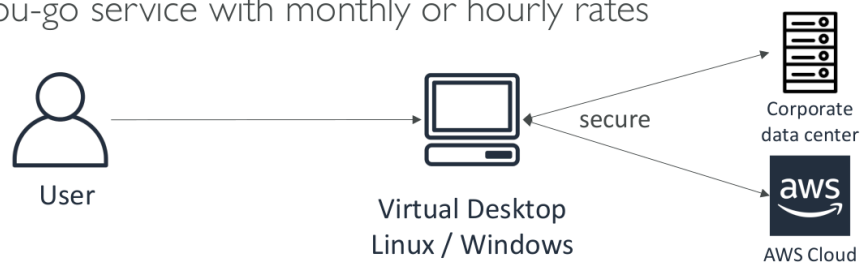


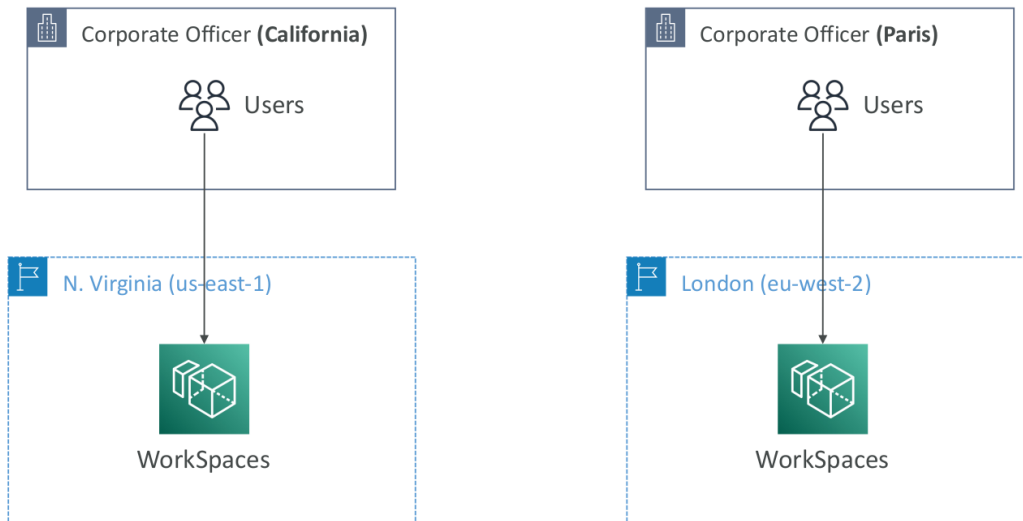
Other Services

WorkSpaces

- The idea is, any user can access a virtual desktop (linux/windows) within AWS through amazon WorkSpaces.
- Managed Desktop as a Service (DaaS) solution to easily provision Windows or Linux desktops
- Great to eliminate management of on-premise VDI (Virtual Desktop Infrastructure)
- Fast and quickly scalable to thousands of users
- Secured data – integrates with KMS
- Pay-as-you-go service with monthly or hourly rates

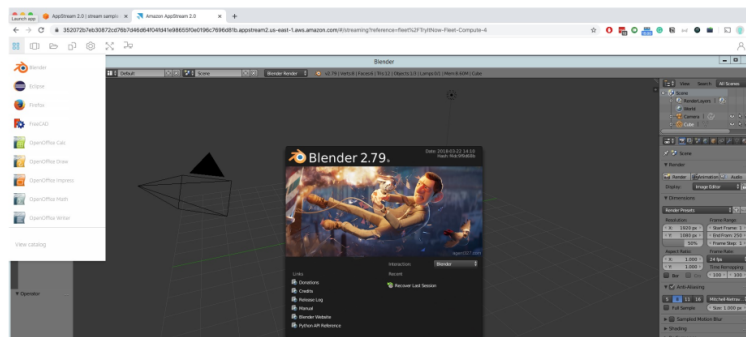


- Also, it is always recommended to deploy your servers close to your users in order to minimize latency. WorkSpaces also allow multi-region deployments for the same reason.



