
amazon EC2
LINUX

❖ EC2 LINUX INSTANCES:

- Linux is a Unix-like OS that was created by Linus Torvalds in 1991. It is developed using open-source principles, which means it is transparent, collaborative, and community-driven.
- An instance is a virtual server in the AWS Cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.
- The instance is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume).

LOGIN USER NAME:

- Amazon, RHEL : ec2-user
- Suse : ec2-user
- Fedora : ec2-user or fedora
- Ubuntu : ubuntu
- centos : centos
- Debian : admin

PORT IN SG:

- **Secure Shell (SSH) : 22**

GETTING ADMIN ACCESS:

- Run a Command **: $sudo -i (or) $sudo su –**

➢ CONNECTING EC2 LINUX INSTANCE:

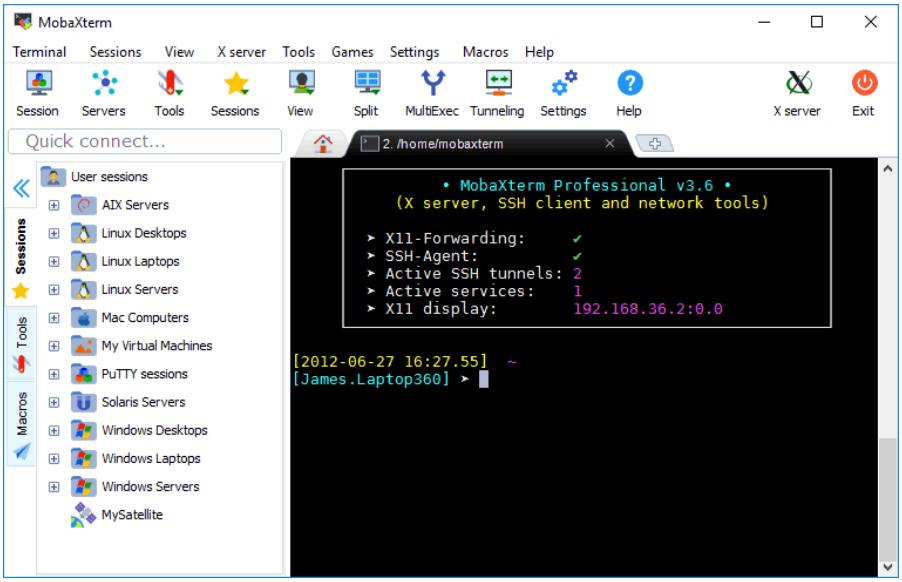
CONNECT TO INSTANCE USING SSH:

- You can connect to your Linux instance using any SSH client. If you are running Windows on your computer, open a terminal and run the ssh command to verify that you have an SSH client installed.

  $ssh -i key-pair-name.pem ec2-user@ip-address / hostname

CONNECT TO INSTANCE USING MOBAXTERM:

- MobaXterm is a remote computing toolbox that provides a variety of functions for Windows users
- MobaXterm is designed for programmers, webmasters, IT administrators, and other users who need to handle remote jobs.
- It's a popular tool for developers in the IT industry.



CONNECT TO INSTANCE USING GIT BASH:

- Git Bash is an application that adds an emulation layer on Microsoft Windows environments to use Git command-line experience.
- Bash is an acronym for Bourne Again Shell.
- A shell is a terminal application used to interface with an operating system through written commands.

$ssh -i key-pair-name.pem ec2-user@ip-address / hostname

CONNECT TO INSTANCE USING LINUX TERMINAL:

- You can connect to your Linux instance using any SSH client.
- If you are running Linux on your computer, open a terminal and run the ssh command to verify that you have an SSH client installed.

Go to Key Pair Location:

$cd Downloads

Change Key Pair File permissions:

$chmod 400 key-pair-name

Connect Instance:

$ssh -i key-pair-name.pem ec2-user@ip-address / hostname

CONNECT TO INSTANCE USING PUTTY:

PUTTY:

- PuTTY is a **free** and **open-source** terminal for **SSH** and **telnet.**
- It is a **Windows client. Mac** and **Linux ports** are existed.

