# IAM Section – Summary

- **Users:** mapped to a physical user, has a password for AWS Console
- **Groups:** contains users only
- **Policies:** JSON document that outlines permissions for users or groups
- **Roles:** for EC2 instances or AWS services
- **Security:** MFA + Password Policy
- **AWS CLI:** manage your AWS services using the command-line
- **AWS SDK:** manage your AWS services using a programming language
- **Access Keys:** access AWS using the CLI or SDK
- **Audit:** IAM Credential Reports & IAM Access Advisor

# Classic Ports to know

- 22 = SSH (Secure Shell) - log into a Linux instance
- 21 = FTP (File Transfer Protocol) – upload files into a file share
- 22 = SFTP (Secure File Transfer Protocol) – upload files using SSH
- 80 = HTTP – access unsecured websites
- 443 = HTTPS – access secured websites
- 3389 = RDP (Remote Desktop Protocol) – log into a Windows instance

# EC2 Instances Purchasing Options

- **On-Demand Instances** – short workload, predictable pricing, pay by second
- **Reserved** (1 & 3 years)
  - **Reserved Instances** – long workloads
  - **Convertible Reserved Instances** – long workloads with flexible instances
- **Savings Plans** (1 & 3 years) –commitment to an amount of usage, long workload
- **Spot Instances** – short workloads, cheap, can lose instances (less reliable)
- **Dedicated Hosts** – book an entire physical server, control instance placement
- **Dedicated Instances** – no other customers will share your hardware
- **Capacity Reservations** – reserve capacity in a specific AZ for any duration

# EC2 Section – Summary

- **EC2 Instance:** AMI (OS) + Instance Size (CPU + RAM) + Storage + security groups + EC2 User Data
- **Security Groups:** Firewall attached to the EC2 instance
- **EC2 User Data:** Script launched at the first start of an instance
- **SSH:** start a terminal into our EC2 Instances (port 22)
- **EC2 Instance Role:** link to IAM roles
- **Purchasing Options:** On-Demand, Spot, Reserved (Standard + Convertible), Dedicated Host, Dedicated Instance

# EC2 Instance Storage - Summary

- EBS volumes:
  - network drives attached to one EC2 instance at a time
  - Mapped to an Availability Zones
  - Can use EBS Snapshots for backups / transferring EBS volumes across AZ
- AMI: create ready-to-use EC2 instances with our customizations
- EC2 Image Builder: automatically build, test and distribute AMIs
- EC2 Instance Store:
  - High performance hardware disk attached to our EC2 instance
  - Lost if our instance is stopped / terminated
- EFS: network file system, can be attached to 100s of instances in a region
- EFS-IA: cost-optimized storage class for infrequent accessed files
- FSx for Windows: Network File System for Windows servers
- FSx for Lustre: High Performance Computing Linux file system

# Scalability vs Elasticity (vs Agility)

- Scalability: ability to accommodate a larger load by making the hardware stronger (scale up), or by adding nodes (scale out)

- Elasticity: once a system is scalable, elasticity means that there will be some "auto-scaling" so that the system can scale based on the load. This is "cloud-friendly": pay-per-use, match demand, optimize costs

- Agility: (not related to scalability - distractor) new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes.

# ELB & ASG – Summary

- **High Availability** vs **Scalability** (vertical and horizontal) vs **Elasticity** vs **Agility** in the Cloud
- **Elastic Load Balancers (ELB)**
  - Distribute traffic across backend EC2 instances, can be Multi-AZ
  - Supports health checks
  - 4 types: Classic (old), Application (HTTP – L7), Network (TCP – L4), Gateway (L3)
- **Auto Scaling Groups (ASG)**
  - Implement Elasticity for your application, across multiple AZ
  - Scale EC2 instances based on the demand on your system, replace unhealthy
  - Integrated with the ELB

