Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

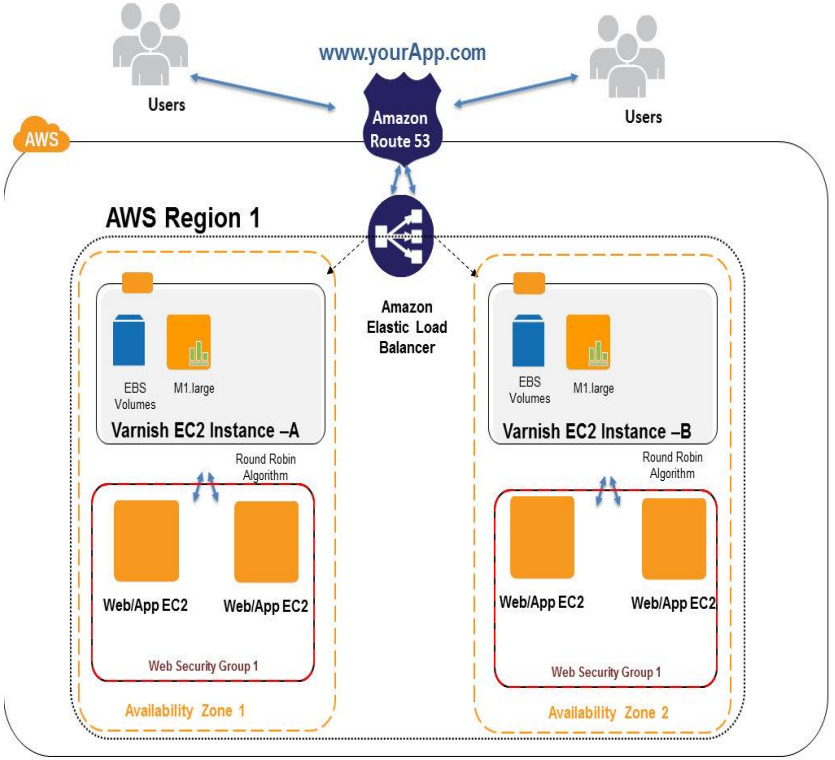
AWS Developer

What is AWS?

- Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.

- The AWS Cloud provides a broad set of infrastructure services, such as computing power, storage options, networking and databases, delivered as a utility: on-demand, available in seconds, with pay-as-you-go pricing.

AWS Regions



AWS Global Presence 16 Regions | 42 AZ

- Region
  - A region is geographical location of the AWS data centre
- Availability Zone
  - Each availability zone is isolated location within the same region

All communication between regions is across the public Internet. Data transfer between regions is charged at the Internet data transfer rate for both the sending and the receiving instance.

AWS Registration

- To use AWS services register with AWS
  - URL : https://aws.amazon.com
- AWS Pricing
  - Pricing varies depending on the services used
- Monthly calculator
  - https://calculator.s3.amazonaws.com/index.html

EC2 – Elastic Compute Cloud

- EC2 is a compute server
- You have choice to run windows and Linux virtual machines
- Instance type can vary based on the requirement

Introduction to Developing on AWS

- AWS provide services for application development i.e compute, storage, database, caching, security etc
- Developer use these services and develop the applications

Choosing a DataStore

- EBS – elastic block storage
- S3 – Simple storage service – Web accessible storage
- RDS – Relation Data Service
- DynamoDB – NoSQL database
- ElastiCache – In memory database
- Glacier – Archiving Storage

EC2 Instance configuration

- T2 Instance are used in dev environment where there is less usage of CPU
- M3 Instance used for data processing tasks that require additional memory, for running backend servers on SAP
- C3/C4 Instance type used for high performance front-end ,web servers
- G2 Instance type used for Game streaming, Video streaming, 3d application streaming
- HS1 instance type used for parallel systems

EC2 Instance configuration

- **EC2 Instance Metadata and User Data**
  - *Instance metadata* is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories.
  - You can also use instance metadata to access *user data* that you specified when launching your instance.
  - For example, you can specify parameters for configuring your instance, or attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

EC2 Instance configuration

- **EC2 Instance Metadata and User Data -***Retrieving Instance Metadata*
  - *Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the public hostname of your instance from instance metadata to manage a connection.*
  - *To view all categories of instance metadata from within a running instance, use the following URI:*
  - *http://169.254.169.254/latest/meta-data/*
  - *Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.*
  - *You can use a tool such as cURL, or if your instance supports it, the GET command; for example:*
  - *[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/*
  - *[ec2-user ~]$ GET http://169.254.169.254/latest/meta-data/*
  - curl http://169.254.169.254/latest/meta-data/public-hostname

EC2 Instance configuration

- **Running Commands on Your Linux Instance at Launch- EC2 Bootstrapping**

- When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.
- You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives. You can also pass this data into the launch wizard as plain text, as a file (this is useful for launching instances using the command line tools), or as base64-encoded text (for API calls).

EC2 Instance configuration

- **Running Commands on Your Linux Instance at Launch- EC2 Bootstraping**

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
```

- **To access - http://my.public.dns.amazonaws.com**
- **To view and update - /action/instance-setting/View-Change user data**

Demo

- Create EC2 instance

S3- Simple Storage Service

- Companies today need the ability to simply and securely collect, store, and analyze their data at a massive scale.
- Amazon S3 is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices.
- It is designed to deliver 99.999999999% durability, and stores data for millions of applications used by market leaders in every industry.
- S3 provides comprehensive security and compliance capabilities that meet even the most stringent regulatory requirements.
- It gives customers flexibility in the way they manage data for cost optimization, access control, and compliance.

S3- Simple Storage Service

- Amazon S3 is easy to use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web.
- You can store static files docs, ppt ,xls , pdf ,audio and video files any other files
- Amazon S3 provides cost-effective object storage for a wide variety of use cases including backup and recovery, big data analytics, content distribution

S3- Simple Storage Service

- S3 uses buckets to store files. Buckets act as a container for storage
- By default all buckets are private, you need to make it public to access
- Each developer account has limit of 100 buckets
- By default three replicas of file are made in same region
- Replication across region is allowed but it comes with additional charges. The charges are based on the region

S3- Simple Storage Service

https://console.aws.amazon.com/s3/home?region=ap-southeast-2

Demo

- Creating S3 bucket
- Upload, download files
- Delete bucket
- Static website configuration

AWS CLI

- The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell.
- To install the AWS CLI using the MSI installer.
  - Download the AWS CLI MSI installer for Windows (64-bit) - https://docs.aws.amazon.com/cli/latest/userguide/install-windows.html#install-msi-on-windows
  - Download the AWS CLI MSI installer for Windows (32-bit) - https://s3.amazonaws.com/aws-cli/AWSCLI32PY3.msi
  - Run the downloaded MSI installer or the setup file.
  - By default, the CLI installs to C:\Program Files\Amazon\AWSCLI (64-bit version) or C:\Program Files (x86)\Amazon\AWSCLI (32-bit version).
  - To confirm the installation, use the aws --version command at a command prompt
    - C:\> aws --version

AWS CLI

- Configuring the AWS CLI

  - For general use, the aws configure command is the fastest way to set up your AWS CLI installation.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE1
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bP2xRfiCYEXAMPLEKEY
Default region name [None]: ap-south-1
Default output format [None]: json
```

Use IAM service to create access key and access key id.

AWS CLI

- Examples
  - The following mb command creates a bucket.

    aws s3 mb s3://mybucket
  - List S3 objects and common prefixes under a prefix or all S3 buckets.

    aws s3 ls s3://mybucket
  - Copies a local file or S3 object to another location locally or in S3.

    aws s3 cp test.txt s3://mybucket/test2.txt

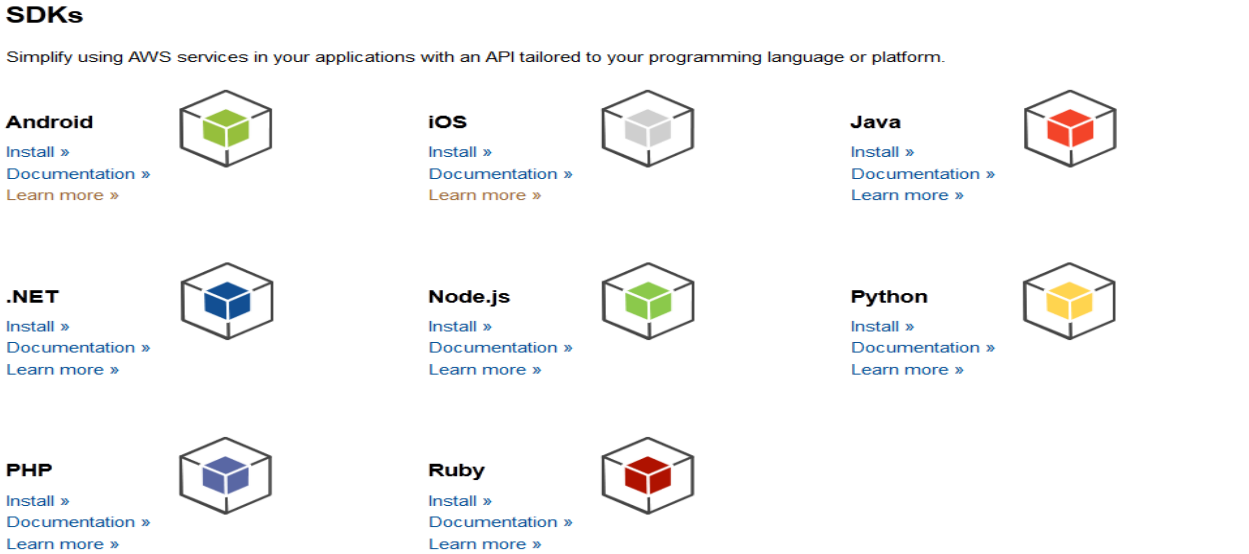
    aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt
  - Deletes an S3 object.

    aws s3 rm s3://mybucket/test2.txt
  - Deletes an empty S3 bucket.

    aws s3 rb s3://mybucket

AWS SDK

- **The AWS SDK provides API for AWS services which you can use to build applications.**



SDKs

Simplify using AWS services in your applications with an API tailored to your programming language or platform.

