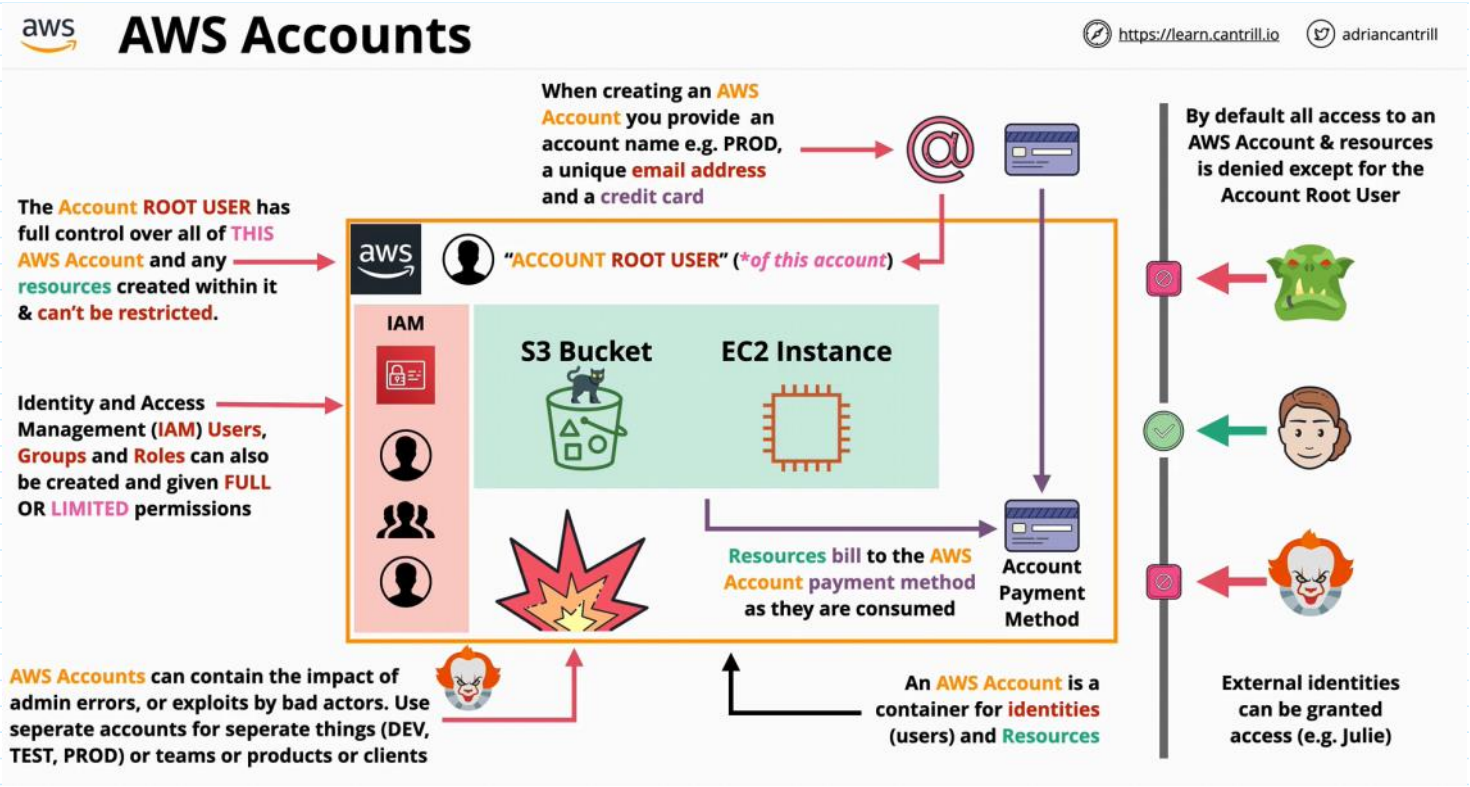
AWS Accounts

Thursday, June 3, 2021 1:16 PM

ROOT	**has full access in the AWS account**
IAM	**has no access at all initially** **access needs to be granted by the root user**



Management/General	dora.theodora89+AWS1@gmail.com	MDLT
Production	dora.theodora89+AWS1@gmail.com	MDLT

IAM identities start with **no permissions** on an AWS Account, but can be granted permissions (almost) up to those held by the Account Root User.
IAM also is trusted full by the account.

What is cloud computing

Friday, June 4, 2021 3:31 PM

1. Introduction

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

1.2 Purpose and Scope

Cloud computing is an evolving paradigm. The NIST definition characterizes important aspects of cloud computing and is intended to serve as a means for broad comparisons of cloud services and deployment strategies, and to provide a baseline for discussion from what is cloud computing to how to best use cloud computing. The service and deployment models defined form a simple taxonomy that is not intended to prescribe or constrain any particular method of deployment, service delivery, or business operation.

1.3 Audience

The intended audience of this document is system planners, program managers, technologists, and others adopting cloud computing as consumers or providers of cloud services.

1

2. The NIST Definition of Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Essential Characteristics:

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability[1] at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Service Models:

Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure[2]. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.[3] The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Deployment Models:

Private cloud. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

[1] Typically this is done on a pay-per-use or charge-per-use basis.

[2] A cloud infrastructure is the collection of hardware and software that enables the five essential characteristics of cloud computing. The cloud infrastructure can be viewed as containing both a physical layer and an abstraction layer. The physical layer consists of the hardware resources that are necessary to support the cloud services being provided, and typically includes server, storage and network components. The abstraction layer consists of the software deployed across the physical layer, which manifests the essential cloud characteristics. Conceptually the abstraction layer sits above the physical layer.

[3] This capability does not necessarily preclude the use of compatible programming languages, libraries, services, and tools from other sources.

On demand self-service: Cloud can provision capabilities as needed **without requiring human interaction.**
- virtual machines, storage databases, networking
- using a web page, terminat to access those
- no delay

Broad network access: Capabilities are available over the **network** and accessed through **standard mechanisms.**
- HTTP, HTTPS, SSH, Remote Desktop, VPN

Resource pooling:
- There is sense **of location independence**... no control or knowledge over the exact location of the resources
- Resources are **pooled** to serve multiple consumers using **a multi-tenant model**
  - pooling: isolates the customer: your data is only visible for you.

Rapid Elasticity:
- Capabilities can be **elastically provisions and** released to **scale** rapidly outward and inward with the demand
- To the consumer, the capabilities available for provisioning often **appear unlimited**.

Measured Service: Resource usage can be **monitored, controlled, reported** and **billed**.
- on demand billing
  - usage is monitored and **then** billed

1 On-Demand Self-Service	**Provision and Terminate using a UI/CLI without human interaction.**
2 Broad Network Access	**Access services over any networks, on any devices, using standard protocols and methods.**
3 Resource Pooling	**Economies of scale, cheaper service.**
4 Rapid Elasticity	**Scale UP (OUT) and DOWN (IN) automatically in response to system load**
5 Measured Service	**Usage is measured. Pay for what you consume.**

[3] This capability does not necessarily preclude the use of compatible programming languages, libraries, services, and tools from other sources.

Public & Private & Hybrid & Multi Cloud

Saturday, June 5, 2021 12:49 PM

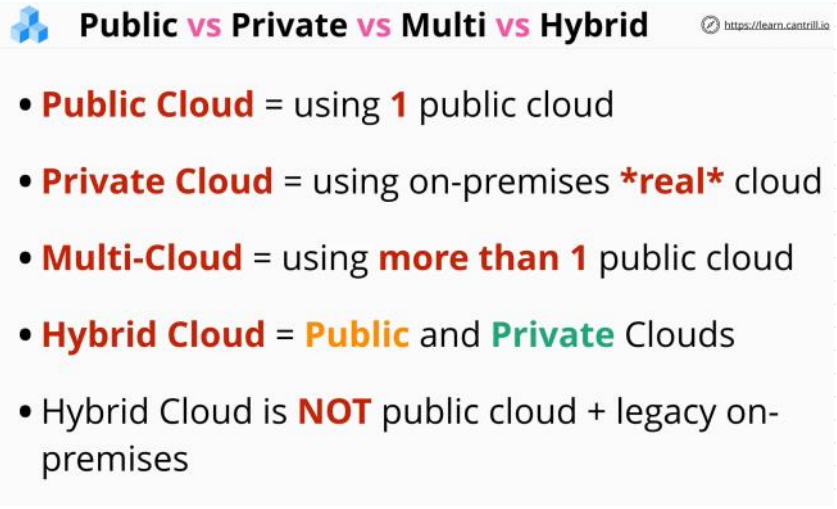
Public Cloud
- classify as a cloud
- available to the general public

Multiple-Cloud
- one app using multiple clouds (AWS, Azure, Google, etc)
- if a vendor fails, part of the system can still work
- multiple clouds can be used a **one** single environment
    - but in this situation they rely on the lowest common feature set
    - loosing what makes each vendor unique

Private Cloud
- meets all 5 characteristics of the cloud

Hybrid Cloud
- using **private** cloud in **conjuncture** with **public** cloud
- private cloud and public cloud working in a single environment
- using same tooling, interfaces and process to interact with both

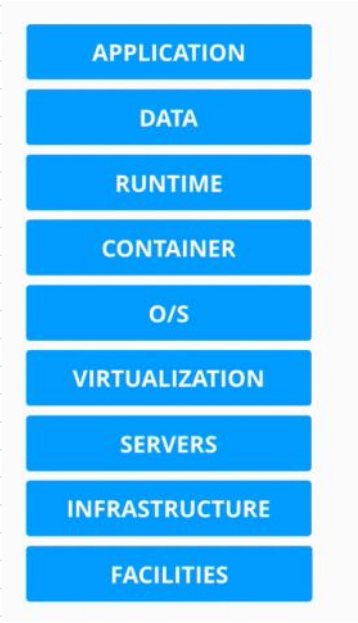
Hybrid environment
- using public cloud with a traditional infrastructure



Cloud service modules

Saturday, June 5, 2021 1:04 PM

Infrastructure stack Some parts are managed by **you**, some are managed by the vendor.



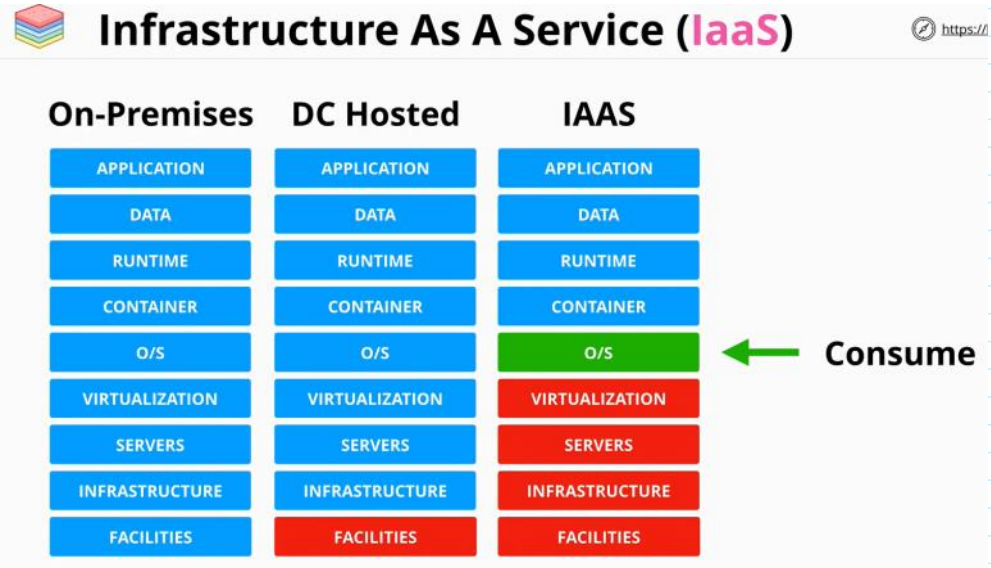
Unit of consumption - what you need to pay that you had consumed.
- part that are you are responsible to manage

In a "on-premises" server (business owns this server)
- the business needs to own/buy each part of the infrastructure
- the business needs to manage all parts of the infrastructure

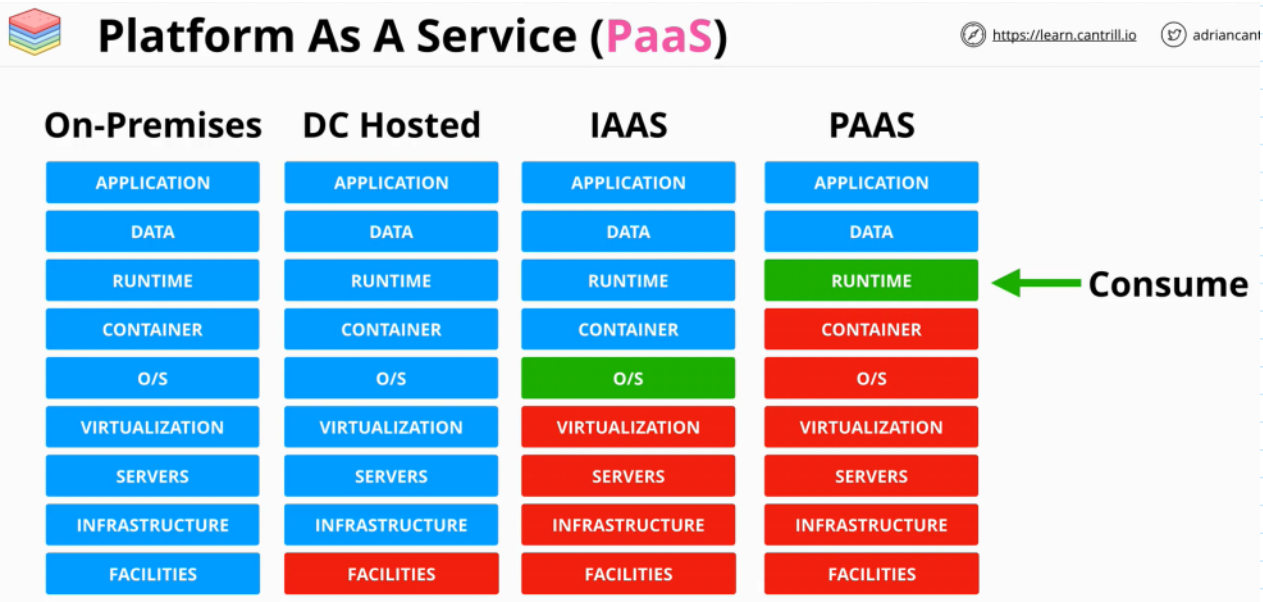
In "DC Hosted"
- the vendor paid an managed the building, the power, air condition and staff

Infrastructure as a service **(IAAS)**

With IAAS you pay per second/minute/hour for the virtual machine - when you use the virtual machine.
- the client ignores the building, maintain the infrastructure, hardware, and staff costs
- EC2

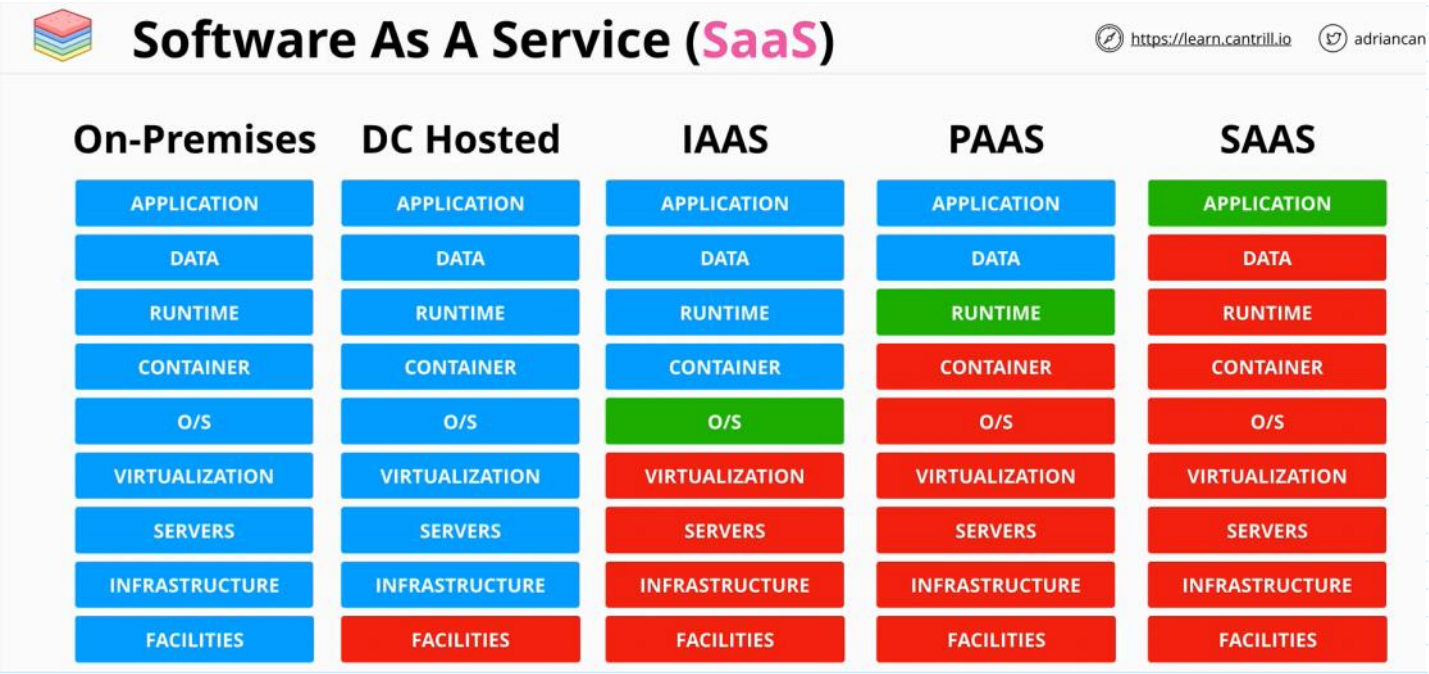


Platform as a service (**PAAS**)
With PAAS is aimed for developers who have an application and only want to run it, without worrying about any of the infrastructure.
- the apps is put in the runtime environment
- the vendor manages the containers, OS, virtualization, servers, etc



Software as a service (**SAAS**)
In this case you consume the application. The application is what you get a service. (Email, Netflix, Dropbox, OneDrive, office, etc)

- there is no much control how the app works



AWS Fundamentals

Thursday, June 10, 2021 1:52 PM

Public vs Private Services

Thursday, June 10, 2021 2:36 PM

Public and **private** keyword in AWS services are referring to the networking **only**!

When connecting to AWS:

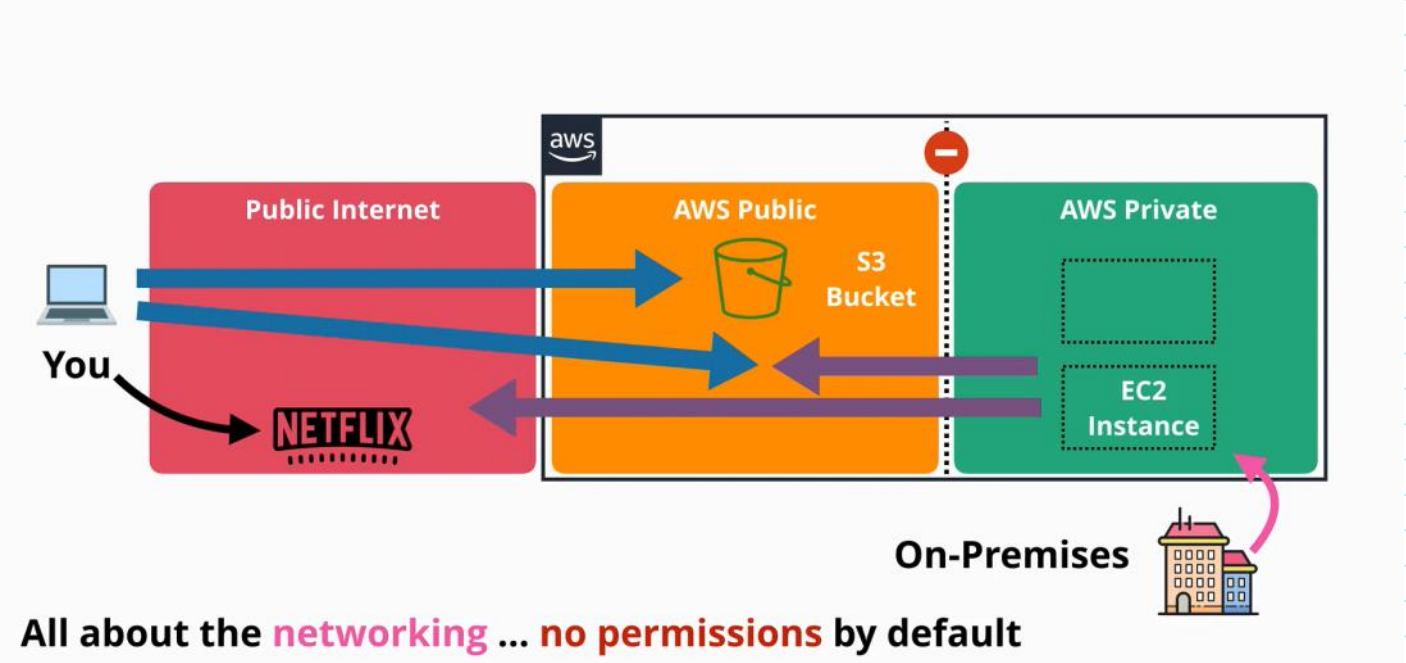
1. Connectivity is required?
2. Permissions granted for the user?

AWS is a **public cloud**, it can be connected over the public internet.

AWS has a zone called the AWS public zone, which is where public services run from, and is connected and can be accessed from the public internet.

The private zone is isolated by default, from the AWS public zone, and from the public internet. But this private zones can be configured to be public, meaning that part of them, is projected into the public zone and then accessed by a public zone.

By default private services can be accessed only from the same private networks or any on-premises networks connected to them.



AWS public service is located in the AWS Public Zone and anyone can connect but permissions are required to access the service.

AWS private service is located in a VPC, accessible from the VPC is located in and accessible from other VPCs or on-premises networks as long as private networking is configured.

AWS global infrastructure

Thursday, June 10, 2021 3:53 PM

AWS is a global cloud platform, which is a collection of smaller groupings of infrastructure connected together by a global high speed network.

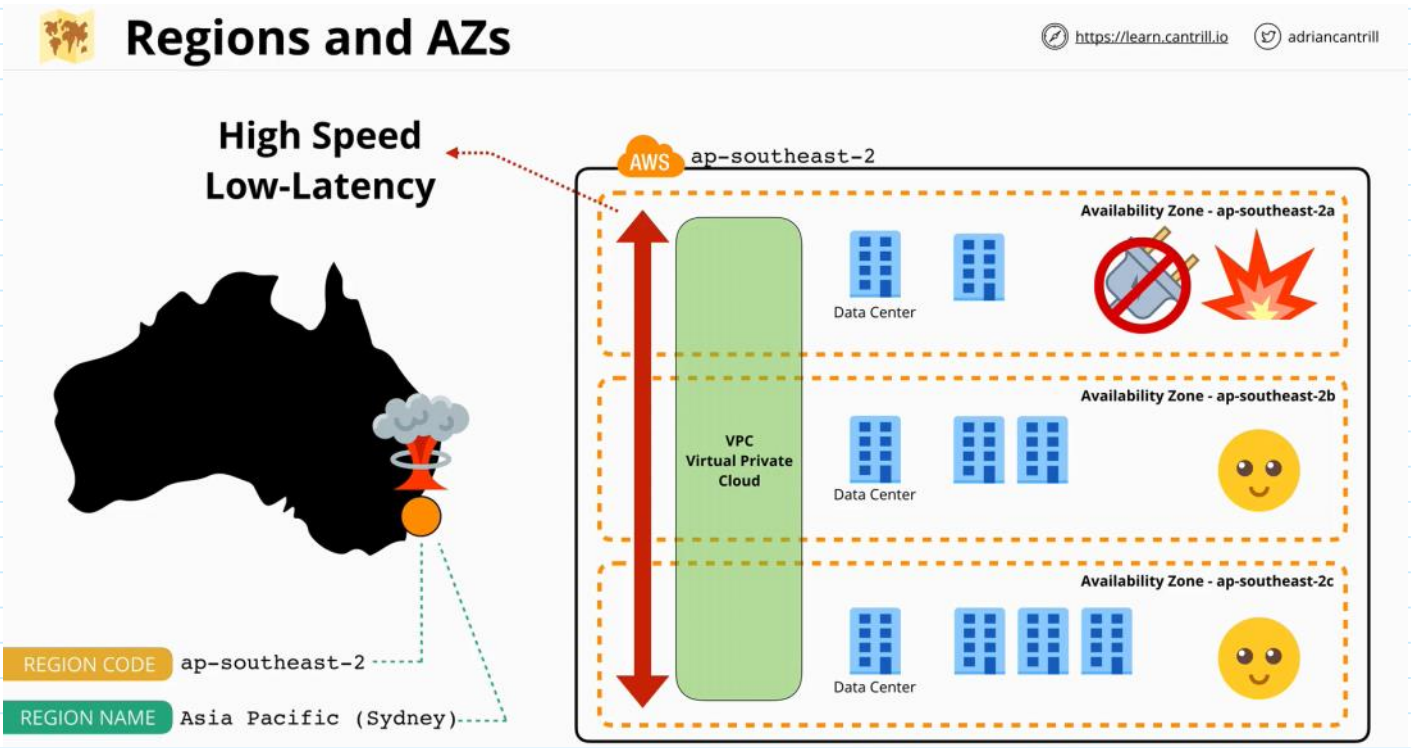
A **region** in AWS context, is an area of the world they had selected that encapsulates a full deployment of AWS infrastructure:

- compute services
- database products
- storage
- AI
- analytics
- etc

Edge locations are smaller than regions, and generally they only have content distribution services, and some types of edge computing.
The edge locations are placed in many more places than regions.
Usually, the further the services are from the clients, the slower the transfer is, and higher the latency.

Usually regions and edge location are used together by solutions architects.

Regions are 100% isolated from each other, meaning that if a disaster happens in one, the others won't be affected.
Regions have geopolitics and governance separation - the client is affected by the laws and regulations of the of the region of where the infrastructure is located.
Data saved in one region, won't leave that region unless configured otherwise.
Regions give location control, which allows the tunning of infrastructure and architecture.