# S3 Storage Classes Comparison

| | Standard | Intelligent-Tiering | Standard-IA | One Zone-IA | Glacier Instant Retrieval | Glacier Flexible Retrieval | Glacier Deep Archive |
|---|---|---|---|---|---|---|---|
| Durability | 99.999999999% == (11 9's) | | | | | | |
| Availability | 99.99% | 99.9% | 99.9% | 99.5% | 99.9% | 99.99% | 99.99% |
| Availability SLA | 99.9% | 99% | 99% | 99% | 99% | 99.9% | 99.9% |
| Availability Zones | >= 3 | >= 3 | >= 3 | 1 | >= 3 | >= 3 | >= 3 |
| Min. Storage Duration Charge | None | None | 30 Days | 30 Days | 90 Days | 90 Days | 180 Days |
| Min. Billable Object Size | None | None | 128 KB | 128 KB | 128 KB | 40 KB | 40 KB |
| Retrieval Fee | None | None | Per GB retrieved | Per GB retrieved | Per GB retrieved | Per GB retrieved | Per GB retrieved |

# AWS Snowball

- Highly-secure, portable devices to **collect and process data at the edge**, and **migrate data into and out of AWS**
- Helps migrate up to Petabytes of data

| Device | Compute | Memory | Storage (SSD) |
|---|---|---|---|
| Snowball Edge **Storage** Optimized | 104 vCPUs | 416 GB | **210 TB** |
| Snowball Edge **Compute** Optimized | 104 vCPUs | 416 GB | **28 TB** |

**Snowball Edge**

# AWS Storage Cloud Native Options


# Amazon S3 – Summary

- **Buckets vs Objects:** global unique name, tied to a region
- **S3 security:** IAM policy, S3 Bucket Policy (public access), S3 Encryption
- **S3 Websites:** host a static website on Amazon S3
- **S3 Versioning:** multiple versions for files, prevent accidental deletes
- **S3 Replication:** same-region or cross-region, must enable versioning
- **S3 Storage Classes:** Standard, IA, 1Z-IA, Intelligent, Glacier (Instant, Flexible, Deep)
- **Snowball:** import data onto S3 through a physical device, edge computing
- **Storage Gateway:** hybrid solution to extend on-premises storage to S3

# Redshift Overview

- Redshift is based on PostgreSQL, but **it's not used for OLTP**
- **It's OLAP – online analytical processing (analytics and data warehousing)**
- Load data once every hour, not every second
- 10x better performance than other data warehouses, scale to PBs of data
- **Columnar** storage of data (instead of row based)
- Massively Parallel Query Execution (MPP), highly available
- Pay as you go based on the instances provisioned
- Has a SQL interface for performing the queries
- BI tools such as AWS Quicksight or Tableau integrate with it

# Databases & Analytics Summary in AWS

- **Relational Databases - OLTP:** RDS & Aurora (SQL)
- **Differences between Multi-AZ, Read Replicas, Multi-Region**
- **In-memory Database:** ElastiCache
- **Key/Value Database:** DynamoDB (serverless) & DAX (cache for DynamoDB)
- **Warehouse - OLAP:** Redshift (SQL)
- **Hadoop Cluster:** EMR
- **Athena:** query data on Amazon S3 (serverless & SQL)
- **QuickSight:** dashboards on your data (serverless)
- **DocumentDB:** "Aurora for MongoDB" (JSON – NoSQL database)
- **Amazon Managed Blockchain:** managed Hyperledger Fabric & Ethereum blockchains
- **Glue:** Managed ETL (Extract Transform Load) and Data Catalog service
- **Database Migration:** DMS
- **Neptune:** graph database
- **Timestream:** time-series database

## Other Compute - Summary

- **Docker:** container technology to run applications
- **ECS:** run Docker containers on EC2 instances
- **Fargate:**
  - Run Docker containers without provisioning the infrastructure
  - Serverless offering (no EC2 instances)
- **ECR:** Private Docker Images Repository
- **Batch:** run batch jobs on AWS across managed EC2 instances
- **Lightsail:** predictable & low pricing for simple application & DB stacks

## Lambda Summary

- Lambda is Serverless, Function as a Service, seamless scaling, reactive
- **Lambda Billing:**
  - By the time run x by the RAM provisioned
  - By the number of invocations
- **Language Support:** many programming languages except (arbitrary) Docker
- **Invocation time:** up to 15 minutes
- **Use cases:**
  - Create Thumbnails for images uploaded onto S3
  - Run a Serverless cron job
- **API Gateway**: expose Lambda functions as HTTP API