Android
Install »
Documentation »
Learn more »

iOS
Install »
Documentation »
Learn more »

Java
Install »
Documentation »
Learn more »

.NET
Install »
Documentation »
Learn more »

Node.js
Install »
Documentation »
Learn more »

Python
Install »
Documentation »
Learn more »

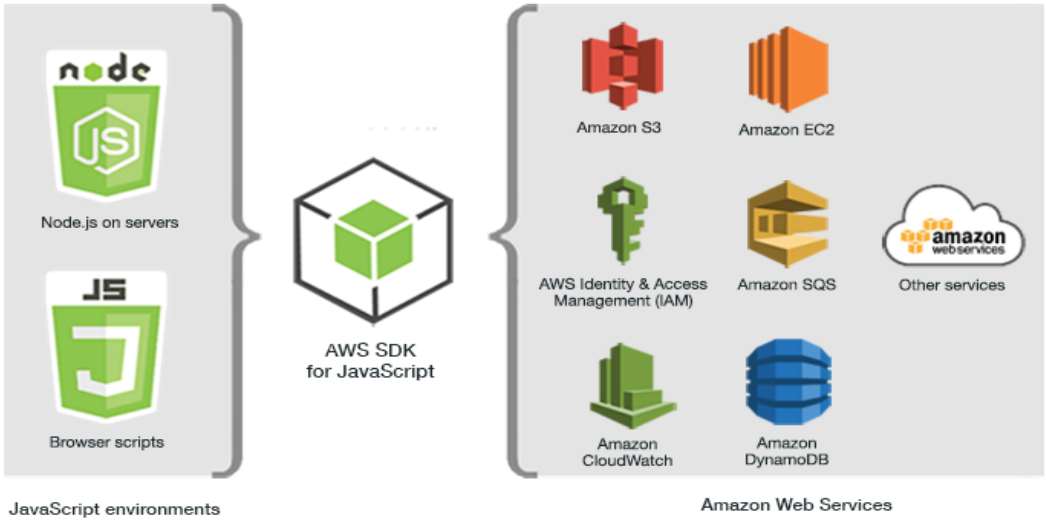
PHP
Install »
Documentation »
Learn more »

Ruby
Install »
Documentation »
Learn more »

↑ Top

AWS SDK

- **AWS SDK for JavaScript**
  - **The AWS SDK for JavaScript provides a JavaScript API for AWS services you can use to build applications for Node.js or the browser.**
  - **The JavaScript API lets developers build libraries or applications that make use of AWS services.**

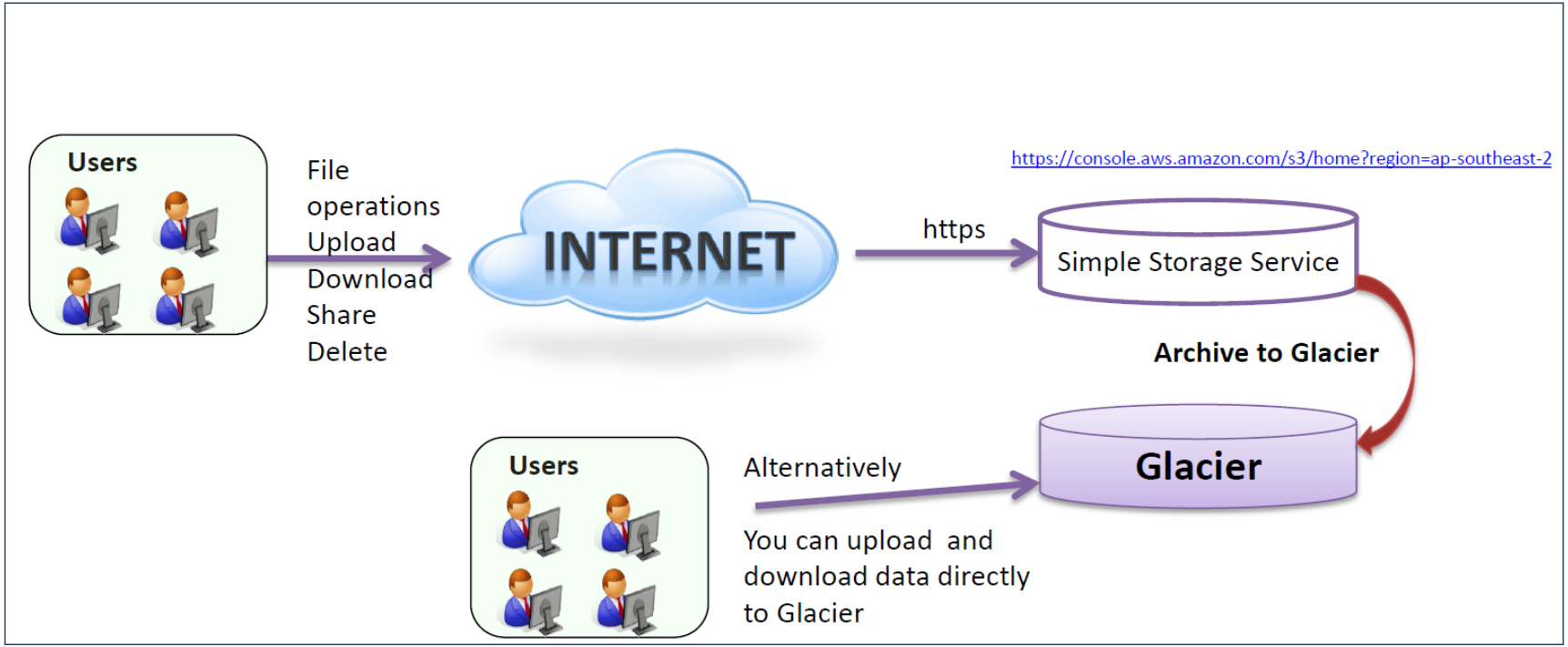


https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/welcome.html

Glacier

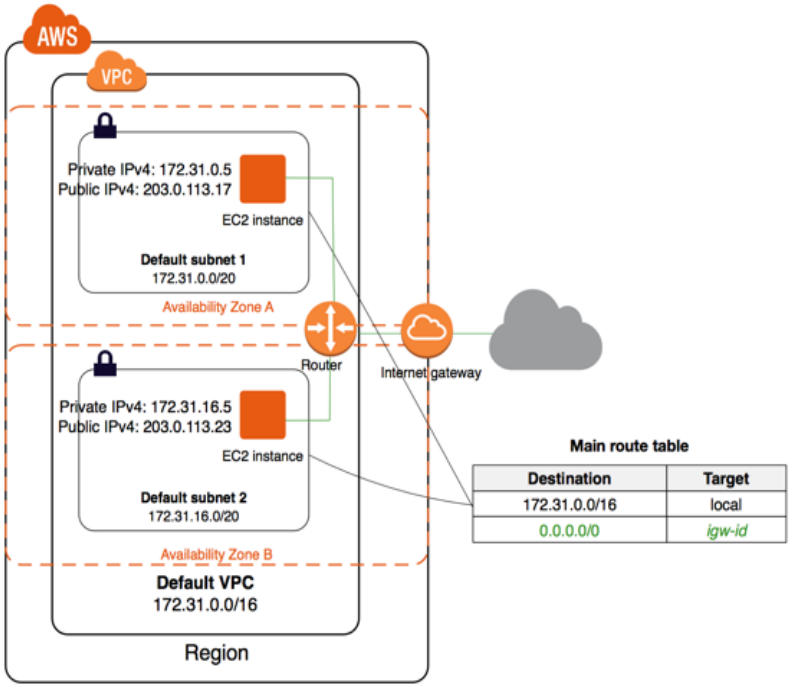
- Amazon Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup
- Through Amazon S3 lifecycle policies, you can optimize your storage costs by moving infrequently accessed objects from Amazon S3 to Amazon Glacier (or vice-versa)
- For long term retention of data, Customers can reliably store large or small amounts of data for as little as $0.007 per gigabyte per month, a significant savings compared to storing on S3
- Amazon Glacier supports data transfer over SSL and automatically encrypts your data at rest
- A single archive can be as large as 40 terabytes

