



| Lab 247

Working with Commands

Student: Mane Zakarian

Bootcamp: Forge AWS re/Start UYMON5

Date: 2023



Objectives

In this lab, you will:

- Use the **tee** command to direct output to a file
- Use the **sort** command to reorganize the contents of a .csv file
- Use the **cut** command to edit the contents of a file
- Use the **sed** command
- Use the **pipe** operator

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.

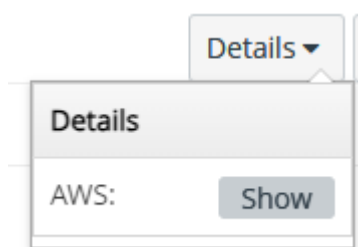


Task 1: Use SSH to connect to an Amazon Linux EC2 instance

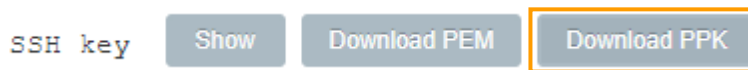
In this task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations.

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.

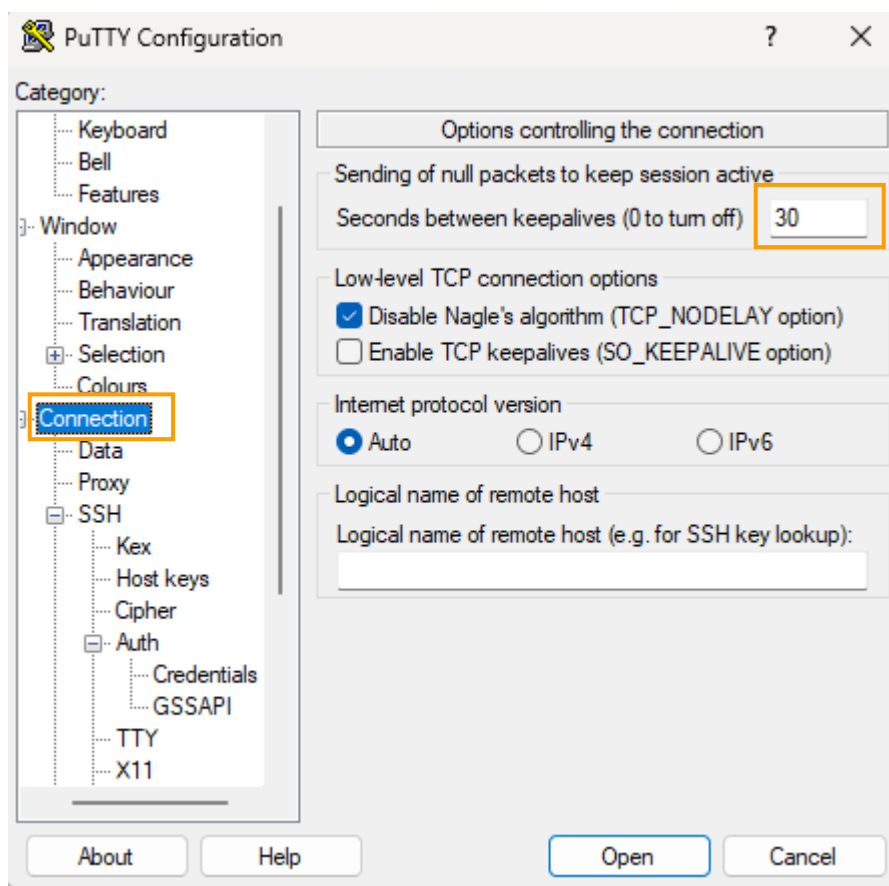
