Script

Dani Yosopov

08/10/2023

Project Two Conference Presentation: Cloud Development

https://www.youtube.com/watch?v=60V5Yd8njF0&ab_channel=DaniYosopov

1. Hello everyone, my name is Dani Yosopov,

   a little bit about myself I am originally from Israel and currently working in IT as a full-stack application developer,

   Recently we had some issues with our server systems trying to keep up with our growth, so I came up with the idea of migrating the application to a serverless application repository.

   I will present to you an illustration of the concepts and processes of the migration from a traditional full stack to a cloud-native web application using AWS.

2. There are seven cloud migration strategies: rehosting, redeployment, repackaging, refactoring, repurchasing, retiring, and retaining.
   Organizations looking to migrate to the cloud should consider which migration strategy best answers their needs.
   Rehost - migrate an application to the cloud without implementing changes to utilize cloud features.
   Relocate - Shift infrastructure to the cloud without the need for new hardware, application rewrites, or adjustments to current operations. Technologies like VMware Cloud enable this migration approach.
   Refactor - Transform an application by altering its architecture and leveraging cloud-native features to enhance agility, performance, and scalability. This usually involves porting the operating system and database.
   Repurchase - Transition to another product, often by adopting a SaaS model instead of a traditional licensing approach.

Replatform - Transfer an application to the cloud, incorporating some optimization to benefit from cloud capabilities.
Retain - Maintain applications in their original environment. This can include applications that need significant refactoring, which can be deferred to a future time, or legacy applications that remain in place due to a lack of business justification for migration.
Retire - Discontinue or eliminate applications that are no longer required in the source environment.

3. So, what is A container, and why do we use it?
   Containers are packages of software that contain all the necessary elements to run in any environment. In this way, containers virtualize the operating system and run anywhere, from a private data center to the public cloud or even on a developer's personal laptop. Containers allow applications to be more rapidly deployed, patched, or scaled. Containers support agile and DevOps efforts to accelerate development, test, and production cycles.
   There are many powerful Orchestration tools, here are some worth checking OKD, Docker Compose, Operator Hub.
   In this case, I had chosen to work with Docker Compose.
   And the benefits of using it are:
   Single host deployment - This means you can run everything on a single piece of hardware.
   Quick and easy configuration - Due to YAML scripts.
   High productivity - Docker Compose reduces the time it takes to perform tasks.
   Security - All the containers are isolated from each other, reducing the threat landscape.

4. what do I mean by Serverless?
   Serverless is a cloud computing application development and execution model that enables developers to build and run application code without provisioning or managing servers or backend infrastructure.

   Using a Serverless framework can offer a few advantages over traditional infrastructure. For many developers, serverless architectures offer greater scalability, more flexibility, and quicker time to release, all at a reduced cost.

5. In this project, I had used Amazon Simple Storage Service, or in other words S3, which is an object storage service that offers industry-leading scalability, data availability, security, and performance.
   With the S3 storage, you can store and retrieve any amount of data at anytime from anywhere.

   There are some key differences between S3 storage and local storage, with each having its own benefits, the choice of which one is the best depends on the purpose of use.

For example, S3's main strengths are time-saving, security, and a user-friendly interface.
Local storage allows you to store your objects locally, your privacy is yours to manage, so you can customize it to meet all your demands and requirements.
Amazon S3 offers virtually unlimited storage, and you won't have to worry about guessing how much space you'll need.
While physical hard drives which have a limit on their space by default.

6. What are APIs you ask? Well, the job of the API is to be like a messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.
   There are a lot of benefits in Serverless API but today I will talk about the API that I had used in my project which is Amazon Web Services API gateway.
   It helps with managing traffic, authorizes end users, monitors performance, and it offers Amazon firewall servers that protect your web application from common web attacks.
   I used the Lambda logic which allowed my API to interact with my database by creating a JavaScript index file with the proper code for each method like Get, Post, and Delete.

   The steps are very simple first we create the Lambda method by giving Its name, then we create the fitting code that would perform the wanted action, that part can be tested by creating many tests to see the results with the code you created and then we must hit deploy in order for it to take effect.

7. MongoDB and DynamoDB are databases that are used to store application data.
   There are some differences between those two databases.
   The major one is the amount of data we can store.
   in MongoDB, we use JSON-based documents to store up to 16MB document size.
   On the other hand, DynamoDB has a Maximum of 400KB record size.
   MongoDB is very versatile with other platforms unlike the DynamoDB which is only working with AWS.
   In the project, I used 3 queries: Answer without filter, question with filter, and question without filter.
   With those test events, we can pull all the questions and their corresponding answers that are in the database.
   This was done with a JavaScript JSON script file which is as part of test for database table.

8. One of the most important principles is Elasticity which refers to the capability to adjust the deployment scale of a system as the business volume changes. This eliminates the need to prepare fixed hardware or software resources based on capacity planning.
   In this graph, we can see the cloud elasticity which can be seen as the actual usage in the red line vs the planned capacity in a physical data center which is the blue line.

With this principle, we can save a lot of money because we pay only for the amount of data that is being used on a regular basis.

9. With the use of AWS, we are able to use a feature called IAM.
   It’s a web service that helps you securely control access to AWS resources.
   With IAM, you can centrally manage permissions that control which AWS resources users can access.
   You use IAM to control who is signed in and authorized to use resources.
   This can help with data breaches, identity theft, and illegal access to valuable organization data.
   it's all done with a role identity that can be edited with different policies, which defines the permissions of the role identity.
   For example, in this project I had created a policy to my role that allow my Lambda functions to access the DynamoDB and make the desired changes.

   For the security method between the API and Lambda AWS uses a special key that is auto generated to perform secure API calls
   the same method is using between Lambda the DynamoDB.
   The nice thing about the storge security is that AWS makes it private by default.
   On top of all that AWS offers the firewall services that is fully maintained by amazon.

10. In conclusion
    Cloud-Based Development allows us to be more flexible with application development instead of scaling and managing the servers.
    Cloud-Based Development helps us to save a lot of money when it comes to business volume changes.
    And the most important part, it has a user-friendly interface that makes development much easier.