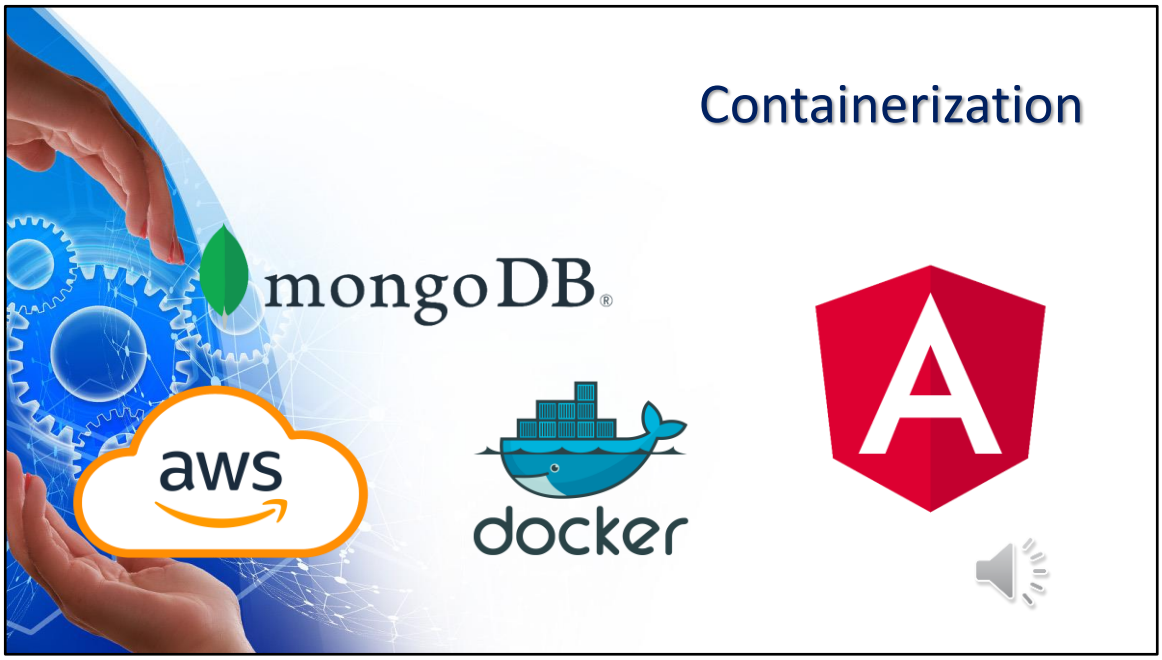
Christopher Dechene
Feb/19/2023
Assignment: CS 470 Project Two Conference Presentation: Cloud Development
Video link: https://youtu.be/taqfrxKB3ac

Hello everyone, my name is Christopher Dechene. I am a student at SNHU, finishing my last term with Full Stack Development II. This presentation today will be about discussing cloud development with our application.

Overview

I will be going over the process of migrating a full stack web application to a cloud-native web application with the use of Amazon Web Services. A Full stack web application includes the user interface of the site which is the front end, essentially how the site appears to the user. Meanwhile the back end is the business logic of the site. Essentially the structure of the site, its performance, logic, and handling of requests to the site.

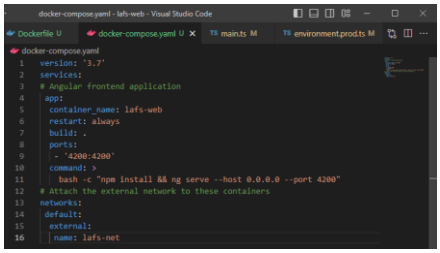
Cloud services specifically AWS allow developers to migrate web applications from a local server to a cloud server so that development can be readily available for remote workers, to increase security of the application, to allow greater storage of data, and to reduce costs of development with pay as you go price options.



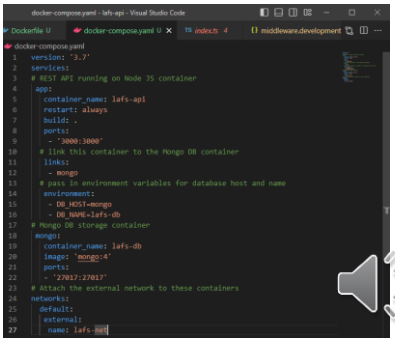
For this process we are using an Angular application that will be migrated onto Amazon Web Services. In order to prepare the angular app, we need to implement containers so that the application is functioning and can be transferred. We use Docker Desktop, Docker Compose and MongoDB to create a shared network for this app to operate on. These tools are necessary for containerization, to reliably get software to run when moved from one computing environment to another. In our case from a local storage to a cloud service. These containers will contain the entire runtime environment of our program.