Resilience:

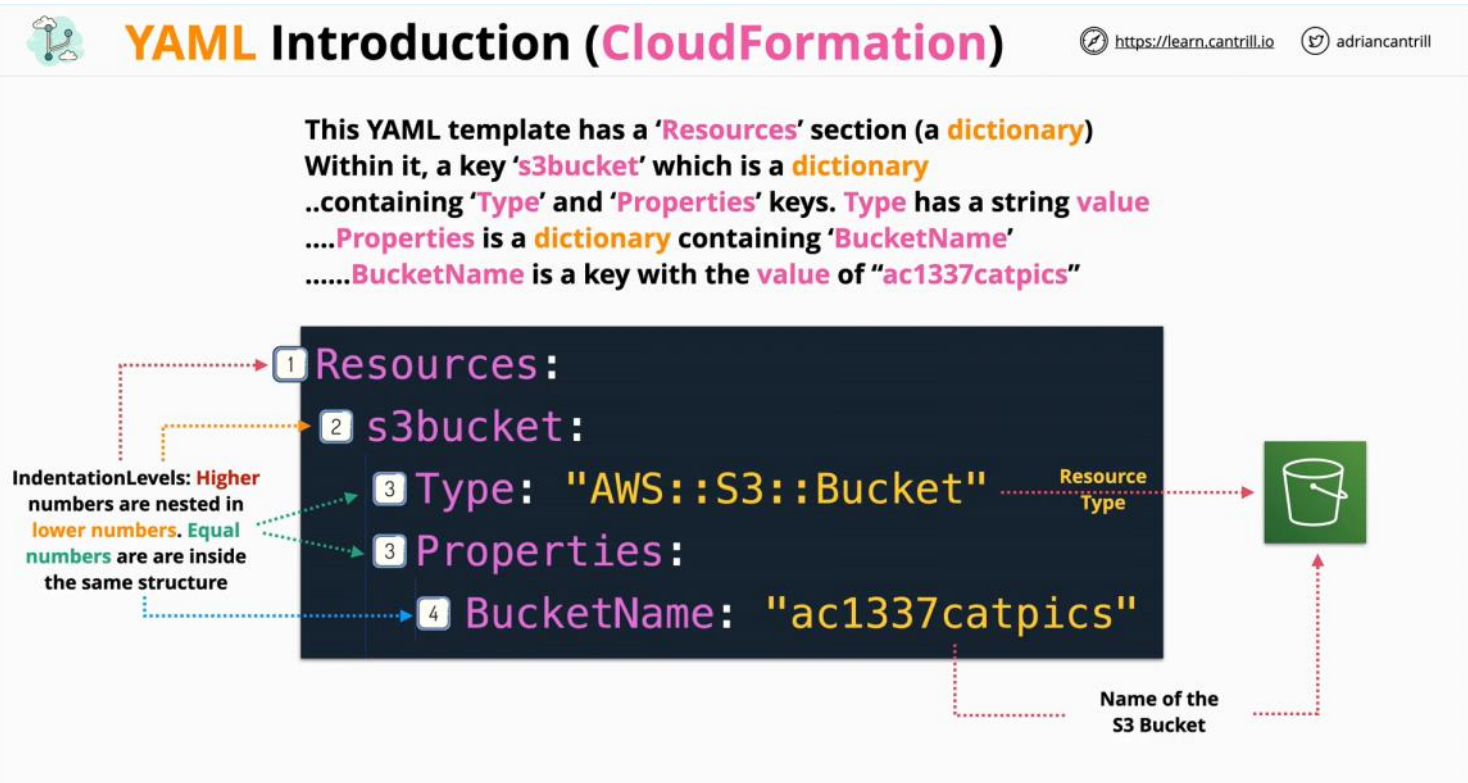
1. **Globally** means that a service operates globally as a single product, and its data is replicated across multiple regions. This means that if a region fails, the services continue to run. (IAM and Route53 are global).
2. **Region** - this are services that operate within a single region, with one set of data per region. They replicate data in multiple **availability zone** within that region.
3. **AZ** -they run from a single availability zone.

YAML

Saturday, June 5, 2021 1:25 PM

YAML is one of the languages, that Cloud uses as a template.

- human readable
- define data
- configuration
- Key:Value pair
- supports:
  - strings
  - integers
  - floats
  - booleans
  - nulls
  - Lists cats : ["a","b","c"] or cats:
    - "a"
    - "b"
    - "c"
  - dictionary
    - turtle:
      - name: "Pixel"
        color: ["green", "grey"]
      - name: "Sid"
        color: ["black", "green"]
- indentation matters

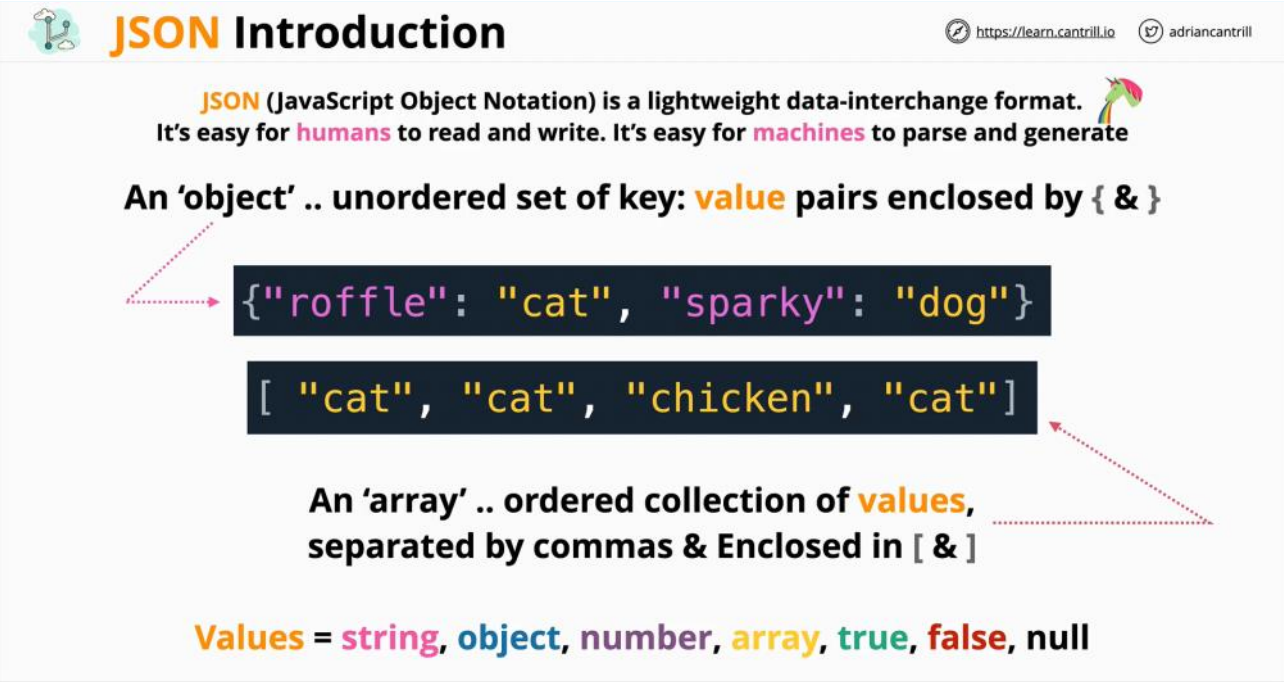


JSON

Saturday, June 5, 2021 1:39 PM

JavaScript Json Notation (JSON) - used broadly
- used for data-interchange format
- easy for humans to read and write
- easy for machines to parse and generate
- does not care about indentation - has brackets

JSON **object** is an unordered set of key:value pairs enclosed by curly brackets {} - dictionary
JSON **array** is an ordered collection of values, separated by commas and enclosed in square brackets [] - list
Values can be:
- string
- object
- integers
- array
- booleans
- null



COLLECTION OF KEY: VALUE PAIRS

{ JSON OBJECT }

```
{
  "cats": ["roffle", "truffles", "penny", "winkie"],
  "colors": ["mixed", "mixed", "grey", "white"],
  "numofeyes": ["2", "2", "2", "1"]
}
```

Each key - "cats", "colors", "numofeyes" has a value
For each example above, this is an array

The top level is a collection of unordered key:value pairs, where the value is a JSON object

Each Nested Object is a collection of key: value pairs where the value can be a scalar, a list or a JSON object

```
{
  "roffle" : {
    "color" : "mixed",
    "numofeyes": "2"
  },
  "truffles" : {
    "color": "mixed",
    "numofeyes": "2"
  },
  "penny" : {
    "color": "grey",
    "numofeyes": "2"
  },
  "winkie": {
    "color" : "white",
    "numofeyes": "1"
  }
}
```

Four Nested JSON Objects

{ JSON OBJECT }

This JSON template has a 'Resources' key, its value is a JSON OBJECT
Within it, a key 's3bucket', its value is a JSON OBJECT
..containing 'Type' and 'Properties' keys. Type has a string value
....Properties is a JSON OBJECT containing 'BucketName'
.....BucketName is a key with the value of "ac1337catpics"

```
{
  "Resources": {
    "s3bucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "BucketName": "ac1337catpics"
      }
    }
  }
}
```

Resource Type

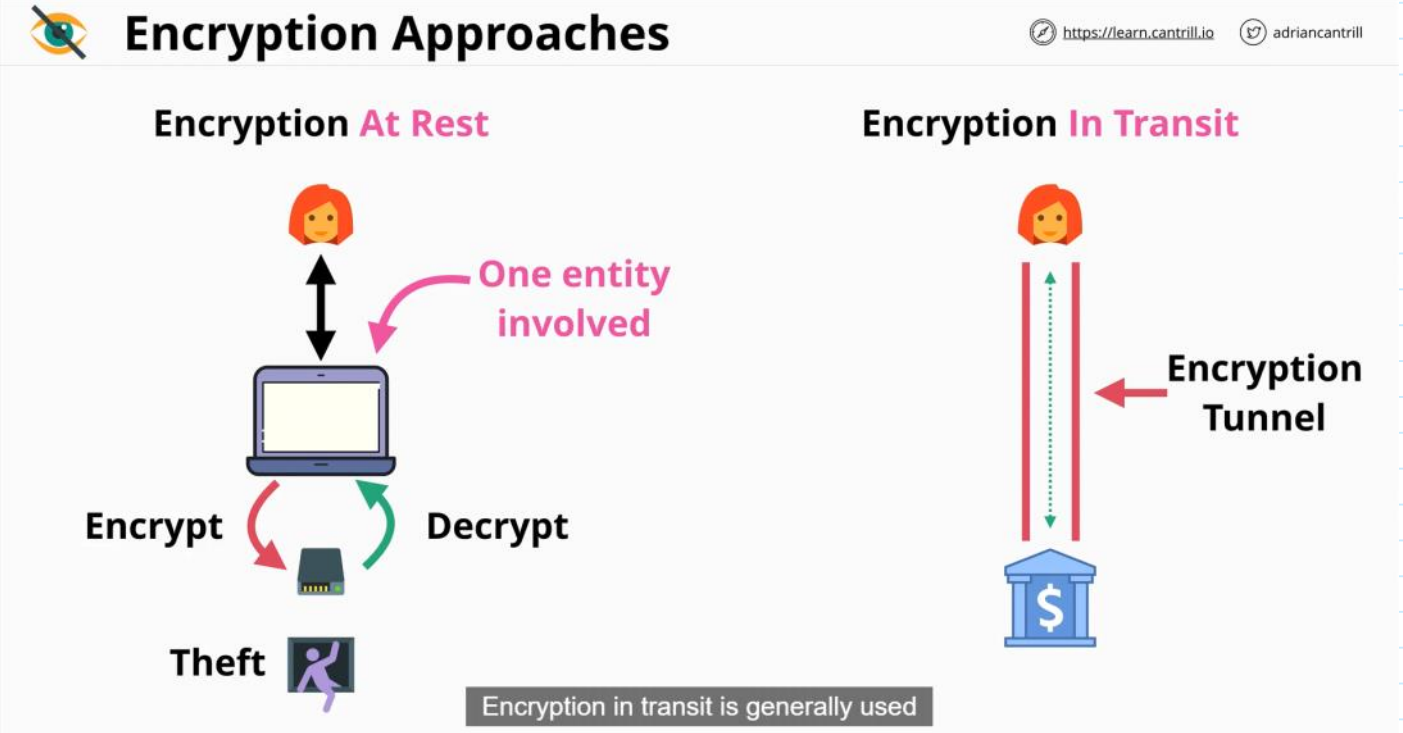
Name of the S3 Bucket

Encryption

Saturday, June 5, 2021 1:51 PM

Encryption Approaches

Encryption at rest	Encryption in transit
• Designed to protect against theft • Encrypts all the data written to the storage • Decrypts the data as is read from the storage • Is usually used if only one entity involved • who know the encryption and decryption key • Used by cloud environments as data is saved in shared hardware	• Aims to protect data as is in transit between two places • The data is encrypted on the "way" and decrypted when it arrives to the destination • Is usually used when multiple entities are involves (send and receiver)



Encryption concepts

Plaintext - text, images or anything that can be read by a person
Algorithm - code/maths which combined with an encryption key can encrypt plaintext ->outputs **cypher text**
Key - is a password
Cypher text - encrypted data
Decryption - taxes cyphertext and the key to output plain text

Keys

Symmetric encryption	Asymmetric encryption
• all entities use the same algorithm	• all entities use the same algorithm

• all entities have the same key
• the key is transferred to all entities in a secure way
  • transporting the key is a problem
  • usually is transferred before the cypher text
	• the entities agree on keys
	  • a public key
	  ○ used to generate cypher text
	  • a private key
	  ○ used to decrypt the cypher text
	• The key do not need to be exchanged in advance
	  • the reader needs to have the private key
	  • the writer needs just the public key
	• **Encryption does not prove identity**
	• Signing - using the private key, where the receiver uses the public key of the sender to check the signature
	  • is used to verify identity
	  • is called **key signing**

Steganography

Sometimes encryption is not enough. Steganography is another layer of protection. Hiding a message within a picture, or something else.

IAM

Friday, June 4, 2021 1:38 PM

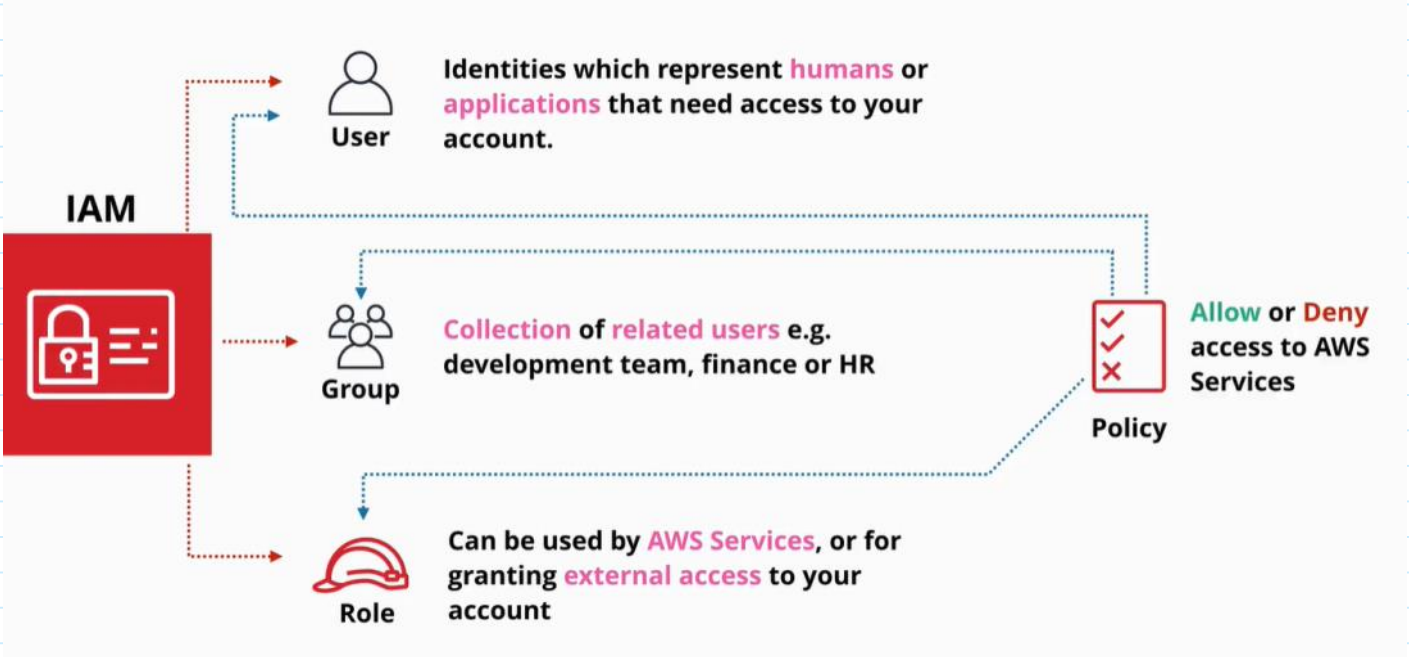
Each account has a **root** user that has **full permissions** on the AWS accounts. - unrestricted access
The **root** user cannot be restricted in anyway.

Other accounts to have access to AWS account can be users, groups or applications. These accounts should be restricted to what they needs to achieve, and are called **least privilege access**.

Every AWS accounts comes with a trusted **IAM** and its own database. This **IAM** belonging to the accounts, is an instance dedicated to **this** account, and is separated from the other AWS accounts.
When a user is added, the account automatically trusts the new user as much as it trusts IAM.

IAM allows to create policies for each users, group, role. These policies state what is allowed or restricted for each identity.

IAM identities start with no permissions on an AWS Account, but can be granted permissions (almost) up to those held by the Account Root User.



IAM has 3 main jobs:

1. CRUD identity (IDP)
2. Authenticate identities
3. Authorize identities. Allows them or denies them actions based on authentication.

IAM is:

- free, but there are some limits.
- is a local service, for the AWS account/infrastructure
- controls what identities can do : allows/denies via policies
- controls only local permissions

There is limit of **5000 IAM user/account**
IAM is a **global service** so this is a global limit
An **IAM user** can be a member of maximum **10 groups**

Create admin (IAM) user

Friday, June 4, 2021 1:57 PM

1. click to IAM console
2. click on user (left)
3. click add user
4. for an admin account, select the:
    a. Attach existing policies
    b. **AdministratorAccess**

This account now has the same privileges as the root user of the account, but is privileges can be restricted by the root user.

IAM Access Keys

Friday, June 4, 2021 2:34 PM

1. user and password are required when using the AWS console
2. access keys are required when using the terminal

An IAM user has 1 user and 1 password, and it cannot have more than that.

- **username** is public
- **password** is private
- **MFA** is private

IAM Access Keys

- **Access Key ID** - like a username
- **Secret Access Key** - as password (can only be seen once and cannot be changed)

Rotated Access Keys - change the access keys (delete and create a new one)

- can be used only by IAM **users** (roles can use one)

Create access keys: (only 2 access keys can be on one user)

1. go to security credentials
2. create access key

Configure the terminal:

aws configure --profile name-of-the-profile	configure a profile
aws s3 ls --profile iamadmin-management	ls s3

Identity Policies

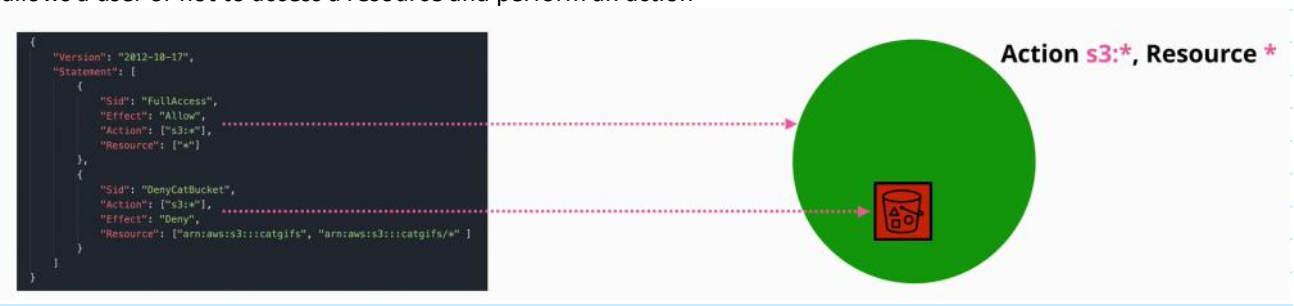
Thursday, June 24, 2021 1:34 PM

IAM identities are users, groups and roles.
An IAM policy is a set of security statements. It grants or denies access to AWS products and features to any identity that uses that policy.

Identity policies also known as **policy documents**, are created using JSON.
A policy document is one or more statements.
Once an entity is authenticated, and AWS knows what policies an identity has.

A statement only applies if the interaction the identity has with the AWS matches the action and the resource.
A **statement has:**

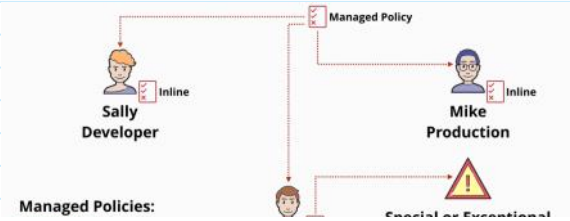
- **SID** (statement ID)
  - optional filed
  - identifies a statement
- **Action**
  - matches one or more actions
  - it can be very specific
  - service:action
  - wildcards can be used to match all actions of a service S3:*
  - there are three options
    - a specific individual action
    - a wild card
    - a list of multiple independent actions
- **Resource**
  - are the same as actions, but they match a **resource**
  - matches individual AWS resources or a list of AWS resources, wild cards are acceptable as well
- **Effect**
  - can be **allowed** or **denied**
  - allows a user or not to access a resource and perform an action



  - in the picture above, effect allowed and denied are granted over a S3 bucket.
    - in this case **overall access** is **allowed**, but **constrained** on the catgifs bucket
  - **Explicit deny wins always**
  - **Explicit allow** wins but not in favor of **explicit deny**
  - **If an effect is not specified, than is implicit denied**
  - **By default a policy is DENIED**
  - **RULE: deny, allow, deny**

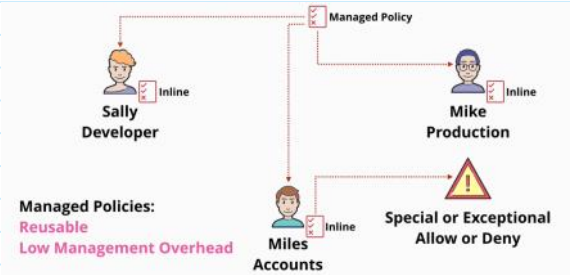
Inline policies - are granted for each entity individually
- used for exceptions - special circumstances

Managed policy - one policy attached to many identities
- AWS managed policies
- customer managed policies - created by the owner



Inline policies - are granted for each entity individually
- used for exceptions - special circumstances

Managed policy - one policy attached to many identities
- AWS managed policies
- customer managed policies - created by the owner



IAM Users and ARNs

Thursday, June 24, 2021 3:55 PM

IAM Users are an identity used for anything requiring long-term AWS access e.g. Humans, Applications or service accounts

These are called **PRINCIPLES**, and for the principles to be able to do anything it needs to authenticate and be authorized.

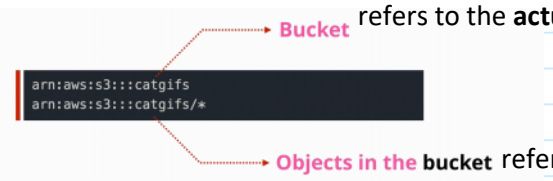
The process is like this:

1. the principle makes a request to IAM to access/interact resources
2. the principle need to authenticate as an IAM used
    a. can be achieved in two ways: **login credentials** or **access keys**
        i. **humans used login credentials**
        ii. **applications use access keys**
3. Now the principle is an authenticated entity and can start interacting with AWS
4. AWS knows which polies applies to the identity **(authorization)**

AMAZON RESOURCE NAME (ARN)

ARN are uniquely identify resources within any AWS accounts

ARNs can always identify single resources in the same account weather they are individual resources in the same account or different accounts.
ARNs are used in IAM policies which are generally attached to identities such as IAM users, and they have a defined format.



refers to the **actual bucket**

refers to **all the objects in the bucket**

ARN structure:

arn:partition:service:region:accountID:resourceType/resourceID

if any are omitted it means it does not refer to one in particular (like regions) - DOES NOT MEAN IT INCLUDES ALL!!!

There is limit of **5000 IAM user/account**
IAM is a **global service** so this is a global limit
An **IAM user** can be a member of maximum **10 groups**

IAM Groups

Thursday, June 24, 2021 5:00 PM

IAM groups are containers for IAM users.

- you cannot login into a group
- the group is used to manage policies easier for users
- facilitates the management of IAM users

An **IAM user** can be part of multiple **IAM groups!**

- it does **not EXIT** by **default** a general **IAM Group** that applies to all IAM users.
  - if a group like this is needed, it needs to be created and managed by the admin
- **groups within groups does not exist**
- **IAM groups** contain users
- **IAM** groups contain permissions attached
- **there is a limit of 300 groups/account** (can be extended by a support ticket)
- can hold identity permissions
- admin groupings of IAM users

Policies can be attached to resources

- these policies can grant access to different IAM users
- allows or denies access to IAM users to identities
- id does this by referencing this identities using an ARN
- a policy on a resource and reference IAM users and IAM roles by using ARN
- groups are not identities and cannot be referenced as principal in a policy
  - resources cannot grant access to IAM groups

IAM Roles

Sunday, June 27, 2021 1:54 PM

A single **principle** uses an IAM user.

IAM roles are also identities

- used by multiple principles (not just one as users)
  - they can be multiple AWS users
  - they can me multiple external applications, users, etc.
- if you cannot identify the number of principles that use an identify then is a candidate for an IAM role
- if you have more than 5000 users (the limit of IAM users) it can also be a candidate for an IAM role
- used on temporary basis
- a role **represents a level of access inside an AWS account**
- **an IAM user is a representation of a user for long term**
- **a role borrows the permissions on a short period of time**

IAM users have permission policies attached to them in inline or managed policies

- these policies define what permission an identity gets inside AWS

IAM roles

- have two types of policies that can be attached to them
  - trust policy
    - which identities can assume the role
    - can refer different identities: IAM users, other roles and AWS services, identities in other AWS accounts
    - if an identity is allowed, AWS generates temporary security credentials (like access keys)
      - they are time limited
      - can be renewed
      - the identities can access the resources specified in the **permission policy**
  - permission policy
    - every time the temporary credentials are used are checked against the permission policy

Roles are real identities, as IAM users, roles can be referenced in resource policies.

- are used in AWS organizations, allowing us to log into one account in the organization and access different account without having to login again
- they are really useful when having a large number of accounts

Temporary credentials, for roles, are generated by a service called Secure Token Service (STS). sts:AssumeRole

When and where to use IAM Roles

YES	NO
Services that need permissions	
Anything that is not an identity	
Very useful in emergency situations - break glass situation - used when absolutely required	
When adding AWS in an existing corporate environment **Roles are often used when you want to reuse the existing identities for use in AWS** - external accounts cannot be used directly - allowing a role to be assumed by an external account (or multiple) - called **ID federation**	
Giving access to users of an app,	

Hardcoding access keys is not advisable as is a security concern, and whenever the keys change, they need to be hardcoded again.
In this scenario, the **IAM Role** is the perfect solution.

to some resources - called web identity federation	

AWS Organizations

Sunday, June 27, 2021 5:11 PM

AWS Organizations is a product that **allows companies to manage multiple AWS accounts** in a cost effective way.

An account is needed to create an organization.

