Lab 1 - Ready, Set, Go!

Friday, December 25, 2020 10:37 AM

Developing on AWS - Lab 1 - Ready, Set, Go!

2 hours Free Rate Lab

aws training and certification

.Net version

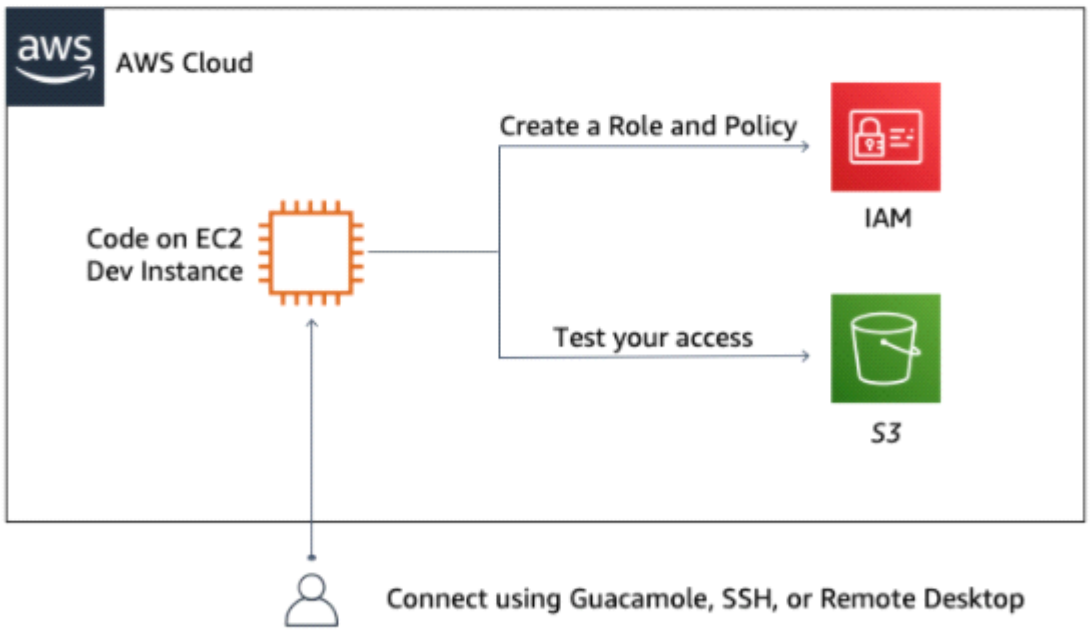
Corrections, feedback, or other questions? Contact us at *AWS Training and Certification*.

Overview

In this lab, you will get started by connecting to your Development environment. You work with a Windows EC2 instance for your lab.

After connecting, you will check your AWS developer credentials and verify your setup by running a basic application.

Finally, you will use AWS Identity and Access Management (IAM) to see the steps required to create an IAM role and policy and assign an existing policy to an existing role.



Objectives

After completing this lab, you will be able to:

- Set up your AWS developer credentials.
- Describe key aspects related to using the AWS SDK.
- Create an IAM policy.
- Create an IAM role.

- Create an IAM policy.
- Create an IAM role.
- Assign a policy to a role.

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi running Microsoft Windows or macOS.
- An Internet browser such as Chrome, Firefox, or IE9+ (previous versions of Internet Explorer are not supported).
- You will need a Microsoft Remote Desktop client to connect to your development EC2 instance.

Note
You can use an iPad or tablet device to access these directions in the lab console.

Duration

This lab will require around **30 minutes** to complete.

Start Lab

1. At the top of your screen, launch your lab by choosing **Start Lab**

   This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

for your resources to be provisioned before continuing.

ℹ If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing Open Console

   This opens an AWS Management Console sign-in page.

