Monitoring & Reporting

EC2

Default Cloudwatch metrics EC2:

- CPU
- Disk
- Network
- Status Check

Standard monitoring is 5 minutes

Detailed monitoring is minimum granularity of 1 minute

Create EC2 role => Access to Cloudwatch

Custom metrics (Memory)

```
sudo apt-get update
sudo apt-get install unzip
sudo apt-get install libwww-perl libdatetime-perl

curl https://aws-
cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip
-O

unzip CloudWatchMonitoringScripts-1.2.2.zip && \
rm CloudWatchMonitoringScripts-1.2.2.zip && \
cd aws-scripts-mon

/home/admin/aws-scripts-mon# ls
awscreds.template    LICENSE.txt               NOTICE.txt
AwsSignatureV4.pm    mon-get-instance-stats.pl
CloudWatchClient.pm  mon-put-instance-data.pl

echo "*/1 *   * * *   root    /home/admin/aws-scripts-mon/mon-put-instance-
data.pl --mem-util --mem-used --mem-avail" >> /etc/crontab
```

Go to Cloudwatch => Metrics => Linux System => InstanceId

EBS

Compare Volume types

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	**General Purpose SSD (gp2)***	**Provisioned IOPS SSD (io1)**	**Throughput Optimized HDD (st1)**	**Cold HDD (sc1)**
Use Cases	**Recommended for most workloads** **System boot volumes** **Virtual desktops** **Low-latency interactive apps** **Development and test environments**	**Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume** **Large database workloads, such as: •MongoDB •Cassandra •Microsoft SQL**	**Streaming workloads requiring consistent, fast throughput at a low price** **Big data Data warehouses Log processing** **Cannot be a boot volume**	**Throughput-oriented storage for large volumes of data that is infrequently accessed** **Scenarios where the lowest storage cost is important** **Cannot be a boot volume**

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of transactional workloads	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/Volume	10,000	20,000	500	250
Max. Throughput/ Volume†	160 MiB/s	320 MiB/s	500 MiB/s	250 MiB/s
Max. IOPS/Instance	65,000	65,000	65,000	65,000
Max. Throughput/ Instance	1,250 MiB/s	1,250 MiB/s	1,250 MiB/s	1,250 MiB/s
Dominant Performance	IOPS	IOPS	MiB/s	MiB/s

IOPS

gp2:

- 3 IOPS/Gb
- Burst up to 3000 IOPS
- I/O credits
- Burst up to 2997 IOPS when using 1 Gb Volume
- max of 10 000 IOPS (more => use Provisioned IOPS)

- Burn all I/O credits if burst for 30 minutes

If you creat a volume from snapshot from s3 => use pre-warming ebs volume for maximum performance => What this basically means is just read every data block which has data on your ebs volume before using it.

Metrics

Metric	Description
VolumeReadBytes	Provides information on the read operations in a specified period of time. The Sum statistic reports the total number of bytes transferred during the period. The Average statistic reports the average size of each read operation during the period, except on volumes attached to a Nitro-based instance, where the average represents the average over the specified period. The SampleCount statistic reports the total number of read operations during the period, except on volumes attached to a Nitro-based instance, where the sample count represents the number of data points used in the statistical calculation. For Xen instances, data is reported only when there is read activity on the volume. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Bytes

Metric	Description
VolumeWriteBytes	Provides information on the write operations in a specified period of time. The Sum statistic reports the total number of bytes transferred during the period. The Average statistic reports the average size of each write operation during the period, except on volumes attached to a Nitro-based instance, where the average represents the average over the specified period. The SampleCount statistic reports the total number of write operations during the period, except on volumes attached to a Nitro-based instance, where the sample count represents the number of data points used in the statistical calculation. For Xen instances, data is reported only when there is write activity on the volume. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Bytes
VolumeReadOps !	The total number of read operations in a specified period of time. To calculate the average read operations per second (read IOPS) for the period, divide the total read operations in the period by the number of seconds in that period. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Count

