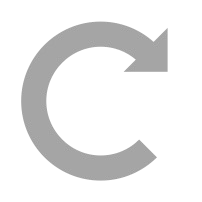
## KMS Keys Types

* KMS Keys is the new name of KMS Customer *Master* Key
* Symmetric (AES-256 keys)
  + Single encryption key that is used to Encrypt and Decrypt
  + AWS services that are integrated with KMS use Symmetric CMKs
  + You never get access to the KMS Key unencrypted (must call KMS API to use)
* Asymmetric (RSA & ECC key pairs)
  + Public (Encrypt) and Private Key (Decrypt) pair
  + Used for Encrypt/Decrypt, or Sign/Verify operations
  + The public key is downloadable, but you can’t access the Private Key unencrypted
  + Use case: encryption outside of AWS by users who can’t call the KMS API

## AWS KMS (Key Management Service)

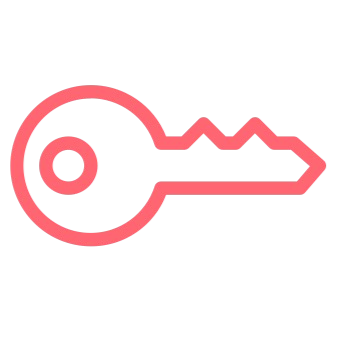
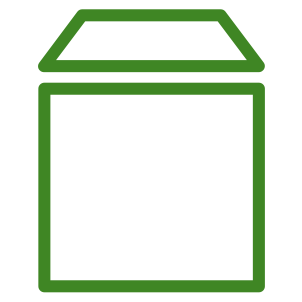
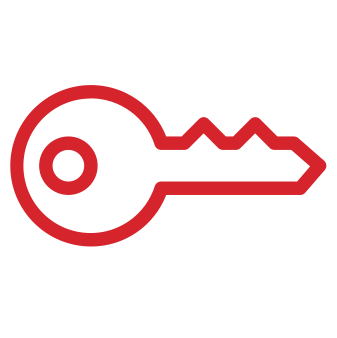
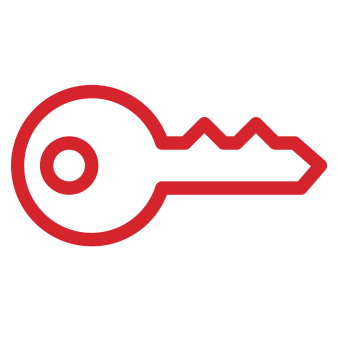
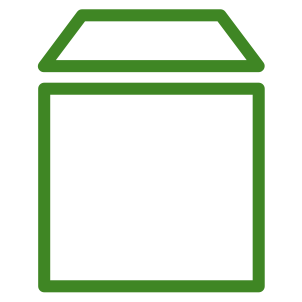
* Three types of KMS Keys:
  + AWS Managed Key: free (aws/*service-name*, example: aws/rds or aws/ebs)
  + Customer Managed Keys (CMK) created in KMS: $1 / month
  + Customer Managed Keys imported (must be 256-bit symmetric key): $1 / month

• + pay for API call to KMS ($0.03 / 10000 calls)

 • Automatic Key rotation:

* + AWS-managed KMS Key: automatic every 1 year
  + Customer-managed KMS Key: (must be enabled) automatic every 1 year
  + Imported KMS Key: only manual rotation possible using alias

## Copying Snapshots across regions



**Region eu-west-2**

**Region ap-southeast-2**

**EBS Volume Encrypted With KMS**

KMS Key A

**EBS Volume Encrypted With KMS**

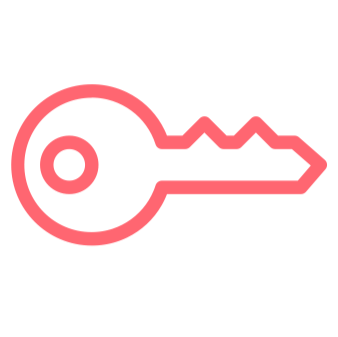
KMS Key B

**EBS Snapshot Encrypted With KMS**

KMS Key A

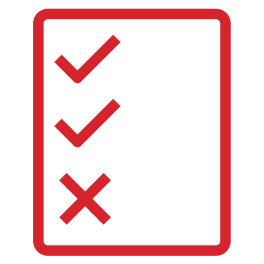
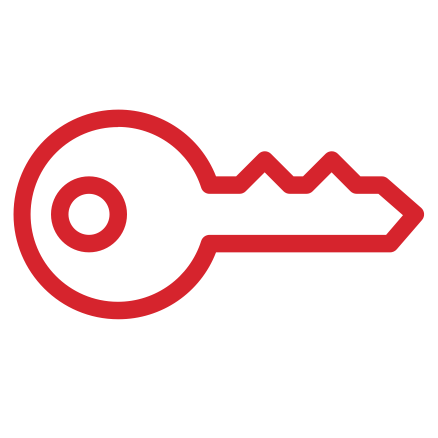
**EBS Snapshot Encrypted With KMS**

