CS 470 Project Two Conference Presentation: Cloud Development

Eric Wallace

06/2023

Overview

- What is Containerization and Orchestration
- The tools required for implementing containerization and orchestration.
- Benefits of using Orchestration
- Concept and advantages of Serverless
- Introduce popular AWS serverless services used for cloud computing.
- Securing cloud applications

Models For Migrating a Full Stack Application To The Cloud

The models for migrating a full stack application to the cloud can be somewhat complex depending on the size of the application.

If the application is containerized, moving an application to a cloud-based Kubernetes or Docker Swarm might be ideal.

If migrating an application to service based such as AWS or Azure then it would involve moving the data and functionality of a container to a service which serves the same functionality.

Additional setup would be required for connecting services to one another along with securing services that are publicly accessed.

Containerization

Containerization is the process of consolidating an application including all dependencies in a single unit. They can then be deployed and run on any platform.

Docker is the most widely used tool used for containers.

Orchestration

Orchestration is the automating deployment, scaling and management of containers. The value of Orchestration is it gives developers the ability to share scripts which very easily generates an environment to which an application can be developed.

Docker Compose is a tool commonly used for orchestration of Docker containers. It allows multiple containers to be created with the use of a single file and usually only running one command.

The Serverless Cloud

Serverless

- Serverless computing is a cloud execution model for creating, deploying and running applications in which businesses do not have to worry about the underlying server infrastructure.
- Advantages of using serverless computing includes but is not limited to:
  - Automatic scaling
  - Built-in high availability
  - Eliminates provisioning and server patching
  - Reduced costs
  - Faster development of applications

Serverless Cloud - S3 Storage

AWS S3 storage is a serverless storage service, it allows developers to store and access data from the cloud. When comparing S3 to local data storage, S3 has many advantages such as:

- High scalability
- High availability
- Security
- Cost

Storing data locally means scaling is done manually with the need for additional hardware, availability is subject to internet services at the local site, security is maintained by onsite IT and purchasing hardware for the storage of data can be quite expensive. With S3 there is none of that, it is all handled by the provider of the service.

API & Lambda

Lambda is a serverless, event-driven service that runs code for virtually any type of application when triggered without the need of provisioning or managing servers. Lambda logic is the code that runs on the service and can handle requests made by other AWS services such as API gateway or S3 buckets configured as websites and returns a response. Nodejs scripts are produced which handle the logic for a function.

To integrate a frontend to a backend with Lambda:

1. Create a Lambda function
2. Create an API Gateway with the proper methods
3. Configure the API Gateway to utilize the Lambda function
4. Setup headers including CORS on the API Gateway request.

The advantages of serverless API is:

- Scalability
- Pay-for-use billing
- High availability
- Connection to other AWS services
- Security

The Serverless Cloud

Database

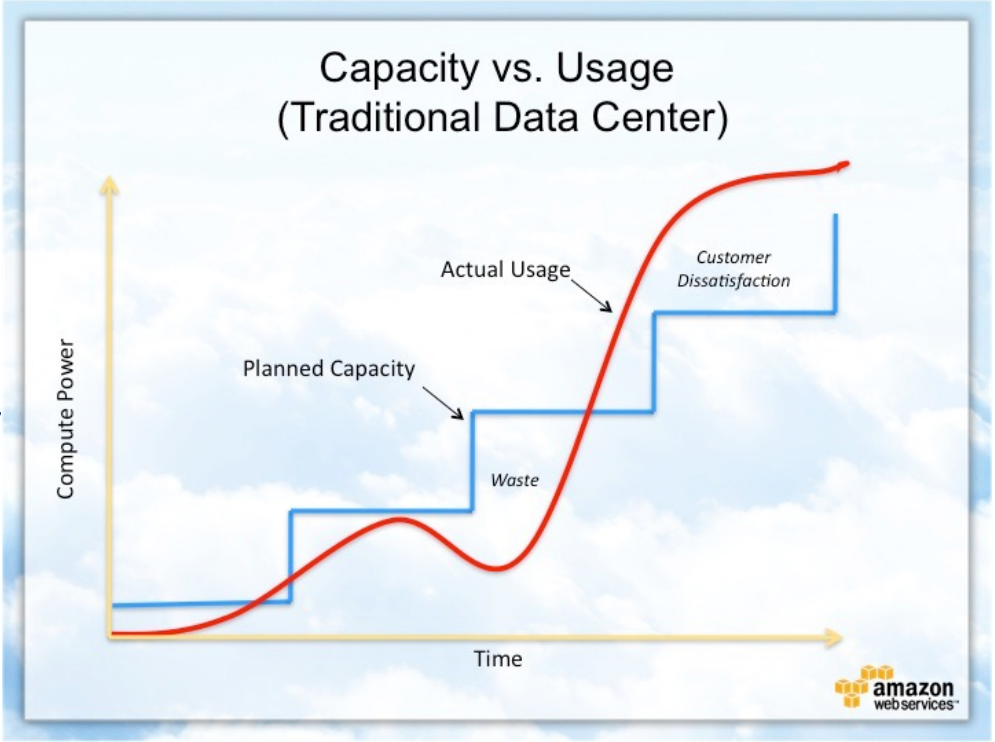
Below is a table comparing the two databases

