Experiment No. 1

Aim: To study and implement Infrastructure as a Service using AWS

Objective: To demonstrate the steps to create and run virtual machines inside Amazon EC2

Theory:

IAAS: Infrastructure as a service (IaaS) is a type of cloud computing service that offers essential compute, storage, and networking resources on demand, on a pay-as-you-go basis. IaaS is one of the four types of cloud services, along with software as a service (SaaS), platform as a service (PaaS), and serverless.

In an IaaS service model, a cloud provider hosts the infrastructure components that are traditionally present in an on-premises data center. This includes servers, storage and networking hardware, as well as the virtualization or hypervisor layer.

IaaS providers also supply a range of services to accompany those infrastructure components. IaaS customers access resources and services through a wide area network (WAN), such as the internet, and can use the cloud provider's services to install the remaining elements of an application stack. For example, the user can log in to the IaaS platform to create virtual machines (VMs); install operating systems in each VM; deploy middleware, such as databases; create storage buckets for workloads and backups; and install the enterprise workload into that VM. Customers can then use the provider's services to track costs, monitor performance, balance network traffic, troubleshoot application issues and manage disaster recovery.

AWS: AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions. You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments.
AWS is designed to allow application providers, ISVs, and vendors to quickly and securely host your applications – whether an existing application or a new SaaS-based application.

EC2: Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

Steps:

- We will start by creating an EC2 instance. Go to EC2 Console https://console.aws.amazon.com/ec2/v2/home and Click on Launch instance. Where you will find multiple AMIs, Scroll down and select Ubuntu Server

- Once you select the server, follow further steps, and launch. Make sure while launching download the PEM file as it will need in the future.

- Now go to the EC2 dashboard and look for a public IP address and Copy that as we will need for PUTTY configuration, Though you can use Public DNS as well.

- We are done with AWS configuration, now open PUTTY and follow a few steps:

- Add copied IP address in the Host Name field and name the Saved session and save it. Will keep port as it is i.e., 22 as we have only opened the 22 port as we want everything should go through the SSH tunnel; though you can open other ports as well there will be less security.

- Add SSH (PPK) key by browsing SSH -> Auth -> Private Key file for authentication -> Browse and again go to Session and Save it.
  Note: By default, AWS only provide a PEM file. We need to convert PEM to PPK. You can find multiple online tools for converting this

- Click on Open, and it will prompt a security alert just choose Yes, and you will able to see the command prompt.
  On the command prompt, just log in as Ubuntu, as we have added PPK so it won't ask for Password.

- Once you get logged into the server, you need to follow the below. o sudo apt update && sudo apt upgrade o sudo sed -i 's/^PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd_config – To set login credentials o sudo /etc/init.d/ssh restart o sudo passwd ubuntu – This will be your login password to ubuntu machine o sudo apt install xrdp xfce4 xfce4-goodies tightvncserver o echo xfce4-session> /home/ubuntu/.xsession o sudo cp /home/ubuntu/.xsession /etc/skel o sudo sed -i '0,/-1/s//ask-1/' /etc/xrdp/xrdp.ini o sudo service xrdp restart

- Note: In between, if this process asks for some kind of choice, then select the local version installed option.

- Now, we need to configure the tunnel on PUTTY to the route of the request.

- Open PUTTY again and go to the Connection – > SSH -> Tunnels ->

1. Add Source Port (This could be any in my case I have added 8888).
2. Add Destination Port – This will be your instance private IP address following with 3389.
3. Click on Add

- Once done, now click on the open button and you will see a Command prompt where login Id will be Ubuntu.
  Now you can check either we are listening on the same port or not using the following command: netstat -antp

- Now we are done with the setup.