KMS Key B



**KMS ReEncrypt with KMS Key B**

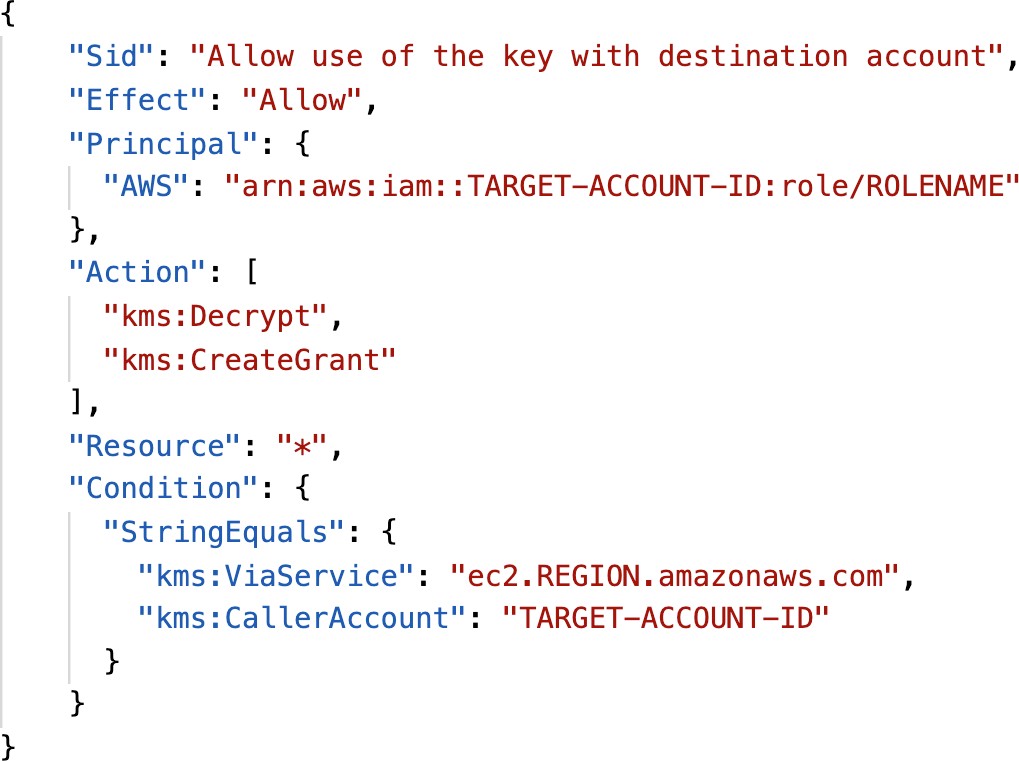
## KMS Key Policies



* Control access to KMS keys, “similar” to S3 bucket policies
* Difference: you cannot control access without them
* Default KMS Key Policy:
  + Created if you don’t provide a specific KMS Key Policy
  + Complete access to the key to the root user = entire AWS account
* Custom KMS Key Policy:
  + Define users, roles that can access the KMS key
  + Define who can administer the key
  + Useful for cross-account access of your KMS key

## Copying Snapshots across accounts

1. Create a Snapshot, encrypted with your own KMS Key (Customer Managed Key)



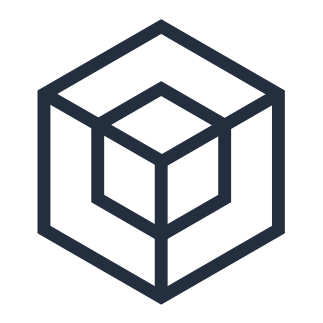
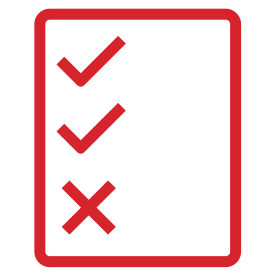
1. Attach a KMS Key Policy to authorize cross-account access
2. Share the encrypted snapshot
3. (in target) Create a copy of the Snapshot, encrypt it with a CMK in your account
4. Create a volume from the snapshot