The account that creates the AWS account, becomes the management account for that organization.
The management account is also called master account.
Using the management account

- you can invite other standard AWS accounts into the organization
  - the invited standard AWS accounts need to accept the invitation to the organizations
  - if the standard AWS accounts accept the invitation they become part of the organization
  - these accounts are changed from being standard AWS accounts to **member accounts of that organization**
- an organization has **only one** master accounts, and **zero or more member accounts**
- **the organization can create accounts directly within it**
  - it isn't an invite process
- with organizations, IAM users do not need to exit within each member AWS account
  - IAM roles can be used to allow IAM users to access other AWS accounts

The Organization has a hierarchy (like a reverted tree)

- the **root container** of the organization is at the top of the tree
  - do **not** confuse it with the **root user**
- other containers can be created - known as **organizational units (OUs)**

Organizational billing

- the billing method for each member account are removed
- the member accounts pass their billing through the management account of the organization also called **payer account**
- the single monthly bill generated for AWS organization, covers all the accounts of the organization

The organization has a service called Service Control Policies (SCPs)

- this service can restrict what each member account can do

Service Control Policies (SCP)

Friday, July 2, 2021 12:10 PM

SCP is a feature of AWS organizations, which can be used to restrict AWS accounts.

SCP is a JSON document (policy document) and can be attached to the
- organization as a whole
    - by attaching them to the root container
- one or more organizational units (OU)
- to individual AWS accounts

Service Control Policies inherit down the organizational tree.
- if the are attached to the organization it affects all accounts within the organization
- if they are attached to an organizational unit, they impact all the account inside the organizational unit
    - if there are nested OUs , it will affect the OUs inside the organization as well

The management account of the organization is special. If this account has policies attached directly, or through an organization unit, **the management account is not affected by any SCP!**

Service Control Policies are account permission boundaries and they limit what an account can do.

As mentioned before, we cannot restrict the account root user of an AWS account, but within an organization with the SCP we can restrict what an AWS account can do (with all the entities in that account)! Therefore, restricting indirectly the account root user.

SCP do not grant any permissions, they define the limit of what isn't allowed (and what is). You still need to give the identities in that AWS account permissions to AWS resources, **but SCPs will limit the permissions that can be assigned to individual identities.**

SCPs can be used in two ways:
1. Block by default and allow certain services: using an allow list
2. Allow by default and deny certain services: using a deny list (default)

When SCP is enabled in an account, AWS applies a default policy which is called a **full AWS access. This is applied to all OUs within the organization. - nothing is restricted**
- **this maintains the current functionalities in the organization (in the moment of creating the SCP)**

To use the allow list, the **FULL AWS access** list must be removed, and allowed services should be specified independently into a new policy. - **security improved**

SCP do not grant access to anything, they specify WHAT CAN AND CANNOT BE ALLOWED by identity policies within that account

Only what is allowed in the SCP and the identity policies is actually allowed.

Policy Interpretation Deep Dive

Monday, July 12, 2021 4:28 PM

1. identify how many statements make up the policy document
    - each document has a single statement or a list of statements
        - a list of statements has **[{statement}, {statement}, {etc}]**
2. identify what each statement actually does
    - each statement has an **effect** which is **allowed** or **denied**
        - this states if that certain statement **allows** or **denies** a set of **activates**
3. start with the **ALLOWED** statements
4. end with **DENIED** statements
5. **check for NOT operations**

- The priority orders is **Deny, Allow, Deny**
    - **by default is denied** - no permissions
        - if a permission is not explicitly allowed in the policy is by default denied
    - **if something is denied explicitly that always wins**
    - **if something is allowed explicitly then is allowed, unless is ALSO denied**
    - **if something is no explicitly denied or allowed, than is implicitly by default denied**
- **if conditions are present in a statement, the statement applies only if the condition is a met**
- **if a policy contains only a DENY, is usually used with another policy, as a single DENY will do nothing to permissions, as by default permissions are DENIED**

```
{
    "Version": "2012-10-17",
    "Statement":
    [
      {
        "Effect":"Allow",
        "Action":[
           "s3:PutObject",
           "s3:PutObjectAcl",
           "s3:GetObject",
           "s3:GetObjectAcl",
           "s3:DeleteObject"
        ],
        "Resource":"arn:aws:s3:::holidaygifts/*"
      },
      {
        "Effect": "Deny",
        "Action": [
          "s3:GetObject",
          "s3:GetObjectAcl"
        ],
        "Resource":"arn:aws:s3:::holidaygifts/*",
        "Condition": {
            "DateGreaterThan": {"aws:CurrentTime": "2020-12-01T00:00:00Z"},
            "DateLessThan": {"aws:CurrentTime": "2020-12-25T06:00:00Z"}
        }
      }
    ]
}
```

check for NOT operations

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyNonApprovedRegions",
            "Effect": "Deny",
            "NotAction": [
                "cloudfront:*",
                "iam:*",
                "route53:*",
                "support:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestedRegion": [
                        "ap-southeast-2",
                        "eu-west-1"
                    ]
                }
            }
        }
    ]
}
```

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::cl-animals4life",
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "",
                        "home/",
                        "home/${aws:username}/*"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::cl-animals4life/home/${aws:username}",
                "arn:aws:s3:::cl-animals4life/home/${aws:username}/*"
            ]
        }
    ]
}
```

- **List everything inside buckets**
- **List a specific bucket - animals4life**
  - **no prefix**
  - **all home/ prefix**
  - **only the buckets that match the name of the IAM user using the policy and everything inside that bucket -** and no other "folders"
- **Allows any operation, as long as the prefix matches animals4life/home/iamusername - on the folder and inside it**

Policy Evaluation Logic

Tuesday, July 13, 2021 11:19 AM

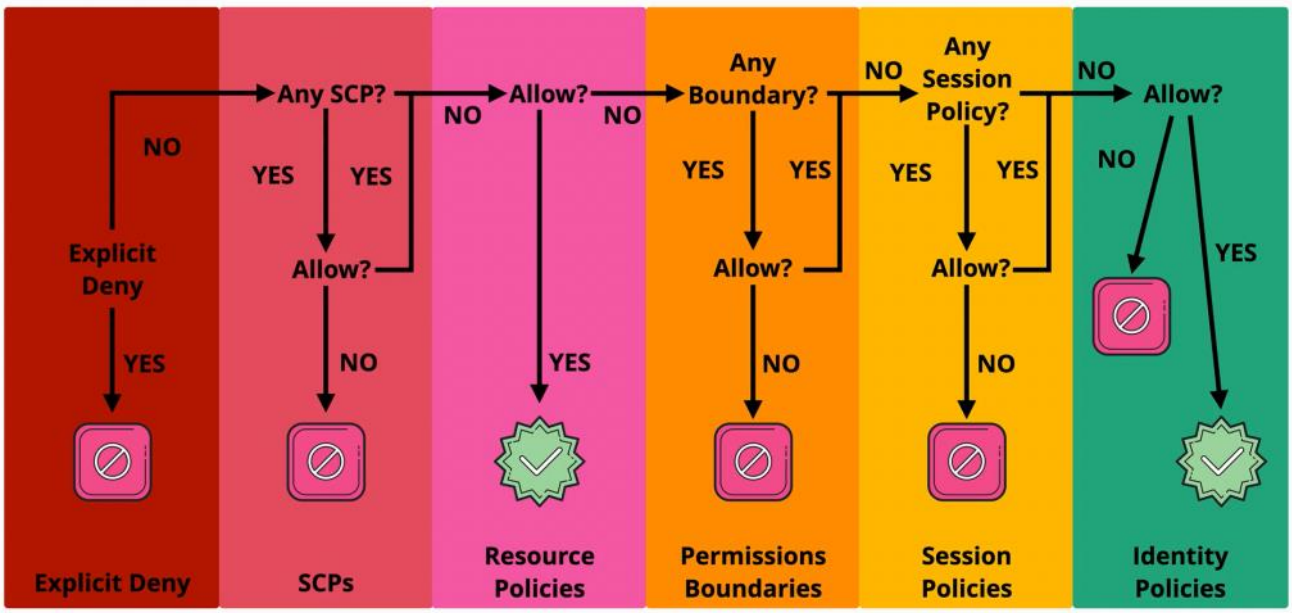
Security architecture can have multiple levels of identities and account boundaries, and even multiple AWS accounts.

When evaluating effective permissions, there a few components involved:

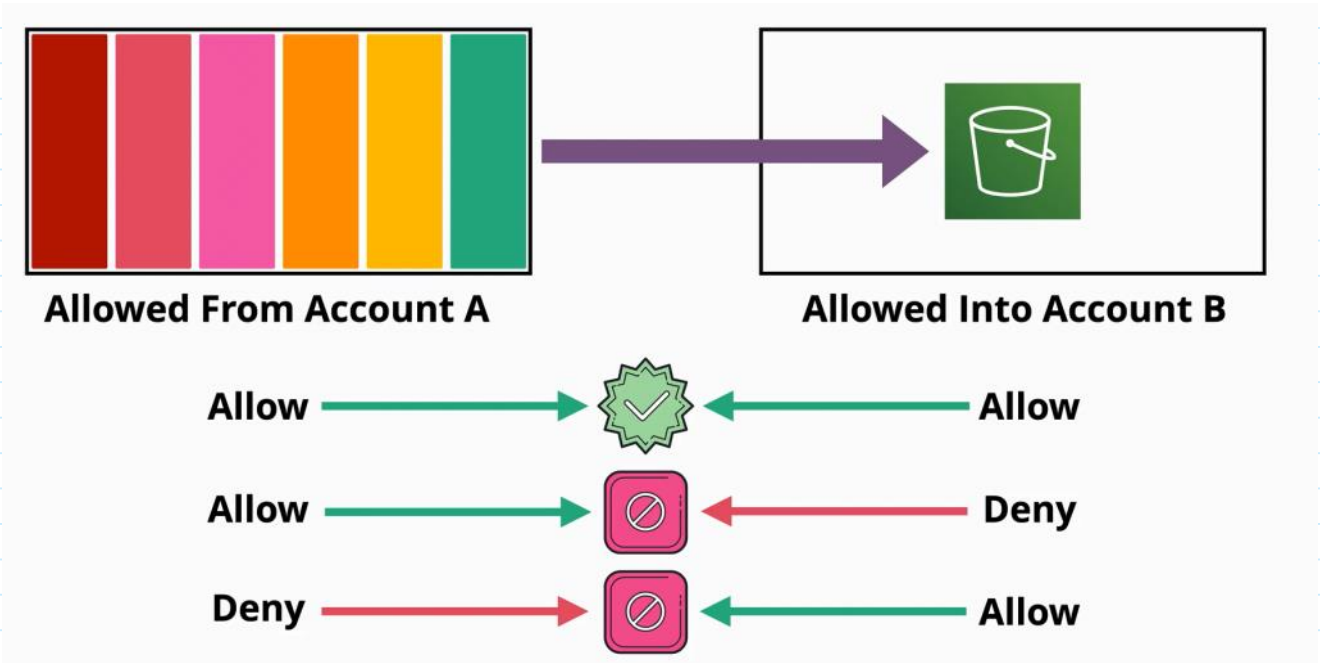
1. consider any AWS organization service control policies
    - this impact what identities inside an organization can do
2. consider the resource policies
3. IAM identity boundaries
4. session policies - what a role is allowed
5. identity policies

Handling permissions - same account

- AWS gathers all policies which apply to an identity accessing a particular resource
    1. **EXPLICIT DENY -** searching for an **explicit deny** in what the identity is trying to do with a certain resource or service
        - if there are explicit deny then, the actioned is denied
        - if there is no explicit deny then the processing continues
    2. **SERVICE CONTROL POLICY -** if there are no explicit denies, then service control policies (SCP - part of the organization product) are processed
        - if service control policies exist, then they are checked if they allow the action
        - if the service control policy allows or if the service control policy does not exist, then the processing continues
        - if the service control policy does not allow the process, then the access is not permitted
    3. **RESOURCE POLICY** - if the resource policy contains an **allow** then the actioned is allowed
        - otherwise, the processing continues
    4. **PERIMISSION BOUNDARY** - following the same process
        - if there is a no boundary or if the actioned is allowed then processing continues
        - if the a boundary exists and denies the action then processing stops, and the action is denied
    5. **SESSSION POLICY -** is checked if a role is used
        - if the action is not explicitly allowed, then is implicitly denied and the processing stops denying the action
        - if there is session policy, or if there is one allowing the action then processing moves on
    6. **IDENTITY POLICY**
        - if the action is explicitly allowed then the action is permitted
        - if the action is explicitly denied then the action is denied
        - by default any action is **denied**



Handling permissions - multiple accounts



Cloud HSM

Tuesday, July 13, 2021 12:38 PM

Cloud HSM is a **cloud**-hosted Hardware Security Module (**HSM**) service that allows you to host encryption keys and perform cryptographic operations in a cluster of FIPS 140-2 Level 3 certified HSMs. Google manages the **HSM** cluster for you, so you don't need to worry about clustering, scaling, or patching.

KMS

KMS is the key management service within AWS

- used for encryption within AWS and integrates with other AWS products
  - generates keys
  - manages keys
- but has one security concern - it is a shared service
  - your part of KMS is isolated under the cover it is shared with other accounts
  - while the permissions in AWS are strict, AWS has a certain level of access to the KMS product - they manage the hardware and software which provides the KMS product to its customers
- behind the scenes KSM uses HSM (Hardware Security Model) - industry standard pieces of hardware which are designed to manage keys and perform cryptographic operations
- is **Level 2 - some Level 3**
- all operations are performed with AWS standard API, and all permissions are controlled with IAM permissions

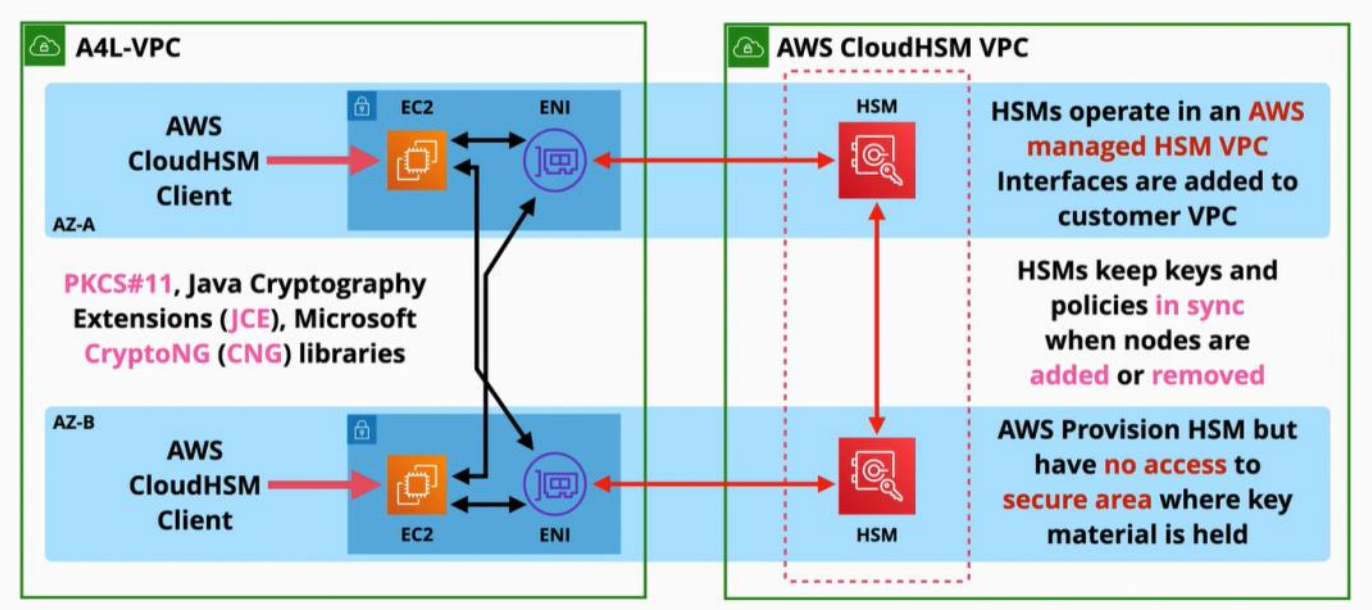
HSM

HSM is a TRUE "SINGLE TENANT" Hardware Security Module (HSM)

- Cloud HSM - hosted by AWS
  - HSM is hosted by AWS and they are reasonable for hardware maintenance
  - **they have no access to the part of the unit where the keys are stored and managed**
  - is not very integrated with AWS
    - you access it with Industry Standard API: PKCS 11, Java Cryptography Extensions (JCE), Microsoft CryptoNG (CNG)
- On-Premise HSM Device
- is **FIPS 140- 2 Level 3**

Architecture

Cloud HSM are not deployed inside a VPC that you control. They are deployed into an AWS managed **Cloud HSM VPC ,** that you have no visibility of.



Overlaps

- A feature from KMS called **Custom Key Store**, which can use **Cloud HSM** to get many benefits from Cloud HSM and integration with AWS.

When to be used

- there is no integration between AWS standard APIs and HSM
- HSM can be used to perform client side encryption
  - encrypt on local device and then upload the encrypted object to AWS
- HSM can be used to offload the SSL/TLS processing for Web Servers
  - the server does not need the perform those cryptographic operations
  - accelerates the process
- enables Transparent Data Encryption (TDE) for different products such as Oracle Databases
- the client is responsible for managing the keys, and the encryption operations
- can be used to protect and manage Private Keys for Certificate Authorities (CA)
- **the overall theme is that anything that isn't specific to AWS, anything which expects to have access to a hardware security module using Industry Standard APIs, then the ideal product for that is Cloud HSM**
- **for anything using standards, for anything that has to integrate with products which aren't AWS then Cloud HSM is ideal**
- **for anything that requires AWS integration, then natively cloud HSM isn't suitable**

CloudWatch

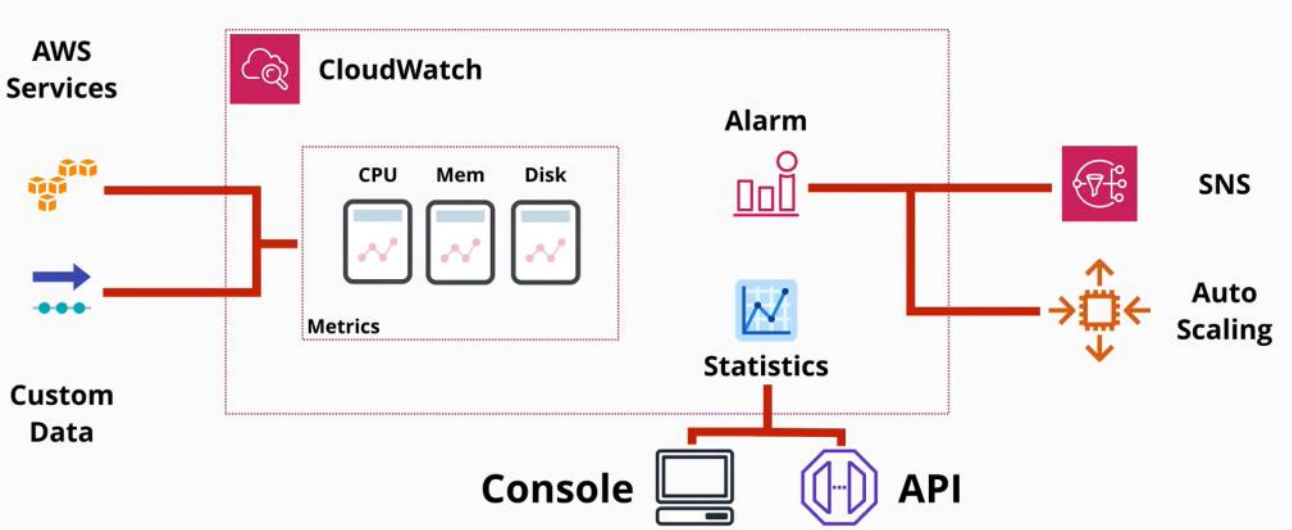
Tuesday, June 22, 2021 1:46 PM

Cloud Watch is a support service used by almost all AWS services - especially for operational services and monitoring.

Cloud watch performs three main jobs:
It collects and manages operational data

a. data that is generated by an environment

1. a collection of metrics, monitoring of metrics and actions based on metrics
    a. data relating to AWS products
        i. CPU utilization of EC2
        ii. numbers of visitors
2. Cloud Watch logs
    a. logging data
3. Cloud Watch events
    a. EC2 terminated, started or stop
    b. schedule an event



Namespace

- is a container for monitoring data, separating things in different areas
- all AWS data go into a namespace is **AWS/service** (e.g. AWS/EC2)

Metric is a collection of related datapoints in a time ordered structure.

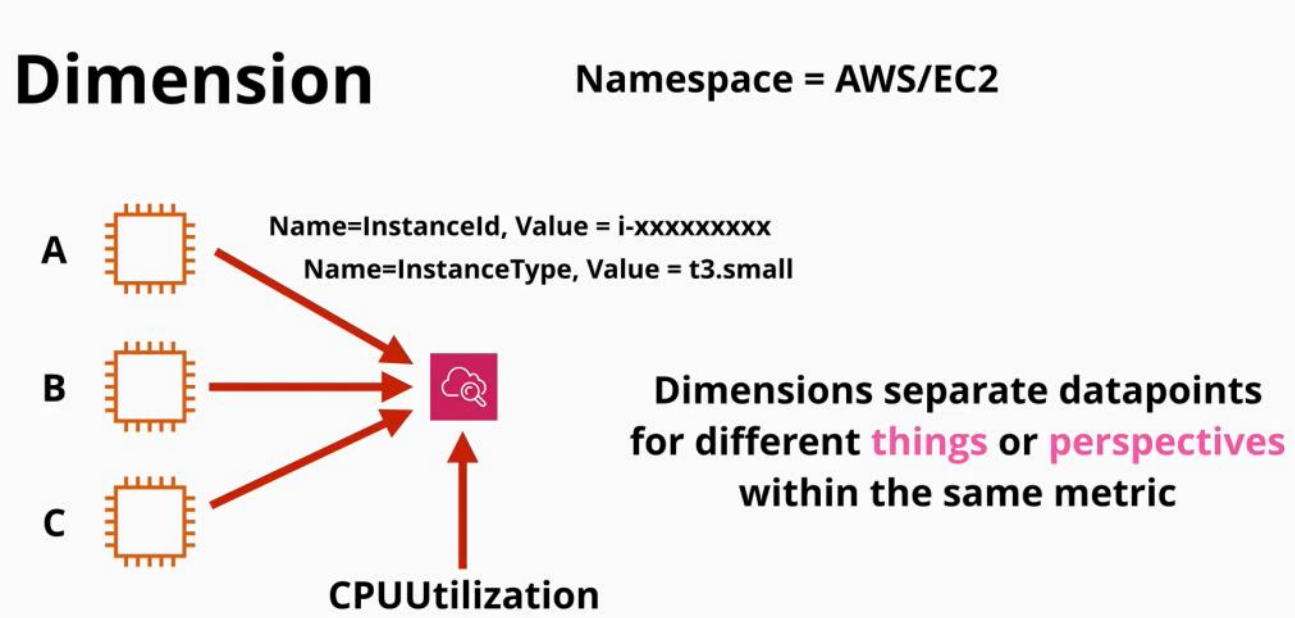
- CPU utilization
- network in an out
- disk utilization
- will start monitoring when the service is started, and it will stop when the service is disabled

Datapoints

- consists of two things:
    - time when measurement was conducted
    - and a value that represents the monitoring

Dimension

- are name value pairs which allow CloudWatch to separate things



Alarms

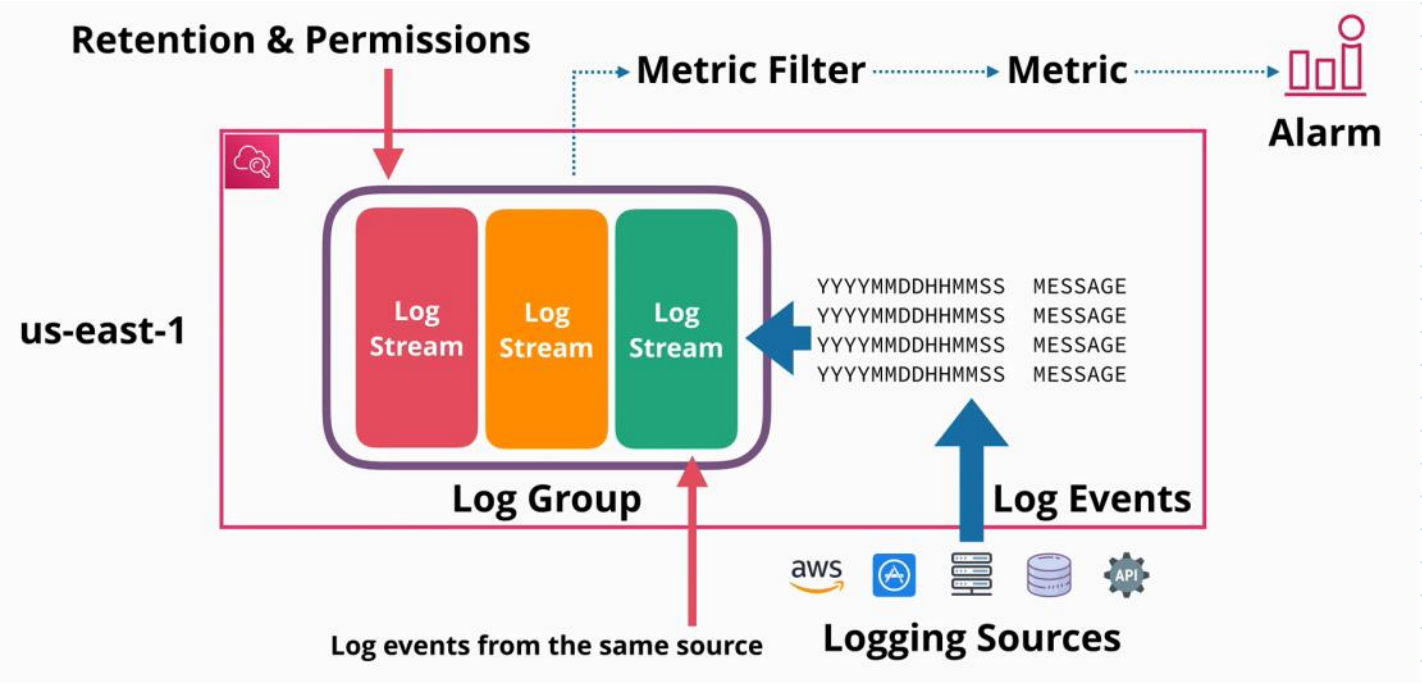
- are created and linked to a specific metric
- based on the alarm's configuration, an action can be performed if the condition for the alarm is met

CloudWatch Logs

Friday, July 2, 2021 1:22 PM

CloudWatch logs is a public service. YH endpoint in which applications connect to is hosted in the AWS public zone.