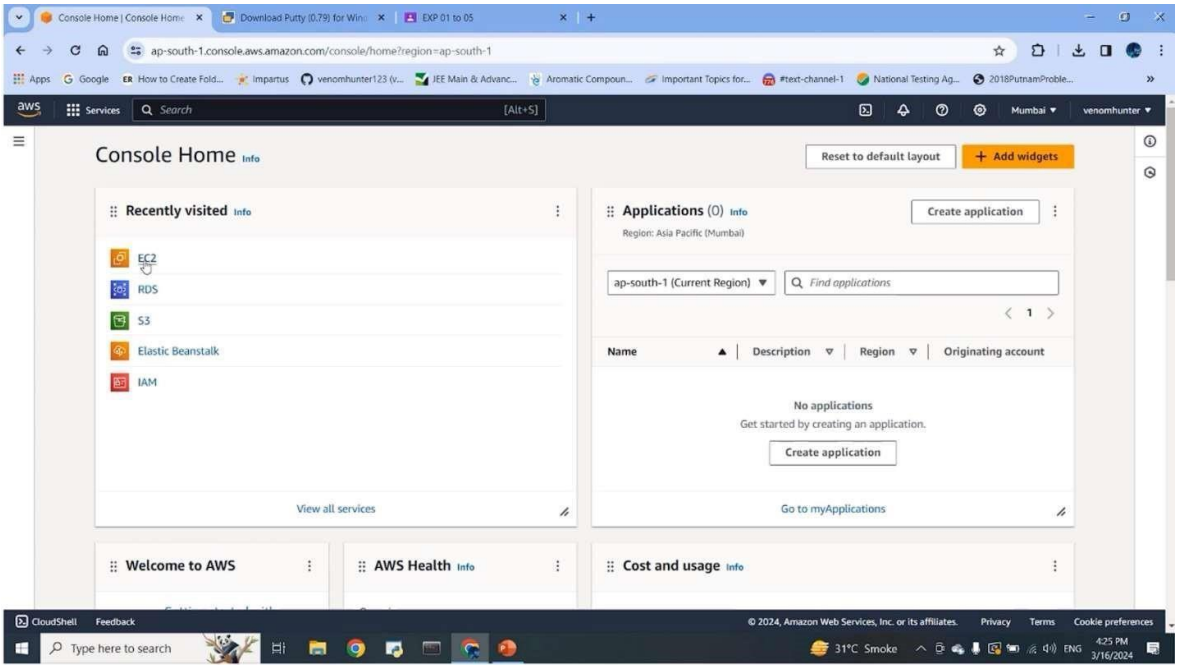
- Open Remote Desktop Connection

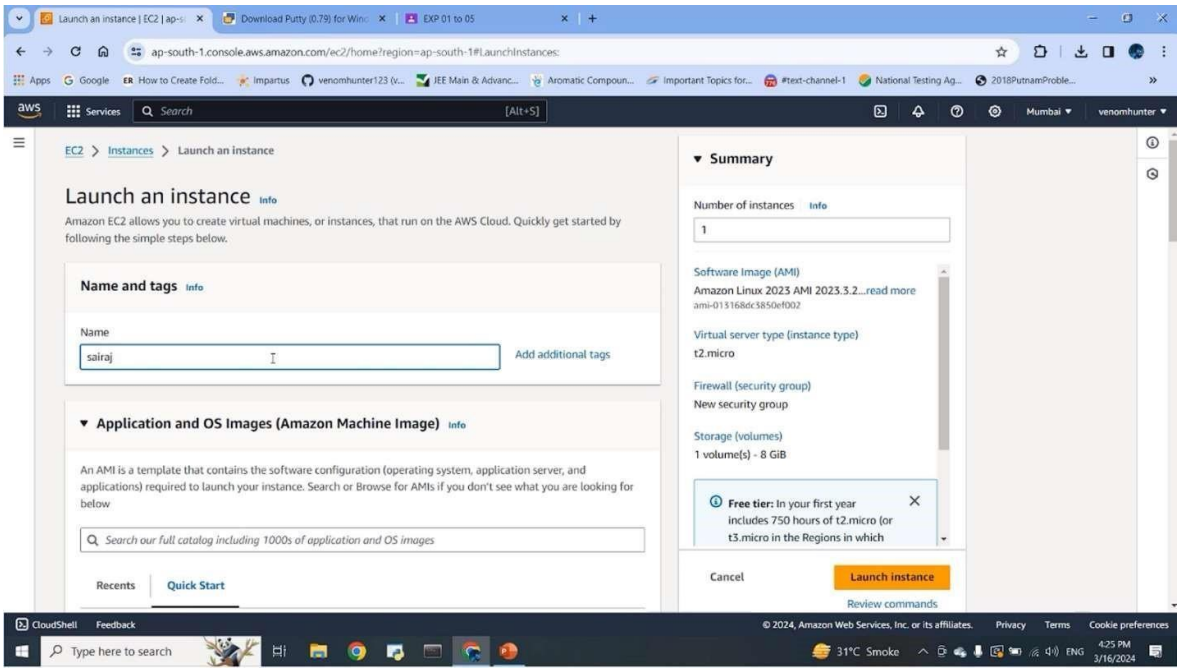
- Add localhost:8888 or 127.0.0.1:8888 and Connect