PUTTYGEN:

- PuTTYgen is a key generator tool for creating SSH keys for PuTTY.
- The basic function is to create **public** and **private key pairs.**
- PuTTY stores keys in its own format in **.ppk** files.

➢ APACHE WEB SERVER (HTTPD):

- A web server is a network service that serves content (**web pages**) to a client over the web.
- Web servers are also known as **HTTP** (Hypertext Transport Protocol) servers.
- Apache is also an option to use the **SSL protocol**, making websites safe and secure.

MAIN CONFIGURATION FILES

- **Package** : httpd
- **Document Root Location** : /var/www/html
- **Configuration File** : /etc/httpd/conf/httpd.conf
- **Default Web Page** : /etc/httpd/conf.d/welcome.conf
- **Log Files Location** : /var/log/httpd/
  access_log & error_log
- **Ports:** : http-80, https-443
- **Service and Daemon** : httpd

➢ EC2 LINUX BOOTSTRAPPING / USER-DATA:

- Run commands on your Linux instance at launching.
- Refers to a self-starting process or set of commands without external input.
- We can bootstrap the instance for installing packages, running updates and configuring various settings).

USER DATA AND SHELL SCRIPTS:

- If you are familiar with shell scripting, this is the easiest and most complete way to send instructions to an instance at launch. Adding these tasks at boot time adds to the amount of time it takes to boot the instance. You should allow a few minutes of extra time for the tasks to complete before you test that the user script has finished successfully.
- User data shell scripts must start with the #! characters and the path to the interpreter you want to read the script (commonly /bin/bash).
- Scripts entered as user data are run as the root user, so do not use the sudo command in the script.

- **Run Commands for http Web Server:**

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "Welcome To Apache Web Server...!" >/var/www/html/index.html
```

OUTPUT LOG FILE:

- The **cloud-init** output log file captures console output so it is easy to debug your scripts following a launch if the instance does not behave the way you intended.
- To view the log file, connect to the instance and open /var/log/cloud-init-output.log.

```
$cat /var/log/cloud-init-output.log
```

NOTE: When a user data script is processed, it is copied to and run from **/var/lib/cloud/instances/instanceid/.** The script is not deleted after it is run. Be sure to delete the user data scripts from /var/lib/cloud/instances/instance-id/ before you create an AMI from the instance.

UPDATE THE INSTANCE USER DATA:

- To update the instance user data, you must first stop the instance. If the instance is running, you can view the user data but you cannot modify it.

To modify instance user data

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Instance state**, **Stop instance**. If this option is disabled, either the instance is already stopped or its root device is an instance store volume.
4. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions**, **Instance settings**, **Edit user data**.
6. Modify the user data as needed, and then choose **Save**.
7. Start the instance. The new user data is visible on your instance after you start it; however, user data scripts are not run.

➢ HOW DO I UTILIZE USER DATA TO AUTOMATICALLY RUN A SCRIPT WITH EVERY RESTART OF MY AMAZON EC2 LINUX INSTANCE:

- By default, user data command and **cloud-init** directives run only during the first boot cycle when an EC2 instance is launched.
- However, you can configure your user data command and **cloud-init** directives with a mime **multi-part** file.
- A mime multi-part file allows your command to override how frequently user data is run in the **cloud-init** package. Then, the file runs the user command.

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment;
 filename="cloud-config.txt"

#cloud-config
cloud_final_modules:
- [scripts-user, always]
--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "Welcome To UPDATED DATA...!" >/var/www/html/index.html
--//--
```

ACCESS INSTANCE USERDATA FROM WITHIN AN EC2 INSTANCE:

- To retrieve user data from an instance, use one of the following URIs.

  Create a token:

  $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/user-data

  Get user data:

  $curl http://169.254.169.254/latest/user-data/

➢ META-DATA:

- Metadata is data about your instance that you can use to configure or manage the running instance.
- This data is divided into various categories such as hostname, events, and security groups, among others.
- The metadata server stores and provides access to this instance-specific information, allowing users and applications to retrieve relevant details about the instance's configuration and environment.