Glacier



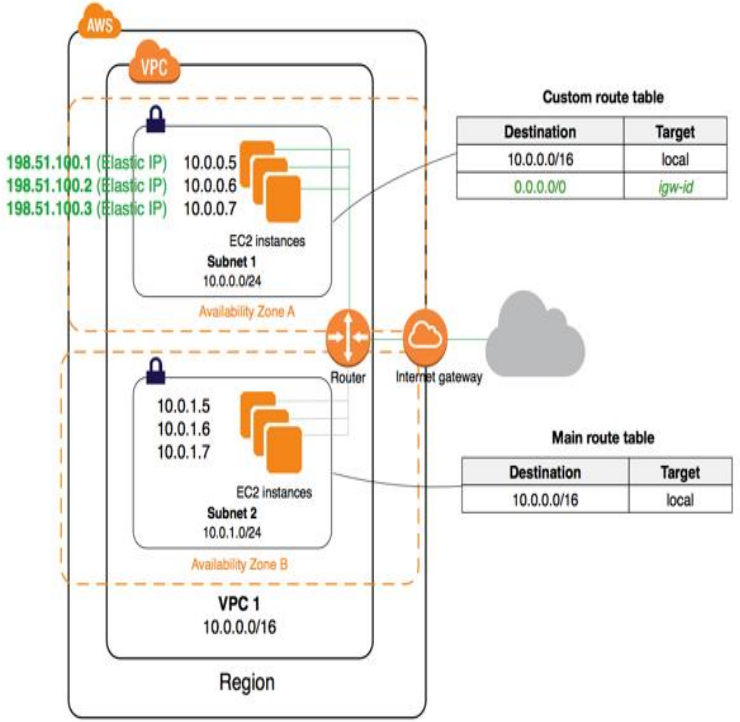
VPC

- A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.
- You can specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.



VPC

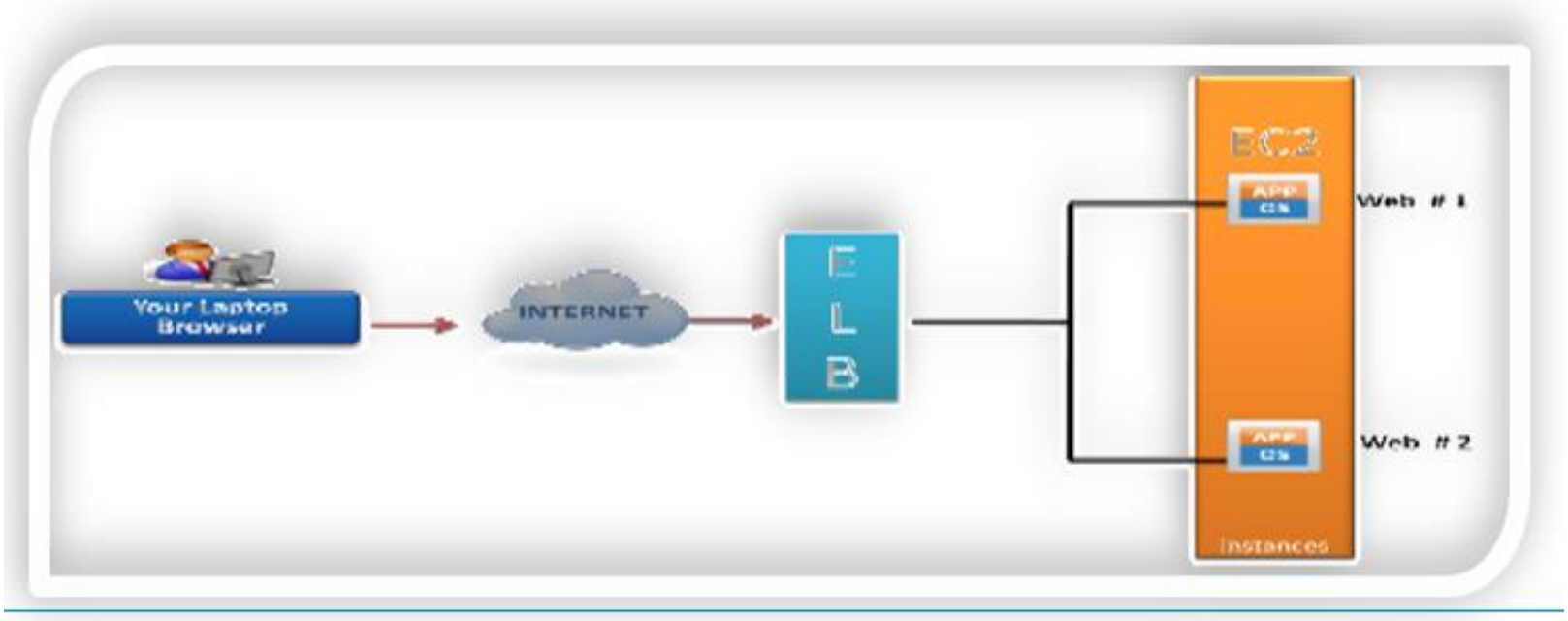
- A *subnet* is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet.
- To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).



ELB- Elastic Load Balancer

- Elastic Load balancer automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud
- Achieve higher levels of fault tolerance for your applications by using Elastic Load Balancing to automatically route traffic across multiple instances and multiple Availability Zones
- Additionally, Elastic Load Balancing offers integration with Auto Scaling to ensure that you have back-end capacity to meet varying levels of traffic without requiring minimal intervention

ELB- Elastic Load Balancer

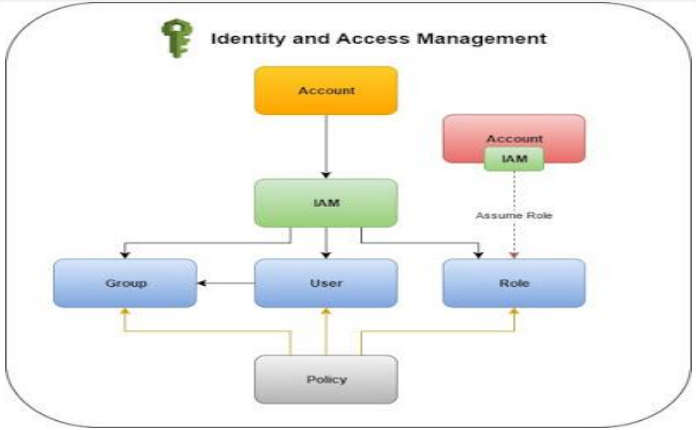


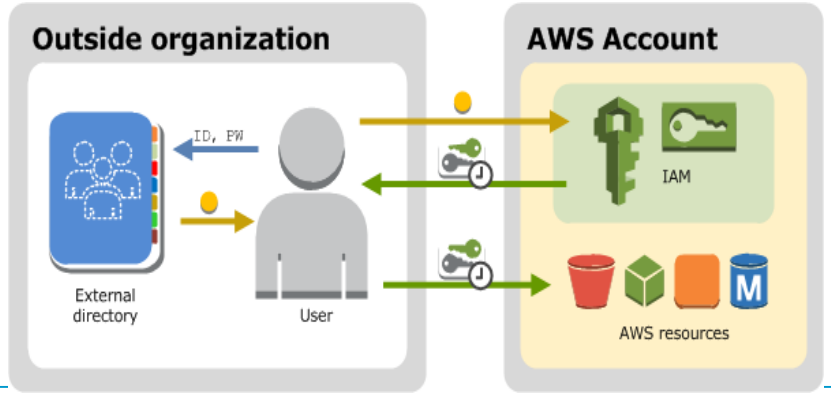
Demo

- Configuring Load Balancer

IAM

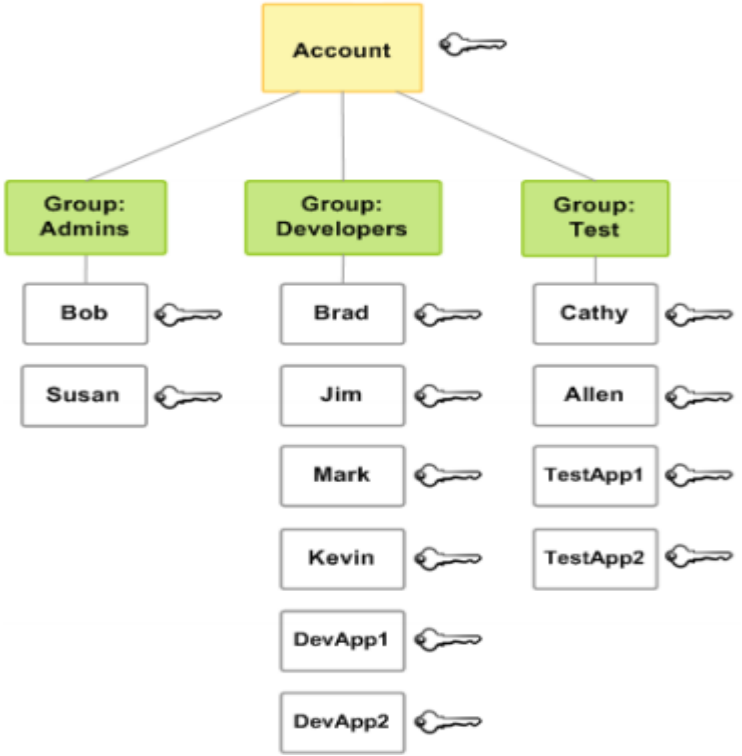
- AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources.
- You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.
- When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account.





