re/start FORGE FUNDACIÓN

Lab 231

Editing Files

Student: Mane Zakarian
Bootcamp: Forge AWS re/Start UYMON5
Date: 2023

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

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**.

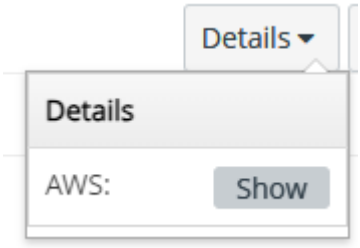
4. Arrange the AWS Management Console tab so that it displays alongside these instructions. Ideally, you will be able to see both browser tabs at the same time so that you can follow the lab steps more easily.

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

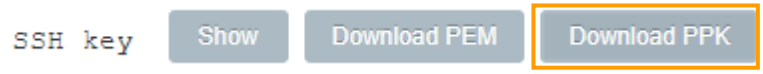
In this task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations.

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.



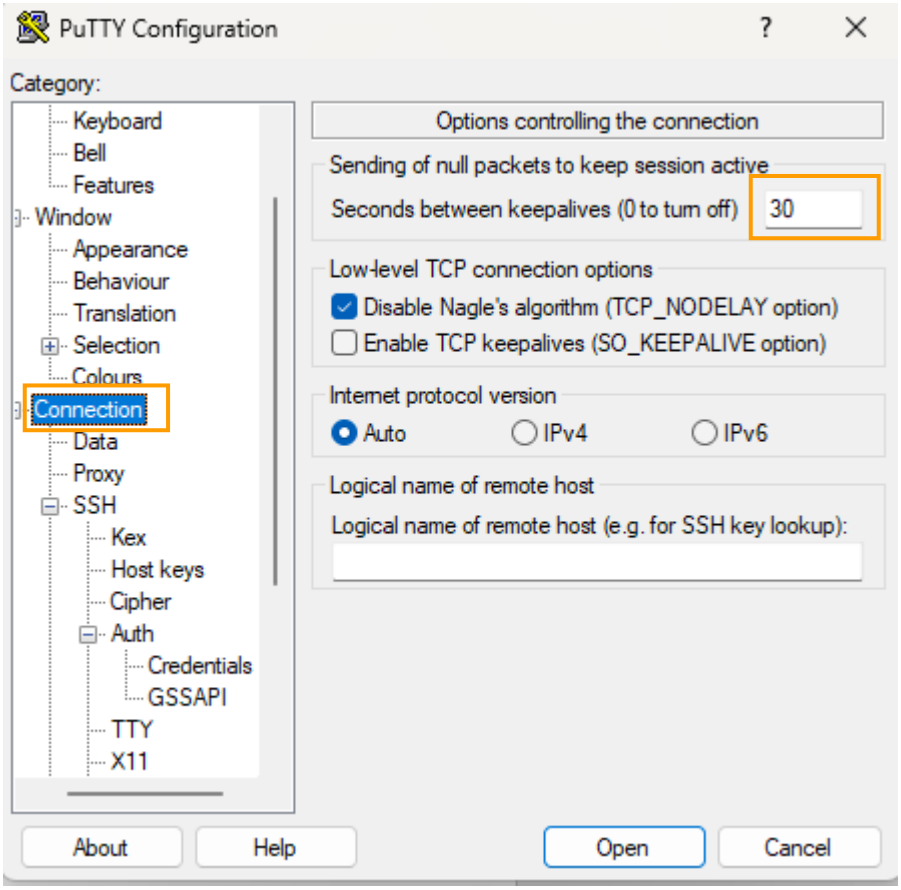
2. Select the **Download PPK** button and save the **labsuser.ppk** file.