Orchestration

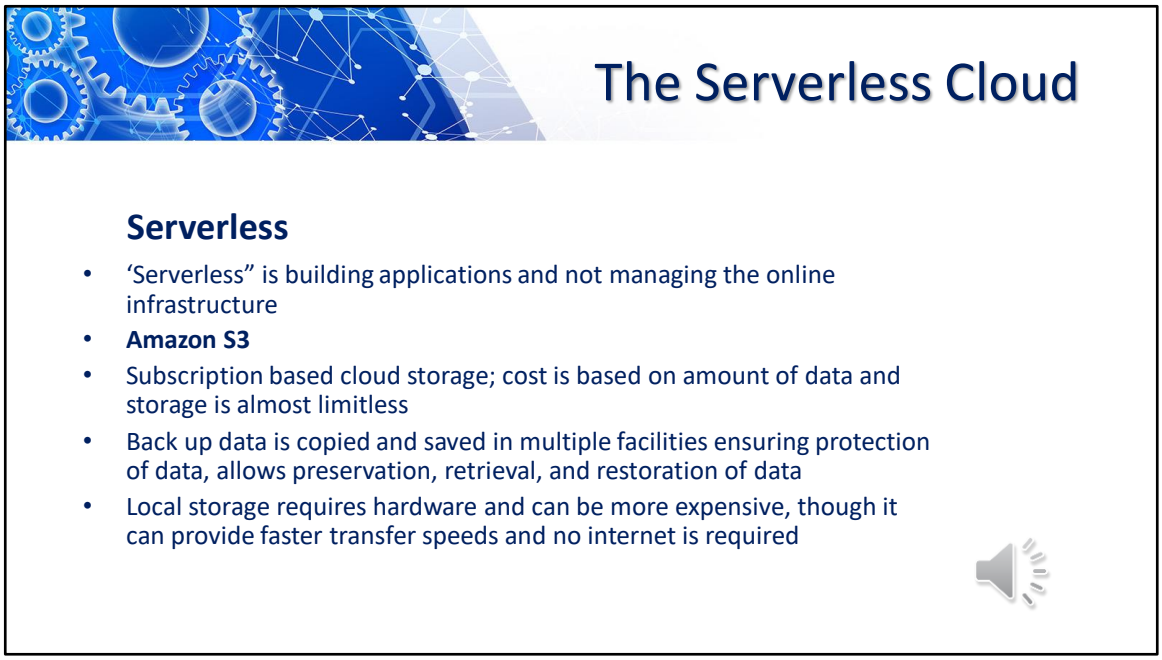
Compose for Front End



Compose for Back End



Once the files are prepared in our database to be transferred we still need to connect everything. This is were docker compose comes in. We have our data in the containers and now we create a bridge network so the containers can communicate with one another. One docker-compose.yaml file for the front end of our application, and the other for our back end. These files interact with our MongoDB database to run our angular application. Once our angular application is up and running it is then ready to be deployed for the cloud.



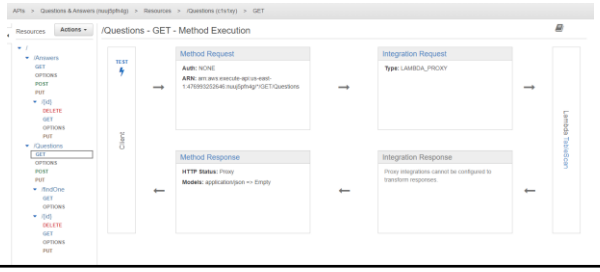
This cloud service is known as a "Serverless cloud". Web applications need to run on a server so users can connect to your application and so it can carry out its functions. Now with a serverless cloud you can upload your app to a cloud and there's no server for you (the developer) to worry about. AWS handles the server of your app for you, so you can focus on the development.

