Cloud Comparison

Presented by: Shivani

Ahmet

Danish

Rohit

Serverless Computing

- AWS
  - 1M free request/month
  - Beyond 1M, $0.2/million
  - Available in most of the region
  - Supports range of runtime environments like java, nodejs, python etc
  - Can be triggered by most of the AWS services
  - Unlimited functions
  - Execution time: 5 min/request

Serverless Computing

- GCP
  - 2M free request/month
  - Beyond 2M, $0.4/million
  - Available only in the central region
  - Supports only NodeJS runtime environment
  - Can be triggered by using Pub/Sub topic, HTTP
  - Limited to 1000 function/project
  - Execution time: 9 min/request

Serverless Computing

- Azure
  - 1M free request/month
  - Beyond 1M, $0.2/million
  - Available in most of the region
  - Supports range of runtime environments like java, PHP, python etc
  - Can be triggered by most of the Azure services and HTTP
  - Functions up to 1536MB
  - Execution time: 5 min/request

Load Balancing

- AWS
  - Cannot handle sudden spikes in incoming traffic
  - Pricing based on deployed load balancers per hour
  - Additional charges for the amount of data processed

Load Balancing

- GCP
  - Sudden spikes in incoming traffic are automatically handled
  - Pricing based on number of forwarding rules
  - Additional charges for the amount of data processed

Load Balancing

- Azure
  - Can handle sudden spikes in incoming traffic
  - The Basic Azure Load Balancer is free of charge. The Standard Azure Load Balancer has a charge associated with it.
  - Additional charges on number of rules and amount of data processed

Auto Scaling

- AWS
  - Provides auto scaling service as IaaS EC2 public cloud
  - Can also specify minimum and maximum number of instances in each Auto Scaling group
  - Auto scaling in AWS uses load balancers to distribute traffic across the instances within auto scaling technique along with the elastic load balancing technique

Auto Scaling

- GCP
  - Managed instance groups offer autoscaling capabilities that allow you to automatically add or remove instances from a managed instance group based on increases or decreases in load.
  - Can also specify minimum and maximum number of instances in auto scaling policy
  - Unmanaged instance groups are not supported for auto scaling
  - For Google Kubernetes Engine groups, use Cluster Autoscaling

Auto Scaling

- Azure
  - Azure does not implement any embedded auto scaling solution to its users, rather it supports Paraleap software which automatically scales resources in Azure to respond to changes on demand
  - Auto scaling is achieved by changing the instance count in the service configuration
  - Azure provides an API that can be extended for auto scaling

Thank You