- which means you can used the product within the AWS VPCs or from on-premises environments and even other cloud platforms as long as you have internet and permissions
- allows you to **store, monitor** and **access** logging data
    - logging data is a piece of information data, and a timestamp
- has some built in integrations with AWS services such as EC2, VPC flow logs, lambda, CloudTrail, R93 and more
- each service that interacts with CloudWatch can store data directly inside the product
- AWS services can log into CloudWatch directly
- unified CloudWatch agent can be used to integrate outside services
- can generate metrics based on logs using the **metric filter**
- **is a regional service**
- **log streams** is a collection of log events from the same source
- **log groups** are containers for multiple log streams for the same type of logging
    - in log groups settings are located as well
        - this settings apply to all log streams inside that group
    - metric filters are defined here as well



CloudTrail Essentials

Friday, July 2, 2021 3:59 PM

Almost everything that can be done with an AWS account can be logged with CloudTrail product.

CloudTrail:

- logs APIs calls and account activities
  - each is called a **CloudTrail event**
    - is an activity in an AWS account
    - an activity is an action taken by an user, role or a service
- by default logs the last 90 days activity in the CloudTrail Event history
  - is enabled by default
  - is free
  - can be customized
- can be two types
  - **management events** (by default the CloudTrail only logs this)
    - provides information about operations performed on resources (control)
  - **data events** (not enabled by default)
    - contains information about resource operations performed on or in a resource
- stored the events in a definable S3 bucket (only if you configure a trail, otherwise you don't get an S3 bucket)
  - stored in JSON format
- a **CloudTrail trail** is a unit of **configuration** within the CloudTrail product
  - how configuration is done
  - **a trail logs events for the AWS region that it's created in**
  - a trail can be configured for one region, or for all regions.
- can be integrated with CloudWatch logs
  - in addition of adding into the S3 buckets dedicated for events, it adds it to the CloudWatch logs
- it is **not real-time** - has a delay (15 minutes)
- an organizational CloudTrail is available
  - can be created from the management account of the organization
  - can store all the information from **all the accounts** within the organization

AWS services are largely split up into regional services and global services. So when this different type of services log to CloudTrail they either log in the region the event is generated in, or if **they are** global (IAM, STS, CloudFront) they log into **us-east-1**. A trail needs to be enabled to capture that data!

S3 Buckets

Wednesday, June 16, 2021 6:49 PM

Simple storage service - S3

S3 is:
- global storage platform - regional resilient
- is a public service
- unlimited data
- multi-user access
- any type of data
- is economical
- **default** storage on AWS
    - **objects** - any data
    - **buckets -** containers for objects

Objects are files, made of:
- object **key** (file name)
- object **value** (the content of the file being stored)
- size range from 0MB to 5TB/object

Buckets
- **all buckets by default are private**
- data inside the bucket has a primary region (than can be changed)
- data never leaves the region (unless configured)
- **the name of the bucket needs to be unique across all regions in AWS**
- can hold unlimited number of objects
- has a flat structure - all files are saved on the root file (no folders in folders)
    - the user can save folders in folders, but in reality is a **flat structure**
    - folders **are called prefixes**
- **bucket names needs to be**
    - all lower case
    - no underscores
    - 3 - 63 characters - characters or digits
- there is a limit of number of buckets for an AWS account, and **it is a soft 100, and a hard limit 1000**
- is good for large scale, data storage, distribution or upload

Simple Storage S3 - is an AWS Public Service, is an object storage system and buckets can store an unlimited amount of data

S3 Security

Saturday, July 3, 2021 11:35 AM

S3 is private by default

- by default the account root user has access to the bucket only
- any other permissions need to be explicitly granted through:
  - **resource policy** - **who can access a resource**
    - can grant access to identities from other accounts - (other AWS accounts)
    - can allow or deny anonymous principals - (sharing a bucket with the world)
    -
```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"PublicRead",
      "Effect":"Allow",
      "Principal": "*",
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::secretcatproject/*"]
    }
  ]
}
```
    - **principal filed defines what principals have access to a resource**
      - **if is there, is a resource policy**
  - **identity policy** - **what resources the identity can access**
    - access can be granted to identities in your account only

Bucket policies

Can be used:

- to control who can access objects (even targeting IP addresses)

```
{
    "Version": "2012-10-17",
    "Id": "BlockUnLeet",
    "Statement": [
      {
        "Sid": "IPAllow",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::secretcatproject/*",
        "Condition": {
           "NotIpAddress": {"aws:SourceIp": "1.3.3.7/32"}
        }
      }
    ]
}
```

  - this permission allows only identities from 1.3.3.7, otherwise the statement applies!
- is a type of **resource policy** is applied to a bucket
- a bucket can have **only one policy** but it can **have multiple statements**
- when an entity accesses a bucket, both the identity and the bucket policy apply

Access Control Lists (ACLs)

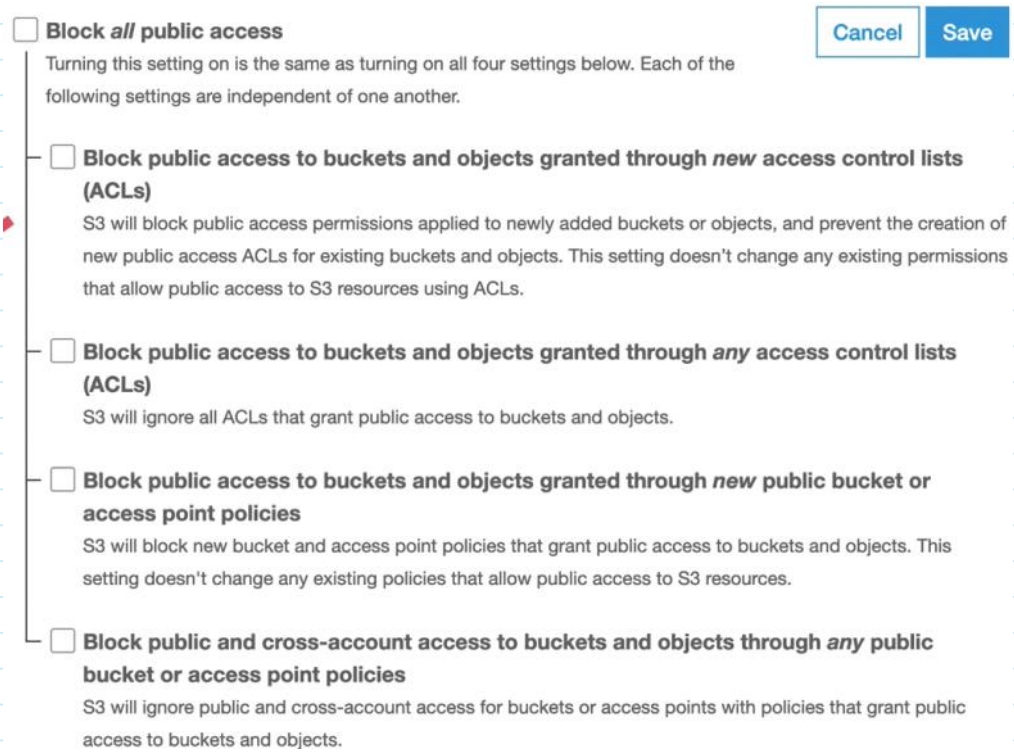
(not really used anymore)

- is a way to apply security to objects and buckets
- is a sub resource
- they are inflexible and allow very simple permissions
- you can use one ACL for multiple objects
    - one ACL per object and one ACL per bucket

Permission	Bucket	Object
READ	Allows grantee to list the objects in the bucket	Allows grantee to read the object data and its metadata
WRITE	Allows grantee to create, overwrite, and delete any object in the bucket	Not applicable
READ_ACP	Allows grantee to read the bucket ACL	Allows grantee to read the object ACL
WRITE_ACP	Allows grantee to write the ACL for the applicable bucket	Allows grantee to write the ACL for the applicable object
FULL_CONTROL	Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket	Allows grantee the READ, READ_ACP, and WRITE_ACP permissions on the object

Block Public Access

- Adds another level of security, which apply no matter what the **bucket policy says**.
    - the Apply only for public access
    - the do not apply to AWS identities



1. **if you are granting and denying permissions on lots of resources across AWS accounts, then** identity policies **should be used, as not all resources have** resource policies.

2. **if you have a preference to manage resources all in one place that single place needs to be** IAM**.**
3. **if you are working with permissions within the** same account**, then IAM is the place to create policies, as IAM** can manage policies only within that account identities.
4. **If you want to grant permission to a single resource to multiple users or everybody in one account then is much more useful to use** resource policies that give level permissions
5. **if you want to directly allow** anonymous identities access**, then again** resource polies are the best fit, as identity policies cannot control permissions for principles outside the account
6. **never user ACL unless you absolutely have to!**

S3 Static Website Hosting

Saturday, July 3, 2021 12:59 PM

S3 Static Website Hosting allows access to buckets and object via HTTP.

- when enabling it, you need to set an **index** and an **error** document
  - index home page - default page
  - error page - is accessed when something goes wrong
- needs to be HTML
- if you use a domain than the bucket name needs to match the domain's name
- the default endpoint is influenced by the bucket name and the region is in

Offloading

- if a website needs a database to load static media (images), S3 bucket can be used to store the media and retrieve it when needed it
- this is cheaper than other compute services as it is costumed-designed for the storage of large data at scale

Out of band pages

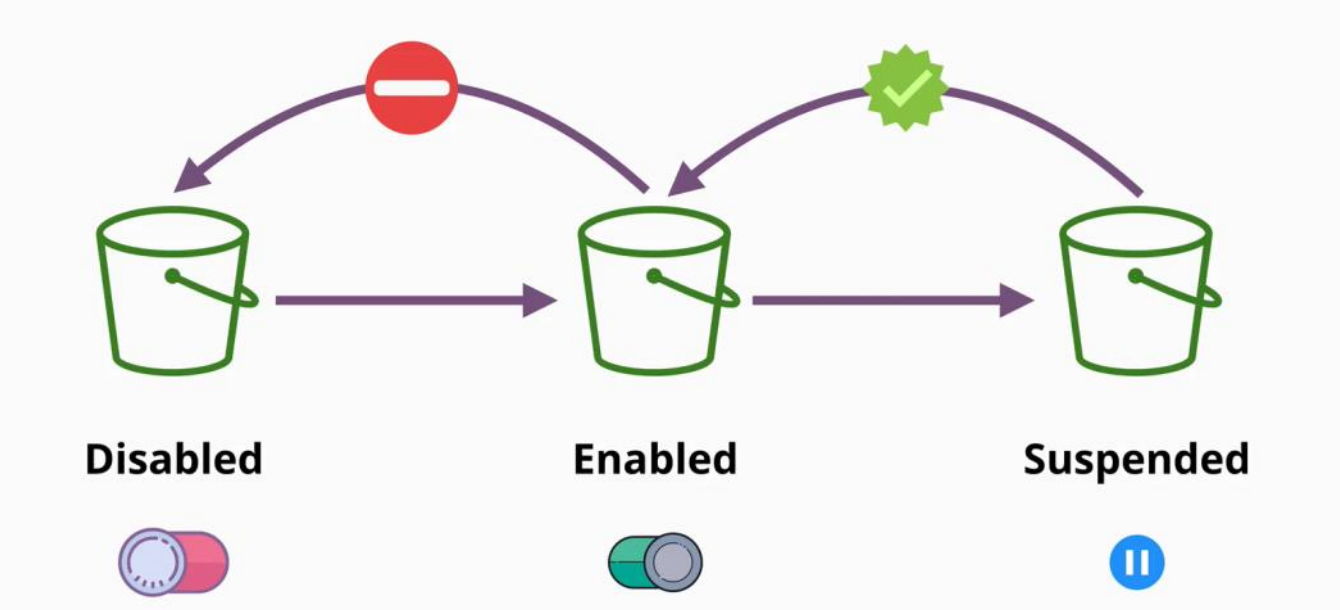
- another services used, in case the main server is offline or has any other interruptions, then we can change our DNS and point customers at a backup static website hosted on S3

Object Versioning and MFA Delete

Saturday, July 3, 2021 3:12 PM

Object Versioning

Object versioning starts up at the object level and is disabled by default. Once the object versioning is **enabled it cannot be enabled again, it can only be suspended and re-enabled again.**



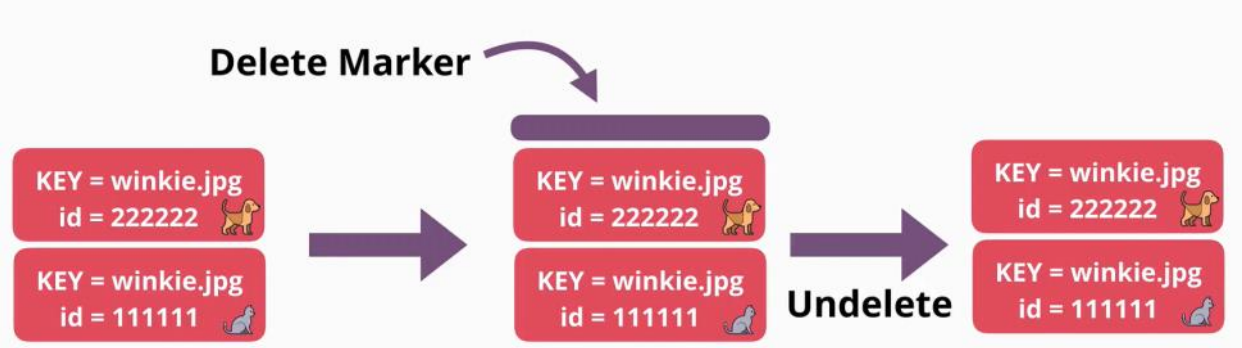
Without versioning enabled on a bucket, each object is identified solely by the object key, its name (which is unique inside the bucket). If an object is modified, the new version of the object replaces the old one.

Versioning allows to store multiple versions of the same object within a bucket. **Any operation that modify objects will generate a new version of that object.**

The latest version of an object is called **the current object** or **latest version.**

When requesting an object from a version bucket, and no version is specified then the current version is returned. Other versions can be requested by **specifying the ID**.

When an object is deleted in a version bucket, the current version of the object is marked as deleted, but the actual object is just hidden. **The delete marker is a special version of an object which hides all previous versions of that object.**
The delete marker can be deleted, which results in restoring the last current version of that object.



To truly delete an object, the version ID needs to be specified. If the current version of the object is deleted, and there are other versions of that object present, the most current version before the current version becomes the current version object.

Space within an bucket **is consumed by all the versions of objects**. (being billed for all). Even if you suspend the version service, the objects and their version withing that bucket are billed.
For zero costs, the bucket should be deleted.

MFA Delete (multi factor auth)

MFA Delete is enabled within the versioning configuration of a bucket.

- when is enabled it means that MFA is required to change bucket versioning state
- MFA is required to delete versions

- Enabled in **versioning configuration**
- MFA is required to change bucket **versioning state**
- MFA is required to **delete versions**
- Serial number (MFA) + Code passed with API CALLS

S3 Performance Optimization

Wednesday, July 7, 2021 12:32 PM

When uploading to S3, and object is loaded as a single blob of data in a single stream

- a file is uploaded and becomes an object via the **S3:PutObject API**. **This happens a single stream**
- **if a stream fails, the whole upload fails**, which needs a full restart
- the speed and reliability of the upload process is limited by the limit of 1 stream of data
- when transferring data between two points, the speed that the data will transfer matches the lower speed from the two points
- any upload with the **put** method, has a limit of maximum 5GB

A solution for this is **multipart upload**

- improves the speed and reliability of uploads to S3
- it does its job by dividing the data into individual parts
    - the minimum size of the original blob of data is 100 MB
    - a blob can be split into maximum of 10,000 parts
        - each part can range from 5MB to 5GB
        - the last part is called a leftover, and can be smaller than 5MB if needed
    - each part can fail and be restarted in isolation
        - the whole data transfer does not need to be restarted if a part fails
    - the transfer rate in this case is the speed of transfer of all parts

Accelerated Transfer

- we do not have control of the path, that the transfer is made from point A to point B
- transfer acceleration uses the network of AWS edge location
- S3 transfer acceleration needs to be enabled, as by default is switched off
    - there are few constrains to enabled it
        - the bucket name needs to not contain any periods
        - needs to be DNS compatible in its naming
- when the transfer acceleration is enabled, the data being uploaded instead of going to the S3 directly it immediately enters the closest best performing AWS edge location
- from here the data is transferred through AWS network, a network that is fully controlled by AWS - direct link
    - the private AWS network is designed to link regions to other regions

s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html

Key Management Service (KMS)

Wednesday, July 7, 2021 1:43 PM

Key Management Service is not part of S3, but is used in most AWS services **that use encryption**. and S3 uses encryption.

KMS is a **regional and public service.** (is a separate product in each region)

- it is public that you need permission to access it
- allows you store, create and manage **cryptographic keys**
- these keys are used to convert plain text to cipher text and vice-versa
- can handle symmetric and asymmetric keys
- can perform cryptographic operations such as encryption and decryption
- **the keys never leave the product** - **provides FIPS 140-2 (Level 2 or 3)**
    - the keys are locked inside KMS service
    - its function is to secure and make sure that the keys never leave the service

Customer Master Key (CMK)

The main thing that KMS manages is **CMS** (Customer master keys)

- are used by KMS for cryptographic operations
- can be used by entities, applications and AWS services
- CMS are logical - a container for the actual physical master keys
- The contain:
    - a key ID - used as an identifier of the key
    - creation date
    - a key policy
    - a description
    - a state of the keys: active or inactive
- can be generated or imported by KMS
- **can be used to encrypt or decrypt data up to 4KB**

The process:

1. After peaking a region a key can be created (a CMK)
2. A CMK logical container is created which contains physical baking key material
    a. This is what KMS creates, stores and manages
    b. the CMK are never stored in a persistent store if they are not encrypted (KMS encrypts the key before storing it on the disk)
3. Request to encrypt some data
    a. this is achieved by making an encryption call, providing a key and the data to be encrypted
4. KMS accept the data and if the entity has permissions to use the CMK
    a. the KMC decrypt the key
    b. uses the decrypted key to encrypt the plain text data to cypher text
5. KMS returns the data to the entity

1. To decrypt the data, an entity needs to request the decrypt
2. KMC does not need to be told which CMK to use to decrypt the data
    a. that information is encoded into the cypher text of the data that needs to be decrypted
3. KMS will decrypt the CMK
4. Will use de decrypted CMS to decrypt the cypher text
5. Will return the data to the entity

During both processes the **CMK does not leave the KMS, the decrypted CMK is not stored on the**

disk and at every step the entity needs permissions to perform this operations

- **EACH OPERATION NEEDS DIFFERENT PERMISSIONS**
  - **Administrative permissions** - delete and create keys
  - **Usage permissions** - encrypt and decrypt

Role separation is where different roles are given different access rights within a product.

Data Encryption Keys (DEKs)

DEKs are another type of key which KMS can generate using a CMK.

- can be used to encrypt data larger than 4 KB
- this is linked to a specific CMK
- KMS does not store the DEK in any way
  - it provides it to the key to the service or entity that needs it and then it discards it

The process:

1. When a data encryption key is generated KMS provides two versions of that encryption key
   a. plain text version
   b. cypher text version
2. Encrypt data with the plain text version key
   a. once the encryption is done, the plain text key is discarded
3. The encrypted key and the encrypted data are stored only

KMS does not perform the encryption or decryption on the data larger than 5 KB

- the identity or the service using KMS does

KMS does not track the usage of the data encryption keys

S3 when using encryption, generates a data encryption key for every object!

Key Concepts

1. **CMKs** are isolated to a region and they never leave that region
2. The never leave the **KMS** service. They cannot be extracted
3. There are two types of CMKs:
   a. **AWS managed CMKs**
      i. created automatically by AWS when a service uses KMS encryption
   b. **Customer Manages CMKs**
      i. are created explicitly by the customer
      ii. are much more configurable
   c. **both support key rotation**
      Rotation is the process when the physical backing material is changed
      - with **AWS Managed Keys** this cannot be disabled and is set to rotate material every 1,095 days (3 years)
      - with **Customer Managed Keys** the rotation is optional and happens once a year if its enabled
        - CMK contains the current backing key and all the previous backing keys used for backing material
        - aliases are a shortcut to a particular CMK - are also per region

Every **CMK has a customer policy**

- **KMS has to be explicitly told they keys trust the AWS account they are in.**
  - the key trusts an account, so that the account can manage the keys
  - **IAM is trusted by the account, and the account needs to be trusted by the key**

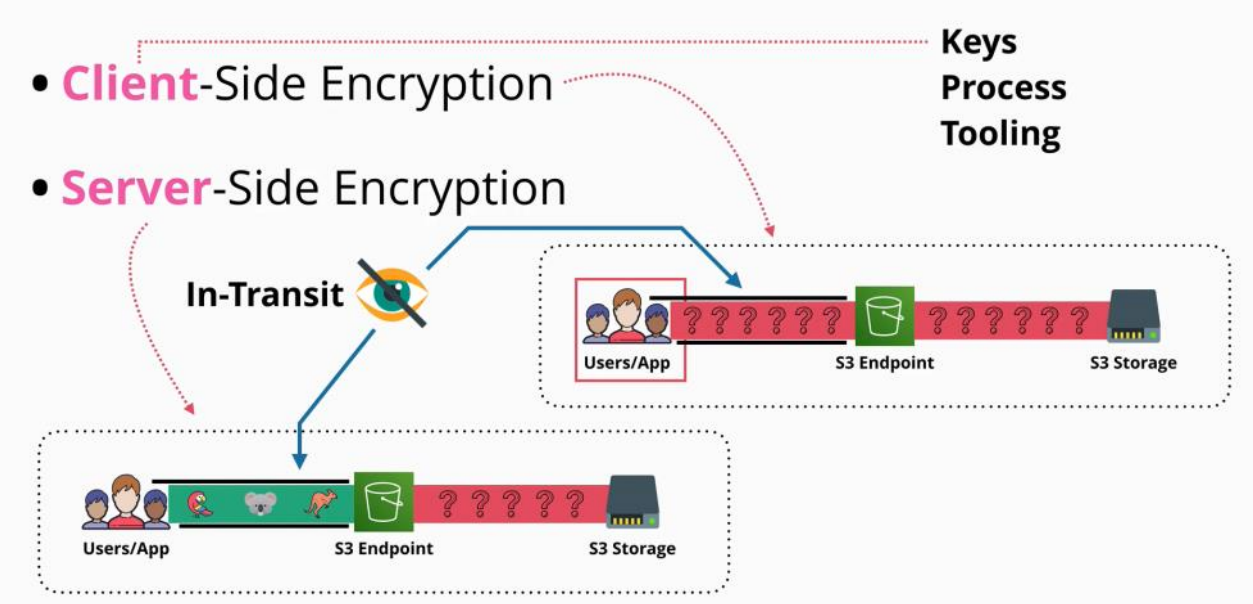
```
# Generate Battleplans

echo "find all the doggos, distract them with the yumz" > battleplans.txt

# Encrypt

aws kms encrypt \
    --key-id alias/catrobot \
    --plaintext fileb://battleplans.txt \
    --output text \
    --query CiphertextBlob \
    --profile iamadmin-general | base64 \
    --decode > not_battleplans.enc

# Decrypt

aws kms decrypt \
    --ciphertext-blob fileb://not_battleplans.enc \
    --output text \
    --profile iamadmin-general \
    --query Plaintext | base64 --decode > decryptedplans.txt
```

S3 Encryption

Wednesday, July 7, 2021 3:49 PM

You do not define encryption at the bucket level

- we define encryption at the object level
  - each object inside a bucket can use **different encryption settings**
- There are two main method of encryption:
  - **client side encryption**
    - the data is encrypted by the client, before they are uploaded
    - the data is cypher text the entire time in transit and on the final destination
    - the data cannot be seen in plain text at any given time
    - **this type of encryption is the client responsibility:**
      - generate a key
      - store that key
      - know what key belongs to what file
      - encryption is done by the client before is uploaded to S3
      - S3 is used for storage **only**
  - **server side encryption**
    - the data is encrypted in transit using HTTPS, the object themselves are not encrypted
    - when it reaches the **S3 endpoint** is in plain text
    - once the data hits **S3 is then encrypted** by the S3 infrastructure
    - with this type of encryption you allow S3 to perform **some or all those process**
  - **both of them refer to encryption at REST** (when they are stored on the disk)
- encryption in standard comes by default with S3 service - data is encrypted in transit

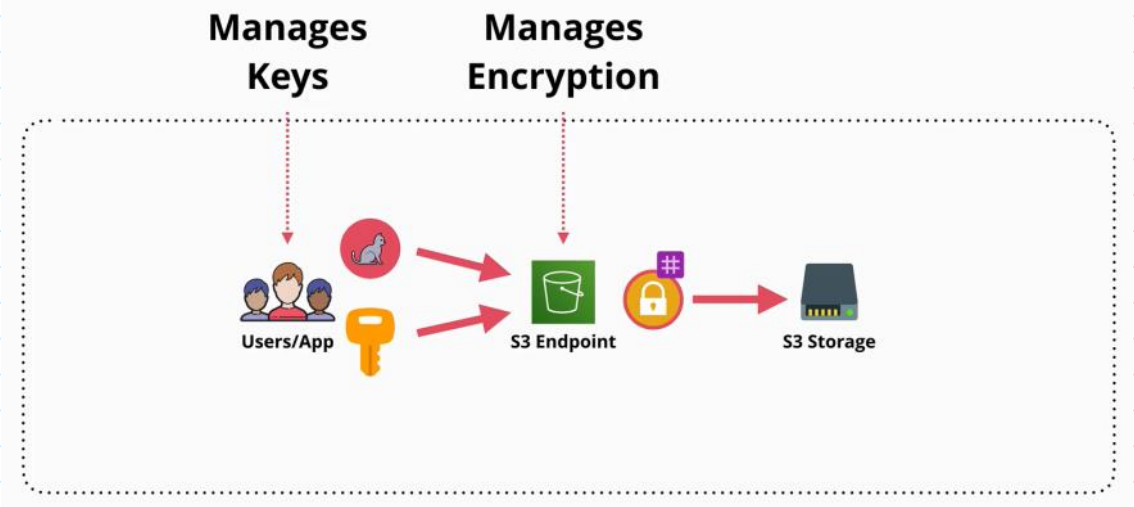


Server side encryption

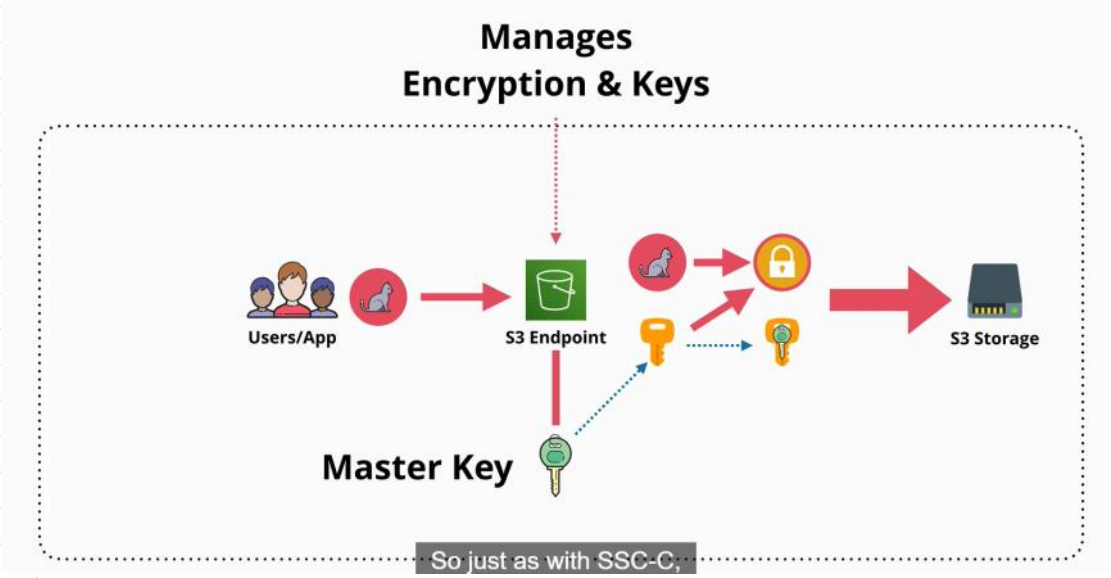
There are 3 types of server side encryption:

1. **SSE-C** (Server side encryption with customer provided keys)
   - The customer is responsible to manage the encryption and decryption keys
   - The server manages the encryption/decryption process
   - When using this service you are required to provide the **object** and the **key**
   - When the key and the object arrive on the server, the object is encrypted using that key and store the object on the server with a hash of that key (the key is discarded after is

hashed)
    - The hash of the key is one way (cannot be used to get the key). It is used in the decryption process to check the authenticity of the key provided to decrypt the object

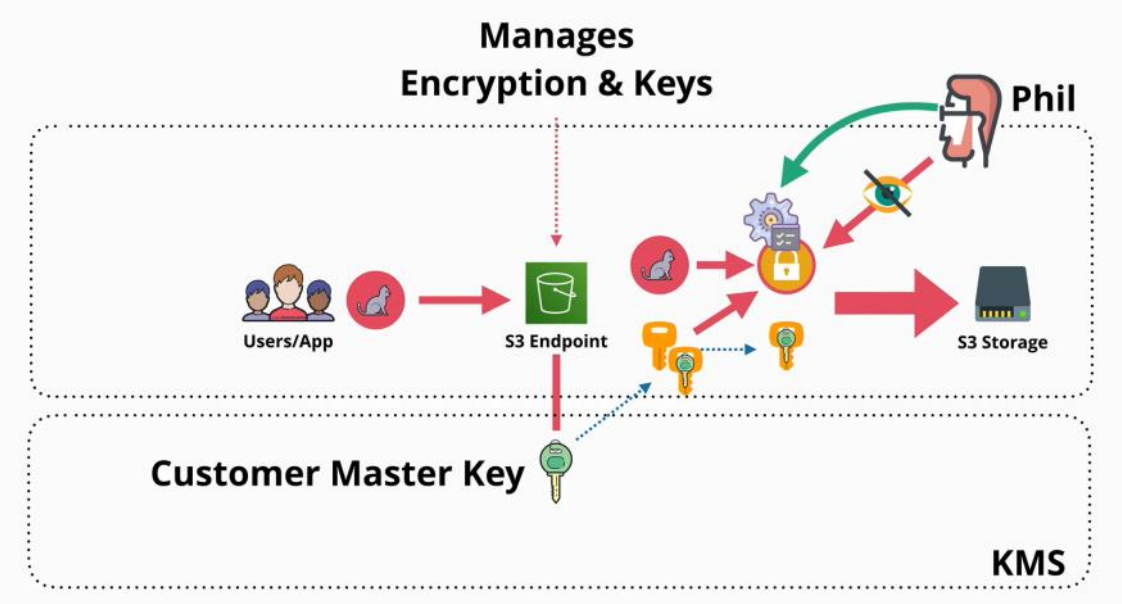


2. **SSE-S3** (Server side encryption with Amazon S3 managed keys)
    - With this service, Aws manages the encryption and the decryption process as well as the key **generation and management**
    - In this case, the client provides just the data in plaintext
    - AWS creates a **master key** to be used for the plaintext data to be encrypted
        - Is handled end to end by AWS can cannot be configured
    - each object will have a unique key
        - AWS generates a key for each object, and uses that key to encrypt that object
        - the **master key** is used to encrypt the key of the object after the object was encrypted
        - the un-encrypted version of that key is discarded, and the encrypted key is stored with that encrypted object



    - **It does come with some disadvantages:**
        - **you cannot control the rotation of the key**
        - **you cannot use role separation**
        - **you cannot control or manage the keys, and you cannot control who uses those key**
3. **SSE-KMS** (Server side encryption with customer master keys stored in AWS key management service)
    - is very much like SSE-S3 but it allows the KMS service to manage the keys
    - every object that is uploaded to a bucket requires a CMK

        - this CMK is used to generate one unique encryption key for every object that is encrypted using the SSE-KMS
    - This gives the option of not using the default CMK that S3 creates. You can create and use a **customer managed CMK** that allows:
        - role separation
            - S3 administrators - not being able to read the objects (decrypt the objects)
        - key policies



Summary

Method	Key Management	Encryption Processing	Extras
Client-Side	YOU	YOU	
SSE-C	YOU	S3	
SSE-S3	S3	S3	
SSE-KMS	S3 & KMS	S3	Rotation Control Role Separation

Default Bucker Encryption

Using the **x-amz-server-side-encryption** header, objects are encrypted. Without the header, the objects will not use encryption.

When the headers is not used, but as default the **AES256** (Advanced Encryption Standard), then SSE-S3 would be used when nothing is set on an object level.

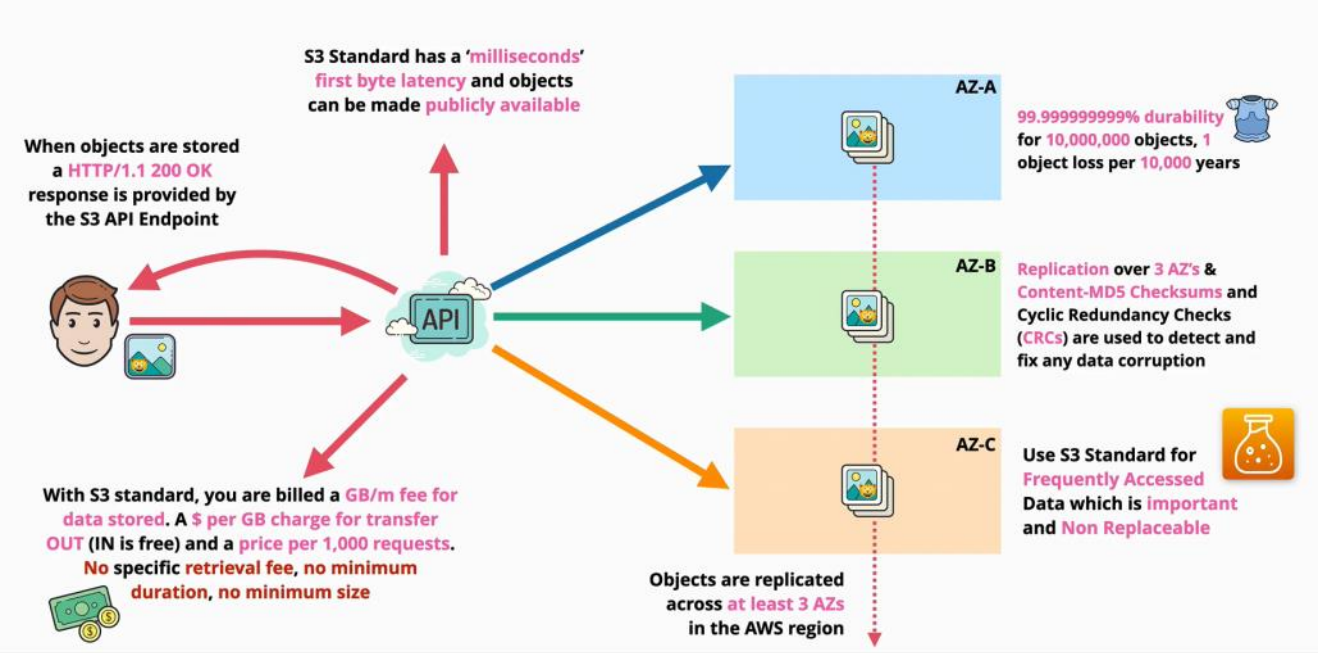
S3 Storage Classes

Thursday, July 8, 2021 1:51 PM

S3 Standard

The default S3 storage class is called **S3 Standard**.
When using S3 standard:

- objects are saved in at least 3 availability zone
- this means that S3 standard is able to cope to availability zone failures
- this level of replication means:
  - 11 nines of durability: for 10 million objects within an S3 bucket, on average you might lose an object every 10,000 years
  - uses MD5 checksums with cyclic redundancy to detect and resolve any data issues
- when objects are stored successfully and durably, a **HTTP/1.1 200 ok** response is provided to the **S3 API Endpoint**
- billing a **GB/month fee for data stored**, **a dollar/GB charge for transfer OUT** **(IN is free)** and **a price per 1000/requests**
  - **there is no specific retrieval fee, no minimum duration and no minimum size**
- access to S3 is instantly - has a milliseconds first byte latency (data requested is available in milliseconds)
  - objects can be made publicly available
- ★ **should be used for frequently accessed data**

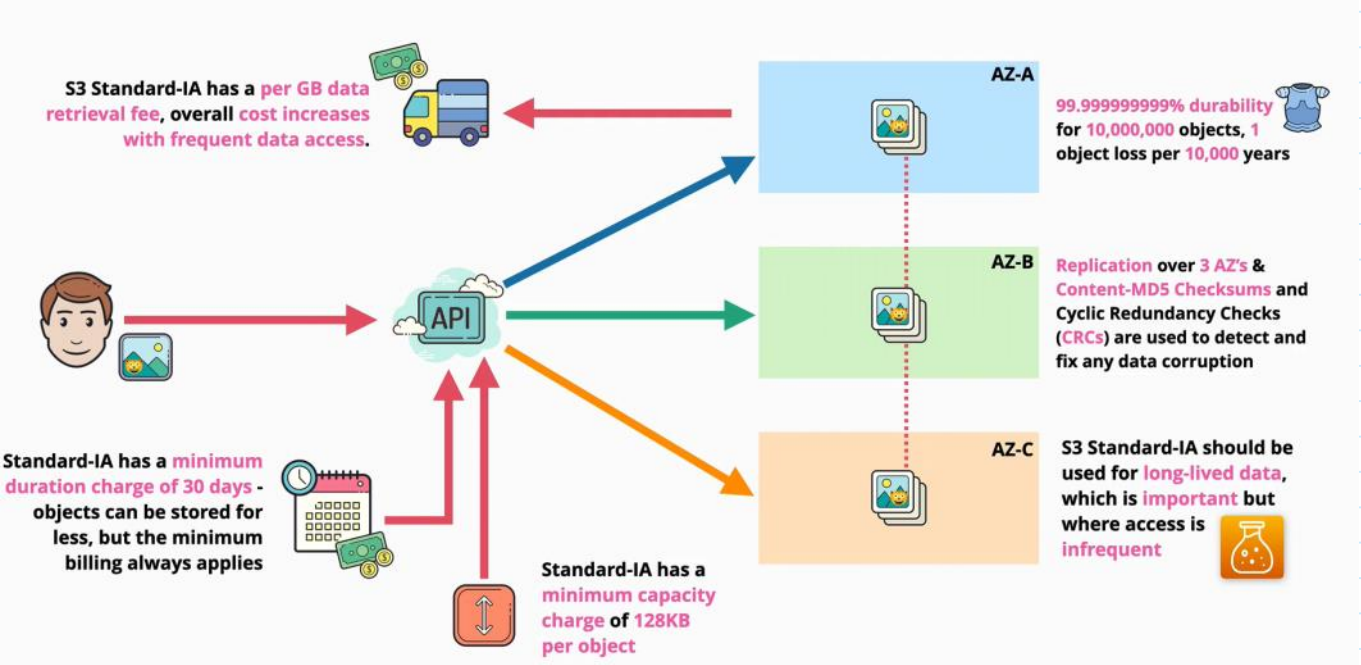


S3 Standard Infrequent Access (S3 Standard-IA)

Standard-IA shares most of the architecture and characteristics with S3 Standard.

- data is replicated over at least three availability zone in the region
  - 11 nines of durability: for 10 million objects within an S3 bucket, on average you might lose an object every 10,000 years
  - uses MD5 checksums with cyclic redundancy to detect and resolve any data issues
- durability, availability, first byte latency and can be made publicly available are the same as in S3 Standard
- Costs are **cheaper** in S3 Standard:
  - request charge and data transfer out has the same price
  - GB/months is about half the price
  - retrieval fee has a cost of GB for retrieval which is expensive for frequent accessed data
  - **this class is designed for infrequently accessed data**

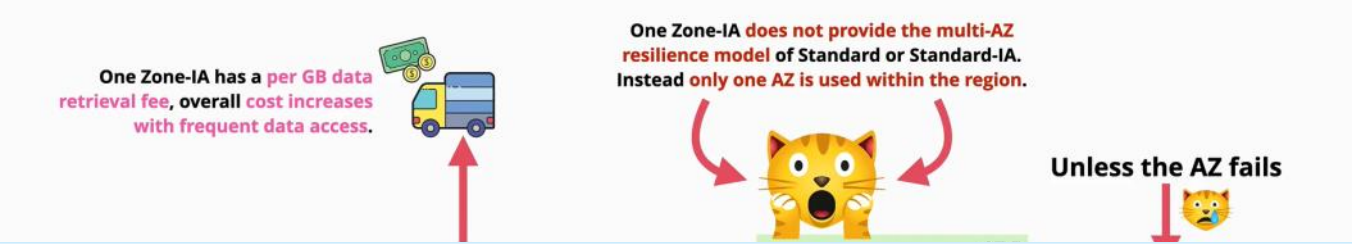
- has a minimum duration charge of 30 days - objects can be stored for less time, but the minimum billing pay applies anyway
- has a minimum capacity charge of 128KB/object. So objects that are smaller than this size, are billed for 128KB
- ★ **this class is cost effective for data as long as the objects are not accessed frequently, as long as the data is not stored for short term and the objects stored are not tiny in size**

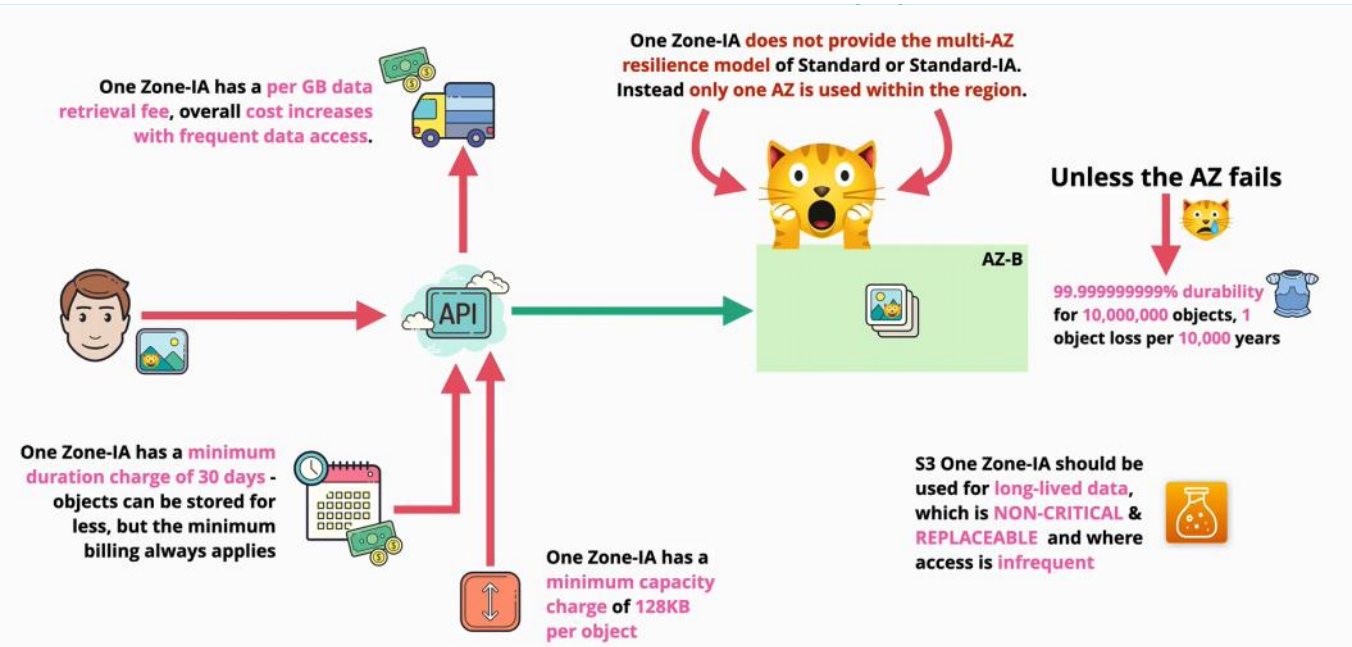


S3 One Zone - Infrequent Access (S3 One Zone - IA)

This class is similar to S3 Standard-IA in many ways.

- it is cheaper than S3 standard and S3 Standard - IA
- retrieval fee has a cost of GB for retrieval which is expensive for frequent accessed data
- has a minimum duration charge of 30 days - objects can be stored for less time, but the minimum billing pay applies anyway
- has a minimum capacity charge of 128KB/object. So objects that are smaller than this size, are billed for 128KB
- ★ **data stored in this class is stored in only one availability zone**
  - **does not have the replication as the other two classes**
  - **less resilience**
  - **the durability is the same - 11 nines**
    - 11 nines of durability: for 10 million objects within an S3 bucket, on average you might lose an object every 10,000 years
      - data is still replicated in that availability zone (multiple copies of data exist) but if the zone fails the data is lost
    - uses MD5 checksums with cyclic redundancy to detect and resolve any data issues
- ★ **should be used for long-lived data (as the size and duration minimums are in place)**
  - not frequent accessed data
  - not small objects in size (less then 128KB)
- ★ **should be used for data that is not critical! (data that can be easily replaced)**

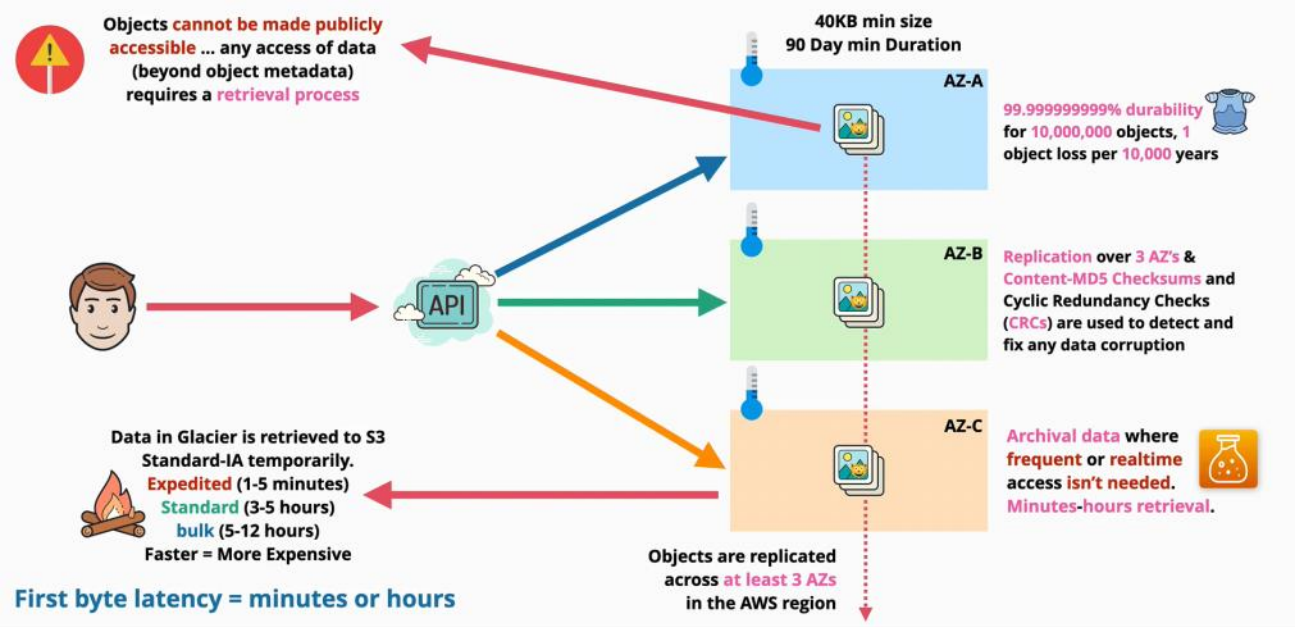




S3 Glacier

S3 Glacier has data replicated over at least three availability zone in the region.

- this level of replication means:
  - 11 nines of durability: for 10 million objects within an S3 bucket, on average you might lose an object every 10,000 years
  - uses MD5 checksums with cyclic redundancy to detect and resolve any data issues
- it is really cost effective as it has the GB/month fee about a fifth of the S3 Standard
- objects stored in S3 Glacier are called **cold objects**
  - ★ **not ready to be used - they are not immediately available**
    - **to access them a retrieval process needs to be made**
    - **when retrieved the objects are stored in S3 Standard - IA temporarily** - they are removed after they are accessed
    - every retrieval process is billed
      - **Expedited** - data is available in 1 - 5 minutes (is the most expensive)
      - **Standard** - data is available in 3 - 5 hours
      - **Bulk** - data is available in 5 - 12 hours (low cost)
      - Faster = more expensive
      - first byte latency can be from minutes to hours
  - ★ **they cannot be made public**
- has a 40KB minimum billing size
- 90 days minimum billing duration
- ★ **used when the object stored are for archives, where frequent or real time data isn't needed**

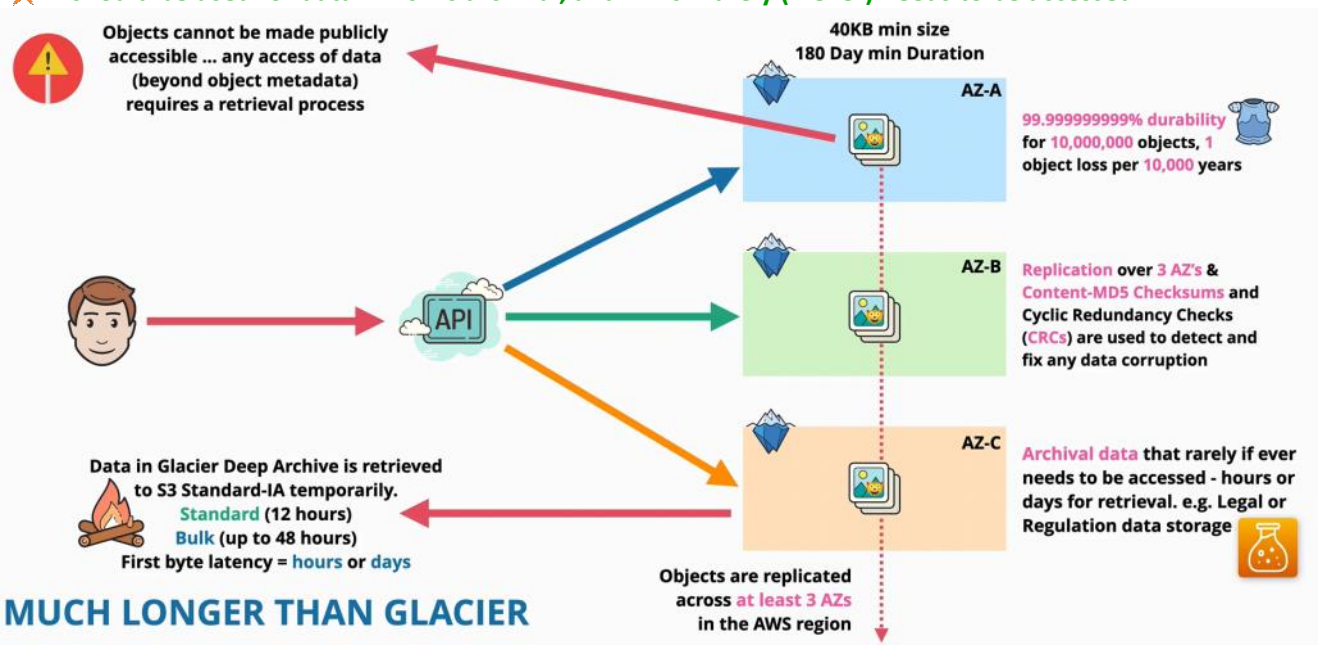


S3 Glacier Deep Archive

S3 Glacier Deep Archive has the price as forth of the S3 Glacier.