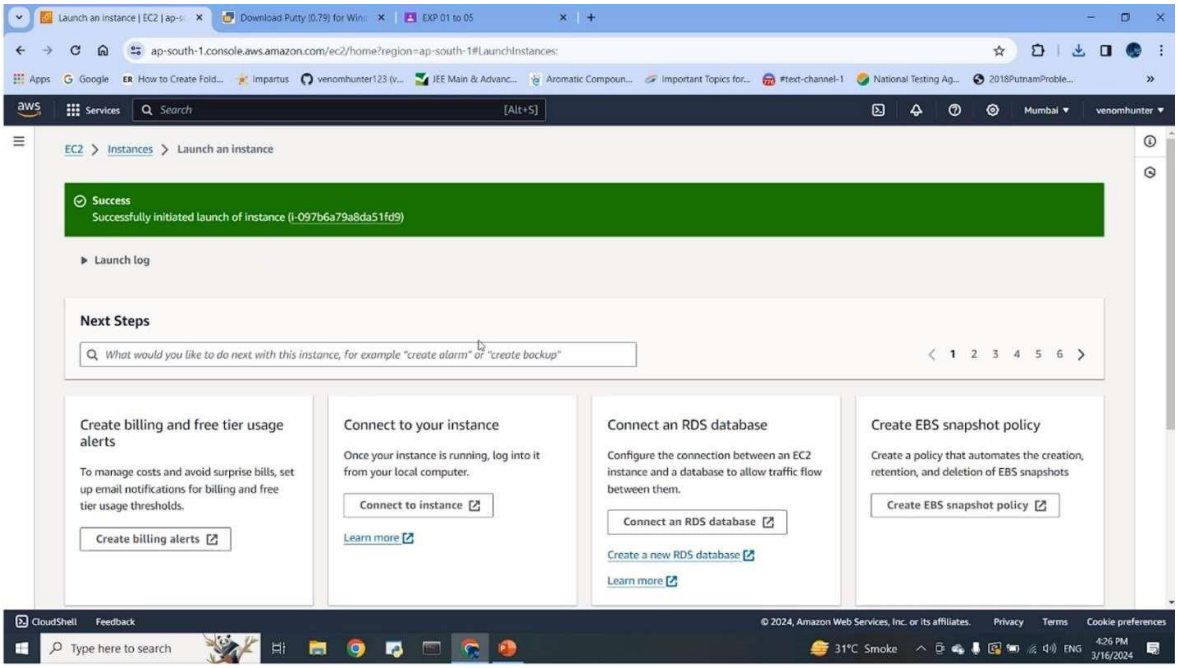
- You will be asked for a username and password; So username could be Ubuntu and Password will be added by you while installing the command. And Yeah, there you go, We have successfully logged into the Remote