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

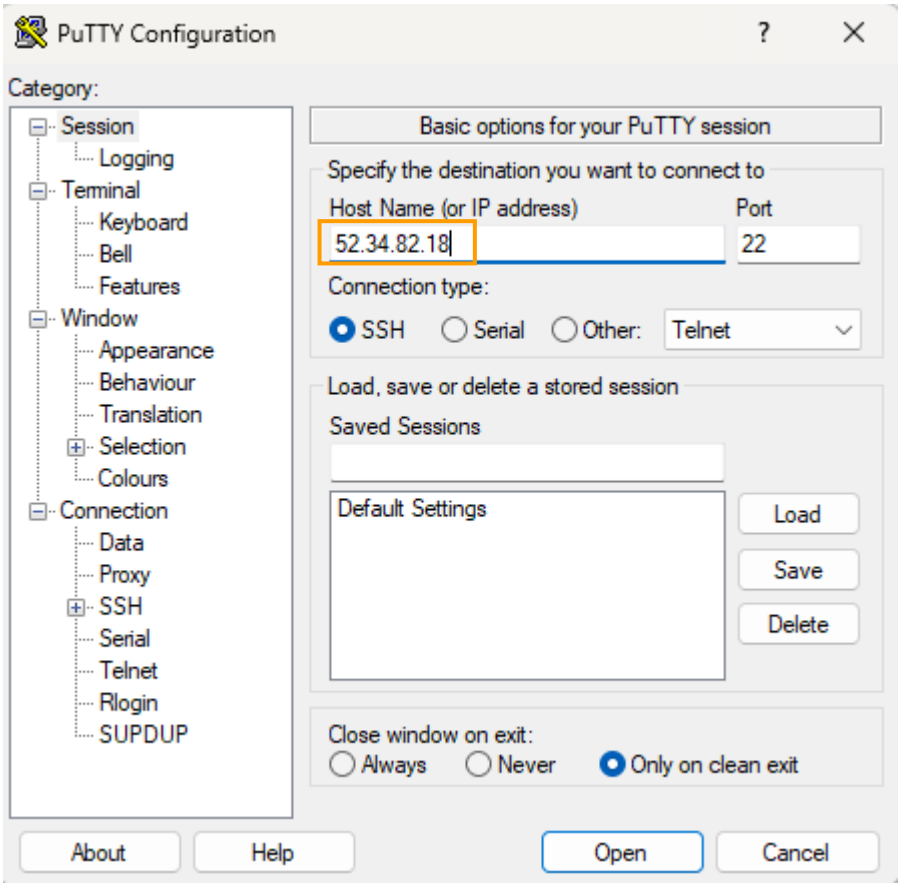
4. Then exit the Details panel by selecting the **X**.
5. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer.
6. Open **putty.exe**
7. Configure PuTTY timeout to keep the PuTTY session open for a longer period of time.:

- Select **Connection**
- Set **Seconds between keepalives** to `30`

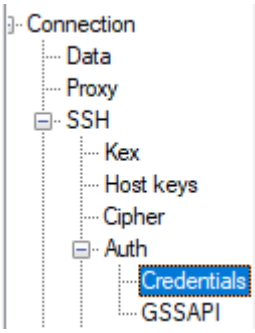


8. Configure your PuTTY session:
   - Select **Session**

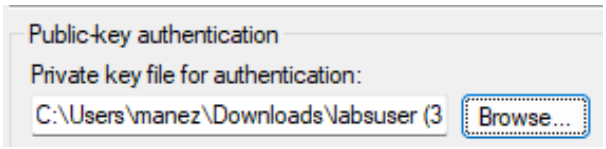
- **Host Name (or IP address):** Paste the **Public DNS or IPv4 address** of the instance you made a note of earlier. Alternatively, return to the EC2 Console and select **Instances**. Check the box next to the instance you want to connect to and in the *Description* tab copy the **IPv4 Public IP** value