Amazon S3 is a subscription based, cloud storage for your applications. The cost is based on the amount of data you have and the price only goes up if you need more data storage. With local storage you need hardware to house the data for the application as well any other data for your database. While local storage can be faster in a bubble, and doesn't need an internet connection, AWS cloud storage can make numerous back ups of your data and copy multiple versions of your data. Should you make any changes to your app, those changes are saved as a separate version while your older copies are kept as well. So if you make an error or something is lost you can access the previous versions of your app. This ensures preservation of lost data, retrieval, and restoration. These back ups are also saved in multiple facilities to ensure better security and protection of your data.

The Serverless Cloud

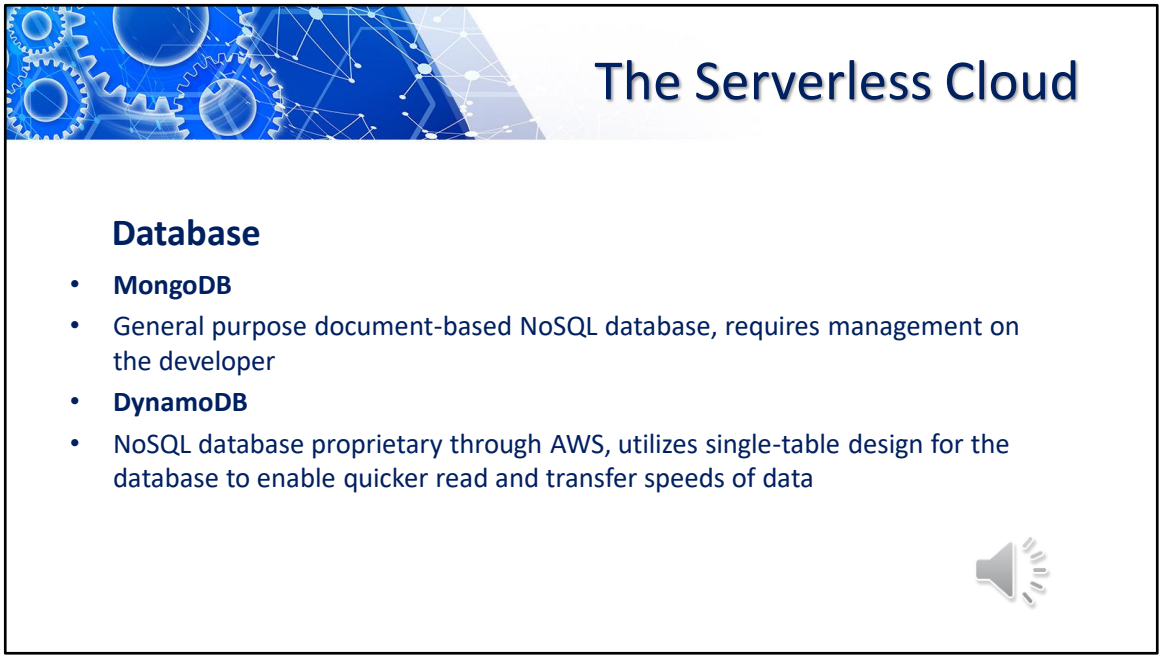
API & Lambda

- Serverless model with code deployed on the platform and it's run for you by AWS
- Build REST APIs and intergrade them with the Lambda function



Once our data is stored on the cloud server we need to implement functions to get it running. Here is where Lambda comes in. A lambda is a serverless model with code deployed, the server is run by AWS for us so we just worry about implementing the functions for the code. The lambda function is run with a java script file and its functions are tested with a .JSON file.

Once we have our lambda functions created we then implement APIs so the functions we create can connect to our database to perform tasks. Tasks that include creating, reading, updating, and deleting data or simply known as CRUD operations. When APIs are set up in AWS, we will be asked to add a permission to the lambda function we create, that enables the functions to communicate and transfer the data.

