In28 Minutes

AWS CERTIFIED CLOUD PRACTITIONER

<< *Exam Review* >>

Getting Started

In28 Minutes

- AWS has *200+ services*. Exam expects knowledge of *40+ services*.
- You might have used these services before.
- But you need to remember all details when attending the exam
  - How do you review everything before the exam?
- **Our Goal** : Enable you to quickly review for **AWS Certified Cloud Practitioner** exam
  - (Remember) This is a crash review course!
- **Our Approach**: Quick Review Videos with Presentations and Comparisons
  - (Recommended) Do not hesitate to replay videos!
  - (Recommended) Have Fun!

EC2

DynamoDB

AWS Lambda

API Gateway

Amazon S3

FASTEST ROADMAPS

in28minutes.com

In28 Minutes

- Google Cloud Certifications
- Azure Certifications
- AWS Certifications
- DevOps
- Java Full Stack
- Java Microservices

Getting Started

What?	Description
Cloud	Elasticity. **On-demand resource provisioning**. Trade **"capital expense (capex)"** for **"variable expense (opex)"** (Pay-as-you-go) **"Go global"** in minutes
Availability	Are the applications available **when the users need them**? 99.99% availability = 4 minutes of downtime in a month
AWS Regions	**20+ regions** around the world (us-east-1,eu-west-2) High Availability and Low Latency
Availability Zones	Each Regions has multiple AZs Discrete data centers, with redundant power, networking, and connectivity **Increase availability** of applications in the same region **us-east-1 has 6 AZs** - us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f **eu-west-2 has 3 AZs** - eu-west-2a, eu-west-2b, eu-west-2c

Compute Services - Overview

In28 Minutes

Service	Description
Amazon EC2 + ELB	Traditional Approach (Virtual Servers + Load Balancing)
AWS Elastic Beanstalk	Simplify management of web applications and batch applications. Automatically creates EC2 + ELB (load balancing and auto scaling)
AWS Elastic Container Service (ECS)	Simplify running of microservices with Docker containers. Run containers in EC2 based ECS Clusters
AWS Fargate	Serverless version of ECS
AWS Lambda	Serverless - Do NOT worry about servers
Amazon Lightsail	Pre-configured development stacks in AWS - LAMP, MEAN. Run websites on WordPress.Low, predictable monthly price.
AWS Batch	Run batch computing workloads on AWS

EC2 (Elastic Compute Cloud)

In28 Minutes

- EC2 - Service to provision Virtual Servers or EC2 instances.
- You can pay by second for Linux instances
- Important Concepts to Remember:
  - **Amazon Machine Image (AMI)**: What operating system and what software do you want on the instance?
  - **Instance Families** - Optimized combination of **compute(CPU, GPU), memory, disk (storage) and networking** for specific workloads
    - General Purpose (m), Compute Optimized (c), Memory Optimized (r - RAM) etc
  - **Security Groups** - **Virtual firewall** to control **incoming and outgoing** traffic to/from AWS resources (EC2 instances, databases etc)
    - Define rules to allow traffic to EC2 instances
    - **Default deny** - If there are no rules configured, no outbound/inbound traffic is allowed
    - **Separate rules** for inbound and outbound traffic

EC2 Pricing Models Overview

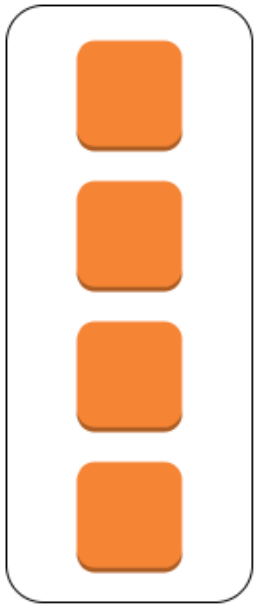
In28 Minutes

Pricing Model	Description
On Demand	Request when you want it. Flexible and Most Expensive. Immediate workloads (web applications/batch programs).
Spot	Cheapest (upto 90% off). Quote the maximum price. Terminated with **2 minute** notice. Cost sensitive, Fault tolerant, Non immediate workloads.
Reserved	Reserve ahead of time. Upto 75% off. 1 or 3 years reservation. **Scheduled**: Reserve for **specific time period** in a day. (5% to 10% off) No Upfront or Partial Upfront or All Upfront Payments
Savings Plans	Commit spending $X per hour on (EC2 or AWS Fargate or Lambda). Upto 66% off. Lot of flexibility. 1 or 3 years reservation. No Upfront or Partial Upfront or All Upfront Payments

EC2 Tenancy - Shared vs Dedicated

In28 Minutes

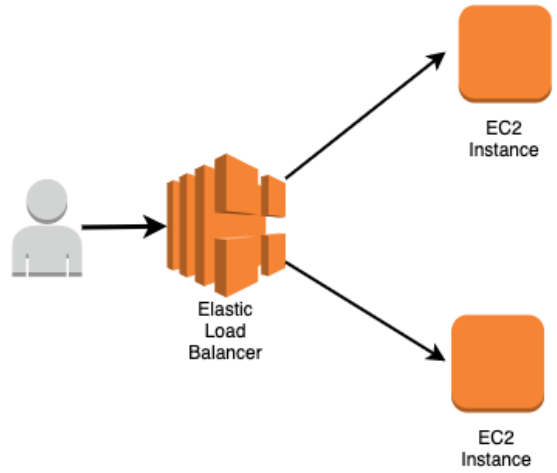
- **Shared Tenancy** (Default)
  - Single host machine can have instances from multiple customers
- **EC2 Dedicated Instances**
  - Virtualized instances on hardware dedicated to one customer
  - You do NOT have visibility into the hardware of underlying host
- **EC2 Dedicated Hosts**
  - Physical servers dedicated to one customer
  - You have visibility into the hardware of underlying host (sockets and physical cores)
  - (Use cases) Regulatory needs or server-bound software licenses like Windows Server, SQL Server



Host

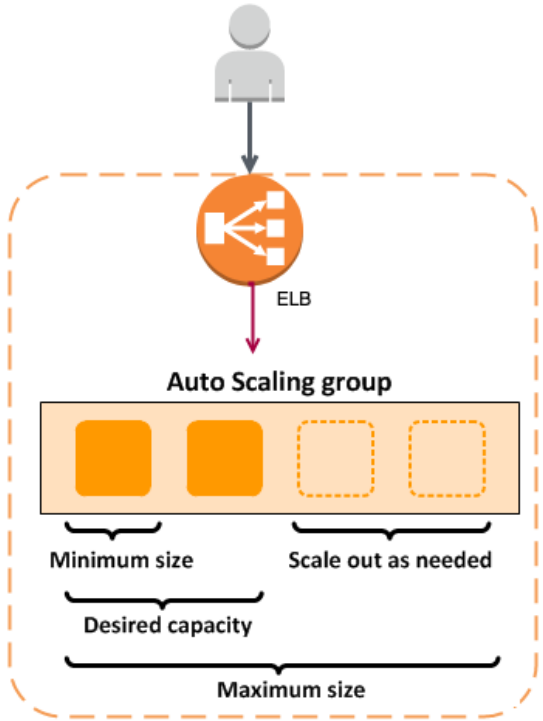
Scalability for EC2 Instances

- Can we handle **growth in users, traffic, or data** without drop in performance?
- **Vertical Scaling** - Deploying to a **bigger instance**:
  - More RAM, CPU, I/O, or networking capabilities
  - Increasing **EC2 instance size**: Example: *t2.micro* to *t2.small*
- **Horizontal Scaling** - Deploying multiple instances:
  - (Typically but not always) Horizontal Scaling is preferred to Vertical Scaling:
    - Horizontal scaling increases availability
  - (BUT) Horizontal Scaling needs additional infrastructure: Load Balancers etc.



Elastic Load Balancer

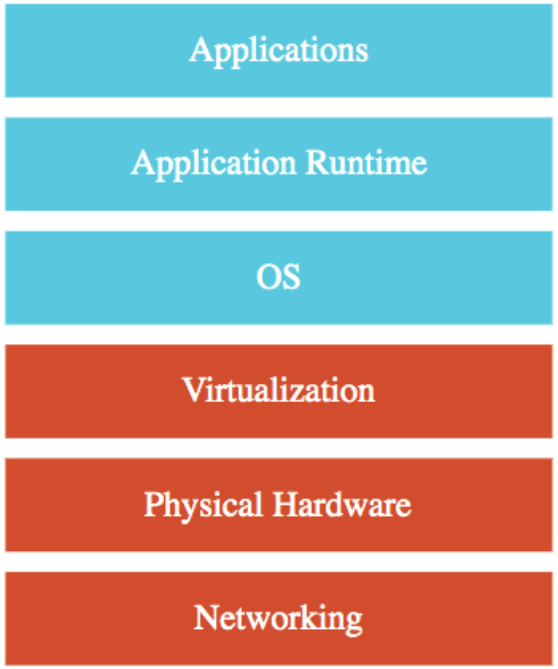
- Distribute traffic to EC2 instances in a single region
  - Can route traffic to IP addresses and Lambda functions also
- **Three Types**:
  - **Classic** Load Balancer: Old generation. Not Recommended by AWS.
  - **Application** Load Balancer: **Most popular** and frequently used ELB in AWS
    - Supports HTTP/HTTPS (Layer 7) and Advanced Routing(Headers, Query Params, Path and Host Based)
  - **Network** Load Balancer: For very high performance usecases
    - Supports TCP/TLS and UDP (Layer 4)
- **Auto Scaling Group**: Scale out and scale in **automatically**
  - Scale-in and scale-out based on auto scaling policies
  - Auto Scaling is done using CloudWatch Alarms



IAAS (Infrastructure as a Service)

In28 Minutes

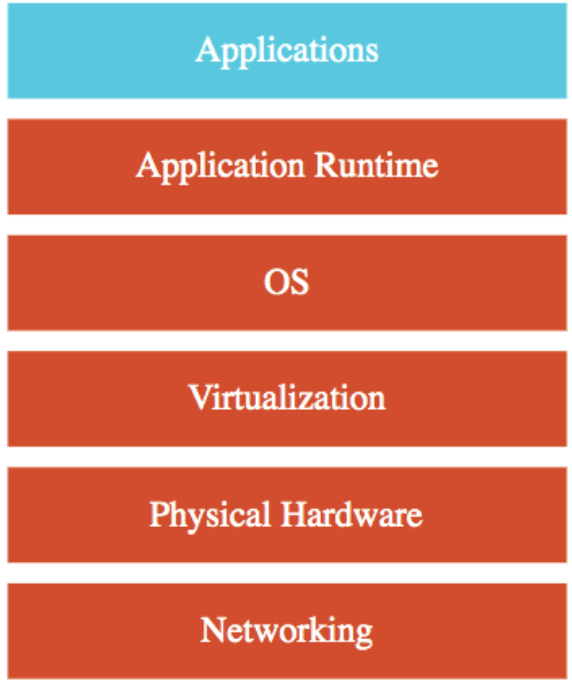
- Use **only infrastructure** from cloud provider
  - **Example**: Using EC2 to deploy your applications
  - **Example**: Using EC2 to create your database
- **Cloud Provider** is responsible for:
  - Physical Infrastructure (Hardware, Networking)
  - Virtualization Layer (Hypervisor, Host OS)
- **Customer** is responsible for:
  - Guest OS upgrades and patches
  - Application Code and Runtime
  - Availability, Scalability etc.