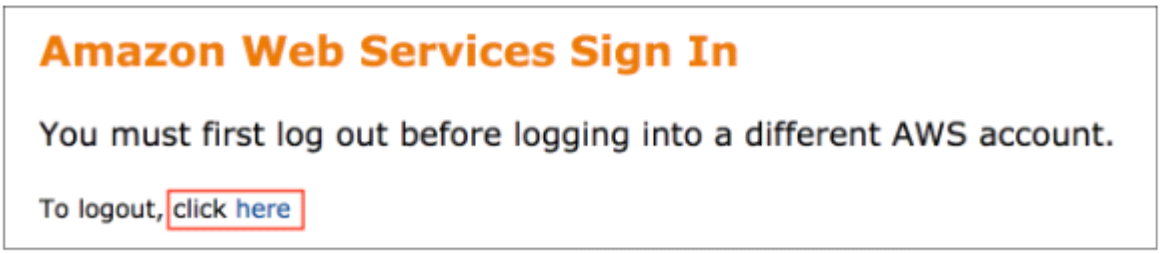
3. On the sign-in page, configure:

   - **IAM user name:** `awsstudent`
   - **Password:** Paste the value of **Password** from the left side of the lab page
   - Choose **Sign In**

⚠ **Do not change the Region unless instructed.**

Common Login Errors

Error: You must first log out



If you see the message, **You must first log out before logging into a different AWS account:**

- Choose **click here**
- Close your browser tab to return to your initial lab window
- Choose Open Console again

Task 1: Connecting to Your Development Environment

4. To connect to your Dev instance, see the Appendix: Connecting to Your Development Environment.

Task 2: Checking Your AWS Credentials

In this section, you will verify the AWS developer credentials that have already been set up on the EC2 instances.

When lab instructions in subsequent sections require a command window, open or use a PowerShell window.

Task 2.1: Check AWS Developer Credentials

In this section, you will check your AWS developer credentials on Windows.

5. View the `%UserProfile%\.aws\credentials` file in a text editor. On the **Windows Dev Instance** you can find this file by browsing to the `C:\Users\Administrator\.aws` folder in Windows Explorer. You can open this file with the **Notepad** application.

this file with the **Notepad** application.

6. The credentials file has the following information:

```
[default]
aws_access_key_id = <your access key id>
aws_secret_access_key = <your secret access key>
```

   You can also find your `aws_access_key_id` and `aws_secret_access_key` to the left of these instructions.

7. **Optional** - Review the following information on credentials in your AWS SDK for .NET applications. Otherwise, continue to Task 3: Running an Application.

   You can manage credentials for your AWS SDK for .NET application in the following ways:

   - Using the SDK Store
   - Using a Credentials File (`C:\Users\<username>\.aws\credentials`)

   For more information, see AWS Access Keys Best Practices.

Task 3: Running an Application

In this section, you will run an application called ReadySetGo to verify that your AWS developer credentials have been set up correctly.

Lab Skeleton Code

Your lab skeleton code has been set up on the **Windows Dev Instance**.

Your lab skeleton code has been set up on the **Windows Dev instance**.

The base working directory located at: `c:\temp\workdir`

8. In Windows Explorer, navigate to `c:\temp\workdir\readySetGoCSharpLab`.

9. Open the *ReadySetGoSolution* file and select **Visual Studio 2017** to launch the lab code.

10. Step through the prompts to setup *Visual Studio* for first time use by selecting **Not now, maybe later.** and **Start Visual Studio**.

11. Build the solution via **Build->Build Solution** or **Ctrl+Shift+B** and then run it via **Debug->Start Without Debugging** or **Ctrl+F5**. The NuGet packages haven't been downloaded yet, so it is fine for the code to show errors until you build the solution for the first time by performing this step.

12. Check the command prompt output to see the result of running the code.

    You should see a welcome message and the *number of S3 buckets in your AWS account* which should be **2**.
    This confirms that your developer credentials have been set up correctly.

    In case of errors, correct the setup and run this command again.

Task 4: Create an IAM Policy and Role

In this section, you step through creating a policy and assigning it to an IAM role.

You can use **IAM Roles** to delegate access to your AWS resources. A role is

You can use **IAM Roles** to delegate access to your AWS resources. A role is similar to a user, in that it is given permissions to call AWS services. However, instead of belonging to one person, a role can be assumed by an authorized user, service or application. When this is done, access credentials are dynamically created, which can then be used to make API calls to AWS services. This is how applications running on an Amazon EC2 instance can obtain credentials to call AWS services such as Amazon S3 and Amazon DynamoDB.

