Cloud & Serverless: Game Changer?

Taqui Syed Shah

March 2021

1 Introduction

Cloud is a topic that has become very common in software development. Cloud enables on-demand computer-related resources, most commonly storage and computing. The underlying hardware is managed by a cloud provider which enables developers to focus on the infrastructure. This enabled the infrastructure as a service(IaaS) model to a greater extent within software development. Recent developments in the cloud have led to a new model called Serverless. Serverless is an abstraction level above the cloud, this created different backend as a service models (BaaS) and functions as a service models (FaaS). "Serverless" refers to storage or computing where the developer does avoid managing the underlying infrastructure. This means that developers manage the code while the underlying infrastructure is managed by the cloud provider. The serverless model has enabled complete autoscaling with a pay as you go model. This has enabled immense cost savings as well as increased simplicity within software development. The most interesting questions are often how can we leverage Serverless?

This essay will explore that how Serverless can be leveraged to DevOpsify infrastructure.

2 DevOps and its relation to the Cloud

DevOps is a widely used term to define principles and practices that automate the process of software development to build, test, and release software more reliably [7]. Essentially, DevOps is about simplifying development and enabling developers to create, build and release software more easily. One aspect of software development is the underlying infrastructure it rests upon. That is the servers, the databases, and file storage, etc. Cloud is a more recent technology that has been used to simplify the creation, utilization, and scaling of an IT infrastructure. Cloud is used to enable on-demand IT-related resources such as computing or storage which is managed and maintained by the cloud provider.[8]. There are some important concepts to understand within the cloud to have a full grasp of the purpose and the usefulness of serverless.

2.1 Concepts of Cloud

The main concepts of infrastructure within the cloud are availability, scalability, and elasticity. Availability refers to the up-time of software, scalability refers to how well the software can allocate new resources depending on the load of the application, elasticity refers to the application being increase and decrease the resources depending on the load of the application, this is to reduce the cost. These concepts relate to DevOps since they are often automated within the cloud to enable a reliable infrastructure that our software rests on top of. An important topic within scalability is vertical scalability vs horizontal scalability. Vertical scalability refers to using utilizing stronger hardware for example a CPU with 16 cores instead of 8 cores. Horizontal scalability refers to utilizing additional compute units for example leveraging 2 CPUs with 8 cores instead of a single CPU with 8 cores. [11]

2.2 Regions and Availability Zones

To ensure high availability cloud providers have created regions and availability zones. Regions are physical locations around the world where cloud providers cluster multiple data centers. Each individual cluster of datacenters is called an availability zones (AZ). This structure enables high availability within the cloud by enabling developers to have their software on multiple AZs. If a data center were to fail or have an electrical outage the application would still be up and running since the other datacenters in the other AZs will still be available. This enables the application to have a huge amount of up-time.[5]

3 Introduction to Serverless

Serverless is used to describe services where the developer does not manage the underlying infrastructure and where the service often deals with scalability, availability, and elasticity. Serverless does leverage servers ultimately but it is abstracted away from the developer. Serverless uses different services for replacing traditional resources in cloud infrastructure. A common misconception is that serverless is purely a FaaS(functions as a service) platform but that is incorrect. Computing, storage, and databases are common resources within the serverless sphere. This means that BaaS (backend as a service) and FaaS services are within the scope of serverless. [12] The traits that are often shared among Serverless services are:

- No server management
- Automatic and Elastic Horizontal Scaling
- High Availability
- Pay as you go model

4 Utilization of Serverless to DevOpsify and build IT-Infrastructure

The question now is "how can serverless be leveraged to DevOpsify an IT-infrastructure?". Some of the applications can be very concrete such as creating DevOps tools using Serverless, other ways are to use already existing serverless DevOps tools such as CloudWatch[4]. However, DevOps is about automating and simplifying the software development process. Serverless can be leveraged instead of traditional cloud resources to simplify and automate the scaling, elasticity, and availability of IT infrastructure. Therefore, just leveraging serverless services for software can be enough to simply and automate a huge portion of the development process. Understanding the services that exist and how they are able to replace more traditional cloud resources can be an aid with the understanding of the benefits of Serverless. In this case, AWS and its services will be used as a base for discussing the benefits of a Serverless platform.