PAAS (Platform as a Service) - Managed Services

In28 Minutes

- Use a platform provided by cloud
- **Cloud provider** is responsible for:
  - Physical Infrastructure (Hardware, Networking)
  - Virtualization Layer (Hypervisor, Host OS)
  - OS (incl. upgrades and patches)
  - Auto scaling, Availability & Load balancing etc..
- **Customer** is responsible for:
  - Application code and/or Configuration
- Examples:
  - **Elastic Load Balancing** - Distribute traffic to multiple targets
  - **AWS Elastic Beanstalk** - Run and Manage Web Apps
  - **Amazon RDS** - Relational Databases - MySQL, Oracle etc
  - And a lot more...



Quick Review of Managed Compute Services

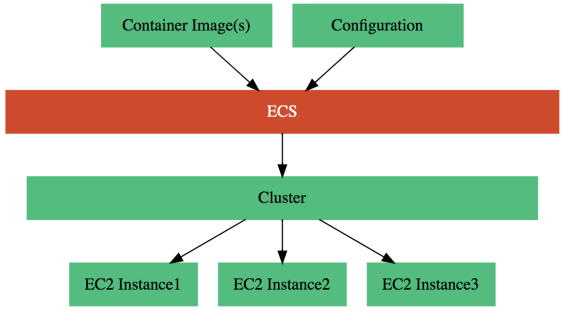
In 28 Minutes

- **AWS Elastic BeanStalk**: Automatic Deployment to EC2 Instances and Load Balance using ELB
  - Supports lot of platforms: Go, Java, Node.js, PHP, Python, Ruby,Tomcat, Containers
  - Managed platform updates and simplified application releases
  - **No usage charges** - Pay only for AWS resources you provision
- **AWS Lambda**: Serverless Compute
  - **Serverless - Don't worry about servers. Focus on building your app**
    - Remember: **Serverless does NOT mean "No Servers"**
    - Pay for use (invocations) and not for servers (EC2 instances)
    - **Focus on code** and the cloud managed service takes care of all that is needed to scale your code to serve millions of requests!
  - Supports Node.js (JavaScript), Java, Python, Go, C# and more..
  - **Pay for Use**: Number of requests, Duration of requests and Memory Configured
    - Free tier - 1M free requests per month

Amazon Elastic Container Service (Amazon ECS)

In 28 Minutes

- Microservices are built in multiple languages (Go, Java, Python, JavaScript, etc)
- Containers simplify deployment of microservices:
  - Step I : Create a self contained Docker image
    - Application Runtime (JDK or Python), Application code and Dependencies
  - Step II : Run it as a container any where
    - Local machine OR Corporate data center OR Cloud
- How do you manage 1000s of containers?
  - **Elastic Container Service (ECS)** - Fully managed service for container orchestration
    - **Step I** : Create a Cluster (Group of one or more EC2 instances)
    - **Step II**: Deploy your microservice containers
  - **AWS Fargate**: Serverless ECS. DON'T worry about EC2 instances.
  - **Cloud Neutral**: Kubernetes
    - AWS - AWS Elastic Kubernetes Service (EKS)



Compute Services - Quick Review

In 28 Minutes

Service	Description
Amazon EC2 + ELB	Traditional Approach (Virtual Servers + Load Balancing)
AWS Elastic Beanstalk	Simplify management of web applications and batch applications. Automatically creates EC2 + ELB(load balancing and auto scaling)
AWS Elastic Container Service (ECS)	Simplify running of microservices with Docker containers. Run containers in EC2 based ECS Clusters
AWS Fargate	Serverless version of ECS
AWS Lambda	Serverless - Do NOT worry about servers
Amazon Lightsail	Pre-configured development stacks in AWS - LAMP, MEAN. Run websites on WordPress.Low, predictable monthly price.
AWS Batch	Run batch computing workloads on AWS

Storage

Storage in AWS - Overview

In28 Minutes

Type	Description
Object	**Amazon S3** (Very Flexible) Store large objects using a key-value approach
Block	Storage connected to one EC2 instance. Your Hard Disks. **Elastic Block Storage**(EBS - Permanent) **Instance Store** (Ephemeral)
File	File Share. Share storage between EC2 instances. **EFS** (Linux) **FSx Windows** **FSx for Lustre** (High Performance)
Archival	**Amazon S3 Glacier** Extremely low cost storage for archives and long-term backups.
Hybrid	**AWS Storage Gateway** Cloud + On Premise

Amazon S3 (Simple Storage Service)

In28 Minutes

- **Most popular, very flexible & inexpensive** storage service
- Provides **unlimited storage**
- Store large objects using a **key-value** approach(**Object Storage**)
- Provides REST API to access and modify objects
- **Store all file types** - text, binary, backup & archives
- Objects are stored in buckets
  - Unlimited objects in a bucket
  - Each object is identified by a **key value pair**
    - **Key is unique** in a bucket
    - Max object size is **5 TB**
- Cost : Storage (how many GB?) + Usage (Retrieval & Transfers)

Amazon S3

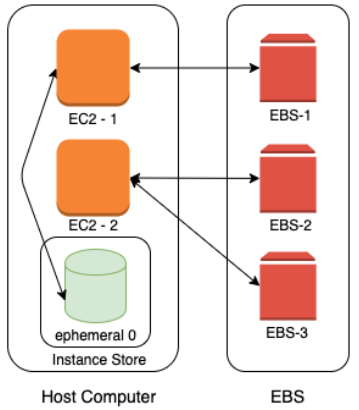
Amazon S3 Storage Classes

In28 Minutes

Storage Class	Scenario	AZs
Standard	Frequently accessed data. First Byte: ms	>=3
Standard-IA	Long-lived, infrequently accessed data (backups for disaster recovery). First Byte: ms	>=3
One Zone-IA	Long-lived, infrequently accessed, non-critical data (Easily re-creatable data - thumbnails for images). First Byte: ms	1
Intelligent-Tiering	Long-lived data with changing or unknown access patterns	>=3
Glacier	Archive data with retrieval times ranging from minutes to hours	>=3
Glacier Deep Archive	Archive data that rarely, if ever, needs to be accessed with retrieval times in few hours	>=3
Reduced Redundancy (Not recommended)	Frequently accessed, non-critical data	>=3

Block Storage

- Two popular types of Block Storage:
  - **Instance Store**: Physically attached to EC2 instance
    - Ephemeral storage - Temporary data (Data lost - hardware fails or instance termination)
    - **CANNOT take a snapshot** or restore from snapshot
    - **Use case**: cache or scratch files
  - **Elastic Block Store (EBS)**: Network Storage
    - More Durable. Very flexible **Provisioned capacity**
    - **Increase size when you need it** - when attached to EC2 instance
    - *99.999% Availability* & replicated within the same AZ
    - **Use case** : Run your custom database

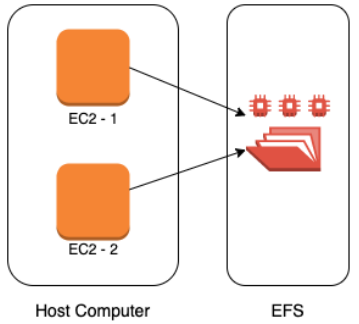


Amazon EBS vs Instance Store

Feature	Elastic Block Store (EBS)	Instance Store
Attachment to EC2 instance	As a network drive	Physically attached
Lifecycle	Separate from EC2 instance	Tied with EC2 instance
Cost	Depends on provisioned size	Zero (Included in EC2 instance cost)
Flexibility	Increase size	Fixed size
I/O Speed	Lower (network latency)	2-100X of EBS
Snapshots	Supported	Not Supported
Use case	Permanent storage	Ephemeral storage

Amazon EFS

- **Petabyte scale, Auto scaling, Pay for use** shared file storage
- Compatible with Amazon EC2 Linux-based instances
- **(Usecases)** Home directories, file share, content management
- (Alternative) Amazon FSx for Lustre
  - File system **optimized for performance**
  - High performance computing (HPC) and media processing use cases
- (Alternative) Amazon FSx Windows File Servers
  - Fully managed Windows file servers
  - Accessible from Windows, Linux and MacOS instances
  - Integrates with Microsoft Active Directory (AD) to support Windows-based environments and enterprises.