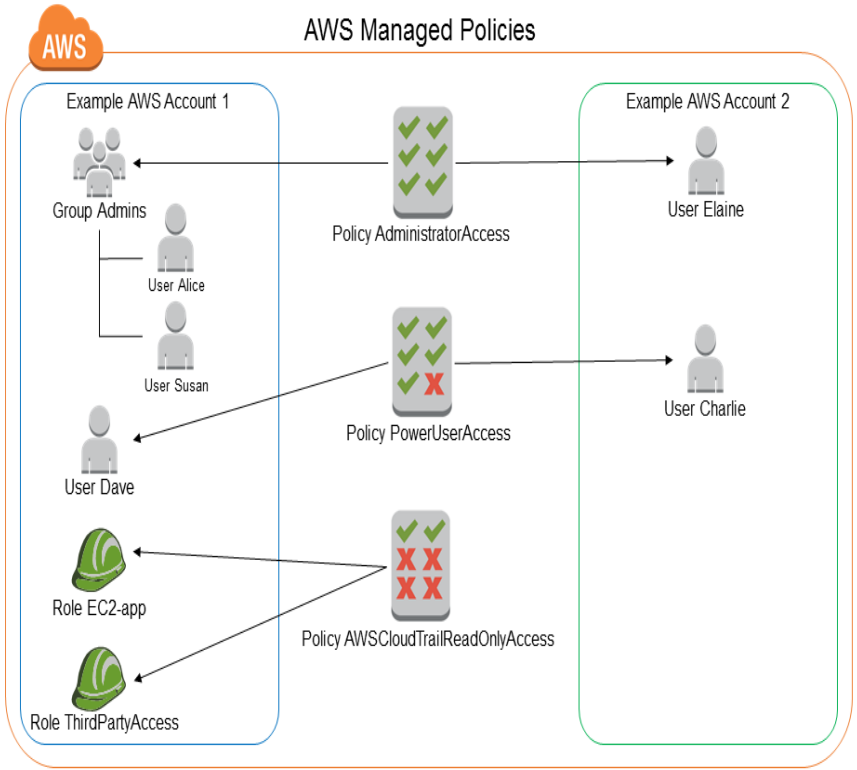
IAM

- This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account.
- It is strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones.



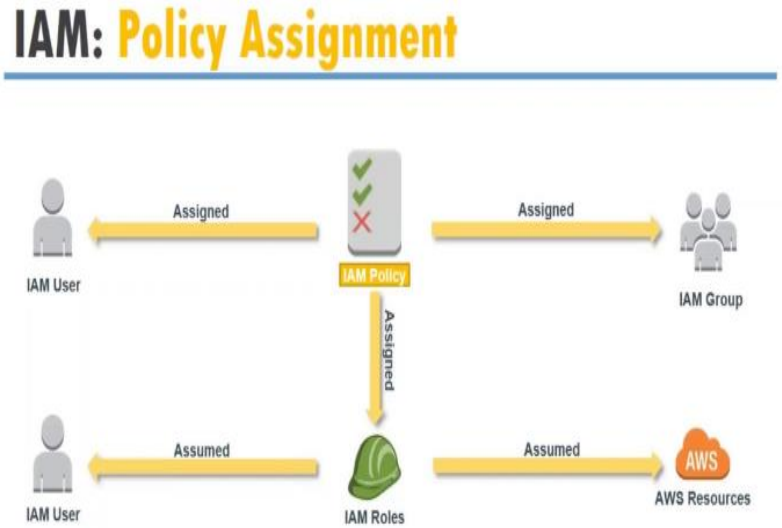
IAM

- You manage access in AWS by creating policies and attaching them to IAM identities or AWS resources.
- A policy is an object in AWS that, when associated with an entity or resource, defines their permissions. AWS evaluates these policies when a principal, such as a user, makes a request.
- Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.



IAM

- IAM policies define permissions for an action regardless of the method that you use to perform the operation.
- When you create an IAM user, you can set up the user to allow console or programmatic access.
- The IAM user can sign in to the console using a user name and password. Or they can use access keys to work with the CLI or API.



AWS Lambda

- AWS Lambda is a compute service that lets you run code without provisioning or managing servers.
- AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second.
- You pay only for the compute time you consume - there is no charge when your code is not running.
- With AWS Lambda, you can run code for virtually any type of application or backend service - all with zero administration.
- AWS Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging.

AWS Lambda

- AWS Lambda supports (currently Node.js, Java, C#, Go and Python).
- You can use AWS Lambda to run your code in response to events, such as changes to data in an Amazon S3 bucket or an Amazon DynamoDB table; to run your code in response to HTTP requests using Amazon API Gateway; or invoke your code using API calls made using AWS SDKs.
- With these capabilities, you can use Lambda to easily build data processing triggers for AWS services like Amazon S3 and AmazonDynamoDB or create your own back end that operates at AWS scale, performance, and security.

AWS Lambda

- **When Should I Use AWS Lambda?**
  - When using AWS Lambda, you are responsible only for your code.
  - AWS Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources. This is in exchange for flexibility, which means you cannot log in to compute instances, or customize the operating system or language runtime. These constraints enable AWS Lambda to perform operational and administrative activities on your behalf, including provisioning capacity, monitoring fleet health, applying security patches, deploying your code, and monitoring and logging your Lambda functions.

API Gateway

- Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.
- With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any web application.
- Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management. Amazon API Gateway has no minimum fees or startup costs. You pay only for the API calls you receive and the amount of data transferred out.

API Gateway

- Benefits
  - LOW COST AND EFFICIENT -With Amazon API Gateway, you pay only for calls made to your APIs and data transfer out. There are no minimum fees or upfront commitments
  - PERFORMANCE AT ANY SCALE - With Amazon CloudFront integration, API Gateway allows you to take advantage of the worldwide network of edge locations to provide your end users with the lowest possible latency for API requests and responses. Amazon API Gateway also helps you manage traffic through throttling, so that back-end operations can withstand traffic spikes. Additionally, Amazon API Gateway helps you improve the performance of your APIs by caching the output of API calls to avoid calling your backend systems unnecessarily.
  - EASILY MONITOR API ACTIVITY -After your API is deployed, Amazon API Gateway provides you with a dashboard to visually monitor calls to your services using Amazon CloudWatch, so you see performance metrics and information on API calls, data latency, and error rates.
  - RUN YOUR APIS WITHOUT SERVERS-Amazon API Gateway tightly integrates with AWS Lambda to allow you to create completely serverless APIs.AWS Lambda runs your code on a high-availability compute infrastructure and performs all the heavy lifting and administration of your compute resources.

API Gateway

- Benefits

  - CREATE RESTFUL API ENDPOINTS FOR EXISTING SERVICES -With Amazon API Gateway, you can create modern resource based APIs, and then use the dynamic and flexible data transformation capabilities to generate the requests in the language your target services expect. API Gateway also helps you protect your existing services by setting throttling rules to avoid overwhelming your back-end infrastructure during unpredictable traffic spikes.

AWS Elastic Beanstalk

- AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
- You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.
- There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications.

AWS Elastic Beanstalk

- Benefits
  - FAST AND SIMPLE TO BEGIN-Elastic Beanstalk is the fastest and simplest way to deploy your application on AWS. You simply use the AWS Management Console, a Git repository, or an integrated development environment (IDE) such as Eclipse or Visual Studio to upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. Within minutes, your application will be ready to use without any infrastructure or resource configuration work on your part.
  - DEVELOPER PRODUCTIVITY-Elastic Beanstalk provisions and operates the infrastructure and manages the application stack (platform) for you, so you don't have to spend the time or develop the expertise. It will also keep the underlying platform running your application up-to-date with the latest patches and updates. Instead, you can focus on writing code rather than spending time managing and configuring servers, databases, load balancers, firewalls, and networks

AWS Elastic Beanstalk

- Benefits
  - COMPLETE RESOURCE CONTROL-You have the freedom to select the AWS resources, such as Amazon EC2 instance type, that are optimal for your application. Additionally, Elastic Beanstalk lets you "open the hood" and retain full control over the AWS resources powering your application. If you decide you want to take over some (or all) of the elements of your infrastructure, you can do so seamlessly by using Elastic Beanstalk's management capabilities.