Metric	Description
VolumeWriteOps !	The total number of write operations in a specified period of time. To calculate the average write operations per second (write IOPS) for the period, divide the total write operations in the period by the number of seconds in that period. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Count
VolumeTotalReadTime	Note This metric is not supported with Multi-Attach enabled volumes. The total number of seconds spent by all read operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds. For Xen instances, data is reported only when there is read activity on the volume. The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Seconds

Metric	Description
VolumeTotalWriteTime	Note This metric is not supported with Multi-Attach enabled volumes. The total number of seconds spent by all write operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds. For Xen instances, data is reported only when there is write activity on the volume. The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Seconds
VolumeIdleTime	Note This metric is not supported with Multi-Attach enabled volumes. The total number of seconds in a specified period of time when no read or write operations were submitted. The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Seconds

Metric	Description
VolumeQueueLength !	The number of read and write operation requests waiting to be completed in a specified period of time. The Sum statistic on this metric is not relevant for volumes attached to Nitro-based instances. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Count
VolumeThroughputPercentage	Note This metric is not supported with Multi-Attach enabled volumes. Used with Provisioned IOPS SSD volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS SSD volumes deliver their provisioned performance 99.9 percent of the time. During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (for example, creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, or accessing data on the volume for the first time). Units: Percent
VolumeConsumedReadWriteOps	Used with Provisioned IOPS SSD volumes only. The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time. I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS. Units: Count

Metric	Description
BurstBalance	Used with General Purpose SSD (gp2), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes only. Provides information about the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket. Data is reported to CloudWatch only when the volume is active. If the volume is not attached, no data is reported. The Sum statistic on this metric is not relevant for volumes attached to Nitro-based instances. If the baseline performance of the volume exceeds the maximum burst performance, credits are never spent. The reported burst balance is either 0% (Nitro-based instances) or 100% (non-Nitro-based instances). For more information, see I/O Credits and burst performance. Units: Percent

Status

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	Insufficient Data

EBS modifications

You can adjust the volume type, size and IOPS on the fly. If you adjust the volume size, you must manually expand the filesystem.

ELB

Cloudwatch monitors performance (Metrics)

Cloudtrail monitors API calls to AWS (Audit)

Cloudwatch monitoring is enabled by default when creating a loadbalancer.

Access logs are not enabled by default (Stored in S3). You can use Sumologic or AWS Athena to query these logs. Once EC2 instances have been deleted, there is no way to recover nginx access logs if you want to debug.

You can use Request tracing on an ALB. It adds or updates the X-Amzn-Trace-id header before sending it through.

ElasticCache

Standard monitoring:

- CPU utilization
- Swap usage
- Evictions

- Concurrent Connections

CPU

Memcached is multi-threaded and can handle loads of up to 90%. Add more nodes to the cluster when it exceeds 90%.

Redis is not multi-threaded. To determine the threshold in which to scale, take 90 and devide by the number of cores.

Swap

If you use 4 Gb of RAM, use 4 Gb of Swapfile

Memcached should have arount 0 swap and should not exceed 50Mb, If it does increase hte memcached_connections_overhead parameter (defines the amount of memory of reserved memcached connections and other miscellanous overhead). Increase the memory of your memcached.

With Redis no SwapUsage metric is shown, instead it uses reserved-memory

Evictions

Evictions => remove data when no new data can be stored.

Memcached => no recommended setting, either scale up or scale out.

Redis => only option the scale out by adding read replicas

Concurrent Connections

No recommended settings. If there is a spike, this is due a spike in traffic or your application is not releasing connections as it should (Set an alarm on the number of connections).

Multiple Regions & Custom Dashboard

Dashboards are internationally! You have to change region if you want a widget with metrics from that region.

Create a billing alarm

Cloudwatch => Create billing alarm => Select time check => Define threshold in USD => Select SNS create topic => Enter your e-mailadres => Confirm email-address.

