**Solution Template 8 - API Gateway and Docker**

Due: Sunday November 4 11:59PM ET

**Name:**

**Problem 1: API Gateway (50 Points)**

**a)** Create an s3 bucket and add some text files (5 points)

*NOTE*: Bucket created doesn't have to be a public bucket.

Paste the screenshot of the bucket with the stuff in it.

|  |
| --- |
|  |

**b)** Create an IAM role with the following policy and attach to a role that has trusted relationship with api gateway( 5 points )

|  |
| --- |
| {  "Version": "2012-10-17",  "Statement": [  {  "Effect": "Allow",  "Action": [  "s3:Get\*",  "s3:List\*",  "s3:Put\*"  ],  "Resource": "\*"  }  ]  } |

Paste the screenshot of the iam role with policy attached.

|  |
| --- |
|  |

**c)** Following the link, Create API Gateway to act as a proxy for s3

<https://docs.aws.amazon.com/apigateway/latest/developerguide/integrating-api-with-aws-services-s3.html>

**d)** Expose GET on the API's root resource to [list all of the Amazon S3 buckets of a caller](https://docs.aws.amazon.com/AmazonS3/latest/API/RESTServiceGET.html).

( 10 points)

Upon successful setup of GET operation on API gateway for s3 at root level, we should be able to test the API & in the response body, see the list of buckets

Paste screenshot of Method execution → Integration Request page , especially to show attached execution role, path override

|  |
| --- |
|  |

Paste the screenshot of the API gateway test response that shows the list of s3 buckets.

|  |
| --- |
|  |

**e)** Expose GET on the API's bucket resource to [list all of the Amazon S3 objects inside the bucket](https://docs.aws.amazon.com/AmazonS3/latest/API/RESTServiceGET.html).

( 15 points)

Upon successful setup of GET operation on API gateway for s3 at bucket level, we should be able to test the API & in the response body, see the list of objects inside the bucket

Expand methods in stages within the API created and paste the screenshot

|  |
| --- |
|  |

Paste the screenshot of the API gateway test response that shows the list of files in the s3 bucket.

|  |
| --- |
|  |

**f)** Expose PUT on the API's bucket resource to be able to place an object inside the desired bucket

( 15 points)

Upon successful setup of PUT operation on API gateway for s3 objects at bucket level, we should be able to test the API & in the response body, see the list of updated objects inside the bucket.

Paste the screenshot of the API gateway test response

|  |
| --- |
|  |

Paste the screenshot of the updated list of files in s3 bucket.

|  |
| --- |
|  |

**Bonus (5 points):**

Use Postman (https://www.getpostman.com) as a client application to put a file into any one of your buckets. To get credit you must show:

* The full command used

|  |
| --- |
|  |

* Screenshot(s) of results

|  |
| --- |
|  |

**Problem 2: Docker (50 Points)**

**a)**Create a free account in Docker hub a cloud based docker registry a.k.a repository.

* + Ref url <https://hub.docker.com/>

**b)** install docker command line utility on your computer and pull down official Apache HTTPD image from docker hub ( 10 points)

- Ref url <https://hub.docker.com/_/httpd/>

Paste the screenshot of the “docker images” command output from your computer’s terminal.

|  |
| --- |
|  |

**c)** Run a httpd container non-interactively from the downloaded docker image using docker command line. Bind the container’s port 80 with host port 8080. ( 10 points)

Paste the screenshot of the “docker run” command output from your computer’s terminal that is used to run the httpd image

|  |
| --- |
|  |

Paste the screenshot of the apache url from the computer’s browser that shows index.html contents.

|  |
| --- |
| Localhost:80 shows the currently running httpd server not inside docker,  localhost:8080 shows the index.html within the docker container |

**d)** ssh into the running container and observe the contents of the index.html file ( 5 points)

Paste the screenshot of the “index.html” file from the docker container created from httpd image

|  |
| --- |
|  |

**e)** Build an index.html file of your choice (may be a bio page from previous homeworks)

**f)** Build a Dockerfile in such a way that would update the index.html file from the httpd image with your custom index.html file. (5 points)

Paste the screenshot of the Dockerfile that shows COPY command to copy updated “index.html” file

|  |
| --- |
|  |

**g)** Build the docker image with your new Dockerfile. Name it with <your-name>-httpd and add a tag “1.0” (5 points)

Eg: **joe-httpd:1.0**

Paste the screenshot of the command “docker images” to display original downloaded httpd image and the image you built.

|  |
| --- |
|  |

**h)** Instantiate a new container from the image built. Bind the container’s port 80 with an un-used port on the host. (10 points)

Paste the screenshot of the apache url from the computer’s browser that shows index.html contents. It should display updated index.html

|  |
| --- |
|  |

Bonus (5 points): Publish built image to your personal account in docker hub.

Paste the screenshot of the docker hub account images section that displays images you pushed from your local computer.

|  |
| --- |
|  |