AWS SYO-CO2

Color Key

Need everything = Training course, AWS DOCs, etc.

Need to read AWS Docs and do labs

Need QwikLab

Need AWS Docs

Complete

Study Guides

<https://tutorialsdojo.com/aws-certified-sysops-administrator-associate/>

Sample Exams

<https://www.examtopics.com/exams/amazon/aws-certified-sysops-administrator-associate/>

<https://portal.tutorialsdojo.com>

# Domain 1: Monitoring, Logging, and Remediation

## 1.1 Implement metrics, alarms, and filters by using AWS monitoring and logging services

### Identify, collect, analyze, and export logs (for example, Amazon CloudWatch Logs, CloudWatch Logs Insights, AWS CloudTrail logs)

* + See Appendix CloudWatch

### Collect metrics and logs using the CloudWatch agent

* + See Appendix

### Create CloudWatch alarms

* + See Appendix CloudWatch

### Create metric filters

* + See Appendix CloudWatch

### Create CloudWatch dashboards

* + See Appendix CloudWatch

### Configure notifications (for example, Amazon Simple Notification Service [Amazon SNS], Service Quotas, CloudWatch alarms, AWS Health events)

* + See Appendix CloudWatch

## 1.2 Remediate issues based on monitoring and availability metrics

### Troubleshoot or take corrective actions based on notifications and alarms

### Configure Amazon EventBridge rules to trigger actions

### Use AWS Systems Manager Automation documents to take action based on AWS Config rules

* + See Appendix AWS Config

# Domain 2: Reliability and Business Continuity

## 2.1 Implement scalability and elasticity

### Create and maintain AWS Auto Scaling plans

* + See Appendix EC2 AutoScaling

### Implement caching

* + See Appendix ElastiCache

### Implement Amazon RDS replicas and Amazon Aurora Replicas

* + RDS Replicas
    - See Appendix
  + Aurora Replicas
    - See Appendix

### Implement loosely coupled architectures

* + See Appendix SQS

### Differentiate between horizontal scaling and vertical scaling

* + Horizonal
    - Scale out
    - Add more instances
  + Vertical
    - Scale up
    - Increase size of instance

## 2.2 Implement high availability and resilient environments

### Configure Elastic Load Balancer and Amazon Route 53 health checks

* + See Appendix ELB
  + See Appendix Route 53

### Differentiate between the use of a single Availability Zone and Multi-AZ deployments (for example, Amazon EC2 Auto Scaling groups, Elastic Load Balancing, Amazon FSx, Amazon RDS)

* + EC2 Auto Scaling
    - See Appendix EC2 AutoScaling
  + Elastic Load Balancing
    - See Appendix ELB
  + Amazon FSx
    - See Appendix FSx
  + Amazon RDS
    - See Appendix RDS

### Implement fault-tolerant workloads (for example, Amazon Elastic File System [Amazon EFS], Elastic IP addresses)

* + EFS
    - See Appendix EFS
  + Elastic IP
    - See Appendix VPC

### Implement Route 53 routing policies (for example, failover, weighted, latency based)

* + See Appendix Route 53

## 2.3 Implement backup and restore strategies

### Automate snapshots and backups based on use cases (for example, RDS snapshots, AWS Backup, RTO and RPO, Amazon Data Lifecycle Manager, retention policy)

* + RDS Snapshots
    - See Appendix RDS

### Restore databases (for example, point-in-time restore, promote read replica)

* + RDS Backup (Point in time Restore
    - See Appendix RDS
  + RDS Failover (Promot Read Replica)
    - See Appendix RDS

### Implement versioning and lifecycle rules

* + See Appendix S3

### Configure Amazon S3 Cross-Region Replication

* + See Appendix S3

### Execute disaster recovery procedures

* + DataSync
    - Move large amount of data to AWS
    - Can sync to
      * S3
      * EFS
      * FSx
    - Move data from on prem NAS or file system via NFS or SMB
    - Scheduled hourly, daily , weekly
    - Datasync agent needs to be installed
    - Setup bandwidth limit to prevent data sync taking over network bandwidth
    - Can sync AWS EFS to AWS EFS
  + Backup
    - Fully managed backup service
    - Centrally managed with auto backups across all AWS Services
      * Fsx
      * EFS
      * DynamoDB  
        EC2
      * EBS
      * RDS
      * Aurora
      * Storage gateway (volume gateway)
    - Supports Cross region and cross account backups
    - Point in Time Recovery
    - On-Demand and Scheduled backups
    - Tag based backup policies
    - Backup plan = Backup policies
      * Frequency
      * Backup window
      * Transition to cold storage (life cycle)
      * Retention Period

# Domain 3: Deployment, Provisioning, and Automation

## 3.1 Provision and maintain cloud resources

### Create and manage AMIs (for example, EC2 Image Builder)

* + Overview
    - Amazon Machine Image
    - Customization of and EC2 Instance
      * You add your own software, config etc
      * Gold image
    - AMI are built for specific region
      * Can be copied across regions
    - You can launch EC2 from
      * Public AMI
        + AWS provide
      * Your own AMI
        + AMI you create
      * AWS Marketplace AMI
        + Ami created by third party
        + May need to purchase
  + No Reboot option
    - Enable you to create AMI without shutting down AMI Instance
    - By default not selected, instance will reboot when creating AMI
      * OS Buffers are not flushed in this mode
    - AWS Backup Plans to create AMI
      * Must use no-reboot option
  + EC2 Instance Migration between AZ
    - Uses AMI
    - Create AMI
    - Restore AMI in new instance in diff AZ
  + Cross account AMI sharing
    - You can share an AMI with another AWS account
    - Sharing does not affect ownership of AMI
    - Share if
      * AMI unencrypted or encrypted with customer managed key
    - If sharing encrypted volumes you must share customer managed key used to encrypt them
    - Cross Account AMI Copy
      * If you copy an AMI that has been shared , you become the owner of the new copy
      * The owner of the source AMI must grant read permissions for the storage that backs the AMI
      * If encrypted owner must share the keys
  + EC2 Image Builder
    - Automate creation of VM or container images
      * Automate creation, maintain, validate and test EC2 AMIs
    - Built AMI can be distributed to different regions
    - Can be run on a schedule
    - Free service
  + Can force users to only launch EC2 instances from pre-approved AMIs (AMIs tagged with specific Tags) using IAM Policies
  + Use AWS Config to find non compliant EC2 instances (instances started with non approved AMI)

### Create, manage, and troubleshoot AWS CloudFormation

* + See Appendix
  + Overview
    - Infrastructure as Code
    - Allows reproducing deployments
    - Declarative way
    - Can be version controlled
    - Templates
      * Uploaded to S3 and referenced in CloudFormation
      * Can’t edit previous template
  + Create
    - Resources created in correct order
    - Parameters defined in template
    - YAML and JSON
      * JSON sucks for CLoudformations
      * YAML Key value pairs
      * Nested objects
      * Arrays (-)
      * Multi line strings (|)
      * Comments
    - Resources
      * Mandatory
      * Diff AWS components
      * Can reference each other
      * AWS::aws-product-name::data-type-name
      * AWS Documentation has the syntax for the YAML
        + [AWS resource and property types reference - AWS CloudFormation (amazon.com)](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html)
      * Type
      * Properties
    - Parameters
      * Provide input to template
      * Reuse templates across company or regions / az
      * Controlled and can prevent errors thanks to types
      * When to use parameters
        + Is the resource config likely to change in the future
      * Settings
        + Type
        + Description
        + Contrainst
        + Etc
      * Reference a parameter
        + !Ref paramname
        + Can also reference other resources in template
        + Pseudo Parameters

Enabled by default

Built in parameters that reference AWS data

AWS::AccountID

AWS::NotificationARNs

AWS::NoValue

AWS::Region

AWS::StackId

AWS::StackName

* + - Mappings
      * Fixed variables within template
      * Hardcoded
      * When to use:
        + If you know in advance all the values that could be used
        + Safer control over template
      * Use mapping
        + Fn::FindInMap

!FindInMap [MapName, TopLevelKey, SecondLevelKey]

* + - Output
      * Optional
      * If exported then they can be imported into other stacks
      * View in console or cli
      * Cross stack collaboration allowing others to create stack
      * Cannot delete template that has output ref by another stack
      * To import
        + Fn::ImportValue function

!ImportValue

* + - Conditions
      * Used to control the creation of resources or outputs
      * Syntax:
        + LogicID: Function [ values ]
        + LogicID is created by you
        + Function

Fn::And

Fn::Equals

Fn::If

Fn::Not

Fn::Or

* + - Intrinsic Functions
      * Fn::Ref
        + Reference parameter or resource within the template
      * Fm::GetAtt
        + Get attribute of a resource
      * Fn::FindInMap
        + See mappings
      * Fn::ImportValue
        + See Output
      * Fn::join
        + Join values with a delimeter
        + Syntax

!Join [ delimeter, [ comman-delimited list of values ] ]

* + - * Fn::sub
        + Substitute
      * Conditionfunctions
        + See Conditions
  + Manage
    - Update
      * Can’t edit existing cloudformation template
      * Must upload / create new template
      * Change set preview
        + Shows what will change between new stack and already deployed stack
    - Delete
      * Delete stack and all resources deleted in correct order
      * All resources created by cloudformations deleted

### Provision resources across multiple AWS Regions and accounts (for example, AWS Resource Access Manager, CloudFormation StackSets, IAM cross-account roles)

* + AWS Resource Access Manager
    - Enables you to share specified AWS Resources that you own with other AWS accounts.
  + CloudFormation StackSets
    - Create update or delete stacks across multipe accounts and regions with single operation
    - Administrator account to create stacksets
    - Trusted accounts to create update delete stack instances from stacksets
    - When you update the stack set, all associated stack instances are updated throughout all accounts and regions
    - Ability to set max concurrent actions on targets (# or %)
    - Ability to set failure tolerance (# or %)
  + IAM Cross-Account Roles

### Select deployment scenarios and services (for example, blue/green, rolling, canary)

* + <https://docs.aws.amazon.com/codedeploy/latest/userguide/welcome.html#blue-green-lambda-compute-type>
  + CodeDeploy
    - Automates app deployment to
      * EC2 instances
      * On-Prem instances
      * Lambda functions
      * ECS Services
  + CodeDeploy deployment types
    - In-Place Deployment
      * Each instance is stopped, latest app installed and the new version is then started
      * Load balancer helps deregister so other nodes stay online
      * Only works for EC2 and on-prem
    - Blue/Green Deployment
      * Can be used to decouple app DB from deployment
      * Depends on computer platform
        + EC2 / On-Prem

Deployment group ( original ) replaced by different instances ( replacement environment )

New isntances provisioned

Updated app installed

New instances registered with ELB

* + - * + Lambda or ECS

Traffic shifted in increments according to

<https://docs.aws.amazon.com/codedeploy/latest/userguide/primary-components.html#primary-components-deployment-configuration>

Canary,

Traffic shifted in two increments

Percentage in each shift configured

Linear

Traffic is shifted in equal increments with an equal number of minutes between each

all-at-once

all traffic is shifted from the original to the new

* + - * + Cloudformation

Traffic shifted from current resources to updated resources as part of stack update.

Only ECS deployments supported

### Identify and remediate deployment issues (for example, service quotas, subnet sizing, CloudFormation and AWS OpsWorks errors, permissions)

See Appendix VPC and OpsWorks

* + Service Quotas
    - See Appendix Cloudwatch Service Quotas
  + Subnet sizing
  + Cloudformation
  + Aws opsworks errors
  + permissions

## 3.2 Automate manual or repeatable processes

### Use AWS services (for example, OpsWorks, Systems Manager, CloudFormation) to automate deployment processes

* + OpsWork
    - See Appendix
  + Systems Manager
    - See Appendix
  + CloudFormation
    - See Appendix

### Implement automated patch management

* + See Appendix

### Schedule automated tasks by using AWS services (for example, EventBridge, AWS Config)

* + EventBridge
    - See Appendix EventBridge
  + AWS Config
    - See Appendix AWS Config

# Domain 4: Security and Compliance

## 4.1 Implement and manage security and compliance policies

### Implement IAM features (for example, password policies, MFA, roles, SAML, federated identity,resource policies, policy conditions)

* + See Appendix IAM

### Troubleshoot and audit access issues by using AWS services (for example, CloudTrail, IAM Access Analyzer, IAM policy simulator)

* + IAM Policy Simulator
    - Test policies

### Validate service control policies and permissions boundaries

### Review AWS Trusted Advisor security checks

* + See Appendix Trusted Advisor

### Validate AWS Region and service selections based on compliance requirements

### Implement secure multi-account strategies (for example, AWS Control Tower, AWS,Organizations)

* + AWS Control Tower
  + AWS Organizations
    - See Appendix AWS Organizations
* AWS Artifacts
  + Not really a service
  + Portal to on-demand compliance documentation and AWS agreements
  + Artifact Reports
    - Download AWS Security and compliance documents from third party auditors
      * PCI
      * SOC
  + Artifact agreements
    - Status of agreements
      * BAA – Business Associate Addendum
      * HIPAA
    - Can be used to support internal audit compliance

## 4.2 Implement data and infrastructure protection strategies

### Enforce a data classification scheme

### Create, manage, and protect encryption keys

### Implement encryption at rest (for example, AWS Key Management Service [AWS KMS])

* + See Appendix AWS KMS

### Implement encryption in transit (for example, AWS Certificate Manager, VPN)

* + See Appendix AWS Cert Manager
  + See Appendix VPN

### Securely store secrets by using AWS services (for example, AWS Secrets Manager, Systems Manager Parameter Store)

* + AWS Secrets Manager
    - See Appendix Secrets Manager
  + Systems Manager parameter Store
    - See Appendix Systems Manager

### Review reports or findings (for example, AWS Security Hub, Amazon GuardDuty, AWS Config, Amazon Inspector)

* + Security Hub
  + Amazon GuardDuty
    - See Appendix Amazon GuardDuty
  + AWS Config
    - See Appendix AWS Config
  + Inspector
    - See Appendix
  + Penetration Testing
    - Pen Test without prior approval for 8 services
      * EC2 instances,NAT Gateways,ELB
      * RDS
      * CloudFront
      * Aurora
      * API Gateway
      * Lambda
      * Lightsail
      * Elastic BeanStalk
    - Prohibitive activities ( anything that looks like an attack on their infrastructure)
      * DNS Zone walking via Route 53 hosted zones
      * DOS,DDOS, SimDOS, SIMDDOS
      * Port flooding
      * Protocol flooding
      * Request flooding
    - Anything else talk with AWS Security

# Domain 5: Networking and Content Delivery

## 5.1 Implement networking features and connectivity

### Configure a VPC (for example, subnets, route tables, network ACLs, security groups, NAT gateway, internet gateway)

* + See Appendix

### Configure private connectivity (for example, Systems Manager Session Manager, VPC endpoints, VPC peering, VPN)

* + See Appendix

### Configure AWS network protection services (for example, AWS WAF, AWS Shield)

* + AWS WAF
    - See Appendix WAF
  + AWS Shield
    - See Appendix Shield

## 5.2 Configure domains, DNS services, and content delivery

### Configure Route 53 hosted zones and records

* + See Appendix Route 53

### Implement Route 53 routing policies (for example, geolocation, geoproximity)

* + See Appendix Route 53

### Configure DNS (for example, Route 53 Resolver)

* + See Appendix Route 53

### Configure Amazon CloudFront and S3 origin access identity (OAI)

* + See Appendix CloudFront

### Configure S3 static website hosting

* + Can host static websites
    - Must be enabled and point to the default doc ( usually index )
  + URL
    - <bucket-name>.s3-website-<AWS-region>.amazonaws.com
  + Requires bucket policy to allow public
    - Or you will get a 403 error

## 5.3 Troubleshoot network connectivity issues

### Interpret VPC configurations (for example, subnets, route tables, network ACLs, security groups)

* + See Appendix

### Collect and interpret logs (for example, VPC Flow Logs, Elastic Load Balancer access logs, AWS WAF web ACL logs, CloudFront logs)

* + See VPC Flow Log Appendix
  + See Appendix ELB
  + See Appendix CloudFront

### Identify and remediate CloudFront caching issues

* + See Appendix CloudFront

### Troubleshoot hybrid and private connectivity issues

* + VPN
  + VPN Gateway
  + DirectConnect

# Domain 6: Cost and Performance Optimization

## 6.1 Implement cost optimization strategies

### Implement cost allocation tags

* + Used to track cost on detailed level
  + AWS Generated tags
    - Auto applied to resource you create
    - Starts with prefix AWS
  + User defined tags
    - Defined by user
    - Starts with prefix User

### Identify and remediate underutilized or unused resources by using AWS services and tools (for example, Trusted Advisor, AWS Compute Optimizer, Cost Explorer)

* + Trusted Advisor
    - See Appendix Trusted Advisor
  + AWS Computer Optimizer
    - See Appendix
  + Cost Explorer
    - See Appendix

### Configure AWS Budgets and billing alarms

* + Budgets
    - <https://tutorialsdojo.com/aws-billing-and-cost-management/>
    - Billing metrics stored in one region Cloudwatch - us east 1
      * But data is for all regions in account
    - Can alert when costs or usage exceed or are forecasted to exceed your budget
    - Can view
      * How close your plan is to your budgeted or to free tier limits
      * Usage to date
        + How much you have used of reserved instances and purchased savings plans
      * Your current estimated charges from AWS and how much your predicted usage will incur in charges by the end of the month
      * How much of your budget has been used
    - Updated three times per day
    - Types of budgets
      * Cost Budget
        + Plan how much you want to spend on a service
      * Usage Budget
        + Plan how much you want to use one or more services
      * RI Utilization Budgets
        + Define a utilized threshold and receive alerts when your RI usage falls below that threshold
      * RI Coverage Budgets
        + Coverage threshold and receive alerts when the number of your instance hours that are voced by RIs fall below that threshold

### Assess resource usage patterns to qualify workloads for EC2 Spot Instances

* + See Appendix EC2

### Identify opportunities to use managed services (for example, Amazon RDS, AWS Fargate, EFS)

* + Fargate
    - Serverless EKS and ECS

## 6.2 Implement performance optimization strategies

### Recommend compute resources based on performance metrics

* + See Appendix Computer Optimiz

### Monitor Amazon EBS metrics and modify configuration to increase performance efficiency

* + See Appendix EBS

### Implement S3 performance features (for example, S3 Transfer Acceleration, multipart uploads)

* + See Appendix S3

### Monitor RDS metrics and modify the configuration to increase performance efficiency (for example, Performance Insights, RDS Proxy)

* + See Appendix RDS

### Enable enhanced EC2 capabilities (for example, enhanced network adapter, instance store, placement groups)

* + See Appendix

## Appendix

Which key tools, technologies, and concepts might be covered on the exam?