	MongoDB Atlas	DynamoDB
Data Model	Document-Based	Key-Value Store
Querying	Single keys, ranges, faceted search, JOINs, graph traversals, and geospatial queries	Primary-key can have a max of 2 attributes, limiting query flexibility
Indexing	Hash, compound, unique, array, partial, TTL, geospatial, sparse, text and wildcard	Hash or hash-range only
Availability	Can be deployed on AWS, Azure and GCP	Only on AWS

Using Lambda functions with test scripts is a perfect way to retrieve, insert, update and delete data from a DynamoDB database table.

Cloud-Based Development Principles

- Elasticity is the ability to automatically scale a service up or down depending on demand
- Pay-for-use model is a billing model cloud service providers use in which they only charge customers for the usage of resources and results in a lower cost to the customer.



Securing Your Cloud App

Access

To prevent unauthorized access to an application or data, industry standards and guidelines must be followed.

Using a cloud service provider increases security to services and data because they use improved and current security standards to protect data.

The implementation of access control lists which only gives users access to relevant data

Using authentication methods such as two-factor authentication can also increase the security of an application.

Securing Your Cloud App

Policies

Roles and policies are two methods used to security data and services. Roles are a set of permissions which dictates what actions a user or service is allowed to perform. Policies are used to control how services or resources be accessed.

For example, a policy can be created to define what actions a Lambda function can perform on a DynamoDB table.

Securing Your Cloud App

API Security

Securing a connection between a Lambda function and an API Gateway could be done through the use roles and policies. API Gateways can also be secured using API keys.

Securing a connection between Lambda and a DynamoDB most commonly would be done using roles and policies. Restricting how a function can interact and to not allow more functionality than what is required.

Depending on the required access to an S3 bucket will dictate the amount of security required. Using roles and policies to limit the data an API has access to and what it can do with that data. Implementing least privilege access is a must. Using S3 bucket access control lists to limit what service can access data. Choose to encrypt data at rest, and data in transit.

CONCLUSION

- Developing or migrating an application to cloud services can reduce business costs, increase availability, leads to faster development and simplifies management.
- Utilizes services offered by AWS makes data more secure and less susceptible to attacks.
- The ability to access services from other services cannot be understated, while there is a learning curve, accessing resources of one service from another service can be quickly setup which could eliminate the need of having to do it within code.

Thank you for your time.

REFERENCES

AWS. (a). *AWS cloud security.* **AWS.** **https://aws.amazon.com/security/**

AWS. (b). *AWS lambda features.* **AWS.** **https://aws.amazon.com/lambda/features/**

MongoDB.*Comparing DynamoDB and MongoDB.* **MongoDB. Retrieved Jun 1, 2023, from** **https://www.mongodb.com/compare/mongodb-dynamodb**

Villanueva, J. C.*Amazon S3 vs local storage - where should you store files uploaded to your file transfer server? | JSCAPE.* **Retrieved Jun 1, 2023, from** **https://www.jscape.com/blog/amazon-s3-vs-local-storage-where-should-you-store-files-uploaded-to-your-file-transfer-server**

Thank you for your time.