

| Lab 251

Bash Shell Scripts

Student: Mane Zakarian

Bootcamp: Forge AWS re/Start UYMON5

Date: 2023





Objectives

In this lab, you will:

- Create a bash script that will automate the backup of a folder

Accessing the AWS Management Console

1. At the top of these instructions, choose **Start Lab** to launch your lab. A **Start Lab** panel opens, and it displays the lab status.

Tip: If you need more time to complete the lab, choose the Start Lab button again to restart the timer for the environment.

2. Wait until you see the message *Lab status: ready*, then close the **Start Lab** panel by choosing the X.
3. At the top of these instructions, choose **AWS**. This opens the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, a banner or icon is usually at the top of your browser with a message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and then choose **Allow pop ups**.

4. Arrange the AWS Management Console tab so that it displays alongside these instructions. Ideally, you will be able to see both browser tabs at the same time so that you can follow the lab steps more easily.

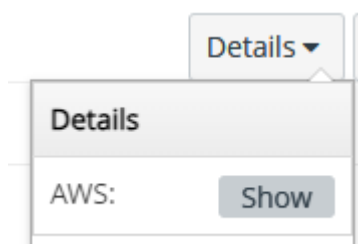


Task 1: Use SSH to connect to an Amazon Linux EC2 instance

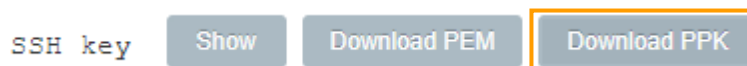
In this task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations.

Windows Users: Using SSH to Connect

1. Select the **Details** drop-down menu above these instructions you are currently reading, and then select **Show**. A Credentials window will be presented.



2. Select the **Download PPK** button and save the **labsuser.ppk** file.



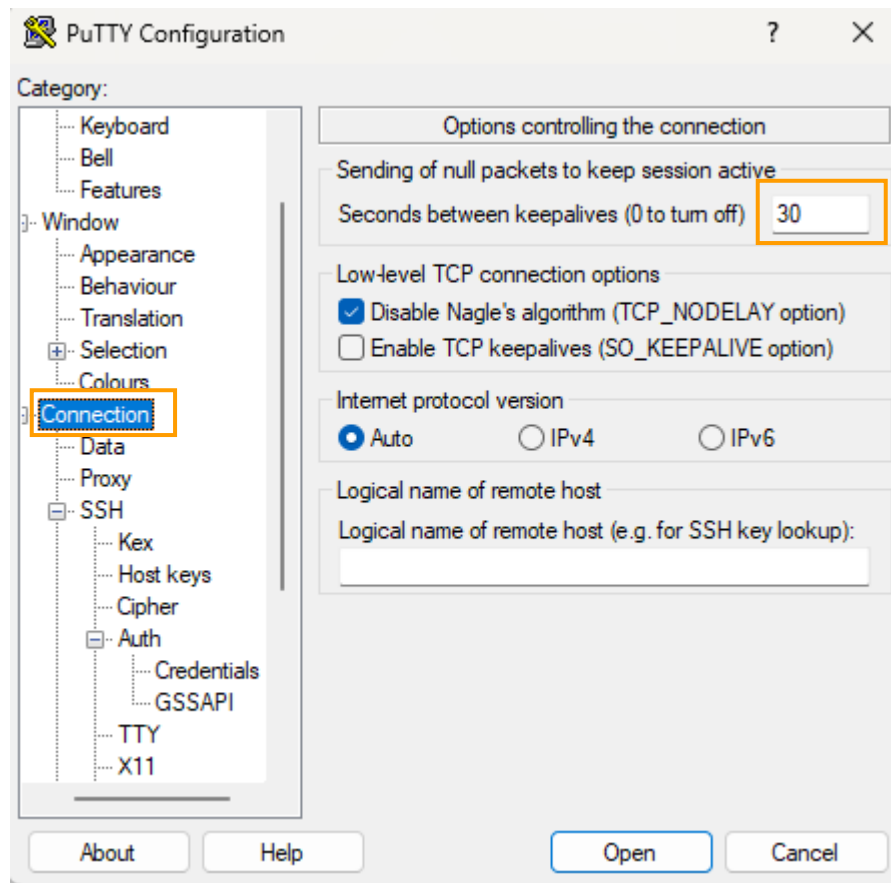
3. Make a note of the **PublicIP** address.



