HYSON MAVEN-DEMO CLASS

To create a security group and open port 22 (SSH) on an AWS EC2 instance, you can follow these steps:

1. **Log in to the AWS Management Console:**
    - Go to AWS Management Console.
    - Sign in with your AWS account credentials.
2. **Navigate to the EC2 Dashboard:**
    - Once logged in, go to the EC2 Dashboard by clicking on the "Services" menu and selecting "EC2" under the "Compute" section.
3. **Create a Security Group:**
    - In the EC2 Dashboard, locate the "Security Groups" option in the left-hand menu under the "Network & Security" section and click on it.
4. **Click on "Create Security Group":**
    - Click the "Create Security Group" button at the top of the page.
5. **Configure Security Group Settings:**
        - Fill in the following details:
        - **Security group name:** Give your security group a descriptive name, e.g., "SSH-Access."
        - **Description:** Optionally, provide a brief description for the security group.
        - **VPC:** Select the Virtual Private Cloud (VPC) associated with your EC2 instance.
6. **Inbound Rules - Open Port 22 (SSH):**
        - In the "Inbound rules" section, click the "Add Rule" button to add a new rule.
        - Configure the rule as follows:
        - **Type:** Choose "SSH" from the dropdown list. This automatically selects "TCP" as the protocol and sets the port range to 22.
        - **Source:** Choose "My IP" to allow SSH access only from your current IP address. Alternatively, you can specify a specific IP range or another source depending on your requirements.
7. **Review and Create the Security Group:**
    - Double-check the configuration to ensure it matches your requirements.
    - Click the "Create security group" button to create the security group.

To create a Red Hat EC2 instance with a T2. medium instance type and 4GB of RAM in AWS, follow these steps:

1. **Navigate to the EC2 Dashboard:**
    - Click on "Services" in the top left corner, and then select "EC2" under the "Compute" section.
2. **Launch an EC2 Instance:**
    - In the EC2 Dashboard, click the "Instances" link in the left-hand menu.
3. **Launch Instance:**
    - Click the "Launch Instances" button.
4. **Choose an Amazon Machine Image (AMI):**
    - In the "Step 1: Choose an Amazon Machine Image (AMI)" section, select a Red Hat Enterprise Linux AMI that meets your requirements. You can use the search bar to find one. Make sure you choose a Red Hat Enterprise Linux AMI.
5. **Choose an Instance Type:**
    - In the "Step 2: Choose an Instance Type" section, scroll down or use the search bar to find the "t2. medium" instance type. Select it. This instance type provides 4GB of RAM.
6. **Configure Instance Details:**
    - In the "Step 3: Configure Instance Details" section, you can specify various instance configuration settings, such as the number of instances, network settings, and more. You can leave most settings at their default values, or adjust them as needed for your use case.
7. **Add Storage:**
    - In the "Step 4: Add Storage" section, you can configure the storage options for your instance. The default settings are usually sufficient, but you can increase the storage size if needed.
8. **Add Tags (Optional):**
    - In the "Step 5: Add Tags" section, you can add tags to your instance for better organization. Tags are key-value pairs that help you identify and categorize your instances.
9. **Configure Security Group:**
    - In the "Step 6: Configure Security Group" section, select the security group you created earlier and that allows SSH access on port 22.
10. **Review Instance Launch:**

- In the "Step 7: Review Instance Launch" section, review your instance's configuration to ensure it matches your requirements.

11. **Launch the Instance:**
    - Click the "Launch" button.
12. **Select or Create a Key Pair:**
    - If you already have an existing key pair, select it. If not, you can create a new key pair. You'll need this key pair to SSH into your EC2 instance.
13. **Launch Instances:**
    - After selecting the key pair, click the "Launch Instances" button.

That's it! You've now created a Red Hat EC2 instance with the T2. medium instance type and 4GB of RAM. You can use this instance for your desired purposes.

Let's break down each command:

1. **sudo hostnamectl set-hostname maven**

- This command sets the hostname of your system to "maven" using `hostnamectl`. The hostname is a label assigned to a computer on a network.

2. **sudo su - ec2-user**

- This switches the current user to `ec2-user` with elevated privileges. It's often necessary to perform system configuration tasks.

3. **cd /opt**

- This changes the current directory to `/opt`, which is typically used for the installation of software and additional packages.

4. **sudo yum install wget nano tree unzip git-all -y**

- This line installs several common utilities using the `yum` package manager:

- `wget`: A command-line utility for downloading files.