# Deployment - Summary

- **CloudFormation:** (AWS only)
  - Infrastructure as Code, works with almost all of AWS resources
  - Repeat across Regions & Accounts
- **Beanstalk:** (AWS only)
  - Platform as a Service (PaaS), limited to certain programming languages or Docker
  - Deploy code consistently with a known architecture: ex, ALB + EC2 + RDS
- **CodeDeploy** (hybrid): deploy & upgrade any application onto servers
- **Systems Manager** (hybrid): patch, configure and run commands at scale

# Developer Services - Summary

- **CodeCommit**: Store code in private git repository (version controlled)
- **CodeBuild**: Build & test code in AWS
- **CodeDeploy**: Deploy code onto servers
- **CodePipeline**: Orchestration of pipeline (from code to build to deploy)
- **CodeArtifact**: Store software packages / dependencies on AWS
- **AWS CDK:** Define your cloud infrastructure using a programming language

# Global Applications in AWS

- **Global DNS: Route 53**
  - Great to route users to the closest deployment with least latency
  - Great for disaster recovery strategies

- **Global Content Delivery Network (CDN): CloudFront**
  - Replicate part of your application to AWS Edge Locations – decrease latency
  - Cache common requests – improved user experience and decreased latency

- **S3 Transfer Acceleration**
  - Accelerate global uploads & downloads into Amazon S3

- **AWS Global Accelerator:**
  - Improve global application availability and performance using the AWS global network

# Global Applications in AWS - Summary

- AWS Outposts
  - Deploy Outposts Racks in your own Data Centers to extend AWS services

- AWS WaveLength
  - Brings AWS services to the edge of the 5G networks
  - Ultra-low latency applications

- AWS Local Zones
  - Bring AWS resources (compute, database, storage, …) closer to your users
  - Good for latency-sensitive applications

# Integration Section – Summary

- SQS:
  - Queue service in AWS
  - Multiple Producers, messages are kept up to 14 days
  - Multiple Consumers share the read and delete messages when done
  - Used to **decouple** applications in AWS
- SNS:
  - Notification service in AWS
  - Subscribers: Email, Lambda, SQS, HTTP, Mobile…
  - Multiple Subscribers, send all messages to all of them
  - No message retention
- Kinesis: real-time data streaming, persistence and analysis
- Amazon MQ: managed message broker for ActiveMQ and RabbitMQ in the cloud (MQTT, AMQP.. protocols)

# Important Metrics

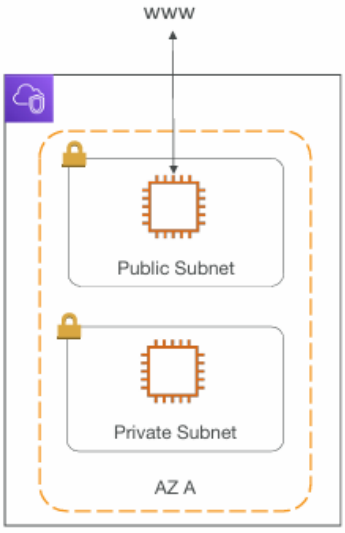
- **EC2 instances:** CPU Utilization, Status Checks, Network (not RAM)
  - Default metrics every 5 minutes
  - Option for Detailed Monitoring ($$$): metrics every 1 minute
- **EBS volumes:** Disk Read/Writes
- **S3 buckets:** BucketSizeBytes, NumberOfObjects, AllRequests
- **Billing:** Total Estimated Charge (only in us-east-1)
- **Service Limits:** how much you've been using a service API
- **Custom metrics:** push your own metrics

# Monitoring Summary

- CloudWatch:
  - **Metrics**: monitor the performance of AWS services and billing metrics
  - **Alarms**: automate notification, perform EC2 action, notify to SNS based on metric
  - **Logs**: collect log files from EC2 instances, servers, Lambda functions…
  - **Events (or EventBridge)**: react to events in AWS, or trigger a rule on a schedule
- **CloudTrail**: audit API calls made within your AWS account
- **CloudTrail Insights:** automated analysis of your CloudTrail Events
- **X-Ray:** trace requests made through your distributed applications
- **AWS Health Dashboard:** status of all AWS services across all regions
- **AWS Account Health Dashboard:** AWS events that impact your infrastructure
- **Amazon CodeGuru:** automated code reviews and application performance recommendations

# VPC & Subnets Primer

- **VPC - Virtual Private Cloud:** private network to deploy your resources (regional resource)
- **Subnets** allow you to partition your network inside your VPC (Availability Zone resource)
- A **public subnet** is a subnet that is accessible from the internet
- A **private subnet** is a subnet that is not accessible from the internet
- To define access to the internet and between subnets, we use **Route Tables**.