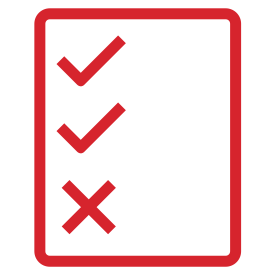
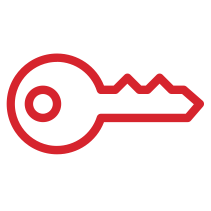
**KMS Key Policy**

## How does KMS work?



**Encrypted Data**

API – Encrypt and Decrypt



**Encrypt** API

**KMS**

Check IAM permissions

**CMK**

Perform encryption

**IAM**

Send encrypted secret

**Decrypt** API

**CMK**

Perform decryption

Check IAM permissions

**IAM**

Send decrypted secret

Decrypted Secret (in plaintext)

Secret (ex: password)

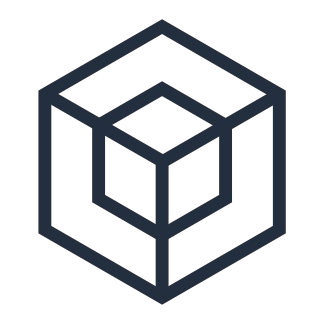
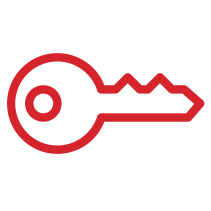
< 4 KB

## Envelope Encryption

* + KMS Encrypt API call has a limit of 4 KB
  + If you want to encrypt >4 KB, we need to use Envelope Encryption
  + The main API that will help us is the GenerateDataKey API
  + For the exam: anything over 4 KB of data that needs to be encrypted must use the Envelope Encryption == GenerateDataKey API

## Deep dive into Envelope Encryption

### GenerateDataKey API



**GenerateDataKey** API

**KMS**

Encrypted DEK

Client side encryption Using DEK

Send plaintext data key Send encrypted data key

Big File (ex: 10MB)

**CMK**

Generate Data Key

Encrypt Data Key with CMK

Check IAM permissions

**IAM**

Plaintext DEK

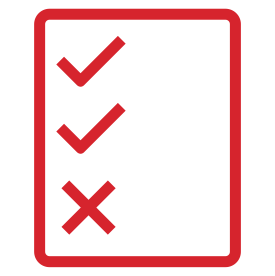
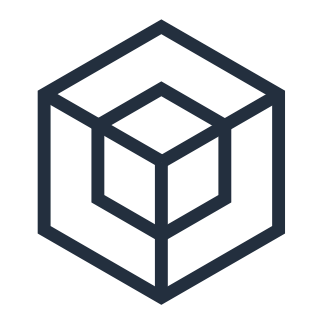
Final file

Encrypted File

## Deep dive into Envelope Encryption

### Decrypt envelope data

**KMS**

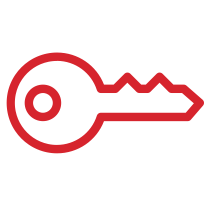


**Decrypt** API

Encrypted DEK

Envelope file

Encrypted File



**CMK**

Check IAM permissions



**IAM**

Send plaintext data key



Plaintext DEK

Client side decryption Using DEK

Decrypt data key using CMK

Decrypted Big File

## Encryption SDK

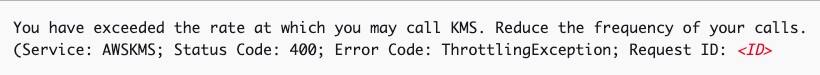
* + The AWS Encryption SDK implemented Envelope Encryption for us
  + The Encryption SDK also exists as a CLI tool we can install
  + Implementations for Java, Python, C, JavaScript
  + Feature - Data Key Caching:
    - re-use data keys instead of creating new ones for each encryption
    - Helps with reducing the number of calls to KMS with a security trade-off
    - Use LocalCryptoMaterialsCache (max age, max bytes, max number of messages)

## KMS Symmetric – API Summary

* + Encrypt: encrypt up to 4 KB of data through KMS
  + GenerateDataKey: generates a unique symmetric data key (DEK)
    - returns a plaintext copy of the data key
    - AND a copy that is encrypted under the CMK that you specify
  + GenerateDataKeyWithoutPlaintext:
    - Generate a DEK to use at some point (not immediately)
    - DEK that is encrypted under the CMK that you specify (must use Decrypt later)
  + Decrypt: decrypt up to 4 KB of data (including Data Encryption Keys)
  + GenerateRandom: Returns a random byte string