- has data replicated over at least three availability zone in the region.
    - this level of replication means:
        - 11 nines of durability: for 10 million objects within an S3 bucket, on average you might lose an object every 10,000 years
        - uses MD5 checksums with cyclic redundancy to detect and resolve any data issues
- objects stored in S3 Glacier are called **freeze objects**
    - has a 40KB minimum billing size
    - 180 days minimum billing duration
- objects cannot be made publicly available
- access of the data requires a retrieval process
    - **standard -** up to 12 hours
    - **bulk -** up to 48 hours
    - first bye latency = hours or days

★ - **should be used for data which is archival, and which rarely (if ever) needs to be accessed**



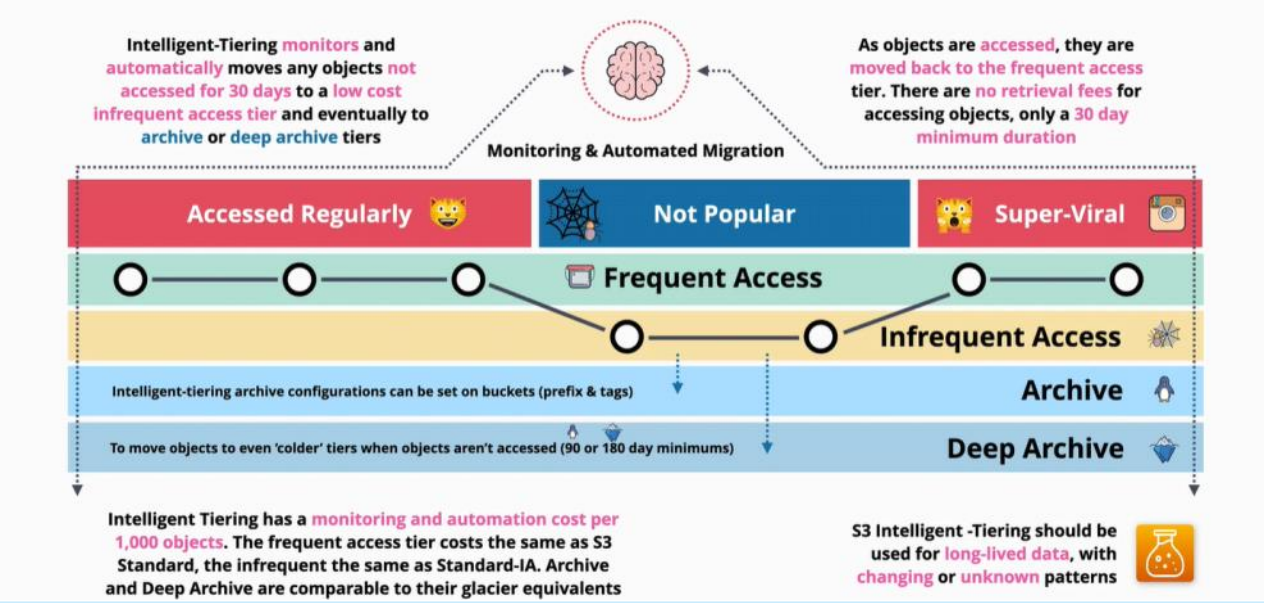
S3 Intelligent-Tiering

Is different from all the other S3 classes

It contains 4 different tiers of storage, there are a number of ways that an object can be stored:

- frequent access tier (like S3 standard)
- infrequent access tier (like S3 standard IA)
- archive (Glacier)
- deep archive (Glacier Deep Archive)

- you do not have to worry about moving objects between tiers
  - the system does it for you
    - the system monitors the usage of the object
      - ★ □ if an object is not used for 30 days it is moved to from frequent access tier to infrequent access tier
      - ★ □ if the object is used even less can be moved to the archive tiers, **but it is optional**
        - 90 days for archive
        - 180 days for deep archive
        - **for this two tires, the access to the objects is not instantly, it is a retrieval time to access them**
      - ★ □ if an object is in the infrequent tier, and the system detects that it is used frequent again it is moved back to frequent tier
- Intelligent tiering has a monitoring and automation cost/1000 objects - instead of the retrieval cost
- the cost of the tires are the same as the S3 storage classes
- ★ **it is ideal for long-lived data, because it has a 30 day minimum billing period, and is also ideal where the usage of objects is changing or is unknown**
  - **ideal for changing access frequency**



S3 Lifecycle Configuration

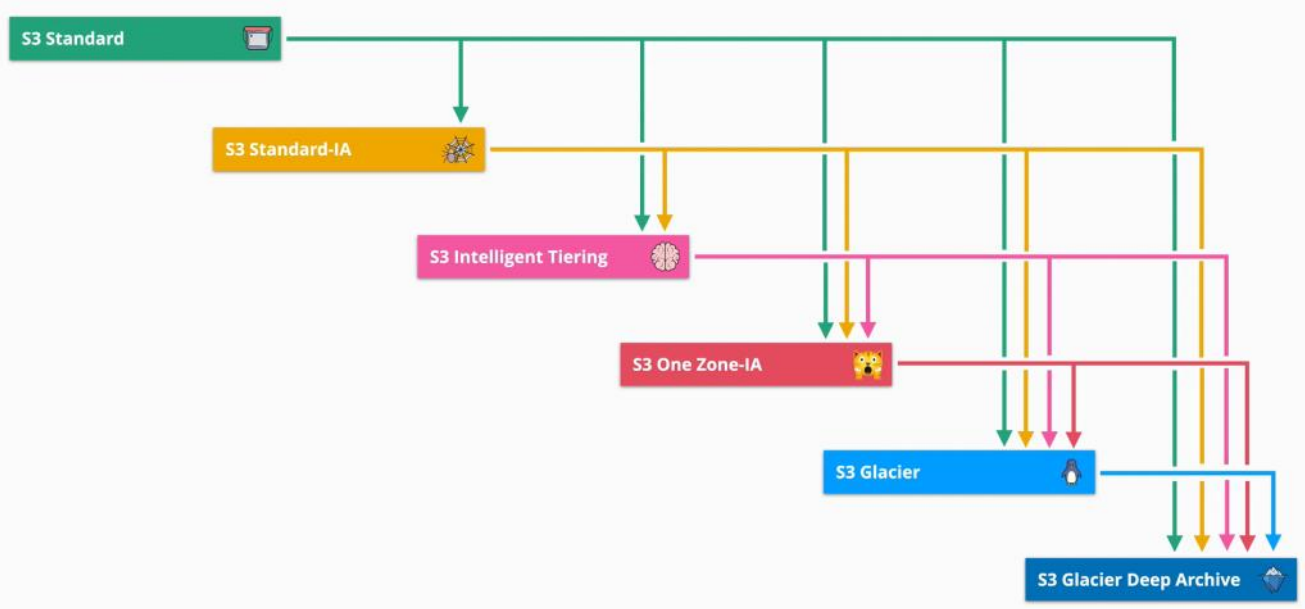
Thursday, July 8, 2021 3:27 PM

Lifecycle rules can be created on S3 buckets which can automatically transition or expire objects in that bucket.

- a great way to optimize costs
- a configuration for S3 buckets is a set of rules which apply to a particular bucket
  - rules consists of actions
- the rules can apply to a bucket or a set of objects defined by prefixes or tags
- there are two types of actions
  - **transition** actions
    - which can change the storage class
  - **expiration** actions
    - which can delete effected objects or version of objects
    - automate the deletion of objects
- these rules are not based on access - not moved according to their accessing intervals, they are moved between classes or they are deleted based on hardcoded rules

Transitions

- lifecycle transitions are like a waterfall
  - starting with S3 Standard, followed by S3 Infrequent Access, S3 Intelligent Tiering, S3 One zone IA, S3 Glacier and S3 Glacier Deep Archive
  - transition flows in a downward direction **only**



- when transitioning **smaller objects** from S3 Standard to IA or Intelligent Tiering because all of these have a minimum storage billing
- to move objects automatically from S3 standard to S3 Standard IA and S3 one Zone IA days, the objects **need** to be 30 days in S3 Standard
- To transition objects from S3 Standard IA, S3 One Zone IA and S3 Intelligent Tiering to Glacier or Glacier Deep Archive, they objects need to be in 30 days in in the S3 Standard IA, S3 One Zone IA and S3 Intelligent Tiering
- You can define two rules:
  - one rule to move an object from Infrequent Access tier
  - second rule to move it from Infrequent Access tier to Glacier
  - **these two rules won't suffer the 30 days minimum**, but it will be charged for the minimum

duration billable duration

- if a single rule is used **then you need to wait for the minim period between transitions**

S3 Replication

Thursday, July 8, 2021 4:01 PM

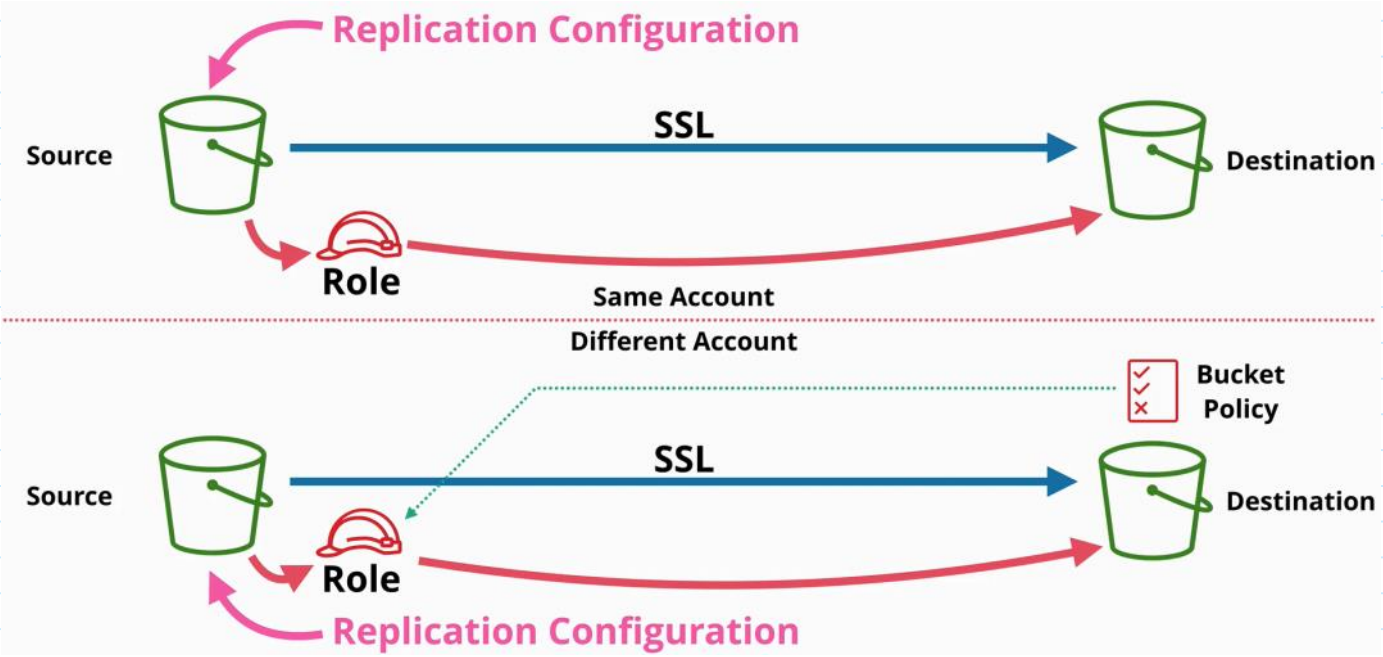
S3 replication is feature that allows the configuration of replication of objects between a source and destination, S3 bucket.

There are 2 types of replication supported by S3:

1. Cross Region Replication (CRR)
    - allows the replication of objects from s source bucket to a destination bucket in **different AWS regions**
    - **needs versioning enabled**
2. Same Region Replication (SRR)
    - allows replication of objects from a source bucket to a destination bucket in the **same region**

Architecture

- **replication configuration is applied on the source bucket**
    - the configuration states the replication from a source bucket to a destination bucket
    - the configuration states the following
        - the destination bucked to be used for the replication process
        - an IAM role to use for the replication process
            - The roles permission policy gives permission to read objects on the source bucket and replicate those objects to the destination bucket
        - the replication is encrypted
    - **same AWS account replication**
        - in this scenario both buckets are owned by the same AWS account
        - they both trust the same IAM service and the IAM role
        - the role automatically has access to both the source and the destination buckets, and to the permission policy that grants the access
    - **different AWS accounts replication**
        - the destination bucket because is in a different AWS account, does **not trust** the other account or the role that is used to replicate the bucket contents
        - the role that is designed to perform the replication isn't by default trusted by the destination AWS account
            - a bucket polity is added on the destination bucket which allows the role in the source account to replicate objects into it



Replication options

- all objects or a subset of objects can be replicated
- the store class of the destination bucket can be selected
    - by default is use the same class as the source bucket
        - for a backup a One Zone IA is good
- ownership of the buckets
    - by default the objects in the destination bucket are owned by the source account
- replication time control (RTC)
    - S3 Replication Time Control is designed to replicate 99.99% of objects within 15 minutes after upload, with the majority of those new objects replicated in seconds. S3 RTC is backed by an SLA with a commitment to replicate 99.9% of objects within 15 minutes during any billing month.

Considerations

1. replication is not retroactive
    - you enable replication with two buckets: a source and a destination bucket
    - only after the source and the destination bucket are named, are objects replicated from source to destination
    - ★ if you enable replication on a bucket that already has objects in it, the existing objects will not be replicated
    - ★ both the source and the destination bucket need to have versioning enabled
2. it is a one way replication
    - ★ from source to destination
        - if objects are added in the destination bucket, they will not be added to the source bucket
3. objects can be replicated:
    - if they are unencrypted
    - or I they are encrypted with SSE-S3 or SS3-KMS
        - cannot replicate objects using SSE-C - because it needs the keys
4. the source bucket owner needs permission on the **all** objects that will be replicated
    - if changes are made in the source bucket by lifecycle management, they will not be replicated to the destination bucket
    - only user events are replicated
5. system events cannot be replicated
6. glacier and glacier deep archive objects cannot be replicated
7. deletes are not replicated

Why use replications?

1. for same region replication (SRR)
    a. log aggregation - merge multiple logs
    b. synchronize production and test accounts
    c. resilience with strict sovereignty requirements
2. for cross region replication (CRR)
    a. global resilience improvements
        i. backups of data copied in different AWS regions to cope with large scale failure
    b. reduce latency
        i. for better performance

S3 PreSigned URLs

Friday, July 9, 2021 1:19 PM

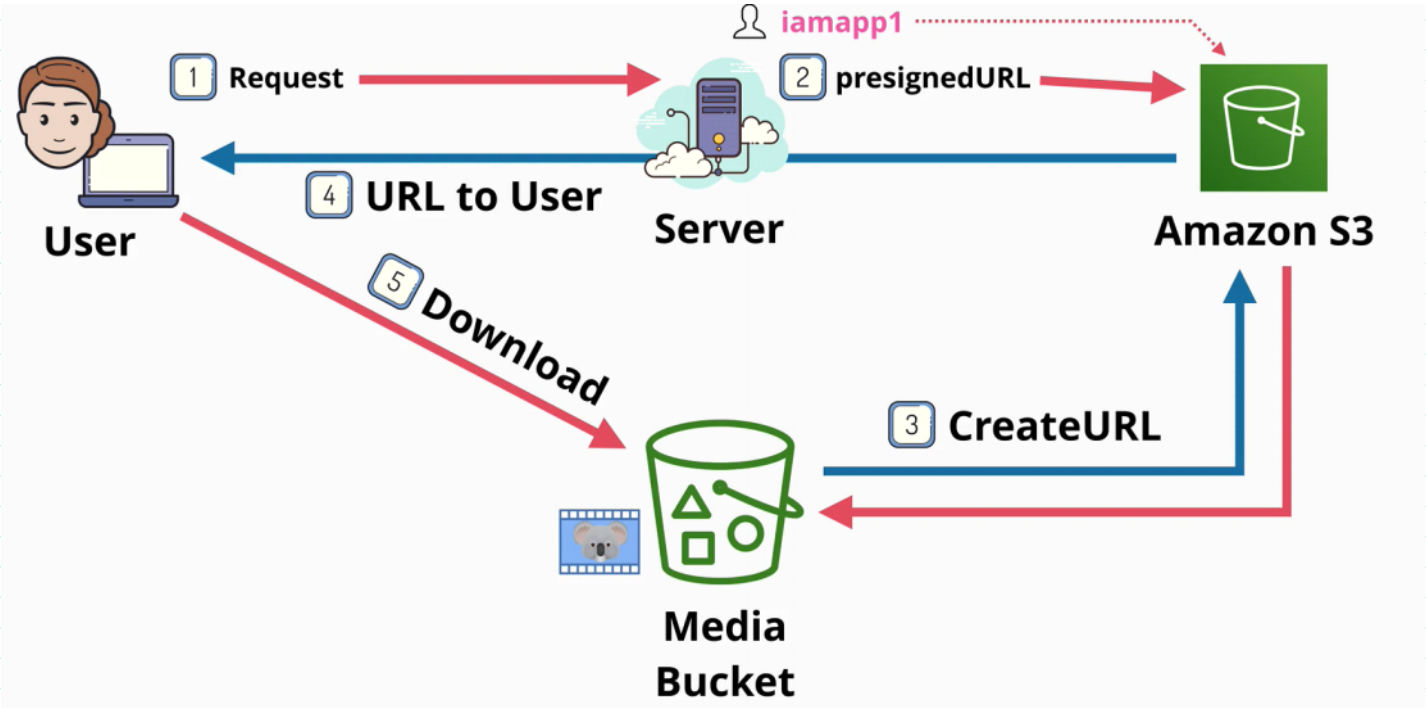
Presigned URLs is way of giving another entity, or application access to an object inside an S3 bucket, using **your credentials** in a **safe and secure way.**

- an S3 bucket that does not have any public access configured - its state is in a private configuration
  - this means, that is an entity needs to access the bucket or its objects needs to authenticate and have permissions to do so
- we can allow access to objects within a bucket or buckets to an unauthenticated entity in three different ways
  1. give the entity an AWS identity
  2. give the entity some AWS credentials
  3. make the bucket or the objects public
  4. presigned URLs
     - the **iamadmin** user make a request to S3 to generate a presigned URL
       - the iamadmin user needs to provide its credentials
       - the bucket name
       - an object key
       - expiry date and time
       - specify how the objects will be accessed
     - the S3 creates a presigned URL and return it to iamadmin
       - the URL has encoded inside it the details that the iamadmin provided
     - the URL can be passed to the mystery identity
       - the identity can use the URL to access the specific S3 bucket or objects
         - until the URL expires
       - the mystery identity that uses the URL interacts with the S3 as the person who generated it
     - the URL can be used to GET (download) or to PUT (upload)

Example

We have an application server in the cloud, which hosts the application. This is a video processing application. Large video files had been migrated from the application server to a media S3 bucket.
Previously the videos were hosted on the application server and it could control the access of the video files. Hosted in the S3 bucket, either every user needs an AWS identity to access the videos, or the videos need to be public. Neither are a good solution.
However, we can use the presigned URLs, to keep the bucket private and create a iamuser for this application.

1. when an application user interacts with the web application it makes a request
2. the request is for a page that has a video associated with it
3. the app running on the server knows that it can directly return the information that is requests, but the video it is hosted on a private S3 bucket
4. therefore, the application initiates a request to S3 asking to generate a presigned URL for that particular video associated with the web request
5. the S3 services creates an URL which has encoded within the authenticated information for that IAM user, to access the video on short time basis (for 2 3 hours)
6. the S3 service returns the URL to the server application, and the server application returns the response including the video to the end user
7. the web application running on the user device, uses the presigned URL to securely access the particular object (the requested video)



- S3 presigned URL are used when buckets are private for different reasons and the access needs to be controlled.
  Presigned URLs are used to access objects within a private S3 bucket, with the access rights of the entity that generates them. The URLs are time limited and they encode all of the authentication information needed inside. They can be used to upload or download objects.

Exam power UP

1. **you can create a URL for an object you DO NOT have access to**
   - **the generated URL will also have no access to that object**
2. **when using a presigned URL, the permission MATCH the identity which generated it**
   - **matches the identity that generated that URL**
   - **matches the permissions that the identity that generated the URL has at the time when the mystery identity accessed the URL**
   - **cannot generate presigned URL for non existing objects**
3. **DO NOT GENERATE presigned URLs with a role, as the URL stops working when temporary credentials expire**

Commands to create a presigned URL through the command line from the console

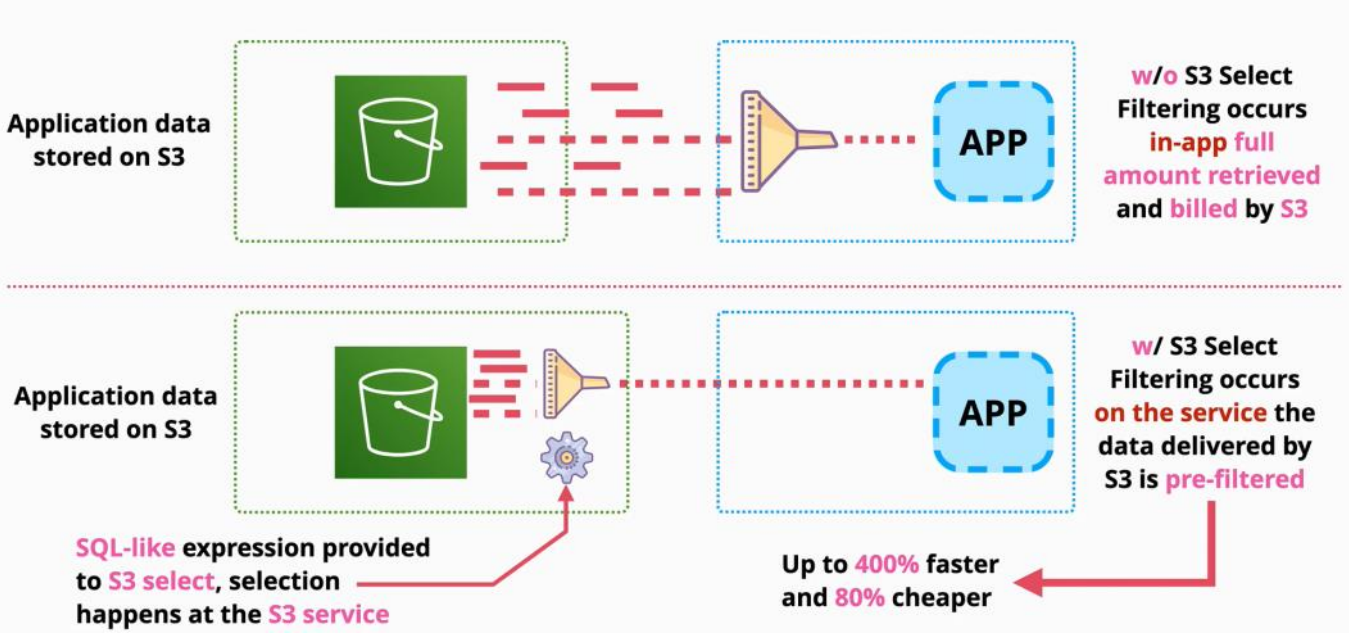
```
aws s3 ls
aws s3 presign objectURI --expires-in 180 // 180 is in seconds
```

S3 Select and Glacier Select

Friday, July 9, 2021 2:41 PM

S3 Select and Glacier Select are ways to retrieve parts of objects, rather than a whole object.

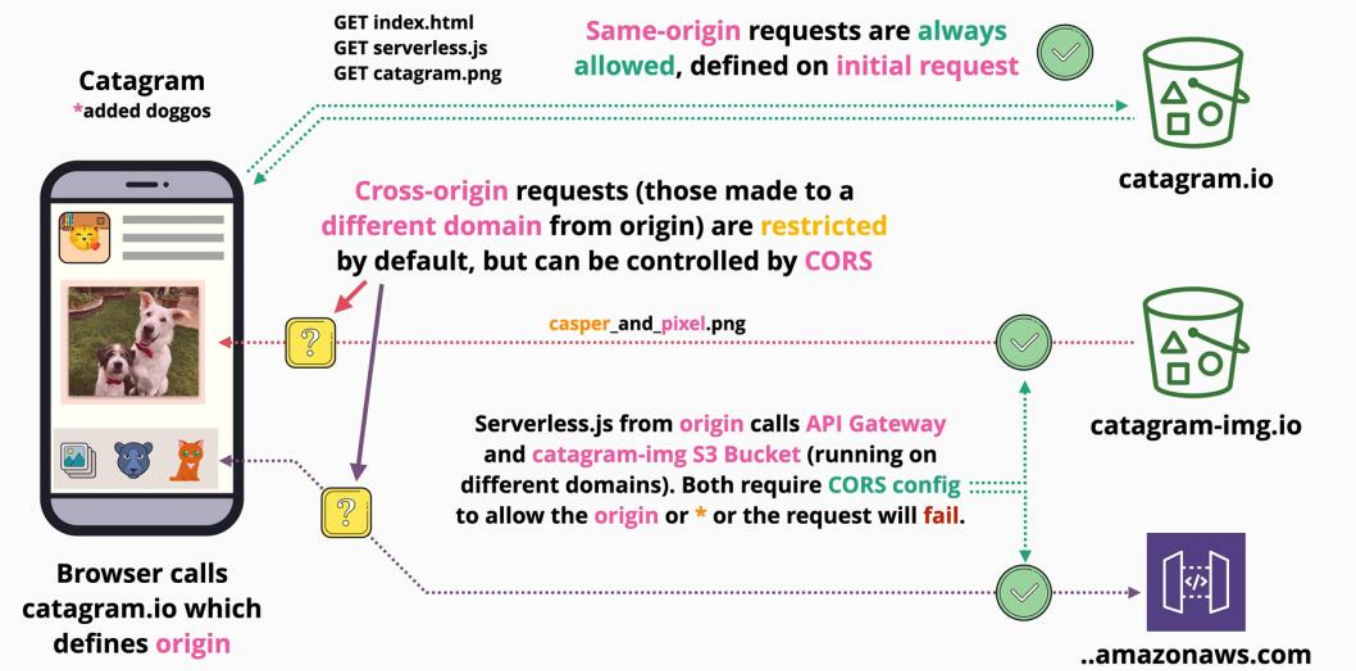
- s3 can store an infinite numbers of object, each object with a maximum size of 5TB
  - if we retrieve a whole 5TB object we consume time and 5TB of transfer
  - this can be filtered at the client side, but the resources had been consumed
- S3 and the Glacier allows the partial download of an object
  - can be obtained with SQL like statement
    - this is a filter on the S3 service
  - allows you to operate in a number of file formats such asL CSV, JSON, Parquet, VZIP2 compression for CSV and JSON



Cross Origin Resource Sharing (CORS)

Friday, July 9, 2021 3:00 PM

- When we open the browser on a device and open a web app such as **catagram.io**, then catagram.io is the origin
  - the website is stored in a bucket
  - in this case the browser makes some web calls to catagram.io
    - get index.html
    - get serverless.js
    - get catagram.png
  - the request get returned without any security issues
    - called same origin request
    - the browsers requests the index.html. which has references to serveless.js and catagram.png
      - all are on the same domain, same origin
  - to load the application, other additional requests
    - an API call is made to an API gateway to get additional application information and pull some images that the users of the application has access to
    - based on the API response, an image is loaded from another bucket
      - there are known as cross-origin requests because they are made to different domains from different origins
      - **by default, cross-origin request are normally restricted** and this can be changed with CORS configuration
  - **CORS** configuration are defined on the other origins
    - if configured they will allow these cross-origin requests
    - they allow from where requests are **allowed**



CORS configuration is made in JSON

There are two different type of requests that can be made, and that require CORS configuration

1. Simple requests
    - directly access a different origin using a cross-origin request
    - as long as the other origin is configured to allow request from the original origin then it is accepted
2. Preflight requests
    - requires a check (known as preflight) which is done in advance to the other origin
    - the browser first sends a HTPP request to the other origin that determinates if the request made is safe to send

Components of CORS Configuration

1. Access Control Allow Origin

- will contain
    - a start (*) allowing any origin to make requests
    - a particular origin to make requests

2. Access Control Max Age

- determinates how long preflight requests can be cashed
    - how long the origin can communicate with the other origin, until another preflight request needs to be made

3. Access Control Allow Methods

- contains a wild card or a list of methods that can be used for cross-origin requests
- GET, PUT or DELETE

4. Access Control Allow Headers

- can be contained in a CORS configuration and within the response to a preflight request
- used to indicate witch HTPP headers can be used with the actual request

S3 Events

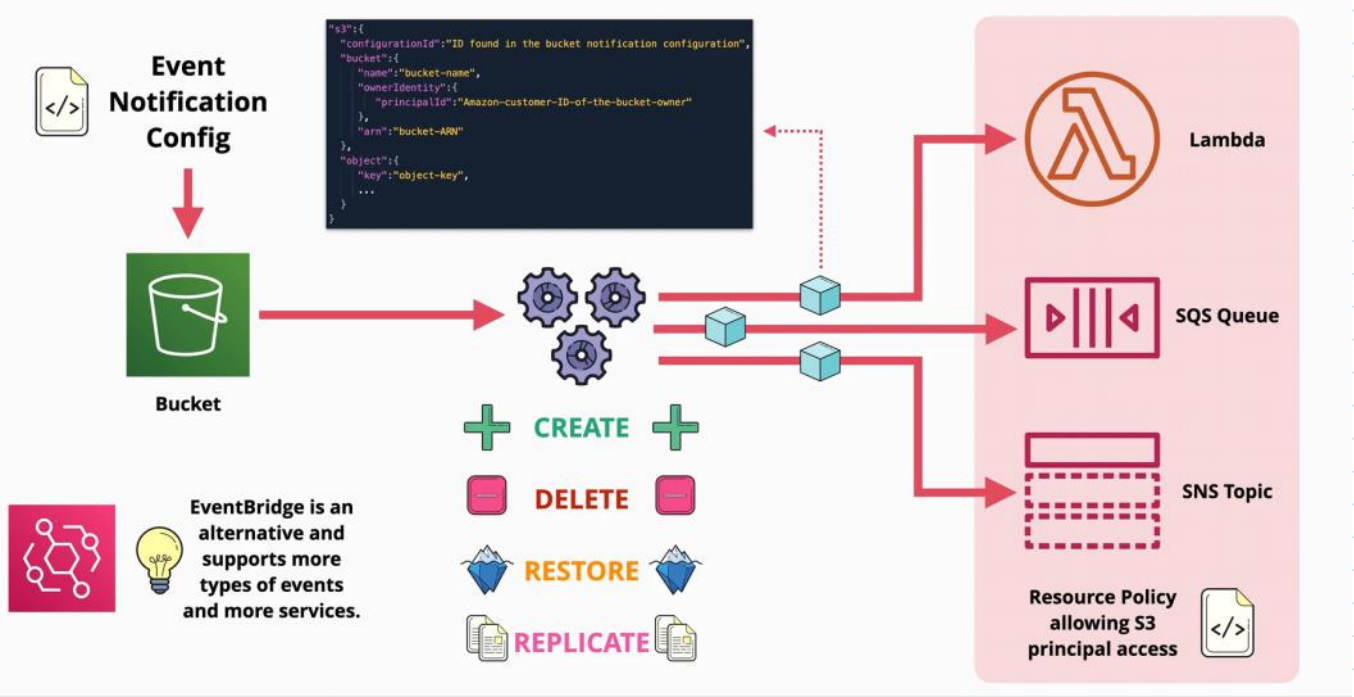
Friday, July 9, 2021 3:52 PM

S3 Events allows the creation event notification configuration on a bucket.