AWS Organizations

Allows you to manage multiple AWS accounts:

- Centrally manage policies across multiple accounts (IAM groups)
- Control access to AWS services (Service control policy => Allow or Deny AWS services)
- Automate AWS account creation and management
- Consolidate Billing across multiple AWS accounts (good for discounts)

Create an organization inside helpful tips, 2 choices:

- Enable all features
- Enable only consolidated billing

Policies:

- Deny (a list of denied services)
- Allow (a list of allowed services)

AWS Rescource Groups & Tagging

Resource groups make it easy to group your resources using tags that are assigned to them. You can group resources that share one or more tags.

Resource groups contain information such as:

- Region
- Name
- Health checks

Specific information:

- EC2: Public and private IP
- ELB: Port configurations
- RDS: Database engine etc

Create resource group based on tags or cloudformation stack based. Add tags and give your group a name.

Via AWS System Manager your can execute automation on resource groups (f.e. Stop Ec2 instances).

Cost explorer & Cost Allocation Tags

You can create csv reports. In Cost allocation tags you can set the tags where you want to know the cost of in cost explorer. It is across multiple accounts.

AWS Config

Aws config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance.

- Configurations snapshots and logs config changes of AWS resources
- Automated compliance checking

Enable it per region. AWS config stores everything in S3 and could trigger a lambda or pulled by Lambda. As soon as a rule is being broken an SNS notification will be send to someone (email).

Terminology:

- Configuration Items: point in time attributes of resource
- Configuration Snapshots: Collection of Config Items
- Configuration Streams: Stream of changed Config Items
- Configuration History: Collection of Config Items for a resource over time
- Configuration Recorder: The configuration of Config that records and stores Config Items => Logs config for account in Region and Stores in S3, Notification through SNS

What we can see:

- Resource Type
- Resource ID
- Compliance
- Timeline:
    - Configuration Details

    - Relationships
    - Changes
    - Cloudtrail Events

Compliance checks:

- Trigger:
    - periodic
    - Configuration snapshot delivery
- Managed Rules:
    - About 40
    - Basic, but fundamental

AWS Config needs Read only permissions to the recorded resources, write access to S3 logging bucket and publish access to SNS.

Health Dashboards

(https://status.aws.amazon.com/)

Service Health Dashboard => Show the health of each AWS Service as a whole per region

Personal Health Dashboard => AWS Personal Health Dashbaord provides alerts and remediation guidance when AWS is experiencing events that may impact you.

Deployment & Provisioning

Deploy an EC2

Running on spot options:

- Add max price
- Persistent request
- Interruption behaviour:
    - Terminate
    - Stop (lose data on RAM)
    - Hibernate (Keep RAM)
- Request valid from/to
- Launch group (only launches when all can launch)

- Placement group: Add instances to one AZ
- Enable Termination protection
- Shutdown Behavior (stop/terminate)
- Enable detailed monitoring
- Tenancy (Shared - Run a shared hardware)
- T2 unlimited (Burst CPU)
- User data (bootscripts)

EC2 Launch Issues

Common issues:

- InstanceLimitExceeded error: Default limit 20 per region limit (AWS support to raise limit)
- InsufficientInstanceCapacity error: AWS does not have enough available on-demand capacity (wait few minutes / request fewer instance types / select other instance types / purchase reserved instances / Submit request without AZ)

EBS Volumes and IOPS

- gp2 : minimum 100 IOPS to max 16 000 IOPS
- io1 (provisioned iops => databases) : minimum 50 IOPS/Gb to max 64 000 IOPS

Hitting limit gp2 iops => I/O request queuing => Application becomes slow

- raise gp2 volume size
- already 16 000 iops => change to io1

Elatic Loadbalancers

- Application loadbalancer (layer 7)
- Network loadbalancer (layer 4 => Handles millions of request per second)
- Classic loadbalancer (X-forwarded and sticky sessions)

Pre-warming loadbalancer => contact AWs support to pre-warm to handle spikes

- start and enddate
- expected req/sec
- total size of typical request

ALB changes ip addresses when scaling

Network loadbalancers create static ip per subnet (good for firewalling)