4. Then exit the Details panel by selecting the X.
5. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer.
6. Open **putty.exe**



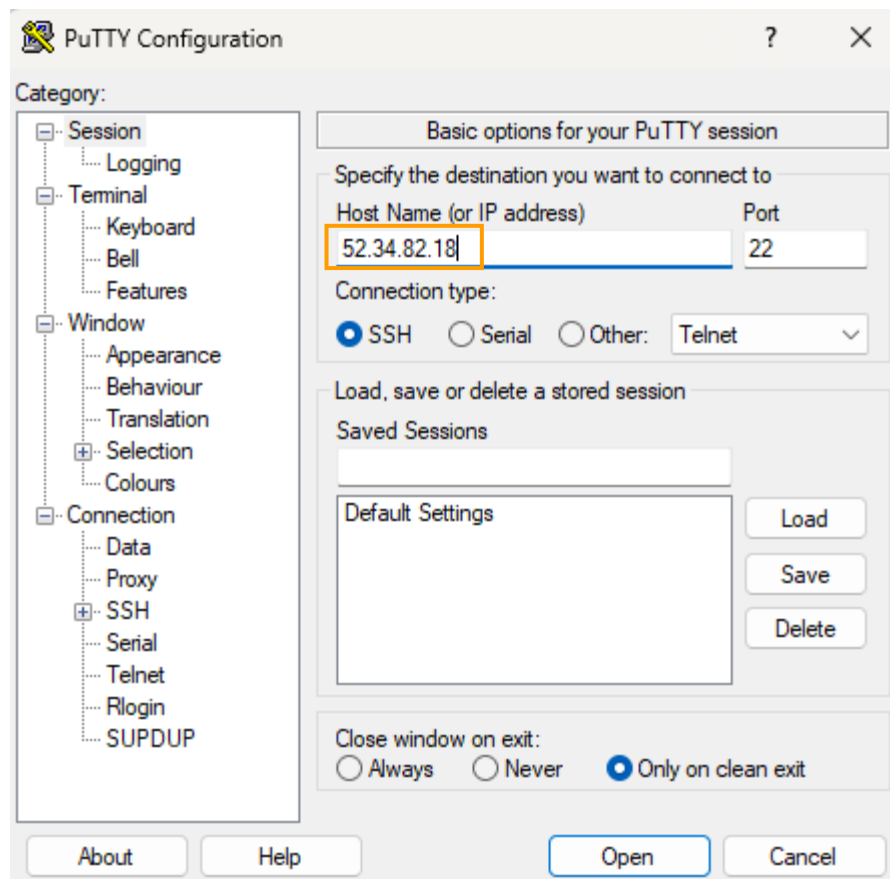
7. Configure PuTTY timeout to keep the PuTTY session open for a longer period of time.
 - Select **Connection**
 - Set **Seconds between keepalives** to 30





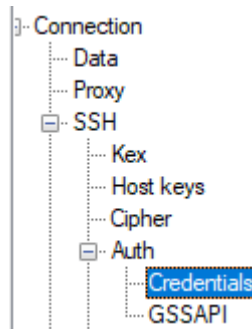
8. Configure your PuTTY session:

- Select **Session**
- **Host Name (or IP address):** Paste the **Public DNS or IPv4 address** of the instance you made a note of earlier. Alternatively, return to the EC2 Console and select **Instances**. Check the box next to the instance you want to connect to and in the *Description* tab copy the **IPv4 Public IP** value

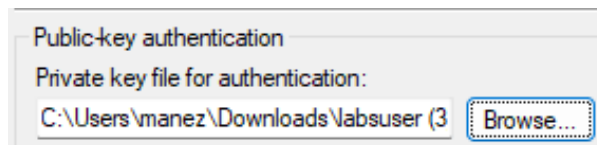




- Back in PuTTY, in the **Connection** list, expand **SSH** and select **Auth** (*don't expand it*)



- Select **Browse** and select the lab#.ppk file that you downloaded



- Select **Open** to select it and then select **Open** again.
9. Select **Yes**, to trust and connect to the host.
10. When prompted **login as**, enter: `ec2-user` This will connect you to the EC2 instance.





Task 2: Write a shell script

In this task, you create a Bash shell script that automates the creation of a backup of the **CompanyA** folder as a compressed archive. The name of the archive will be in the format **date of the day-backup-companyA.tar.gz**.

Helpful Hint

You may have to use **sudo** to complete this task if you are not root.

24. To validate that you are in the home folder, enter the following command, and press Enter.

```
pwd
```

Expected Output:

```
[ec2-user@ip-10-0-10-209 ~]$ pwd
/home/ec2-user
```

25. To create a generic shell script called **backup.sh**, enter the following command, and press Enter.

```
touch backup.sh
```

```
[ec2-user@ip-10-0-10-209 ~]$ touch backup.sh
[ec2-user@ip-10-0-10-209 ~]$
```

26. To change the file privileges to make **backup.sh** be executable, enter the following command, and press Enter.

```
sudo chmod 755 backup.sh
```

```
[ec2-user@ip-10-0-10-209 ~]$ sudo chmod 755 backup.sh
[ec2-user@ip-10-0-10-209 ~]$
```



27. Use your preferred text editor to open the **backup.sh** file for editing.

If you choose nano, please enter the following command:

```
nano backup.sh
```

```
[ec2-user@ip-10-0-10-209 ~]$ nano backup.sh  
[ec2-user@ip-10-0-10-209 ~]$
```

