1. **What is .PEM file?**

A.

* A .pem file you get when creating an AWS EC2 instance is a **private key file** used for securely connecting to that instance via SSH (Secure Shell).
* This file is essential for managing and accessing your cloud server securely.
* It stores the private key that authenticates your login to the EC2 instance, enabling secure, encrypted access.
* Can use this file with an SSH client to connect to your EC2 instance remotely without a password.
* You can only download the .pem file once at the time you create the key pair during instance setup. If you lose it, you cannot download it again from AWS.
* Keep it safe and private. If lost, you would need to create a new key pair and manually update your instance to regain access.

1. **How do we get the EC2 Instance username?**

**A.**

* The default username for an AWS EC2 instance depends on the Amazon Machine Image (AMI) used to launch it.
* You use this username along with the .pem key file to SSH into the instance
* If you are using a custom AMI or an uncommon one, check with the AMI provider or documentation for the default username
* you can change the username on an EC2 instance, but it requires some manual steps because the default username is tied to the initial SSH login and the system's user accounts

Here are some common default usernames by AMI:

1. Amazon Linux or Amazon Linux 2 AMI: **ec2-user**
2. CentOS AMI: **centos** or **ec2-user**
3. Debian AMI: **admin**
4. Fedora AMI: **fedora** or **ec2-user**
5. RHEL AMI: **ec2-user** or **root**
6. SUSE AMI: **ec2-user** or **root**
7. Ubuntu AMI: **ubuntu**
8. Oracle AMI: **ec2-user**
9. Bitnami AMI: **bitnami**
10. **What is SSH and -I while connecting to EC2 instance?**

A.

* SSH stands for **Secure Shell**. It is a network protocol that allows you to securely connect and manage remote computers over an unsecured network, like the internet.
* With SSH, you can remotely log into your AWS EC2 instance's command line interface to run commands and manage your server.
* The -i option in the SSH command specifies the **identity file**—in other words, the private key file (your .pem file) that SSH uses to authenticate your access to the instance.

1. **What is cron job?**

A.

* A cron job is a command or script scheduled to run periodically at fixed times, dates, or intervals.
* Managed by a daemon called cron (runs in the background).
* Cron jobs are listed in a configuration file called a crontab (cron table).
* The system reads the crontab entries and executes commands at the scheduled times.
* Use crontab -e to edit your user’s cron jobs.
* Use crontab -l to list your scheduled cron jobs.
* Use crontab -r to remove your cron jobs.

Syntax:

A typical cron job has a schedule specified by **5-time fields** followed by the command to execute:

\* \* \* \* \* command\_to\_run

1st \*: minute (0-59)

2nd \*: hour (0-23)

3rd \*: date [day of the month] (1-31)

4th \*: month (1-12)

5th \*: day of the week (0-6, Sunday=0)

EX:

0 0 \* \* \* run daily at midnight

\*/15 \* \* \* \* run every 15 minutes

0 22 \* \* 5 Run every Friday at 10 PM

sudo grep CRON /var/log/cron

1. **What is wheel group?**

A.

* The **wheel** group is a special user group on many Linux distributions (including Amazon Linux and Red Hat-based systems)
* Members of the wheel group have **sudo** (administrative) privileges, meaning they can run commands as root using sudo
* Adding a user to the wheel group grants that user the ability to perform administrative tasks without logging in directly as root.

**How to add new user to this group:**

* + Suppose you created a new user called newuser. To grant this user sudo privileges (via the wheel group), run:

sudo usermod -aG wheel newuser

* + -aG wheel adds the user to the wheel group without removing existing group memberships.
  + After this, newuser can run commands with sudo
  + confirm group membership for the user

groups newuser