- when enabled a notification is generated when a certain thing occurs within an S3 bucket
  - these can be delivered to different destinations (SNS, SQS and Lambda Functions)
  - this way you can event driven processes which occur as a result of different things happening within S3
- various types of events are supported:
  - when objects are created (by PUT, POST, COPY and multi-part upload operations is completed)
  - on object deletion (Delete an object, or when a Delete marker is created)
  - restore actions (from Glacier (post initiated, or completed))
  - replication (when an objects is no longer tracker, operations missed, operation failed, etc)

Process

1. create a bucket
2. create an event notification configuration
3. when events occur and are stipulated in the notification configuration, notifications are sent to the configured destination
   - events are generated by the S3 service (S3 principle)
   - we need to add resource policies on each destination service - allowing the S3 service to interact with them
   - the events themselves are JSON objects



EventBridge

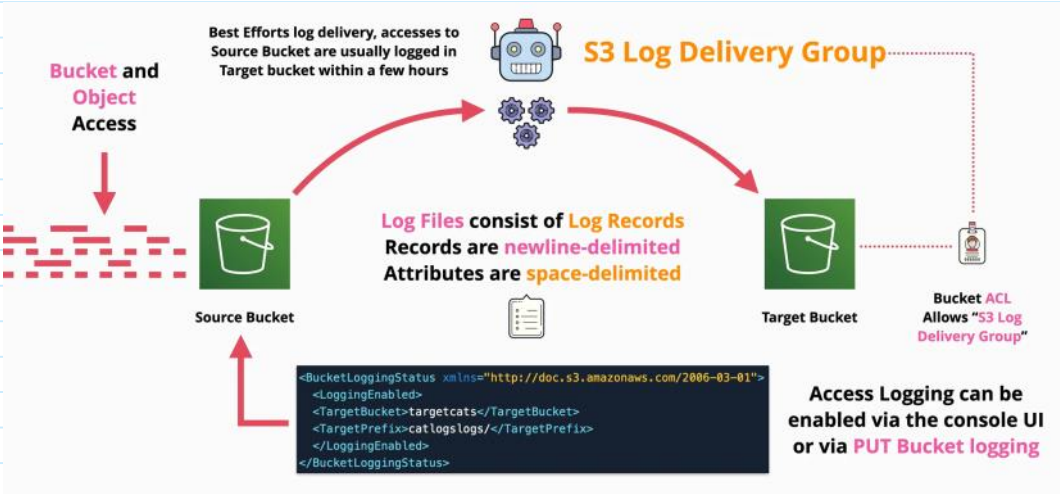
EventBridge is an alternative service, that supports more types of events and more services that notification.

S3 Access Logs

Friday, July 9, 2021 4:18 PM

S3 Access Logs is a feature available in S3 that provides detailed records for the requests that are made to an Amazon S3 bucket.

- a **source bucket** is a bucket that generated records
- a **destination bucket** is where the records are stored
- it is managed by a system known as **S3 log delivery group** which reads the logging configuration which is set to the source bucket
- enabling or changing the configuration to this service, needs a few hours to take effect
- logs are delivered as log files and each file consists of a number of log records and these are **newline-delimited**
- each record consists of attributes such as:
    - date and time of the event
    - the requester
    - the operation
    - status codes
    - error codes
    - etc.
- attributes within a record are **space-delimited**
- **a single target bucket** can be used for many **source buckets -**
    - the records can be separated easily using prefixes
    - this is configured in the logging configuration that is set on the **source bucket**

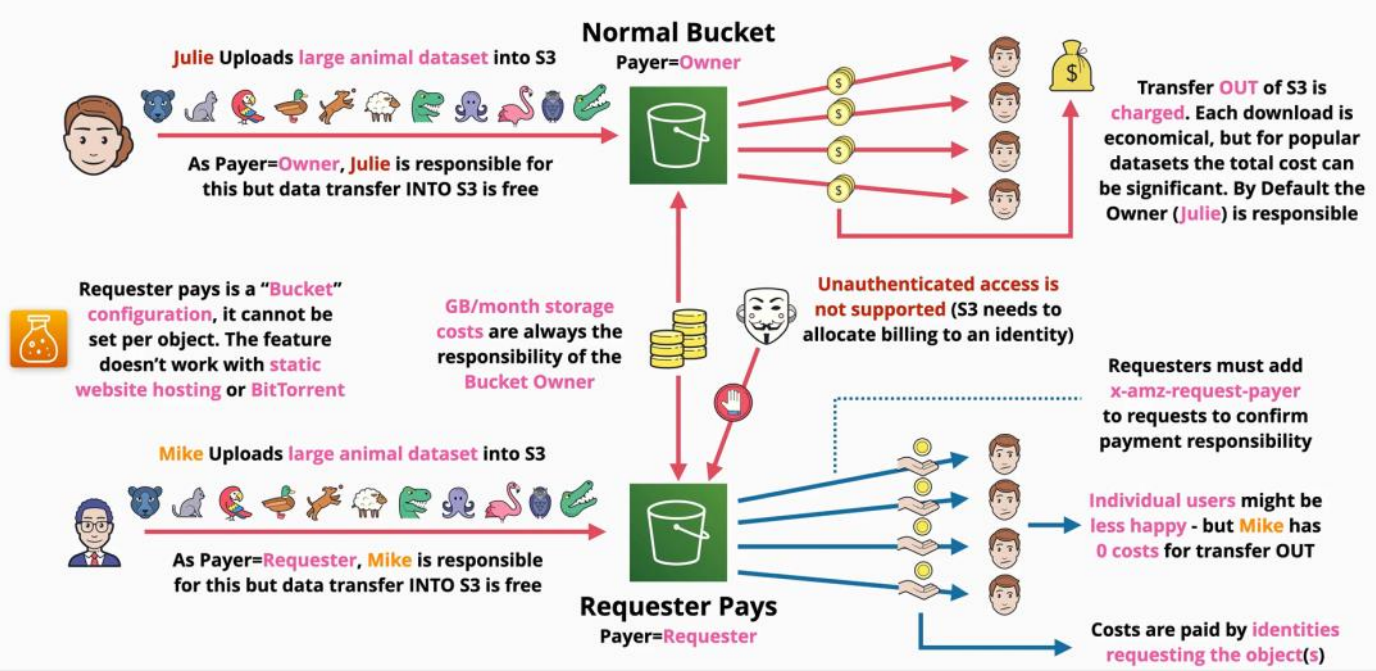


S3 Requester Pays

Friday, July 9, 2021 4:29 PM

With Requester Pays buckets, the requester instead of the bucket owner pays the cost of the request and the data download from the bucket. The bucket owner always pays the cost of storing data. ... The request authentication enables Amazon S3 to identify and charge the requester for their use of the Requester Pays bucket.

- S3 buckets charge for GB stored in a bucket
- IN transfer are free
- OUT transfers are charged
- with **requester pay bucket**
  - ★ this a bucket setting
  - cannot be used the bucket as a static website hosting or BitTorrent
  - ★ **only authenticated identities** can use the bucket
  - ★ any session downloading data from a **requester pay bucket is paid by the consumer**
    - **these users need to supply the x-amz-request-payer header to CONFIRM the payment responsibility**



S3 Object Lock

Friday, July 9, 2021 4:40 PM

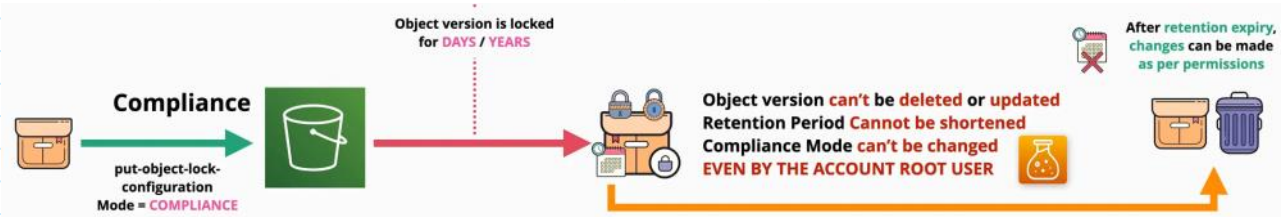
Amazon S3 Object Lock is an Amazon S3 feature that allows you to store objects using a write once, read many (WORM) model. You can use WORM protection for scenarios where it is imperative that data is not changed or deleted after it has been written.

- can be enabled on **new** S3 buckets
  - for existing buckets the AWS support needs to be contacted
- when enabled the versioning setting on the bucket needs to be enabled too
- cannot be disabled, or the versioning cannot be suspended
- write-once-read-many (WORM) architecture
  - object versions cannot be overwritten or deleted
- **requires versioning and individual versions of objects are locked**
- an object version can have
  1. **S3 object lock retention period**
  2. **S3 object lock legal hold**
  - an object can have, both, one or none
  - can be defined on individual object versions or for all object within a bucket

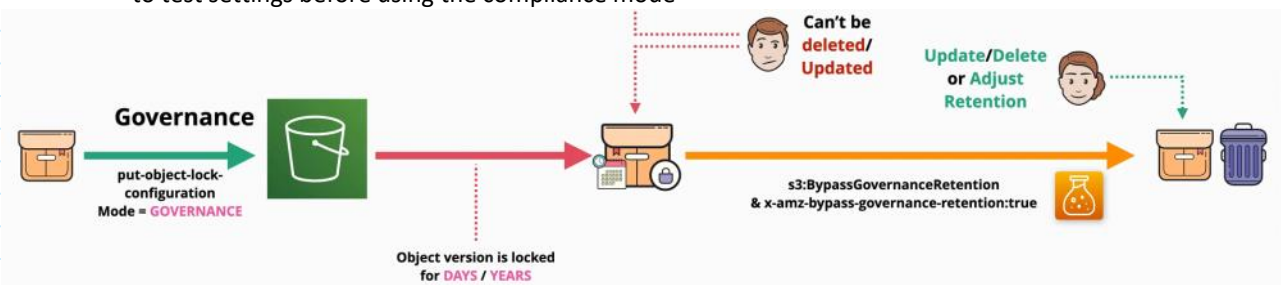
Retention Period

- when used you specify a retention period of day - years

1. **Compliance mode**
   - it means that an object version cannot be deleted or overwritten for the duration of the retention period
   - the retention period itself cannot be reduced and the retention mode cannot be adjusted during the retention period
   - ★ **no changes at all to all object version or retention period settings (this includes the root user)**

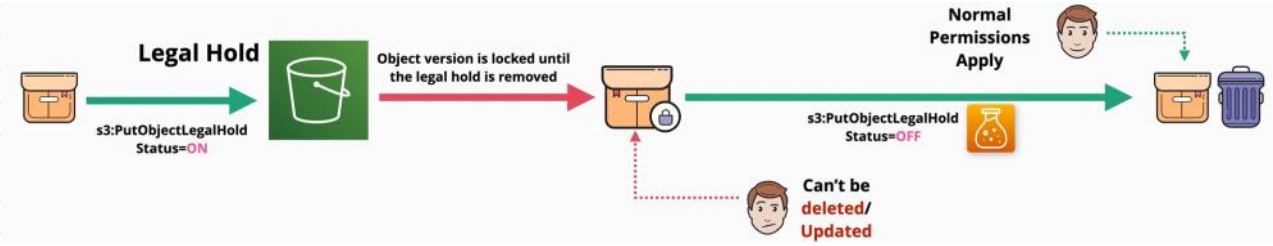


2. **Governance Mode**
   - while active the object version cannot be deleted or changed in anyway
   - you can grant special permissions to allow this to be changed
   - certain identities can change settings and object version
     - permission is required: **s3:BypassGouvernanceRetantion**
     - header is requested: **x-amz-bypass-governance-retention:true**
   - ★ is very useful for:
     - accidental deletion
     - process reasons or governance reasons to keep object versions
     - to test settings before using the compliance mode



3. **Legal hold**
   - with this type there is no retention period set at all
   - is used to set on an object version to be **ON** or **OFF**
   - does not allow deletion or changing of object version
   - to add or remove the legal hold the **s3:PutObjectLegalHold** is required
   - ★ used to:

- prevent accidental deletions of object versions
- when specific object versions need to be flagged



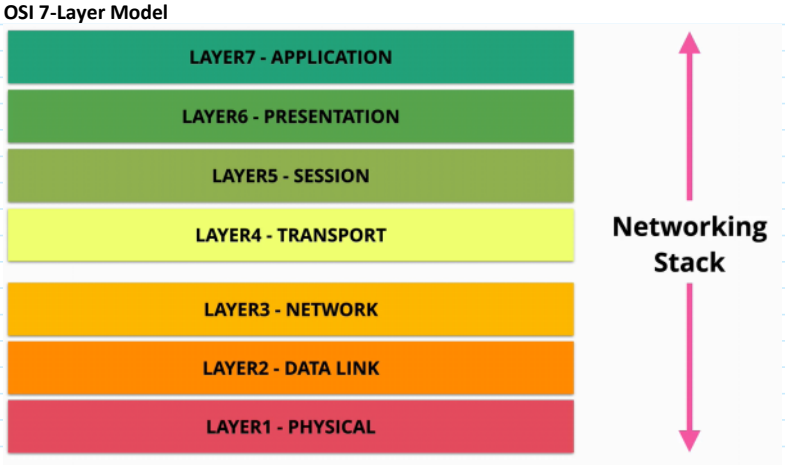
★ THEY CANNOT BE COMBINED

Networking

Monday, June 7, 2021 3:32 PM

Whenever you use a Cloud system, is done via networking.

OSI 7-Layer Model

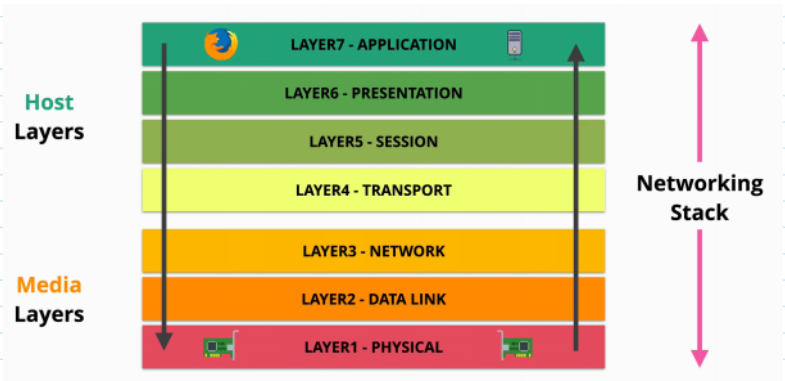


Host Layers

- **How data is chopped off and reassembled, and how is formatted so that is understandable by both sides of the network connection**

Media Layers

- **How date is moved from point A to point B**

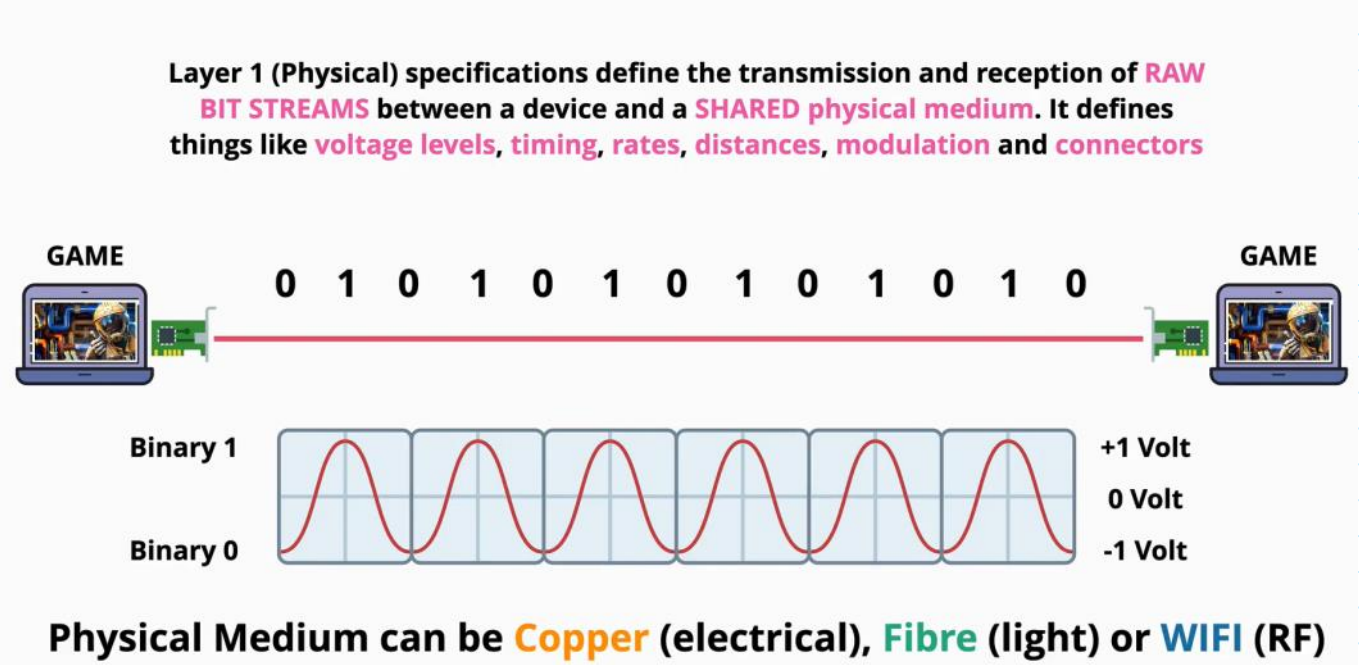


Layer 1 - Physical Layer

Monday, June 7, 2021 3:38 PM

Physical later is the networking **card**.
The connection between cards is called a **physical medium** which is usually a copper cable, optical fiber or WIFI.

For Copper cable, voltage is used to transmit data, using a low voltage as 0, and a high voltage as 1.

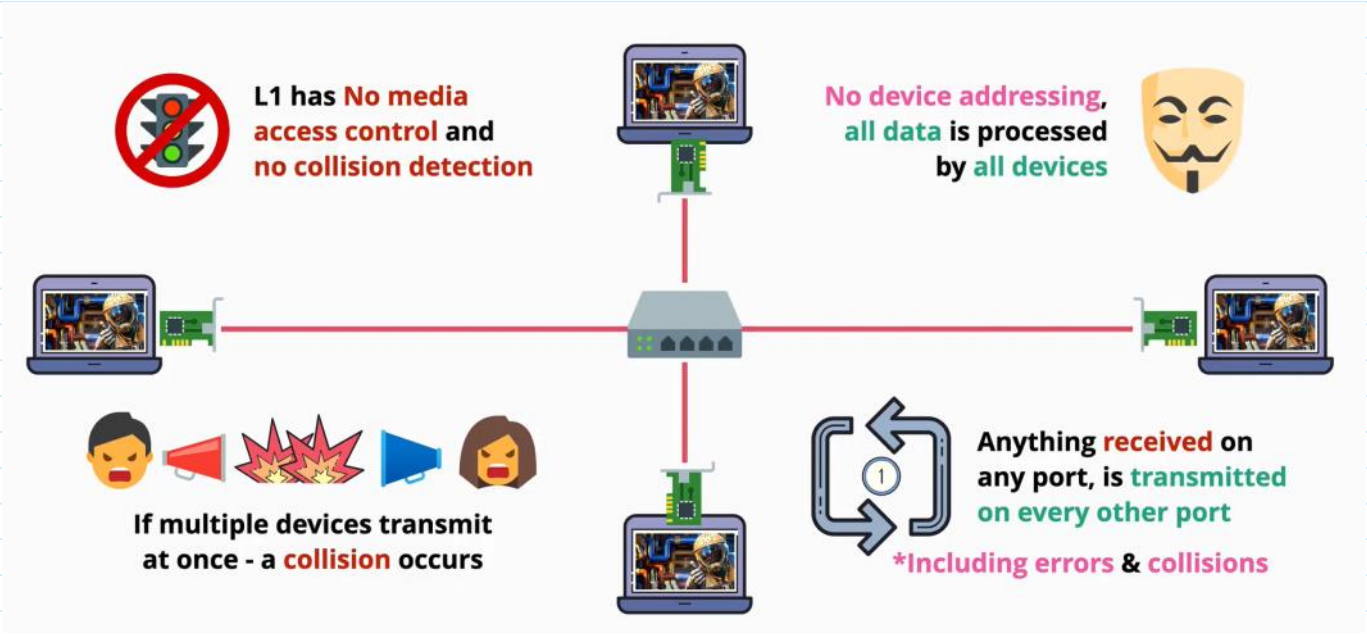


If more devices need to communicate, a **hub** is needed. A **hub** job is to transmit to every device, what is receiving from a device. Anything received on any port, is transmitted to all the other ports!

In this layer, none of the connected device has addressing (identities - names).

Two devices might transmit at the same time, fact that might results into a collision - only one device can transmit at a time, so that the information transmitted can be read by the other devices.

This layer has no media control access, or any collision detection, therefore, is a network is created using only layer 1, collision will be present.


L1 has No media access control and no collision detection
No device addressing, all data is processed by all devices
If multiple devices transmit at once - a collision occurs
Anything received on any port, is transmitted on every other port
*Including errors & collisions

Layer 2 - Data Link Layer

Monday, June 7, 2021 3:56 PM

The Data Link layer runs over the physical layer.

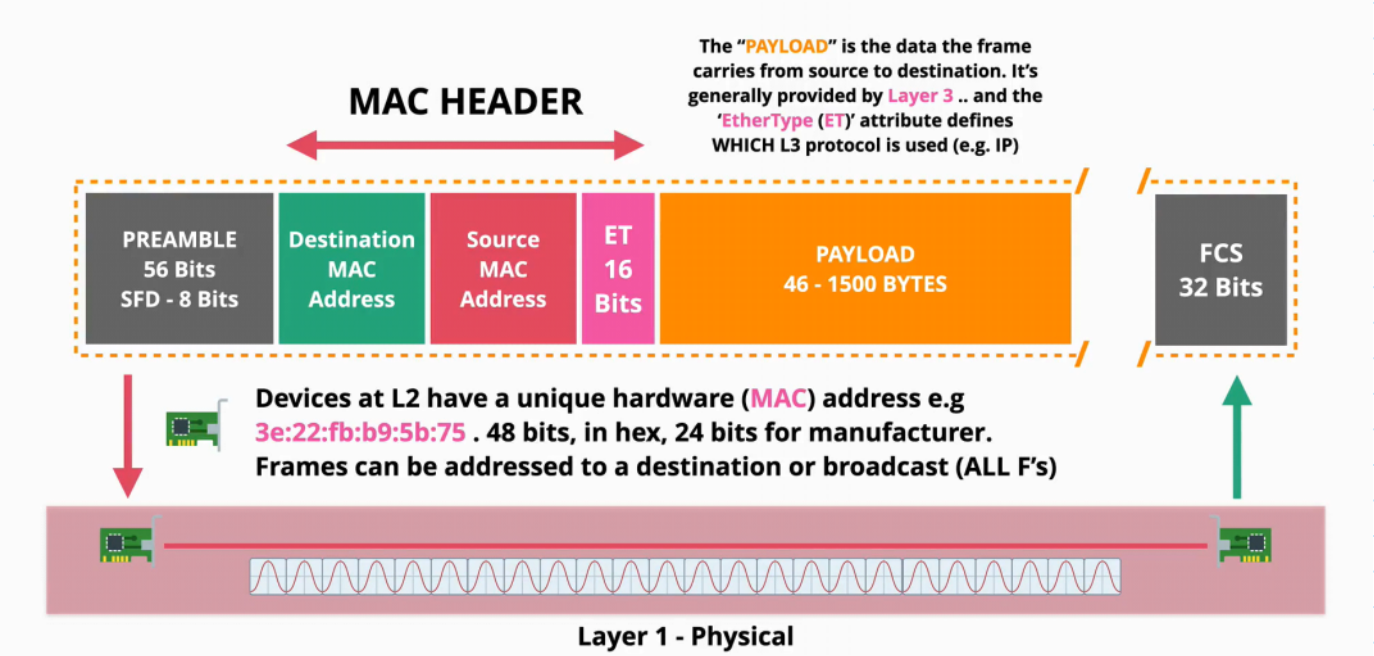
The Data Link layer uses frames to send information between devices. Devices using this layer has a unique hardware address called a **MAC address**, which is represented by hexadecimal values.
This address is not based on any software, it is uniquely attached to a piece of hardware.