## KMS Request Quotas

* + When you exceed a request quota, you get a ThrottlingException:



* + To respond, use exponential backoff (backoff and retry)
  + For cryptographic operations, they share a quota
  + This includes requests made by AWS on your behalf (ex: SSE-KMS)
  + For GenerateDataKey, consider using DEK caching from the Encryption SDK
  + You can request a Request Quotas increase through API or AWS support

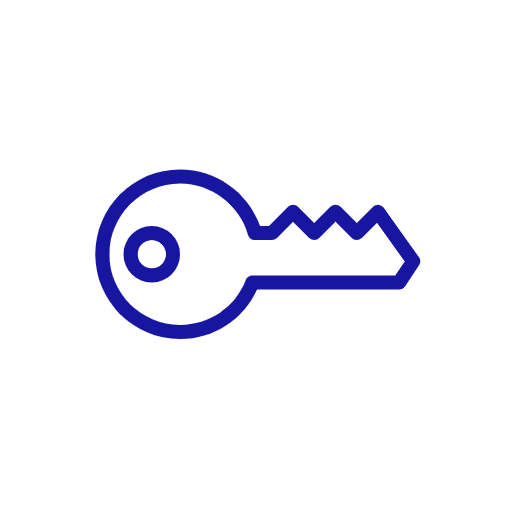
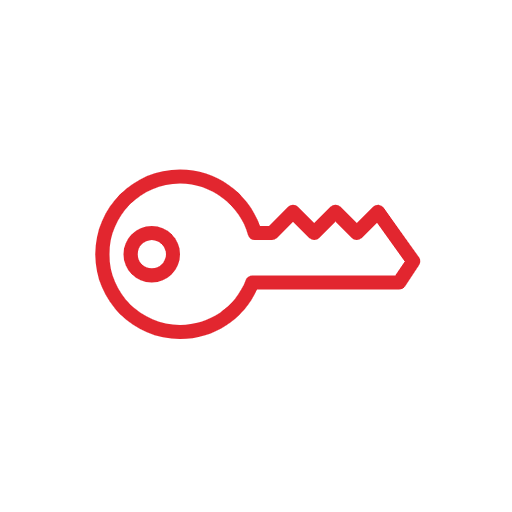
## KMS Request Quotas

|  |  |
| --- | --- |
| **API operation** | **Request quotas (per second)** |
| Decrypt Encrypt  GenerateDataKey (symmetric)  GenerateDataKeyWithoutPlaintext (symmetric) GenerateRandom  ReEncrypt  Sign (asymmetric) Verify (asymmetric) | These shared quotas vary with the AWS Region and the type of CMK used in the request. Each quota is calculated separately.  **Symmetric CMK quota:**   * 5,500 (shared) * 10,000 (shared) in the following Regions:   + us-east-2, ap-southeast-1, ap-southeast-2,   ap-northeast-1, eu-central-1, eu-west-2   * 30,000 (shared) in the following Regions:   + us-east-1, us-west-2, eu-west-1 |
|  | **Asymmetric CMK quota:**   * 500 (shared) for RSA CMKs * 300 (shared) for Elliptic curve (ECC) CMKs |

S3 Bucket Key for SSE-KMS encryption

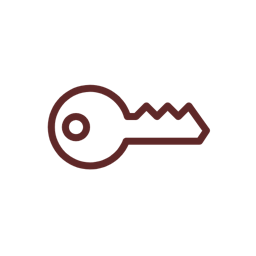
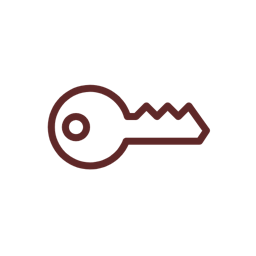
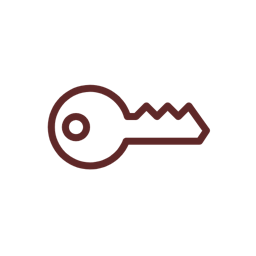
* + New setting to decrease…
    - Number of API calls made to KMS from S3 by 99%
    - Costs of overall KMS encryption with Amazon S3 by 99%

Amazon S3 AWS KMS



* + This leverages data keys
    - A “S3 bucket key” is generated
    - That key is used to encrypt KMS objects with new data keys
  + You will see less KMS CloudTrail events in CloudTrail

