AWS Certified Cloud Practitioner Training Bootcamp

Design Principles - Scalability

Scalability Overview

- Systems that are expected to grow over time need to be built on top of a scalable architecture
- Scalable architectures provide the ability to grow your environment when this is needed (increase in number of users, traffic throughput)
- Cloud computing allows virtually unlimited growth, but the underlying architecture must be designed to support this
- Scale either vertically or horizontally

Scaling Vertically

- Scaling vertically means increasing the capacity of your current server, as an example

- At some point, you discover that your current server can no longer process the amount of data that is constantly increasing => you need to scale, or grow

- For example, you are running your website on an AWS EC2 instance: *a1.medium* (1vCPU , 2GB RAM) and you will migrate your EC2 instance to a more advanced option *m5.24xlarg* (96 vCPU, 384GB RAM)

Scaling Horizontally

- Scaling horizontally means increasing the number of current resources
  - i.e. adding more EC2 instances to support your website

- This is not always possible, depending on the underlying architecture, which can or can not distribute traffic to multiple resources

- We will analyse different scenarios: stateless apps, stateless components, stateful components and distributed processing

Scaling Horizontally – Stateless Applications

- What does stateless or stateful mean ?

- The key difference between stateful and stateless applications is that stateless applications don't "store" any data and connections are independent from one another

- A stateless application is an application that needs no knowledge of previous interactions and stores no session information; i.e. app provides the same response, to any user with the same input

Scaling Horizontally – Stateless Applications

- Why is this important ? ☺
- Stateless applications are a great candidate for horizontal scaling; simply add more EC2 instances in order to run your app and terminate EC2 instances when no longer needed
- The easiest and most popular way to distribute traffic to the EC2 fleet is through an Elastic Load Balancer (ELB)

Scaling Horizontally – Stateless Components

- Most applications need to maintain some kind of state information (web applications need to track whether a user is signed in)
- Some web applications use HTTP cookies to store data on the client side, other scenarios require storing larger files
- For the second option, Amazon S3 or EFS could be used

Scaling Horizontally – Stateful Components

- There are cases where you can not change all your components in your architecture to stateless
- Example: real-time multiplayer online gaming, users are connected to the same server (low latency and best experience this way)
- Horizontal scaling can be achieved in this case also, using "session affinity" – binding all connections from a specific user to only a single server

Scaling Horizontally – Distributed Processing

- This is similar to breaking a problem into smaller pieces
- When a single compute resource can not process that information, because it's too large for example, then the work will be distributed and split into small fragments, to more instances
- This use case is absolute common for Big Data scenarios (processing of large volume data sets)

AWS Certified Cloud Practitioner Training Bootcamp

Thank you