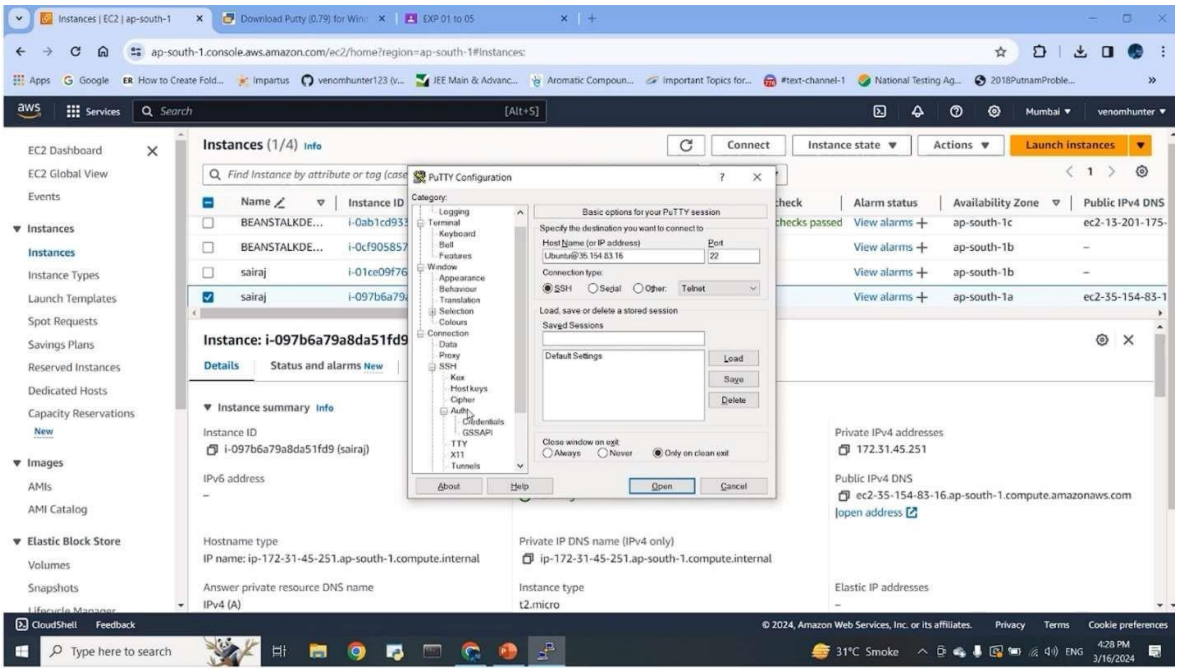
server.