`PublicIP`

`52.34.82.18`

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**

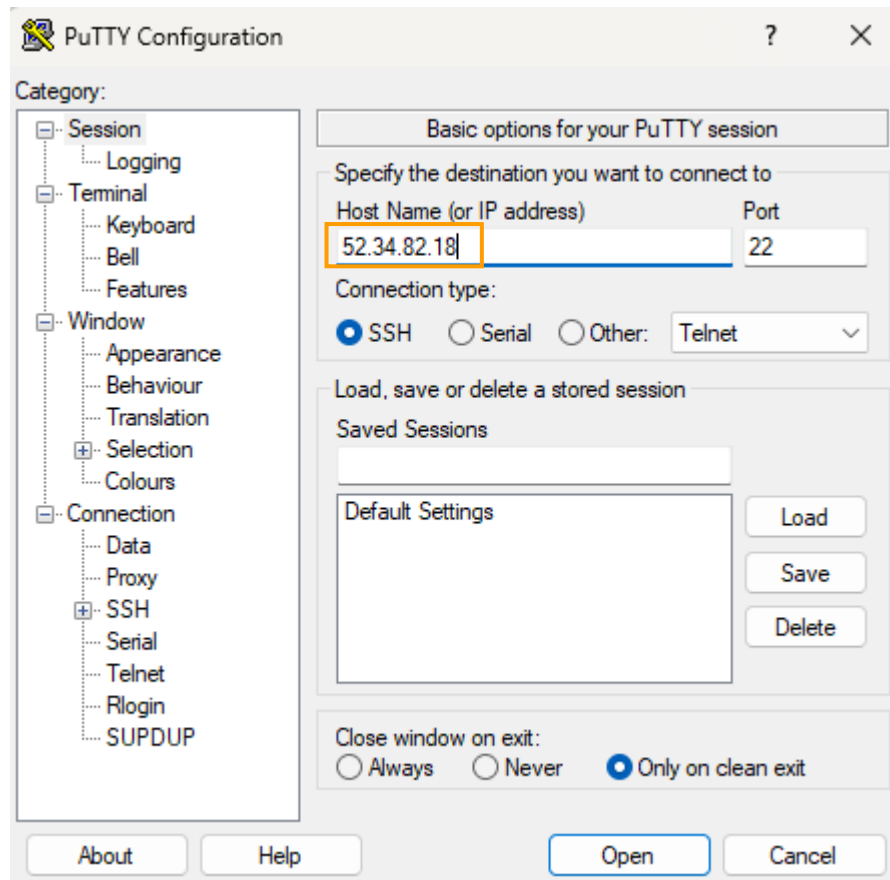


8. Configure your PuTTY session:

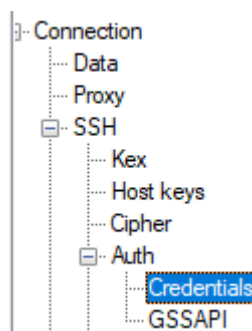
- Select **Session**



- **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

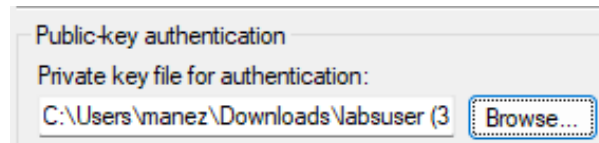


- Back in PuTTY, in the **Connection** list, expand **SSH** and select **Auth** (*don't expand it*)

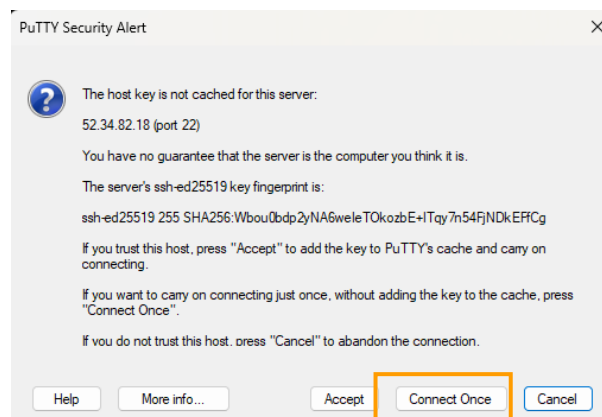




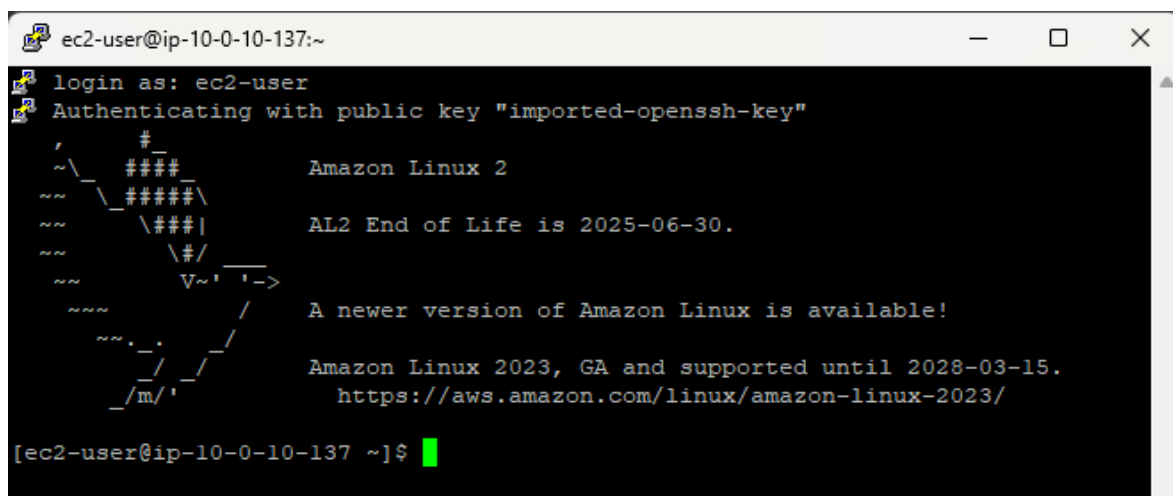
- Select **Browse** and select the lab#.ppk file that you downloaded



- 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: Use the tee command

In this task, you use the **tee** command to display the output to the screen and a file.