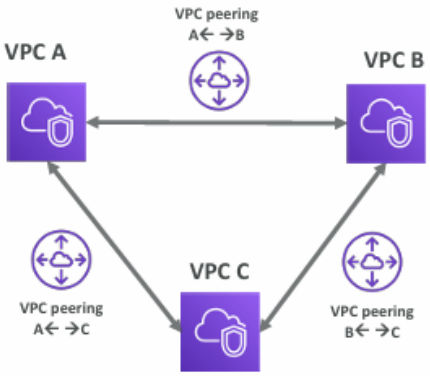



# VPC Flow Logs

- Capture information about IP traffic going into your interfaces:
  - **VPC** Flow Logs
  - **Subnet** Flow Logs
  - **Elastic Network Interface** Flow Logs
- Helps to monitor & troubleshoot connectivity issues. Example:
  - Subnets to internet
  - Subnets to subnets
  - Internet to subnets
- Captures network information from AWS managed interfaces too: Elastic Load Balancers, ElastiCache, RDS, Aurora, etc…
- VPC Flow logs data can go to S3, CloudWatch Logs, and Amazon Data Firehose

# VPC Peering

- Connect two VPC, privately using AWS' network
- Make them behave as if they were in the same network
- Must not have overlapping CIDR (IP address range)
- VPC Peering connection is **not transitive** (must be established for each VPC that need to communicate with one another)




# VPC Closing Comments

- **VPC** – Virtual Private Cloud
- **Subnets** – Tied to an AZ, network partition of the VPC
- **Internet Gateway** – at the VPC level, provide Internet Access
- **NAT Gateway / Instances** – give internet access to private subnets
- **NACL** – Stateless, subnet rules for inbound and outbound
- **Security Groups** – Stateful, operate at the EC2 instance level or ENI
- **VPC Peering** – Connect two VPC with non overlapping IP ranges, nontransitive
- **Elastic IP** –fixed public IPv4, ongoing cost if not in-use

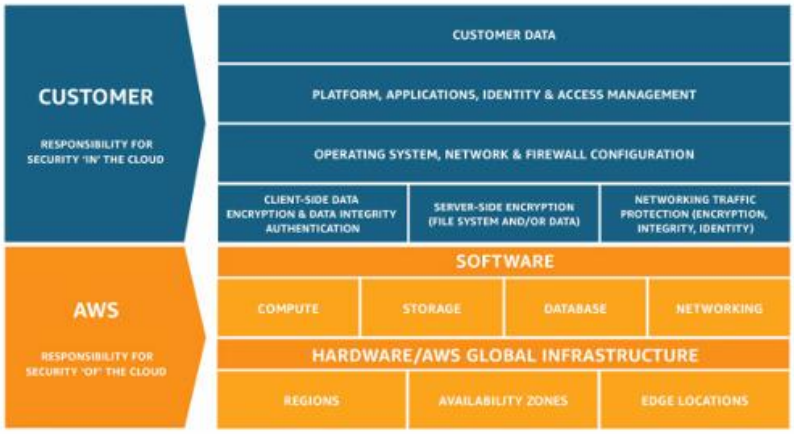
# VPC Closing Comments

- **VPC Endpoints** – Provide private access to AWS Services within VPC
- **PrivateLink** – Privately connect to a service in a 3rd party VPC
- **VPC Flow Logs** – network traffic logs
- **Site to Site VPN** – VPN over public internet between on-premises DC and AWS
- **Client VPN** – OpenVPN connection from your computer into your VPC
- **Direct Connect** – direct private connection to AWS
- **Transit Gateway** – Connect thousands of VPC and on-premises networks together