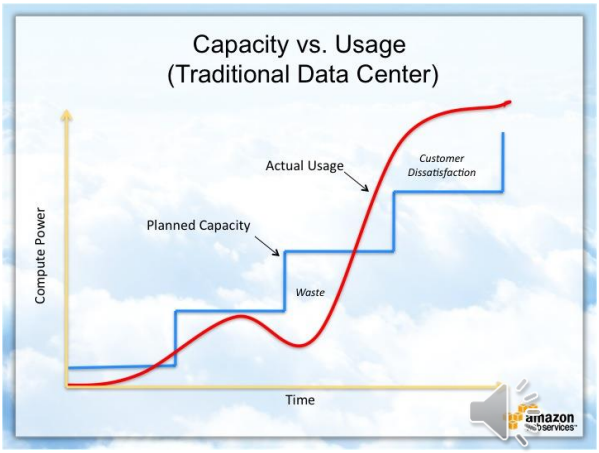


Now that we have our lambda function and APIs created there is also the database for our questions and answers. When developing a local application MongoDB is the go to for a NoSQL database. It is an open source document based database that offers high flexibility for users storing data. With its only main downside being that the potential of Mongo falls on the developers to utilize it.

AWS however has it's own database management tool, known as DynamoDB. Dynamo runs in tandem with the rest of AWS as it takes advantage of its integration tools and can be used almost immediately while developing. What makes Dynamo stand out is it's single-table design for the database. Most databases use what is known as foreign keys to interact with the various tables in a database, these act as pointers to refer from a record in one table to another one. Where as Dynamo has pre-joining of the data, the items in the single table share a primary key and a sort key. This allows the data to be searched and accessed at a much faster rate, greatly increasing it's performance.

Cloud-Based Development Principles

- Elasticity
- Pay-for-use model



Developing applications can become quite expensive, as I have mentioned before AWS has it's pay for use model. The more data you need, the more you pay to store it. With local storage you need to purchase hard drives, more PCs, and server chassis based on the amount you are working with. If you are making apps by yourself or work in a small business you won't really need much. Then again Amazon has a free tier so very low level projects can take advantage of standard storage without paying anything.

AWS offers developers flexibility with their data storage. Once you don't need that space you just don't pay for it anymore. It's much easier than trying to sell back used hardware that you don't need. Same with increasing data, as you reach the next data cap you are just charged more. Where as local storage means your company needs to purchase and install new hardware, these costs can go up ridiculously high. That being said companies may still want their data to be hosted locally and can even have both local and use cloud storage as a back up. Though it can't be stated enough that cloud storage is a very cost effective solution for large scale development projects.

Securing Your Cloud App

Access

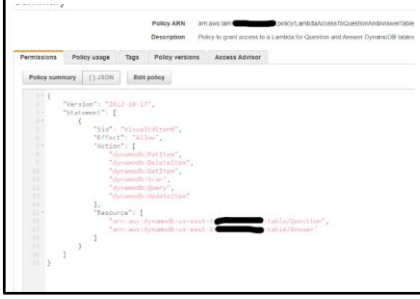
- Permission policies through IAM Roles and Policies

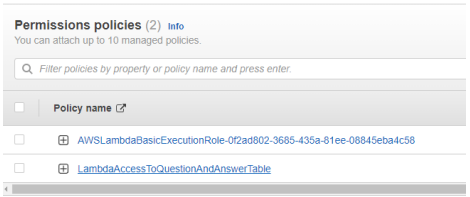
Policies

- Custom policy to enable CRUD functionality with DynamoDB

API Security

- Permissions
- API Keys
- Configuring a web application firewall





Going back to development, we have security with the cloud application. For any development project you don't want unauthorized users tampering with the project, and at the same time you don't want programs interfering with each other as they run. Through IAM roles and policies we can enable permissions on the project itself. We created a custom policy to enable CRUD functionality between the DynamoDB and our lambda function. The ability to read, write, access, and delete data on our application.

Other types of security on APIs include API keys that are unique identifiers that connect and perform calls that send or retrieve data. Another type of security can be a web application firewall to reject any unwanted traffic by filtering HTTP: requests.

CONCLUSION

- **Cloud Development…**
- Is cost effective
- Saves resources
- Allows more freedom

Thank you for your time.

In conclusion, migrating to a cloud service can have many advantages for development. As explained previously with capacity vs usage, costs can drastically go up on a local storage as you need to house more data. Using cloud storage can mitigate these costs so you can devote more resources to developing the project itself. You won't need to worry about hardware for storage and running a server. With less resources at hand to develop, your team will be allowed more freedom on the project. You can have remote workers reliably develop the project. You can have more time and effort put into various aspects such as performance, security, and maintenance.

Well that is all I have for today. Thank you for your time.