The following is a non-exhaustive list of the tools and technologies that could appear on the exam. This list

is subject to change and is provided to help you understand the general scope of services, features, or

technologies on the exam. The general tools and technologies in this list appear in no particular order.

AWS services are grouped according to their primary functions. While some of these technologies will likely

be covered more than others on the exam, the order and placement of them in this list is no indication of

relative weight or importance:

AWS services and features

### Analytics:

* Amazon Elasticsearch Service (Amazon ES) now called OpenSearch
  + <https://tutorialsdojo.com/amazon-elasticsearch-amazon-es/>
  + Managed version of ElasticSearch (Open source)
  + Runs on server
  + Use cases
    - Log analytics
    - Real time application monitoring
    - Security analytics
    - Full Text Search
    - Clickstream analytics
    - Indexing
  + Access policy
    - IP Based policy
      * Resource-based policies used to restric access to ES domain to IP addresses or CIDR blocks
      * Allows unsigned requests to ES Domain
  + Kibana Authentication
    - Does not natively support IAM users and roles
    - Can control
      * HTTP Basic Authentication
        + Internal User database stored in ElasticSearch Index
      * SAML
        + Use existing 3rd party id provider to log into Kibana
      * Cognito
        + Easy AD integration
  + Production Setup
    - Use 3 dedicated master nodes
      * Spread across AZ (at least 3) for HA
      * One active 2 backup
    - Use at least 2 data nodes per AZ for replication
    - Create one replica for each index in the cluster
* Application Integration:
* Amazon EventBridge (Amazon CloudWatch Events)
  + Overview
    - Realtime stream of system events
    - Intercept events from AWS services
    - Intercept API call with cloud trail integration
    - Schedule event
    - Json payload for event passed to target
    - Can connect to on prem apps and third party SaaS apps
  + EventBridge Bus
    - Default bus = Cloudwatch Events (AWS services)
    - Partner Event BUS = events from SaaS services or applications (third parties)
    - Custom event BUS = events from own applications
    - Can be access from other accounts
    - Rules to process events
  + Can analyze events depending on schema
  + Schema registry
    - Allows you to generate code for your application that will know in advance how data is structured in the event bus
    - Schema can be versioned
  + Concepts
    - Events
      * Change to AWS environment
    - Targets
      * Process Events
    - Rules
      * Matches incoming events and routs them to targets
* Amazon Simple Notification Service (Amazon SNS)

#### Amazon Simple Queue Service (Amazon SQS)

* + <https://tutorialsdojo.com/amazon-sqs/>
  + Fully managed message queueing service
  + Allows you to decouple applications Scale microservices, distributed systems and serverless apps
  + Pull based (polling) not push based
  + Can be accessed via VPC endpoint
  + Benefits
    - You control who can send to and reveive messages from
    - Server side encryption
    - Durable
    - HA
    - Scalable
    - Multiple senders and multiple revievers at same time
  + Configs
    - Can be subscribed by SNS topic
    - Can trigger lambda function
      * Must be in same region
    - Purging delets all messages in queue
    - Long polling
      * Reduce cost by eliminating number of empty responses
    - Visibility Timeout
      * Period of time SQS prevents other consumers from receiving and processing the message
      * Prevents message redundancy
    - Dead Letter queues
      * Messages that can’t be processed correctly
    - Delay Queues
      * Delay delivery
  + Standard Queue
    - Unlimited throughput
    - At least once delivery
      * May deliver more than one copy
    - Best effort ordering
  + FIFO Queues
    - <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/FIFO-queues.html>
    - First in First out delivery
    - Messages received in order they were sent
    - Exactly once processing
      * No duplicates
    - Message remains in queue until consumer deletes
    - Supports message groups
    - Use case
      * User entered data received in correct order
      * Display correct price by sending price modifications in right order
      * Prevent student from enrolling in a course before registering for an account

### AWS Cost Management:

* AWS Cost and Usage Report
  + <https://tutorialsdojo.com/aws-billing-and-cost-management/>
  + Info about your use of AWS resources and estimated costs for that usage
  + CSV or collection of CSVs
  + Stored in S2 bucket
  + Track your Reserved instance utilization charges and allocation
  + Time can choose
    - Hourly
    - Daily
    - Monthly
  + Can be uploaded to Redshift or Quicksight for analysis
  + Most comprehensive
* AWS Cost Explorer
  + <https://tutorialsdojo.com/aws-billing-and-cost-management/>
  + Helps visualize costs and usage associated with your TOP FIVE cost-acruing AWS services
  + Detailed breakdown of services in the table view
  + Must be enabled before you can use it
    - Must be owner of account to enable
  + Enabling in organization, enables it for all group accounts
  + Create forecasts
  + Available reports
    - EC2 Monthly cost and usage report
      * All AWS costs over past two months
      * Current month to date costs
    - Monthly costs by linked account report
      * View distribution of costs across your organization
    - Monthly running costs report
      * Overview of all your running costs over past three months
      * Provides forecasts
* Savings Plans
  + Based on what you use
  + Alternative to Reserved Instances

### Compute:

#### AWS Application Auto Scaling

#### Amazon EC2

* + <https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>
  + AMI
    - Amazon Machine Image
    - Package OS and additional installations in a reusable template
    - Include
      * Template for root volume
        + OS
        + Apps
        + Applications
      * Launch permissions that control which accounts can use the AMI
      * Block device mapping that specifies the volumes to attach to the instance when its launched
  + Instances
    - Single VM
    - Types
      * T and M
        + General Purpose
      * C
        + Compute optimized
      * R and X and Z
        + Mem optimized
      * D and h and i
        + Storage optimized
      * F and g and p
        + Accelerated computing
    - Key pair secure login
    - Instance Volume Store
      * Non persistent
    - EBS storage
      * Persistent storage
    - Security groups are firewalls for instance
    - User Data
      * Scripts run on instance when first started
      * Can be used to configure or install
      * View user data
        + http://168.254.169.254/latest/user-data
    - Host Recovery for EC2
      * Automatically restarts your instances on a new host in the event of an unexpected hardware failure on a dedicated Host
    - Root Device Volumes
      * Contains boot image
      * Instance store backed instances
        + Data deleted when instance terminated or it fails
      * EBS Backed instances
        + Data is not lost
        + When stopped the instance can be modified
        + By default EBS root volumes are deleted when the instance is terminated
    - Metadata
      * Data used to configure and manage running instance
      * Can view
        + http://192.254.168.254/latest/meta-data
  + Change instance
    - Only works with EBS Storage.
    - Stop instance
    - Change instance type
    - Restart instance
    - Note
      * EC2 Instance does not loose data
  + Networking
    - Elastic IP
      * Fixed IP like Static IP
      * When an instance starts it changes its IP (usually not the same all the time). Like DHCP
      * Attached to one instance at a time
      * Do not pay if attached to server
      * Can only have 5 per account
        + Should avoid using elastic IP
      * Static IPv4
      * Can be rapidly remapped to another instance
      * If not using Elastic IP then you need to enable auto-assign public IP for instances
      * Can only be used within a region
    - Instance either has public IP or Private IP
    - Elastic Network Interface (ENI)
      * Logical network component in VPC (Virtual Network Card)
    - Primary Network Interface
      * Default network interface for VPC
      * Cannot detach primary network interface from instance
  + Enhanced Networking
    - SR-IOV
      * Higher bandwitch, higher PPS (packets per second), lower latency
      * Option I
        + Elastic Network Adapter
        + Up to 100 Gbps
      * Option 2
        + Intel 82599VF
        + Up to 10 Gbps – Legacy
      * Works for newer generation of EC2
    - Elastic Fabric Adapter
      * Improved for HPC, only works in linux
      * Greate for inter-node communications, tightly couples workloads
      * Leverages message Passing Interface (MPI)
      * Bypasses the underlying Linux OS to provide low-latency, reliable transport
  + EC2 Placement Group
    - Sometime you want control over the EC2 placement strategy
    - Strategy
      * Cluster
        + Instances grouped together in low same AZ, same rack, same hardware
        + Low-latency
        + High risk
        + Use

Big data jobs

Most traffic between instances in group

Low latency network requirements

* + - * Spread
        + Instances spread across underlying hardware ( max 7 instances per group per AZ)
        + No two instances share same hardware
        + Limited to single region
        + Minimize failure risk
        + Use

Application needs to be maximized high availability

Critical applications

* + - * Partition
        + Spread instances across diff partitions

Rely on different sets of racks within AZ

* + - * + Each partition does not share hardware with another partition.

Instances within the partition could share hardware

* + - * + 7 partitions per AZ
        + Scales to 100s of instances per group
        + use

Works good for Hadoop, Casssandre, Kafka

* + - Rules
      * Names must be unique
      * Can’t merge placement groups
      * Instance can be launched in one placement group at a time
  + Shutdown and Termination Protection
    - [Terminate your instance - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/terminating-instances.html#Using_ChangingDisableAPITermination)
    - Shutdown Behavior
      * Shutdown from within OS
        + Stop

Default

Ec2 instance will stop as well

* + - * + Terminate

Stops and deletes the Instance

Deletes instance store or EBS backed EC2 volumes

Attached volumes are not deleted by default

* + - * + AWS CLI

InstanceInitiatedShutdownBehavior

* + - Termination Protection
      * [Terminate your instance - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/terminating-instances.html#Using_ChangingDisableAPITermination)
      * Enabled = protect against accidental termination (deletion) in AWS Console or CLI
      * DidableApiTermination
        + Attribute controls whether instance can be terminated using console, cli or api
        + Can still shutdown instance within OS
      * Cant enable on spot instances
      * Does not prevent Auto scaling from terminating instance
        + Use Instance Scale-in protection to prevent this
  + Troubleshooting EC2 Launch
    - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/troubleshooting-launch.html#troubleshooting-launch-internal
    - InstanceLimitExceeded
      * reached your limit of max number vCPUs per region
      * on-Demand and Spot instance limits are set per region
      * Resolution
        + Launch instance in diff region or request AWS to increase your limit
    - InsufficientInstanceCapacity
      * AWS does not have enough On-Demand capacity in the AZ
      * Resolution
        + Wait for capacity to become available
        + If requesting more than one instance, reduce the number requested

Possibly request each instance one by one instead of all at once.

* + - * + Request diff instance type
        + Launch in diff AZ
    - InstanceTerminatesImmediately
      * Pending to terminate
      * Reached your EBS volume limit
      * EBS snapshot is corrupt
      * Root EBS volume is encrypted and you do not have permissions to access the KMS key for decryptions
      * The instance store-backed AMI used is missing a required part
      * Examin EC2 console
        + Description tab will tell you why
    - SSH
      * Unprotected Private Key file
        + Incorrect permission on private key (pem file)
      * Host key not found, Permission denied, or Connection closed
        + Incorrect username
      * Connection Timeout
        + Security Group not configured correctly
        + NACL not configured correctly
        + Route table incorrect
        + Instance does not have IPv4
        + CPU load on instance is high
  + EC2 Launch Types
    - On-Demand
      * Short workloads
      * Predictable pricing
      * Highest cost but no upfront payment
      * No long-term commitment
    - Reserved
      * Minimum 1 year or 3 years
      * Up to 75% discount
      * Reserved Instances
        + Long workloads
      * Classes
        + Standard

Can only modify some attributes

Can be sold

* + - * + Convertible Reserved workloads

Long workloads with flexible instances

Can change the EC2 instance type over time

Less discount

Cannot be sold

* + - * No upfront , partial upfront, all upfront payment options
        + More upfront = greater discount
      * Scheduled-Reserved Instanaces
        + Deprecated
        + Launch within time window you reserve
        + When you require a fraction of day/week/month
    - Spot Instances
      * Up to 90% discount
      * Short workloads or resilient workloads that don’t loose what they have already done.
      * Cheapest
      * Can lose instances if someone pays more for it ( spot price changes over time )
      * No critical jobs or databases
      * Spot Block
        + Block, prevent, AWS from reclaiming your instance for 1 to 6 hours
        + Deprecated
      * One time request type
        + Will not launch again after it has met its price once.
      * Persistent request type
        + Will launch everytime the spot price is below your max
      * Terminate spot instance
        + Cancel request
        + Terminate spot instances running
      * Spot Fleet
        + Set of spot instances and on-demand instances
        + Will try to meet target capacity with price constraints
        + Strategies

Lowest price

Lauch instances from pools with lowest price first

Good for short workloads

Diversified

Distributed across all pools

Great for availability and long workloads

Capacity optimized

Pool with the optimal capacity of the number of instances

* + - * Spot Instance Pool
        + Set of unused EC2 instances with the same instance type, OS, AZ, network platform
        + Can stop and start at will
        + EBS Backed
        + Can modify instance type
        + Application strategy for spot instances

Lowest price

Instances come from a pool with lowest price

Default strategy

Diversified

Spot instances distributed across all pools

Capacity Optimized

Instances come from the pool with optimal capacity for number of instances launching

InstancePoolsToUseCount

Distributed across the number of spot pools you specify

* + - Dedicated Hosts
      * Entire physical server, your software/os on dedicated server
      * Control instance placement
      * Use if you have compliance requirements
      * Allow you to use existing server-bound software licenses
      * 3 year commitment
      * More expensive
    - EC2 Dedicated Instances
      * Instances running on hardware that’s dedicated to you, VMs running on host dedicated to you.
      * May share hardware with other instances in your account
      * No control over instance placement
  + Burstable Instances
    - Instance can burst CPU for unexpected processing
    - Burst Credits
      * Burst credits are used during burst periods
      * If low credits then you probably are using the wrong instance type
      * If out of credits then cpu will drop way down
    - Unlimited instances do not use burst credits
      * Cost more when bursting
      * [Work with burstable performance instances - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/burstable-performance-instances-how-to.html)
  + Monitoring EC2
    - Cloudwatch for EC2
      * See Cloudwatch
    - Status Checks
      * Automatic checks to ID hardware and software issues
      * System Status Checks
        + Monitors problems with AWS systems
        + Issues on the physical host

Loss of network connectivity

Loss of system power

Software issues on the physical host

Hardware issues on the physical host that impact network reachability

* + - * + Personal Health Dashboard

To check if there is any scheduled critical maintenance by AWS

* + - * + If any problems you will need to stop and restart the EC2 instance

This migrates the instance to another host

* + - * Instance status check
        + Monitors software / network config of your instance

Incorrect networking or startup configuration

Exhausted memory

Corrupted file system

Incompatible kernel

* + - * + Reboot the instance or change instance config
    - Automate Recovery
      * Cloudwatch Metrics and Recovery
        + Monitor

SttusCheckFailed\_**System**

Monitors

Loss of network

System power

Physical host software issues

Hardware issues

StatusCheckFailed\_**Instance**

Monitors

Instance software

Network issues

StatusCheckFailed **(both**)

* + - * + Cloudwatch Alarm

Monitor and recover instance with same private/public IP etc

Send notification using SNS

* + - * + AutoScaling Group

Set min/max/desired to 1

If issue then new instance will be started

* + EC2 Hibernate
    - In memory RAM is preserved
    - Like pause
    - OS is not stopped / restarted
    - Under the Hood
      * RAM must be written to the EBS root volume
      * Volume must be encrypted
    - Use cases ( when it makes sense to hibernate as opposed to restart )
      * Long running processing
      * Saving the RAM state
      * Services that take time to initialize
    - Only supports the following instances
      * C3,C4,C5,M3,M4,M5,R3,R4,R5
    - Instance RAM must be less than 150 GB
    - Not supported for bare metal
    - AMI: Linux 2, Linux AMI, Ubuntu and Windows
    - Root Volume must be EBS and encrypted
    - Only avail for on Demand and reserved instance
    - Cannot be hibernated more than 60 days
  + Security
    - IAM to control access to instances
    - Security Group
      * Firewall for Instance
      * All outbound by default
      * Stateful
  + EC2 Scheduled Events
    - [Scheduled events for your instances - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-instances-status-check_sched.html)
    - Fully managed by AWS, you cannot schedule events for your instances
    - If event affect your instance, an email notification will occur
    - Types of events
      * Instance Stop
        + Instance stopped.
        + When started again it is migrated to new host
        + Only for EBS backed instances
      * Instance retirement
        + Stopped if backed by EBS
        + Terminated if backed by instance store
      * Instance Reboot
      * System Reboot
        + Host for instance is rebooted
      * System maintenance
        + Instance might be temp affected by network maint or power maint

#### Amazon EC2 Auto Scaling

* + Overview
    - Scale out to match an increased load
    - Scale in to match decreased load
    - Ensure we have a minimum and maximum number of machines running
    - Automatically register / deregister instances to a load balancer
    - Attributes
      * Launch configuration
        + AMI+instance type
        + EC2 User Data
        + EBS Volumes
        + Security Groups
        + SSH Key Pair
      * Min Size / Max Size / Initial Capacity
      * Network + Subnets information
      * Load balancer information
      * Scaling policies
    - Auto Scaling Alarms
      * Possible to scale an ASG based on Cloudwatch Alarms
      * Alarm can monitor a metric
      * Metrics are computed for the overall ASG instances
      * Based on alarms
        + Scale-out policies
        + Scale-in Policies
    - Auto Scaling Rules
      * Target Average CPU Usage (average over all instances)
      * Number of requests on the ELB per instance
      * Average Network in
      * Average Network out
      * Custom Metric
    - Brain Dump
      * Scaling policies can be on CPU, Networ, etc and can even be on custom metrics or based on schedule
      * Can use launch configs or Launch templates
      * Update ASG means to provide new Launch config / template
      * IAM roles attached to ASG will get assigned to EC2
      * ASG is free pay for only what is launched
      * If instance gets terminated, then ASG will create new instance automatically
      * ASG can terminate instances marked as unhealthy by a LB and then replace them
    - Available for scalable AWS Resources
      * EC2 ASG
        + Launch or terminate EC2 instances
      * EC2 Spot Fleet requests
        + Launch or terminate instances from spot fleet request
        + Automatically replace instances that get interrupted for price or capacity reasons
      * ECS
        + Adjust ECS service desired count up or down
      * DynamoDB (table or global secondary index)
        + <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AutoScaling.html>
        + Adjust write capacity and read capacity with Auto Scaling
        + Uses AWS Application AutoScaling
        + Dynamically adjust throughput capacity
        + Scaling policy