# AWS Shared Responsibility Model

- AWS responsibility - Security **of** the Cloud
  - Protecting infrastructure (hardware, software, facilities, and networking) that runs all the AWS services
  - Managed services like S3, DynamoDB, RDS, etc.
- Customer responsibility - Security **in** the Cloud
  - For EC2 instance, customer is responsible for management of the guest OS (including security patches and updates), firewall & network configuration, IAM
  - Encrypting application data
- Shared controls:
  - Patch Management, Configuration Management, Awareness & Training

# Shared Responsibility Model diagram




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

# DDOS Protection on AWS

- **AWS Shield Standard:** protects against DDOS attack for your website and applications, for all customers at no additional costs
- **AWS Shield Advanced:** 24/7 premium DDoS protection
- **AWS WAF:** Filter specific requests based on rules
- **CloudFront and Route 53:**
  - Availability protection using global edge network
  - Combined with AWS Shield, provides attack mitigation at the edge
- Be ready to scale – leverage **AWS Auto Scaling**

# AWS Security Hub

- **Central security tool** to manage security **across several AWS accounts** and **automate security checks**
- Integrated dashboards showing current security and compliance status to quickly take actions
- Automatically aggregates alerts in predefined or personal findings formats from various AWS services & AWS partner tools:
  - Config
  - GuardDuty
  - Inspector
  - Macie
  - IAM Access Analyzer
  - AWS Systems Manager
  - AWS Firewall Manager
  - AWS Health
  - AWS Partner Network Solutions
- Must first enable the AWS Config Service

# Section Summary: Security & Compliance

- **Shared Responsibility on AWS**
- **Shield:** Automatic DDoS Protection + 24/7 support for advanced
- **WAF:** Firewall to filter incoming requests based on rules
- **KMS:** Encryption keys managed by AWS
- **CloudHSM:** Hardware encryption, we manage encryption keys
- **AWS Certificate Manager:** provision, manage, and deploy SSL/TLS Certificates
- **Artifact:** Get access to compliance reports such as PCI, ISO, etc…
- **GuardDuty:** Find malicious behavior with VPC, DNS & CloudTrail Logs
- **Inspector:** find software vulnerabilities in EC2, ECR Images, and Lambda functions
- **Network Firewall:** Protect VPC against network attacks

# Section Summary: Security & Compliance

- **Config:** Track config changes and compliance against rules
- **Macie:** Find sensitive data (ex: PII data) in Amazon S3 buckets
- **CloudTrail:** Track API calls made by users within account
- **AWS Security Hub:** gather security findings from multiple AWS accounts
- **Amazon Detective:** find the root cause of security issues or suspicious activities
- **AWS Abuse:** Report AWS resources used for abusive or illegal purposes
- **Root user privileges:**
  - Change account settings
  - Close your AWS account
  - Change or cancel your AWS Support plan
  - Register as a seller in the Reserved Instance Marketplace
- **IAM Access Analyzer:** identify which resources are shared externally
- **Firewall Manager:** manage security rules across an Organization (WAF, Shield…)



# AWS Machine Learning - Summary

- **Rekognition:** face detection, labeling, celebrity recognition
- **Transcribe:** audio to text (ex: subtitles)
- **Polly:** text to audio
- **Translate:** translations
- **Lex:** build conversational bots – chatbots
- **Connect:** cloud contact center
- **Comprehend:** natural language processing
- **SageMaker:** machine learning for every developer and data scientist
- **Kendra:** ML-powered search engine
- **Personalize:** real-time personalized recommendations
- **Textract:** detect text and data in documents

# AWS Organizations

- Global service
- Allows to manage **multiple AWS accounts**
- The main account is the master account
- Cost Benefits:
  - **Consolidated Billing** across all accounts - single payment method
  - Pricing benefits from **aggregated usage** (volume discount for EC2, S3…)
  - **Pooling of Reserved EC2 instances** for optimal savings
- API is available to **automate AWS account creation**
- **Restrict account privileges using Service Control Policies (SCP)**

# Compute Pricing – EC2

- **On-demand instances:**
  - Minimum of 60s
  - Pay per second (Linux/Windows) or per hour (other)
- **Reserved instances:**
  - Up to 75% discount compared to On-demand on hourly rate
  - 1- or 3-years commitment
  - All upfront, partial upfront, no upfront
- **Spot instances:**
  - Up to 90% discount compared to On-demand on hourly rate
  - Bid for unused capacity
- **Dedicated Host:**
  - On-demand
  - Reservation for 1 year or 3 years commitment
- **Savings plans** as an alternative to save on sustained usage