5 Serverless Computing

FaaS or often referred to as Serverless Computing is a service that enables server-side logic on stateless compute containers that are fully managed by the cloud provider. The server-side logic is a function but since these functions are run in a serverless context the functions are often called serverless functions. Making the functions run in a stateless compute container is what enables such ease of scaling because all functions are just a replica of each other and do not have any state. Serverless functions often do not last for a long time, depending on the cloud provider they last approximately 900s or 15 minutes[6]. This is why more long-running functions will not be appropriate for a serverless architecture. FaaS is considered to be part of event-driven computing, which means that the program is reactive to identifiable events.[14] This means that serverless functions are only called based on a certain event whether that is based on time, an email, or an HTTP request. Serverless functions often rely on other services within the serverless sphere, which can be either storage or a database. [12]

How can Serverless Computing be used to replace more traditional computing within the cloud and how can it be a benefit to leverage serverless computing?

Serverless Computing can be used to start replacing virtual machines such as EC2 instances in AWS. EC2 instances that are used in an event-driven manner can be replaced by serverless functions. If an application needs to train a machine learning model then serverless computing is not appropriate since there is a constantly running server. [9]

One benefit is automatic horizontal scaling. In AWS autoscaling groups are leveraged to horizontally scale an infrastructure. Instead of building out the infrastructure for the EC2 instances with autoscaling groups we can just create

a simple serverless function and have the cloud provider deal with the issues with scaling.[3]

The pay-as-you-go model enables enormous cost-saving since the developer only pay per request rather than for each EC2 instance that is running[3]. This means that when the system is idle the entire infrastructure is free. This can be extremely helpful for start-ups since paying for up-time can be quite expensive. Serverless functions are often also free to a certain amount of requests meaning that for a start-up the entire backend infrastructure can basically be free until it scales to large size and by that the business is probably making revenue to cover the costs.[2] [9]

6 Serverless Storage/Persistence

Storage can often refer to two different forms of storage, object-based and block-based.[15] To keep it simple object-based storage refers to files such as pictures, music, or large text documents. Object-based storage is used by the operating system for hosting databases, or for having your operating system for a virtual machine. In the serverless sphere, storage can often be split in two ways Serverless Object Storage and Serverless Relational Databases. In this case, serverless object storage replaces classic object storage you might find with EC2 instances and Serverless Databases aims to deal with the need for block storage for databases. [10] A distinction between storage and computing is availability is enabled by replicating the storage between multiple AZs.

6.1 Serverless Object Storage

Serverless Object Storage aims to deal with the use of object-based storage that was used in more traditional EC2 based applications. AWS offers a service called S3 which is serverless object storage that deals with availability, scalability, and pay-for-usage model. Serverless storage often offers buckets which is the main way you store object-based storage. Security is an aspect of serverless storage which makes it incredibly interesting since it can offer powerful access management through granular policy creation using JSON. [13]

In an application, Serverless Object Storage can replace certain block storage that is used in an application. EBS(Elastic Block Storage) is one of the main ways to generate block-based storage for an EC2 instance. Previously applications needed to leverage EBS or EFS to store object-based storage. This also applies to before the cloud was used as people also used FTP servers for file storage. S3 solves issues like this by dealing with the possible complexities of building object storage infrastructure since you can just leverage S3 instead of creating EBS volumes or FTP Servers. Scaling also becomes easier since EBS does not scale easily. Security is an aspect that enables S3 to be a lot more efficient compared to EBS since to EBS you will have to create an application layer for determining which have access to which resources, however with S3

you are able to create S3 bucket policies which make easier to manage access control.

6.2 Serverless Databases

A common use for block-storage is databases. To eliminate the need to block-storage in a serverless architecture most cloud providers offer some form of serverless database service. Cloud providers often provide both serverless relational databases and non-relational databases. These serverless services automatically deal with availability, scalability, and elasticity similar to S3. [13]

The common serverless relational database service is Aurora, and the common non-relational database service is DynamoDB.