Specifies whether you want to scale read capacity or write capacity or both

Max / min provisioned capacity

Target utilization

Percentage of consumed provisioned throughput at a point in time

Target Tracking

Algorithm to adjust the provisioned througput

* + - * Aurora
        + Dynamic Read Replicas
  + Scaling Policies
    - Dynamic Scaling Policies
      * Target Tracking Scaling
        + Most simple
        + Adjust capacity depending on target metic
        + Optimize for availability = 40%
        + Balance availabity and cost = 50%
        + Optimize for cost = 70%
        + Can choose own metric and target value
        + Can set

Disable scale in

Cooldown period

Warmup time

* + - * Simple / Step Scaling
        + When CloudWatch alarm is triggered then add / remove x units
      * Scheduled actions
        + Anticipate scaling based on known usage patterns
    - Predictive Scaling
      * Continuously forecast load and schedule scaling ahead
    - Metrics to scale on
      * CPUUtilization
        + Average CPU utilization across instances
      * RequestCountPerTarget
        + Number of requests per EC2 instance
      * Average Network In / Out
      * Any custom metric
    - Scaling cooldown
      * Period after scaling activity (in or out)
      * Default 300 seconds
      * During cooldown ASG will not launch or terminate additional instances
      * Allows for metrics to stabilize
      * Suggest using ready to use AMI so they start / stop faster
  + Lifecycle Hooks
    - By default an instance goes into service immediately
    - Hooks allow you to add extra steps before the instance goes into service or terminates
    - Pending State
      * You can perform steps between pending and inService states
    - Terminate State
      * Perform action before system terminates
    - Use Cases:
      * Cleanup
      * Log extraction
      * Special health checks
    - Can integrate with EventBridge, SNS and SQS
  + Launch Configuration vs Launch Templates
    - Both
      * Can specify
        + AMI
        + Instance Type
        + Key pair
        + Security groups
        + Etc
      * Can’t edit
    - Launch Config
      * Depricated
      * Must be re-created every time
    - Launch Template
      * Versioning
      * Parameter subset
        + Partial config for reuse and inheritance
      * Provision using both On-Demand and Spot or mixed
      * Supports
        + Placement Groups
        + Capacity Reservations
        + Dedicated hosts
        + Multiple instance types
        + T2 unlimited burst feature
    - SQS with ASG
      * Scale ASG based on messages in SQS
      * Create cloudwatch metric / alarm on queue length
    - ASG Healthchecks
      * High availability = 2 instances in at least 2 AZ ( multi-AZ ASG )
      * Types
        + EC2 status Checks
        + ELB Health Checks
        + Custom health checks

Send health via CLI or SDK

* + - * ASG will Terminate and relaunch new instance if one becomes unhealthy
      * CLI
        + Set-Instance-Health

Use with custom health checks

* + - * + Terminate-instance-in-auto-scaling-group
    - Troubleshooting
      * Cant launch new instances when other instances are already running
        + ASG group has reached the limit set by the Max Capacity
        + Need to increase this value
        + Capacity issue in AZ
      * Launching EC2 instances is failing
        + Security group does not exist
        + Key pair does not exist
      * If there are issue launching EC2 instances over 24 hours, ASG will suspend scaling processes (administration suspension
  + CloudWatch for ASG
    - Metrics are collected every 1 min
    - ASG Metrics (must enable at ASG to see metrics)
      * GroupMinSize
      * GroupMaxSize
      * GroupDesiredCapacity
      * GroupInServiceInstances
      * GropuPendingINstances
      * GroupStandbyInstances
      * GroupTerminatingInstances
      * GroupTotalInstances
    - EC2-Level Metrics ( enabled by default)
      * CPUUtilization, etc
      * Basic = 5 minute
      * Detailed = 1 minute

#### Amazon EC2 Image Builder

#### AWS Lambda

* When associated to custom resource, function invoked whenever the custom resource is created, updated or deleted (via cloudformation)
* Cloud formation calls API to invoke function

### Database:

#### Amazon Aurora

* + Overview
    - AWS Proprietary tech
    - Compatible with Postgres and MySQL
    - AWS Cloud optimized
      * Better performances over MySQL and Postgres on RDS
    - Aurora storage auto grows from 10 Gb to 64 TB
    - Read Replicas
      * Up to 15 read replicas
      * Replication is faster
      * Supports cross region replication
      * Can associate a Priority
        + Controls which Read Replica will become the master during failover
        + Lowest tier ( number ) will get promoted
        + If two have the same priority then the largest in size will get promoted
    - Failover is instantaneous as it is HA native
    - Can migrate RDS MySQL Snapshot to Aurora MySQL Cluster
  + Cloudwatch metrics
    - AuroraReplicaLag
      * Amount of lag when replicating updates from the primary instance
    - AuroraReplicaLagMaximum
      * Max amount of lag across all DB instances in cluster
    - DatabaseConnections
      * Current number of connections to a DB instance
    - InsertLatency
      * Average duration of insert operations
    - Eventual Consistency
      * High lag means users will get diff experience depending on what replica they read from
  + High Avalability and Read Scaling
    - 6 copies across 3 AZ
    - Only 4 of 6 copies needed for writes
    - 3 of 6 copies needed for reads
    - Self healing
      * Backend peer to peer replication if data is corrupted
    - Striped across 100s of volumes
    - Master
      * Only one instance writes
      * But failover is quick less than 30 seconds
    - Aurora writer endpoint
      * Points to master
      * Client speaks to endpoint
    - Can autoscal read replicas
      * Reader endpoint
        + Client speak to endpoint which load balances across read replicas
  + Security
    - Encryption at rest with KMS
    - Automated backups, snapshots and replicas also encrypted
    - Encryption in flieght using SSL
    - Possible to auth with IAM tokens same as RDS
    - You are responsible for security groups
  + Backup
    - Auto backups
      * Retention
        + 1 – 35 days
    - Allows Point in Time Retention to within 5 minutes of current time
    - Restores to new DB Cluster
    - Backtracking
      * Rewind DB cluster back and forth in time upto 72 hours
      * In place restore - Doesn’t create new DB Cluster
      * MySQL only
    - DB Cloning
      * Create new DB Cluster that uses the same Volume as the original cluster
      * Copy on write protocol
        + Copies to the new cluster volume
      * Uses
        + Test environments using prod data

#### Amazon ElastiCache

* + <https://tutorialsdojo.com/amazon-elasticache/>
  + Overview
    - Help reduce load from DB by putting data in memory
    - Makes applications stateless
    - Managed service
    - Requires app code changes
  + Use cases
    - DB
    - Session data (makes stateless app)
  + REDIS
    - Multi AZ with auto failover
    - Read replicas to scale reads and have high avail
    - Data durability using AOF persistence
    - Backup and restore features
    - Replication Cluster Modes
      * Cluster mode Disabled
        + One Primary up to 5 replicas
        + Asych replication
        + Primary node used for read and write
        + Replicas read only
        + One shard

All nodes have all the data

* + - * + Guards against dataloss
        + Multi-AZ
        + Scale read performance
        + Scaling

Horizontal

Scale by adding/removing read replicas (up to 5)

Vertical

Scale up/down to larger/smaller node (instance) types

To do this Elasticache does this behind the scenes

Creates new node group

Replicates data from old to new

Updates DNS

* + - * Cluster Mode Enabled
        + Data partitioned across many shards

Help to scale writes

Like striping across shard

* + - * + One primary up to 5 replcias
        + Multi-AZ
        + 500 nodes per clustr spread primary and replicas
        + Two modes of Scaling

Online Scaling

Continue serving requests during the scaling process

No downtime

Some degradation of performance

Offline scaling

Take cluster offline

Can make more config changes

* + - * + Horizontal scaling

ReSharding

Add / remove shards

Shard Rebalancing

Equally distribute the keyspaces among the shards as possible

Supports online or offline scaling

* + - * + Vertical Scaling

Change larger or smaller node type

Online scaling

* + - Metrics
      * Evictions
        + Number of non-expired items the cache evicted to allow space for new writes
        + Memory Overfilled
        + Solution

Choose eviction policy to evict expired items

Scal up to larger note type ( more memory ) or add more nodes

* + - * CPUUtilization
        + CPU Util for entire host
        + Solution

Scale up to larger note or add more nodes

* + - * SwapUsage
        + Should not exceed 50 MB
        + Solution

Verify you have enough reserved mem configured

* + - * CurrentConnections
        + Number of concurrent and active connections
        + Solution:

Application behaviour

* + - * DatabaseMemoryUsagePercentage
        + Memory utilization
      * NetworkBytesIn/Out
      * NetworkPacketsIn/Out
      * Replication Bytes
        + Voluem of data being replicated
      * Replication Lag
        + How far behind the replica is from the primary node
  + MemCache
    - Multi-node for partitioning of date ( sharding )
    - No high availability
    - Not persisteant
    - No bakup / restore
    - Multi-thread
    - Scaling
      * 1-40 nodes
      * Horizontal
        + Add/remove nodes from cluster
        + Auto-discovery allow your app to find nodes

All nodes in the cluster maintain list of metadata with info for other nodes

* + - * Vertical
        + Larger/smaller node types
        + Process

Create new cluster

Update application to use new endpoints

Delete old cluster

* + - * + Clusters / nodes start out empty (no backup)
    - Metics
      * Evictions
        + Number of non-expired items the cache evicted to allow space for new writes
        + Memory Overfilled
        + Solution

Choose eviction policy to evict expired items

Scal up to larger note type ( more memory ) or add more nodes

* + - * CPUUtilization
        + CPU Util for entire host
        + Solution

Scale up to larger note or add more nodes

* + - * SwapUsage
        + Should not exceed 50 MB
        + Solution

Verify you have enough reserved mem configured

* + - * CurrentConnections
        + Number of concurrent and active connections
        + Solution:

Application behaviour

* + - * Freeable Memory
        + Amount of free memory on the host

#### Amazon RDS

* + Overview
    - Relational Database Service
    - Managed SQL language DB
    - Types
      * Postgres
      * MySQL
      * MariaDB
      * Oracle
      * Microsoft SQL Server
      * Aurora
    - Managed by AWS
      * Automated provisioning
      * Continuous backups
        + Automatically enabled

Daily full

Transaction every 5 minutes

Point in time restore within 5 minutes

* + - * + 7 day retention (can be up to 35)
        + DB Snapshots

Manually triggered by user

Retain as long as you want

* + - * Monitoring
      * Read replicas
      * Multi az
      * Maintenance windows
      * Scaling
        + Auto scaling

Scales when storage gets close to threshold

Max storage threshold. (don’t grow over this)

* + - * Storage backed by EBS
  + Read Replicas for Scalability
    - Up to 5 replicas
    - Same AZ, Cross AZ, Cross Region
    - Async replication
      * Reads eventually consistent
    - Replica can be promoted to its own database to accept reads
    - Application must update connection string to point to all replicas
    - Only for SELECT Statements
      * Writing to read replca can break the replication
    - Cost
      * Replica in same region not cost to transfer between AZ
      * Cross region replication pay for transfer
    - Read replicas can be setup as Mult AZ for DR
    - If the value for Max\_Allowed\_Packet parameter for a read replica is less than the Max\_allowed\_packet for the source DB instance, replica error can occur
      * Custom parameter
      * Specify the max size of data manipulation language (DML) that can be run on the DB
      * [Troubleshooting for Aurora - Amazon Aurora](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_Troubleshooting.html#CHAP_Troubleshooting.MySQL.ReplicaLag)
  + Multi AZ
    - Standby replica in different AZ
      * Sync replication
      * One DNS name for master and replicas
      * Auto failover to replicas if primary fail
      * No manual intervention
    - Not used for scaling. Replica is used for standby
    - Config
      * Zero downtime
      * Modify DB and enable multi AZ
    - Failover conditions
      * when the Primary DB
        + Failed
        + Is being patched
        + Unreachable due to network
        + Modified (DB instance changed)
        + Busy and unresponsive
        + Underlying storage failure
      * AZ outage
      * Manual failover of the DB instance initiated using Reboot with failover
  + RDS Encryption and Security
    - At rest encryption
      * Master db and replicas can be encrypted
      * Default at launch
      * If master is not encrypted, the read replica cannot be encrypted
      * Transparent data Encryption (TDE) available for Oracle and SQL
    - In Flight Encryption
      * SSL Certs
      * To enforce SSL
        + Postgress

Rds.force\_ssl=1 in AWS RDS Console

* + - * + MySQL

GRANT USAGE ON \*>\* TO ‘mysqluser’@’%’ REQUIRE SSL;

* + - * Operations
        + Encrypt RDS backup

Snapshot retain RDS database encryption/unencryption

Can copy a snapshot into an encrypted one

* + - * + Encrypt Un-encrypted RDS Database

Create snapshot of the un-encrypted database

Copy snapshot to encrypted snapshot

Restore DB from encrypted snapshot

Migrate app to use new DB

* + - Network and IAM Security
      * RDS DB deployed in private subnet
        + Security Groups
      * IAM policies control who can manage AWS RDS
      * Username and password to login to DB
      * IAM based authentication can be used to login to RDS MySQL and PostgresSQL
        + Uses Auth token

Lifetime 15 minutes

* + - * + Benefits

Network must be encrypted SSL

IAM used to centrally manage users instead of DB

Leverage IAM roles ad EC2 instances

* + RDS Proxy
    - Allows resources (Lambda Functions ) not in your RDS VPC to access the RDS DB
    - Handles cleaning of idle connections
    - Autoscales
  + RDS Parameter Groups
    - Configure DB engine
    - Dynamic parameter
      * Applied immediately
    - Static Parameter
      * Applied after reboot
    - Modify parameter group associated with DB
      * Requires reboot
    - PostgresSQL /SQL Server:
      * Rds.force\_ssl=1
      * Force sql connections
  + RDS Backups and Snapshots
    - Backups
      * Continuous
      * Point in time recovery
      * During maintenance windows
      * Can retain backups when deleting DB
      * Retention period
        + 0-35 days
      * Disable backups = retention period of 0
    - Snapshots
      * Takes IO operations and can stop the DB from seconds to minutes
      * Snapshots taken on Multi AZ DB don’t impact the master – only the standby
      * Incremental snapshot
      * Can copy and share across accounts
        + Manual snapshots can be shared
        + Auto snapshots must be copied first
        + Can only share unencrypted snapshots or encrypted snapshots with Customer Managed Key (must also share key)
      * Manual snapshots don’t expire
      * Final snapshot when delete snapshot
    - Restoring from backup or snapshot will create new DB instance
  + RDS Events and logs
    - RDS keeps record of events
      * DB instances
      * Snapshots
      * Parameter groups…
    - Can send to SNS, EventBridge, etc
    - Log files
      * Send logs to cloudwatch logs
        + Metric filter
        + Trigger alarms
  + RDS Cloudwatch
    - [Overview of monitoring metrics in Amazon RDS - Amazon Relational Database Service](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/MonitoringOverview.html)
    - Metrics (gathered from hyper visor)
      * Database Connections
      * SwapUsage
      * ReadIOPS/WriteIOPS
      * Read Latency / Write Latency
      * Read Throughput/Write Throughput
      * DiskQueueDepth
      * FreeStorageSpace
    - Enahanced monitoring
      * OS level monitoring
      * Gathered from agent running on instance
      * Useful when you need to see different processes or threads used by CPU
      * Access to over 50 new CPU, mem, file system, and disk I/O
  + RDS Performance Insights
    - Visualize DB performance
    - Analyze why there are issues
    - Filter by
      * Waits
        + Resource that is the bottleneck (CPU,IO,Lock, etc)
        + What it waits on by the most
      * SQL Statements
        + Find which statement is use most resources
      * Host
        + What application server is using DB the most
      * Users
        + What user is using the DB the most
    - DBLoad
      * Number of active sessions for the DB Engine
    - Can view what SQL queries are putting load on DB

Management, Monitoring, and Governance:

* AWS CloudFormation
  + <https://tutorialsdojo.com/aws-cloudformation/>
  + Template to create collection of related AWS resources and provision them in an orderly way
  + User Data
    - User scripts
    - Pass entire script using Fn::Base54
    - User data script log is in /var/log/cloud-init-output.log
  + Cfn-init
    - CloudFormation Init
    - Stack must include AWS::CloudFormation::Init resource metadata
    - Makes complex scripts readable
    - Logs = /var/log/cfn-init.log
    - More declarative
  + Cfn-signal
    - Wait conditions
      * Wait for template to wait until it gets signal
      * Can be used to detect fail or success
      * Works with cfn-init
      * Wait condition in metadata
    - Failures troubleshooting
      * Ensure the AMI has the AWS Cloudformation Helper scripts installed
      * Verify cfn-init and cfn-signal command run successfully by checking logs
        + Must disable rollback to view logs
      * Must have connection to internet
        + This is how it talks to cloud formation service
        + Curl -l <https://www.amazon.com> to test internet connection
    - Creation Policy (ASG only plus a few others)
      * Waits for signal of x number to show the ASG has been created
    - Update Policy (ASG Only plus a few others)
      * Attributes
        + AutoScalingReplacingUpdates

Creates new ASG with new config and terminate old one

* + - * + AutoScalingRollingUpdates

Specifies how many instances remain online and how many will be updated at one time.