If you choose vim enter the following command and press Enter to go to the next line.

```
vi backup.sh
```





28. To activate insert mode, enter `i`

```
-- INSERT --
```

29. On line 1 of the script, enter `#!/bin/bash` to add the shebang line, and press Enter to go to the next line.

```
#!/bin/bash
```

30. To create a variable for the current date, enter `DAY="$(date +%Y_%m_%d_%T_%H_%M)"` and press Enter to go to the next line.

```
#!/bin/bash
DAY="$(date +%Y_%m_%d_%T_%H_%M)"
```

Note:

You can use the `date +%Y_%m_%d` command to retrieve the current date and time. This command formats this information as follows: 2021_08_31

31. To create a variable for the backup file for the day, enter `BACKUP="/home/$USER/backups/$DAY-backup-CompanyA.tar.gz"` and press Enter to go to the next line.

```
#!/bin/bash
DAY="$(date +%Y_%m_%d_%T_%H_%M)"
BACKUP="/home/$USER/backups/$DAY-backup-CompanyA.tar.gz"
```

Note:



Lab 251 | Bash Shell Scripts

`$USER` returns the current user, which is `ec2-user` in this lab. This is the equivalent of entering the `whoami` command in the shell. The created archive will be located in `/home/ec2-user/backups`.

32. On the next line, enter `tar -csvgzf $BACKUP /home/$USER/CompanyA` and press Enter.

```
#!/bin/bash
DAY="$(date +%Y_%m_%d_%T_%H_%M)"
BACKUP="/home/$USER/backups/$DAY-backup-CompanyA.tar.gz"
tar -csvgzf $BACKUP /home/$USER/CompanyA
```

Contents of `backup.sh` script written so far:

```
#!/bin/bash
DAY="$(date +%Y_%m_%d)"
BACKUP="/home/$USER/backups/$DAY-backup-CompanyA.tar.gz"
tar -csvgzf $BACKUP /home/$USER/CompanyA
```

33. With your current text editor, save your script and exit from the editor. If you have vim editor, press the Esc key, enter `:wq` and press Enter.

If you have nano, save the change with: `Ctrl + X`, then `Y` and Enter

```
[ec2-user@ip-10-0-10-209 ~]$ nano backup.sh
[ec2-user@ip-10-0-10-209 ~]$
```



34. To run **backup.sh**, enter the following command, and press Enter.

```
./backup.sh
```

```
[ec2-user@ip-10-0-10-209 ~]$ ./backup.sh
tar: Removing leading `/' from member names
/home/ec2-user/CompanyA/
/home/ec2-user/CompanyA/Management/
/home/ec2-user/CompanyA/Management/Sections.csv
/home/ec2-user/CompanyA/Management/Promotions.csv
/home/ec2-user/CompanyA/Employees/
/home/ec2-user/CompanyA/Employees/Schedules.csv
/home/ec2-user/CompanyA/Finance/
/home/ec2-user/CompanyA/Finance/Salary.csv
/home/ec2-user/CompanyA/Finance/Hourly.csv
/home/ec2-user/CompanyA/HR/
/home/ec2-user/CompanyA/HR/Managers.csv
/home/ec2-user/CompanyA/HR/Assessments.csv
/home/ec2-user/CompanyA/IA/
/home/ec2-user/CompanyA/SharedFolders/
```

35. To verify that the archive is created in the **backups** folder, enter the following command, and press Enter.

```
ls backups/
```

Expected Output:

```
[ec2-user@ip-10-0-10-209 ~]$ ls backups/
2023_10_30_23:22:55_23_22-backup-CompanyA.tar.gz
```

You can schedule this type of script via cron to create a daily backup of the folder. You can also use other commands to copy this archive to other servers, but this option is beyond the scope of this lab.



Lab Complete



Congratulations! You have completed the lab.

36. Choose **End Lab** at the top of this page, and then select **Yes** to confirm that you want to end the lab.
37. A message *Ended AWS Lab Successfully* is briefly displayed, indicating that the lab has ended.



Commands Used:

On this lab we used several commands to perform different tasks. Here is a summary of the commands used:

Command	Description
<code>touch</code>	Creates an empty file.
<code>sudo</code>	Used to execute a command with superuser privileges.
<code>vi</code>	A text editor used to open, edit, and save text files.
<code>nano</code>	Another text editor often used for simple text editing.
<code>i</code>	In the context of <code>vi</code> , it enters insert mode for editing.
<code>day =</code>	Assigns a value to a shell variable named <code>day</code> .
<code>#!/bin/bash</code>	The shebang line that specifies the interpreter (Bash).
<code>BACKUP=</code>	Assigns a value to a shell variable named <code>BACKUP</code> . Denotes the current directory or is used to execute a script located in the current directory.