**Serverless**

**Lambda**

1. History of Cloud
   1. Data Centers (Rackspace) => IaaS (EC2) => PasS (Elastic Beanstalk) => Container (Docker) => Serverless
2. What is Lambda
3. Use cases of Lambda
   1. Event Driven => Asynchronous
   2. Handle http request (API Gateway) => Synchronous
4. Supported Languages (7 languages)
   1. Node.js
   2. Python
   3. Java
   4. C#
   5. Ruby
   6. GO
   7. PowerShell
5. Price => Number of requests / Duration / Memory allocation
6. Lambda features
   1. Scale out automatically
   2. Each event => Each invocation
   3. Lambda can trigger other lambdas
   4. X-ray is used to debug Lambdas

**Hand-on**

Simple

**Build an Alexa Skill – Demo**

Simple

**Serverless Application Model**

1. What is SAM
   1. CloudFormation extension, Supports anything CloudFormation
   2. Run serverless applications locally
   3. Package and deploy serverless applications
2. Common commands
   1. sam init
   2. sam build
   3. sam deploy – (samconfig.toml, the deploy config is saved here)

**Elastic Container Service (ECS)**

1. Containers and Docker
   1. Provides isolation and faster starts than VMs
   2. Portable and offers consistent environment
2. What is ECS – Managed Container Orchestration Server
   1. Create clusters to manage fleets of container deployments
      1. On EC2 instances
      2. Or on Fargate instances
   2. Schedule containers for optimal placement
   3. Defines rules for CPU and memory requirements
   4. Monitor resource utilization
   5. Easy to deploy, update, or rollback containers
   6. ECS is free – Charge is on containers and Fargate tasks
   7. Integrate with VPC, security groups, EBS volumes
   8. Integrate with ELB
   9. Integrate with CloudTrail and CloudWatch
3. ECS 6 basic Components
   1. Cluster – Logical collection of ECS resources, either EC2 or Fargate instances
   2. Task Definition – Define the tasks, can contain multiple containers
   3. Container Definition – Inside a task definition, it defines the individual containers a task uses, controls CPU and memory allocations and port mappings
   4. Task – A single running copy of any containers. A task can be standalone without a service - <https://docs.aws.amazon.com/AmazonECS/latest/userguide/ecs_run_task.html>
   5. Service – Allow task definitions to scale by adding tasks, Defines minimum and maximum values (Service tries to maintain desired number of running tasks)
   6. Registry – ECR (Elastic Container Registry), stores the container images
4. Fargate – Serverless Container Engine
   1. Works with both ECS and EKS
   2. Why use EC2 over Fargate
      1. Compliance Requirements
      2. Require more customization
      3. Requires GPU
5. EKS – Elastic Kubernetes Service
   1. Open source – K8s
   2. Same toolset on premises and cloud
   3. Containers are grouped in **pods (Task in ECS)**
   4. Support both EC2 and Fargate (Like ECS)
   5. Why use EKS
      1. Already using EKS
      2. Easy migrate to AWS environment
6. ECR - Managed Docker container Registry
   1. Integrate with ECS and EKS
   2. Work with on-premises deployments
   3. Multiple AZ
   4. Integrate with IAM
   5. Pay for storage and date transfer (Similar to S3)
7. ECS + ELB
   1. ALB, ELB, Class LB
   2. Supported by both EC2 and Fargate launch types
   3. ALB is recommended over NLB and CLB
8. Security
   1. Instance Role – Role applies to instance (Not ideal)
   2. Task Role – Role applies to tasks (Better – least privilege)

**Summary**