Task 2: Create a Lambda Function

In this task, we are going to create a Lambda function by providing the required configurations.

1. Navigate to **Lambda** by clicking on **Services** menu. **Lambda** is available under **Compute** section.
2. Click on the **Create** function button.
   * Choose **Author from scratch**
   * Function name : Enter ***install\_request\_module***
   * Runtime: Select **Python 3.8**
   * Permissions: Click on **Change default execution role** and Select **Use an existing role.**
   * The role **lambda\_Role\_<RANDOM\_NUMBER> will be already created for you.**

A screenshot of a computer

Description automatically generated with low confidence

* Leave everything as default and click on **Create function** button.

A screenshot of a computer

Description automatically generated with low confidence

     3. Copy the below code :

|  |
| --- |
| import json  import requests  def lambda\_handler(event, context):      r = requests.get('https://github.com')      print(r.status\_code)      print(r.headers['Date'])      print(r.headers['server'])      return {          'statusCode': 200,          'body': json.dumps('success')      } |

      4. Scroll down to the **Function code** section,  remove the existing code and paste the new code.

1. Code Explanation:
   * Import requests python module to the code
   * HTTP request to GitHub URL
   * Print the connection status, server date and server name.
2. Deploy the code by clicking on **Deploy** button.
3. Now click on the **Test Dropdown then click on** configure test event.
   * Event template : Select **hello-world**
   * Event name : Enter ***TestCode***
   * Click on the **Save** button.
4. Now again click on the **Test** button and you will get the below error message in **Execution results**.

A screenshot of a computer error

Description automatically generated with low confidence

     9. This is because the request module is not available in Lambda, so we need to install this library package to run this code.

Task 3: Create an AWS Cloud9 Development Environment

1. Navigate to Cloud9 by clicking on **Services** menu. **Cloud9** is available under **Developer Tools**section.
2. Now click on the **Create environment** button.
3. On the **Name environment** page :
   * Name : Enter ***whiz\_lambda\_env***
   * Description : Enter ***Creating a new Cloud9 Environment for Lambda***
4. Scroll down a bit.

A screenshot of a computer

Description automatically generated with medium confidence

1. choose to **Create a new EC2 instance for the environment (direct access)** which Launches an Amazon EC2 instance that AWS Cloud9 can connect to directly over SSH

A screenshot of a computer

Description automatically generated with medium confidence

1. Under Network settings select Secure Shell (SSH) and Click on **Create**button.  
   A screenshot of a computer

   Description automatically generated with medium confidence
2. Now it will take a few minutes to launch the Environment. Click on **Open**A screenshot of a computer

   Description automatically generated with low confidence
3. If AWS Cloud9 doesn't display the IDE after at least five minutes, there might be a problem with your web browser or instance or others.

Task 4: Import the Lambda function to Cloud9

1. Once the Environment is ready it looks like the below screenshot.

A screenshot of a computer

Description automatically generated with medium confidence

1. On the left side you can see the Environment **whiz\_lambda\_env** folder.
2. Now on the left side click on the AWS icon  , click on the , then click on .
3. Now select  , Right click on the it and select .
4. It asks to **select a workspace folder for your new project** , do nothing just hit the [enter] key in your keyboard.
5. Now the lambda code that you saved in the Lambda function is available in the Cloud9 environment.

A screen shot of a computer code

Description automatically generated with medium confidence

1. Now Select  on the left side above the  icon.
2. In the Folder hierarchy, You will be able to see a folder created on your Lambda function name **install\_request\_module.**

A screenshot of a computer

Description automatically generated with medium confidence

Task 5: Install requests Library and download it

1. Now select the **install\_request\_module** folder and Right click  and select .
2. Now a **bash** window opens up in the lambda function folder path.

A screen shot of a computer

Description automatically generated with low confidence

1. Run the below command to install requests python module in the lambda function module.
   * Syntax :

pip install --target ~/environment/<Your lambda function name>/ requests

* + Example : **pip install --target ~/environment/install\_request\_module/ requests**



1. Now on the left side module you will be able to see the requests package got installed inside the lambda function name folder.
   * Move to the **install\_request\_module** folder by pasting the following command:

cd install\_request\_module/

1. Run the below command to zip all the files and folders inside the Lambda function name folder, **install\_request\_module**
   * Syntax :

zip -r ../<zip folder name>.zip \*

* + Example : **zip -r ../my-deployment-package.zip \***



1. Now you will be able to see the zip folder created in the lambda function name folder.



1. Right click on the Zip file **my-deployment-package** that was created in the above step and click on the  option.
2. Now the Zip file will be downloaded to your local machine.

Task 6: Upload the deployment package and test

1. Click on the link to open the lambda console <https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions>
2. Navigate to your **Functions**from the left side menu.
3. Click on your lambda function name, ie **install\_request\_module.**
4. Scroll down to the Code source part and click on the  button and select **.zip file** .

A picture containing text, screenshot, font, white

Description automatically generated

1. Click on the **Upload** button.
2. Now select the zip file **my-deployment-package.zip** from your local machine and click on **Save**button.
3. Now the whole deployment package is added to the Lambda function.

A screenshot of a computer

Description automatically generated with medium confidence

      7. Click on the **Test** button and you will get the below error message in **Execution results**.

A screenshot of a computer

Description automatically generated with medium confidence

      8. Now we have successfully installed the requests module to lambda and run the code successfully. Like this you can install any external dependencies in the lambda function.

Task 7: Validation Test

1. Once the lab steps are completed, please click on the A green and white sign

   Description automatically generated with low confidence button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :

A screenshot of a computer

Description automatically generated with medium confidence

Task 8: Delete AWS Resources

Deleting Lambda Function

1. Make sure you are in the **US East (N. Virginia) us-east-1** Region.
2. Navigate to Lambda by clicking on **Services** menu. **Lambda** is available under **Compute** section.
3. Select the lambda function **install\_request\_module** and click **Actions** and click **Delete**button.

A screenshot of a computer

Description automatically generated with low confidence

1. Click the **Delete** button to confirm deletion.

Deleting AWS Cloud9 Environment

1. Navigate to Cloud9 by clicking on **Services** menu. **Cloud9** is available under **Developer Tools** section.
2. Under **Your environments**, you will be able to see the environment that you created, **whiz\_lambda\_env**
3. Select the Environment and click on **Delete** button.
4. Enter ***Delete*** in the confirmation textbox and click on the **Delete** button.
5. Now it will take a few minutes to delete your Environment.

**Completion and Conclusion**