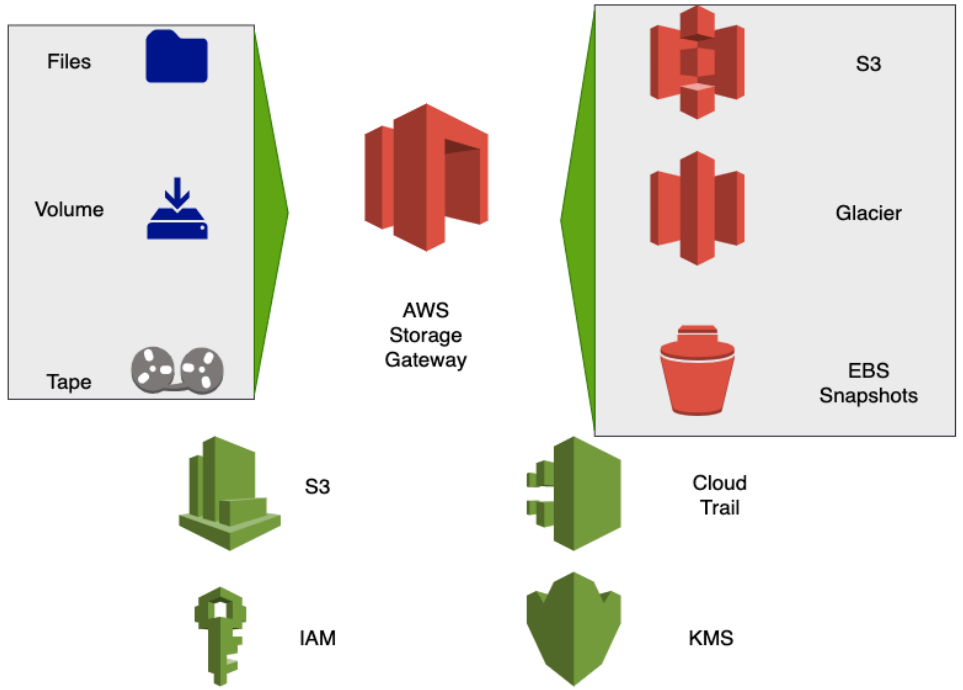
Review of storage options

In28 Minutes

Type	Examples	Latency	Throughput	Shareable
Block	EBS, Instance Store	Lowest	Single	Attached to one instance at a time. Take snapshots to share.
File	EFS, FSx Windows, FSx for Lustre	Low	Multiple	Yes
Object	S3	Low	Web Scale	Yes
Archival	Glacier	Minutes to hours	High	No

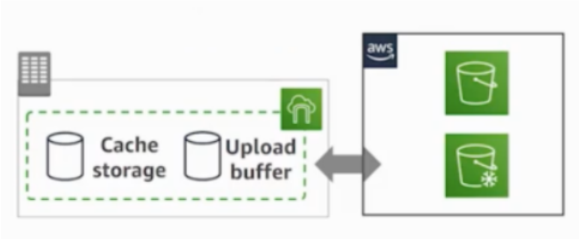
AWS Storage Gateway

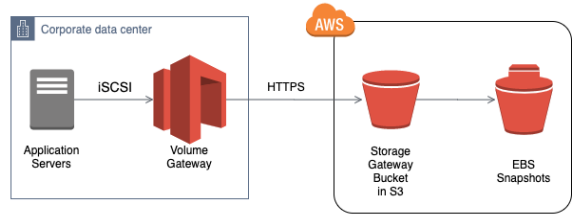
- **Hybrid storage** (cloud + on premise)
- Unlimited cloud storage for on-premise software applications and users with good performance
- (Remember) Storage Gateway and S3 Glacier **encrypt data** by default
- **Three Options**
  - AWS Storage File Gateway
  - AWS Storage Tape Gateway
  - AWS Storage Volume Gateway



AWS Storage Gateway - Types

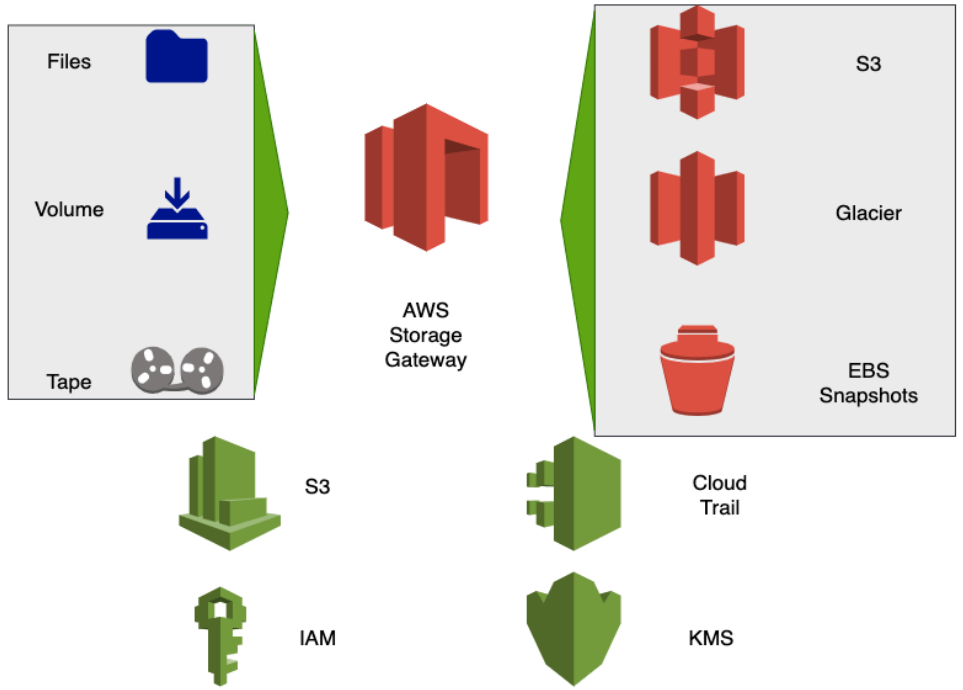
- **Storage File Gateway** - Storage for file shares
  - Files **stored** in **Amazon S3 & Glacier**
- **Storage Tape Gateway** - Virtual tape backups
  - Tapes stored in Amazon S3 & Glacier
    - Avoid complex physical tape backups (wear and tear)
    - **No change needed** for tape backup infrastructure
- **Storage Volume Gateway** : Cloud Block Storage
  - Use cases: Backup, Disaster Recovery, Cloud Migration
  - (Option 1) **Cached** (Gateway Cached Volumes):
    - Primary Data Store - **AWS - Amazon S3**
    - **On-premise cache** stores frequently accessed data
  - (Option 2) **Stored** (Gateway Stored Volumes):
    - Primary Data Store - **On-Premises**
    - Asynchronous copy to AWS
    - Stored as EBS snapshots





AWS Storage Gateway - Review

- Key to look for : **Hybrid storage** (cloud + on premise)
- File share moved to cloud => **AWS Storage File Gateway**
- Tape Backups on cloud => **AWS Storage Tape Gateway**
- Volume Backups on cloud (Block Storage) => **AWS Storage Volume Gateway**
  - High performance => **Stored**
  - Otherwise => **Cached**



Databases in AWS

Databases - Overview

In 28 Minutes

Database Type	AWS Service	Description
Relational OLTP databases	Amazon RDS	Transactional usecases needing **predefined schema** and very **strong transactional** capabilities
Relational OLAP databases	Amazon Redshift	Datewarehouse, reporting, analytics & intelligence apps
Document & Key Databases	Amazon DynamoDB	Apps needing **quickly evolving** semi structured data (**schema-less**) **Terabytes of data** with **millisecond responses** for **millions of TPS** Content management, catalogs, user profiles, shopping carts, session stores and gaming applications
In memory databases/caches	Amazon ElastiCache	Applications needing **microsecond** responses **Redis** - persistent data **Memcached** - simple caches

Amazon RDS (Relational Database Service)

In 28 Minutes

- **Managed relational database service** for OLTP use cases
  - Supports Amazon Aurora, PostgreSQL, MySQL, MariaDB (Enhanced MySQL), Oracle Database, and SQL Server
  - Manage setup, backup, scaling, replication and patching of your relational databases
- **Features:**
  - Multi-AZ deployment (standby in another AZ)
  - Read replicas (Same AZ or Multi AZ (Availability+) or Cross Region(Availability++))
  - Automated backups (restore to point in time)
  - Manual snapshots
- Amazon Aurora: MySQL and PostgreSQL-compatible
  - Provides "Global Database" option
    - Up to five read-only, secondary AWS Regions (Low latency for global reads)

Amazon RDS (Relational Database Service) - Remember

In28 Minutes

- AWS is responsible for
  - Availability, Durability, Backups, Scaling (according to your configuration) and Maintenance (patches according to your configuration)
- You are responsible for
  - Choosing database type
  - Managing database users
  - Creating schema (tables, indexes etc)
  - Schema optimization
- You CANNOT
  - SSH into database EC2 instances or setup custom software (NOT ALLOWED)
  - Install OS or DB patches. RDS takes care of them (NOT ALLOWED)

Amazon Redshift

In28 Minutes

- **Amazon Redshift** - Relational DB for OLAP (reads >>>> writes)
  - **Petabyte-scale distributed data ware house** based on PostgreSQL
    - Traditional ETL(Extract, Transform, Load), OLAP and Business Intelligence (BI) use cases
    - Supports SQL, Tables and Relationships
  - Massively parallel processing (MPP):
    - A single row of data might be stored across multiple nodes (Columnar Storage)
    - A query to Redshift leader node is distributed to multiple compute nodes for execution

Amazon Redshift vs Alternatives

In 28 Minutes

Alternative	Scenario
Amazon Redshift	Run complex queries (SQL) against data warehouse - housing structured and unstructured data pulled in from a variety of sources
Amazon EMR	Managed Hadoop. Large scale data processing with high customization (machine learning, graph analytics)
Amazon Redshift Spectrum	Run queries directly against S3 without worrying about loading entire data from S3 into a data warehouse. Scale compute and storage independently.
Amazon Athena	Quick ad-hoc queries without worrying about provisioning a compute cluster (serverless) Amazon Redshift Spectrum is recommended if you are executing queries frequently against structured data.