* + - * + AutoScalingScheduledActions

Allows you to ignore ASG actions that would grow or shrink the number of systems running

* + Rollback
    - Stack creation fails it will rollback (all deleted)
    - Can disable rollback
    - If updating stack and it fails then the stack will rollback to previous know working stack
      * Here you will be able to see what happened it the logs
    - Update\_rollback\_failed
      * Can’t rollback
      * Fix error manually
      * Or skip resources marked as rollback
    - Can set rollback triggers using cloudwatch alarms
  + Stacks
    - Template
    - Will rollback if resource cannot be provisioned
    - Update methods
      * Direct Update
      * Creating and executing Change sets
    - Output
      * Export output to share between stacks
      * Output stays within account and region
  + Nested Stacks
    - Allow you to isolate repeated patterns /common components in separate stacks and call them from other stacks
  + ChangeSets
    - Preview proposed changes to stack
    - Won’t say if it will be successful but they show what will change during an update
    - Creating change set can update stack
  + Drift
    - Manual changes can be made to stack
    - Drift detects any changes to the stack not made by cloudformation
  + Deletion Policy
    - Controls what happens when the cloudformation template is deleted
    - DeletionPolicy
      * Retain
        + Preserve / backup in case of cloudformation deletes
      * Snapshots
        + EBS vol, ElastiCache cluster, elasticache replication group, RDS DBInstance, RDS DBClustr, Redshift cluster
        + Instance deleted but snapshot made before deletion
      * Delete
        + Default behavior
        + (except for AWS:RDS::DBCluster default is Snapshot)
        + To delete S3 bucket all objects need to be deleted from bucket first
  + Termination protection
    - Prevents accidental deletion
  + DependsOn
    - Resource will not be created until the dependson resource has been created.
    - Allows one resource to be completed first
    - Orders resource creation
  + Stackset
    - Provision common set of AWS Resources across multiple accounts and reagions
    - Can use lambda to create custom actions
    - Admin account
      * AWS account where the stackset is created
    - Target Account
      * Create, update, delete on or more stacks in stackset
  + Stack Policies
    - <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/protect-stack-resources.html#stack-policy-reference>
    - JSON file
    - Similar to IAM policy
    - Can prevent resources from being updated accidentally
      * Protected resources
    - Can overrule policy during update
      * Create temp policy to override and apply duringin update creation
  + Artifacts
    - Stack Template File
      * Defines resources that cloudformation provisions
      * JSON or YAML
    - Template Configuration File
      * Json
      * Specifies template parameters values or a stack policy
  + Monitoring
    - Integrates with CloudTrail
  + Security
    - Use IAM and cloudformation to control what users can do with cloudformations (create, delete stacks )
    - Service Role
      * Allows CF to make calls to resources on your behalf
    - Use Interface VPC Endpoint to increase VPC Security
* AWS CloudTrail
  + Overview
    - Governance, compliance, audit
    - Enabled by default
    - Get history of events/ api calls made within your aws account aby
      * Console
      * SDK
      * Cli
      * Aws services
    - Can move logs from CT to CT Logs or S3
    - Trail can be applied to all regions (Default) or a single region
    - Events
      * Management Events
        + Operations performed on resources in AWS account
        + Default management events are logged
        + Types

Read Events

No modification

Write Events

Make changes

* + - * Data Events
        + Not logged by default
      * CloudTrail Insights Events
        + Paid service
        + Detects unusual activity
        + Monitor write type of events
    - Retention
      * 90 days by default
      * To keep longer move to S3
    - Log File Integrity Validation
      * [Validating CloudTrail log file integrity - AWS CloudTrail (amazon.com)](https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-intro.html)
      * Digest Files (Hash)
        + Reference log files for last hour contain Hash
        + Stored in same S3 bucket
        + Helps you determine if log file tampered with
      * Should still protect S3 bucket
    - Integrate clouttrail with EventBridge
      * React to any API call
      * Delivers an event 15 minutes
      * Log files to s3 with in 5 minutes
    - Organization Trails
      * Log events for all member accounts
      * Member accounts can only view
* Amazon CloudWatch
  + <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html>
  + <https://tutorialsdojo.com/amazon-cloudwatch/>
  + Over View
    - Monitoring service for AWS cloud resources and the applications you run on AWS
    - Collects and tracks metrics
    - Collect and monitor log files
      * Centralize logs
    - Set alarms
    - Basic metrics collected every 5 minutes (free)
    - Detailed metrics collected every 1 minute (paid)
    - Retention
    - Monitor and store logs to help better understand and operate your systems and applications
    - Real-Time Application and System Monitoring
      * Monitor using log data
      * Can track number of errors that occur in your application logs and send you a notification whenever the rate of errors exceeds a threshold
    - Long-term log retention
      * Store log data indefinaitely in highly durable and cost effective storage without worrying about hard drives running out of space
      * Quickly move rotated and non-rotated logs files off host
    - Metrics do not aggregate across regions
      * Each region has separate counts
  + Metrics are provided automatically for a number of AWS products and Services
    - EC2 instances
      * CPU
        + CPU Utilization
        + Credit Usage/Balance (burst)
      * Network
        + Network In / Out
      * Disk
        + Read / Write for Ops / Bytes ( only for instance store )
      * Status Checks
        + Is the instance healthy or not
        + Instance status = check EC2VM health
        + System Status = Check the underlying hardware
      * RAM is not inclued
      * Custom Metrics
        + Basic every 1 minute
        + Hi resolution every 1 second
        + RAM
        + Application Level Metrics
        + Instance custom metrics requires an IAM Role to push to Cloudwatch
        + Requires cloudwatch agent
    - EBS Volumes
    - Elastic Load Balancer (ELB)
    - Auto Scaling
    - EMR job flow
    - RDS DB instances
    - DynamoDB tables
    - ElastiCache Cluster
    - Redshift clusters
    - OpsWorks stacks
    - Route53 health checks
    - SNS topics
    - SQS Queues
    - SQF workflows
    - Storage gateways
  + Unified Cloudwatch Agent
    - Overview
      * For virtual servers
        + EC2 Instances
        + On Prem Servers
        + Etc
      * Collect additional system-level metrics
        + RAM
        + Processes
        + Used disk space
        + Etc
      * Collect logs to send to cloudwatch logs
        + Default EC2 instance does not send logs
      * Centeralized configuration of agent using SSM Parameter Store
      * IAM Permissions need to be correct to push to Cloudwatch from the agent
      * Default namespace for metics collected by agent start with CWAgent
        + Configurable
      * Procstat Plugin
        + Collect metrics and monitor system utilization of individual processes
        + Supports linux and windows
        + Select which processes to monitor

Pid\_file: name of process identification number files they create

Exe: process name that match string you specify (regex)

Pattern: Command lines used to start he process (regex)

* + - * + Metrics collected by procstat plugin begin with procstat
  + Metrics
    - Provides metrics for every service in AWS
    - Namespaces
      * Collection of Metrics
    - Dimension
      * Name/value pair
      * Attribute of a metric
        + Instance ID
        + Environment
        + ETC
      * Up to 10 per metic
    - Timestamp
    - EC2 Metrics
      * Basic Monitoring
        + Metrics collected every 5 minutes (free)
      * Detailed Monitoring
        + Metrics every 1 minute (cost)
        + Can be used to scale faster
      * Mem is not pushed. Must be custom metric
      * Disk usage is not pushed
    - Can get metrics for two weeks in the past or two hours in the future.
    - Statistics
      * Aggregation over specified time
    - Metric Math
      * Query multiple cloudwatch metrics and use math expressions to create new time series based on new metrics
  + Custom Metrics
    - Define own metrics
    - PutMetricData
      * API call to push metric
    - Metric Resolution (StorageResolution API parameter)
      * Standard
        + 1 minute
      * High Resolution
        + 1/5/10/30 seconds (costs more)
    - Can use metric data points two weeks in the past and two hours in the future
  + CloudWatch Metric Streams
    - Continuous near real-time stream of metrics
  + CloudWatch Dashboard
    - Shows metrics and alarms
    - Single pane view
    - Graphs from diff accounts and regions
      * Dashboards are global not regional
      * Requires detailed monitoring
    - Can change timezone and time ranges
    - Can setup auto refresh
    - Share dashboard with non AWS accounts
    - Cross-account and cross reagion
  + CloudWatch Logs
    - Log aggregator
    - Log Groups
      * Groups of logs
    - Log Streams
      * Log files coming from instance containers, etc
    - Log expiration policy
      * How log to keep logs
    - Sources
      * SKD
      * CloudWatch Log Agent (deprecated)
      * CloudWatch Log Unified Agent
      * Elastic Beanstalk
      * ECS container logs
      * AWS Lambda Functions
      * VPC Flow logs
      * API Gateway
      * CloudTrail based on filter
      * Route 53
    - Metric Filter
      * Find specific data about log files
      * Count what is found
      * Filters can be used to trigger alarms
    - Insights
      * <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html>
      * Can be used to query logs and add queries to cloudwatch dashboards
      * auto discover fields in logs from AWS Services
      * single request can query up to 20 log groups, timeout after 15 minutes, results available for 7 days
      * Interactive, pay as you go and integrated log analytics capability for cloudwatch logs
      * Search and visualize logs
      * Allows you to understand your applications and make improvements and find and fix problems quickly
    - S3 Export
      * Can export logs directly to S3 bucket
      * Takes up to 12 hours to become available for export
      * API Call : CreateExportTask
      * Not real time
    - Subscriptions
      * Filter to logs then send to destination (another AWS Service)
      * Realtime
    - Log Aggregation across accounts and across reagions
      * Use subscription to aggregate and send on
  + Cloudwatch Alarms
    - Trigger notifications from any metric
      * Including metric Filters
    - States
      * OK
        + Not triggered
      * Insufficient\_data
        + Just started, the metric is not avail or not enough data
      * Alarm
        + Over threshold
    - Period
      * How long in seconds to evaluate metric
    - Targets
      * EC2 Instance
        + Stop, Terminate,Reboot, Recover
      * Trigger Auto Scaling Action
      * Send notification to SNS
        + Lambda
        + Subscription
        + Etc
    - AWS CLI SetAlarmState
      * Can be used to test alarms
  + Service Quotas Alarms
    - Service Quota Overview
      * Limits for each AWS service
      * Usually region specific
      * Can request increase
      * Service quota console will show limits
    - Alarms
      * Notify you when close to quota
      * Create on Service Quota Console
      * Alternative
        + Trusted Advisor with CW Alarms
        + Not as robust
* AWS Command Line Interface (AWS CLI)
* AWS Compute Optimizer
  + Reduce cost and improve performance by recommending optimal AWS resources for your workloads
  + Lower cost upto 25 %
  + Right size workloads
  + Uses machine learning and tracking cloudwatch metrics
  + Supports
    - EC2 instances
    - EC2 Auto scaling groups
    - EBS Volumes
    - Lambda functions
* AWS Config
  + Audits and records compliance
  + Record config and changes over time
  + Example
    - Is there unrestricted SSH to my security groups
    - Do my buckets have public access
    - How ALB config changed over time
    - Etc
  + Can recieve alerts via SNS notification for any change
  + AWS config is per region service
  + Can aggregate across regions and accounts
    - Aggregator created in one region
    - Aggregate rules , resources across multiple accounts and region
      * Rules at each region
    - If using AWS organizations no need for authorization
    - Ideal to used CloudFormation
  + Config can be stored in S3
  + Config Rules
    - Types
      * AWS managed rules ( built by AWS )
      * Custom rules
        + Defined in AWS Lambda
    - Rules can be evaluated / triggered
      * For each config change
      * And / or evaluated at regular intervals
    - AWS Config rules does not prevent changes from happening
    - No free tier
  + Config Resource
    - View the compliance of a resource over time
    - View configuration of a resource over time
  + Can be linked to cloudtrail api calls over time
  + Remediation
    - SSM automation document
      * Including retries up to 5 times
  + Notifications
    - Eventbridge to trigger notifications for not compliance
    - Or send configuration changes and compliance state notifications to SNS
      * All events
      * Use SNS Filtering or client side filtering
* AWS Control Tower
  + Way to set up and govern a secure and compliant multi-account AWS environment based on best practices
  + Automate setup
  + Benefits
    - Automate the setup of your environment
    - Automate ongoing policy management using guardrails
    - Detect policy violations and remediate them
    - Monitor compliance through an interactive dashboard
  + Runs ontop of organizations
    - Auto sets up AWS Organizations to organize accounts and implement SCPs
* AWS License Manager
* AWS Management Console
* AWS OpsWorks
  + Overview
    - Use Chef and Puppet to perform server config automatically or repetitive actions
    - Work with EC2 or onPrem
    - Managed Check and Puppet
    - Alternative to SSM
* AWS Organizations
  + OverView
    - Global service
    - Allows management of multiple AWS Accounts
    - Main account is master and can’t be changed
    - Other accounts member accounts
      * Only part of one organization at a time
      * Can be migrated to another organization
        + Remove member account from old org
        + Send invite to account from new org
        + Accept invite
      * Migrate root account
        + Remove all accounts
        + Then migrate
    - Consolidated Billing across all accounts
      * Single payment
      * Price benefit with volume discounts
    - Api to automate AWS Account creation
  + Why
    - Multiple accounts per department, per cost center, per dev/test/prod
    - Based on regulatory restrictions
    - Beter resource isolation
    - Separate per-account service limits
  + Use tagging standards for billing purposes
  + Enable cloudtrail on all accounts to send logs to central S3 Account
  + Cross account roles for admin purposes
  + Organizational Unit
    - Organize accounts
  + Service Control Policies (SCP)
    - Whitelist or blacklist IAM Actions
    - Applied at the OU or Account level
      * Does not apply to master Account
      * Applies to all users and roles including root
    - Does not apply to service linked roles
      * Service linked roles enable other AWS services to integrate with AWS organizations and can’t be restricted by SCP
    - By default nothing is allowed, so explicit Allow is required
    - Use cases
      * Restrict access to certain services
      * Enforce PCI compliance by explicitly disabling services
    - Precidence
      * Explicit deny wins
  + Aws:PrincipalOrgID
    - Condition key in IAM Policy
    - Restrict access to IAM principals from accounts in org
    - Use to allow/deny access to any account without specifying each account
  + TAG Policies
    - Enforce tagging in all accounts
  + Reserved Instance Sharing
    - Turned on by default
    - All accounts in the org can reveive the hourly cost benefit of Reserved instances that are purchased by any other account
    - Can be set by organization for each account
    - If turned off
      * Discounts apply only to the account that purchased the RIs
      * RI Discounts from other accounts in the organization don’t apply
      * Charges accrued on that account are still added to the organizations bill
* AWS Personal Health Dashboard
  + Global service
  + Show outages directly impact you
  + Show impact to your resources
  + List issues and how to fix
  + Shows maintenance events form AWS
  + AWS Health API
  + Aggregates across multiple accounts of AWS organization
  + Notifications
    - EventBridge
      * Reacts to changes for AWS Health events
      * Can’t be used to return public events from Service Health Dashboard
* AWS Service Catalog
  + Quick self service portal to lauch set of authorized products defined by admsin
  + Products = CloudFormation Templates
  + Portfolio = Collection of Products
  + Control = IAM permissions to allow access
  + User can pic products with not knowing how to setup and configure services
  + Sharing catalogs
    - Share catelog with accounts or orgs
    - Sharing Options
      * Share a reference portfolio
        + Import the ref into recipient account
        + Stays in sync
      * Deploy Copy of portfolio
        + Copy port
        + Must be redeployed to update changes
  + Tag Option
    - Defined in service catalog
    - Can be associated with Portfolios and products
    - Tags get inherited by services started with products
* AWS Systems Manager
  + <https://tutorialsdojo.com/aws-systems-manager/>
  + OverView
    - Helps you manage EC2 and on Prem systems at scale
    - Get operational insights about the state of infrastructure
    - Easily detect problems
    - Patching automation for enhanced compliance
    - Linux and Windows
    - Integrated with CloudWatch metrics / dashboards
    - Integrated with AWS Config
    - Free Service
    - SSM agent must be installed on system we will control
    - Installed on Linux 2 AMI by default
    - Can push to EC2 Instance
    - EC2 instanc needs IAM roles to allow SSM actions
  + Tags and Resource Groups
    - Tags can be used for resource grouping, etc
    - Resource groups allow you to work on multiple resources at one time
  + Documents
    - JSON or YAML
    - Define parameters
    - Define actions
    - Used to
      * Run command
      * State manager
      * Patch manager
      * Automation
      * Retrieve info from parameter store
  + Run Command
    - Execute a document
    - Run command across multiple instances (using resource groups)
    - Rate control ( how many instances to run against
    - Error Control
    - Integrated with IAM and CloudTrail
    - No need for SSH
    - Command output can be shown in
      * Console
      * Sent to S3 bucket
      * Cloudwatch logs
    - Send notifications to SNS about status
    - Can be invoked using EventBridge
  + SSM Automation
    - Simplifies common maintenance and deployment tasks of EC2 instances and other resources
    - Automation runbook
      * SSM Document of type automation
      * Defines actions performed on your EC2 instance or resource
      * Predefined runbook or custom runbooks
    - Triggered
      * Manually by AWS Console, AWS CLI or SKD
      * EventBridge
      * Schedule using maintenance windows
      * SQS Config rule remediation
  + SSM Inventory
    - Collect meta data from managed instances ( including on Prem)
    - Metadata includes
      * Installed software
      * OS Drivers
      * Config
      * Isntalled updates
      * Running service
      * Etc
    - Can be viewed in AWS Console or stored in S3 and query and analyze using Athena and QuickSight
    - Can spcify metadata collection interval
    - Query data from multiple AWS Accounts and regions
    - Create custom Inventory for custom metadata
  + State Manager
    - Used to automate process of keeping your managed instances in a state that you define
    - Use cases:
      * Bootstrap instances with software
      * Patch OS/Software update on schedule
      * Etc
    - State Manager Association
      * Defines the state that you want to maintain
      * Specify schedule when this will be applied
    - Use SSM Documents to create an association
  + Patch Manager
    - Automate process of patching instances
    - Both AWS and on prem
    - Linux and windows
    - On demand or on schedule
    - Scan instances and generate patch compliance report (missing patches)
      * Which can be sent to S3
    - Patch Baseline
      * Which patches should / should not be installed
      * Ability to create custom patch baselines ( specifying approved/rejected patches)
      * Patches can be autoapproved
      * By default install only critical patchs and security patches
    - Patch Groups
      * Associate set of instances with patch baseline
      * Instances should be defined with the tag key Patch Group
      * An instance can only be in one Patch Group\
      * Patch group can only be registered with one patch baseline
  + SSM Maintenance Window
    - Defines a schedule for when to perform actions on your instances
    - Contains
      * Schedule
      * Duration
      * Set of registered instances
      * Set of registered tasks
  + Session Manager
    - Allows you to start secure shell ( SSH ) on your EC2 instances and on prem servers
    - Access it through AWS Console, AWS CLI, or Sesseion manager SDK
    - Do not need SSH access, bation host or ssh keys
    - Requires
      * SSM agent needs to be installed
      * IAM Permissions
        + Control with user/groups can access Session Manager and which instances
        + Use tags to restric access to only specific EC2 instances
        + Access to SSM + write to S3 + write to cloud watch
        + Restrict commands a user can run in a session
    - Supports linude , MacOS and Windows
    - Log connections to your instances and Executed command ( regular SSH does not do this)
      * Can be sent to S3 or Cloudwatch logs
    - CloudTrail can intercept startSession events
* AWS Systems Manager Parameter Store
  + Securly store config and secrets
  + Optional seamless encryption using KMS
  + Serverless, scalable, durable, easy SDK
  + Version tracking of config / secrets
  + Config management using path and IAM
  + Notifications with CloudWatch Events
  + Integration with CloudFormation
  + Hierarchy
    - Custom created
  + Reference Secrets from secret manager
  + Reference AWS parameters
  + Parameter store tiers
    - Standard
      * Free
      * 10000
    - Advanced
      * .05 per parameter per month
      * 100000 parameters
  + Parameter Policies
    - Only for Advanced Tier
    - Allow to assign TTL to parameter to force updating or deleting sensitive data
    - Can assign multiple polices at one time
  + Cheaper than Secrets Manager
  + Simple API
  + No secret Rotation
  + KMS encryption optional
  + Can pull secret using SSM Parameter Store API
* AWS tools and SDKs
* AWS Trusted Advisor
  + <https://tutorialsdojo.com/aws-trusted-advisor/>
  + Analyzes AWS environment and provides best practice recommendations in the following categories
    - Cost Optimization
      * Low utilization EC2 instances
      * ELB idle
      * Under Utilized EBS
      * Reserve instances and saving plans optimizations
    - Performance
      * High utilization EC2 instance, cloudfront optimizations
      * EC2 to EBS throughput optimizations, alias records recommendations
    - Security
      * MFA enabled on root account
      * IAM key rotaiong
      * Exposed access keys
      * S2 bucket permission for public access, security groups, unrestricted ports
    - Fault Tolerance
      * EBS Snapshot age
      * AZ balance
      * Auto scaling group multi-AZ, RDS, Multi-AZ, ELB Config
    - Service Limits
  + Access to seven core checks are available to all
  + Tiers
    - Core checks and recommendations
      * All customers
      * Weekly email notification
    - Full trusted Advisor
      * Business and enterprise support plan
      * Cloudwatch alarms
      * Programmatic access using support API