→ **Make an HTTP request:**
$curl http://169.254.169.254/latest/meta-data

→ **Create a token:**
$export TOKEN=$(curl -X PUT -H "X-aws-ec2-metadata-token-ttl-seconds: 300" http://169.254.169.254/latest/api/token)

→ **Fetch the metadata using the token:**
$curl http://169.254.169.254/latest/meta-data -H "X-aws-ec2-metadata-token: $TOKEN

→ **Choose a metadata category:**
$curl http://169.254.169.254/latest/meta-data/instance-id -H "X-aws-ec2-metadata-token: $TOKEN

➢ DYNAMIC DATA:

- **Dynamic data** includes an instance identity document that is generated when the instance is launched.

→ **To retrieve dynamic data from within a running instance, use one of the following URIs.**

$curl http://169.254.169.254/latest/dynamic/instance-identity/

$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"

$curl http://169.254.169.254/latest/dynamic/instance-identity/

➢ NETWORK INTERFACES (ENI's):

- **Elastic network interface** is a logical networking component that represents a virtual network card.
- Each instance has a default network interface, called the **primary network interface**.
- You can create and attach additional network interfaces.
- **It can include the following attributes:**
  - A primary private IPv4 address from the IPv4 address range of your VPC
  - One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
  - One Elastic IP address (IPv4) per private IPv4 address
  - One public IPv4 address
  - One or more IPv6 addresses
  - One or more security groups
  - A MAC addresses
  - A source/destination check flag
  - A description

➢ ELASTIC IP-ADDRESS (EIP):

- An EIP is a **Static Ipv4 address** designed for **Dynamic cloud computing**.
- Attaching an EIP to an instance will replace its default public IP address for as long as it is attached.
- By default, all AWS accounts have a quota of **five (5) Elastic IP addresses** per Region, because public (IPv4) internet addresses are a scarce public resource. We strongly recommend that you use Elastic IP addresses primarily for their ability to remap the address to another instance in the case of instance failure, and to use DNS hostnames for all other inter-node communication.

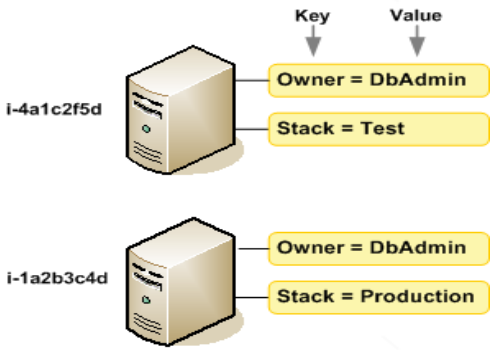
➢ EC2 SERVICE QUOTAS (LIMITS):

- When you create your AWS account, we set default limits on AWS resources on a per-Region basis.
- The Service Quotas console is a central location where you can view and manage your quotas for AWS services, and request a quota increase for many of the resources that you use.

NOTE: You can request a quota increase for each Region.

➢ RESOURCES & TAGS:

- To manage your instances, images, volumes and snapshots, you can optionally assign your own metadata to each resource in the form of tags.
- Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment.
- A tag is a label that you assign to an AWS resource. Each tag consists of a **key** and an optional **value**.



➢ SERIAL CONSOLE:

- EC2 Serial Console provides a simple and secure way to troubleshoot boot and network connectivity issues by establishing a connection to the serial port of an instance.
- It provides a one-click, text-based access to an instances' serial port as though a monitor and keyboard were attached to it.
- Previously, you could get serial console output as logs or a screenshot through the EC2 management console, API or CLI.
- You can interactively run troubleshooting commands for resolving boot and network configuration issues.
- EC2 Serial Console is ideal for situations where you are unable to connect to your instance via normal SSH or RDP.

NOTE: EC2 Serial Console Access is available for **Nitro virtual instances** running either Windows or Linux