Solution: Put an ALB behind a network loadbalancer for static ip

ELB Error Messages

Classic and ALB:

- 200 => success
- 4xx client side error
- 5xx server side error

Client side error:

- 400 => Bad/malformed request (header malformed)
- 401 => unauthorized
- 403 => Forbidden (blocked by WAF access control list)
- 460 => Client closed connection before loadbalancer could respond
- 463 => Loadbalancer received X-forwarded-For header with > 30 ips

Server side error:

- 500 => Internal server error (loadbalancer)
- 502 => Bad Gateway (application server closed connection)
- 503 => Service unavailable (no registered targets)
- 504 => Gateway timeout
- 561 => unauthorized (identity provider)

ELB Cloudwatch Metrics

Loadbalancer have default Cloudwatch metrics and also for the backends

- BackendConnectionErrors => number of unsuccessful connections to the backend instances
- HealthyHostCount
- UnHealthyHostCount
- HTTPCode_Backend_2xx,3xx,4xx,5xx
- Latency => number of second taken for instance to respond

- RequestCount => number of request completed
- SurgeQueueLength => number of pending requests - max of 1024 (Classic only)
- SpolloverCount => number of requests rejected when surge queue is full (Classic only)

AWS Systems Manager

- Management-tool which give you control over AWS infrastructure.
- Integrates with Cloudwatch allowing you view your dashboards, view operation data & detect problems.
- Includes Run command which automates operational tasks across resources - f.e. security patches, package installs.
- Organize your inventorym grouping resources together by application or environment (including on-premise)

Run-command

- Allow you to run pre-defined command on one or more EC2 instances.
- Stop, restart, terminate, resize instance
- Attach/detatch EBS volumes
- Create snapshots, backup DynamoDB tables
- Apply patches and updates
- Run an Ansible playbook
- Run shell scripts

Use

- Create role in IAM for EC2 (EC2RoleforSSM)
- Attach role to EC2
- In SSM, find resource group (create resource group)
    - Build-in insights
        - view Cloudtrail
        - AWS Config
        - Personal Health Dashboard
        - Trusted Advisor
            - Cost optimizations
            - Performance

        - Security recommendations
        - Fault Tolerance
        - Service Limits
    - Dashboards in Cloudwatch
    - Inventory (Top OS, Top Services, ...)
    - Compliance (see patches applied)
    - Automation (Run commands on instances, automated/in steps, create your own documents)
    - Run command (pre-configured or own scripts f.e AWS-RunShellScript document)
        - SNS notification
        - Output to S3
        - Shell script box
        - Apply shows you the output
    - Patch Manager
    - Maintenance Windows (cron scheduler, duration)
    - State Manager (ensure consistent state - reapply when state is no compliant)
    - Managed instances
    - Activations (Register EC2 instances and on-premise - Install ssm agent)
    - Documents (Create your own documents / view existing)
    - Parameter Store (secrets)

Placement Groups

By default AWS places instances across different physical hardware. This minimizes the impact of a hardware failure. Not so great for low latency, high network throughput applications.

Types:

- Cluster: Instances are all created in a single AZ
    - Full line rate of 10 Gbps
    - Not for high availability
- Partition: Instances are created in logical segments called partitions, each located in a separate rack, with independent network and power. Some instances could be in the same rack.
    - Great for HDFS, HBase, and Cassandra

- Spread: Each instance is created in a separate rack, with independent network and power.
    - Maximum availability

High availability

Elasticity (Short term) & Scalability (Long term)

EC2:

- Scalability: Increase instance size
- Elasticity: Increase number of instances

DynamoDB:

- Scalability: Unlimited amount of storage
- Elasticity: Increase additional IOPS for additional spikes in traffic, decrease when spike stops.

RDS:

- Scalability: Increase instance size
- Elasticity: not very elastic, can scale RDS base on demand

Aurora => Scalability: Modify instance type, Elasticity: Aurora Serverless

RDS and Multi-AZ Failover

Automatically failover to other AZ (which already replicated) for Disaster Recovery. Small outage of 1 minute can happen during failover. It update the private DNS for the database endpoint.