Task 4.1: Test your user access to S3 buckets

13. To list all of the buckets you have access to, run the following command:

```
aws s3 ls
```

    You should see a list of S3 buckets.

14. Choose **Services** and select **IAM**.

15. In the left navigation pane, select **Users**.

16. Choose **awsstudent**.

17. Choose the **Security credentials** tab.

18. Scroll down to the **Access keys** section.

    For the **Access key ID** listed, you can see the status is **Active**.

19. Select **Make inactive**.

    Now you can see the status is *Inactive*.

20. Wait a few seconds and then go back to the Dev Instance PowerShell

Now you can see the status is *Inactive*.

20. Wait a few seconds and then go back to the Dev Instance PowerShell window and run the following command:

    ```
    aws s3 ls
    ```

    You will see the *InvalidAccessKeyId* message.

Task 4.2: Create a Policy

In this section, you will see how to create a policy and a role, assign the policy to the role, and assign the role to our EC2 instance.

Note
You will walk through the steps to learn the process, but won't actually create the policy nor the role. However, you will attach the existing policy to the existing role, and assign the role to the EC2 instance.

21. In the left navigation pane, select **Policies**.

22. Choose **Create policy**.

23. For **Service**, select **Choose a service**.

24. Choose **S3**.

25. For **Actions**, select the check box next to **All S3 actions (s3:*)**

    This action gives you all access to list, read, write, and all permissions management levels.

26. Choose **Resources** and select **All resources**.

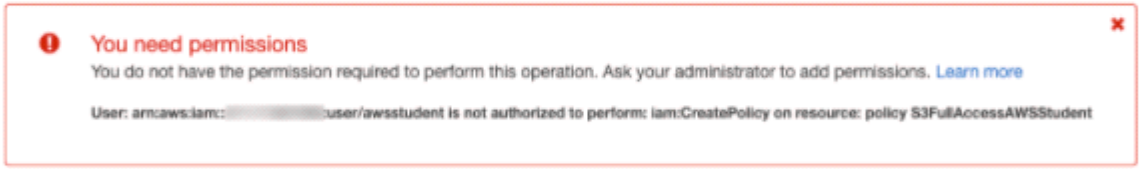
27. Choose **Review policy**.

27. Choose **Review policy**.

28. For **Name**, enter: `S3FullAccessAWSStudent`

29. Choose **Create policy**.

    You will encounter the following error.



    You walked you through these steps so you can learn how to create the policy from the IAM console, but the policy has already been created for you.

    From the bottom of the screen, select **Cancel**.

Task 4.3: Create a Role

30. In the left navigation pane, select **Roles**.

31. Choose **Create role**.

32. Under **Select type of trusted entity**, select **AWS service**, and then select **EC2**.

33. Choose **Next: permissions**.

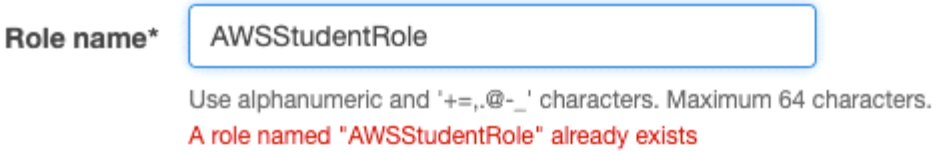
34. Type the name of the policy `S3FullAccessAWSStudent` and select the check box next to the policy name.

35. Choose **Next: tags**, then select **Next: Review**.

36. For **Role name**, enter: `AWSStudentRole`

    You will encounter the following error since the role was already created for

You will encounter the following error since the role was already created for you.



Select **Cancel** at the bottom of the screen.

37. You will be directed back to the **Roles** page. Next, attach the existing policy to the existing role. In the **Search** field, type `AWSStudentRole` and select the **AWSStudentRole** link in the results output.

38. Select the **Permissions** tab if not already selected. In the **Permissions policies** section, select **Attach policies**.

39. Type `S3FullAccessAWSStudent` in the search field.

40. Select the checkbox next to the results field reading `S3FullAccessAWSStudent` and select **Attach policy**.

    You should see a message reading, **Policy S3FullAccessAWSStudent has been attached for the AWSStudentRole**.