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

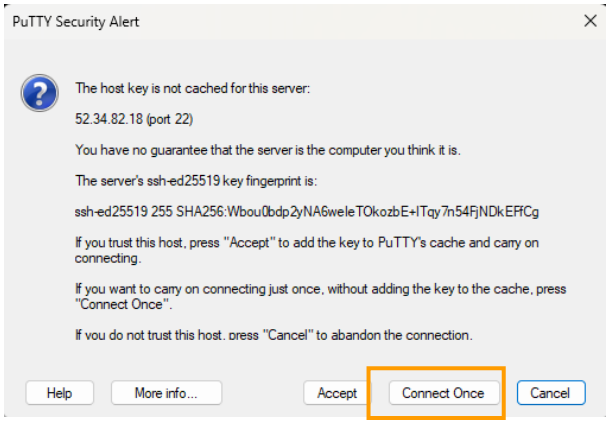


- Select **Browse** and select the lab#.ppk file that you downloaded

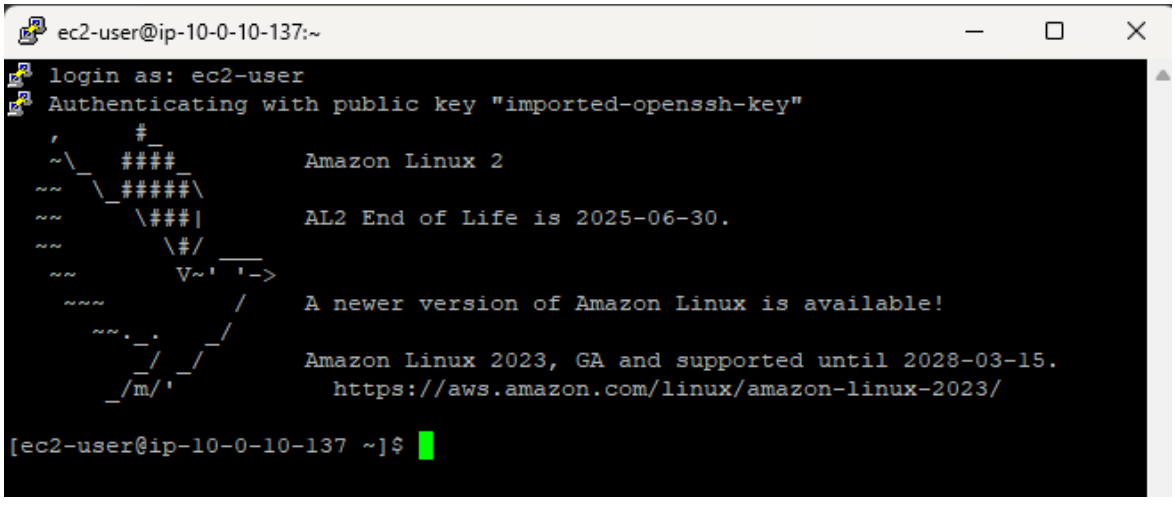


- Select **Open** to select it and then select **Open** again.

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: 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.

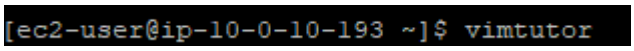
24. From your current location in the terminal, enter `vimtutor` and press Enter. This step starts the vimtutor session:

```
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.



```
===============================================================================
=   W e l c o m e   t o   t h e   V I M   T u t o r   -   Version 1.7       =
===============================================================================

   Vim is a very powerful editor that has many commands, too many to
   explain in a tutor such as this.  This tutor is designed to describe
   enough of the commands that you will be able to easily use Vim as
   an all-purpose editor.

   The approximate time required to complete the tutor is 30 minutes,
   depending upon how much time is spent with experimentation.

   ATTENTION:
   The commands in the lessons will modify the text.  Make a copy of this
   file to practice on (if you started "vimtutor" this is already a copy).

   It is important to remember that this tutor is set up to teach by
   use.  That means that you need to execute the commands to learn them
   properly.  If you only read the text, you will forget the commands!

   Now, make sure that your Caps-Lock key is NOT depressed and press
   the   j   key enough times to move the cursor so that lesson 1.1
   completely fills the screen.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                     Lesson 1.1:  MOVING THE CURSOR


 ** To move the cursor, press the h,j,k,l keys as indicated. **
           ^
           k               Hint:  The h key is at the left and moves left.
      < h       l >               The l key is at the right and moves right.
           j                      The j key looks like a down arrow.
           v
1. Move the cursor around the screen until you are comfortable.

