware-assisted Virtualization:** Takes advantage of specific hardware features (such as Intel VT or AMD-V) to enhance virtualization performance. This is commonly used in conjunction with full virtualization.

5. **Operating System-level Virtualization (OS-level Virtualization):** Enables multiple isolated user spaces (containers or virtual environments) on a single operating system instance. Examples include LXC (Linux Containers) and OpenVZ.

6. **Desktop Virtualization (VDI - Virtual Desktop Infrastructure):** Involves running multiple desktop environments on a centralized server, with users accessing them remotely. VirtualBox and VMware are examples of desktop virtualization software.

Each type of virtualization in Linux caters to different use cases and has its advantages and trade-offs. The choice depends on factors such as performance requirements, resource efficiency, and the level of isolation needed for the intended applications or workloads.