41. Choose **Services** and select **EC2**.

42. Choose **Instances**.

43. Select your Windows Dev Instance and choose **Actions**

44. Choose **Security** and then select **Modify IAM role**.

45. For **IAM role**, from the drop-down menu, choose **AWSStudentRole**.

46. Choose **Save**.

46. Choose **Save**.

Task 4.4: Test the configuration

47. You need to remove the Access Key ID and Secret Access Key from the EC2 instance for the awsstudent user. Go back to the Dev Instance and open a PowerShell window.

48. Run the following command:

```
rm ~/.aws/credentials
```

49. Without any hardcoded credentials on the instance anymore, the only option left for the SDK or CLI to find credentials is to look into the Instance Metadata where the temporary credentials for your role are. To see if applying the role gave you the credentials you expected, run the following command:

```
aws s3 ls
```

This time you should see the original list of buckets you saw before making the `awsstudent` access keys inactive.

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

experience.

50. Return to the AWS Management Console.

51. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.

52. Choose **End Lab**

53. Choose **OK**

54. (Optional):

    - Select the applicable number of stars ☆
    - Type a comment
    - Choose **Submit**

        - 1 star = Very dissatisfied
        - 2 stars = Dissatisfied
        - 3 stars = Neutral
        - 4 stars = Satisfied
        - 5 stars = Very satisfied

You may close the window if you don't want to provide feedback.
Congratulations! You are done!

Additional Resources

For more information about AWS Training and Certification, see *http://aws.amazon.com/training/*.

http://aws.amazon.com/training/.

Your feedback is welcome and appreciated.

If you would like to share any feedback, suggestions, or corrections, please provide the details in our *AWS Training and Certification Contact Form*.

Appendix: Connecting to Your Development Environment

In this section, you will sign in to an Amazon EC2 instance that has been pre-configured for you with various development tools. You will work with the Windows EC2 instance.

- On the Windows EC2 instance, you will have access to the Visual Studio 2017 IDE.

55. You can connect to your Dev instance by using one of the following methods:

    - Use Apache Guacamole to connect to your Windows Dev instance.
    - Use Remote Desktop to connect to your Windows Dev instance.

    To connect to the **Windows EC2 instance** by using Guacamole (Recommended), see the following directions:

    - Connect to Your Windows Dev Instance by Using Apache Guacamole

    To connect to the **Windows EC2 instance** by using RDP, see the following directions:

directions:

- Connect to Your Windows Dev Instance from a Windows Machine
- Connect to Your Windows Dev Instance from a macOS Machine

Connect to Your Windows Dev Instance by Using Apache Guacamole

56. In the **Connection Details** section, go to the bottom for the Guacamole information. Copy the **GuacamoleLink** and paste it into a browser.