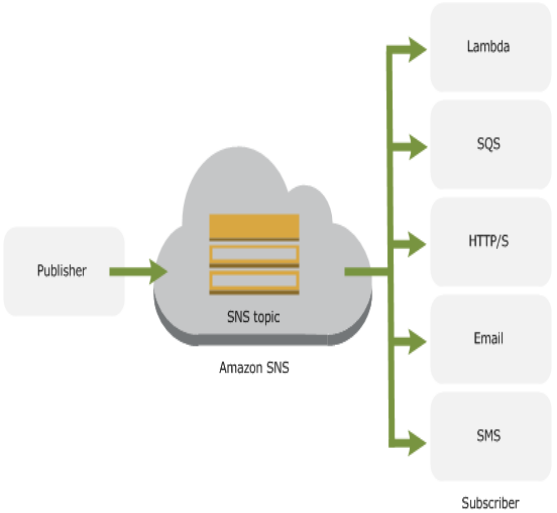
  - IMPOSSIBLE TO OUTGROW- Elastic Beanstalk automatically scales your application up and down based on your application's specific need using easily adjustable Auto Scaling settings. For example, you can use CPU utilization metrics to trigger Auto Scaling actions. With Elastic Beanstalk, your application can handle peaks in workload or traffic while minimizing your costs.

Amazon Simple Notification Service (SNS)

- Amazon Simple Notification Service (SNS) is a flexible, fully managed pub/sub messaging and mobile notifications service for coordinating the delivery of messages to subscribing endpoints and clients.
- With SNS you can fan-out messages to a large number of subscribers, including distributed systems and services, and mobile devices. It is easy to set up, operate, and reliably send notifications to all your endpoints – at any scale.
- You can get started using SNS in a matter of minutes using the AWS Management Console, AWS Command Line Interface, or using the AWS SDK with just three simple APIs. SNS eliminates the complexity and overhead associated with managing and operating dedicated messaging software and infrastructure.

Amazon Simple Notification Service (SNS)

- Amazon SNS Pub/Sub Messaging
  - Amazon SNS is a fully managed pub/sub messaging service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications.
  - With SNS, you can use topics to decouple message publishers from subscribers, fan-out messages to multiple recipients at once, and eliminate polling in your applications.
  - SNS supports a variety of subscription types, allowing you to push messages directly to Amazon Simple Queue Service (SQS) queues, AWS Lambda functions, and HTTP endpoints. AWS services, such as Amazon EC2, Amazon S3 and Amazon CloudWatch, can publish messages to your SNS topics to trigger event-driven computing and workflows.
  - SNS works with SQS to provide a powerful messaging solution for building cloud applications that are fault tolerant and easy to scale.



Amazon Simple Queue Service (SQS)

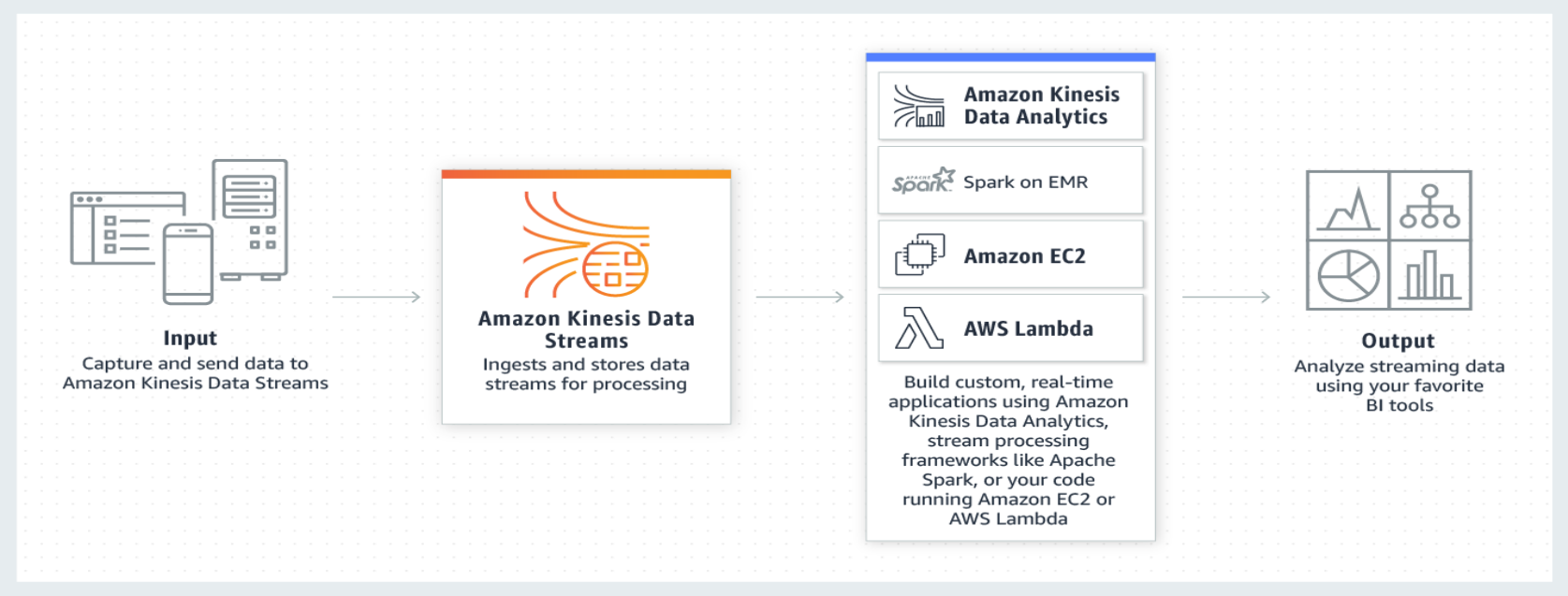
- Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.
- SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work.
- Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of your choice, and three simple commands.
- SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

Amazon Kinesis

- Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information.
- Amazon Kinesis offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application.
- With Amazon Kinesis, you can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications.
- Amazon Kinesis enables you to process and analyze data as it arrives and respond instantly instead of having to wait until all your data is collected before the processing can begin.

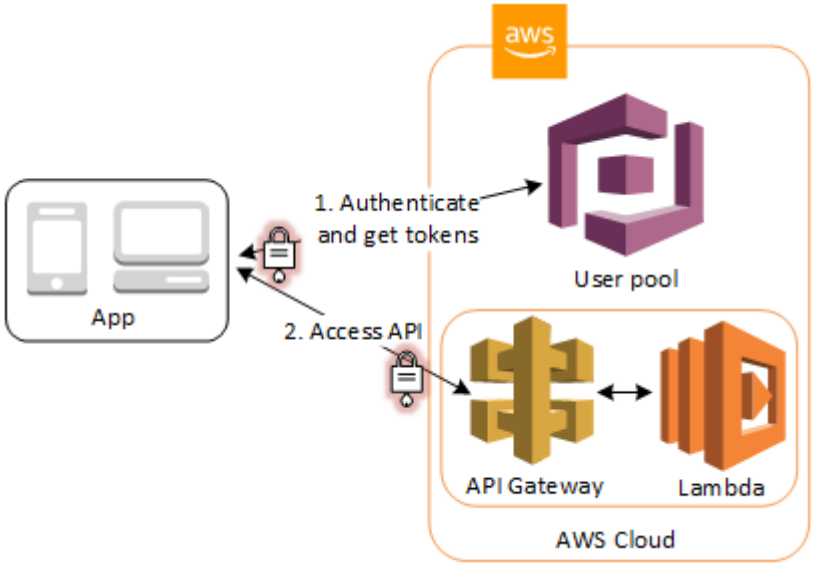
Amazon Kinesis

- Data Stream



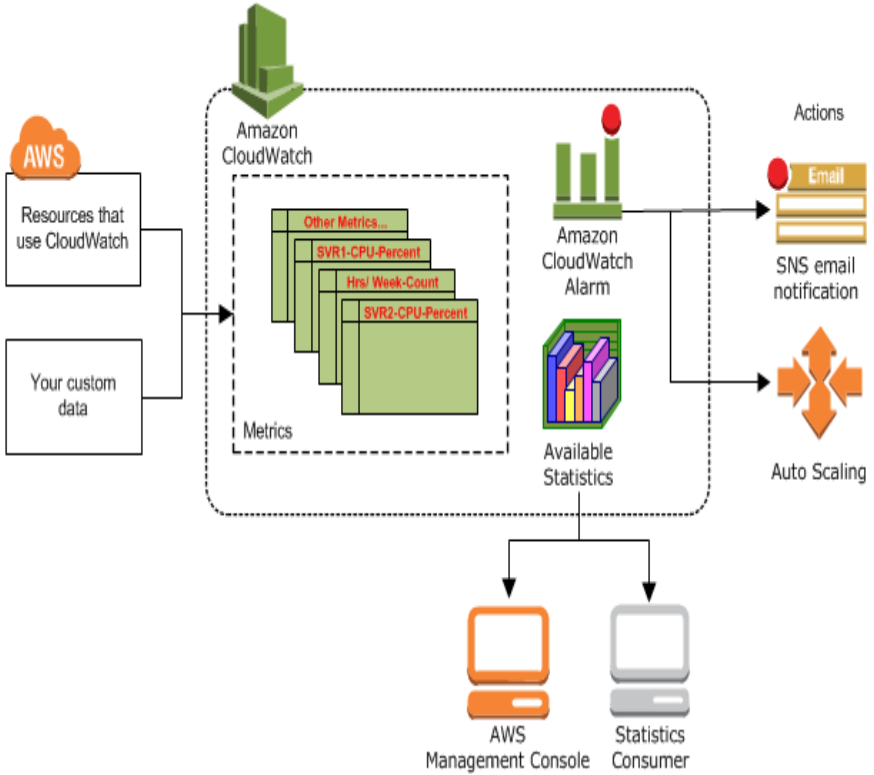
Amazon Cognito-Simple and Secure User Sign-Up, Sign-In, and Access Control

- Spend your time creating great apps. Let Amazon Cognito handle authentication.
- Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily.
- Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.