Amazon DynamoDB

- Schemaless NoSQL key-value & document database
- Single-digit millisecond responses for million of TPS
- Do not worry about scaling, availability or durability
- No need to create a database:
  - Create a table and configure read and write capacity (RCU and WCU)
  - Automatically scales to meet your RCU and WCU
- Use cases: User profiles, shopping carts, high volume read write applications

```
{
    "id": 1,
    "name": "Jane Doe",
    "username": "abcdefgh",
    "email": "someone@gmail.com",
    "address": {
      "street": "Some Street",
      "suite": "Apt. 556",
      "city": "Hyderabad",
      "zipcode": "500018",
      "geo": {
        "lat": "-3.31",
        "lng": "8.14"
      }
    },
    "phone": "9-999-999-9999",
    "website": "in28minutes.com",
    "company": {
      "name": "in28minutes"
    }
}
```

DynamoDB vs RDS

In28 Minutes

Feature	DynamoDB	RDS
Scenario	Millisecond latency with millions of TPS	Stronger consistency (schema) and transactional capabilities
Schema	Schemaless (needs only a primary key - Great for use cases where your schema is evolving)	Well-defined schema with relationships
Data Access	Using REST API provided by AWS using AWS SDKs or AWS Management Console or AWS CLI	SQL queries
Complex Data Queries Involving Multiple Tables	Difficult to run	Run complex relational queries with multiple entities
Scaling	No upper limits	64 TB

Amazon ElastiCache

In28 Minutes

- Retrieving data from memory is faster than from disk
- Highly scalable & low latency in-memory data store
- Used for distributed caching
- (Option 1) ElastiCache Memcached:
  - Low maintenance simple caching solution
  - Easy horizontal scaling (multiple nodes)
  - Use case: Speed up database-driven websites by caching data
- (Option 2) ElastiCache Redis:
  - Persistence
  - Advanced Features:
    - Publish subscribe messaging
    - Read replicas and failover
  - Usecases: gaming leader boards, queues, real-time analytics

ElastiCache

Databases - Review

In28 Minutes

Database Type	AWS Service	Description
Relational OLTP databases	Amazon RDS	Transactional usecases needing **predefined schema** and very **strong transactional** capabilities
Relational OLAP databases	Amazon Redshift	Reporting, analytics & intelligence apps needing **predefined schema**
Document & Key Databases	Amazon DynamoDB	Apps needing **quickly evolving** semi structured data (**schema-less**) Scale to **terabytes of data** with **millisecond responses** upto **millions of TPS** Content management, catalogs, user profiles, shopping carts, session stores and gaming applications
In memory databases/caches	Amazon ElastiCache	Applications needing **microsecond** responses **Redis** - persistent data **Memcached** - simple caches

Databases - Scenarios

In28 Minutes

Scenario	Solution
A start up with quickly evolving tables	DynamoDB
Transaction application needing to process million transactions per second	DynamoDB
Very high consistency of data is needed while processing thousands of transactions per second	RDS
Cache data from database for a web application	Amazon ElastiCache
Relational database for analytics processing of petabytes of data	Amazon Redshift

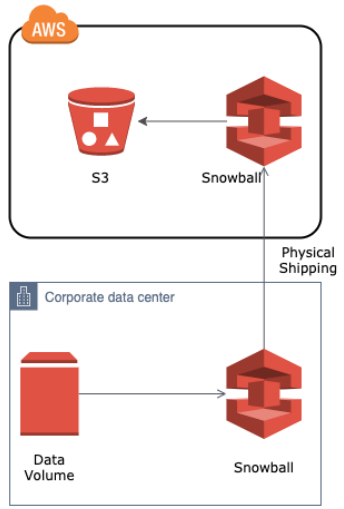
Other Database/Storage Services

In28 Minutes

- Amazon DocumentDB
  - Managed document database service
  - Compatible with MongoDB
- Amazon Keyspaces
  - Managed service for Apache Cassandra
- AWS Backup
  - Centrally manage and automate backups across AWS services
  - Automate backup compliance and monitoring

AWS Snowball and AWS Snowmobile

- AWS Snowball: Transfer 100s of terabytes to petabytes data from onpremises to AWS
  - 100TB (80 TB usable) per appliance with automatic encryption (KMS)
  - Simple Process: Request for Snowball, Copy data and Ship it back
  - Manage jobs with AWS Snowball console
- Current versions of AWS Snowball use Snowball Edge devices
  - Provide both compute and storage(Storage or Compute or GPU Optimized)
  - Pre-process data (using Lambda functions)
- Use **Snowmobile** Trucks (100PB per truck) for dozen petabytes to exabytes



Networking in AWS

Amazon VPC (Virtual Private Cloud) and Subnets

In 28 Minutes

- VPC (Virtual Private Cloud) - Your **own isolated network** in AWS cloud
  - Network traffic within a VPC is isolated (not visible) from all other Amazon VPCs
  - You **control all the traffic** coming in and going outside a VPC
  - **(Best Practice)** Create AWS resources **in a VPC**
    - Secure resources from unauthorized access AND
    - Enable secure communication between your cloud resources
    - Each VPC is created in a Region
- **Subnet - Separate public resources from private resources** in a VPC
  - **Create different subnets** for public and private resources
    - Resources in a public subnet **CAN** be accessed from internet
    - Resources in a private subnet **CANNOT** be accessed from internet
    - BUT resources in public subnet can talk to resources in private subnet
    - Each Subnet is created in an Availability Zone
    - VPC - us-east-1 => Subnets - AZs us-east-1a or us-east-1b or ..

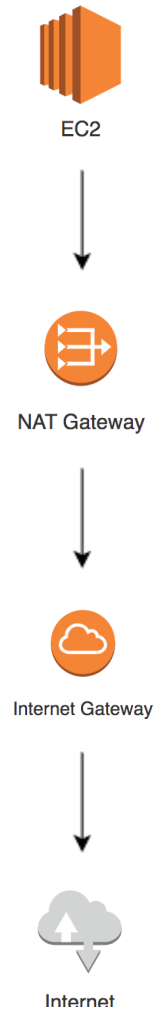
Public Subnet vs Private Subnet

- **Public Subnet**: Communication allowed - Internet to Subnet
- An **Internet Gateway** enables internet communication for public subnets
  - **Public Subnet**: Subnet having a route to an internet gateway
  - **Private Subnet**: Subnet **DOES NOT** have route to an internet gateway

Private Subnet - NAT Devices - Download Patches

In28 Minutes

- **Allow instances in a private subnet to download software patches** while denying inbound traffic from internet
- Three Options:
  - **NAT Instance**: Install a EC2 instance with specific NAT AMI and configure as a gateway
    - You are taking complete responsibility of availability
  - **NAT Gateway**: Managed Service (PREFERRED - No maintenance, more availability & high bandwidth)
  - **Egress-Only Internet Gateways**: For IPv6 subnets (NAT Gateway supports **IPv4 ONLY**)



Network Access Control List

- Security groups control traffic to a specific resource in a subnet
- NACL provides **stateless firewall** at subnet level
  - Stop traffic from **even entering the subnet**
- Each subnet **must** be associated with a NACL
  - **Default NACL** allows all inbound and outbound traffic.
  - **Custom created NACL** denies all inbound and outbound traffic by default.
  - Rules have a priority number.
    - Lower number => Higher priority.

Security Group vs NACL

Feature	Security Group	NACL
Level	Assigned to a specific instance(s)/resource(s)	Configured for a subnet. Applies to traffic to all instances in a subnet.
Rules	Allow rules only	Both allow and deny rules

VPC - Important Concepts

In28 Minutes

- **VPC Peering** - Connect VPCs from same or different AWS accounts (across regions)
  - Allows private communication between the connected VPCs
  - Peering uses a request/accept protocol (Owner of requesting VPC sends a request)
  - Peering is not transitive.
- **VPC Endpoint** - Securely connect your VPC to another service
  - Gateway endpoint: Securely connect to Amazon S3 and DynamoDB
  - Interface endpoint: Securely connect to AWS services EXCEPT FOR Amazon S3 and DynamoDB
    - Powered by PrivateLink (keeps network traffic within AWS network)
  - (Avoid DDoS & MTM attacks) Traffic does NOT go thru internet
  - (Simple) Does NOT need Internet Gateway, VPN or NAT

VPC Flow Logs

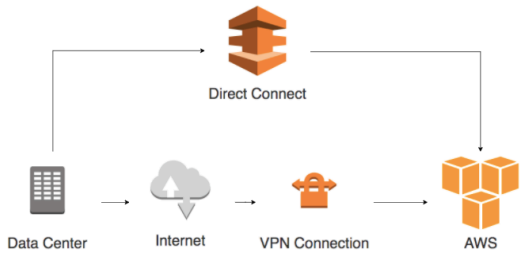
In28 Minutes

- Monitor network traffic
- Troubleshoot connectivity issues (NACL and/or security groups misconfiguration)
- Capture traffic going in and out of your VPC (network interfaces)
- Can be created for
  - a VPC
  - a subnet
- Publish logs to Amazon CloudWatch Logs or Amazon S3
- Flow log records contain ACCEPT or REJECT
  - Is traffic is permitted by security groups or network ACLs?

VPC Flow Logs

AWS and On-Premises - Overview

- **AWS Managed VPN**: Tunnels from VPC to on premises
  - Traffic over internet - encrypted using IPsec protocol
  - **VPN gateway** to connect one VPC to customer network
  - **Customer gateway** installed in customer network
    - You need a Internet-routable IP address of customer gateway
- **AWS Direct Connect (DX)**: Private dedicated network connection to on premises
  - (Advantage) Reduce your (ISP) bandwidth costs
  - (Advantage) Consistent Network performance (private network)
  - Connection options: Dedicated (1 Gbps or 10 Gbps) or Hosted (Shared 50Mbps to 10 Gbps)
  - (Caution) Establishing DC connection takes a month
  - (Caution) Establish a redundant DC for maximum reliability
  - (Caution) Data is NOT encrypted (Private Connection ONLY)



VPC - Review

- **VPC**: Virtual Network to protect resources and communication from outside world.
- **Subnet**: Seperate private resources from public resources
- **Internet Gateway**: Allows Public Subnets to connect/accept traffic to/from internet
- **NAT Gateway**: Allow internet traffic from private subnets
- **VPC Peering**: Connect one VPC with another VPC
- **VPC Flow Logs**: Enable logs to debug problems
- **AWS Direct Connect**: Private pipe from AWS to on-premises
- **AWS VPN**: Encrypted (IPsec) tunnel over internet to on-premises

Security and Encryption in AWS

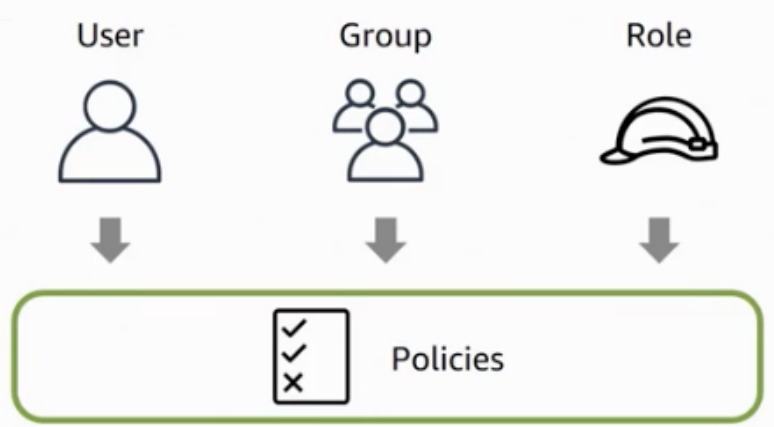
AWS Identity and Access Management (IAM)