S3 Bucket key



Data keys

Customer master key



S3 Bucket

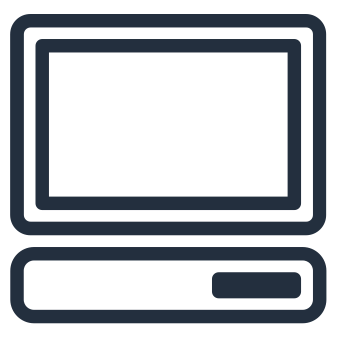
## SSM Parameter Store

* + Secure storage for configuration and secrets
  + Optional Seamless Encryption using KMS
  + Serverless, scalable, durable, easy SDK
  + Version tracking of configurations / secrets
  + Security through IAM
  + Notifications with Amazon EventBridge
  + Integration with CloudFormation

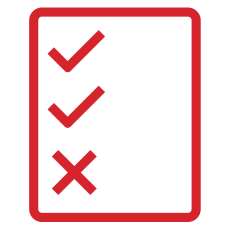
Plaintext configuration

**Applications**

Encrypted configuration



**SSM Parameter Store**



**Check IAM permissions**

Decryption Service

**AWS KMS**



## SSM Parameter Store Hierarchy

* + /my-department/
    - my-app/
      * dev/
        + db-url
        + db-password
      * prod/
        + db-url
        + db-password
    - other-app/
  + /other-department/

GetParameters or GetParametersByPath API

**Dev Lambda Function**

**Prod Lambda Function**

* + */aws/reference/secretsmanager/secret\_ID\_in\_Secrets\_Manager*
  + */aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86\_64-gp2 (public)*

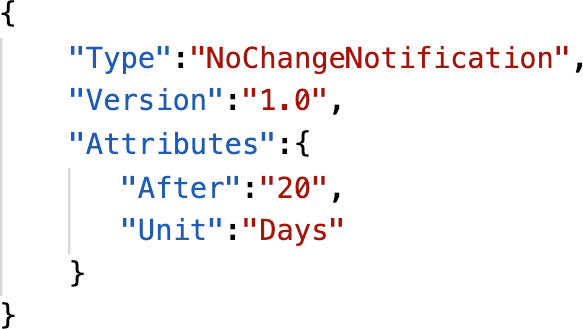
## Standard and advanced parameter tiers

|  |  |  |
| --- | --- | --- |
| **Standard** | | **Advanced** |
| Total number of parameters | 10,000 | 100,000 |
| allowed |  |  |
| (per AWS account and |  |  |
| Region) |  |  |
| Maximum size of a parameter value | 4 KB | 8 KB |
| Parameter policies available | No | Yes |
| Cost | No additional charge | Charges apply |
| Storage Pricing | Free | $0.05 per advanced parameter per month |

Parameters Policies (for advanced parameters)

* + Allow to assign a TTL to a parameter (expiration date) to force updating or deleting sensitive data such as passwords
  + Can assign multiple policies at a time

**Expiration (to delete a parameter) ExpirationNotification (EventBridge) NoChangeNotification (EventBridge)**



## AWS Secrets Manager

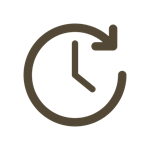
* + Newer service, meant for storing secrets
  + Capability to force rotation of secrets every X days
  + Automate generation of secrets on rotation (uses Lambda)
  + Integration with Amazon RDS (MySQL, PostgreSQL, Aurora)
  + Secrets are encrypted using KMS
  + Mostly meant for RDS integration

## SSM Parameter Store vs Secrets Manager

* + Secrets Manager ($$$):
    - Automatic rotation of secrets with AWS Lambda
    - Lambda function is provided for RDS, Redshift, DocumentDB
    - KMS encryption is mandatory
    - Can integration with CloudFormation
  + SSM Parameter Store ($):
    - Simple API
    - No secret rotation (can enable rotation using Lambda triggered by CW Events)
    - KMS encryption is optional
    - Can integration with CloudFormation
    - Can pull a Secrets Manager secret using the SSM Parameter Store API

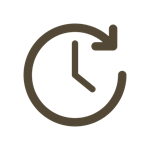
## SSM Parameter Store vs. Secrets Manager Rotation

