Jason Cote

06/16/21

Project two Conference presentation: Cloud Development

https://www.youtube.com/watch?v=6Y7lSsX-4KE

- **Purpose of Presentation**
    - Hello my name is Jason Cote I am a CS student at SNHU and this is my final project before I graduate.
    - This presentation is going to help articulate the intricacies of cloud development to both technical and nontechnical audiences.

- **Containerization and Orchestration**
  - The model I used to migrate a full stack application to the cloud was AWS . This help store everything and be able to access it to deploy a website.
  - The tools that are necessary to be able to use containerization is
    - Docker-helped create the container
    - Powershell-helped run different commands to install and run applications, angular, backend rest api
    - VScode-helps write code
  - Why use Docker Compose?
    - Docker compose has a lot of great features. you can create an environment that is isolated and that can be interacted with. Helps you set different containers to run specifications with compose, which helps getting things started. Different containers can talk to each other. There is a way to set up automated testing environments.
    -

The Serverless Cloud

  - Serverless
    - What is “serverless”

        - Serverless is a way to build and run different applications without having to worry about servers
    - what are its advantages?
        - Serverless has a much lower cost and hosting a server on your own, security is handled by AWS, so you can focus on building and not worry about security. This can be developed much faster.
    - What is S3 storage and how does it compare to local storage?
        - Amazon s3 storage you can use it and do not need to worry about how much storage you are going to need in the future. You can access this storage anywhere you are in the world if you have internet.
        - With local disk storage there is a only a certain amount of storage you can have before you have to keep adding costing you more money. You are not able to access it unless the equipment is local to you. You must worry about security yourself and figure out what you want for security and possible have to pay for it.
        - The images is a UML diagram of a serverless and local storage
- advantages of using a serverless API.
    - The advantages of using serverless api There are different policies that you can create and use. They have policies that can control who can create and manage the different APIs. You can also control different things like access control, and tracking.
    - Lambda API logic.
        - Everything is set up after you create the lambda function. You can check for certain things by testing.
    - What scripts are produced in order to make this happen?
        - You can run the script
        - Export.handler = async (events =>
            - Const response
            - Status code:200
            - Body.json.stringify('hello from lambda
            - Return response
            - When run correctly you will receive a statues code 200, and a body that says hello from lambda.

    - How to integrate the frontend with the backend.
        - The way to integrate the frontend and the backend is to make sure you are creating a fork of the different repository's in your account and then making sure you are deploying the backed and then deploy the front end.
- What are the data-model differences between MongoDB and DynamoDB?
    - Mongo and Dynamo are both database engines, that can be used and accessed in applications from a cloud. MongoDB is open source; it does have an intergraded language which is C++. You can run mongo on many different OS, Linux, OS, Solaris, windows. MongoDB can use many different programming languages. There is a server-side script that is JavaScript. When it comes to DynamoDB this is an engine that is an Amazon web service database engine. You must pay for the license to use DynamoDB. It uses RESTful HTTP API. There are not as many supported programming languages that are support by Dynamo compared to MongoDB. You can define and apply roles to who can use and access Dynamo. There is no server that you must manage with Dynamo since it is all taken care of with AWS.
    -
    - What queries did you perform?
        - The queries I performed was adding items and attributes

These are a few scripts I used to add the items and attributes. `{`

```
  "resource": "/Questions",
  "httpMethod": "GET",
  "queryStringParameters": {
    "filter":
"{\"include\":{\"relation\":\"answers\"},\"where\":{\"categorySlug\":\"a
ngular\"}}"
  },
  "multiValueQueryStringParameters": {
    "filter": [

"{\"include\":{\"relation\":\"answers\"},\"where\":{\"categorySlug\":\"a
ngular\"}}"
    ]
  }
}
```

        -
- **Cloud-Based Development Principles**
    - Elasticity
        - The elasticity is how the system is able to adapt to different workloads and being able to change the resource, being able to allocate the different resources to where it is needed, always changing and on demand.
    - Pay-for-Use Model
        - This is a method at which you pay for what you are using and charged by what was used.
        - Suggested image: An AWS graph comparing Capacity vs. Usage has been included in your template.
- **Securing Your Cloud Application**
    - How can you prevent unauthorized access?
        - The way to prevent unauthorized access is to make sure passwords are correct and updated as needed, users removed and the system is monitored.
    - Explain the relationship between roles and policies.
        - The different between roles and polices is that roles help set up the different permissions for a user and pilices help define the permission that the admin requires for the different user.
    -
    - What custom policies were created?
        - The different polices that was created were being able to have the different parts be able to communicate to each other.
    - How can you secure the connection between Lambda and Gateway?
        - The way that you secure the connection between lambda and the gateway is to create a JSON policy. That defines the specific attributes that you want to be able to be taken as actions.
    - Lambda and the database
        - Setting up a json policy to keep the database and lambda secure.
    - S3 Bucket
        - Set up the security configuration based on needs, make sure there is encryption and make sure there are role-based access set up for users.
    - Conclusion

- Serverless
    - Understanding serverless cloud computing will help you realize that it will save cost and time being able to not worry about having to buy more space or worry about creating your server.
-
- Testing
    - Testing in cloud computing has become very simple and easy to make sure different things are working as they should, being able to write the test and test different thing right can really help understanding where there is problems.
- Security
    - Adding security into cloud development is very similar just apply the policy and set users to there different groups will make sure everything is secure and worry free.