- `nano`: A simple text editor.

- `tree`: A command-line program for displaying directory trees.

- `unzip`: A utility for extracting compressed ZIP files.

- `git-all`: The complete Git package that includes Git, Git GUI, and other tools.

- The `-y` flag automatically answers "yes" to any prompts during the installation process.

5. **sudo yum install java-11-openjdk-devel java-1.8.0-openjdk-devel -y**

- This command installs both OpenJDK 11 (Java Development Kit) and OpenJDK 1.8 (Java Development Kit) using `yum`. This is important because you need Java installed to run Maven, and different applications might require different Java versions. The `-y` flag automatically answers "yes" to any prompts during the installation process.

6. java -version

- This command verifies the Java installation by displaying the version of Java currently installed on the system.

7. git --version

- This command verifies the Git installation by displaying the version of Git currently installed on the system.

The provided commands are used to download and install Apache Maven on your system. Here's an explanation of each step:

1. **Download the Maven Software:**
    - **sudo wget https://dlcdn.apache.org/maven/maven-3/3.9.4/binaries/apachemaven-3.9.4-bin.zip**
    - This command uses **wget** to download the Apache Maven 3.9.4 binary distribution (ZIP file) from the specified URL. It's a direct download link for the Apache Maven distribution.
2. **Unzip the Maven Software:**
    - **sudo unzip apache-maven-3.9.4-bin.zip**
    - This command unzips the downloaded ZIP file, extracting the contents into the current directory. It will create a directory called **apache-maven-3.9.4** with Maven's files.
3. **Remove the Downloaded ZIP File:**
    - **sudo rm -rf apache-maven-3.9.4-bin.zip**
    - This command removes the downloaded ZIP file to save disk space, as it's no longer needed once Maven has been extracted.
4. **Rename the Maven Directory:**
    - **sudo mv apache-maven-3.9.4/ maven**
    - This command renames the **apache-maven-3.9.4** directory to simply **maven** for easier reference and usage. It moves the entire directory to the new name.

The provided instructions are for setting environment variables for a specific user, ec2-user, in their ~/.bash_profile file. These environment variables are necessary for configuring Apache Maven. Here's how you can do it:

1. **Edit the .bash_profile for ec2-user:**
    - Use the **vi** text editor to edit the **.bash_profile** file for the **ec2-user**:

      vi ~/.bash_profile

2. **Add the Environment Variables:**
    - Inside the **.bash_profile** file, add the following lines at the end of the file:

      export M2_HOME=/opt/maven export

      PATH=$PATH:$M2_HOME/bin

These lines set the **M2_HOME** variable to the Maven installation directory (**/opt/maven**) and append the Maven binary directory to the system's **PATH**.

3. **Save and Exit the Editor:**
    - If you're using **vi**, save and exit the file by following these steps:
    - Press **Esc** key to exit insert mode (if you are in insert mode).
    - Type: **wq** and press **Enter**. This saves the file and exits **vi**

Refreshing the profile file and verifying if Maven is running is a good practice after making changes to the environment variables. Here are the steps to do that:

1. **Refresh the .bash_profile:**
    - To apply the changes made to the **.bash_profile**, run the following command:

      source ~/.bash_profile

This command reloads the profile for your current session, ensuring that the updated environment variables (**M2_HOME** and **PATH**) are in effect.

2. **Verify if Maven is Running:**
    - To check if Maven is properly installed and running, you can run the following command:

      mvn -version

This will display information about the installed Maven version, as well as the Java version it is using.

Here's how you can create a directory for your Git repository:

1. **Change to Your Home Directory:**
    - If you're not already in your home directory, you can use the following command to navigate to it: **cd ~**
2. **Create a Directory for Git Repositories:**
    - You can create a new directory to hold your Git repositories. For example, let's create a directory named "maven-project": **mkdir maven-project**
3. **Navigate to the New Directory:**
    - Change into the "maven-project" directory:

      cd maven-project

Now you have a directory named " maven-project" within your home directory, and you can use it to clone your Git repositories. You can clone a Git repository into this directory using a command like:

git clone https://github.com/Landmark-Technologies/maven-standalone-application

The repository will be cloned into the " maven-project" directory, which you created in your home folder.

Navigate in to this project and run mvn package to generate a target directory which would contain the .war, .jar or the .ear file.

THE END.