MySQL, MariaDB, Oracle and PostgreSQL engines utilize synchronous physical replication

SQL Server engine uses synchronous logical replication.

Advantages:

- Backups are taken from secondary which avoids I/O suspension to the primary
- Restore's are taken from secondary which avoids I/O suspension to the primary

! You can force a failover from one AZ to another by rebooting your instance. Via AWS console or by using RebootDBInstance API call.

RDS & Using Read Replicas

Create a Read Replica via the AWS console or by CreateDBInstanceReadReplica API. It will use engine native asynchronous replication.

Useful for Read-heavy databases to scale. Point your application to read endpoint of the RDS instance. (Business reporting or data warehousing)

For Bi solutions => create read replica or import data into Redshift.

Aurora uses SSD backend virtualized storage layer for database workloads. Aurora replicas share the same underlying storage as the source instance, lowering the costs and avoiding to need to copy data to the replica nodes.

Creating Read Replicas

- AWS takes a snapshot of the database
- If Multi-Az is not enabled, AWS will take the snapshot of your primary database. This will create a brief I/O suspension for around 1 minute
- If Multi-Az is enabled, AWS will take the snapshot from your secondary database.

Read Replicas can be promoted to its own database. Will break the replication.

You can also create a read replica of a read replica in a other region. Can have latency.

You can have up to 5 Read replicas for Mysql, Postgresql and Mariadb

DB snapshots and Automated Backups cannot be taken of read replicas.

Key Metric to look for is REPLICA LAG (How long does the replication take)

RDS & Using Read Replicas

- Storage autoscaling (start storage and threshold)
- Deletion protection
- You can not create a read replica if automated backups are turned off
- creating mutli-az could have potential downtime.

- You cannot create read replica of read replica when automated backups are turned off

```
aws rds describe-db-instances --region eu-west-1
```

RDS - Encryption RDS Snaps

- Take a snap of existing RDS instance
- Copy the snap to the same/different region. (enable encyption)
- Encrypt the copy during the copy process.
- Restore the snap

You can share DB encrypted snapshot using AES-256 encryption with other AWS accounts:

- Create a Custom KMS encryption key
- create RDS snap using custom key
- share the custom AWS KMS encryption key
- Use AWS Management console, cli or rds api to share the encrypted snapshot.

You cant share the following:

- encrypted snapshot as public
- Oracle or Microsoft SQL Server snapshot that are encrypted using Transparent Data Encryption (TDE).
- Snapshots encrypted using default AWS KMS encryption key.

Which Services have Maintenance windows

- RDS
- Elasticache
- Redshift
- DynamoDB DAX
- Neptune
- Amazon DocumentDB

Elasticache

Redis has Master/Slave replication and Multi-AZ, Memcached doesn't

If your databases is stressed out by read operation, use ElastiCache or Redshift for OLAP transactions.

Aurora

Supports:

- MySQL (5 times better performance)
- PostgreSQL (3 times better performance)

Specs:

- Starts with 10 GB storage and increments per 10 GB up to 64 TB
- Compute resources can scale up to 64 vCPUs and 488 GiB Memory
- 2 copies of your data in each AZ, with minimum of 3 AZ. 6 copies of you data.
- Can lose 2 copies of your data without effecting write availability
- Can lose 3 copies of your data without effecting read availability
- Self Healing are continuously scanned for errors
- Cluster volume across AZ's

Aurora Replica:

- Aurora replicas (up to 15)
- Mysql read replicas (up to 15)

100% of CPU utilization?

- Write causing issue: Scale up (increase instance size)
- Read causing issue: Scale out (increase number of read replicas)

Aurora Serverless: Automatically scale up, shuts down base on capacity. You pay per second basis for database capacity, and you can migrate between standard and serverless configuration with a few clicks in the Amazon RDS Management Console.

Lowest number tier will be used for the failover (failover priority in configuration).