In28 Minutes

- **Authentication** (is it the right user?) and
- **Authorization** (do they have the right access?)
- **Identities** can be
  - AWS users or
  - Federated users (externally authenticated users)
- Provides very **granular** control
  - Limit a single user:
    - to perform single action
    - on a specific AWS resource
    - from a specific IP address
    - during a specific time window

Important IAM Concepts

- **IAM users**: Users created in an AWS account
  - Has credentials attached (name/password or access keys)
- **IAM groups**: Collection of IAM users
- **Roles**: Temporary identities
  - Does NOT have credentials attached
  - (Advantage) Expire after a set period of time
  - Used to give access to federated users or EC2 instances
- **Policies**: Define permissions



IAM Best Practices - Recommended by AWS

- **Users** – Create individual users
- **Groups** – Manage permissions with groups
- **Permissions** – Grant least privilege
- **Auditing** – Turn on AWS CloudTrail
- **Password** – Configure a strong password policy
- **MFA** – Enable MFA for privileged users
  - (Hardware device - Gemalto, Virtual device - An app on a smart phone)
- **Roles** – Use IAM roles for Amazon EC2 instances
- **Sharing** – Use IAM roles to share access
- **Rotate** – Rotate security credentials regularly
- **Root** – Reduce or remove use of root

KMS and Cloud HSM

- You need keys to encrypt your data in various storage and database options in AWS
- **KMS** & **Cloud HSM** - Generate, store, use and replace your keys in AWS
- **KMS**: Create & manage **cryptographic keys**
  - **KMS** integrates with all storage and database services in AWS
  - **Automatically rotate master keys** once a year
  - **Schedule key deletion** to verify if the key is used
    - Mandatory minimum wait period of 7 days (max-30 days)
- **CloudHSM**: **Dedicated single-tenant** HSM for regulatory compliance
  - (Remember) AWS KMS is a multi-tenant service
  - **Use Cases**: Compliance to Regulations, Very high security

AWS KMS

Cloud HSM

AWS Shield

- Shields from Distributed Denial of Service (DDoS) attacks
  - Disrupt normal traffic of a server by overwhelming it with a flood of Internet traffic
  - Protect Amazon Route 53, CloudFront, EC2 instances and Elastic Load Balancers (ELB)
- AWS Shield Standard is automatically enabled (ZERO COST)
  - Protection against common infrastructure (layer 3 and 4) DDoS attacks
- Enable AWS Shield Advanced ($$$) for Enhanced Protection:
  - 24x7 access to the AWS DDoS Response Team (DRT)
  - Protects your AWS bill from usage spikes as a result of a DDoS attack

AWS Shield

AWS WAF - Web Application Firewall

In 28 Minutes

- AWS WAF protect your web applications from OWASP Top 10 exploits, CVE and a lot more!
  - OWASP (Open Web Application Security Project) Top 10
    - List of broadly agreed "**most critical security risks to web applications**"
    - Examples : SQL injection, cross-site scripting etc
  - Common Vulnerabilities and Exposures (CVE) is a list of information-security vulnerabilities and exposures
- Can be deployed on Amazon CloudFront, Application Load Balancer, Amazon API Gateway
- Customize rules & trigger realtime alerts (CloudWatch Alarms)
- Web traffic filtering : block attacks
  - Filter traffic based on IP addresses, geo locations, HTTP headers and body (block attacks from specific user-agents, bad bots, or content scrapers)

AWS WAF

Other Important Security Services

In28 Minutes

Solution	Description
Amazon Macie	Fully managed data security and privacy service Uses machine learning to identify sensitive data in Amazon S3 (Recommendation) When migrating data to AWS use S3 for staging and Run Macie
Amazon GuardDuty	Continuously monitor AWS environment for suspicious activity (Intelligent Threat Detection) Analyze AWS CloudTrail events, VPC Flow Logs etc
Certificate Manager	Provision, manage, deploy, and renew SSL/TLS certificates on the AWS platform
Penetration Testing	Testing application security by simulating an attack You do NOT need permission from AWS to do penetration testing on a limited set of services (EC2 instances, ELB, RDS, CloudFront, API Gateway, Lambda, Elastic BeanStalk)
AWS Single Sign On	Cloud-based single sign-on (SSO) service. Centrally manage SSO access to all of your AWS accounts (SAML and Microsoft AD support. Integrates with AWS Organizations)

Other Important Security Services

In28 Minutes

Solution	Description
AWS Artifact	Self-service portal for on-demand access to AWS compliance reports, certifications, accreditations, and other third-party attestations. Review, accept, and manage your agreements with AWS.
AWS Security Hub	Consolidated view of your security status in AWS. Automate security checks, manage security findings, and identify the highest priority security issues across your AWS environment.
Amazon Detective	Investigate and quickly identify the root cause of potential security issues. Automatically collect log data from your AWS resources and uses machine learning to help you visualize and conduct security investigations.

Monitoring and Governance in AWS

AWS CloudTrail and AWS Config

In28 Minutes

Solution	Description
AWS CloudTrail	Track events, API calls, changes made to your AWS resources. Who (made the request), What (action, parameters, end result) and When? **Multi Region Trail** - One trail for all AWS regions vs **Single Region Trail** - Only events from one region
AWS Config	**Auditing**: Complete inventory of your AWS resources **Resource history and change tracking** - Find how a resource was configured at any point in time **Governance** - Customize Config Rules for specific resources or for entire AWS account and Continuously evaluate compliance against desired configuration
AWS Config vs AWS CloudTrail	AWS Config - What did my AWS resource look like? AWS CloudTrail - Who made an API call to modify this resource?

Monitoring AWS with Amazon CloudWatch

In 28 Minutes

- Monitoring and observability service
- Collects monitoring and operational data in the form of logs, metrics, and events
- Set alarms, visualize logs, take automated actions and troubleshoot issues
- Integrates with more than 70 AWS services:
  - Amazon EC2
  - Amazon DynamoDB
  - Amazon S3
  - Amazon ECS
  - AWS Lambda
  - and ....

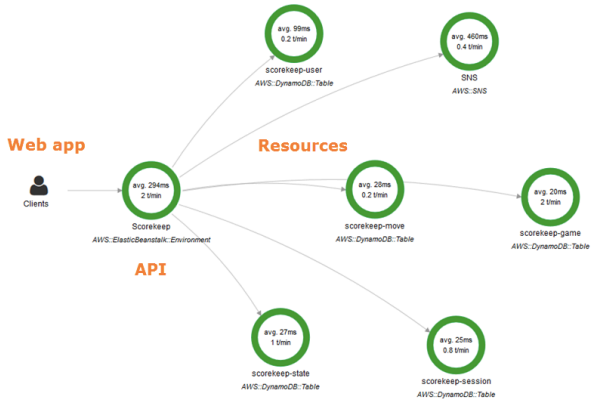
Cloudwatch

Amazon CloudWatch - Options

Solution	Description
Amazon CloudWatch Metrics	Metrics for AWS services Example **EC2**: CPUUtilization, NetworkIn, NetworkOut
Amazon CloudWatch Logs	Monitor and troubleshoot using system, application and custom log files
Amazon CloudWatch Alarms	**Create alarms** to take immediate action Execute an Auto Scaling policy based on CPU utilization of EC2 instance or Amazon SQS queue length Send a SNS event notification (email) based on Amazon DynamoDB table throughput
Amazon CloudWatch Events	**Act** based on **events on AWS resources** Call a AWS Lambda function or send an email when an EC2 instance starts (ADDITIONAL FEATURE) **Schedule events** - Schedule hourly call to Lambda function
Amazon EventBridge	Extends CloudWatch Events and helps you build event driven architectures

X-Ray

- Trace request across microservices/AWS services
  - Analyze, Troubleshoot errors, Solve performance issues
  - Gather tracing information
    - From applications/components/AWS Services
  - Tools to view, filter and gain insights (Ex: Service Map)
- How does Tracing work?
  - **Unique trace ID** assigned to every client request
    - `X-Amzn-Trace-Id:Root=1-5759e988-bd862e3fe`
  - Each service in request chain sends traces to X-Ray with trace ID
    - X-Ray gathers all the information and provides visualization

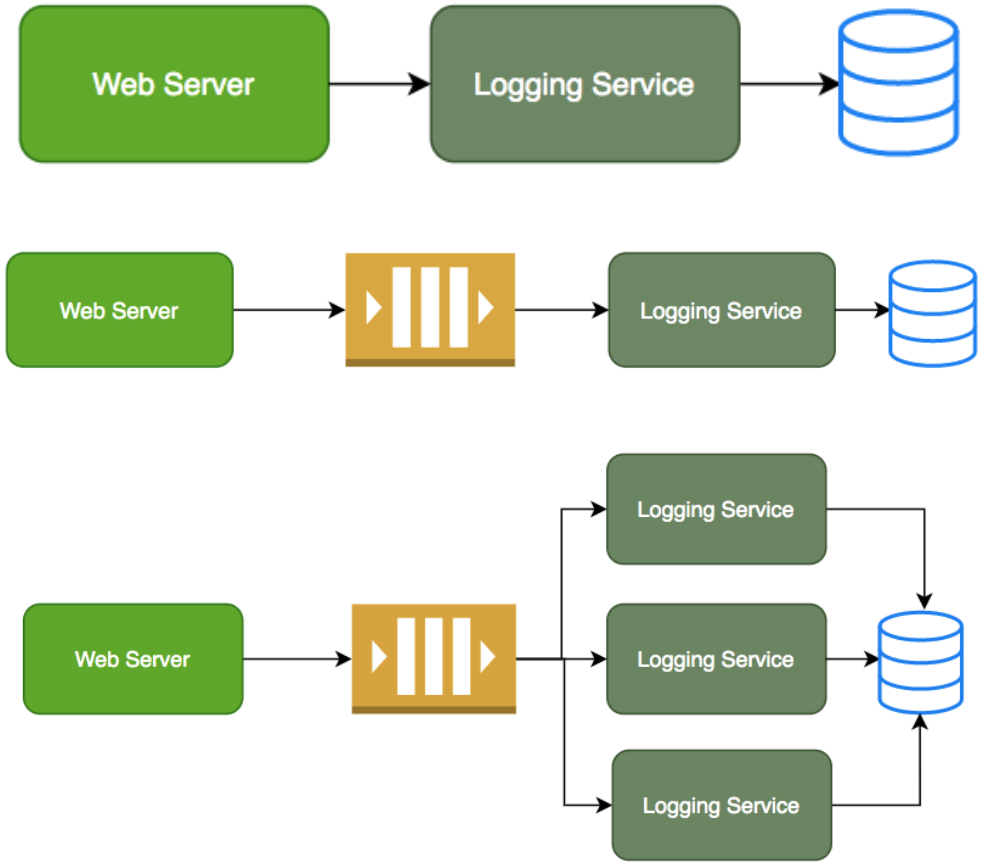


Decoupling Applications - SQS, SNS and Kinesis

Why do we need Asynchronous Communication?