AppStream 2.0

- The idea is that you want to stream to any number of end users without acquiring and provisioning infrastructure. End users can easily access the applications through a web browser interface.
- Desktop Application Streaming Service
- Deliver to any computer, without acquiring, provisioning infrastructure
- The application is delivered from within a web browser



Amazon AppStream 2.0 vs WorkSpaces

- **Workspaces**

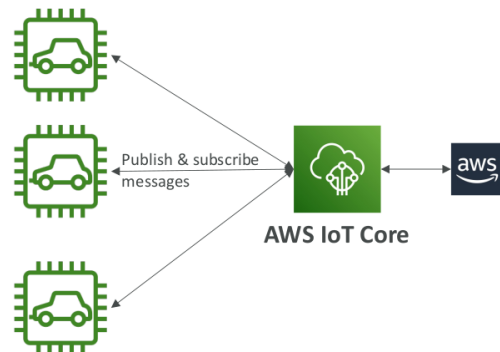
- Fully managed VDI and desktop available
- The users connect to the VDI and open native or WAM applications
- Workspaces are on-demand or always on

- **AppStream 2.0**

- Stream a desktop application to web browsers (no need to connect to a VDI)
- Works with any device (that has a web browser)
- Allow to configure an instance type per application type (CPU, RAM, GPU)

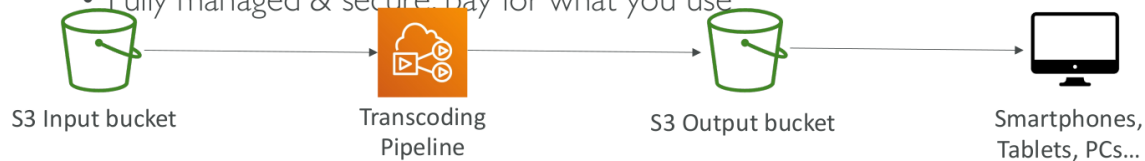
AWS IoT Core

- IoT stands for “Internet of Things” – the network of internet-connected devices that are able to collect and transfer data
- AWS IoT Core allows you to **easily connect IoT devices to the AWS Cloud**
- **Serverless, secure & scalable** to billions of devices and trillions of messages
- Your applications can communicate with your devices even when they aren't connected
- Integrates with a lot of AWS services (Lambda, S3, SageMaker, etc.)
- Build IoT applications that gather, process, analyze, and act on data



Elastic Transcoder

- Elastic Transcoder is used to **convert media files** stored in S3 into media files in the formats required by consumer playback devices (phones etc..)
- Benefits:
 - Easy to use
 - Highly scalable – can handle large volumes of media files and large file sizes
 - Cost effective – duration-based pricing model
 - Fully managed & secure, pay for what you use



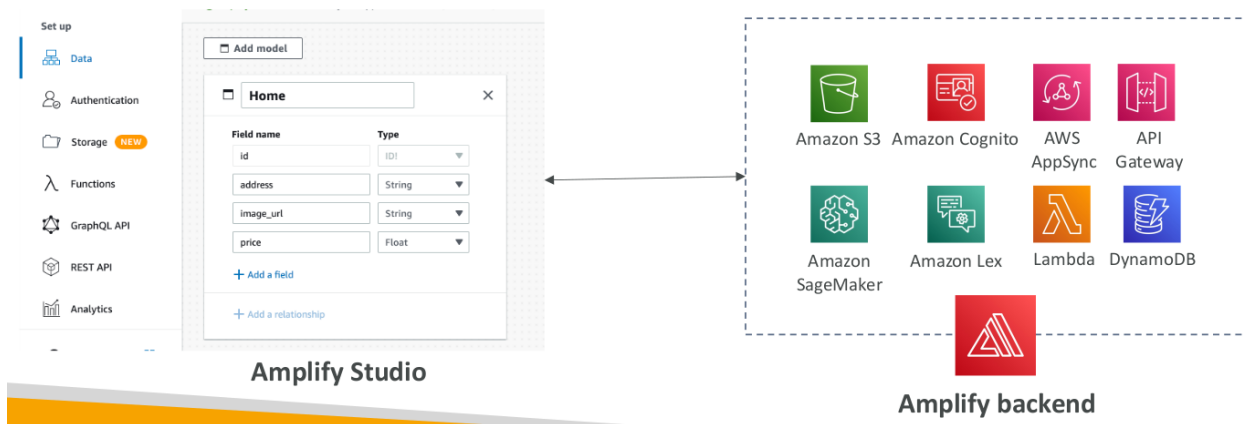
AWS AppSync

- The idea with AppSync is to build a backend for your mobile and web applications.
- Exam = storing data for applications using **GraphQL** = AppSync
Leverages AWS Amplify to build backend in the background = AppSync
- Store and sync data across mobile and web apps in real-time
- Makes use of GraphQL (mobile technology from Facebook)
- Client Code can be generated automatically
- Integrations with DynamoDB / Lambda
- Real-time subscriptions
- Offline data synchronization (replaces Cognito Sync)
- Fine Grained Security
- AWS Amplify can leverage AWS AppSync in the background!