57. Go back to the lab console and copy the **WindowsPassword** to the clipboard.

58. Go to the Apache Guacamole sign in in the browser. Sign in by using the following steps:

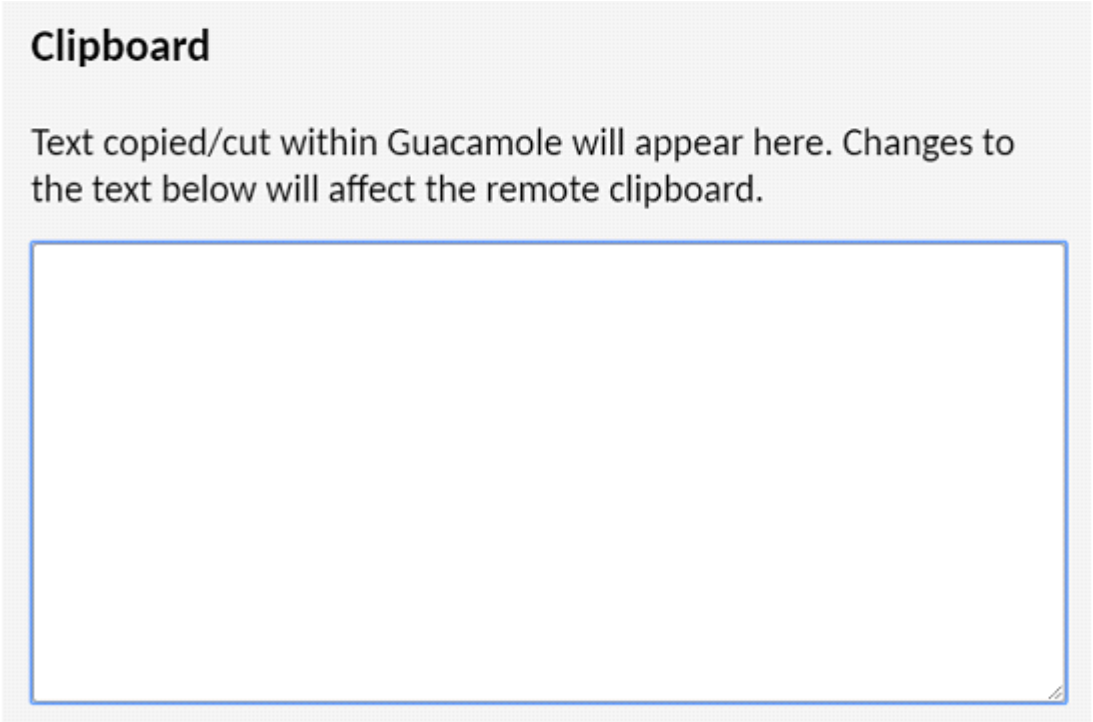
    - For **Username**, enter: `student`
    - For **Password**, paste the **WindowsPassword** from the clipboard.
    - Choose **Log In**.

    Your connection to your remote instance should start momentarily. Once you open a connection, you will see an image of the Dev instance desktop. You can interact with this image just as you would your normal desktop, or any remote desktop client.

    You are now connected to your Windows Dev instance in the browser via Guacamole.

    Tip Web browsers don't provide access to clipboard data, which means synchronization between your local clipboard and the remote clipboard is impossible. To copy and paste when using Guacamole, you must use the Clipboard editor. To open the Clipboard editor, press **Ctrl -> Alt -> Shift**.

    Clipboard



Copy your text and paste it to the Clipboard editor. This will set the clipboard of your Dev instance to what you just pasted. You can also edit the text that you place in the Clipboard editor before pasting into your remote desktop.

To close the Clipboard editor, select **Ctrl -> Alt -> Shift**.

To continue this lab, move on to Task 2: Checking Your AWS Credentials.

Connect to Your Windows Dev Instance from a Windows Machine

In this task, you will connect to a Windows EC2 instance from your Windows machine.

Note

Perform the steps in this task only if you are connecting to **Windows Dev**

Note

Perform the steps in this task only if you are connecting to **Windows Dev Instance** from a Windows machine.

59. In the lab console, go to the **Connection Details** section and copy the **WindowsInstanceIP** to the clipboard.

60. Open the Remote Desktop Connection application on your computer.

    - On Windows 7, select the **Start** icon, and in the **Search programs and files** textbox, type `Remote Desktop Connection`. Choose the application when it appears in the **Programs** list.
    - On Windows 8, activate the Charms menu by moving the cursor into the lower right corner of the screen, and select the **Search** icon. Type in `Remote Desktop Connection`. Choose the application when it appears in the **Programs** list.
    - On Windows 10, select the **Start** icon, and select the **Search** icon. Type in `Remote Desktop Connection`. Choose the application when it appears in the **Programs** list.

61. In Remote Desktop Connection, for **Computer**, paste the IP of your Windows instance that you copied.

62. Choose **Connect**.

63. Remote Desktop Connection will prompt you with a Login dialog asking for your username and password. By default, the application will use your current Windows username and domain. To change this, select **Use another account**.

    Note

    On Windows 10, select **More Choices** before selecting **Use a different account**.

64. Go back to the lab console and copy the **WindowsPassword** to the clipboard.

65. For your login credentials, use the following values:

65. For your login credentials, use the following values:

    - For **User name**, enter: `\Administrator`
    - For **Password**, paste the password from the clipboard.

    Note
    The `\` in the user name is important, as it tells Remote Desktop Connection that you are logging in as the local Administrator, and not as a domain user.

