# 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

# 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 + Scheduled), 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

# 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
  - 3 types: Application LB (HTTP – L7), Network LB (TCP – L4), Classic LB (old)
- 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

# 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 Access Logs:** log requests made within your S3 bucket
- **S3 Replication:** same-region or cross-region, must enable versioning
- **S3 Storage Classes:** Standard, IA, 1Z-IA, Intelligent, Glacier, Glacier Deep Archive
- **S3 Lifecycle Rules:** transition objects between classes
- **S3 Glacier Vault Lock / S3 Object Lock:** WORM (Write Once Read Many)
- **Snow Family:** import data onto S3 through a physical device, edge computing
- **OpsHub:** desktop application to manage Snow Family devices
- **Storage Gateway:** hybrid solution to extend on-premises storage to S3

# 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 QLDB:** Financial Transactions Ledger (immutable journal, cryptographically verifiable)
- **Amazon Managed Blockchain:** managed Hyperledger Fabric & Ethereum blockchains
- **Glue:** Managed ETL (Extract Transform Load) and Data Catalog service
- **Database Migration:** DMS
- **Neptune:** graph 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
- **OpsWorks** (hybrid): managed Chef and Puppet in AWS

# 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
- **CodeStar**: Unified view for allowing developers to do CICD and code
- **Cloud9**: Cloud IDE (Integrated Development Environment) with collab
- **AWS CDK:** Define your cloud infrastructure using a programming language

# Global Applications in AWS - Summary


- 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 Apache MQ in the cloud (MQTT, AMQP.. protocols)

# 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
- Service Health Dashboard: status of all AWS services across all regions
- Personal Health Dashboard: AWS events that impact your infrastructure
- Amazon CodeGuru: automated code reviews and application performance recommendations

# 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:** For EC2 only, install agent and find vulnerabilities

# 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

# 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
- **Forecast:** build highly accurate forecasts
- **Kendra:** ML-powered search engine
- **Personalize:** real-time personalized recommendations
- **Textract:** detect text and data in documents

# 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

# 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 + free tier dashboard
- **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

# 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 AWS 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
- **Single Sign-On (SSO):** one login for multiple AWS accounts & applications