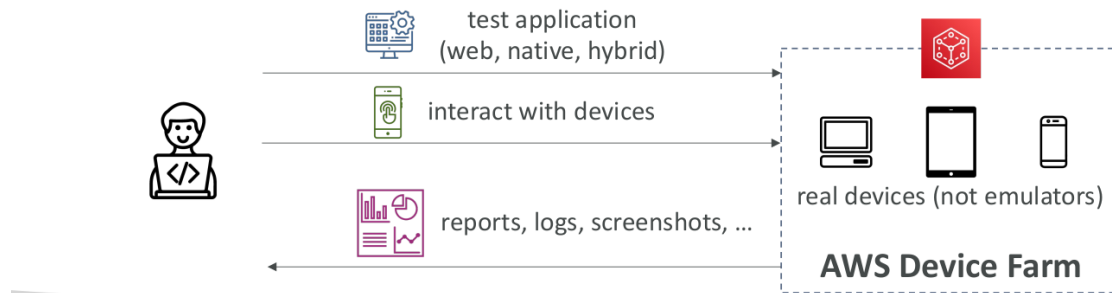
AWS Amplify

- A set of tools and services that helps you develop and deploy scalable full stack web and mobile applications
- Authentication, Storage, API (REST, GraphQL), CI/CD, PubSub, Analytics, AI/ML Predictions, Monitoring, Source Code from AWS, GitHub, etc...
- All at one place. Amplify is like Elastic Beanstalk for web and mobile applications.
- You can set up everything you need within the Amplify Studio on AWS. This will then, get linked to the backend of Amplify which allows you to leverage other AWS services for your application.



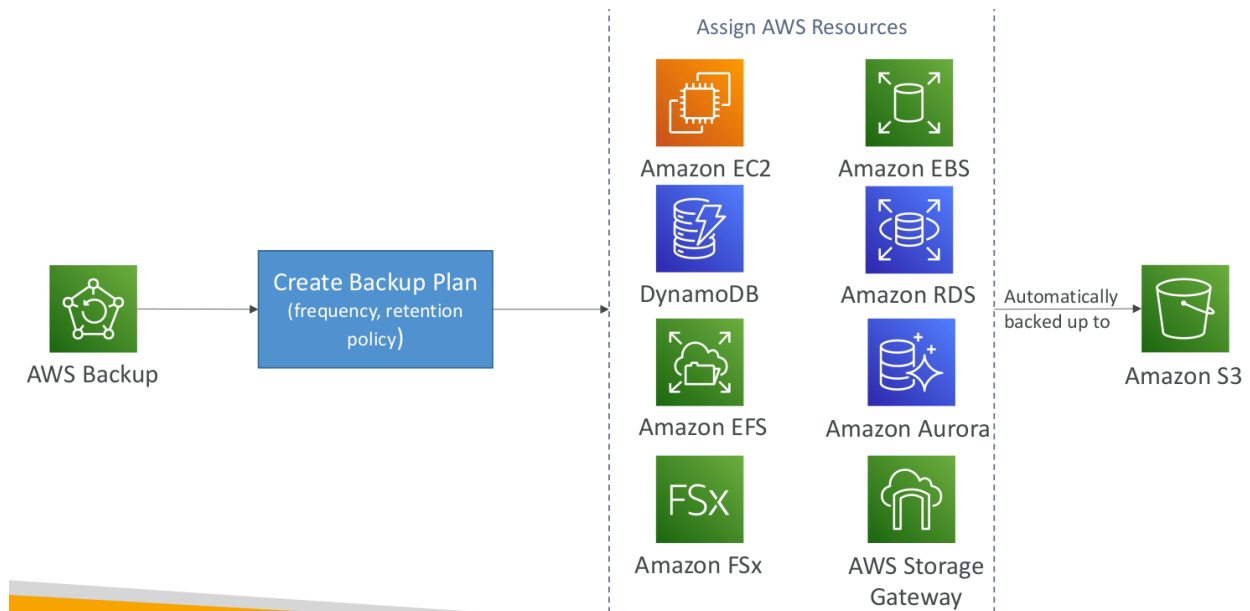
Device Farm

- Fully-managed service that tests your web and mobile apps against desktop browsers, real mobile devices, and tablets
- Run tests concurrently on multiple devices (speed up execution)
- Ability to configure device settings (GPS, language, Wi-Fi, Bluetooth, ...)