66. To connect to your instance, select **OK**. If you receive a prompt that the certificate used to verify the connection was not a known, trusted root certificate, select **Yes**.

    Result

    Your connection to your remote instance should start momentarily. When lab instructions in subsequent sections require a command window, open or use a Powershell window.

    To continue this lab, move on to Task 2: Checking Your AWS Credentials.

Connect to Your Windows Dev Instance from a macOS Machine

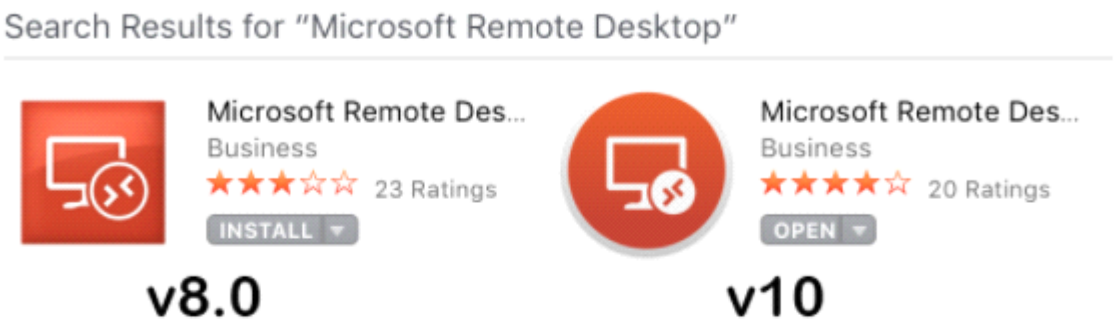
In this section, you will connect to a Windows EC2 instance from your macOS machine.

67. In the lab console, go to the **Connection Details** section and copy the **WindowsInstanceIP** to the clipboard.

WindowsInstanceIP to the clipboard.

68. Install Microsoft Remote Desktop if it is not already installed. To install, complete the following:

    - At the top of the screen, select the Apple icon.
    - Select **About This Mac**.
    - Take note of your macOS version.
    - From the Dock, launch **App store**.
    - Search for the following string: `Microsoft Remote Desktop`
    - If your macOS version is OSX 10.11+ select **Microsoft Remote Desktop 10**.
    - If your macOS version is OSX 10.9 - 10.10 select **Microsoft Remote Desktop 8.0**



    - Choose **Install** or **GET** to install the appropriate version.

69. To open **Microsoft Remote Desktop**, on the Dock, select **Launchpad**. Then, select **Microsoft Remote Desktop**.

70. To create a new connection, select **New** (v8.0) or **Add desktop** (v10). **Note** For v10, once you select Add Desktop, you will need to select **Show More** to enter the following details:

    Use the following values:

    - For **Connection name (v8)/ Friendly Name (v10)**, enter: `Windows Dev Instance`

- For **Connection name (v8)/ Friendly Name (v10)**, enter: `Windows Dev Instance`
- For **PC Name**, paste in the IP address of your Windows Server instance that you copied to the clipboard.
- For**User name (v8.0)**, enter: `\Administrator`
- For**User name (v10)**, enter: `Administrator`

Note

For v10 users, you will need to select **User Account** and **Add User Account** to be able to paste the username and password.

71. Go back to the lab console and copy the **WindowsPassword** to the clipboard.

72. Go back to your Microsoft Remote Desktop connection window and enter the following value:

    - For **Password**, paste in the password that you copied to the clipboard.

    Note

    The `\` in the user name is important, as it tells Remote Desktop Connection that you are logging in as the local Administrator, and not as a domain user.

73. **For v8.0 only**: Close the *Edit Remote Desktops* window by selecting the button on the top left corner.

74. In the *Microsoft Remote Desktop* window, double click the connection titled **Windows Dev Instance**.

75. In the *Verify Certificate* dialog, select **Continue** to complete the connection.

    Result

    Your connection to your remote instance should start momentarily. When lab instructions in subsequent sections require a command window, open or use a Powershell window.

To continue this lab, move on to Task 2: Checking Your AWS Credentials.

To continue this lab, move on to Task 2: Checking Your AWS Credentials.