In28 Minutes

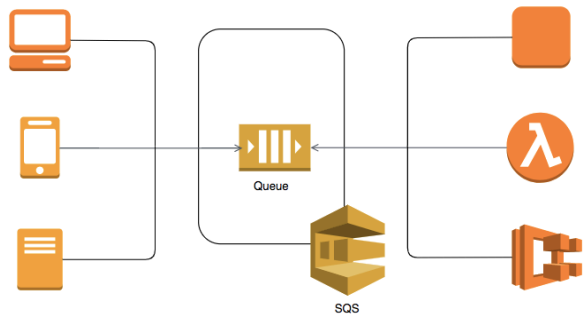
- Synchronous Communication:
  - What if your logging service goes down?
    - Will you applications go down too?
  - What if there is high load?
    - Log Service unable to handle and goes down
- Asynchronous Communication:
  - Create a queue or a topic
    - Your applications put the logs on the queue
    - Picked up when the logging service is ready
  - Good example of decoupling!
  - (Possible) Multiple logging service instances reading from the queue!



Asynchronous Communication - Pull Model - SQS

In28 Minutes

- Producers put messages. Consumers poll on queue.
  - Only one of the consumers will successfully process a message
- **Advantages**:
  - Scalability: Scale consumer instances under high load
  - Availability: Producer up even if a consumer is down
  - Reliability: Work is not lost due to insufficient resources
  - Decoupling: Make changes to consumers without effect on producers worrying about them
- **Features**:
  - Reliable, scalable, fully-managed message queuing service
  - High availability
  - Unlimited scaling
    - Auto scale to process billions of messages per day
  - Low cost (Pay for use)



Standard and FIFO Queues

In28 Minutes

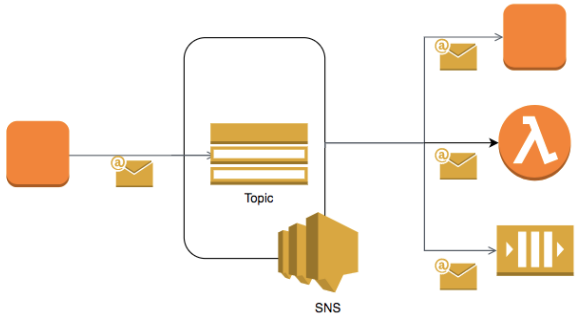
- Standard Queue
  - Unlimited throughput
  - BUT NO guarantee of ordering (Best-Effort Ordering)
  - and NO guarantee of exactly-once processing
    - Guarantees at-least-once delivery (some messages can be processed twice)
- FIFO (first-in-first-out) Queue
  - First-In-First-out Delivery
  - Exactly-Once Processing
  - BUT throughput is lower
    - Upto 300 messages per second (300 send, receive, or delete operations per second)
    - If you batch 10 messages per operation (maximum), up to 3,000 messages per second
- Choose
  - Standard SQS queue if throughput is important
  - FIFO Queue if order of events is important

Amazon SQS

Asynchronous Communication - Push Model - SNS

In28 Minutes

- How does it work (Publish-Subscribe(pub-sub))?
  - Create an SNS Topic
  - Subscribers can register for a Topic
  - When an SNS Topic receives an event notification (from publisher), it is broadcast to all Subscribers
  - (Advantage) Decoupling: Producers don't care about Consumers
  - (Advantage) Availability: Producer up even if subscriber is down
- Use Cases : Monitoring Apps, workflow systems
- Provides mobile and enterprise messaging services
  - Push notifications to Apple, Android, FireOS, Windows devices
  - Send SMS to mobile users and Emails
- REMEMBER : SNS does not need SQS or a Queue



Amazon Kinesis

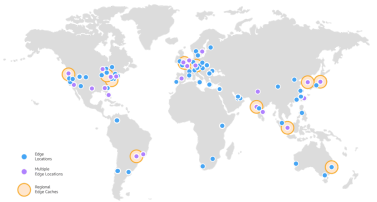
- Handle streaming data (NOT recommended for ETL Batch Jobs)
- **Amazon Kinesis Data Streams** (Alternative for Kafka)
  - Limitless Real time stream processing with Sub second latency
    - Supports multiple clients(Each client can track their stream position)
    - Retain and replay data (max 7 days & default 1 day)
- **Amazon Kinesis Firehose**: Data ingestion for streaming data
  - Receive > Process (transform - Lambda, compress, encrypt) > Store (S3, Elasticsearch, Redshift and Splunk)
    - Use existing analytics tools based on S3, Redshift and Elasticsearch
- **Amazon Kinesis Analytics**: Continuously analyze streaming data
  - Run SQL queries and write Java apps (find active users in last 5 minutes)
- **Amazon Kinesis Video Streams**: Monitor video streams - web-cams
  - Integrate with machine learning to get intelligence (Examples: traffic lights)

Routing and Content Delivery

Amazon CloudFront - Content Delivery Network

In 28 Minutes

- Deliver content to your global audience:
  - AWS provides 200+ edge locations around the world
  - Provides high availability and low latency
  - Serve users from nearest edge location (based on user location)
  - If content is not available at the edge location, it is retrieved from the origin server and cached
- Content Source - S3, EC2, ELB or External Websites
- Use Cases: Static web apps, Downloads (media/software)
- Integrates with: AWS Shield (Avoid DDoS attacks)
  - AWS WAF (protect from SQL injection, cross-site scripting)
- Cost Benefits: Zero cost for transfer from S3 to CloudFront
  - Reduce compute workload for your EC2 instances



Route 53 = Domain Registrar + DNS (Domain Name Server)

In28 Minutes

- What would be the steps in setting up a website with a domain name (for example, in28minutes.com)?
  - Step I : Buy the domain name in28minutes.com (Domain Registrar)
  - Step II : Setup your website content (Website Hosting)
  - Step III : Route requests to in28minutes.com to the my website host server (DNS)

- Route 53 = Domain Registrar + DNS
  - Domain Registrar - Buy your domain name
  - DNS - Setup your DNS routing for in28minutes.com
    - Configure Records for routing traffic for in28minutes.com
      - Route api.in28minutes.com to the IP address of api server
      - Route static.in28minutes.com to the IP address of http server
      - Route email (ranga@in28minutes.com) to the mail server(mail.in28minutes.com)
      - Each record is associated with a TTL (Time To Live) - How long is your mapping cached at the routers and the client?

Route 53 Routing Policies

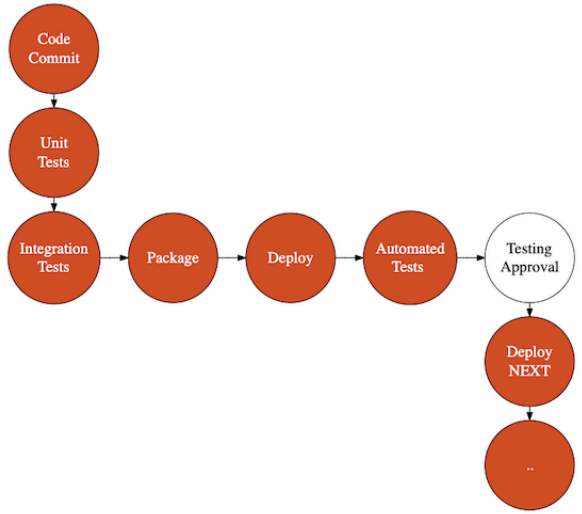
In28 Minutes

Policy	Description
Simple	Maps a domain name to (one or more) IP Addresses
Weighted	Maps a single DNS name to multiple weighted resources 10% to A, 30% to B, 60% to C (useful for canary deployments)
Latency	Choose the option with minimum latency Latency between hosts on the internet can change over time
Failover	Active-passive failover. Primary Health check fails (optional cloud Watch alarm) => DR site is used
Geoproximity	Choose the nearest resource (geographic distance) to your user. Configure a bias.
Multivalue answer	Return multiple healthy records (upto 8) at random You can configure an (optional) health check against every record
Geolocation	Choose based on the location of the user

DevOps

DevOps

- Get Better at "**3 Elements of Great Software Teams**"
  - **Communication** - Get teams together
  - **Feedback** - Earlier you find a problem, easier it is to fix
  - **Automation** - Testing, deployment, monitoring etc
- CI/CD Practices:
  - **Continuous Integration**: Continuously run tests & packaging
  - **Continuous Deployment**: Continuously deploy to test env
  - **Continuous Delivery**: Continuously deploy to production
- CI/CD Tools:
  - **AWS CodeCommit** - Private source control (Git)
  - **AWS CodePipeline** - Orchestrate CI/CD pipelines
  - **AWS CodeBuild** - Build and Test Code (packages and containers)
  - **AWS CodeDeploy** - Automate Deployment(ECS, Lambda etc)



DevOps - Practices - IAC (Infrastructure as Code)

- Treat infrastructure the same way as application code
- Track your infrastructure changes over time (version control)
- Bring repeatability into your infrastructure
- Two Key Parts
  - **Infrastructure Provisioning**
    - Provisioning compute, database, storage and networking
    - Open source cloud neutral - Terraform
  - **Configuration Management**
    - Install right software and tools on the provisioned resources
    - Open Source Tools - Chef, Puppet, Ansible

DevOps - IAC (Infrastructure as Code) - AWS

- **Infrastructure Provisioning**
  - Open Source: Terraform
  - AWS:
    - AWS CloudFormation: Provision AWS Resources
    - AWS SAM (Serverless Application Model):Provision Serverless Resources
- **Configuration Management**
  - Open Source Tools: Chef, Puppet, Ansible
  - AWS Service: OpsWorks (Chef, Puppet in AWS)
- (Remember) Most DevOps Tools AutoScale
  - CodeCommit, CodePipeline, CodeBuild, CodeDeploy, CloudFormation, OpsWorks