### Migration and Transfer:

* AWS DataSync
  + Replicate all file attributes and metadata
* AWS Transfer Family

### Networking and Content Delivery:

#### AWS Client VPN

* + Software used to VPN to AWS Virtual Private Gateway
  + Use if only a few need connection to VPC

#### Amazon CloudFront

* + <https://tutorialsdojo.com/amazon-cloudfront/>
  + Content Deliver Network
  + Improves Read Performance
    - Cached at AWS edge locations
  + DDOS protection
  + Integrations with AWS Shield and AWS Web Application Firewall
  + Can expose external HTTPS with cert and can talke to internal HTTPS backends
  + Origins
    - S3 buckets
      * Distribute files global
      * Caching
      * CloudFront Origin Access Identity OAI
        + Enhanced security
        + Allows Como only between cloudfront and S3 bucket
        + IAM role
        + Use

Monitor

Cookies

* + - * Ingress to upload files
    - Custom Origin ( HTTP )
      * Application load balancer (must be public)
      * EC2 Instance (must be public)
      * S3 website (must first enable bucket as static S3 website
      * Any HTTP Backend
      * Security group must allow all cloudfront public IPs
  + CloudFront GEO Restriction
    - Restrict who can access
      * White list
        + Users from these countries can access
      * Black list
        + Users from these countries cannot access
    - Country is determined by third party GEO-IP database
    - Use case
      * Copyright laws
  + CloudFront vs S3 Cross Region Replication
    - Global Edge Network
    - Files are cached
    - Great for static content available everywhere
    - S3
      * Must be set up for each region
      * Read only
      * Real time replication
      * Dynamic content low latency updates
  + CloudFront Access Logs
    - Logs request made to cloudfront into logging S3 Bucket (diff than Origin Bucket)
    - Reports
      * Based on data from access Logs
      * Types
        + Cache Statistics Report
        + Popular Objects Report
        + Top Referrers Report
        + Usage Reports
        + Viewers Report
  + Troubleshooting
    - Caches HTTP 4xx and 5xx status codes returned by S3 (or origin server)
      * 403
        + User does not have access to underlying bucket
      * 404
        + Object user is requesting is not found
      * 5xx
        + Gateway issues
  + CloudFront Caching
    - Cache based on
      * Headers
      * Configure Cache
        + Forward all headers to origin

No caching

Every request goes to origin

TTL = 0

* + - * + Forward whitelist of headers

Caching based on headers

* + - * + No headers

Forward only default headers

No caching based on request headers

Best caching performance

* + - * Session Cookies
        + Specific header with key/value pairs
        + Settings

Default: do not process

Caching is not based on cookies

Cookies are not forwarded

Forward WhiteList of cookies

Caching based on values in all the specified cookies

Forward all cookies

Worst caching performance

* + - * Query String Parameters
      * Cache behaviour for query strings
        + Query string parameters are in the URL

Default

Do not process the query string

Cachin is not based on query strings

Forward whitelist of query strings

Caching based on parameter whitelist

Forward all query strings

Worst performance

* + - Cache resides on edge locations
    - Want to maximize cache hit
      * Separate distributions
        + static

No headers / session caching rules

* + - * + dynamic

cache bases on headers and cookies and queries

* + - * increase Cache Hit Ratio
        + Monitor cloudwatch metric

CacheHitRate

* + - * + Specifiy how long to cache your objects

Cache Control Max age header

* + - * + Specify non or minimally required

Header

Cookies

Query string parameters

* + - TTL – Time To Live
      * How long info stays in cache without being queried
      * Control via
        + Cache Control Header

Cache-Control: max-age

Preferred to expires header

If origin always sends back Cache-Control header then TTL can be controlled by that header

Min/Max boundaries can be set by customize object caching settings

This can set a default value if header is missing

* + - * + Expires Header
    - Can invalidate part of cache with CreateInvalidation API
    - Origin Custom Headers
      * Set a constant header / header value for all requests
      * All requests will have some headers
      * Use
        + Tell origin request from cloudfront

Cloudfront-is-Desktop-Viewer

Cloudfront-is-Movile-Viewer

CloudFront-is-SmartTV-Viewer

Cloudfront-is-Tablet-Viewer

* + - Behaviour Settings
      * Set whitelist of headers to forward
      * Cache related settings
  + CloudFront with ALB Sticky Sessions
    - Must forward / whitelist cookie that controls session affinity to the origin to allow the session affinity to work
    - Set TTL to value lesser than whenthe authentication cookie expires

#### Elastic Load Balancing

* + Scalability
    - Virtical
      * Increase size of your instance
      * Scale up
    - Horizontal (elasticity)
      * Increase the number of instances
      * Scale out
      * Distributed systems
      * Auto scaling group
      * Load balancer
  + High availability
    - At least two AZ
    - Goal to survive AZ loss
    - Linked to Horizontal scaling
    - Auto scaling group multi AZ
    - Load balancer AZ
  + Over View
    - Forward traffic to multiple backend servers (downstream)
    - Only expose single point of access to app (Single DNS )
    - Seamlessly handle failures of downstream instances
    - Regular health checks
      * Knows if resource is properly working
      * ELB disables traffic to node if the health is bad
      * Uses port and route to check health
    - SSL Termination
    - Stickiness with cookies
    - High availability across zones
    - Separate public traffic from private traffic
    - Managed load balancer (
      * AWS guarantees it is working
      * Takes care of upgrades, maintenance high availability
      * Provies only few configuration settings
    - Costs less than setting up your own
    - Integrated with many AWS services
  + Types of load balancers
    - Classic
      * V1 – old generation 2009
      * http, https (Layer 7), tcp (Layer 4), ssl
      * do not use (deprecated)
      * Health checks are TCP or HTTP based
      * Fixed hostname
        + Xxxx.region.elb.amazonaws.com
      * OutofService
        + Health check failed
        + EC2 instance not registered with CLB yet
    - Application load balancer
      * [What is an Application Load Balancer? - Elastic Load Balancing (amazon.com)](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html)
      * 2016
      * http https (Layer 7), websock
      * mutlple HTTP applications
      * can have multiple apps on same machine (different ports)
      * listener
        + checks for connection requests from clients
        + listens on protocol/port
        + rules determine how traffic is routed
        + must define a default rule
      * Target Group
        + Attached to listeners
        + EC2 instances

Can be auto-scale group

* + - * + ECS tasks
        + Lambda functions
        + IP Address
        + Health check done at target group level
        + Can be in one or more AZ
      * Route Routing
        + Routing tables to different target groups
        + Based on path in URL (example.com/users)
        + Based on hostname in URL (users.example.com)
        + Routing based on query string, headers
      * ALB good fit for micro services and container-base apps
      * Port mapping feature
        + Allows redirect to dynamic port in ECS (elastic Container Service)
      * Fixed hostname
        + Xxxx.region.elb.amazonaws.com
      * Application Serves do not see IP of client directly
        + Tru IP of client in X-Forwarded-For
        + Can also get Port (X-Forwarded\_Port) and Proto (X-Forwarded-Proto)
    - Network Load Balancer
      * 2017
      * TCP, tls, udp (Layer 4)
      * Handle millions of request per second
      * Less latency – for extreme performance
      * One static IP per AZ
      * Support assigning Elastic IP
      * Not in the AWS free tier
      * Target groups
        + EC2 instances
        + IP Address
        + Application Load Balancer
        + Lambda
      * Server Security Group needs to allow traffic from anywhere not just the NLB security group because the NLB passes traffic through
    - Gateway Load Balancer
      * 2020
      * Operates at lay 3(Network) IP Protocol
      * Deploy scale and manage fleet of 3rd party network virtual appliances ins AWS
        + Firewalls, intrusion detection and preventions systems, deep packet inspection systems…
      * CAN BE used to analyze traffic
      * Combines the following functions
        + Transparent Network Gateway

Single entry/exit for all traffic

* + - * + Load Balancer

Distributes traffic to your virtual appliances

* + - * Uses GENEVE protocol on port 6081
      * Target groups
        + EC2 instances
        + IP Addresses
  + Security Groups for LB
    - Users are allowed to Load balancer
    - Resource security group should only allow security group traffic from ELB
  + Sticky Sessions
    - Session Affinity
      * Client is always redirected to the same instance behind LB
    - Classic load balancer and Application load balancer
    - Cookies used with expiration date for stickiness
    - Use case:
      * Make sure user does not loose session data
    - Could cause imbalance
    - Cookies
      * Application bases cookie
        + Custom cookie

Generated by application

Each target group gets a cookie name

Can’t use

AWSALB, AWSALBAPP,AWSALBTG

* + - * + Application Cookie

Generated by the LB

Cookie name = AWSALBAPP

* + - * Duration based Cookie
        + Generated by LB
        + Name = AWSALB or AWSELB (clb)
  + Cross Zone Load Balancing
    - The load balancer in each AZ will distribute traffic evenly amongst all EC2 instances in all AZ
    - ALB this is always on and can’t disable
      * Not charge for inter AZ data
    - NLB disabled by default
      * You pay for inter AZ data transfer
    - CLB disabled by default
      * No charge for inter AZ
  + SSL Certificates
    - LB does HTTP SSL Termination
    - Certificates are managed by AWA Certificate Manager
    - Server Name Indication
      * Solves the problem of loading multiple SSL certificates onto one webserver
      * New protocol requires the client to indicate the hostname of the target server in the initial ssl handshake
      * Only works for ALB and NLB, or CloudFront
  + Connection Draining
    - CLB – Connection Draining
    - ALB or NBL – Deregistration Delay
    - Before stopping an instance, the LB gives some time for existing requests to complete
    - 1 to 3600 seconds
    - Default 300
    - 0 disables
  + Health Checks
    - Statuses
      * Initial
        + Registering the target
      * Health
      * Unhealthy
      * Unused
        + Target is not registered
      * Draining
        + Target is de-registering
      * Unavailable
        + Health checks disabled
    - If target group contains only unhealthy targets, ELB routes requests across all targets
  + Monitoring
    - Metrics push to Cloudwatch from all LB
      * BackendConnectionErrors
      * HealthyHostCount
        + Number of targets healthy
      * UnHealthyHostCount
        + Number of targets considered unhealthy
      * HTTPCode-Bakckend
        + 2xx
        + 3xx
        + 4xx
        + 5xx
      * Latency
      * RequestCount
      * RequestCountPerTarget
      * SergeQueueLength
      * SpillOverCount
        + Request rejected because the queue was too long
  + Troubleshooting
    - Successful request = 200
    - Unsuccessful at client = 4xx
      * 403 = client sent malformed request that does not meet HTTP Specifications
    - Unsuccessful at server = 5xx
      * 503 = Service Unavaiable
      * 504 = Gateway Timeout
  + Logging
    - <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html>
    - Disabled by default
    - Access logs from LB can be store in S3
      * Bucket must be in same region as Load Balancer
    - Contain
      * Time
      * Client IP
      * Latencies
      * Request Path
      * Server Response
      * Trace ID
    - Helpful for compliance
    - Already encrypted
  + Tracing
    - Request tracing – each HTTP request has an added custom header added
      * X-Amzn-Trace-ID
    - Help track single request
    - ALB not working with X-Ray
  + Target Group Attributes
    - Detregistration\_delay.timeout\_seconds
      * Time the LB waits before deregistering a target
    - Slow\_start.duration\_seconds
      * Default target receives full share of requests once it is registered
      * Gives healthy target time to warmup before the load balancer sends them a full request
      * Exit slow start when:
        + Duration period ends
        + Target unhealthy
    - Load\_balancing.algorithm.type
      * How the load balancer selects targets when routing requests
        + Round Robin

Equally choose the targets from group

ALB and CLB

* + - * + Least Outstanding Requests

Next instanc to receive the request is the instance with the lowest number of pending/unfinished requests

Works with ALB and CLB

* + - * + Flow Hash

NLB

Target based on protocol, source/destination IP, source /destination port and TCP sequence number

Each TCP/UDP connection is routed to a single target for the life of the connection (suedo sticky)

* + - Stickiness.enabled
    - Stickiness.type
      * Application-based or duration-based
    - Stickiness.app\_cookie.cookie\_name
      * Name of the application cookie
    - Stickiness.app\_cookie.duration\_seconds
      * Application-based cookie expiration period
    - Stickiness.lb\_cookie.duration\_seconds
      * Duration-based cookie expiration perios
  + ALB Rules
    - Default rule last rule
    - Process in order
    - Supported actions
      * Forwarded
      * Redirect
      * Fixed response
    - Rule conditions
      * Host-header
      * HTTP-requested method
      * Path-pattern
      * Source-ip
      * http-header
      * query-string
    - target group weighting
      * specify weight fro each target group on a single rule
      * blue / green deployment

#### AWS Firewall Manager

#### AWS Global Accelerator

#### Amazon Route 53

* + <https://tutorialsdojo.com/amazon-route-53/>
  + Overview
    - HA, scalable, fully managed and authoritative DNS
      * Authoritative = customer can update DNS records
    - Domain registrar
      * Registers domain names
    - Can check health of resources
    - 100% SLA
  + Domain Registration
    - If register with Route 53, will make it auto owner
      * Creates hosted zone that is same name as your domain
      * Assigns a set of four name servers to the hosted zone (ns records)
      * Gets servers from hosted zone and adds them to domain
    - If register with another service
      * Transfer to Route 53
  + DNS Records
    - Domain/subdomain name
    - Record type
      * A
        + Maps hostname to IPv4
      * AAAA
        + Mpas hostname to IPv6
      * CNAME
      * Maps hostname to another hostname
      * Only works with non root domain name
      * Alias
        + Point hostname to AWS Resource

ELB

CLoudfront

API Gateway

Elastic Beanstalk

S3 websites

VPC interface endpoints

Global accelerator

Route 53 record in the same hosted zone

* + - * + Works for root domain and non root domain
        + Free
        + Native health check built in
        + AWS only
        + Always of type A/AAAA
        + Can’t change TTL
      * NS
        + Name servers for hosted zone
        + IP for DNS servers own zone
    - Value – IP Address
    - Routing policy
      * How route 53 responds to queries
      * Routing policies
        + Simple

Route traffic to single resource

Can specify multiple values in same record

Client chooses random answer

Alias enabled can only specify one resource

Cannot associate health checks

* + - * + Weighted

Control % of requests that go to each resource

Weights don’t have to sum to 100

DNS records must have same name and type

Can use health checks

Use case

Load balancing between regions,

testing new application versions

weight of 0 stops sending traffic

if all have weight of 0 then all resources have equal weight

* + - * + Failover

Must use health check on primary instance

If primary is fails health check then Route 53 returns IP to secondary instance

* + - * + Latency Based

Redirect to resource with lowest latency

Based on traffic between user and resource

Can use health checks

* + - * + Geolocation

Based on where user is located

Specify on

Continent

Country

U.S. State

Should create default record to use if there is not location match

Use case

Website localization

Restrict content distribution

Load balancing

Can use health checks

* + - * + Multi-Value

Route to multiple resources

Can use health checks

Up to 8 healthy records returned

Not substitute for ELB

client chooses randomly

* + - * + GeoProximity

Route 53 traffic flow feature

Simplify process of creating and maintaining records in large and complex configurations