AWS Backup

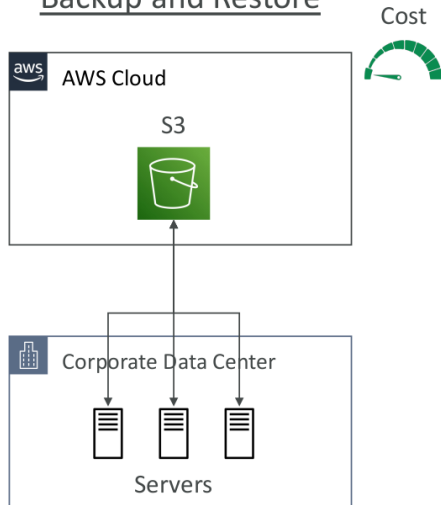
- Fully-managed service to centrally manage and automate backups across AWS services
- On-demand and scheduled backups
- Supports PITR (Point-in-time Recovery)
- Retention Periods, Lifecycle Management, Backup Policies, ...
- Cross-Region Backup
- Cross-Account Backup (using AWS Organizations)



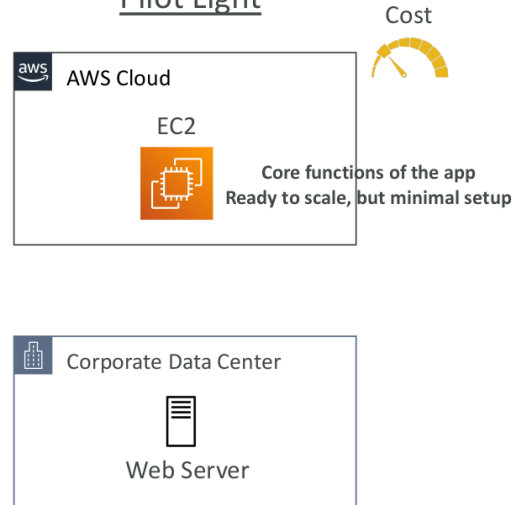
Disaster Recovery Strategies

- **Cost:** Backup & Restore < Pilot Light < Warm Standby < Multi-site or Hot-site
- **Instantly ready to use and scale = Multi-site strategy** as it has the full version of the app running at full size within the cloud in case of any disasters.
- **Not running, just storing in the cloud = Backup & Restore strategy** as it has all the files of your application stored away in a S3 bucket. Minimal cost, not ready to replace the damaged infrastructure instantly.

