20S503: Fundamentals of Scalable Computing – Project

Build your own Cloud Web Application on AmazonAWS

During this semester for the scalable computing course, you will develop a scalable cloud based web application. You can choose whatever application you want and have to implement and some samples are given below. Examples:

- Build a ride sharing application
- Build an online book store application
- Swachh City application
- Build an online bakery.

As part of the application, you will have to design some REST apis to work with the application, deploy the REST API implementation on a docker container and deploy the docker container on AWS.

In order to help you design the REST APIs for the application, a few sample APIs for teh online bakery application are given at the end. The only condition is that you have to implement at least 3 APIs for this application - one GET, one PUT/POST and one DELETE. In order to store data, you can store to a file.

The entire application will be built using Amazon Web Services. You should start with a AWS educate starter account.

Team Members

Each project should be done in a team of 3 members. Please update your team information and the project details to the link given below. Projects and design must be shared by March 31st
https://forms.gle/2v9s339g6HVHjveP8

Scope of Project

For this project, you will package your entire application and the rest services in a Docker container. You must ensure that the web server is also containerized as part of the docker container.

Deliverables

1. REST API design for your application for each one of the APIs must be implemented with proper status codes
2. These APIs must be exposed on the public IP address, so that we can run a test script to test for correctness of the functioning of the API. Hence it is important that you must stick to the specification. Major focus of the credits must be on this. You can use *postman (https://www.postman.com)*
3. Docker image with the web server and your services
4. Start the docker image as a container and then
5. Your source code must be documented and all your development must be done on github. You will share your github link with me.
6. Please submit a one page report based on the template highlighting your design

Marks: 10

Due Date: Apr 17, 2020

Evaluation Criteria

- REST API design (3 marks)
- Creation of docker image (2 marks)
- Creating a container with the docker image (2 marks)
- Viva (3 Marks) Also, if you implement anything over and above the specification, you can get additional credit for it based on the instructors decision. Additional things could be adding additional APIs for users.

Docker container tutorials

1. https://www.tutorialspoint.com/docker/index.htm
2. https://docs.docker.com/get-started/

Sample API design

1. **List all items**

   Route: /api/v1/items

   HTTP Request Method: GET

   Relevant HTTP Response Codes: 200, 204, 405

   Request: {}

   Response:

   ```
   {
       "item1Name": 200, //price of the item
       "item2Name": 150,
       ...
   }
   ```

Comments:

1. Each itemname name in the response body must be unique (case-sensitive).

2. Add an item to the shopping cart

Route: `/api/v1/addItem`

HTTP Request Method: `POST`

Relevant HTTP Response Codes: `201, 400, 405 (500 also if the service is not available?)`

Request: `{`

```
    "itemName": "Black Forest Cake",
    "number": "2"
}
```

Response: `{}`

Comments:

1) What will be the effect of repeating this command multiple times?

3. View the shopping cart

Route: /api/v1/cart

HTTP Request Method: GET

Relevant HTTP Response Codes: 200, 204, 405

Request: {}

Response:

{

"item1Name": "3", //#of items in the shopping cart

...

}

Comments: if the cart does not exist, a new cart is created and then elements are added.

4. Clear the shopping cart

Route: /api/v1/cart

HTTP Request Method: DELETE

Relevant HTTP Response Codes: 200

Request: {}

Response:

{

"item1Name": "3", //#of items in the shopping cart

...

}