Visual Editor to manage complex routing decisions

Configurations can be saved as Traffic Flow Policy

Can be applied to different Rout 53 hosted zones

Supports versioning

Route based on location of users and resources

Shift more traffic based on defined bias

More traffic = expand bias (1 to 99)

Less traffic = shrink bias (-1 to -99)

Resources

AWS resources (region

Non aws resources (lat and long)

* + - Health Checks
      * Check health of public resources ( mainly)
      * Allow automated failover
        + Monitor endpoint

About 15 global health checkers will check endpoints

Health/Unhealthy threshold – 3 (default)

Interval – 30 sec (default) or 10 sec (fast health check)

http/https/tcp

if > 18% of health checksers report endpoint is healthy, Route 53 considers it healthy

can choose which locations you want Route 53 to use for health checks

Health checks pass with 2xx or 3xx status codes

Can be set up to pass / fail based on ttext in first 5120 bytes of response

* + - * + Monitor other health checks (Calculated Health Checks)

Combine the results of multiple health checks into a single health check

Can use OR, AND, NOT

Up to 256 child health chekcs

Specify how many need to pass to make parent pass

* + - * + Monitor cloudwatch alarms (helpful for private resources)

Create cloudwatch metric with alarm

Health check then monitors alarm

* + - * + Integrated with CloudWatch Metrics
        + Router/firewall must allow incoming requests from Route 53 Health Checkers
      * Target Health Flag
        + If true will route 53 will not answer with that resource
    - TTL
      * TTL of cache
      * TTL balances traffic to Route 53 and outdated records if IP changes
      * Mandatory for all records except Alias records
  + Hosted Zone
    - Container for records that define how route traffic to a domain and its subdomains
    - Public Hosted Zones
      * Records for traffic on the internet
    - Private Hosted zones
      * Route traffic within one or more VPCs
  + S3 Website
    - Create S3 bucket with exact same name as the DNS website name
    - Enable S3 as website and public
    - Create alias record
    - Only works for HTTP not HTTPS (requires CloudFront)

#### Amazon Route 53 Resolver

* + By default auto answers queries for
    - Local domain names for EC2 instances
    - Record in private hosted zone
    - Records in public name servers
  + Hybrid DNS
    - Resolve queries between VPC and other networks
      * VPC itself / Peered VPC
      * On-prem Network ( connected through Direct Connect or VPN )
  + Resolver Endpoints
    - Inbound Endpoint
      * DNS resolver on your network can forward DNS queries to Route 53 resolver
      * Allows DNS resolver to resolve domain names for AWS resources or records created in Route 53
    - Outbound Endpoint
      * Route 53 resolver conditionally forwards DNS queries to your DNS resolver
      * User Resolver Rules to forward DNS queries to your DNS Resolver
    - Associate with one or more VPC in same Region
    - Create in two AZs for HA
  + Resolver Rules
    - Controls which DNS queries are forwarded to DNS resolvers on your network
    - Rules
      * Conditional Forwarding Rules
        + Forward DNS queries for specific domain and all its sub domains to target IP addresses (DNS servers)
      * System Rules
        + Override forwarding rules
      * Auto-defined System Rules
        + Defines how DNS queries for selected domains are resolved
    - If multiple rules match most specific will be chosen

#### AWS Transit Gateway

* + Transit Gateway
    - Transitive peering between thousands of VPC and on prem datacenter including direct connected and VPN
    - Hub and spoke connection
    - Cross account and cross region (using Resource Access Manager)
    - Route tables = limit which VPC can talk with other VPC
    - Supports IP Multicast ( only AWS service )
    - ECMP
      * Equal Cost Multipath routing
      * Routing strategy to allow forward packet over multiple best path
      * Use case:
        + Create multiple site-to-site VPN connections to increase bandwidth of your connection to AWS

#### Amazon VPC

* + <https://tutorialsdojo.com/amazon-vpc/>
  + Overview
    - Virtual Private Cloud
    - Can have multiple VPC in a region
      * Soft limit max 5 per region
    - IP Ranges
      * Max CIDER per VPC is 5
        + Min /28 = 16 IP addr
        + Max /16 = 65536 IP Addr
        + Only private IP addr allowed

10.x.x.x

172.16.x.x

192.168.x.x

* + - * + CIDR should not overlap other VPC or corp net
      * AWS reserves first 4 IP address and last IP for each subnet
        + Example

10.0.0.0/24

10.0.0.0 = Network address

10.0.0.1 = VPC Router

10.0.0.2 = Reserverd for mapping to Amazon provide DNS

10.0.0.3 = reserved for future use

10.0.0.255 = broadcast address (broadcast is not supported in VPC so the address is reserved and cannot be used)

* + - Default VPC
      * Each account has one
      * EC2 instances are started here by default
      * Has internet connectivity
      * All ec2 instances have IPv4 public in this VPC
      * Public and private DNS name
    - Internet Gateway
      * Allows resources in VPC to connect to internet
      * Scales horizontally and is highly available and redundant
      * Must be created separately from VPC (not during VPC creationg)
      * One VPC can only be attached to one IGW and vice versa
      * On their own IGW do NOT allow internet access
        + Route tables to allow traffic to and from the internet
    - Bastion Host
      * EC2 instance in public subnet of VPC that we connect to and allows us to pass thru to private subnet inctances
      * Should only have port 22 open from your IP address
      * Private EC2 instances should allow 22 from bastion host
    - NAT Instances
      * Outdated
      * Allow EC2 instances in private subnets to connect to the internet
      * Must be launched in the public subnet
      * Must disable Source destination check
        + Nat instance must be able to send and receive traffic whenthe source or destination is not itself (acts on behalf of private EC2 instance)
      * Must have elastic IP attached to it
      * Route tables must be configured to route traffic from private subnets to the NAT instance
      * Preconfigured Amazon Linux AMI
        + EOL Dec 31 2020
        + Not highly available / resilient

You have to configure all of this

* + - * + Internet traffic depends on EC2 instance
        + Manage security group rules

Allow SSH from your home network

Allow HTTP/HTTPS traffic from Private subnets

All HTTP/HTTPS out to internet

* + - NAT Gateway
      * Replacing NAT Instances
      * Allows instances in Private subnet to access internet but prevents internet from initiating contact
      * Aws managed
      * Higher bandwidth,
      * high availability
        + redundant within AZ
        + should have one in another AZ for high availability
      * Pay per hour for usage and bandwidth
      * Can’t be used by EC2 instances in same subnet
        + Requires an IGW ( Private Subnet =>NATGQ(public subnet) =>IGW
      * 5 Gbps bandwidth auto scale to 45 Gbps
      * No Security Groups to manage / required
      * Connectivity Types
        + Public

Default

Private instances can connect to internet but cannot receive connections from internet initiated

Must be in public subnet

Must have Elastic IP

NATGW created in a specific AZ with Elastic IP

Can use public NATGW to connect to other VPCs or to ON-Prem network

Route traffic from NATGW through

Transit Gateway

VPN Private Gateway

* + - * + Private

Private instances can connect to other VPCs or on prem network

Route traffic from NATGW through

Transit Gateway

VPN Private Gateway

If you connect internet gateway, traffic will be dropped

* + - VPC DNS Resolution
      * EnableDNSSupport
        + Defaults to true

Allows DNS resolution from Route 53 resolver server to VPC

Queries Amazon Provider DNS server at 169.254.169.253 or the reserved IP address at the base of the VPC IPv4 network range plus two

* + - * + False

Must have your own DNS server within the VPC

* + - * EnableDNSHostnames
        + Default

True = Default VPC

False = Newly created VPC

* + - * + Requires EnableDNSSupport to be true
        + True = assigns public hostname to EC2 instance if it has a public IPv4
        + When to use

If you use custom DNS domain names in a privete hosted zone in rout 53, you must set both of these

* + - Security Groups and NACLs
      * Security Group
        + Apply to resource
        + Stateful

Whatever is allowed in is allowed out

Whatever is allowed out, answer is allowed in

* + - * + Allow rules only
        + All rules evaluated
      * NACL
        + Apply to Subnet
        + Stateless

Outbound rules need to be allowed

Inbound rules need to be allowed

* + - * + Allow and deny rules
        + One NACL per subnet
        + Rules are ordered lower number looked at first

First rule match wins

* + - * + Last rule Asterisk deny all
        + Newly created NACL will deny everything
        + NACL are great at blocking specific IP
        + Default NACL

Accepts everything inbound/outbound with the subnets its associated with

* + - * + Ephemeral Ports

Clients connect to server on defined port

Expects a response on an ephemeral port ( port on client machine )

Diff OS use different ports

These ports need to be opened on NACL or return info will not happen

* + - VPC Endpoints
      * Private endpoint within VPC to connect to AWS resources without using the public internet
      * Also called AWS Private link
      * Redundant and scal horizontially
      * Remove need to IGW, NATGW, …
      * Troubleshooting
        + Check DNS Setting Resolution in VPC
        + Check Route tables

Must have route to resource

* + - * Types
        + Interface Endpoints

Provision ENI as an entry point to AWS Service

Must attach Security Group

Supports most AWS Services

* + - * + Gateway Endpoints

Gateway must be used as a target in a route table

Supports both S3 and Dynamo DB

* + - VPC PrivateLink – VPC Endpoint Services
      * Powers VPC Endpoint
      * Most secure and scalable way to expose a service to a lot of VPCs (own or customer accounts)
      * Requires
        + Network load balacer

On your service VPC

* + - * + ENI

On your client VPC

* + - * If NLB is in multi zones, then ENI needs to be in those multi Zones to create fault tolerance
    - Reachability Analyzer
      * Net diag tool to troublehoot network connectivity between two endpoints in your VPC
      * Builds a model of the network configuration then checks reachability based on these configs with no packets
    - VPC Peering
      * Privately connect two VPCs useing AWS network
      * Behave as they are in the same network
      * CIDR must be diff
      * Not transitive
      * Must update route tables in each VPC to ensure instances can commo with each other
      * Good to know
        + Peer between VPC in different Accounts and reagions
        + You can reference a security roup in a peered VPC
    - VPC Flow Logs
      * Capture info about IP Traffic going into your interfaces
        + VPC Flow Logs
        + Subnet Flow Logs
        + Elastic Network Interface (ENI) Flow logs
      * Helps to monitor and troubleshoot connectivity issues
      * Flow log data can go to S3 / cloudwatch logs
      * Capture info for AWS managed interfaces
      * Can be used for analytics on usage patterns or malicious behavior
      * Query VPC flow logs using athena on S3 or cloudwatch logs insights
    - Site to Site VPN
      * VPN between VPC and Corp datacenter
      * Requires
        + Virtual Private Gateway

<https://docs.aws.amazon.com/vpn/latest/s2svpn/SetUpVPNConnections.html#vpn-configure-routing>

VPN Concentrator on the AWS side

Created and attached to the VPC from which you want to create the site to site VPN

Possibility to customize the ASN (autonomous system number)

* + - * + Customer Gateway

Software or physical device in corp datacenter

Can be public IP or private IP (with NAT-T device for internet access)

* + - * + Must enable route propagation for the VPGateway in the route table associated with your subnets

To reach VPC from onprem

Create route in Customer VPN Gateway to the Virtual Private Gateway

* + - * + To ping security groups must allow ICMP
      * AWS VPN CloudHub
        + If you have more than one Datacenter connect all of them via VPN to AWS VPNGatewa and with cloudhub all data centers can talk to each other over VPN
        + Dynamic route and route tables required
    - Direct Connect (DX) and Direct Connect Gateway
      * Dedicated private connection from DC to AWS
      * Uses AWS direct connect Locations
      * Allows you to access public and private AWS resources over same connections
      * Requires
        + VP Gateway on your VPC
      * Benefits
        + Lower cost
        + Increase bandwidth
        + More consistent experience
        + Hybrid environments
        + IPv4 and v6
      * Direct Connect Gateway
        + Allows you to create a connection to AWS and then using the gateway connect to multiple VPCs in different reagions
      * Connection types
        + Dedicated Connection

1 Gbps and 10 Gbps

Physical ethernet port dedicated to customer

Request made to AWS first then completed by AWS direct connect partners

* + - * + Hosted Connections

50 Mbps, 500 Mbps up to 10 Gbps

Request made via AWS Direct Connect Partners

Capacity can be added or removed on demand

* + - * Lead times are often longer than 1 month to establish a new connection
      * No encryption but private
      * AWS + VPN with IPsec-encrypted is you want encryption
      * Resiliency
        + High Resiliency for critical workloads

Multiple direct connect locations

* + - * + Max resiliency

Two direct connect location each with two connections

* + - AWS Classlink
      * Deprecated
      * Run instances in single network shared with others
      * Replaced by VPC
      * Classic link is a link between EC2-classic and VPC
    - Transit Gateway
      * Transitive peering between thousands of VPC and on prem datacenter including direct connected and VPN
      * Hub and spoke connection
      * Cross account and cross region (using Resource Access Manager)
      * Route tables = limit which VPC can talk with other VPC
      * Supports IP Multicast ( only AWS service )
      * ECMP
        + Equal Cost Multipath routing
        + Routing strategy to allow forward packet over multiple best path
        + Use case:

Create multiple site-to-site VPN connections to increase bandwidth of your connection to AWS

* + - VPC traffic Mirroring
      * Capture and inspect network traffic in your VPC
      * Route the traffic to security appliances that you manage
      * Capture traffic
        + From (source) – ENIs
        + To (targets) – and ENI or a Network Load Balancer
      * Capture all packets or filter by what you are interested in
      * Source and target same VPC or diff VPC if peered
    - VPC IPv6
      * Every ipv6 is public no private range
      * Cannot disable IPv4
      * When enable EC2 instances get both IPv4 and IPv6 IP
      * Troubleshooting
        + If can’t launch IPv6 EC2 instance in subnet probably because IPv4 subnets ran out
    - Egress only Internet Gateway
      * <https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html>
      * IPv6 only
      * Outbound only , no inbound initiated calls
      * Must update route table

#### Amazon VPC Traffic Mirroring

* + VPC traffic Mirroring
    - Capture and inspect network traffic in your VPC
    - Route the traffic to security appliances that you manage
    - Capture traffic
      * From (source) – ENIs
      * To (targets) – and ENI or a Network Load Balancer
    - Capture all packets or filter by what you are interested in
    - Source and target same VPC or diff VPC if peered

### Security, Identity, and Compliance:

* AWS Certificate Manager (ACM)
  + <https://tutorialsdojo.com/aws-certificate-manager/>
  + Provision, manage and deploy public and privet SSL certs
  + Removes manual purchase, upload and renewing certs
  + Manages cert renewal
    - Can automate
  + Integrated with
    - ELB
    - Cloudfront
    - Elastic Beanstalk
    - API Gateway
    - CloudFormation
  + Public Certs
    - Manages renewal and deployment
    - Cannot install certs directly. Must be deployed via ACM
  + Private Certs
    - Ways to create
      * Delegate private cert management to ACM
        + ACM can auto renew and deploy
      * Export private cert from ACM and use in EC2 instances, containers,on-prem servers and IOT Devices
        + Auto renew these certs and sends a CloudWatch notification when renewal is complete
        + Must manually deploy or create own code to deploy
      * Private CA
        + Create own keys, generate CSR, issue cert
        + Manage certs yourself
        + Manually renew
  + Imported Certs
    - ACM can deploy but renewal is manual
  + Domain Verification for certs
    - Before ACM can issue certs it must verify ownership
      * Email Validation
        + Sends 3 emails to 3 contacts
      * DNS Validation
        + CNAME records to validate
* Amazon Detective
* Amazon GuardDuty
  + Intelligent threat discovery to protect AWS account
  + Use Machine Learning to analys logs
  + Input data
    - Cloudtrail logs
    - VPC flow logs
    - DNS Logs
  + Cloudwatch event rule to be notified if guardduty finds something
  + Events can target SNS or Lambda
  + Can protect against CryptoCurrentcy Attacks
* AWS IAM Access Analyzer
  + Find out which resources are shared externally
  + Define Zone of Trust = AWS Account or AWS Organization
  + Findings = anything outside of Zone of Trust
* AWS Identity and Access Management (IAM)
  + IAM Credentials Report (Account Level)
    - Lists all accounts users ad status of their various credentials
  + IAM Access Advisor
    - User level
    - Permissions granted to user and when those service were last accessed
    - Use to revise policy
  + Controls Internal Team accounts. NOT USER ACCOUNTS
  + Policy document bound to IAM User
    - Entity in AWS defines permissions for identity or resource
    - Allow statements
    - Deny statements
    - Policy types
      * Identity-based policy (IAM Policy)
        + Attach managed and inline policies to IAM identities

Users

Groups

roles

* + - * Resource-based policies
        + Attach inline polices to resources
        + S3 bucket policies and IAM Role Trust policies most common
      * AWS Organizations Service Control Policies (SCP)
        + Define max permissions for account members of an organization or OU
      * Access control Lists (ACL)
        + Control which principals in other accounts can access a resource to which the ACL is attached.
        + Similar to resource based policies
        + Does not use JSON structure
    - Policy Elements (JSON)
      * Effect
        + Allow or Deny
        + Implicit Deny
        + Most restrictive wins (Deny)
      * Principal
        + Account, user, role or federated user to which you want to allow or deny access
        + Only used with Resource-based policy
      * Action
        + List of actions the policy allows or denys
      * Resource
        + List of resources to which the action apply
        + Required for IAM Policy
        + Optional for Resource-based Policy
      * Condition
        + Circumstances under which the policy grants permission.
  + Authentication
    - Sign in to AWS
    - Users
      * Username and password
    - Federated users
      * Lets user outside of AWS to assume IAM Roles
      * Single Sign On
      * External Identity store (AD, Azure AD, LDAP, etc)
      * Assume IAM Role
      * SAML Federation
        + MS AD