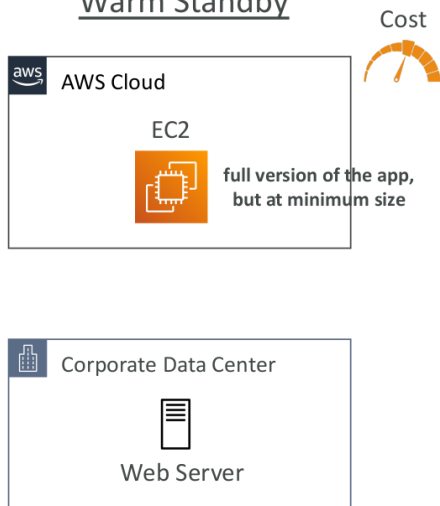
Backup and Restore



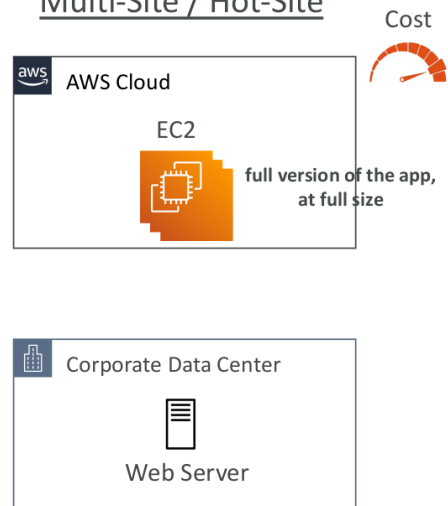
Pilot Light



Warm Standby

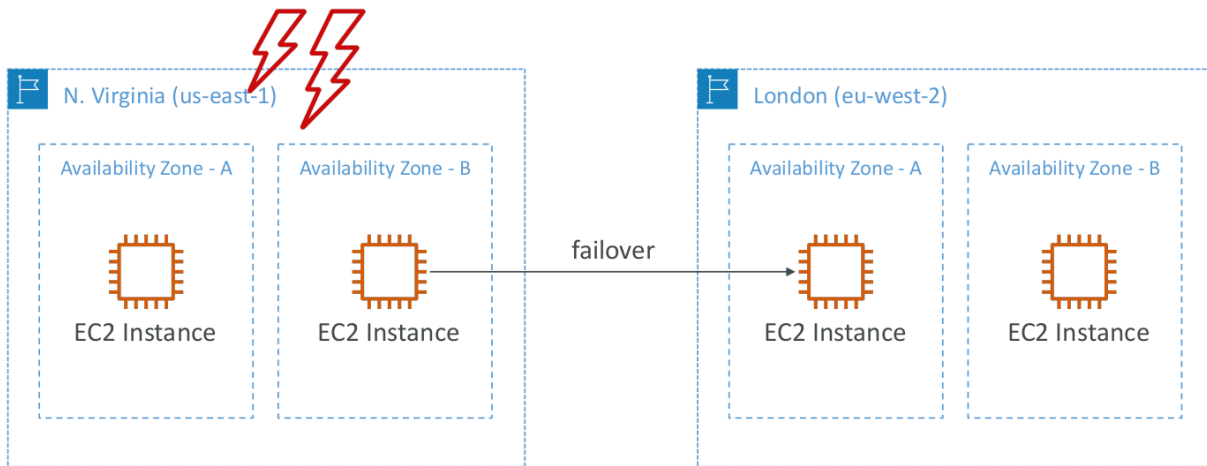


Multi-Site / Hot-Site



- In case of a disaster in one region, the traffic can be directed towards the failover region to take over. This can be done through Route 53.

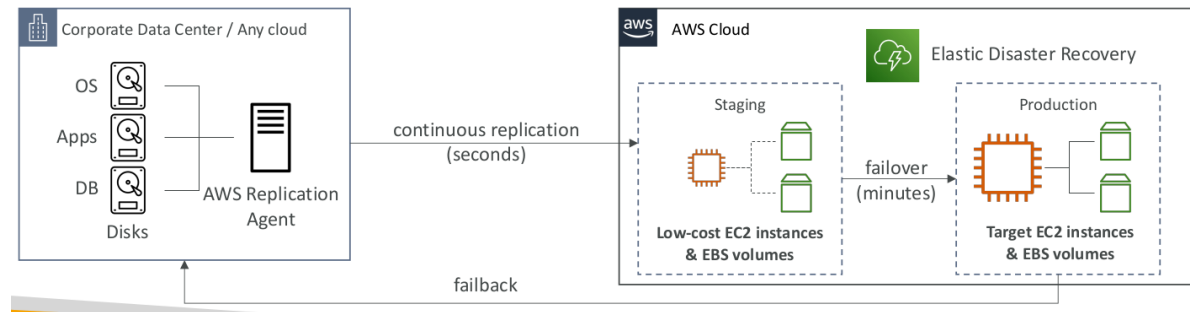
Typical DR Setup for Cloud Deployments



AWS Elastic Disaster Recovery (DRS)

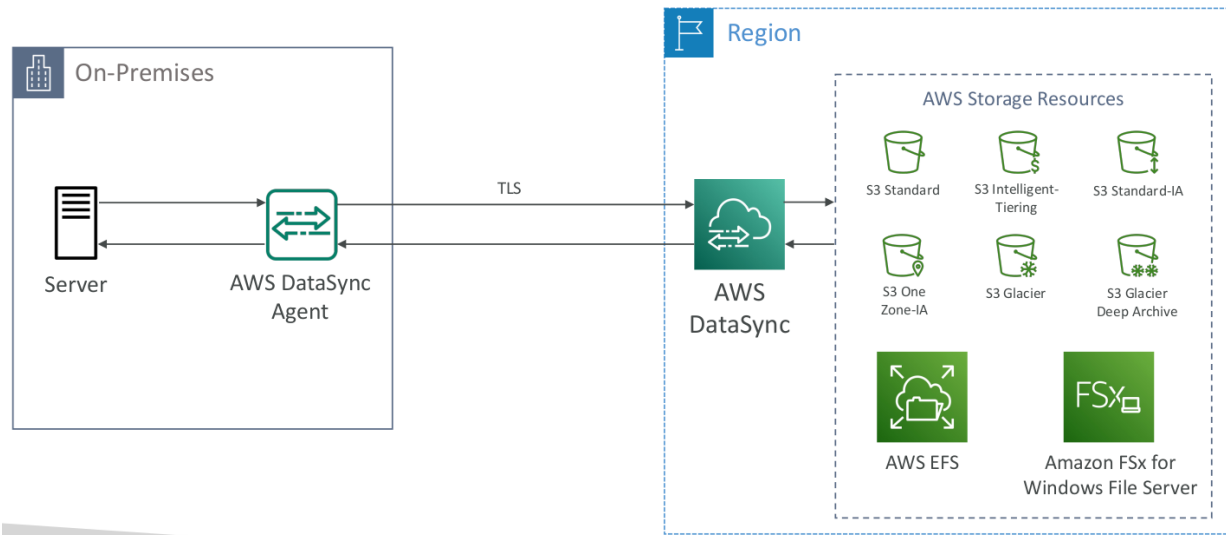
- Used to be named “CloudEndure Disaster Recovery”
- Quickly and easily **recover** your physical, virtual, and cloud-based servers into AWS
- Example: protect your most critical databases (including Oracle, MySQL, and SQL Server), enterprise apps (SAP), protect your data from ransomware attacks, ...
- Continuous block-level replication for your servers
- How this basically works is, **AWS Replication Agent** allows you to continuously replicate your whole infrastructure (from other cloud or on-premises) which then goes through **DRS**. First, it goes into **Staging** where your infrastructure is backed up using cost efficient EC2 instances and EBS volumes. Now, as soon as disaster hits your original data centre, your infrastructure within the staging area moves to the **failover Production** set up in minutes. This allows you to keep running your application even in case of the worst disasters. Then, once your data centre is back online, your infrastrue