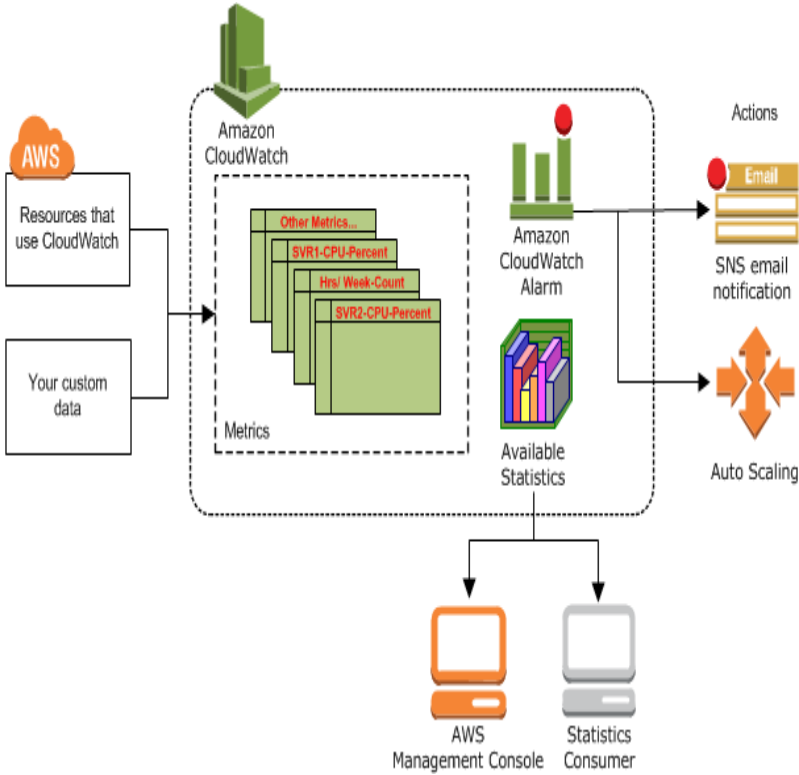
Amazon CloudWatch- Complete Visibility of Your Cloud Resources and Applications

- Amazon CloudWatch is a monitoring and management service built for developers, system operators, site reliability engineers (SRE), and IT managers.
- CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.



Amazon CloudWatch- Complete Visibility of Your Cloud Resources and Applications

- CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers.
- You can use CloudWatch to set high resolution alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to optimize your applications, and ensure they are running smoothly.

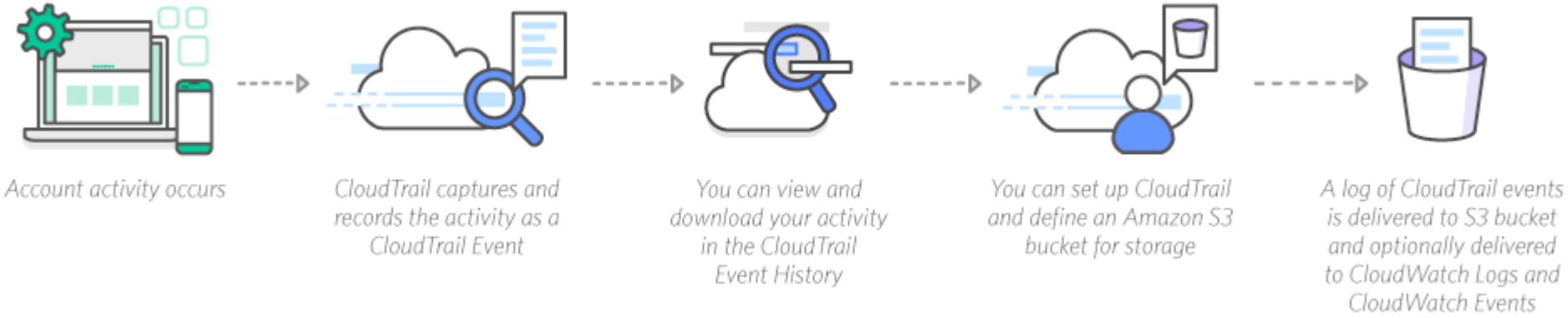


AWS CloudTrail - Track user activity and API usage

- AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account.
- With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure.
- CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services.
- This event history simplifies security analysis, resource change tracking, and troubleshooting.

AWS CloudTrail - Track user activity and API usage

- How it works



https://docs.aws.amazon.com/lambda/latest/dg/with-cloudtrail-example.html

What is RDS?

- RDS (Relational Database Service) : Amazon Relational Database Service (Amazon RDS) is a web service that makes it easy to set up, operate, and scale a relational database in the cloud .Started in 2009

RDS

- Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud
- It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business
- Amazon RDS provides you six familiar database engines to choose from

Amazon RDS Database Engines

RDS-Benefits

- No need to launch EC2 instance
- No need to install and manage database servers
- No need to setup replication (primary database on instance A and secondary database on instance B)
- Even minor upgrade of your database version are taken care by AWS

AWS DynamoDB

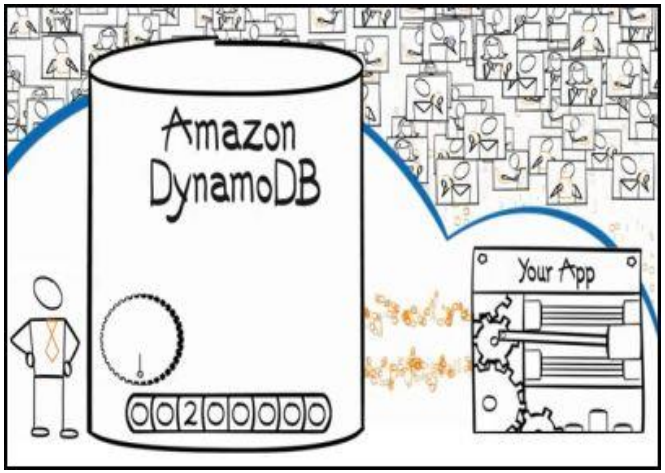
- Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.
- You can use Amazon DynamoDB to create a database table that can store and retrieve any amount of data, and serve any level of request traffic.
- Amazon DynamoDB automatically spreads the data and traffic for the table over a sufficient number of servers to handle the request capacity specified by the customer and the amount of data stored, while maintaining consistent and fast performance.
- Amazon DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

Introduction to DynamoDB

- Fully managed NoSQL database service by Amazon
- Database type: Key-value stores
- Designed to address the core problems of database management, performance, scalability, and reliability

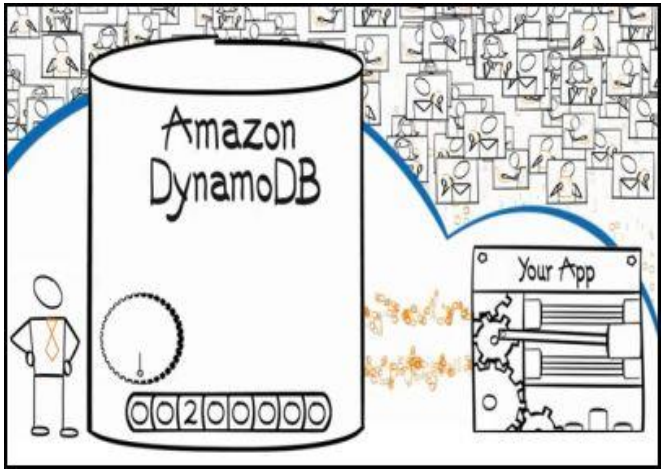
Features

- Scalable
  - Provisioned Throughput
  - Fully Distributed, Shared Nothing Architecture
- Fast Performance
  - Average service-side latencies < 10 ms
  - The service runs on Solid State Disks - consistent, fast latencies at any scale
- Easy Administration and Cost Effective
  - a fully managed service by Amazon
- Fault-tolerant
  - Synchronous replication across multiple zones in a region

