Lab 1: Introduction to AWS Cloud9

Errors or corrections? Email us at aws-course-feedback@amazon.com

Other questions? Contact us at https://aws.amazon.com/contact-us/aws-training

---

Introduction to AWS Cloud9

Overview

AWS Cloud9 is a web-based integrated development environment (IDE) that contains a collection of tools that you use to code, build, run, test, debug, and release software in the cloud. In this lab, you will open a preconfigured AWS Cloud9 development environment and create a `Hello World` program in Node.js, Python, or Ruby.

Objectives

After completing this lab, you will be able to:

- Use the AWS Cloud9 development environment to create, edit, and run code files.

Duration

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

Accessing the AWS Management Console

1. At the top of these instructions, click Start Lab to launch your lab.
2. A Start Lab panel opens displaying the lab status.
3. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.
4. At the top of these instructions, click AWS

This will open 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, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

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, to make it easier to follow the lab steps.

Task 1: Connect to and Prepare Your AWS Cloud9 Development Environment

An AWS Cloud9 environment has been created for you to use in this lab. The AWS Cloud9 integrated development environment (IDE) provides you with a console to work from and filesystem to use.

An Amazon Elastic Compute Cloud (Amazon EC2) instance is the driving force behind this AWS Cloud9 environment. However, this underlying system is hidden from you behind the AWS Cloud9 console, allowing you to focus on interacting with your code and command line, and the respective AWS resources, for example, Amazon Simple Storage Service (Amazon S3).

To connect to your AWS Cloud9 development environment:

5. From the AWS Management console, from the **Services** list, choose **Cloud9**.
6. To open the AWS Cloud9 environment you are provided, choose **Open IDE**.
7. To seed your AWS Cloud9 filesystem, go to the AWS Cloud9 bash terminal (at the bottom of the page) and run the following `wget` command:

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/DEV-ILT-TF-200-ACCDEV-1/lab-1-cloud9.zip -P /home/ec2-user/environment
```

8. To unzip the *lab-1-cloud9.zip* file, run the following command:

```
unzip lab-1-cloud9.zip
```

This process might take a few moments.

9. To clean up your environment, remove the .zip and README files by running the following commands:

```
rm *.zip
rm README.md
```

Task 2: Explore Your AWS Cloud9 Development Environment

10. Take a few minutes to explore the AWS Cloud9 development environment before you continue to the next step.

Notice that the AWS Cloud9 filesystem (at the left side of the IDE) contains the following folders under the **cloud9-start** root folder:

- **Node_8.10.0**
- **Python_3.6.8**
- **Ruby_2.6.0**

You don't need to do anything with these folders at this time. For now, be aware that they are available.

11. To see the full pathname of your current working directory, in the AWS Cloud9 terminal, run the following *print working directory* command:

```
pwd
```

You should see the following result:

```
/home/ec2-user/environment
```

12. To check that you can access your AWS resources from AWS Cloud9, run the following command from the AWS Cloud9 terminal:

```
aws s3 ls
```

You should see a list of your S3 buckets, *similar* to this example:

Congratulations! You have completed this task. You now know how to issue command-line requests and interact with AWS resources form the AWS Cloud9 terminal.

Task 3: Create a Text File

In this task, you will create a text file and use the AWS Cloud9 editor to add some text to it.

13. From the **File** menu, choose **New File**.
14. From the **File** menu, choose **Save as**.
15. In the **Filename** box, enter `hello_world.txt`.
16. Choose **Save**.

You should see the new file that you created in your AWS Cloud9 filesystem.

17. If it is not already open, open (double-click) your new `hello_world.txt` file.
18. Add the following sample text to the file: `Hello world!`.
19. **Save** the file.

Congratulations! You have completed this task. You now know how to create and edit files in AWS Cloud9.

Task 4: Run Solution Code that Displays *Hello World!*

In most of the AWS Academy Cloud Developing labs, you can choose a programming language to work in. These labs are available in **Node.js** (version 8.10.0), **Python** (version 3.6.8), and **Ruby** (version 2.6.0).

20. Decide which language you will use.

Recall the three folders in the AWS Cloud9 filesystem that you saw earlier.

21. In the **Environment** window, select the folder for the language that you will use to expand it.

    You should now see a `helloworld` file and a `solution` folder.

22. Select the `solution` folder to expand it, and then double-click the `helloworld` file that is inside this folder.

    You should have just opened the `solution` version of the `helloworld` file.

23. Take a few minutes to read the code in the `helloworld` file to understand what the code will do.

    When the code runs, it will print `hello <yourname>`.

Now, you will run the solution `helloworld` file.

24. To set the AWS Cloud9 terminal path to the folder for your language of choice, in the following command replace the characters `<FMI>` (which stand for `<Fill Me In>`) with the folder name for your language, and run the command:

```
cd <FMI>
```

TIP: Your choices for language folder are: `node_8.10.0`, `python_3.6.8`, and `ruby_2.6.0`

For example, if you choose to use Node.js, then you would set the terminal path to the Node.js folder by running the following command:

```
cd node_8.10.0
```

Now, you will run the solution file.

Each language has its own preferred way of being run. In the AWS Cloud9 terminal, you must use the run command for the respective language to run the `helloworld` solution file.

25. From the AWS Cloud9 terminal, run the solution file by using the run command for the language you chose, which is listed in the following table:

TIP: The run command requires you to pass in a variable, that is, your name. Be sure to replace `<FMI>` with your name.

Lanauage	Command
Node_8.10.0	`node solution/helloworld.js <FMI>`
Python_3.6.8	`python3 solution/helloworld.py <FMI>`
Ruby_2.6.0	`ruby solution/helloworld.rb <FMI>`

NOTE: Because you are running the solution version of the file (and not your own code yet), you must use the *full path* from your folder.

For example, if you are using Node.js and your name is Alex, you would already be in the `node_8.10.0` path. You would run a command like the one in the table above in the AWS Cloud9 terminal:

```
node solution/helloworld.js Alex
```

You would see output similar to this example:

Congratulations! You have completed this task. You now know how to run a file in your preferred coding language by targeting files in your filesystem, and are also able to pass a `variable` into the program at runtime.

Challenge: Go multilingual!

Try running the solution `helloworld` file from another code solution file. You will see that although the syntax is different across lanaguages, the concepts are similar.

For example, if you just ran the solution file for Node.js, try running the solution file for Ruby or Python. Be sure to navigate to the correct language folder inside the environment folder, and replace `<FMI>` with your name, as in the following example:

```
ruby ~/environment/ruby_2.6.0/solution/helloworld.rb <FMI>
```

Task 5: Edit and Run the `helloworld` Code File

In this task, you will use the AWS Cloud9 editor to edit a code file to display the text *Hello World!*

26. First, close any `helloworld` solution code tabs that are open from the previous task.
27. Open (double-click) the `helloworld` file that is inside your respective code folder. (**NOTE**: *Do not* open the `helloworld` file that is in the solution folder this time.)
28. Read the contents inside this file.

You will notice some differences between this file and the solution file you looked at and ran before.

Namley, you will see `<FMI>`s in the code.

29. Do your best to replace the `<FMI>` section with the missing code.

TIP: If you can't complete this step, refer to the respective `helloworld` solution code you ran last time. You can easily just copy the solution code and paste it into your task code, then run it.

30. After you have modified the code inside your `helloworld` file, save your changes.
31. Navigate to the correct folder for the language that you chose. To do this, in the following command replace `<FMI>` with the language you chose, and run the command in the AWS Cloud9 terminal:

```
cd ~/environment/<FMI>
```

For example, if you want to use Node.js, you would run the following command:

```
cd ~/environment/node_8.10.0
```

32. In the AWS Cloud9 terminal, run the `helloworld` file that you just edited by using the run command for the language you chose, which is listed in the following table. Replace the `<FMI>` with your name.

Lanauage	Command
Node_8.10.0	`node helloworld.js <FMI>`
Python_3.6.8	`python3 helloworld.py <FMI>`
Ruby_2.6.0	`ruby helloworld.rb <FMI>`

Confirm That Your Code Works

33. Verify that you get the same result as the result from when you ran the solution code.

You should see output a bit like the image below:

Congratulations! You have completed this task and have also completed the first lab. You have edited code files, replaced `<FMI>` sections in both the terminal and in the code, referred to solution files (if needed), navigated to the files you wanted to run, ran a file in the language of your choosing, and observed the outputs.

The next labs in this course will require you to do similar tasks. These labs will be more challenging, and you will replace more `<FMI>` sections with your code.

However, future labs have links to the AWS documentation to help you with your code. Remember that each code file that you must edit has a solution. You can copy the solution if you can't complete the code challenges.

See you in the next lab!

Lab Complete

Congratulations! You have completed the lab.

- Click End Lab at the top of this page and then click 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."
- Click the **X** in the top right corner to close the panel.

For feedback, suggestions, or corrections, please email us at: aws-course-feedback@amazon.com