goes through the **failback** stage which allows it to run the original infrastructure again.



AWS DataSync

- Move large amount of data from on-premises to AWS
- Can synchronize to: Amazon S3 (any storage classes – including Glacier), Amazon EFS, Amazon FSx for Windows
- Replication tasks can be scheduled hourly, daily, weekly
- The replication tasks are incremental after the first full load



Application Discovery Service

- Plan migration projects by gathering information about on-premises data centers
- Server utilization data and dependency mapping are important for migrations
- **Agentless Discovery (AWS Agentless Discovery Connector)**
 - VM inventory, configuration, and performance history such as CPU, memory, and disk usage
- **Agent-based Discovery (AWS Application Discovery Agent)**
 - System configuration, system performance, running processes, and details of the network connections between systems
- Resulting data can be viewed within AWS Migration Hub

1. Agentless Discovery:

- **Description:** This method relies on existing technologies and protocols to gather information without installing any additional software (agents) on the target systems.
- **How it Works:** Utilizes standard protocols like WMI (Windows Management Instrumentation) for Windows or SSH (Secure Shell) for Linux to retrieve configuration and performance data.
- **Advantages:** Simpler to implement as it doesn't require deploying and managing agents on each system. It's generally less intrusive.

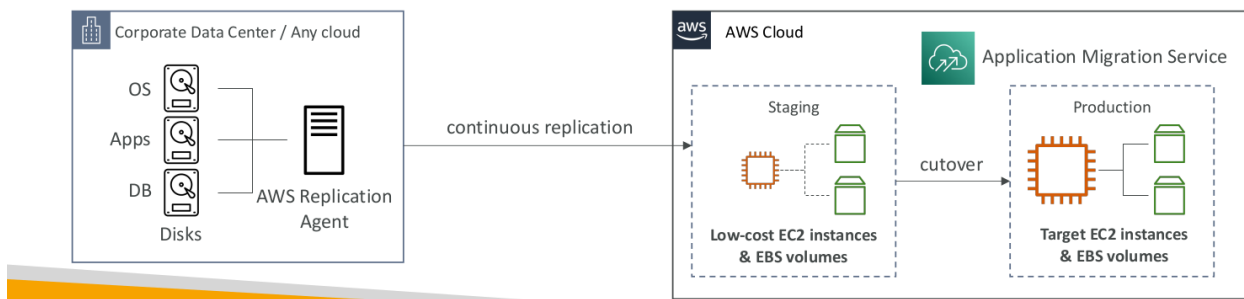
2. Agent-Based Discovery:

- **Description:** Involves installing lightweight agents on the target systems to collect detailed information about applications and dependencies.
- **How it Works:** Agents are deployed on each machine, and they actively gather data about the installed software, processes, and communication patterns to provide a more comprehensive view of the environment.
- **Advantages:** Offers more detailed and accurate insights into the applications and dependencies. Can be particularly useful for complex environments with custom or non-standard applications.

- The simplest way to move from on-premises to AWS is to use:

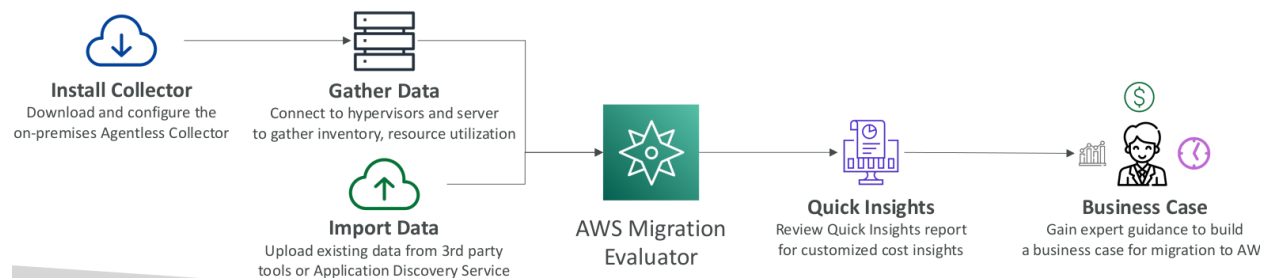
- **Application Migration Service (MGN)**