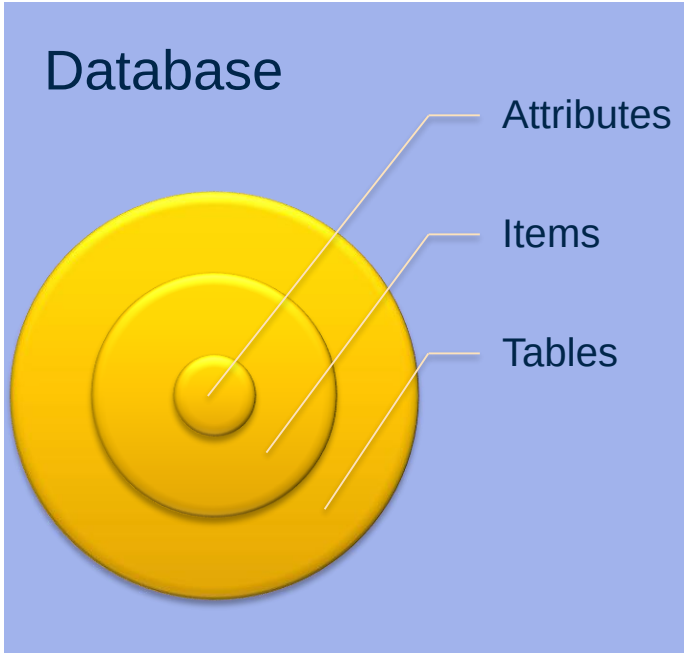


Features

- Flexible
  - Does not have a fixed schema
- Efficient Indexing
  - Every item identified by a primary key
- Strong consistency
  - Implemented with Atomic Counters
  - Disk-only writes
- Secure with Monitoring
  - AWS Identity and Access Management
  - CloudWatch for monitoring request throughput, latency and resource consumption
- Amazon Elastic MapReduce Integration
- Amazon Redshift Integration



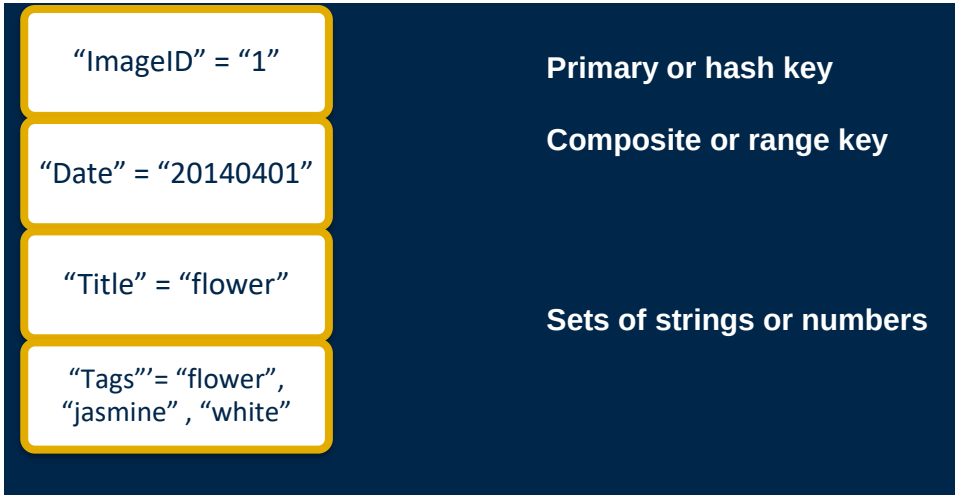
Data model concepts



- Except for the primary key, DynamoDB is schema-less
- Each item can have any number of attributes
- An attribute is a name-value pair
  - can be single valued or multi-valued set

Data model concepts

- Primary keys
  - Hash type primary key
  - Hash and range type primary key
- Secondary Indexes
  - Local secondary index
  - Global secondary index
- DynamoDB data types
  - Scalar data types
  - Multivalued data types



Supported Operations

- Table operations
  - create, update and delete tables
- Item operations
  - add, update and delete items from a table
  - retrieve a single item (GetItem) or multiple items (BatchGetItem)
- Query and Scan
  - query a table using the hash attribute and an optional range filter.
  - If the table has a secondary index, you can also Query the index using its key
  - Scan operation reads every item in the table or secondary index

Supported Operations

- Data Read and Consistency considerations
  - Multiple copies of each item to ensure durability
  - Eventually Consistent Reads
  - Strongly Consistent Reads

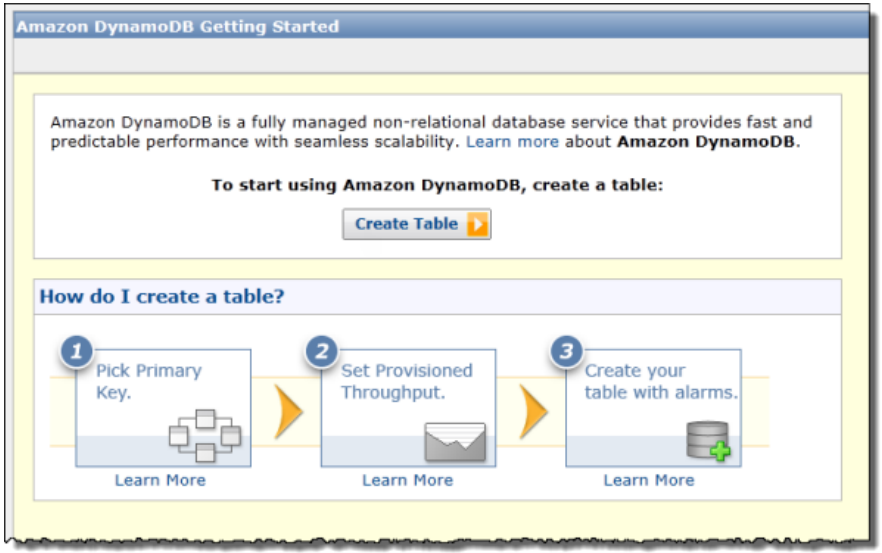
- Conditional updates and concurrency control
  - updates made by one client don't overwrite updates made by another client
  - “conditional write” and “atomic counter”

Considerations

- Item size is limited to 64KB
- Attribute values can not be null or empty
- Hash primary key attribute value is limited to 1024 bytes
- Range primary key attribute value is limited to 2048 bytes
- Up to 5 local as well as global secondary indexes per table

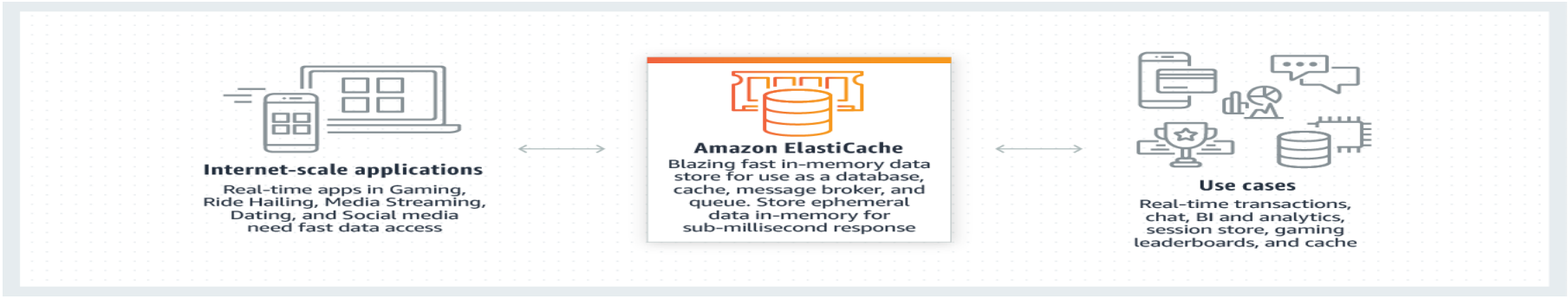
Start using DynamoDB!

- The AWS Management Console for Amazon DynamoDB is available at https://console.aws.amazon.com/dynamodb/home



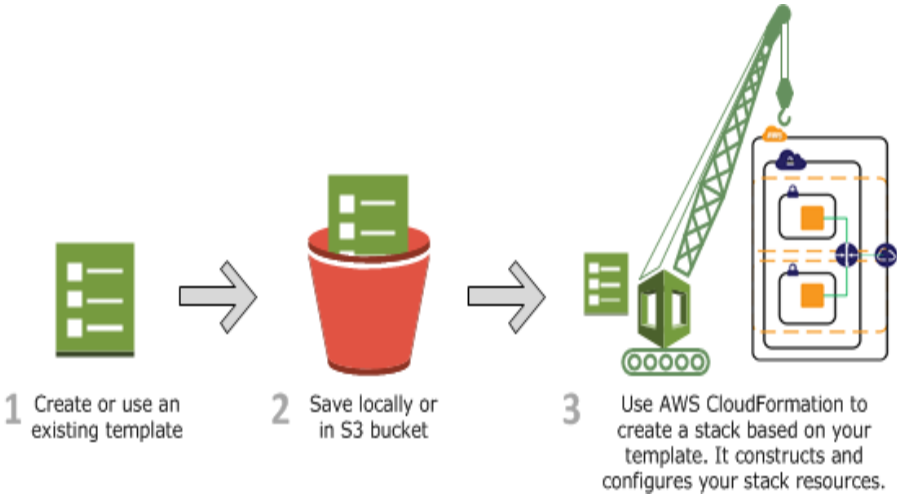
Amazon ElastiCache-Managed, Redis or Memcached-compatible in-memory data store.