2. Hold down the down key (j) until it repeats.
   Now you know how to move to the next lesson.

3. Using the down key, move to lesson 1.2.

TE: If you are ever unsure about something you typed, press <ESC> to place
    you in Normal mode.  Then retype the command you wanted.
```

25. Complete lessons 1-3 in vimtutor.

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

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

```
[ec2-user@ip-10-0-10-193 ~]$ vimtutor
[ec2-user@ip-10-0-10-193 ~]$
```

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.

27. 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.

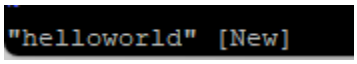


Figure: The command vim followed by the file name, in this example the file is named helloworld.

28. 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.

29. 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.

30. 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:

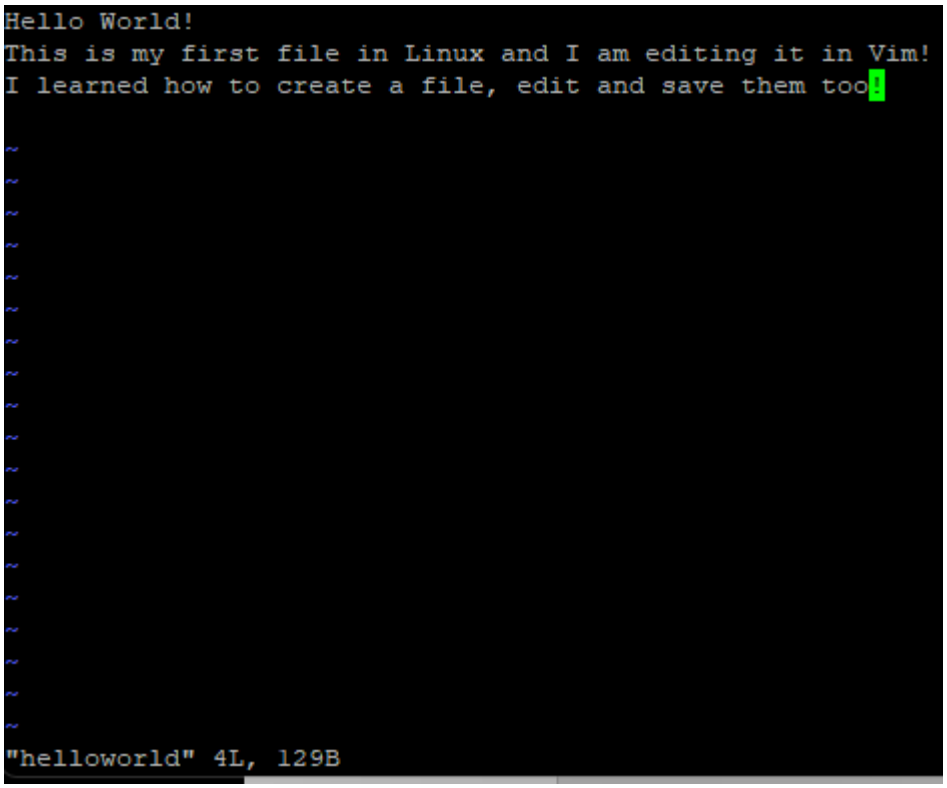
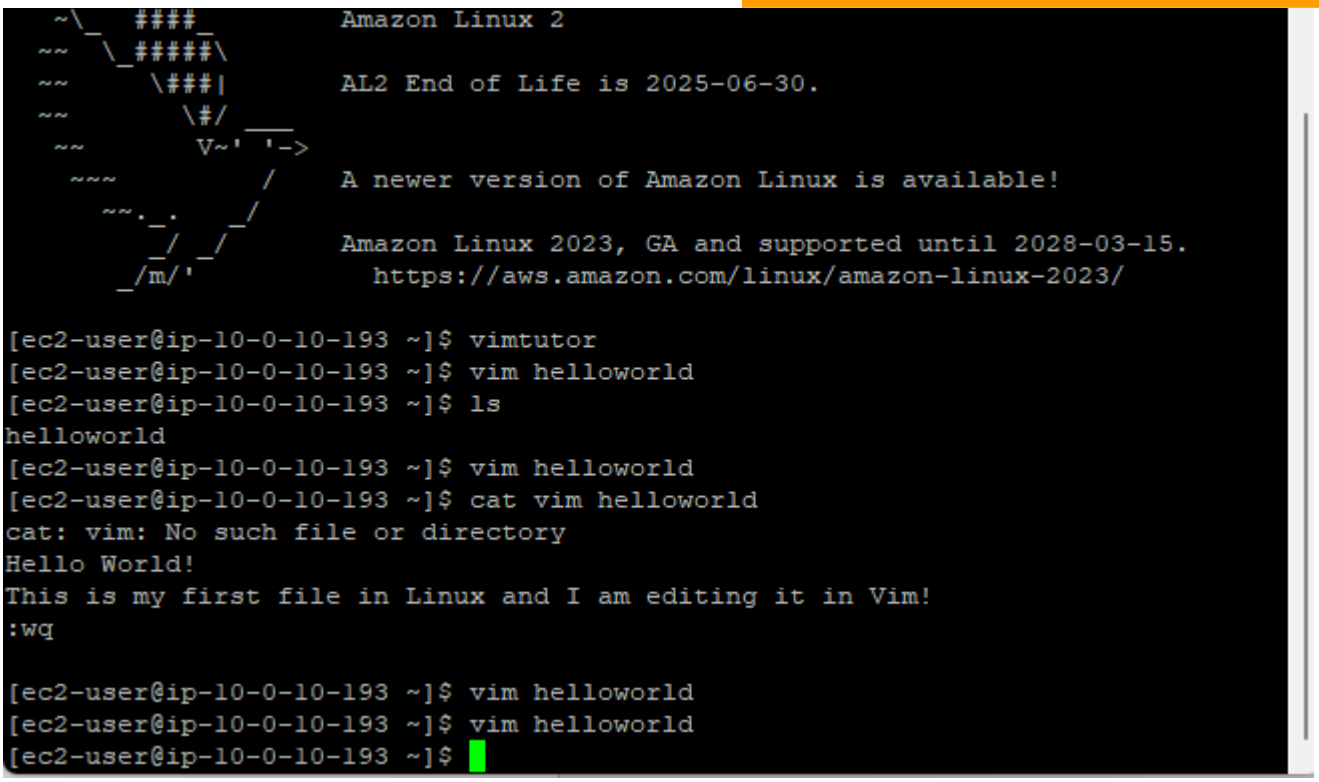


Figure: The example displays the second line that has been placed in the terminal.

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