In AWS the most common way to have relational databases is using Amazon RDS which is used to create a SQL database of any flavor. However, when creating such a database you will often have to deal with the backups, scaling using read replicas, and dealing with possible infrastructure failures and failover to your standby database backups.[1] If AWS Aurora is leveraged that would not be a problem since the service is implicitly available since it replicates to different AZs similar to S3. This is also very similar for non-relational databases, but in this case, the service you would use is DynamoDB as a replacement for the database.

7 Conclusion

DevOps defines the principles and practices that automate the process of software development to build, test, and release software more reliably. Serverless and Cloud have a clear spot in DevOps since these technologies enable extreme automation by automating the infrastructure management of availability, scalability, and elasticity. Serverless has enabled enormous strides in simplifying the creation of software by completely abstracting the infrastructure which enables a developer to purely focus on the code rather than the underlying infrastructure the code rest on top of. The future for Serverless is bright as it enables developers to build software with a simple and cheap infrastructure which hopefully enables more companies to scale with extreme simplification of the infrastructure and with a reduction in cost.

References

[1] Amazon relational database service (rds). "`https://aws.amazon.com/rds/`", 2020. Accessed: 2020-04-07.

[2] Aws lambda pricing. "`https://aws.amazon.com/lambda/pricing/`", 2020. Accessed: 2020-04-07.

[3] Aws lambda vs ec2: Comparison of aws compute resources. "`https://www.simform.com/aws-lambda-vs-ec2/`", 2020. Accessed: 2020-04-07.

[4] Amazon cloudwatch. "https://aws.amazon.com/cloudwatch/", 2021. Accessed: 2020-04-07.

[5] Regions and availability zones. "https://aws.amazon.com/about-aws/global-infrastructure/regions_az", 2021. Accessed: 2020-04-02.

[6] Serverless architectures with aws lambda: Timeout. "https://docs.aws.amazon.com/whitepapers/latest/serverless-architectures-lambda/timeout.html", 2021. Accessed: 2020-04-07.

[7] Christophe Capel. Devops: Breaking the development-operations barrier. "https://www.atlassian.com/devops", 2021. Accessed: 2020-04-02.

[8] T. Dillon, C. Wu, and E. Chang. Cloud computing: Issues and challenges. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pages 27–33, 2010.

[9] an NTT Data Company Flux7. Serverless vs ec2 vs containers: A comparative study. "https://cdn2.hubspot.net/hubfs/411552/Whitepapers/Whitepapers\%20in\%20the\%20sales\%20library/WP_\%20\%20Serverless\%20vs\%20EC2\%20vs\%20Containers_\%20A\%20Comparative\%20Study.pdf". Accessed: 2020-04-07.

[10] Gali Kovacs. Aws ebs and s3: Object storage vs. block storage in the aws cloud. "https://cloud.netapp.com/blog/block-storage-vs-object-storage-cloud", 2017. Accessed: 2020-04-07.

[11] Matt Rasband and Eugene Ciurana. Scalability and high availability. "https://dzone.com/refcardz/scalability?chapter=1", 2017. Accessed: 2020-04-02.

[12] Mike Roberts. Serverless architectures. "https://martinfowler.com/articles/serverless.html#WhatIsServerless". Accessed: 2020-04-06.

[13] Rajind Ruparathna. What makes a storage service truly serverless? "https://dzone.com/articles/what-makes-a-storage-service-truly-serverless", 2019. Accessed: 2020-04-07.

[14] PIOTR SROCZKOWSKI. Cloud: Iaas vs paas vs saas vs daas vs faas vs dbaas. "https://brainhub.eu/blog/cloud-architecture-saas-faas-xaas/", 2021-01-11. Accessed: 2020-04-06.

[15] Tony Piscopo Yadin Porter de León. Object storage versus block storage: Understanding the technology differences. "https://www.druva.com/blog/object-storage-versus-block-storage-understanding-technology-differences/", 2019. Accessed: 2020-04-07.