AWS CloudFormation

In28 Minutes

- Lets consider an example:
  - I would want to create a new VPC and a subnet
  - I want to provision a ELB, ASG with 5 EC2 instances & RDS database
- AND I would want to create 4 environments
  - Dev, QA, Stage and Production!
- CloudFormation can help you do all these with a simple script!
- Advantages (Infrastructure as Code - IAC & CloudFormation) :
  - Automate deployment of AWS resources in a controlled, predictable way
  - Avoid mistakes with manual configuration
  - Think of it as version control for your environments
- Free to use - Pay only for the resources provisioned
  - Get an automated estimate for your configuration

CloudFormation

Management Services in AWS

AWS Organizations

- **AWS Organizations**: Simple management for multiple AWS accounts
  - Organize accounts into Organizational Units (OU)
  - Consolidated bill for AWS accounts
  - Centralized management for AWS Config Rules
  - Send AWS CloudTrail data to one S3 bucket (across accounts)
  - AWS Firewall Manager to manage firewall rules (WAF, Shield and Security Groups)
- Use **Service control policies(SCP)** to define cross account restrictions
  - Require Amazon EC2 instances to use a specific type
  - Require MFA to stop an Amazon EC2 instance
  - Require a tag upon resource creation
- Use **AWS Resource Access Manager** to share AWS resources:
  - Share AWS Transit Gateways, Subnets, AWS License Manager configurations, Amazon Route 53 Resolver rules with other AWS accounts or your AWS Organization

(Optimize costs)

AWS Trusted Advisor

- Cost optimization, performance, security & fault tolerance recommendations
- **4 FREE Checks**:
  - Service limits (usage > 80%)
  - Security groups having unrestricted access (0.0.0.0/0)
  - Proper use of IAM
  - MFA on Root Account
- Enable **Business or Enterprise AWS support plan** for over 50 checks
  - Cost Optimization: Unused resources, Other opportunities (ex: reserved instances)
  - Security : Settings to make your AWS solution more secure (ex: security group)
  - Fault Tolerance: Redundancy improvements, over-utilized resources
  - Performance: Improve speed and responsiveness of your AWS solutions
  - Service Limits: Is your usage is more than 80% of service limits?

Billing and Cost Management Services/Tools

In 28 Minutes

Service	Description
AWS Billing and Cost Management	Pay your AWS bill, monitor your usage **Cost Explorer** - View your AWS cost data as a graph (Filter by Region, AZ, tags etc. See future cost projection.) **AWS Budgets** - Create a budget (Create alerts (SNS)) Recommendation: Enable Cost allocation tags. Helps you categorize your resource costs in Cost Management.
AWS Compute Optimizer	Recommends compute optimizations to reduce costs (Ex: Right-sizing - EC2 instance type, Auto Scaling group configuration)
AWS Pricing Calculator (NEW)	Estimate cost of your architecture solution
AWS Simple Monthly Calculator (OLD)	Estimate charges for AWS services
TCO - Total Cost of Ownership Calculator (OLD)	Compare Cost of running applications in AWS vs On Premise

More Management Services

Service	Description
AWS Marketplace	Digital catalog to find, test, buy, and deploy licensed software solutions using flexible pricing options: Bring Your Own License (BYOL), free trial, pay-as-you-go, hourly, monthly etc.
Resource Groups	Group your AWS resources. Automate Tasks using AWS Systems Manager. Get group related insights from AWS Config and CloudTrail.
AWS Systems Manager	Run commands(operational tasks) on Amazon EC2 instances. Manage your OS and Database patches.
Personal Health Dashboard	Personalized alerts when AWS is experiencing events that may impact you Provides troubleshooting guidance

Other AWS Services

Service	Description
AWS Professional Services	Get help from AWS for your cloud migration Get technical expertise and advise from AWS Teams for Application Migration, Application Modernization etc
AWS Partner Network	Consulting and technology firms that help enterprises make the best use of AWS Get help with design, architecture, build, connectivity and migration to AWS
AWS Service Quotas	AWS account has Region-specific default quotas or limits for each service (You don't need to remember all of them) Service Quotas allows you to manage your quotas for over 100 AWS services, from one location

More Serverless Architecture (Beyond Lambdas)

Amazon API Gateway

In28 Minutes

- Most applications today are built around REST API:
  - Resources (`/todos`, `/todos/{id}`, etc.)
  - Actions - HTTP Methods - GET, PUT, POST, DELETE etc.
- Management of REST API is not easy:
  - You've to take care of authentication and authorization
  - You've to be able to set limits (rate limiting, quotas) for your API consumers
  - You've to take care of implementing multiple versions of your API
  - You would want to implement monitoring, caching and a lot of other features..
- **"Amazon API Gateway"** - **"front door"** to your APIs
  - **Fully managed - "publish, maintain, monitor, and secure APIs at any scale"**
  - Integrates with AWS Lambda or any web application
  - Supports HTTP(S) and WebSockets
  - Serverless. **Pay for use** (API calls and connection duration)

Amazon API Gateway - Remember

- Run multiple versions of the same API
- Rate Limits(request quota limits) and Throttling
- Fine-grained access permissions using API Keys for Third-Party Developers

Amazon Cognito

- Add authentication and authorization to your mobile and web apps
  - **Integrate with web identity providers** (ex: Google, Facebook)
  - Add **multi-factor authentication (MFA)**, phone and email verification
  - Sync user data across devices, platforms, and applications
- **User Pools**: Create **your own secure and scalable user directory**
  - Create **sign-up (or registration) pages**
  - Customizable **web UI to sign in users** (with option to social sign-in)
  - Integrates with Application Load Balancer and API Gateway
  - Provides triggers to customize workflow - **Pre Authentication Lambda Trigger**, **Pre Sign-up Lambda Trigger**,**Post Confirmation Lambda Trigger** etc
- **Identity pools**: Provide **access to AWS resources** to **your users**
  - Integrate with your own user pool or OpenID Connect provider (Amazon, Apple, Facebook, Google+, Twitter) or SAML identity providers (Corporate)
  - Allows **multiple** authentication (identity) providers

Serverless Application Model

In28 Minutes

- 1000s of Lambda functions to manage, versioning, deployment etc
- Serverless projects can become a maintenance headache
- How to test serverless projects with Lambda, API Gateway and DynamoDB in your local?
- How to ensure that your serverless projects are adhering to best practices?
  - Tracing (X-Ray), CI/CD(CodeBuild, CodeDeploy, CodePipeline) etc
- Welcome SAM - Serverless Application Model
  - Open source framework for building serverless applications
  - Define a YAML with all the serverless resources you want:
    - Functions, APIs, Databases etc
  - BEHIND THE SCENES : Your configuration is used to create a AWS CloudFormation syntax to deploy your application

AWS Step Functions

In 28 Minutes

- Create a serverless workflow in 10 Minutes using a visual approach
- Orchestrate multiple AWS services into serverless workflows:
  - Invoke an AWS Lambda function
  - Run an Amazon Elastic Container Service or AWS Fargate task
  - Get an existing item from an Amazon DynamoDB table or put a new item into a DynamoDB table
  - Publish a message to an Amazon SNS topic
  - Send a message to an Amazon SQS queue
- Build workflows as a series of steps:
  - Output of one step flows as input into next step
  - Retry a step multiple times until it succeeds
  - Maximum duration of 1 year

Architecture and Best Practices

Well Architected Framework

In28 Minutes

- Helps cloud architects build application infrastructure which is:
  - Secure
  - High-performing
  - Resilient and
  - Efficient
- Five Pillars
  - Operational Excellence
  - Security
  - Reliability
  - Performance Efficiency
  - Cost Optimization

Operational Excellence Pillar

In28 Minutes

- Avoid/Minimize effort and problems with:
  - Provisioning servers, Deployment, Monitoring and Support
- Recommendations:
  - Use Managed Services: No worry about managing servers, availability etc
  - Go serverless: Prefer Lambda to EC2!
  - Automate with Cloud Formation: Use Infrastructure As Code
  - Implement CI/CD to find problems early: CodePipeline, CodeBuild, CodeDeploy
  - Perform frequent, small reversible changes
- Recommended Approach:
  - Prepare for failure: Game days, Disaster recovery exercises
    - Implement standards with AWS Config rules
  - Operate: Gather Data and Metrics
    - CloudWatch (Logs agent), Config, Config Rules, CloudTrail, VPC Flow Logs and X-Ray (tracing)
  - Evolve: Get intelligence (Ex:Use Amazon Elasticsearch to analyze your logs)


AWS Lambda


CloudFormation


Codepipeline


AWS Config


Cloudwatch

Security Pillar

- Principle of least privilege for least time
  - Use temporary credentials when possible (IAM roles, Instance profiles)
  - Enforce MFA and strong password practices
  - Rotate credentials regularly
- Security in Depth - Apply security in all layers
  - VPCs and Private Subnets (Security Groups and Network Access Control List)
  - Use hardened EC2 AMIs (golden image) - Automate patches for OS, Software etc
  - Use CloudFront with AWS Shield for DDoS mitigation
  - Use WAF with CloudFront and ALB (Protect web apps from XSS, SQL injection etc)
  - Use CloudFormation (Automate provisioning infra that adheres to security policies)
- Protect Data at Rest
  - Enable Versioning (when available)
  - Enable encryption - KMS and Cloud HSM (Rotate encryption keys)


AWS IAM


AWS Shield


AWS WAF


AWS KMS


Cloud HSM

Security Pillar - 2

- Protect Data in Transit
  - Data coming in and going out of AWS
  - By default, all AWS API use HTTPS/SSL
  - You can also choose to perform client side encryption for additional security
  - Ensure your data stays in AWS network when possible(VPC Endpoints and AWS PrivateLink)
- Detect Threats: Actively monitor for security issues
  - Monitor CloudWatch Logs
  - Use Amazon GuardDuty to detect threats and continuously monitor for malicious behavior
  - Use AWS Organization to centralize security policies for multiple AWS accounts