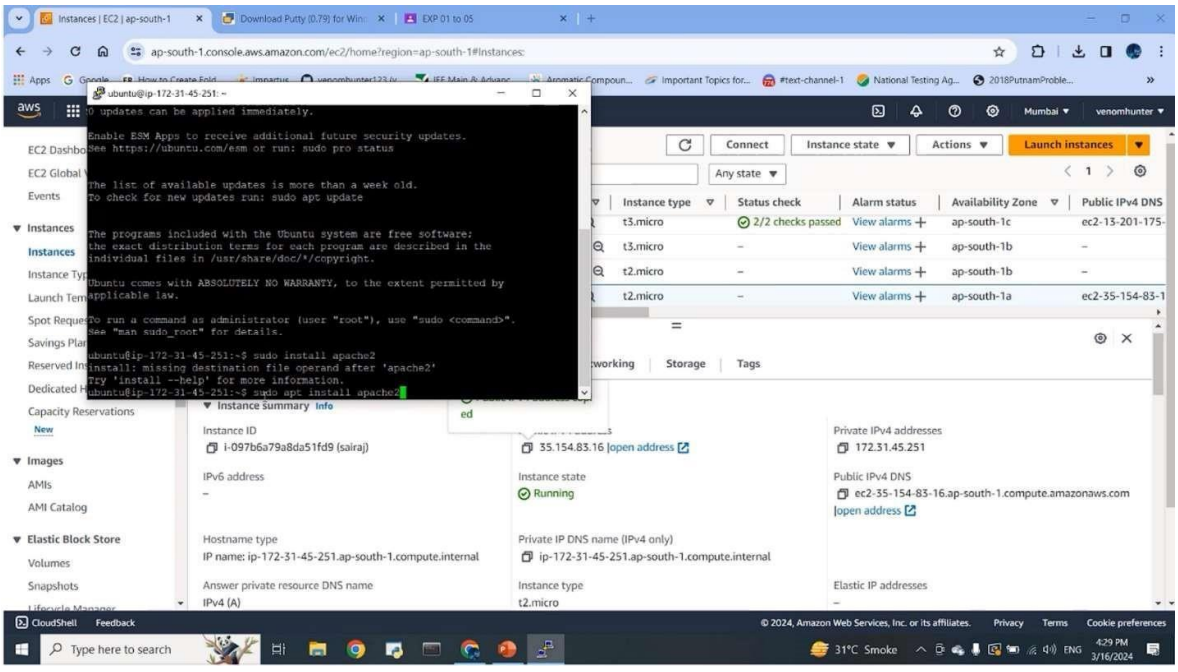
Output and Observation:




Launch an instance | EC2 | ap-s
Download Putty (0.79) for Win
EXP 01 to 05
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
Services
Search
[Alt+S]
Mumbai
venomhunter
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
sairaj
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
Recents
Quick Start
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-01316adc3850ef002
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Smoke
ENG
4:25 PM
3/16/2024


Launch an instance | EC2 | ap-s
Download Putty (0.79) for Win
EXP 01 to 05
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
Services
Search
[Alt+S]
Mumbai
venomhunter
EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-097b6a79a8da51fd9)
Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
1 2 3 4 5 6
Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
Create billing alerts
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more
Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
Create EBS snapshot policy
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Smoke
ENG
4:26 PM
3/16/2024


PuTTY Configuration
Instances (1/4) Info
Instance: i-097b6a79a8da51fd9
Details
Status and alarms New
Instance summary Info
Instance ID
i-097b6a79a8da51fd9 (sairaj)
Private IPv4 addresses
172.31.45.251
Public IPv4 DNS
ec2-35-154-83-16.ap-south-1.compute.amazonaws.com
Hostname type
IP name: ip-172-31-45-251.ap-south-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-45-251.ap-south-1.compute.internal
Instance type
t2.micro


ubuntu@ip-172-31-45-251: ~
Enable ESM Apps to receive additional future security updates.
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-45-251:~$ sudo install apache2
install: missing destination file operand after 'apache2'
Try 'install --help' for more information.
ubuntu@ip-172-31-45-251:~$ sudo apt install apache2
Instance summary Info
Instance ID
i-097b6a79a8da51fd9 (sairaj)
Private IPv4 addresses
172.31.45.251
Public IPv4 DNS
ec2-35-154-83-16.ap-south-1.compute.amazonaws.com
Instance state
Running
Hostname type
IP name: ip-172-31-45-251.ap-south-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-45-251.ap-south-1.compute.internal
Instance type
t2.micro

Conclusion:

Amazon Elastic Compute Cloud (EC2) offers numerous advantages as an Infrastructure as a Service (IaaS) solution. Firstly, EC2 provides scalable computing capacity, allowing users to easily adjust their resources based on demand fluctuations, thus optimizing costs and performance. Secondly, it offers a wide selection of instance types tailored to various workloads, ensuring flexibility and efficiency in resource allocation. Additionally, EC2's integration with other AWS services facilitates seamless deployment and management of complex applications, enabling rapid development cycles and improved operational efficiency. Overall, EC2's robust infrastructure, scalability, and integration capabilities make it a preferred choice for organizations seeking reliable and scalable cloud computing solutions.