- The “AWS evolution” of CloudEndure Migration, replacing AWS Server Migration Service (SMS)
- Lift-and-shift (rehost) solution which simplify **migrating** applications to AWS
- Converts your physical, virtual, and cloud-based servers to run natively on AWS
- Supports wide range of platforms, Operating Systems, and databases
- Minimal downtime, reduced costs



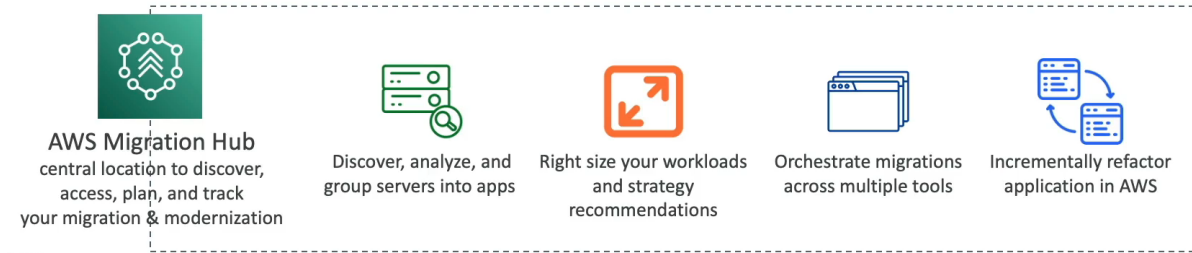
AWS Migration Evaluator

- A tool that helps you understand the potential costs and benefits of moving your on-premises workloads to AWS.
- In simpler terms, it analyzes your existing IT environment, considers how it would work in the AWS cloud, and provides insights into the estimated costs, performance improvements, and other factors you might want to know before making the move.
- It does so by installing Agentless Collector to conduct broad-based discovery, which then takes a snapshot of on-premises footprint, server dependencies and eventually, analyze current state, define target state, then develop migration plan



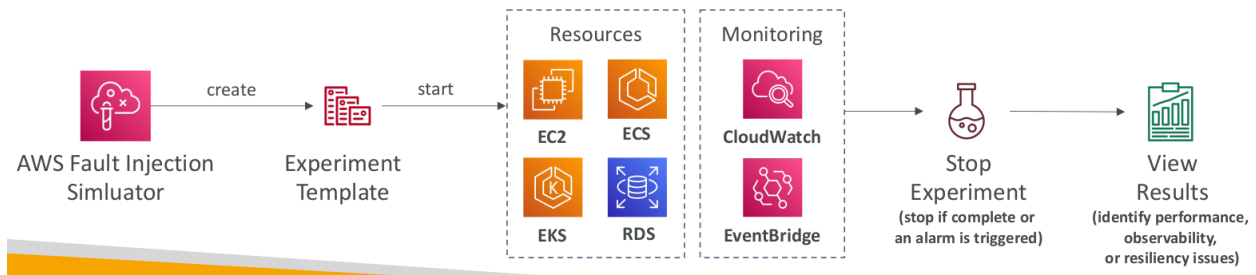
AWS Migration Hub

- Central location to collect servers and applications inventory data for the assessment, planning, and tracking of migrations to AWS
- Helps accelerate your migration to AWS, automate lift-and-shift
- **AWS Migration Hub Orchestrator** – provides pre-built templates to save time and effort migrating enterprise apps (e.g., SAP, Microsoft SQL Server...)
- Supports migrations status updates from **Application Migration Service (MGN)** and **Database Migration Service (DMS)**



Fault Injection Simulator (FIS)

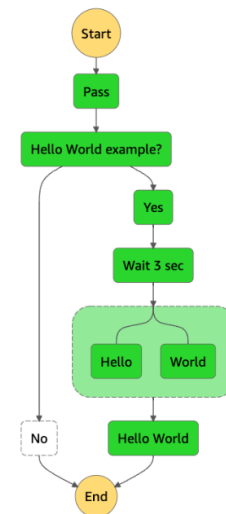
- Fault injection is a testing technique used to assess the resilience and reliability of a system by intentionally introducing faults or errors into the system. The purpose is to evaluate how well the system can detect, tolerate, and recover from these faults or errors.
- Various types of faults, such as software bugs, hardware failures, or network issues, are intentionally introduced to observe the system's behavior under adverse conditions.
- 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