1. Auth with AD ( ID Provider )
2. Get SAM Assertion
3. Send assertion to AWS SSO endpoint
4. SSO checks with STS
5. Redirected to Consolw
   * + - Custom ID Broker
         * Custom programed Code
         * Checks with STS in AWS
       - AWS Cognito
         * Public applications
         * Cognito ID Pools

Federated Identities

Get identities for user to get temp creds

ID Source can include

Amazon

Facebook

Google

AWS CUP

OpenID connect providers

SAML providers

Developer authenticated identities (custom login)

Guest access (unauthenticated)

AWS Credendentials have

IAM polices defined in Cognito (users mapped)

Default IAM role for auth users or guest users

Define rules to choose the role for each user

Can be customized based on user\_id for fine grain control

* + - * + Cognito User Pools ( CUP )

Serverless DB for web and mobile apps user

Simple login Username(email) password

Email and phone verification

MFA

Federate -- Associates ID Pool user to email account

Google

Facebook

Twitter

Etc

Can block users if creds compromised elsewhere

Login sends back JSON Web Toke (JWT)

Integrates with API gateway and Application Load Balancer

* + - Single Sign on ( SSO )
      * Centrally manage SSO to access multiple accounts and third party business apps
      * Integrates with AWS Organizations
      * Intgegraes with SAML
      * Integrates with AD
      * Centralized permission management
      * Centralize auditing with Cloudtrail
    - Applications
      * Permissions to access AWS Resources can be assigned
      * Access/secret keys
        + Api level credentials
    - IAM Role
      * Policy document bount to IAM Role
  + Authorization
    - What you can do
    - Policy document
      * Explicitly grants permissions
        + Users

Can belong to multiple groups

* + - * + Groups

Collection of IAM Users

Cannot be nested

Not a principal

IAM document policy

* + - Default = New users have not permissions
  + MFA
    - Second factor auth
  + Access keys rotate
    - AWS Config can be used to identify access keys past due for rotation
  + Remove unnecessary credentials
    - Last time user logged in ( over x days) should probably be removed.
  + AWS STS
    - Security Token Service
    - Allows grant limited and temp access to AWS resource
    - Token valid for 1 hour
    - API Calls
      * AssumeRole
        + Within own account for enhanced security
        + Cross account access
        + How this works

Define IAM Role within your account or cross account

Define which principals can access this IAM Role

Use AWS STS to retrieve creds and impersonate the IAM Role

Creds temp 15 minutes to 1 hour

* + - * AssumeRolewithSAML
        + Return credentials for users logged in with SAML
      * AssumeRoleWithWbIdentity
        + Return credentials for users logged in with IDP

Facebook, google, etc

* + - * + AWS recommends against using this (use Cognito)
      * GetSessionToken
        + For MFA from a user or AWS account root user
* Amazon Inspector
  + Automated security assessments for EC2 instances
  + Check OS against
    - Known vulnerabilities
    - Unitended network accessibility
  + Inspector Agent must be installed on OS
  + Report can be sent to SNS
* AWS Key Management Service (AWS KMS)
  + Encryption Overview
    - Encryption in flight SSL
      * https
      * data encrypted before sending
      * decrypted after receiving
      * SSL Certs
      * Prevents man in the middle attack
    - Server side encryption
      * Encryption at rest
      * Data encrypted on disk after being received
      * Data decrypted before being sent
      * Encryption keys decrypt/encrypt
      * Keys must be stored somewhere
    - Client side encryption
      * Data encrypted by client and not decrypted by server
        + Server should not be able to decrypt
      * Data decrypted by receiving client
      * Could leverage envelope encryption
  + KMS
    - Control who and what can access data
    - Fully integrated with IAM authorization
    - Can use with cli / sdk
    - Customer Master Key (CMK) Types
      * Symmetric ( AES-256 keys )
        + First offering
        + Use single key fore encrypt/decrypt
        + Needed for envelope encrypt
        + Must use KMS API
      * Asymmentric Key (RSA and ECC key pairs)
        + Public key = encrypt
        + Private key = Decrypt
        + Encrypt/decrypt or sign/verify
        + Encryption outside of AWS for users who can’t call KMS API
    - Fully manage keys
      * Create
      * Rotation policies
      * Disable
      * Enable
    - Cloud trail can audit usage
    - Types of CKM Keys
      * AWS Managed Service Default CMK
        + Free
      * User keys created in KMS
      * User Keys imported
        + Must be 256 bit symmetric key
    - When to use
      * Any time you need to share sensitive data
    - The value of KMS is that the CMK used to encrypte can never be retrieved by the user and it is rotated regularly
    - Never store secrets in plaintext
    - Can only encrypt 4KB of data per call
    - Give access
      * Key policy allows user to access key
      * IAM policy allow API call
    - Key bound to region
    - Key policy
      * Cannot control access without
      * Default KMS policy
        + Create if you don’t provide specific key polic
        + Root user gets full access
        + Give IAM policy to KMS Policy
      * Custom KMS Key Policy
        + Define users, roles that can access KMS Key
        + Define who can administer the key
        + Useful for cross account access
    - Cannot change encryption keys used by EBS volume
      * If you want to
        + Then create EBS Snapshot
        + Create new volume from snapshot
        + Specify new KMS key
    - Sharing KMS encrypted RDS DB Snapshots
      * [Allowing users in other accounts to use a KMS key - AWS Key Management Service (amazon.com)](https://docs.aws.amazon.com/kms/latest/developerguide/key-policy-modifying-external-accounts.html)
      * Cross-account access requires permission in the key policy of the KMS key and in the IAM policy in the external user account
        + Key Policy - Must share KMS CMK with target account using key policy (giving Other account IAM user / role permissions)
        + IAM policy - in external account must delegate the key policy permissions to the users and roles
    - Key Deletion
      * Schedule key for deletion
      * Has waiting period of 7-30 days in case you want to cancel
        + Pending Deletion
        + Can not be used for cryptographic operations

Can’t decrypt/encrypt

* + - * + Key is not rotated
      * You can create a cloudwatch alarm to check if key is being used
  + KMS Key Rotation
    - Can be automatic for customer managed CMK
      * 1 year schedule
      * Previous key will remain enabled to decrypt old data
      * CMK ID will remain same but backend key will change
    - Manual key rotation
      * New key will have diff CMK ID
      * Keep previous key active so you can decrypt old data
      * Beter to use aliases when encrypt / decrypt
        + Hides CMK ID
        + UpdateAlias API Call
  + CloudHSM
    - KMS = AWS managed software for encryption
    - HSM = AWS provision encryption hardware
    - Dedicated Hardware
    - Manage own keys
    - Symmetric and asymmetric keys
    - Good for SSE-C (customer KMS)
* AWS License Manager
* AWS Secrets Manager
  + Meant for storing secrets
  + Can force rotation every X days
  + Automate generation of secrets on rotation with lambda
  + Integrates with RDS
    - Mostely meant for RDS
  + Encrypted with KMS
  + Monitoring
    - Cloudtrail captures API calls
    - Cloudtrail captures other related events that might have a security or compliance impact on your AWS account or might help troubleshoot
      * No API Service Events
        + RotationStarted Event
        + RotationSucceeded Event
        + Rotation Failed Event
        + RotationAbandoned event

Manual change instead of auto rotation

* + - * + StartSecretVersionDelete Event
        + CancelSecretVersionDelte
        + EndSecretVersionDelete Event
    - Can combine events with cloudwatch logs and alarms
  + Troubleshooting
    - Rotation
      * Lambda function logs to see why rotation failed
  + More expensive than Secretes Parameter store
  + Auto rotation with Lambda
  + Lambda functions provided for RDS, Redshift, documentDB
  + KMS Mandatory
* AWS Security Hub
* AWS Service Dashboard
  + Shows all regions
  + All service health
  + Shows historical info foreach day
  + RRS Feed
  + <https://status.aws.amazon.com>
* AWS Shield
  + Shield standard
    - Enabled by default at no cost
    - Protects against DDOS attacks
      * Protect most common DDOS
  + Shield Advanced
    - Premium service
    - Protects against more sophisticated attacks
    - Access to response team
    - Protects against higher fees during usage spikes
* AWS WAF
  + Filter requests based on rules
  + Web Application Firewall
  + Layer 7
    - http(s)
  + APP Load balance
  + Cloudfront
  + API Gateway
  + Define Web ACL
    - Rules can include
      * IP Address
      * HTTP Headers
      * HTTP Body
      * Or URI strings
    - Protect against common attack
      * SQL injection
      * Cross-site scripting (XSS)
    - Size constraints
    - Geo-match
      * Block certain countries
    - Rate-based rule
      * Count occurrences of events
        + DDOS protection

### Storage:

#### Amazon Elastic Block Store (Amazon EBS)

* + Overview
    - Elastic Block Store
    - Network drive
    - Persist data
    - Can only be mounted to one instance at a time
    - Bound to specific AZ
      * Must be attached to EC2 instance in same AZ
    - Like a shared drive
    - Commo via network
    - Can be detached and attached to diff EC2 instance
    - To move to diff AZ must create snapshot first
    - Delete on termination attribute
      * Deletes the volume when EC2 instance is terminated
      * Root vol enabled by default
  + Snapshots
    - Do not need to detach volume ( but recommended )
    - Snapshot can be copied across regions
    - Amazon Data lifecycle manager
      * Creation, retention and deletion of EBS snapshots and EBS Backed AMI
      * Schedule backups, cross-account snapshots copies delete outdated backups
      * Can use tags to select resources
    - Fast snapshot Restore (FSR)
      * Snapshots stored in S3
      * Helps create vol from snapshot that is fully initialized at creation
      * No IO latency
      * Billed per minute ( expensive )
      * Can be enabled via DLM
  + Instance Store
    - High performance
    - Better I/O performance
    - Ephemeral
      * Not persistent
      * Loose data when instance is stopped
    - Buffer, cache, scratch data / temp content
    - Backups and replication up to you
  + EBS Volume Types
    - Types
      * Gp2 / Gp3 (SSD)
        + General purpose
        + SSD
        + Balanced price and performance
        + 1 GB to 16 TB
        + GP3

3000 IOPS – increase to 16000

125 MiB/s – increase to 1000 MiB/s

Independently set IOPS and throughput

* + - * + GP2

Small vol burst to 3000 IOPS

Size of vol and IOPS are linked max 16000 IOPs

* + - * Io1 / Io2 ( SSD )
        + Highest Performance SSD
        + Mission critical
        + Low latency
        + High throughput
        + Provisioned IOPS
        + Critical business apps with sustained IOPs
        + Apps that need more than 16000 IOPS
        + 4GB – 16TB
        + Max IOPS 64000 (Nitro EC2 instances)

32000 for other instances

* + - * + Can increase IOPS independent of size
        + IO2 have more durability and mor IOPS per GB than IO1 at same price
        + Support EBS multi attach

Io1 and io2

Attach same EBS vol to multiple EC2 in same AZ

With full read/write

Clusted use cases

App must manage concurrent writes

* + - * St1 (HDD)
        + Low cost HDD
        + Designed for frequently access, throughput-intensive workloads
        + Cannot be used as boot volume
        + 125 MB – 16 TB
      * Sc1 (HDD )
        + Lowest Cost HDD
        + Les frequently accessed
        + Cannot be used as boot volume
        + Cold HDD
    - Characterized by
      * Size
      * Throughput
      * IOPS
  + EBS Volume Resizing
    - Can increase size and IOPs (IO drives only)
    - Must repartition drive after resizing
    - Volume may go into long optimization phase (Defrag) but is still usable
    - Decrease vol
      * Must create new vol of smaller size
      * Attach and migrate data
      * Remove old bigger drive
  + EBS Migration
    - Locked to specific AZ
    - To move to another AZ / region
      * Snapshot volume
      * Copy to diff region if need be
      * Create a volume from the snapshot in the AZ of your choice
  + EBS Encryption
    - By default when create EBS Vol
      * Data at rest is encrypted
      * All data in flight moving between the instance tnd the volume is encrypted
      * All snapshots are encrypted
      * All volumes created from the snapshot are encrypted
    - Encryption and decryption handled transparently
    - Encryption minimal affect on latency
    - Keys from KMS
    - Copying an unencrypted snapshot allows encryption
    - Enable encryption
      * Create snapshot
      * Encrypt using copy
      * Create new vol from snapshot
      * Attach new vol
    - Can use standard RAID if OS supports
      * Software RAID
      * Types of RAID
        + RAID 0 – Striped

Better I/O Performance

* + - * + RAID 1 – Mirror
        + RAID 5 – Stripe with Parity

Not recommended on EBS as parity write takes up IOPS

* + - * + RAID 6 – Double Parity

Adds another parity bit

Not recommended on EBS

* + Troubleshooting
    - Hangs at attaching
      * Usually indicates device name is already being used
      * Try reattaching to a different device name

#### Amazon Elastic File System (Amazon EFS)

* + <https://tutorialsdojo.com/amazon-efs/>
  + Overview
    - Managed NFS
    - Multi-AZ
    - Multi instance can access
    - Highly available
    - Scalable
    - Expensive
      * Pay per use
    - Like mounting NFS (network share)
    - Use cases
      * Content management
      * Web serving
      * Data sharing
      * Wordpress
    - Uses NFS 4.1
    - Use security groups for access
    - Only works with linux no windows
    - Encrypt at rest use KMS
    - Scale automatically
      * 1000 of connections
      * 10 GB plus throughput
      * Can grow to peta byte
    - Performance Mode
      * Set at EFS vol Creation time
      * Types
        + General Purpose
        + Max I/O

less latency, throughput, highly parallel

* + - * Throughput Mode
        + Bursting

1 TB = 50 MB/S burst up to 100 MB/s

* + - * + Provisioned

Set your throughput regardless of storage size

* + - * + Storage Tiers (lifecycle management feature)

Standard

Frequently accessed files

Infrequent Access (EFS-IA)

Cost to retrieve but less to store

* + EFS Access Points
    - Feature to manage application access to NFS environments
    - Enforce POSIX user and group to use when access the file system
    - Allows you to restrict access to directory within file system
    - Specify different root directory
    - Can restrict access from NFS Clients using IAM policy
  + EFS Operations
    - In place operations ( when EFS is attached and in use)
      * Lifecycle policy ( enable IA or change settings )
      * Througput Mode
      * Provisioned Througput Numbers
      * EFS Access Points
    - Operations that require migration using datasync
      * Migration to encrypted EFS
      * Performance Mode
  + EFS Cloudwatch Metrics
    - PercentIOLimit
      * How close volume is to IO limit
      * If near or at 100% migrate to MaxIO
    - BurstCreditBalance
      * The number of burst credits the file system can use to achieve higher throughput levels
      * Remember if no credits then cannot burst
    - StorageBytes
      * Size in bytes
      * Dimensions
        + Standard
        + IA
        + Total (standard+IA)

#### Amazon FSx

* + [What is FSx for Windows File Server? - Amazon FSx for Windows File Server](https://docs.aws.amazon.com/fsx/latest/WindowsGuide/what-is.html)
  + Launch third party high performance file systems in AWS
  + Fully manged service
    - FSx for Lustre
      * Parallel distributed file system for large scale computing
      * Linux and Cluster = Lustre
      * Machine Learning
      * High Performance Computing (HPC)
      * Scales large lots of IOPs
      * Seamless integration with S#
        + Can read S3 as a file system through FSx
        + Can write the output of computations back to S3 through FSX
      * Can be used on-prem
      * Types of FSx File System Deployment
        + Scratch File System

Temp storage

Data not replicated

High burst

Use

Short term processing

Optimize cost by not replicating

* + - * + Persistent File System

Long term storage

Data is replicated in same AZ

Use

Long term processing

Storage of sensitive data

* + - FSx for Windows File Server
      * EFS for windows
      * Supports SMB and NTFS
      * AD integration
      * ACLs
      * User quotas
      * Built on SSD scale large and lots of IOPs
      * Can be access from On-Prem
      * Data backed up to S3
      * Deployments
        + Single AZ

Replicated within AZ

Single AZ 1 = SSD

Single AZ 2 = SSD and HDD

* + - * + Multi AZ

Sync replicated between AZ

Standby file server in diff AZ ( auto Failover )

* + - FSx for NetApp ONTAP

#### Amazon S3

* + <https://tutorialsdojo.com/amazon-s3/>
  + Buckets
    - S3 buckets must be named unique within entire AWS – globally unique
    - Object storage
    - Fast, durable, highly available, key based
    - Not a file system
    - Defined at the region level
    - Naming convention
      * No uppercase
      * No underscore
      * 3-63 chars long
      * Not an IP
      * Start with letter or number
  + Objects
    - Stored in buckets
    - Key
      * Full path to object
      * Composed of
        + Prefix

Path excluding bucket name and object name

* + - * + Object name

Object name

* + - Value
      * Content of the body
      * Max object size is 5TB
      * Can only upload 5 GB at time
        + Multi-part upload is when you upload multiple 5GB chunks to make up a large file
    - Metadata
      * List of key/value pairs
      * System or user metadata
    - Tags
      * Useful for lifecycle and security
    - Version ID
      * <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html#MultiFactorAuthenticationDelete>
      * Need to be enabled at the bucket level
      * Upload / save object with same name it will create new version
      * Protects against unintended deletes
      * Easy roll back to previous version
      * Notes
        + Any file that is not versioned prior to enabling versioning will have aversion of null
        + Suspending versioning does not delete the previous versions
      * Deleting older version or deleting delete marker is a permanent delete
      * Cannot remove versioning but can suspend versioning
        + Only bucket owner can suspend version
        + Existing versions do not change only future objects changes
  + Object Locks
    - [Using S3 Object Lock - Amazon Simple Storage Service](https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock.html)
    - Uses WORM
      * Write Once Read Many
    - Prevent objects from being deleted or overwritten for fixed amount of time or indefinitely
    - Retention
      * Retention period
        + Fixed period fo time during which object is locked
        + Cannot be overwritten or deleted
      * Legal Hold
        + No expiration date
    - Bucket must be enabled for versioning
    - Retention period and legal hold may be different per object version
  + Encryption for Objects
    - 4 methods to encrypt
      * SSE-S3
        + Encrypts S3 objects using keys handled and managed by AWS
        + Server side encryption (SSE)
        + AES 256
        + Must set header