AWS Secrets Manager SSM Parameter Store



every 30 days

invoke



every 30 days

invoke

AWS Secrets Manager



change password

change value

Lambda Function (can be provided)

change password



CloudWatch Events Lambda Function

SSM Parameter Store

Amazon RDS

Amazon RDS

## CloudWatch Logs - Encryption **+**

* + You can encrypt CloudWatch logs with KMS keys
  + Encryption is enabled at the log group level, by associating a CMK with a log group, either when you create the log group or after it exists.
  + You cannot associate a CMK with a log group using the CloudWatch console.
  + You must use the CloudWatch Logs API:
    - associate-kms-key : if the log group already exists
    - create-log-group: if the log group doesn’t exist yet

## CodeBuild Security **+**

* + To access resources in your VPC, make sure you specify a VPC configuration for your CodeBuild
  + Secrets in CodeBuild:
  + Don’t store them as plaintext in environment variables
  + Instead…
    - Environment variables can reference parameter store parameters
    - Environment variables can reference secrets manager secrets

# Other AWS Services

Quick overview of other services that might have questions on at the exam

Amazon Simple Email Service (Amazon SES)

* + Fully managed service to send emails securely, globally and at scale
  + Allows inbound/outbound emails
  + Reputation dashboard, performance insights, anti-spam feedback
  + Provides statistics such as email deliveries, bounces, feedback loop results, email open
  + Supports DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF)
  + Flexible IP deployment: shared, dedicated, and customer-owned IPs
  + Send emails using your application using AWS Console, APIs, or SMTP
  + Use cases: transactional, marketing and bulk email communications

Users

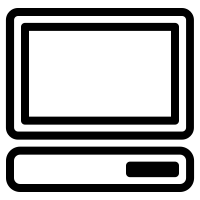
**bulk emails**



Amazon SES

**APIs**

or **SMTP**



Application

## AWS Databases Summary

* + RDS: Relational databases, OLTP
    - PostgreSQL, MySQL, Oracle…
    - Aurora + Aurora Serverless
    - Provisioned database
  + DynamoDB: NoSQL DB
    - Managed, Key Value, Document
    - Serverless
  + ElastiCache: In memory DB
    - Redis / Memcached
    - Cache capability
* Redshift: OLAP – Analytic Processing
  + Data Warehousing / Data Lake
  + Analytics queries
* Neptune: Graph Database
* DMS: Database Migration Service
* DocumentDB: managed MongoDB for AWS

## AWS Certificate Manager (ACM)

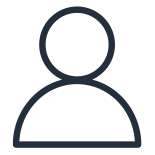
* + Let’s you easily provision, manage, and deploy

SSL/TLS Certificates

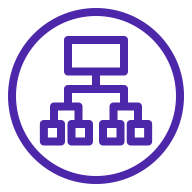
* + Used to provide in-flight encryption for websites (HTTPS)
  + Supports both public and private TLS



provision and



**HTTPS**



Application

certificates

* + Free of charge for public TLS certificates
  + Automatic TLS certificate renewal
  + Integrations with (load TLS certificates on)
    - Elastic Load Balancers
    - CloudFront Distributions
    - APIs on API Gateway

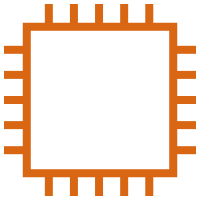
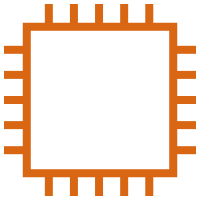
maintain TLS certs

AWS Certificate Manager

HTTP

Load Balancer

Auto Scaling group



EC2 Instance

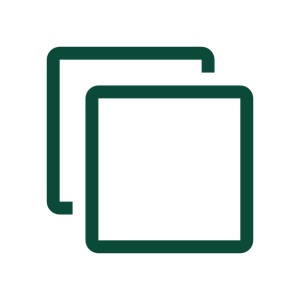
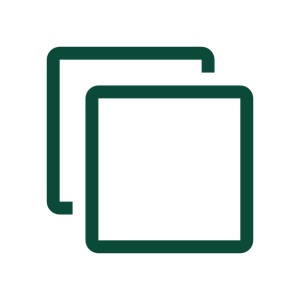
EC2 Instance

## AWS Cloud Map

* + A fully managed resource discovery service
  + Creates a map of the backend services/resources that your applications depend on
  + You register your application components, their locations, attributes, and health status with AWS Cloud Map
  + Integrated health checking (stop sending traffic to unhealthy endpoints)