# Account Best Practices – Summary

- Operate multiple accounts using **Organizations**
- Use **SCP** (service control policies) to restrict account power
- Easily setup multiple accounts with best-practices with **AWS Control Tower**
- **Use Tags & Cost Allocation Tags** for easy management & billing
- **IAM guidelines:** MFA, least-privilege, password policy, password rotation
- **Config** to record all resources configurations & compliance over time
- **CloudFormation** to deploy stacks across accounts and regions
- **Trusted Advisor** to get insights, Support Plan adapted to your needs
- Send Service Logs and Access Logs to **S3 or CloudWatch Logs**
- **CloudTrail** to record API calls made within your account
- **If your Account is compromised:** change the root password, delete and rotate all passwords / keys, contact the AWS support
- Allow users to create pre-defined stacks defined by admins using **AWS Service Catalog**



# Billing and Costing Tools – Summary

- **Compute Optimizer:** recommends resources' configurations to reduce cost
- **Pricing Calculator:** cost of services on AWS
- **Billing Dashboard:** high level overview
- **Cost Allocation Tags:** tag resources to create detailed reports
- **Cost and Usage Reports:** most comprehensive billing dataset
- **Cost Explorer:** View current usage (detailed) and forecast usage
- **Billing Alarms:** in us-east-1 – track overall and per-service billing
- **Budgets:** more advanced – track usage, costs, RI, and get alerts
- **Savings Plans:** easy way to save based on long-term usage of AWS
- **Cost Anomaly Detection:** detect unusual spends using Machine Learning
- **Service Quotas:** notify you when you're close to service quota threshold

# Advanced Identity - Summary

- IAM
  - Identity and Access Management inside your AWS account
  - For users that you trust and belong to your company
- **Organizations:** Manage multiple accounts
- **Security Token Service (STS):** temporary, limited-privileges credentials to access AWS resources
- **Cognito**: create a database of users for your mobile & web applications
- **Directory Services**: integrate Microsoft Active Directory in AWS
- **IAM Identity Center:** one login for multiple AWS accounts & applications

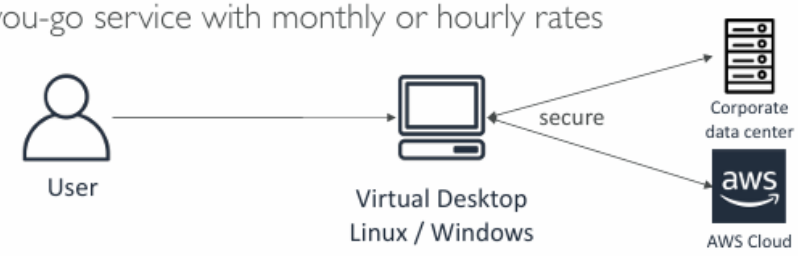
# Amazon AppStream 2.0 vs WorkSpaces

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

- **AppStream 2.0**
  - Stream a desktop application to web browsers (no need to connect to a VDI)
  - Works with any device (that has a web browser)
  - Allow to configure an instance type per application type (CPU, RAM, GPU)

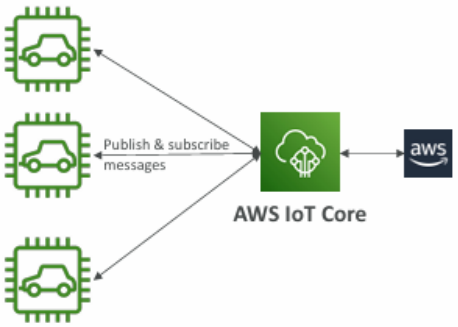
# Amazon WorkSpaces

- Managed Desktop as a Service (DaaS) solution to easily **provision Windows or Linux desktops**
- **Great to eliminate management of on-premise VDI (Virtual Desktop Infrastructure)**
- Fast and quickly scalable to thousands of users
- Secured data – integrates with KMS
- Pay-as-you-go service with monthly or hourly rates



# AWS IoT Core

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




# AWS AppSync

- Store and sync data across mobile and web apps in real-time
- Makes use of GraphQL (mobile technology from Facebook)
- Client Code can be generated automatically
- Integrations with DynamoDB / Lambda
- Real-time subscriptions
- Offline data synchronization (replaces Cognito Sync)
- Fine Grained Security
- AWS Amplify can leverage AWS AppSync in the background!

