1. Global Secondary Index vs Local Secondary Index
2. SWF Faq

**Deployment**

1. Elastic Beanstalk
2. Cloud Formation
3. Serverless Application Model
4. AWS Lambda

**Serverless**

* Compute – AWS Lambda
* API Proxy – API Gateway
* Storage – S3
* Database – DynamoDB
* Interprocess Messaging – Amazon SNS /SQS
* Orchestration – AWS Step Function
* Analytics – Amazon Kinesis / Amazon Athena
* Developer Tools – Tooling

**Event Sources that trigger AWS Lambda**

**Data Stores**

**Amazon S3**

**Amazon DynamoDb**

**Amazon Kinesis**

**Amazon Cognito**

**Endpoints**

**Amazon API Gateway**

**AWS IOT**

**AWS Step Functions**

**Amazon Alexa**

**Development and Management Tools**

**Amazon CloudFormation**

**Cloud Trail**

**Code Commit**

**Amazon CloudWatch**

**Event/ Message Service**

**Amazon SES**

**Amazon SQS**

**Amazon SNS**

**Cron Events**

**Dynamo DB**

**Strong Consistency Reads – 4KB**

**Eventual Consistency Reads – 8KB**

**Strong Writes – 1KB**

**Eventual Consistency Writes – 2KB**

**1RCU = 4Kb**

1. **Question – One Item = 5KB, Read 420 items every minute using strong consistency read. RCU?**

**1RCU = 4KB**

**2RCU = 5KB**

**Items per second = 420/60 = 7**

**Total RCU per second = 7\*2 = 14**

1. **Question – Item – 2KB , 100 items every second WCU = ?**

**1WCU = 1KB**

**2WCU/item = 2KB**

**Items per second = 100**

**Total WCU = 2 \* 100 = 200**

1. **Question – 1200/min (Eventual Read) , Request Size = 2.5KB. RCU?**

**1RCU = 2 \* 4KB**

**1RCU/item = 2.5KB (provides 2 reads per second for 4kb size per item)**

**Items per second = 1200/60 = 20**

**Total = 1RCU \* 20 = 20 RCU /2 = 10**

**DynamoDB – RCU, Strong Consistency Read, Eventual Consistency Read, Write Consistency Units, Read Write Capacity mode**

**DynamoDB - DAX – DynamoDB accelerator**

**DynamoDB – projection expression , getitem, query and scan**

**DynamoDB - Local Secondary Index vs Global Secondary Index**

**DynamoDB – TTL**

**DynamoDB – Programming errors**

**DynamoDB – Streams Lambda**

**DynamoDB – API Reference – BatchGetItem**

**Lambda – Provisioned vs Reserved Concurrency, Concurrency, set concurrency limits**

**SAM – Canary/Linear/All at Once, deploying serverless gradually**

**Lambda – lambda function to access amazon RDS in an Amazon VPC**

**Lambda – Publish Version**

**Lambda – Environment Variables**

**Lambda – Function Metrics**

**Lambda – IteratorAge**

**Lambda – Lambda with Kinesis**

**SAM – Policy Templates**

**API Gateway – Canary Release deployments, setup**

**API Gateway – Enable API caching to Enhance responsiveness**

**API Gateway – Usage Plans for Amazon API Gateway**

**API Gateway – Throttle API request for better throughput**

**API Gateway – Enable Cors**

**KMS – symmetric CMK, Asymmetric CMK , RSA and ECC**

**KMS – Track requests that dynamodb sends to KMS by tracking in cloudtrail**

**KMS – Envelope Encryption**

**KMS – Encryption SDK**

**KMS - Decrypt**