```
:q!
```

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

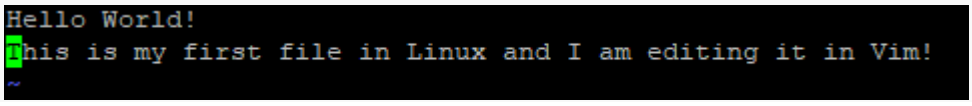


Additional Challenge

Try additional useful commands.

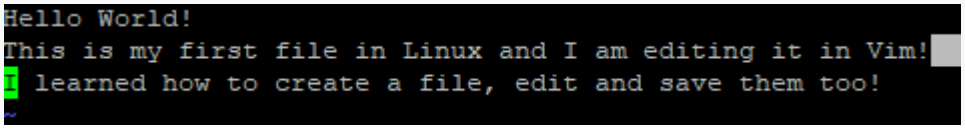
33. Use the following command to delete the entire line:

```
dd
```



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

```
u
```



35. 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.

36. 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.

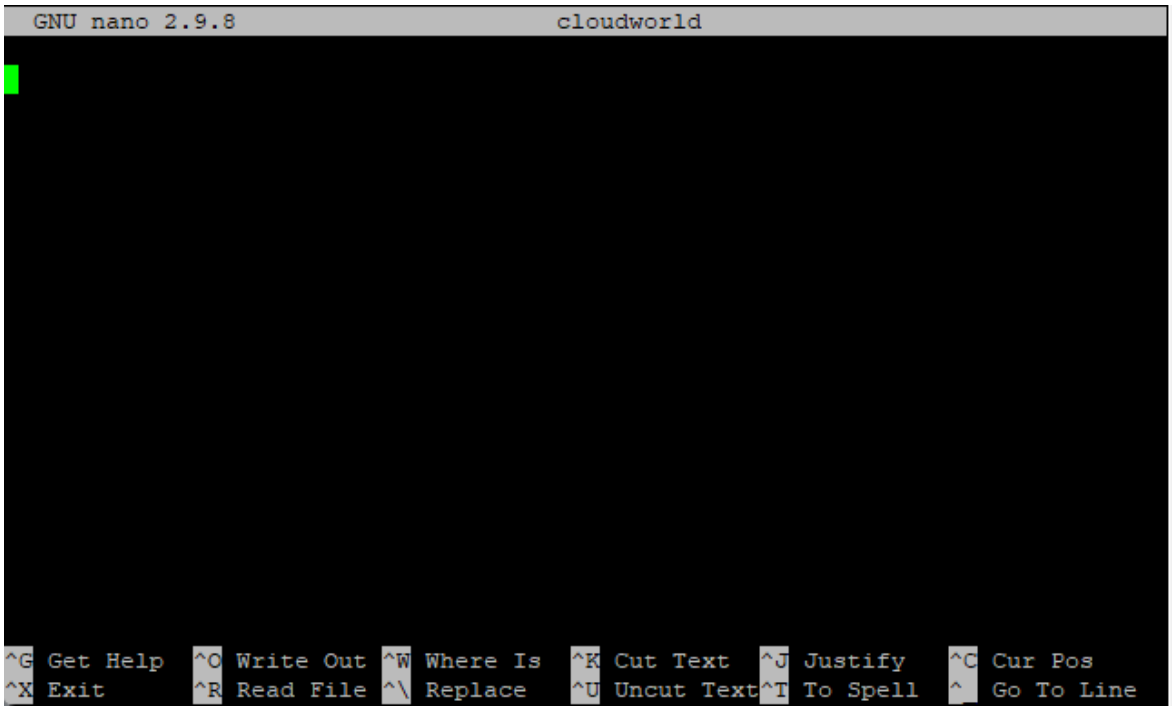
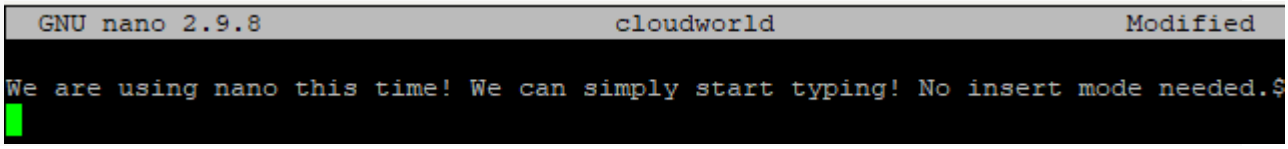


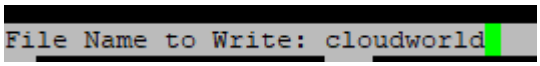
Figure: The command nano followed by the file name, in this example the file is named cloudworld.

37. 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.
```

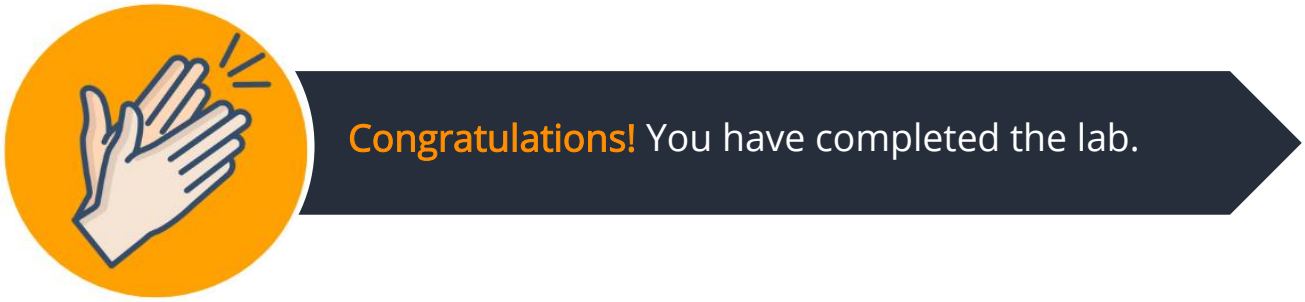


38. To save your changes to the file, press **CTRL+O**. Press Enter to confirm the file name once you save it.



39. Now that you have saved the file, press **CTRL+X** to exit the nano editor.
40. 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:

Lab Complete



41. Choose **End Lab** at the top of this page, and then select Yes to confirm that you want to end the lab.
42. 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
vimtutor	Starts the Vim tutorial.
i	Enters insert mode in Vim.
ESC	Exits insert mode in Vim.
:wq	Saves changes and quits Vim.
:q!	Quits Vim without saving changes.
CTRL+O	Saves changes in Nano.
CTRL+X	Exits the Nano editor.