AWS IAM


AWS Shield


AWS WAF


AWS KMS


Cloud HSM

Reliability Pillar

- Reliability: Ability to recover from infra and app issues
  - Adapt to changing demands in load
- Best Practices
  - Automate recovery from failure
    - Health checks and Auto scaling
    - Managed services like RDS can automatically switch to standby
  - Scale horizontally (Reduces impact of single failure)
  - Maintain Redundancy
    - Multiple Direct Connect connections
    - Multiple Regions and Availability Zones
  - Prefer serverless architectures
  - Prefer loosely coupled architectures: SQS, SNS
  - Distributed System Best Practices
    - Use Amazon API Gateway for throttling requests
    - AWS SDK provides retry with exponential backoff

Loosely coupled architectures

- ELB
  - Works in tandem with AWS auto scaling
- Amazon SQS
  - Polling mechanism
- Amazon SNS
  - Publish subscribe pattern
  - Bulk notifications and Mobile push support
- Amazon Kinesis
  - Handle event streams
  - Multiple clients
  - Each client can track their stream position

Troubleshooting on AWS - Quick Review

In 28 Minutes

Option	Details	When to Use
Amazon S3 Server Access Logs	S3 data request details - request type, the resources requested, and the date and time of request	Troubleshoot bucket access issues and data requests
Amazon ELB Access Logs	Client's IP address, latencies, and server responses	Analyze traffic patterns and troubleshoot network issues
Amazon VPC Flow Logs	Monitor network traffic	Troubleshoot network connectivity and security issues

Troubleshooting on AWS - Quick Review

In28 Minutes

Option	Details	When to Use
Amazon CloudWatch	Monitor metrics from AWS resources	Monitoring
Amazon CloudWatch Logs	Store and Analyze log data from Amazon EC2 instances and on-premises servers	Debugging application issues and Monitoring
AWS Config	AWS resource inventory. History. Rules.	Inventory and History
Amazon CloudTrail	History of AWS API calls made via AWS Management Console, AWS CLI, AWS SDKs etc.	Auditing and troubleshooting. Determine who did what, when, and from where.

Performance Efficiency Pillar: Meet needs with min. resources

In 28 Minutes

- Continue being efficient as demand and technology evolves
- Best Practices:
  - Use Managed Services (Avoid Undifferentiated Heavy Lifting)
  - Go Serverless (Lower transactional costs and less operational burden)
  - Experiment (Cloud makes it easy to experiment)
  - Monitor Performance (Trigger CloudWatch alarms - Perform actions with SQS and Lambda)
- Choose the right solution:
  - Compute: EC2 instances vs Lambda vs Containers
  - Storage: Block, File, Object
  - Database: RDS vs DynamoDB vs RedShift ..
  - Caching: ElastiCache vs CloudFront vs DAX vs Read Replicas
  - Network: CloudFront, Global Accelerator, Route 53, Placement Groups, VPC endpoints, Direct Connect
  - Use product specific features: Enhanced Networking, S3 Transfer Acceleration, EBS optimized instances


AWS Lambda


API Gateway


Cloudwatch


Amazon SQS

Cost Optimization Pillar: Run systems at lowest cost

In 28 Minutes

- Best Practices
  - Match supply and demand
    - Implement Auto Scaling
    - Stop Dev/Test resources when you don't need them
    - Go Serverless
  - Track your expenditure (Use tags on resources)
    - Cost Explorer to track and analyze your spend
    - AWS Budgets to trigger alerts
- Choose Cost-Effective Solutions
  - Right-Sizing : Analyze 5 large servers vs 10 small servers
    - Use CloudWatch (monitoring) and Trusted Advisor (recommendations) to right size your resources
  - Email server vs Managed email service (charged per email)
  - On-Demand vs Reserved vs Spot instances
  - Avoid expensive software : MySQL vs Aurora vs Oracle
  - Optimize data transfer costs using AWS Direct Connect and Amazon CloudFront


AutoScaling


AWS Lambda


Trusted Advisor

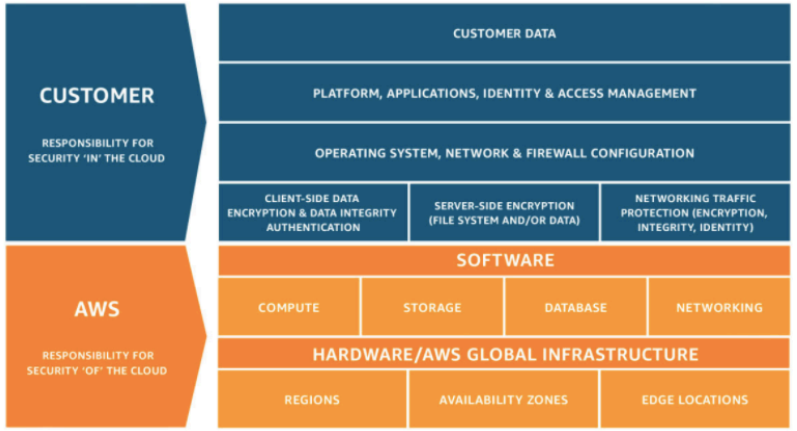

Cloudwatch


CloudFront

Shared Responsibility Model

Shared Responsibility Model

- Security & Compliance - Shared responsibility:
  - AWS manages security of the cloud:
    - Operates, manages & controls Host OS and virtualization layer down to the physical security.
  - YOU are responsible for security in the cloud:
    - Guest OS (patches), Application S/W, Security Groups, Integrating AWS Services with IT environments
- Examples:
  - EC2 - Infrastructure as a Service (IaaS)
    - AWS is responsible for infrastructure layer
    - Your Responsibilities: Guest OS (incl. patches), Application software, Security Groups (or firewalls) etc
  - Amazon S3 - Managed service (PaaS)
    - AWS manages infrastructure, OS, and platform
    - Your Responsibilities: Manage data, data security at rest (encryption) and transit (https, private network), Managing access to data and service (IAM, S3 features)



https://aws.amazon.com/compliance/shared-responsibility-model/

Shared Responsibility Model - Remember

In28 Minutes

- **Compliance responsibilities will be shared**
  - AWS ensures adherence of IT Infrastructure with IT security standards
    - SOC 1, SOC 2, SOC 3, FISMA, DIACAP, FedRAMP, PCI DSS Level 1, ISO 9001, ISO 27001, ISO 27017, ISO 27018 etc
  - AWS provides information on its IT control environment (white papers, certifications etc)
  - Customers perform their evaluation (use AWS control and compliance documentation)
- **IT controls are of three types:**
  - Inherited Controls (Customer fully inherits from AWS) : Physical and Environmental controls
  - Shared Controls (Controls shared by AWS and Customer)
    - Patch Management: AWS (Infrastructure Patches), Customer (Guest OS Patches and Software Patches)
    - Configuration Management: AWS (Infrastructure), Customer (Guest OS, databases, and applications)
    - Awareness & Training
  - Customer Owned Controls
    - Controls based on the applications deployed to AWS
    - Data Security Requirements

Important Things to Remember

Three ways to use AWS

In28 Minutes

- AWS Management Console
  - Mobile App
- AWS CLI (Command Line Interface)
  - Execute Commands
  - Create Scripts
  - Use IAM Users Credentials - access key ID and secret access key
- AWS SDKs (Software Development Kits)
  - Write Code (Java, JavaScript, Python, Go etc) using AWS APIs
  - Integrate into Existing Applications

AWS Support Plans - 1

In 28 Minutes

Feature	Basic	Developer	Business	Enterprise
AWS Trusted Advisor	7 core checks	7 core checks	All checks	All checks
Account Assistance				Concierge Support Team
Technical Account Management				Designated Technical Account Manager (TAM) to pro-actively monitor your environment and assist with optimization and coordinate access to programs and AWS experts
Cost		Starts from $29	Starts from $100	Starts from $15,000

AWS Support Plans - 2

In 28 Minutes

Feature	Basic	Developer	Business	Enterprise
Technical Support	24x7 access to customer service and forums	Business hours email access to Cloud Support Associates Unlimited cases / **1 primary contact**	24x7 phone, email, and chat access to Cloud Support Engineers Unlimited cases / unlimited contacts (IAM supported)	24x7 phone, email, and chat access to Cloud Support Engineers Unlimited cases / unlimited contacts (IAM supported)
Response Times	NA	General guidance - 24 hours System impaired: < 12 hours	General guidance: < 24 hours System impaired: < 12 hours Production impaired: < 4 hours Production down: < 1 hour	General guidance: < 24 hours System impaired: < 12 hours Production impaired: < 4 hours Production down: < 1 hour Business-critical system down: < 15 minutes

Get Ready

Certification Resources

In 28 Minutes

Title	Link
Certification - Home Page	*https://aws.amazon.com/certification/certified-cloud-practitioner/*
Overview of Amazon Web Services	*https://d0.awsstatic.com/whitepapers/aws-overview.pdf*
How AWS Pricing Works	*https://d0.awsstatic.com/whitepapers/aws_pricing_overview.pdf*
AWS Well-Architected Framework	*https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf*
The Total Cost of (Non) Ownership of Web Applications in the Cloud	*https://media.amazonwebservices.com/AWS_TCO_Web_Applications.pdf*
Compare AWS Support Plans	*https://aws.amazon.com/premiumsupport/plans/*

Certification Exam

In28 Minutes

- Multiple Choice Questions
  - Type 1 : Single Answer - 4 options and 1 right answer
  - Type 2 : Multiple Answer - 5 (or more) options and 2 (or more) right answers
- No penalty for wrong answers
  - Feel free to guess if you do not know the answer
- 65 questions and 90 minutes
- Result immediately shown after exam completion
- Email with detailed scores (a couple of days later)

Certification Exam - My Recommendations

In 28 Minutes

- Read the entire question
  - Identify the key parts of the question
- Read all answers at least once
- If you do NOT know the answer, eliminate wrong answers first
- Mark questions for future consideration and review them before final submission

You are all set!

Let's clap for you!

- You have put your best foot forward to be an AWS Certified Cloud Practitioner
- Make sure you prepare well and
- Good Luck!

Do Not Forget!

In 28 Minutes

- Recommend the course to your friends!
  - Do not forget to review!
- Your Success = My Success
  - Share your success story with me on LinkedIn (Ranga Karanam)
  - Share your success story and lessons learnt in Q&A with other learners!

What Next?

FASTEST ROADMAPS

in28minutes.com

In28 Minutes

- Google Cloud Certifications
- Azure Certifications
- AWS Certifications
- DevOps
- Java Full Stack
- Java Microservices