GraphQL

# 1) Operational Excellence

- Includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures
- Design Principles
  - **Perform operations as code** - Infrastructure as code
  - **Make frequent, small, reversible changes** - So that in case of any failure, you can reverse it
  - **Refine operations procedures frequently** - And ensure that team members are familiar with it
  - **Anticipate failure**
  - **Learn from all operational failures**
  - **Use managed services** - to reduce operational burden
  - **Implement observability for actionable insights** - performance, reliability, cost…

Stephane Maarek

# 3) Reliability

- Ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues
- Design Principles
  - **Test recovery procedures** - Use automation to simulate different failures or to recreate scenarios that led to failures before
  - **Automatically recover from failure** - Anticipate and remediate failures before they occur
  - **Scale horizontally to increase aggregate system availability** - Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure
  - **Stop guessing capacity** - Maintain the optimal level to satisfy demand without over or under provisioning - Use Auto Scaling
  - **Manage change in automation** - Use automation to make changes to infrastructure

# 4) Performance Efficiency

- Includes the ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve
- Design Principles
  - **Democratize advanced technologies** - Advance technologies become services and hence you can focus more on product development
  - **Go global in minutes** - Easy deployment in multiple regions
  - **Use serverless architectures** - Avoid burden of managing servers
  - **Experiment more often** - Easy to carry out comparative testing
  - **Mechanical sympathy** - Be aware of all AWS services

# 5) Cost Optimization

- Includes the ability to run systems to deliver business value at the lowest price point
- Design Principles
  - **Adopt a consumption mode** - Pay only for what you use
  - **Measure overall efficiency** - Use CloudWatch
  - **Stop spending money on data center operations** - AWS does the infrastructure part and enables customer to focus on organization projects
  - **Analyze and attribute expenditure** - Accurate identification of system usage and costs, helps measure return on investment (ROI) - Make sure to use tags
  - **Use managed and application level services to reduce cost of ownership** - As managed services operate at cloud scale, they can offer a lower cost per transaction or service

# 6) Sustainability

- The sustainability pillar focuses on minimizing the environmental impacts of running cloud workloads.
- Design Principles
  - **Understand your impact** – establish performance indicators, evaluate improvements
  - **Establish sustainability goals** – Set long-term goals for each workload, model return on investment (ROI)
  - **Maximize utilizatio**n – Right size each workload to maximize the energy efficiency of the underlying hardware and minimize idle resources.
  - **Anticipate and adopt new, more efficient hardware and software offerings** – and design for flexibility to adopt new technologies over time.
  - **Use managed services –** Shared services reduce the amount of infrastructure; Managed services help automate sustainability best practices as moving infrequent accessed data to cold storage and adjusting compute capacity.
  - **Reduce the downstream impact of your cloud workloads** – Reduce the amount of energy or resources required to use your services and reduce the need for your customers to upgrade their devices

# Well Architected Framework 6 Pillars

- 1) Operational Excellence
- 2) Security
- 3) Reliability
- 4) Performance Efficiency
- 5) Cost Optimization
- 6) Sustainability

# CAF Perspectives and Foundational Capabilities
## Business Capabilities






# CAF Perspectives and Foundational Capabilities
## Technical Capabilities

- **Platform Perspective** helps you build an enterprise-grade, scalable, hybrid cloud platform; modernize existing workloads; and implement new cloud-native solutions.

- **Security Perspective** helps you achieve the confidentiality, integrity, and availability of your data and cloud workloads.

- **Operations Perspective** helps ensure that your cloud services are delivered at a level that meets the needs of your business.

# AWS CAF – Transformation Phases

- **Envision** – demonstrate how the Cloud will accelerate business outcomes by identifying transformation opportunities and create a foundation for your digital transformation
- **Align** – identify capability gaps across the 6 AWS CAF Perspectives which results in an Action Plan
- **Launch** – build and deliver pilot initiatives in production and demonstrate incremental business value
- **Scale** – expand pilot initiatives to the desired scale while realizing the desired business benefits