- Amazon ElastiCache offers fully managed Redis and Memcached.
- Seamlessly deploy, run, and scale popular open source compatible in-memory data stores.
- Build data-intensive apps or improve the performance of your existing apps by retrieving data from high throughput and low latency in-memory data stores.
- Amazon ElastiCache is a popular choice for Gaming, Ad-Tech, Financial Services, Healthcare, and IoT apps.



AWS CloudFormation

- AWS CloudFormation is a service that helps you model and set up your Amazon Web Services resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS.
- You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and AWS CloudFormation takes care of provisioning and configuring those resources for you.
- You don't need to individually create and configure AWS resources and figure out what's dependent on what; AWS CloudFormation handles all of that.



AWS CloudFormation

- The following scenarios demonstrate how AWS CloudFormation can help.
  - **Simplify Infrastructure Management –**You can create or modify an existing AWS CloudFormation template. A template describes all of your resources and their properties. When you use that template to create an AWS CloudFormation stack, AWS CloudFormation provisions the Auto Scaling group, load balancer, and database for you. After the stack has been successfully created, your AWS resources are up and running. You can delete the stack just as easily, which deletes all the resources in the stack. By using AWS CloudFormation, you easily manage a collection of resources as a single unit.
  - **Quickly Replicate Your Infrastructure -**If your application requires additional availability, you might replicate it in multiple regions so that if one region becomes unavailable, your users can still use your application in other regions.

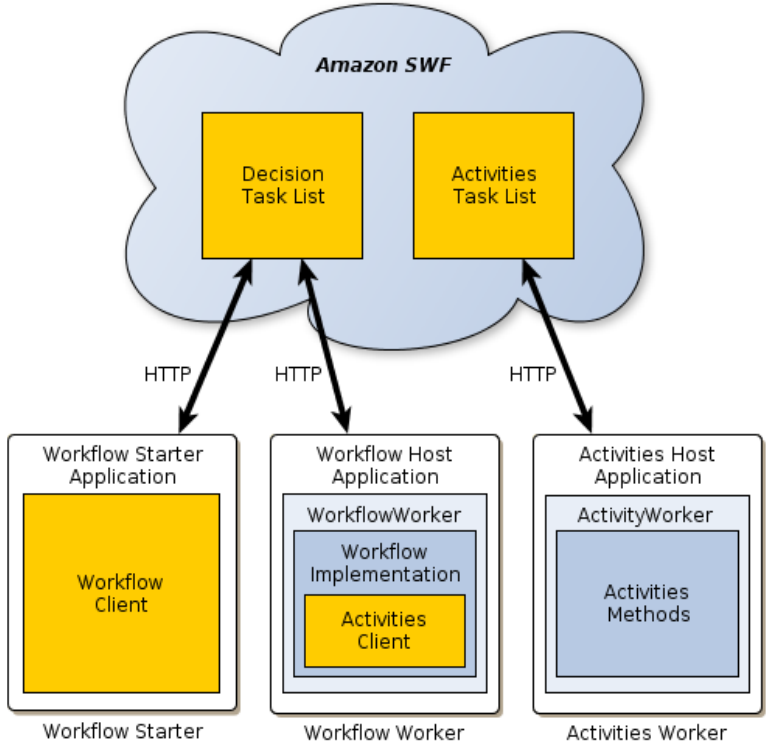
AWS CloudFormation

- The following scenarios demonstrate how AWS CloudFormation can help.

  - **Easily Control and Track Changes to Your Infrastructure-** When you provision your infrastructure with AWS CloudFormation, the AWS CloudFormation template describes exactly what resources are provisioned and their settings. Because these templates are text files, you simply track differences in your templates to track changes to your infrastructure, similar to the way developers control revisions to source code. For example, you can use a version control system with your templates so that you know exactly what changes were made, who made them, and when. If at any point you need to reverse changes to your infrastructure, you can use a previous version of your template.

Amazon Simple Workflow Service (SWF)

- Amazon SWF helps developers build, run, and scale background jobs that have parallel or sequential steps.
- You can think of Amazon SWF as a fully-managed state tracker and task coordinator in the Cloud.
- AWS Flow Framework application consists of three basic components: *workflow starters*, *workflow workers*, and *activity workers*. One or more host applications are responsible for registering the workers (workflow and activity) with Amazon SWF, starting the workers, and handling cleanup.
- The workers handle the mechanics of executing the workflow and may be implemented on several hosts.



Amazon Simple Workflow Service (SWF)

DECIDERS

SWF

Activity Workers

DECIDERS

Responsible for making workflow decisions

Typically a long-running process

```
While True:
    Poll task-list for a decision
    Analyze execution history
    Make a decision on next step(s)
    Return result to SWF
```

Responsible for doing the "work"

Typically a long-running process

Poll ~~queue~~ task-list → Do work

```
While True:
    Poll an SWF ~~queue~~ task-list
    Process the task
    Return result to SWF
```

Activity Workers

SWF Responsibilities

Coordinate system components

Schedule "decision tasks" (a queue?)

Schedule "activity tasks" (a queue?)

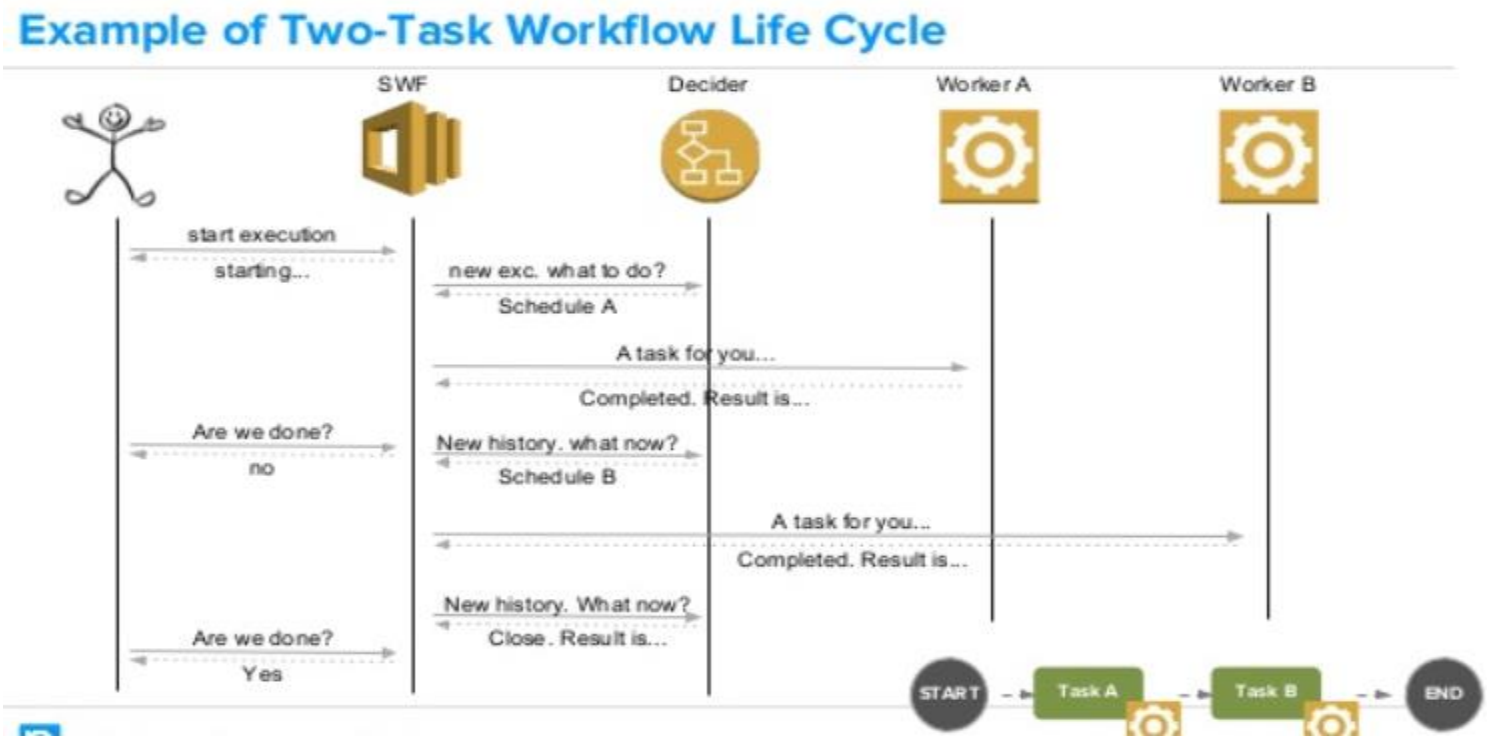
Maintain workflow state

Catch errors and timeouts

Provide an API to track workflow-execution progress

Does No work. Makes No decisions.

Amazon Simple Workflow Service (SWF)



Summary

- We learned various services and understood when to use which service.