“X-amz-server-side-encryption”:”AES256”

* + - * SSE-KMS
        + Leverage AWS Key Management Service to manage encryption keys
        + Advantages

User control

Audit trail

* + - * + KMS is AWS service that you set up
        + Must set header

“x-amz-server-side-ecryption”:”aws:kms”

* + - * SSE-C
        + You manage your own encryption keys outside of AWS
        + Key is not stored in S3 once object is encrypted
        + Must use HTTPS
        + Encryption key must be provide in HTTP headers for every request
        + Requires more management on your end ( manage keys )
      * Client Side Encryption (CSE)
        + Encrypt object before uploading to S3
        + Client library can assist

Such as AWS S3 Encryption Client

* + - * + Must decrypt data when receiving from S3
        + Client fully manages encryption cycle
    - Default Encryption vs Bucket Policies
      * Bucket policy checks headers and deny’s request if the header is not there.
      * Default encryption
        + Encrypts object if it is not encrypted during upload
        + Bucket policies are evaluated before default encryption
  + Batch Operations
    - [New – Amazon S3 Batch Operations | AWS News Blog](https://aws.amazon.com/blogs/aws/new-amazon-s3-batch-operations/)
    - Process s3 objects
      * Copy to another bucket
      * Set tags
      * Set ACL
      * Initiate a restore from Glacier
      * Invoke lambda functions on each object
      * Encrypt / decrypt existing files in the bucket
  + Encryption in transit
    - SSL/TLS
    - S3 exposes
      * HTTP endpoint – non encrypted
      * HTTPS endpoint – encryption in flight
    - HTTPS recommended
    - Most clients use HTTPS by default
  + Security and bucket Policies
    - IAM principal can access an S3 object if
      * IAM Permissions allow it (IAM Policies)
      * OR the resource policy allows it
      * AND no explicit DENY
    - User based
      * IAM policies
        + Which API calls should be allowed for specific user from IAM console
    - Resource Based
      * Bucket Policies
        + Bucket wide rules from the S3 console
        + Allows cross account
        + What principals can / cannot do on the bucket
        + JSON based

Resource

Bucket or objects

Action

Set of API to Allow or Deny

Effect

Allow / Deny

Principal

The account or user to apply policy to

* + - * + Use Cases

Grant public access to the bucket

Force objects to be encrypted at upload

Grant access to another account

Cross account

* + - * + Optional Conditions

Public IP or Elastic IP ( Not Private IP )

Source VPC or Source VPC Endpoint

CloudFrond Origin Identity

MFA

* + - * Object Access Control List ( ACL ) – finer grained
      * Bucket Access Control List ( ACL ) – Less common
    - Block Public Access
      * Prevent company data leaks
      * Granted through
        + NEW ACL
        + ANY ACL
        + NEW public bucket or access point policies
      * Block public and cross-account access to buckets and objects through any public bucket or access point policies
      * Can be set at the account level
    - Other S3 Security
      * Networking
        + Can access S3 buckets from VPC Endpoints (VPC without internet)
      * Logging and Audit
        + S3 Access logs can be stored in other S3 buckets
        + API calls can be logged in AWS CloudTrail
      * User Security
        + MFA Delete

Can be required to delete versioned bucket objects

* + - * + PreSigned URLs

URLs that are valid only for a limited time

* + CORS
    - Cross-Origin Resource Sharing
      * Get resources from different origin
      * Defines a way for client web applications that are loaded in one domain to interact with resources in another different domain
    - Origin
      * Scheme (Protocol)
      * Host (Domain)
      * Port
      * Example
        + <https://www.example.com>

Scheme = https

Domain = [www.example.com](http://www.example.com)

Port = 443 (implied with https)

* + - Web Browser based mechanisms
      * Allow requests to other origins while visiting the main origin only if the resource (other origin ) allows it
      * CORS request won’t be fulfilled unless the other origin allows for the request using CORS headers
        + Ex: Access-Control-Allow-Origin
    - S3 CORS
      * Specify CORS Headers
      * Allow for specific origin or \* all
      * CORS enabled on cross Origin bucket
  + Consistency Model
    - Strong consistency as of DEC 2020
    - After
      * Successful write of new object ( PUT )
      * OR an overwrite or delete of an existing object ( overwrite PUT or DELETE )
    - ANY
      * Subsequent read request immediately receives the latest version of the object (Read after Write Consistency)
      * Subsequent list request immediately reflect changes (List consistency)
  + S3 MFA Delete
    - <https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingMFADelete.html>
    - Forces user to generate code on device before doing important operations on S3
    - Must enable Versioning
    - Need MFA to
      * Permanently delete an object version
      * Suspend versioning on the bucket
    - Don’t need it for
      * Enabling versioning
      * Listing deleted versions
    - Can only be enabled / disabled by bucket owner (root account)
    - MFA Delete can only be enabled via the CLI
    - Delete from CLI
      * Requires the x-amz-mfa request header to permanently delete an object version or change versioning state of the bucket
      * Headers value
        + Auth devices serial number space then code displayed
  + S3 Access Logs
    - Not enabled by default
    - Logging from source saved to target bucket
    - Any access allowed or denied logged to S3 bucket
    - Can by analyzed with AWS Athena
    - Do not set logging bucket to be the monitored bucket
      * Logging loop and grow
  + S3 Replication
    - Replication requires Versioning set up in both buckets
    - Buckets can be in different accounts
    - Asynch copy
    - Must have IAM permissions in each S3 bucket
    - Types
      * Cross Region Replication
        + Use

Compliance

Lower latency

Replication across accounts

* + - * Same Region Replication
        + Use

Log aggregation

Live replication between production and test accounts

* + - Only new objects are replicated
    - For Delete
      * Can replicate delete markers from source to target
      * Deletions with a version ID are not replicated ( to avoid malicious deletes)
    - No chaining of replication
  + S3 Pre-Signed URLs
    - Can generate with SDK or CLI
    - CLI easy of download
    - SDK must be used for uploads
    - Valid for 3600 seconds by default
    - Users given pre-signed URL are given permissions of person who generated the URL for GET/PUT
    - Uses
      * Allow only logged-in users to download a premium video from S3 Bucket
      * Allow ever changing list of users to download files by generating URLs dynamically
      * Allow user to upload a file with temp permissions
  + S3 Inventory
    - <https://docs.aws.amazon.com/AmazonS3/latest/userguide/storage-inventory.html>
    - List all objects including metadata
    - Use
      * Audit and report on the replication and encryption status of your objects
      * Get number of objects in S3 buckets
      * Identify total storage of previous object versions
    - Output
      * CSV, ORC, APACHE Parquet
    - Can query all data using Athena, Redshift, etc
    - Can filter generated report using S3 Select
  + S3 Storage Classes
    - S3 Standard - General Purpose
      * 11 9s of durability
      * 99.99 availability over year
      * 2 concurrent facility failure
      * Uses
        + Big Data analytics
        + Mobile and gaming applications
        + Content distribution
    - S3 Standard – Infrequent Access (IA)
      * Less frequently accessed but requires rapid access
      * Same durability and availability as GP
      * Lower cost because less using
      * Use cases:
        + Data store for DR
        + Backups
        + Or and infrequently used data
    - S3 One Zone – Infrequent Access
      * Same as IA but only stored in single AZ
      * Same durability but less availability
      * Lower cost by 20%
      * Use Case
        + Secondary backup copies of on prem data
        + Any data that we can recreate
    - S3 Intelligent Tiering
      * Same low latency and throughput as standard
      * Small monthly monitoring and auto tiering fee
      * Auto moves objects between two access tiers based on changing access patterns
    - Amazon S3 Glacier
      * Archive/backup
      * Low cost
      * Stored for long time ( minimum 90 days )
      * Alternative to on prem magnetic tape
      * Cost per storage really low
      * Each object is called an archive stored in vaults
      * All data encrypted at rest using AES-256 keys managed by AWS
      * Vault Operations
        + Create and Delete

Delete only when there’s no archives in it

* + - * + Retrieve Metadata
        + Download Inventory
      * Glacier Operations
        + Upload
        + Download

After file retrieval

* + - * + Delete
      * Restore links have expiration date
      * 3 retrieval Options
        + Expidited

1 – 5 minutes

* + - * + Standard

3 – 5 hours

* + - * + Bulk

5 – 12 hours

* + - * Each Vault has (written in JSON)
        + Vault access policy

Like S3 bucket policy

* + - * + Vault lock policy

Lock for regulatory and compliance

File cannot be changed

Uses

Forbid deleting an archive if less than 1 year

Implement WORM Policy ( Write Once Read Many)

Two steps to enable

Generate Vault Policy ID and evaluate

Assign ID to vault

* + - * Notifications for restore operations
        + Vault Notification Configuration

Configure vault to send notification to SNS when job complete

Optionally when job is initiated

* + - * + S3 Event notification

Supports the restoration of objects archived to S3 Glacier storage class

S3:ObjectRestore:Post = Notify when object restoration initiated

S3:ObjectRestore:Completed = Notify when object restoration has completed

* + - Glacier Deep Archive
      * Cheaper still
      * Super long term storage
      * Retrieval options
        + Standard

12 hours

* + - * + Bulk

48 hours

* + - * Minimum storage duration of 180 days
  + S3 Lifecycle Rules
    - Moves data from one class to another automatically
    - Transition actions
      * Defines when objects are transitioned to another storage class
    - Expiration actions
      * Delete objects after time
      * Can be used to delete old versions
      * Can clean up incomplete multi-part uploads
    - Rules can be created for certain prefix ( folder, etc)
    - Rules for specific tags
  + S3 Analytics
    - Can be set up to determine when to transition objects from standard to Standard – IA
    - Does not work for onezone-ia or glacier
    - Updated daily
    - Helps build lifecycle rules ( or improve)
  + S3 Performance
    - Auto scales to high request rates, latench 100-200 ms
    - 3500 PUT/COPY/POST/DELETE and 5500 Get/HEAD requests per prefix in bucket
    - No limit to number of prefixes in a bucket
    - Prefix = Object path (like folder structure)
    - KMS Limitation
      * When you upload GenerateDataKey KMS API
      * Download calls Decrypt KMS API
      * Count towards KMS Quota per second depending on region
    - Optimize
      * Multi-part upload
        + Recommend for files > 100MB
        + Must use for files > 5GB
        + Help parallel uploads speed
        + Max parts 10000
        + Failures will only retry what is failed
        + Lifecycle to clean up partial uploads
      * S3 Transfer Acceleration
        + Increase transfer speed by transferrig file to AWS edge location which will forward data to S3 bucket
      * S3 Byte-Range Fetches
        + Parallelize GETs by requesting specific byte ranges
        + Better resilience in case of faliures
        + Can be used to speed up downloads
        + Can be used to only retrieve partial data
  + S3 Select and Glacier Select
    - Retrieve less data using SQL by performing server side filtering
    - Can filter by rows and colums
    - Less network so less cost
  + S3 Event Notifications
    - Rules on event (can be filtered) can be used to trigger other services
    - S3 Versioning will make sure all events notifications are sent. Without versioning multiple simultaneous evens may not be sent
  + Athena
    - <https://tutorialsdojo.com/amazon-athena/>
    - Serverless query service to perform analytics against S3 objects
    - Uses SQL (Presto open source sql)
    - Supports
      * CSV
      * JSON
      * Avro
      * Parquet
    - Compressed or columnar data can be cost savings
    - Use
      * Business Intelligence
      * Analytics
      * Reporting, analyze and query VPC Flow logs, ELB logs, CLoudwatch, etc
    - Partitioning
      * Partitioning data can restrict amount of data scanned by athena
      * Can partition on any key
    - Queries
      * Can query geospatial data
      * Query diff kinds of logs
      * Querys stored on S3 for 45 days
    - Security
      * Control access
        + IAM Policies
        + ACL
        + S3 Bucket Policies
  + S3 Access Points
    - Defined outside of buckets
    - Mapped to prefix in bucket
    - Removes policies from bucket to access points
    - Each access point gets its own DNS and policy to limit who can access it
    - Can restrict traffic to access point from specific VPC ( private S3 to VPC )
  + VPC EndPoint
    - By default EC2 instances will connect to S3 via internet
    - VPC Endpoint Gateway
      * Private connection to S3 bucket
    - Allows you to create bucket policy to allow onl y traffic from VPC Endpoint. Keeps it private
  + S3 Batch Operations
    - Bulk operations on existing S3 objects with single request
    - Job
      * List of object
      * Action to perform
      * Optional parameters
    - Batch operations manage
      * Retries
      * Tracks progress
      * Sends notifications
      * Generates reports
      * Etc

#### AWS Backup

#### AWS Storage Gateway

* + <https://tutorialsdojo.com/aws-storage-gateway/>
  + Helps with hybrid clouds
  + Bridge between on-prem and AWS proprietary storage S3
  + Uses
    - DR
    - Backup and restore
    - Tiered storage
  + Types of Storage Gateways
    - File Gateway
      * Security using IAM Roles
      * Most recently used data is cached on the file gateway
      * Mounted on many servers on prem
      * Integrates with on prem AD
      * Types
        + S3 File Gateway

Configured S3 buckets are accessible using NFS and SMB

* + - * + FSx File Gateway

Low latency access to FSx for Windows File Server file shares

Native access to FSx for windows

Local cache

SMB, NTFS, AD…

Group file shares / home directories, etc

* + - * File gateway is a linux machine
        + Virtual Software appliance

VM

Installed On-Prem

* + - * Reboot
        + Simply restart VM
    - Volume Gateway
      * Block storage ISCSI
      * Backed by EBS snapshots which can help restore on-prem voluems
      * Types
        + Cached Volumes

Low latency access to most recent data

Only most recent data stored locally

Look at cache efficiency

CacheHitPercent metric ( want high )

CachePercentUsed

Should not too high

Can create a larger cache disk ( steps on the Volume Gateway VM )

Clone volume to larger disk

Select new disk as the cached volume

* + - * + Stored Volume

Entire dataset is on prem with scheduled backups to S3

* + - * Runs as VM on prem
      * Reboot
        + Stop Storage Gateway Service
        + Reboot
        + Start the Storage Gateway Service
    - Tape Gateway
      * Virtual Tape library backed by S3 and Glacier
      * Uses existing tape based backup processes
      * ISCSI interface
      * Can be on prem VM or hardware appliance or EC2 instance
      * Reboot
        + Stop Storage Gateway Service
        + Reboot
        + Start the Storage Gateway Service
  + Usually installed on server
  + Option for Hardware Appliance
  + Activate Gateway
    - Get Activation Key
      * Using Gateway VM CLI
      * Make a web request to the Gateway VM ( port 80 )
        + Older way
    - Troubleshooting activation failures
      * Port 80 open
      * Gateway VM must have correct time with NTP server

#### AWS Beanstalk

* PaaS
* Dev centric view of deploying an application on AWS
* Free but pay for underlying resources
* Managed service
  + Instance config / OS config
  + Handles infrastructure
  + Deployment strategy
* Architecture models
  + Single instance deployment
    - Good for dev
  + LB + ASG
    - Production or pre production
  + ASG
    - Non web apps in production
    - No LB
* Three components
  + Application
    - Collection of Beanstalk components ( configs )
  + Application Version
    - Iteration of your application code
  + Environment Name
    - Collection of AWS resources running an application version
* Deploy application versions to environment
* Can promote application versions to next environment
* Rollback to previous app version
* Full control over lifecycle
* Support for many platfoorms
* Automate replacement of EC2 instances when unhealthy
  + Default health checks set to EC2 instances
  + For ELB to auto replace set to ELB
  + Change within config file

## Out-of-scope AWS services and features

The following is a non-exhaustive list of AWS services and features that are not covered on the exam.

These services and features do not represent every AWS offering that is excluded from the exam content.

Services or features that are entirely unrelated to the target job roles for the exam are excluded from this

list because they are assumed to be irrelevant.

Out-of-scope AWS services and features include the following:

* Amazon API Gateway
* Amazon AppStream 2.0
* AWS Batch
* Amazon Chime
* Amazon Cloud Directory
* Amazon CloudSearch
* AWS CodeBuild
* AWS CodeCommit
* AWS CodeDeploy
* AWS CodeStar
* Amazon Connect
* AWS Deep Learning AMIs (DLAMI)
* AWS Device Farm
* Amazon DynamoDB
* Amazon DynamoDB Accelerator (DAX)
* Amazon Elastic Container Registry (Amazon ECR)
* Amazon Elastic Container Service (Amazon ECS)
* Amazon Elastic Transcoder
* Amazon EMR
* Amazon GameLift
* AWS IoT Button
* AWS IoT Greengrass
* AWS IoT Platform
* Amazon Kinesis
* Amazon Lex
* Amazon Lightsail
* Amazon Lumberyard
* Amazon Machine Learning (Amazon ML)
* Version 2.1 SOA-C02 10 | PAGE
* AWS Managed Services
* AWS Mobile Hub
* AWS Mobile SDK
* Apache MXNet on AWS
* Amazon Pinpoint
* Amazon Polly
* Amazon Redshift
* Amazon Rekognition
* AWS Schema Conversion Tool
* Amazon Simple Email Service (Amazon SES)
* AWS Snowmobile
* Amazon WorkDocs
* Amazon WorkMail
* Amazon WorkSpaces
* AWS X-Ray
  + Log analysis
  + Trace
  + Visual analysis of app

# Sample lab from Udemy course

A company is deploying a new web application. Configure a highly available MySQL 8.0 database with the following

1. Create custom DB parametergroup and set the event\_scheduler parameter to true and use this parameter during DBinstance creation
2. Create a custom AWS Key Management Service (AWS KMS) key and use this key during DBInstance creationg
3. Create a VPC security group that allows TCP port 3306 from the CIDR block 192.168.1.0/24. Use this security group during DBinstance creation
4. Launch the Amazon RDS DBinstance
5. After launch, take a manual RDS DB snapshot