Editing Files

> **Note**
>
> All labs rely on previous courseware and lab information.

Objectives

After completing this lab, you will be able to:

- Use the vimtutor executable to conduct tasks 1-4
- Copy content from the /var/log/secure file, and edit it with nano

Duration

This lab requires approximately **1 hour** to complete.

AWS service restrictions

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

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.

   This lab launches an Amazon Elastic Compute Cloud (Amazon EC2) instance that's named *Command Host*. You will connect to this instance to run Linux commands.

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 your PuTTY session by following the directions in the following link: Connect to your Linux instance using PuTTY

12. 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.

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

    ```
    cd ~/Downloads
    ```

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

    ```
    chmod 400 labsuser.pem
    ```

19. 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>
    ```

20. 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: Exercise - run the Vim tutorial

In this exercise, you run the ./vimtutor and follow all directions in the file for tasks 1-4. Vimtutor is an application that teaches you the basics of how to use Vim, which is one of the text editors for Linux.

21. From your current location in the terminal, enter `vimtutor` and press Enter. This step starts the vimtutor session:

    ```
    vimtutor
    ```

    > **Note**: You may have to use sudo to conduct this step if you are not root. If vimtutor does not work, you may need to install Vim by entering the following command:

    ```
    sudo yum install vim
    ```

    Note: You may have to use sudo to conduct this step if you are not root. If vimtutor does not work, you may need to install Vim using sudo yum install vim.

22. Complete lessons 1-3 in vimtutor.

```
![The terminal window displays output from the command vimtutor.  The output from this command displays tutorial lesson 1.1: moving the cursor from vimtutor](images/vimtutor.jpg)

*Figure: Vimtutor consists of tutorials that teaches a user how to use Vim.*
```

23. Enter `:q!` and press Enter to exit vimtutor.

Task 3: Exercise - edit a file in Vim

In this exercise, you use the Vim command-line editor program. Use Vim to create and edit a file using the following steps.

24. From your current location in the terminal, enter `vim helloworld` and press Enter. You are using Vim to create a file called **helloworld**, and you open this file when you press Enter.

```
![The terminal window displays output from the command vim helloworld. ](images/vimhello.jpg)

*Figure: The command vim followed by the file name, in this example the file is named helloworld.*
```

25. Now that you are in the file that you created called **helloworld**, use Vim to insert a few lines of text. Enter `i` to use insert mode, and enter the following text:

    ```
    Hello World!
    This is my first file in Linux and I am editing it in Vim!
    ```

    > **Note**: The bottom left of the terminal indicates if you are in insert mode.

    Once complete, press ESC to exit insert mode.

26. Save your changes to the file, and enter the following command to quit:

    ```
    :wq
    ```

    Next, type of the following into the command prompt:

    ```
    vim helloworld
    ```

    > **Note**: Because the **vim helloworld** command was the last command that you used, you can use the up arrow to recall the last command and press Enter.

27. Add the following line to the editor:

    ```
    I learned how to create a file, edit and save them too!
    ```

    It should look like the following picture:

```
![The terminal displays a few lines, the first line is text entered in the previously saved vim file. The second line displays the following: I learned how to create a file, edit and save them too!](images/vimSecondLine.jpg)

*Figure: The example displays the second line that has been placed in the terminal.*
```

28. Once complete, press ESC to exit insert mode and use the following command:

    ```
    :q!
    ```

29. You are back in the main terminal. Use Vim to go back to the **helloworld** file and analyze what happened. What was the difference?

Additional Challenge

Try additional useful commands.

30. Use the following command to delete the entire line:

    ```
    dd
    ```

31. Use the following command to undo the last command:

    ```
    u
    ```

32. Use the following command to save changes without quitting:

    ```
    :w
    ```

Task 4: Exercise - edit a file in nano

In this exercise, you use an alternative command-line editor program called nano. Use nano to create and edit a text file.

33. Similar to Vim, in the main terminal, enter `nano cloudworld` and press Enter. You are using nano to create a file called **cloudworld**, and pressing Enter opens this file.

```
![The terminal window displays the command nano followed by the word cloudworld.](images/nano.jpg)

*Figure: The command nano followed by the file name, in this example the file is named cloudworld.*
```

34. Now that you are in the file that you created called **cloudworld**, unlike vim, you do not have to enter insert mode. Instead, you can start typing. Enter the following text:

    ```
    We are using nano this time! We can simply start typing! No insert mode needed.
    ```

35. To save your changes to the file, press **CTRL+O**. Press Enter to confirm the file name once you save it.
36. Now that you have saved the file, press **CTRL+X** to exit the nano editor.
37. Now that you are at the main terminal, check to make sure our file saved correctly. Enter `nano cloudworld` to go back into the file using nano. Confirm that everything is correct as the following image shows, and exit the editor:

```
![The terminal window displays the command CTRL +X which is used to end nano.](images/nanoEnd.jpg)

*Figure: The keys CTRL + X is used to will end the nano editor.*
```

Lab Complete 🎓

🏁 Congratulations! You have completed the lab.

38. 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."
39. 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*, allowing you to scale your resources to the requirements of your target workload.

You will use 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 Limits
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.