The **tee** command reads the standard input. In this example, the standard input is **hostname**. The **tee** command outputs the hostname to the screen (in the shell) and the designated file, which is **file1.txt**.

24. To validate that you are in the `/home/ec2-user` folder, enter `pwd` and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ pwd
/home/ec2-user
```

25. From your current location in the terminal, enter `ls` and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ ls
cities.csv  companyA  file1.txt  test.csv
```

Figure: The output from the command `hostname | tee file1.txt` is `ip-10-0-10-81.us-west-2.compute.internal`.

From the following output, you can see the standard input for **tee** in the output of the command **hostname**. The **tee** command wrote the **hostname** to the **file1.txt** and to the screen.

```
[ec2-user@ ~]$ hostname | tee file1.txt
ip-(xx-xx-xx-xx).(region).compute.internal
```

```
[ec2-user@ip-10-0-10-103 ~]$ hostname | tee file1.txt
ip-10-0-10-103.us-west-2.compute.internal
```

26. To confirm that the **file1.txt** file has been created, enter `ls` and press Enter.

```
ip-10-0-10-103.us-west-2.compute.internal
[ec2-user@ip-10-0-10-103 ~]$ ls
cities.csv  companyA  file1.txt  test.csv
```

Figure: In the current directory, there are two items present: `companyA` and `file1.txt`



Task 3: Use the sort command and pipe operator

In this task, you use the **sort** command to reorder the list within the **test.csv** file. You also use the **pipe** operator to search for the factory in Paris.

27. To validate that you are in the **/home/ec2-user** folder, enter **pwd** and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ pwd
/home/ec2-user
```

28. Enter **cat > test.csv** and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ cat > test.csv
```

29. Enter the following list as shown and press Enter.

```
Factory, 1, Paris
Store, 2, Dubai
Factory, 3, Brasilia
Store, 4, Algiers
Factory, 5, Tokyo
```

```
[ec2-user@ ~]$ cat > test.csv
Factory, 1, Paris
Store, 2, Dubai
Factory, 3, Brasilia
Store, 4, Algiers
Factory, 5, Tokyo
```

30. When you are done, press CTRL+D to exit the file.

31. To verify that the **test.csv** file has been created, enter **ls** and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ ls
cities.csv  companyA  file1.txt  test.csv
```




32. Now that you have created a few items within the **test.csv** file, use the **sort** command to reorder the list. Enter `sort test.csv` and press Enter.

The output should look like the following image. Because you used the **sort** command with no options, it sorted the list with the default action by alphabetical order, which is why **Factory** is listed before **Stores**. The command then sorts by numerical order.

```
[ec2-user@ip-10-0-10-103 ~]$ sort test.csv
Factory, 1, Paris
Factory, 3, Brasilia
Factory, 5, Tokyo
Store, 2, Dubai
Store, 4, Algiers
[ec2-user@ip-10-0-10-103 ~]$
```

Figure: When the command `sort test.csv` is ran, it sorted the contents within the file in the following order: Factory 1 Paris, Factory 3 Brasilia, Factory 5 Tokyo, Store 2 Dubai, and Store 4 Algiers.

```
[ec2-user@ ~]$ sort test.csv
Factory, 1, Paris
Factory, 3, Brasilia
Factory, 5, Tokyo
Store, 2, Dubai
Store, 4, Algiers
```

33. To look for the factory named **Paris** using the **pipe (|)** operator, enter `find | grep Paris test.csv` and press Enter.

In the following output, **find | grep Paris test.csv** searches and lists the content of the **test.csv** file and redirects the results to the **grep** command where it searches for the **Paris** pattern.