You can set up a read replica cross-region. Be aware to turn on Multi-AZ for this. If the replication is disrupted, you have to set this up again.

Encryption at is rest is set by default.

Troubleshooting Autoscaling

- Associated Key Pair does not exist
- Security group does not exist
- Autoscaling config is not working correctly
- Autoscaling group not found
- Instance type specified is not supported in the AZ
- AZ is no longer supported
- Invalid EBS device mapping
- Autoscaling service is not enabled on your account (check IAM)
- Attempting to attach an EBS block device to an instance store AMI

Cloud Front & Cache Hit Ratios

- Edge location: location where the content will be cached
- Origin
- Distribution: name given to the CDN
- Web Distribution: For websites
- RTMP: Used for media streaming

Maximize cache hit ratios:

- Specify how long cloudfront caches object (Cache-Control max-age)
- Caching Based on Query String parameters
- Caching Based on Cookie Values
- Caching Based on Request Headers
- Remove Accept-Encoding Header when compression is not needed
- Serving Media content using http

Storage & Data Management

S3

- Files can be from 0 Bytes to 5 TB
- Unlimited storage
- https://s3-eu-west-1.amazonaws.com/{yourname}

Objects consist of:

- Key
- Value
- Version ID (Important when versioning is enabled)
- Metadata (Data about data you are storing)
- Subresources (Bucket specific configuration)
    - Bucket policies, Access control lists
    - CORS
    - Transfer acceleration

Charged for:

- Storage per GB
- Requests (Get, Put, Copy, etc.)
- Storage Management pricing:
    - Inventory, Analytics, and Object tags
- Data Management pricing:
    - Data transferred out of S3
- Transfer Acceleration
    - Use Cloudfront to optimize

Example Life Cycle policies

- Transition objects to IA storage class 90 days after you created them (f.e. logs)
- Archive objects to Glacier 1 year after creation
- Expire object 1 year after creating them (S3 will auto-delete) => Server access logging in s3 can accumulate many log files

MFA Delete

- When versioning on a bucket is enabled, a delete action doesn't delete the object version, but applies a delete marker.
- To permanently delete, provide object version id in the delete request

MFA delete provides and additional layer of protection to s3 Versioning. Once enabled, MFA Delete will enforce 2 things:

- You will need a valid code form your MFA device in orderr to permanently delete an object version

- MFA also needed to suspend / reactivate versioning on an S3 Bucket

S3 Encryption

Types of encryption:

- In Transit (SSL/TLS)
- At Rest:
  - Server Side Encryption
    - S3 Managed keys - SSE-S3 (master key encryption + rotate)
    - AWS Key Management Service, Managed Keys, SSE-KMS (envelope key)
    - Server Side Encryption with Customer Provided Keys - SSE-C (you own key and rotation)
  - Client Side Encryption (encrypt before upload)

Enforce Encryption:

- x-amz-server-side-encryption:AES256 (SSE-S3)
- x-amz-server-side-encryption:aws:kms (SSE-S3)

=> Use a bucket policy which denies PUT requests which doesn't include x-amz-server-side-encryption

The Expect: 100-continue Header in the PUT request is for accept of deny the body sent to S3

EC2 Volume Types

- Instance store is know as ephemeral storage which is non-persistent
- EBS is Elastic Block storage which allows persistence

Root device can be EBS or Instance Store:

- Instance store root has maximum size of 10 Gb
- EBS can be up to 1 or 2 Tb
- All Instance store volumes are removed on termination of EC2
- Instance store persist with reboot of EC2

Upgrading EC2 Volume Types

Create ec2 will create all ebs on it in the same AZ. Change size and volume type on the fly. Snapshots exists on S3 (not visible), there incremental. Snapshots needs to be encrypted to be shared with other AWS accounts.

Migrating EBS to another AZ:

- create snapshot of EBS
- create volume from snapshot

Migrate to other region:

- Copy snapshot (select region)
- Create Image of ebs snapshot
- Launch EC2