Without Cloud Map

code change is required



**Backend service**

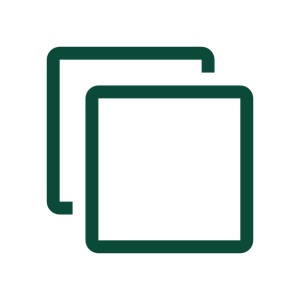
Version 1

Version 2

**Frontend service**

With Cloud Map

**Backend service**



**Frontend service**

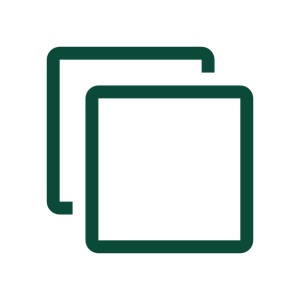
* + Your applications can query AWS Cloud Map using AWS SDK, API, or DNS

1. Dynamic lookup to find the location

of v2

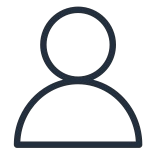
1. Connect to v2

Version 1



Version 2

AWS Cloud Map **1.** Update endpoint



from v1 to v2

## AWS Fault Injection Simulator (FIS)

* + A fully managed service for running fault injection experiments on AWS workloads
  + Based on Chaos Engineering – stressing an application by creating disruptive events (e.g., sudden increase in CPU or memory), observing how the system responds, and implementing improvements
  + Helps you uncover hidden bugs and performance bottlenecks
  + Supports the following AWS services: EC2, ECS, EKS, RDS…
  + Use pre-built templates that generate the desired disruptions

Monitoring

**CloudWatch**

**EventBridge**

View Results

**(identify performance, observability,**

**or resiliency issues)**

Stop Experiment

**(stop if complete or an alarm is triggered)**

Experiment Template

AWS Fault Injection Simluator

start

create

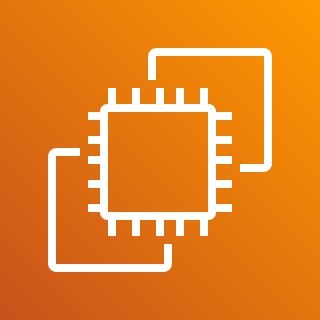
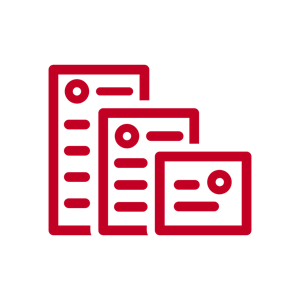
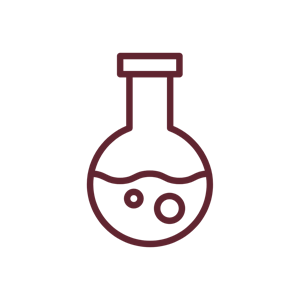
**RDS**

**EKS**

**ECS**

**EC2**

Resources



## AWS DataSync

* + Move large amount of data to and from
    - On-premises / other cloud to AWS (NFS, SMB, HDFS, S3 API…) – needs agent
    - AWS to AWS (different storage services) – no agent needed
  + Can synchronize to:
    - Amazon S3 (any storage classes – including Glacier)
    - Amazon EFS
    - Amazon FSx (Windows, Lustre, NetApp, OpenZFS...)
  + Replication tasks can be scheduled hourly, daily, weekly
  + File permissions and metadata are preserved (NFS POSIX, SMB…)
  + One agent task can use 10 Gbps, can setup a bandwidth limit

## AWS DataSync

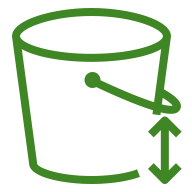
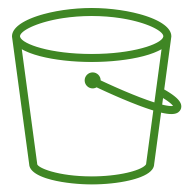
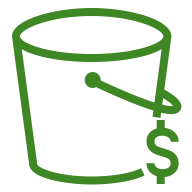
### NFS / SMB to AWS (S3, EFS, FSx…)

Region

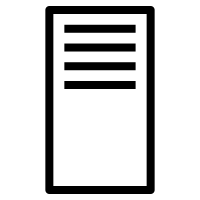
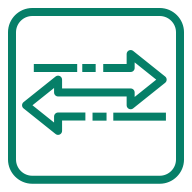


On-Premises

AWS Storage Resources



NFS or SMB

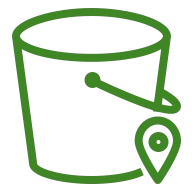
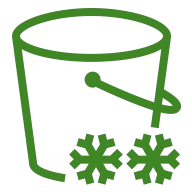
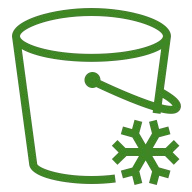