```
[ec2-user@ip-10-0-10-103 ~]$ find | grep Paris test.csv
Factory, 1, Paris
[ec2-user@ip-10-0-10-103 ~]$
```

*Figure: When the command `grep Paris test.csv` is ran, it searches for the word **Paris** and returns the following: **Factory, 1, Paris**.*



Task 4: Use the cut command

In this task, you use the `cut` command to edit the `test.csv` file.

34. To validate that you are in the `/home/ec2-user` folder, enter `pwd` and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ find | grep Paris test.csv
Factory, 1, Paris
[ec2-user@ip-10-0-10-103 ~]$ pwd
/home/ec2-user
```

35. Enter `cat > cities.csv` and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ cat > cities.csv
```

36. Enter the following list as shown and press Enter.

```
[ec2-user@ip-10-0-10-103 ~]$ cat > cities.csv
Dallas, Texas
Seattle, Washington
Los Angeles, California
Atlanta, Georgia
New York, New York
```

37. When you are done, press CTRL+D to exit the file.



38. Next, you use the **cut** command to cut sections from lines of text by character. You use the **-d** (delimiter) option, the **,** option, and the **-f** (field) option. The combined command and options extract the first field of each record.

Enter the following command `cut -d ',' -f 1 cities.csv`

As you can see from the following output, the **cut** command removed everything after the **,**.

```
[ec2-user@ip-10-0-10-103 ~]$ cut -d ',' -f 1 cities.csv
Dallas
Seattle
Los Angeles
Atlanta
New York
```

*Figure: After the cut command is ran, the following is left: Dallas, Seattle, Los Angeles, Atlanta, and New York. *

```
[ec2-user@ ~]$ cut -d ',' -f 1 cities.csv
Dallas
Seattle
Los Angeles
Atlanta
New York
```



Additional Challenge:

Use only the **sed** command to make changes or do all the changes in one line. (You can use command chaining using the pipe character (|).)

Remember, the **sed** command is mainly used to replace some text in a file for different text.

```
sed 's/word being replaced/replacement word/' file name
```

The **sed** command searches the file text for an occurrence of the first string, and will replace any matches with the second.

For example, if we want to change the name of Dallas for Prueba2, we run the following command:

```
sed 's/Dallas /Prueba2/' /home/ec2-user/cities.csv
```

```
[ec2-user@ip-10-0-10-103 ~]$ sed 's/Dallas/Prueba2/' /home/ec2-user/cities.csv
Prueba2, Texas
Seattle, Washington
Los Angeles, California
Atlanta, Georgia
New York, New York
```

To find the full path of a file in Linux, you can use the **realpath** or **readlink** command. Both commands will provide you with the full path of a file.

```
realpath cities.csv
```

```
[ec2-user@ip-10-0-10-103 ~]$ realpath cities.csv
/home/ec2-user/cities.csv
```

39. Use the **sed** command to replace the first comma (,) with periods (.) in both the **cities.csv** and **test.csv** files.

```
sed 's/, /./' /home/ec2-user/cities.csv
```

```
[ec2-user@ip-10-0-10-103 ~]$ sed 's/, /./' /home/ec2-user/cities.csv
Dallas.Texas
Seattle.Washington
Los Angeles.California
Atlanta.Georgia
New York.New York
```



```
sed 's/,./' /home/ec2-user/test.csv
```

```
[ec2-user@ip-10-0-10-103 ~]$ sed 's/,./' /home/ec2-user/test.csv
Factory.1, Paris
Store.2, Dubai
Factory.3, Brasilia
Store.4, Algiers
Factory.5, Tokyo
```

Lab Complete



Congratulations! You have completed the lab.

40. Select **End Lab** at the top of this page and then select **Yes** to confirm that you want to end the lab. A panel will appear, indicating that "DELETE has been initiated... You may close this message box now."
41. Select the X in the top right corner to close the panel.



Commands Used:

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

| <i>Command</i> | <i>Description</i> |
|-----------------------|---|
| <code>pwd</code> | Displays the current working directory, showing your current location in the file system. |
| <code>ls</code> | Lists files and directories in the current directory. Can be used with options to modify the output, such as <code>-R</code> for recursive listing. |
| <code>cat ></code> | Creates a new file or overwrites an existing file with the input provided. Used to input data into a file from the terminal. |
| <code>sort</code> | Sorts the lines of text in a file or from input. Can be used with options to control sorting behavior, like sorting by fields or numerically. |
| <code>find</code> | Searches for files and directories within a specified location. Often used in combination with other commands. |
| <code>grep</code> | Searches for text patterns or regular expressions within files. Used to extract or filter lines that match a specified pattern. |
| <code>cut</code> | Extracts specific fields or columns from lines of text based on a delimiter. The <code>-d</code> option specifies the delimiter, and <code>-f</code> specifies the field. |
| <code>sed</code> | Processes and transforms text. Mainly used to replace text in a file with different text based on patterns or expressions. |
| <code>realpath</code> | Display the full, absolute path of a file or directory. |