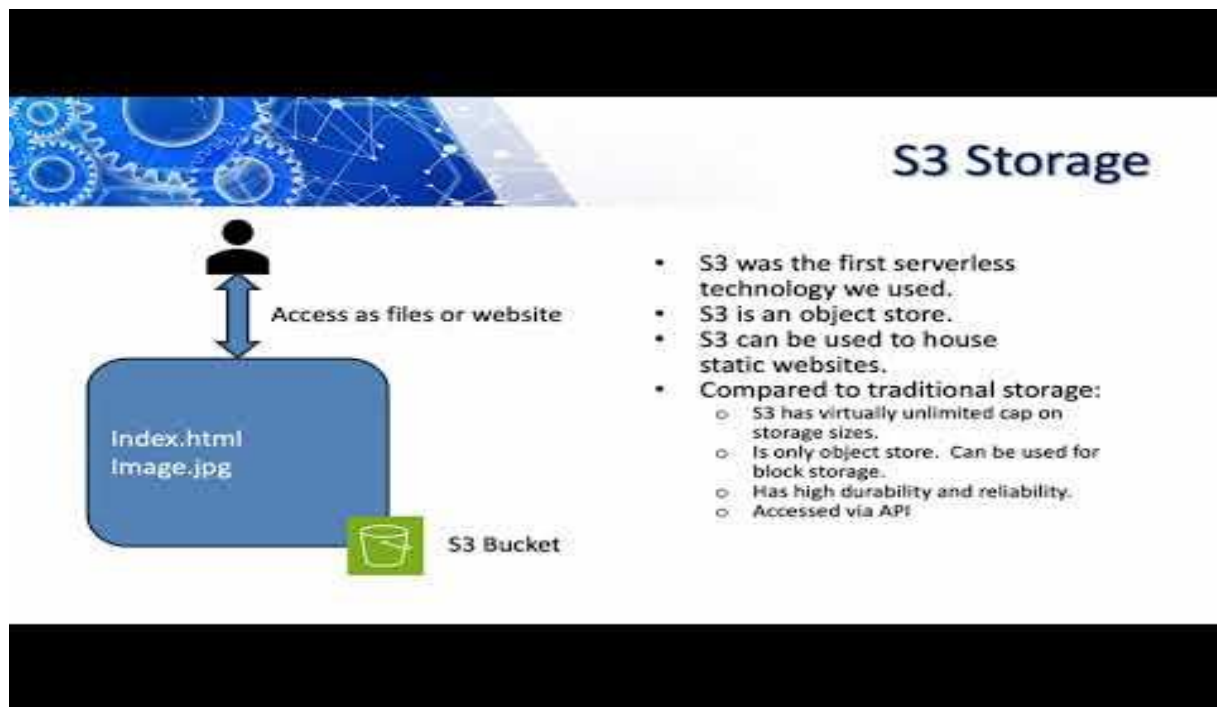


Michael Ross

4/28/2024

CS470 Final Reflection

<https://youtu.be/ZrqYrxVH8l0>



Final Reflection

Experiences and Strengths:

Having hands-on experience with one of the most used cloud platforms is invaluable in being marketable to any organization which may develop software toward a cloud platform. As someone who previously made hiring decisions for cloud platform engineers, I can say that anyone who can demonstrate some level of cloud understanding, and hands-on experience was much more likely to make it back to the second round of interviews.

If I were asked to describe my strengths as a software engineer, I would say it lies in the willingness and the passion for learning new technologies. Like any other activity in this life, you are going to gravitate to and be more likely to put the time into things that you love. Having a passion for technology allows me to excel in the things that I know but also to learn quickly. With the ever-changing pace of technology, being able to come up to speed quickly is an especially important skill to have.

As part of any new job I would want, I can see myself starting to gravitate toward a full software development track rather than the infrastructure and platform work I am doing now. I am already well established in my current career so any change would have to be gradual and well planned out.

Planning for Growth:

The application we migrated was straightforward with no authentication or integrations with 3rd party applications. The user load is also incredibly low, and one API frontend combined with the Lambda backend was sufficient for handling the load. As the application grows, I can see a need for more services.

For the API layer, because that was implemented in API Gateway, the availability and scaling of that gateway is handled for us by AWS. One aspect which is not handled automatically however is ensuring that the APIs can be utilized from anywhere in the world with low latency and through a common front-end. By integrating with CloudFront we can ensure low access times coupled with a common front-end URL that users worldwide can use.

Currently, there is no visibility into the tech stack we are using from a monitoring perspective. By centralizing our logs in CloudWatch and creating dashboards, we can be alerted if anything in the application starts to go wrong and the centralized logging will help us better troubleshoot the application.

Another point that would need to scale would be the database. The DynamoDB instance used for the application can be set up to auto-scale if there is a surge in traffic. This will keep performance metrics within our KPIs. When the traffic is reduced, the DynamoDB instance will scale back down to save on cost.

Speaking of cost, AWS provides ways to reduce cost in the cloud. By using the cost optimization hub, organizations can keep costs down by implementing the recommended settings and policies to ensure that no money is wasted.

While serverless provides remarkably high amounts of scalability and flexibility, it is also charged by the transaction. If some code change causes a spike in the number of transactions or if there is an attack on a Lambda service, the cost for that service can go up unexpectedly. Conversely, with containers, the cost is more stable because hosts must be provisioned for them to run on, and you pay for the host capacity instead of per transaction.

In general, because of the spiky or unpredictable nature of cloud spending, organizations need to develop policies and comfortability around cloud spending. Cloud providers give this level of comfortability by offering pre-paid 1-, 2- or 3-year terms for a reserved amount of cloud spending which flattens the spikiness of cloud billing and provides a way to safely budget for future cloud spending.