TLS

S3 Standard-IA

S3 Intelligent- Tiering

S3 Standard



NFS or SMB

Server

AWS DataSync Agent

AWS

S3 Glacier Deep Archive

S3 Glacier

S3 One Zone-IA



DataSync



AWS Snowcone (agent pre-installed)

Amazon FSx

AWS EFS

## AWS DataSync

Transfer between AWS storage services



Amazon S3

Amazon EFS



Amazon FSx



Amazon S3

Amazon EFS



Amazon FSx





AWS DataSync

copy data and metadata between AWS Storage Services

Exam Review & Tips

©

## State of learning checkpoint

* Let’s look how far we’ve gone on our learning journey
* https://aws.amazon.com/certification/certified-developer-associate/

## Practice makes perfect

* If you’re new to AWS, take a bit of AWS practice thanks to this course before rushing to the exam
* The exam recommends you to have one or more years of hands-on developing and maintaining an AWS based applications
* Practice makes perfect!
* If you feel overwhelmed by the amount of knowledge you just learned, just go through it one more time

## Ideas for practicing…!

* Take one of your existing applications
* Try deploying it manually on EC2
* Try deploying it on Elastic Beanstalk and have it scale
* Try creating a CICD pipeline for it
* Try decoupling components using SQS / SNS
* If possible, try running it on AWS Lambda & friends
* Write automation scripts using the CLI / SDK
  + Idea 1: Shut down EC2 instances at night / start in the morning
  + Idea 2: Automate snapshots of EBS volumes at night
  + Idea 3: List all under-utilized EC2 instances (CPU Utilization < 10%)

## Proceed by elimination

* Most questions are going to be scenario based
* For all the questions, rule out answers that you know for sure are wrong
* For the remaining answers, understand which one makes the most sense
* There are very few trick questions
* Don’t over-think it
* If a solution seems feasible but highly complicated, it’s probably wrong

## Skim the AWS Whitepapers

* You can read about some AWS White Papers here:
  + AWS Security Best Practices
  + AWS Well-Architected Framework
  + Architecting for the Cloud AWS Best Practices
  + Practicing Continuous Integration and Continuous Delivery on AWS Accelerating Software Delivery with DevOps
  + Microservices on AWS
  + Serverless Architectures with AWS Lambda
  + Optimizing Enterprise Economics with Serverless Architectures
  + Running Containerized Microservices on AWS
  + Blue/Green Deployments on AWS
* Overall we’ve explored all the most important concepts in the course
* It’s never bad to have a look at the whitepapers you think are interesting!

## Read each service’s FAQ

* FAQ = Frequently asked questions
* Example: https://aws.amazon.com/lambda/faqs/
* FAQ cover a lot of the questions asked at the exam
* They help confirm your understanding of a service

## Get into the AWS Community

* Help out and discuss with other people in the course Q&A
* Review questions asked by other people in the Q&A
* Do the practice test in this section
* Read forums online
* Read online blogs
* Attend local meetups and discuss with other AWS engineers
* Watch re-invent videos on Youtube (AWS Conference)

## How will the exam work?

* You’ll have to register online at https://[www.aws.training/](http://www.aws.training/)
* Fee for the exam is 150 USD
* Provide one identity documents (ID, Passport, details are in emails sent to you…)
* No notes are allowed, no pen is allowed, no speaking
* 65 questions will be asked in 130 minutes
* Use the “Flag” feature to mark questions you want to re-visit
* At the end you can optionally review all the questions / answers
* To pass you need a score of a least 720 out of 1000
* You will know within 5 days if you passed / failed the exams (most of the time less)
* You will know the overall score a few days later (email notification)
* You will not know which answers were right / wrong
* If you fail, you can retake the exam again 14 days later

Congratulations & Next Steps!

## Congratulations!

* Congrats on finishing the course!
* I hope you will pass the exam without a hitch 
* If you passed, I’ll be more than happy to know I’ve helped
  + Post it in the Q&A to help & motivate other students. Share your tips!
  + Post it on LinkedIn and tag me!
* Overall, I hope you learned how to use AWS and that you will be a tremendously good AWS Developer

## Next Steps

* We’ve spent a lot of time getting an overview of each service
* Each service on its own deserves its own course and study time
* Find out what services you liked and get specialized in them!
* My personal favorites: AWS Lambda, CloudFormation, EC2 & ECS
* Happy learning!