

# Working with Commands

---

## Note

All labs rely on previous courseware and lab information.

## 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

## Duration

---

This lab requires approximately **30 minutes** to complete.

## AWS service restrictions

---

In this lab environment, access to AWS services and service actions might be restricted to the ones that you need to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that this lab describes.

## 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 along side 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. The following instructions vary slightly depending on whether you are using Windows or Mac/Linux.

### Windows Users: Using SSH to Connect

These instructions are specifically for Windows users. If you are using macOS or Linux, [skip to the next section](#).

5. Select the **Details** drop-down menu above these instructions you are currently reading, and then select **Show**. A Credentials window will be presented.
6. Select the **Download PPK** button and save the **labsuser.ppk** file.  
*Typically your browser will save it to the Downloads directory.*
7. Make a note of the **PublicIP** address.
8. Then exit the Details panel by selecting the **X**.
9. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer, [download it here](#).
10. Open **putty.exe**
11. Configure PuTTY timeout to keep the PuTTY session open for a longer period of time.:
  - Select **Connection**
  - Set **Seconds between keepalives** to **30**
12. 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**
  - Select **Auth** (*don't expand it*)
  - Select **Browse**
  - Browse to and select the lab#.ppk file that you downloaded
  - Select **Open** to select it
  - Select **Open** again.
13. Select **Yes**, to trust and connect to the host.
14. When prompted **login as**, enter: **ec2-user**  
This will connect you to the EC2 instance.

15. Windows Users: [Select here to skip ahead to the next task.](#)

## macOS and Linux Users

These instructions are specifically for Mac/Linux users. If you are a Windows user, [skip ahead to the next task.](#)

16. Select the **Details** drop-down menu above these instructions you are currently reading, and then select **Show**. A Credentials window will be presented.
17. Select the **Download PEM** button and save the **labsuser.pem** file.
18. Make a note of the **PublicIP** address.
19. Then exit the Details panel by selecting the **X**.
20. Open a terminal window, and change directory `cd` to the directory where the *labsuser.pem* file was downloaded. For example, if the *labsuser.pem* file was saved to your Downloads directory, run this command:

```
cd ~/Downloads
```

21. Change the permissions on the key to be read-only, by running this command:

```
chmod 400 labsuser.pem
```

22. Run the below command (*replace **<public-ip>** with the **PublicIP** address you copied 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.:

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

23. Type `yes` when prompted to allow the first connection to this remote SSH server. Because you are using a key pair for authentication, you will not be prompted for a password.

## 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.
25. From your current location in the terminal, enter `hostname | tee file1.txt` and press Enter.

```
[ec2-user@ip-10-0-10-81 ~]$ hostname | tee file1.txt
ip-10-0-10-81.us-west-2.compute.internal
```

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
```

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

```
[ec2-user@ip-10-0-10-81 ~]$ ls
companyA  file1.txt
```

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

```
[ec2-user@ ~]$ ls
companyA 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.

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

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

```
[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.

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-10-0-10-81 ~]$ sort test.csv
Factory, 1, Paris
Factory, 3, Brasillia
Factory, 5, Tokyo
Store, 2, Dubai
Store, 4, Algiers
```

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-10-0-10-81 ~]$ find | grep Paris test.csv
Factory, 1, Paris
```

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

```
[ec2-user@ ~]$ find | grep Paris test.csv
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.

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

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

```
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-10-0-10-81 ~]$ 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.

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

**Expected output:**

```
Dallas. Texas
Seattle. Washington
Los Angeles. California
Atlanta. Georgia
New York. New York
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  at the top of this page and then select  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.

## About the AWS component

Amazon EC2 provides a wide selection of *instance types* optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more *instance sizes* so that you can scale your resources to the requirements of your target workload.

This lab uses a **t3.micro** instance, which should be selected by default. This instance type has 1 virtual CPU and 1 GiB of memory.

## Additional resources

---

- [Amazon EC2 Instance Types](#)
- [Amazon Machine Images \(AMI\)](#)
- [Status Checks for Your Instances](#)
- [Amazon EC2 Service Quotas](#)
- [Terminate Your Instance](#)

For more information about AWS Training and Certification, see <https://aws.amazon.com/training/>.

*Your feedback is welcome and appreciated.*

If you would like to share any suggestions or corrections, please provide the details in our [AWS Training and Certification Contact Form](#).

*© 2022 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.*