AWS Step Functions

- Workflows in form of flow charts or graphs.
- Build serverless visual workflow to orchestrate your Lambda functions
- **Features:** sequence, parallel, conditions, timeouts, error handling, ...
- Can integrate with EC2, ECS, On-premises servers, API Gateway, SQS queues, etc...
- Possibility of implementing human approval feature
- **Use cases:** order fulfillment, data processing, web applications, any workflow

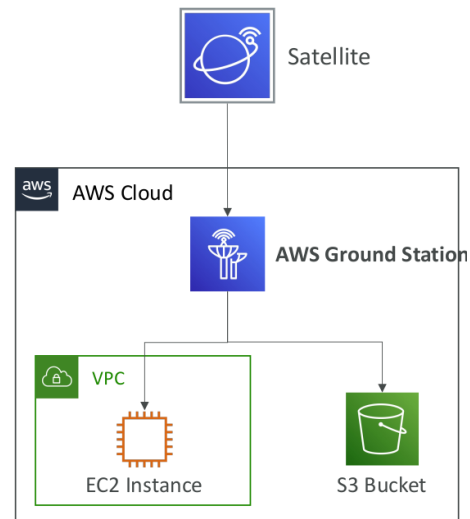
■ In Progress ■ Succeeded ■ Failed ■ Cancelled ■ Caught Error



AWS Ground Station

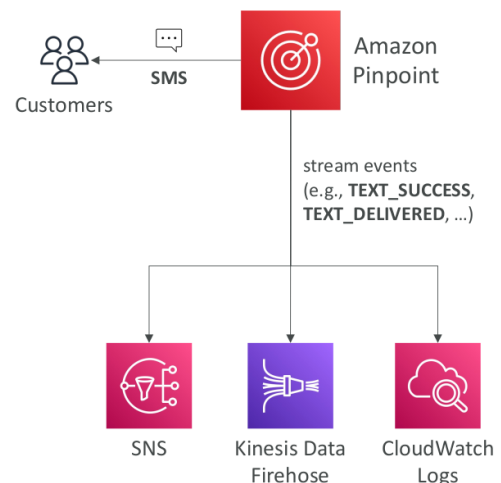
- Satellite data = AWS Ground Station

- Fully managed service that lets you control satellite communications, process data, and scale your satellite operations
- Provides a global network of satellite ground stations near AWS regions
- Allows you to download satellite data to your AWS VPC within seconds
- Send satellite data to S3 or EC2 instance
- Use cases: weather forecasting, surface imaging, communications, video broadcasts



Amazon Pinpoint

- Full blown scalable marketing 2-way communications.
- A level higher to SNS and SES.
- Allows you to target groups of people by defining categories and segments.
- Scalable 2-way (outbound/inbound) marketing communications service
- Supports email, SMS, push, voice, and in-app messaging
- Ability to segment and personalize messages with the right content to customers
- Possibility to receive replies
- Scales to billions of messages per day
- Use cases: run campaigns by sending marketing, bulk, transactional SMS messages
- **Versus Amazon SNS or Amazon SES**
 - In SNS & SES you managed each message's audience, content, and delivery schedule
 - In Amazon Pinpoint, you create message templates, delivery schedules, highly-targeted segments, and full campaigns



Quiz



Good job!

Amazon Elastic Transcoder is media transcoding in the cloud. It is used to convert media files from their source format into versions that will play back on devices like smartphones, tablets, and PCs.

Question 1:

You would like to convert an S3 file so it can be played on users' devices. Which AWS service can help?



Transcribe



Elastic Transcoder



AppStream 2.0



Sumerian

**Good job!**

AWS Device Farm is an application testing service that lets you improve the quality of your web and mobile apps by testing them across an extensive range of desktop browsers and real mobile devices; without having to provision and manage any testing infrastructure.

Question 3:

Which AWS service can be used to test your application across real desktop browsers and mobile devices?



IoT Core



AppStream 2.0



WorkSpaces



Device Farm

**Good job!**

AWS IoT Core lets you securely connect IoT devices to the AWS Cloud and other devices without the need to provision or manage servers.

Question 4:

Which AWS service is serverless and lets you connect billions of devices to the AWS Cloud?



Transit Gateway



Connect



Elastic Transcoder



IoT Core

**Good job!**

AWS Backup is a centralized backup service that makes it easy and cost-effective for you to backup your application data across AWS services in the AWS Cloud. CloudEndure Disaster Recovery minimizes downtime and data loss by providing fast, reliable recovery into AWS of your physical, virtual, and cloud-based servers.

Question 5:

CloudEndure Disaster Recovery is used to centrally automate backups across AWS services while AWS Backup is used to quickly and easily recover servers into AWS.



True



False