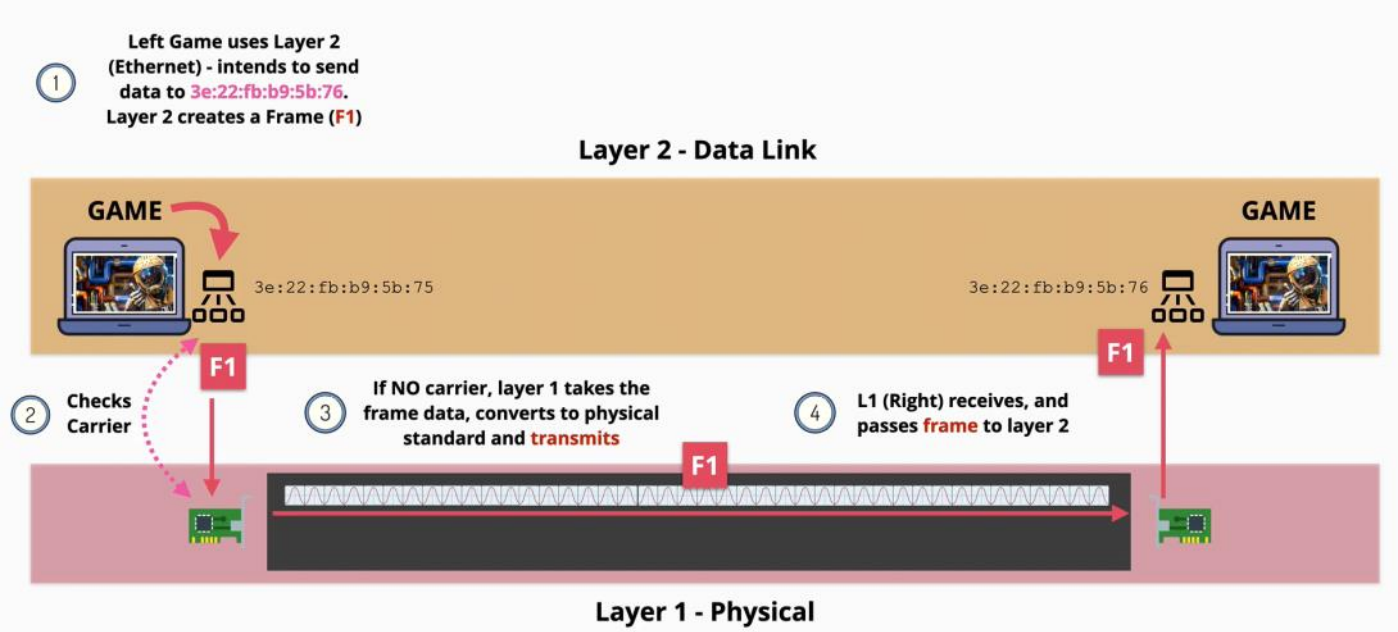
The MAC address is made by two parts:

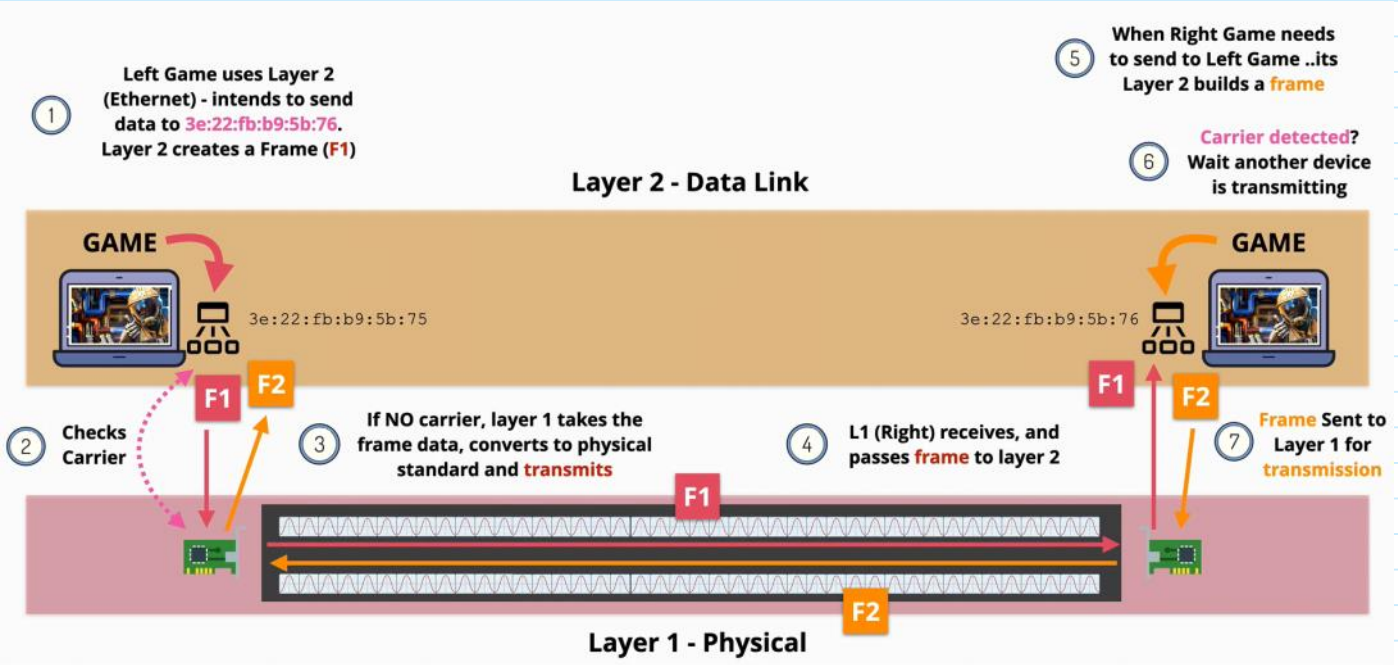
1. The OUI - organizational unique identifier
2. The NIC - network interface controller

As the data link layer is based on layer one, the information shared by layer 1. For simplicity, layer 2 transmits a frame, via the layer 1, even layer 1 does not understands what it is sharing.

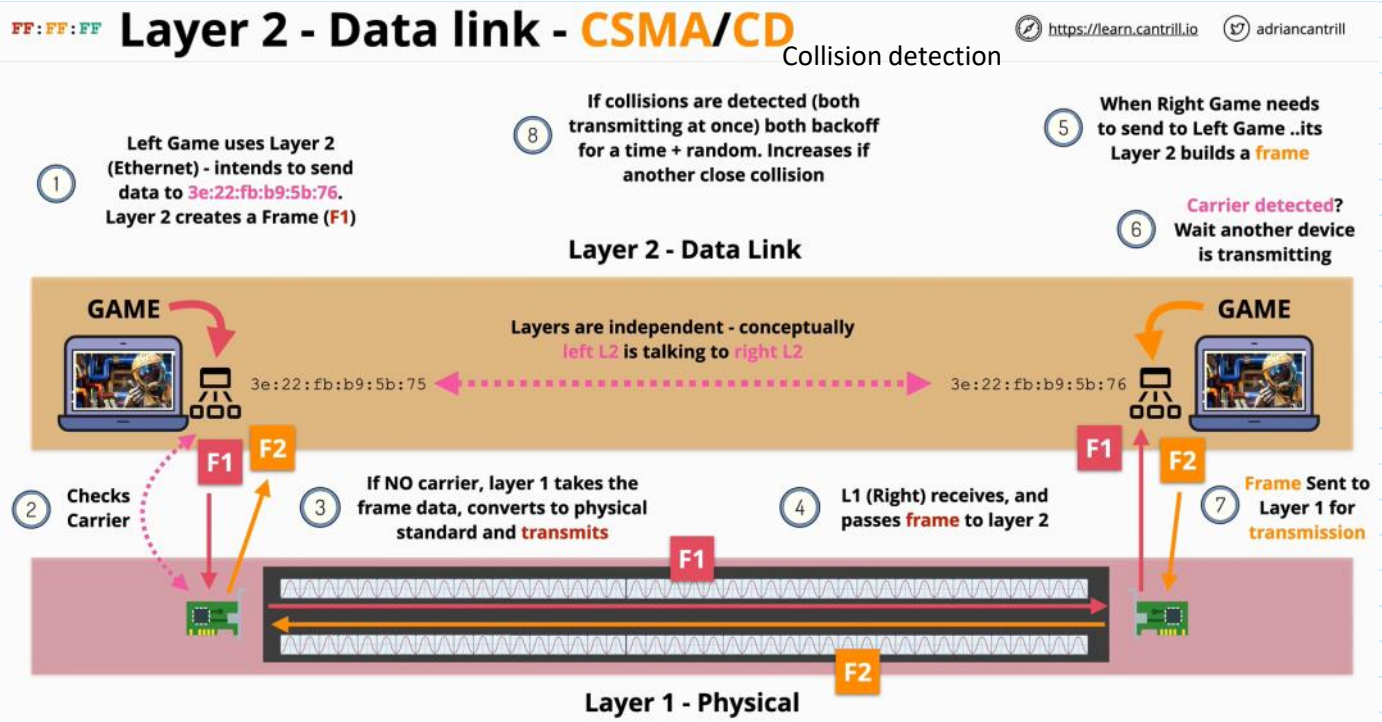
Layer 2 brings identifiable devices, senders and receivers to life.



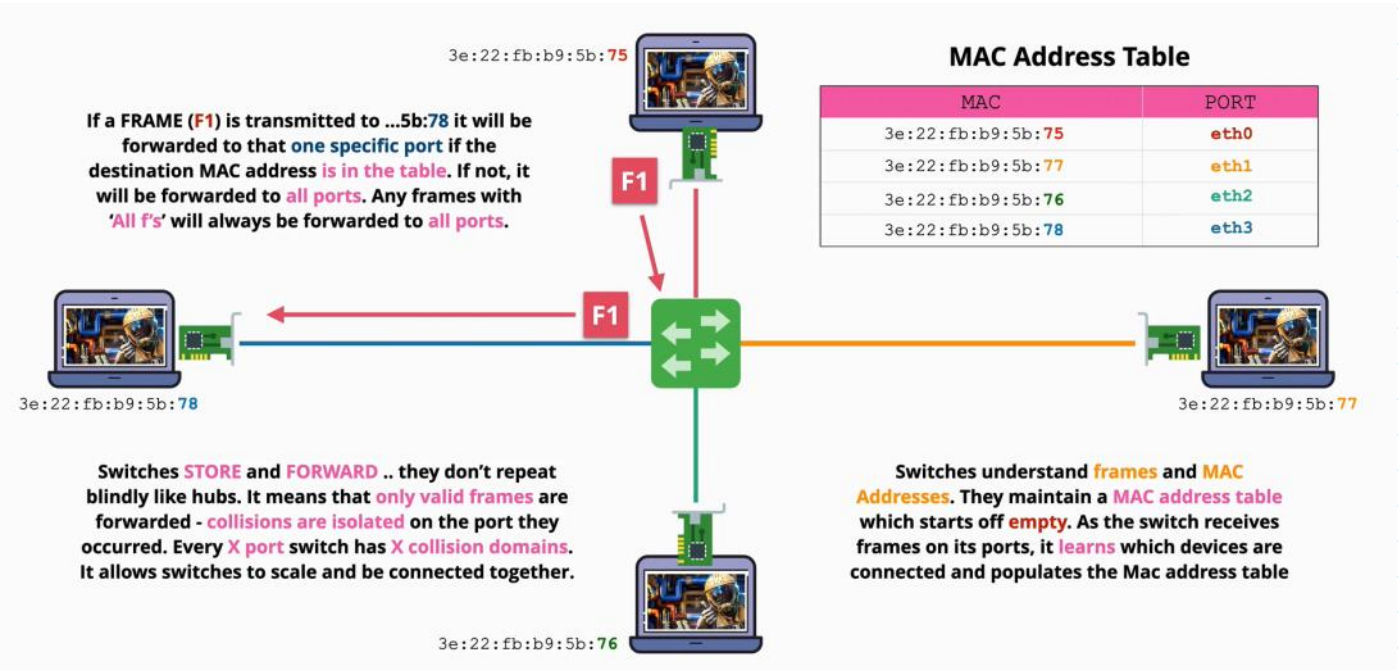




The data is **encapsulated** into the frame when transmitted, and extracted when it reaches the receiver.



A **switch** is a layer two hub, that understands frames.



Decimal <=> Binary

Monday, June 7, 2021 4:35 PM

133.33.33.7 -> dotted decimal notation

133	10000101
33	00100001
33	00100001
7	00000111

Position	1	2	3	4	5	6	7	8
Binary Position Value	128	64	32	16	8	4	2	1
Binary Value								

128	64	32	16	8	4	2	1

Layer 3 - Network Layer

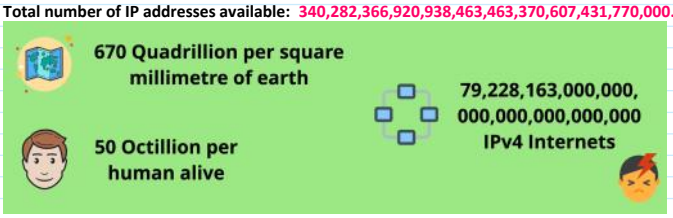
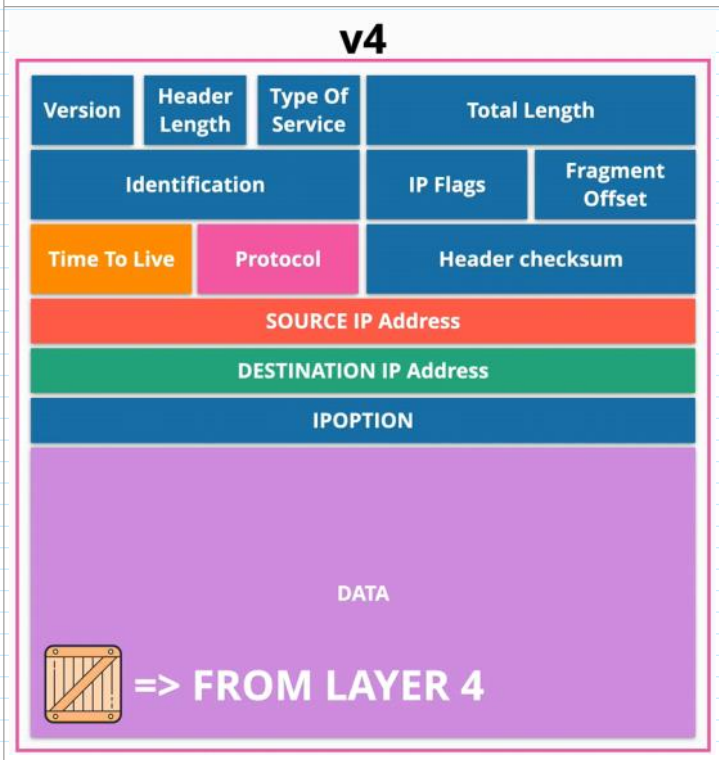
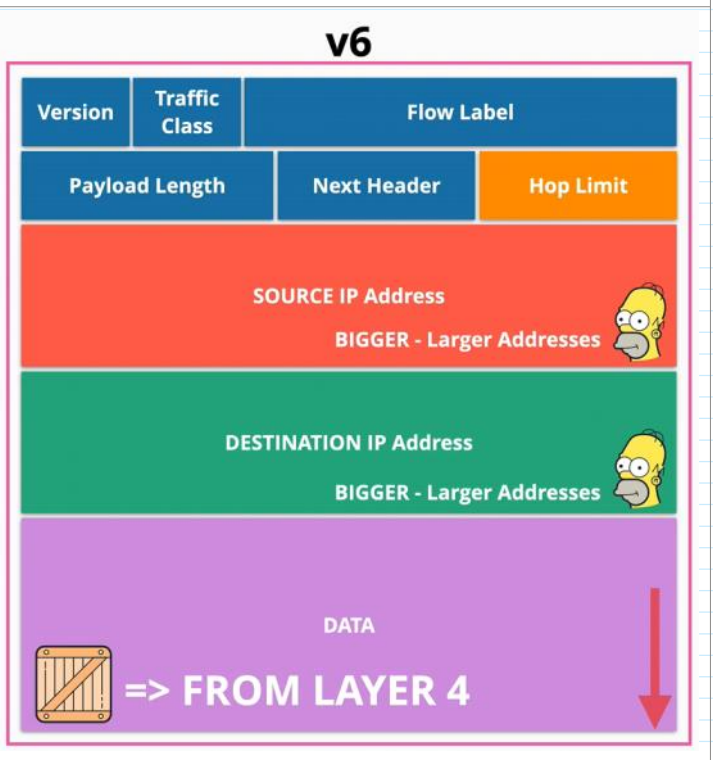
Tuesday, June 8, 2021 2:42 PM

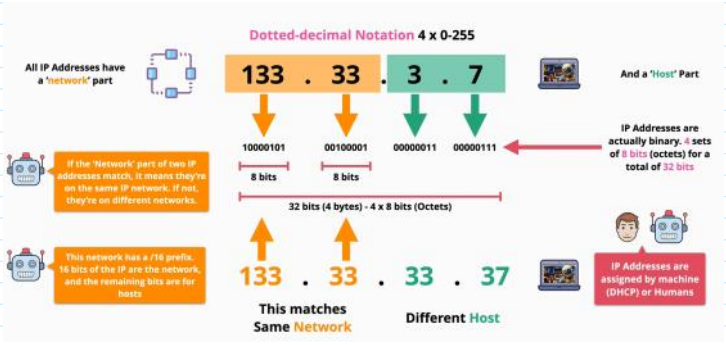
Ethernet is the most popular layer 2 connection.

Layer 3 is the common protocol which can spam multiple different Layer 2 networks.
IP address comes with Layer 3.
Routers are layer 3 devices which move packets of data across different networks.

Packets - data unit used by the layer 3 protocol.
- are very similar to frames
  - within frames the source and the destination are local
  - with packets, the destination and source could be on different sides of the planet
- packets never change when transitioning from source to destination
  - the frames are changed as the packets go from one local network to another
  - the packets are encapsulate in the frame

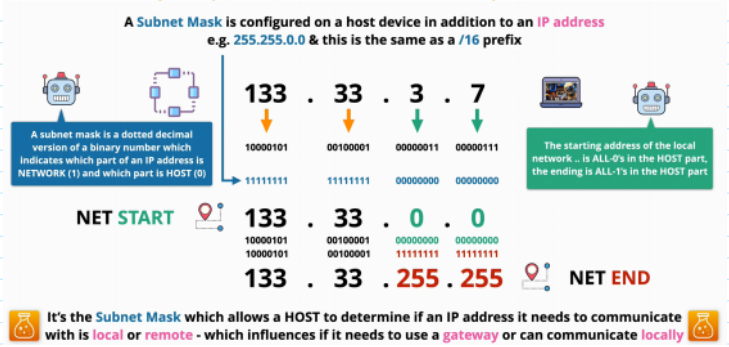
IP addresses

version 4	**version 6**
• destination IP • source IP • protocol - which protocol is used • ICMP, TCP, UDP • data • time to live (TTL) - if the packets cannot reach its destination, this number declares how many hops the packet can take (so that it won't travel forever) before being discarded • **Total number of IP addresses available:** 4,294,967,296	• destination IP • source IP • data • hop limit (time to live) • **Total number of IP addresses available:** 340,282,366,920,938,463,463,370,607,431,770,000… • 
	



First two digits - **network**
Last two digits - **host**

Subnet masks allows a host to determine if an IP address is local or remote. The result allows the sender to know if it needs a gateway (router) or it can communicate locally.



255.255.0.0 -> 11111111.11111111.00000000.00000000 /16 (ones)

everything with ones represents the network
everything with zeros represents the host

All data generated by your router is by default sent through the internet service provider.
The internet provider has multiple interfaces route table to forward the data

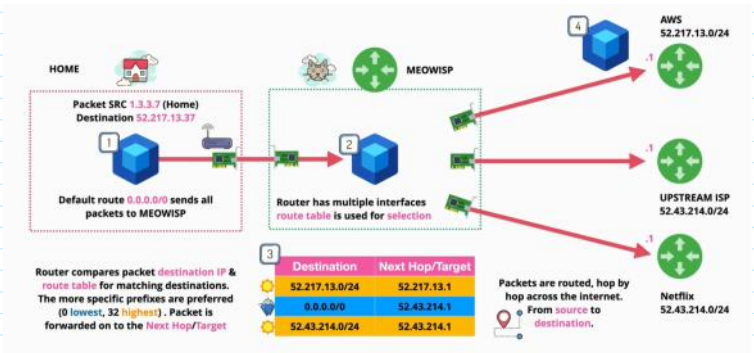
1. **each router has multiple route tables - a collection of routes**
2. **every packet that arrives to the internet provider will check its destination, and if a match is found the data is sent to the next hop according to its destination**

3

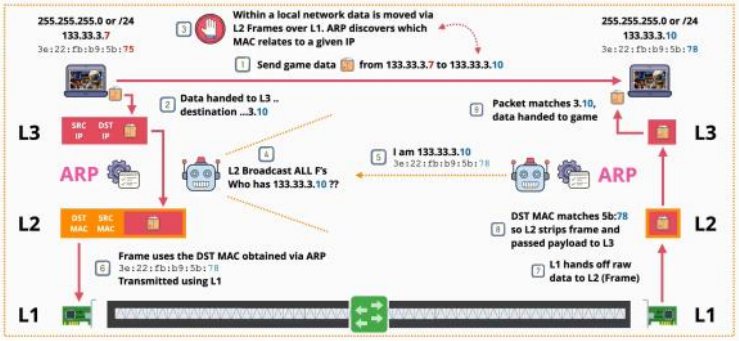
Destination	Next Hop/Target
52.217.13.0/24	52.217.13.1
0.0.0.0/0	52.43.214.1
52.43.214.0/24	52.43.214.1

0.0.0.0/0 - default route

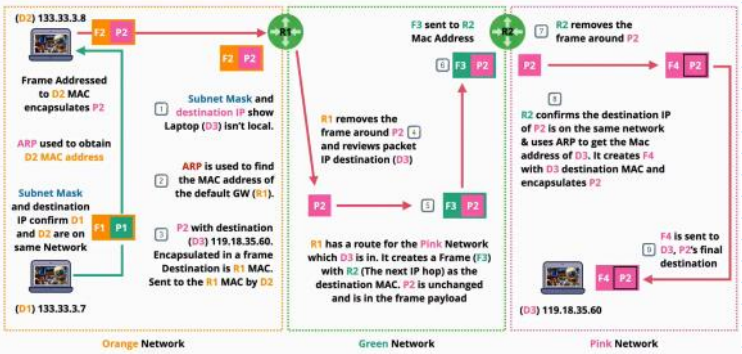
3. **each time a hop is made, the frame where the packet is encapsulated is changed, according to the new destination**



Address Resolution Protocol (ARP)

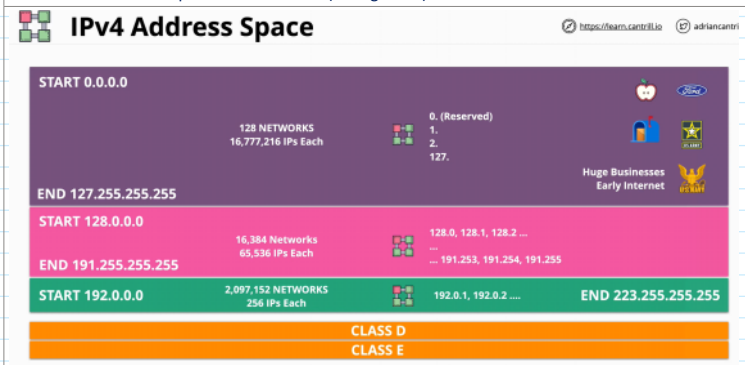


ARP is responsible of finding the **MAC** address of the destination/next hop **host.**



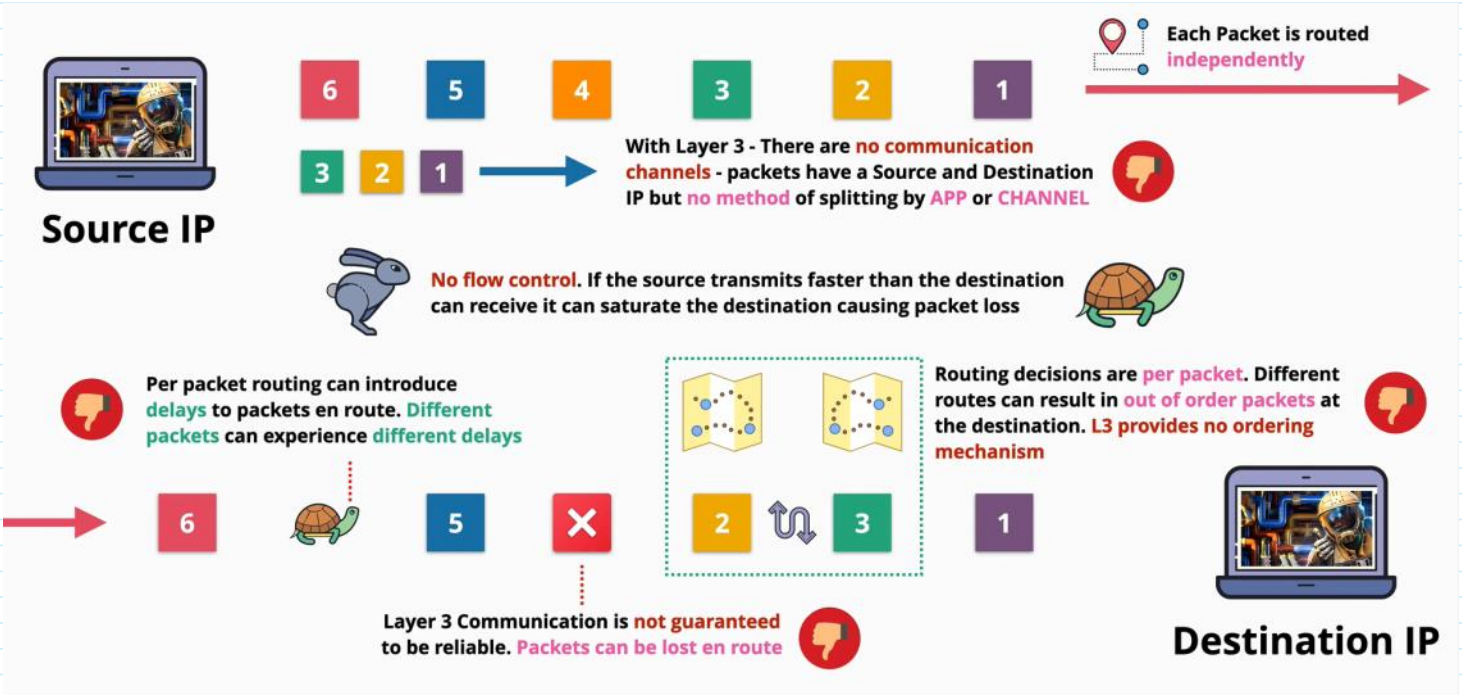
1. **If routers receive packets within a frame that are not for them, they are checking the routing table, match the next hop and create a new frame for the packet, before it sends the frame to a new router**
2. **However, if a host receives a frame that is not destinated for it, it drops the frame and packet.**

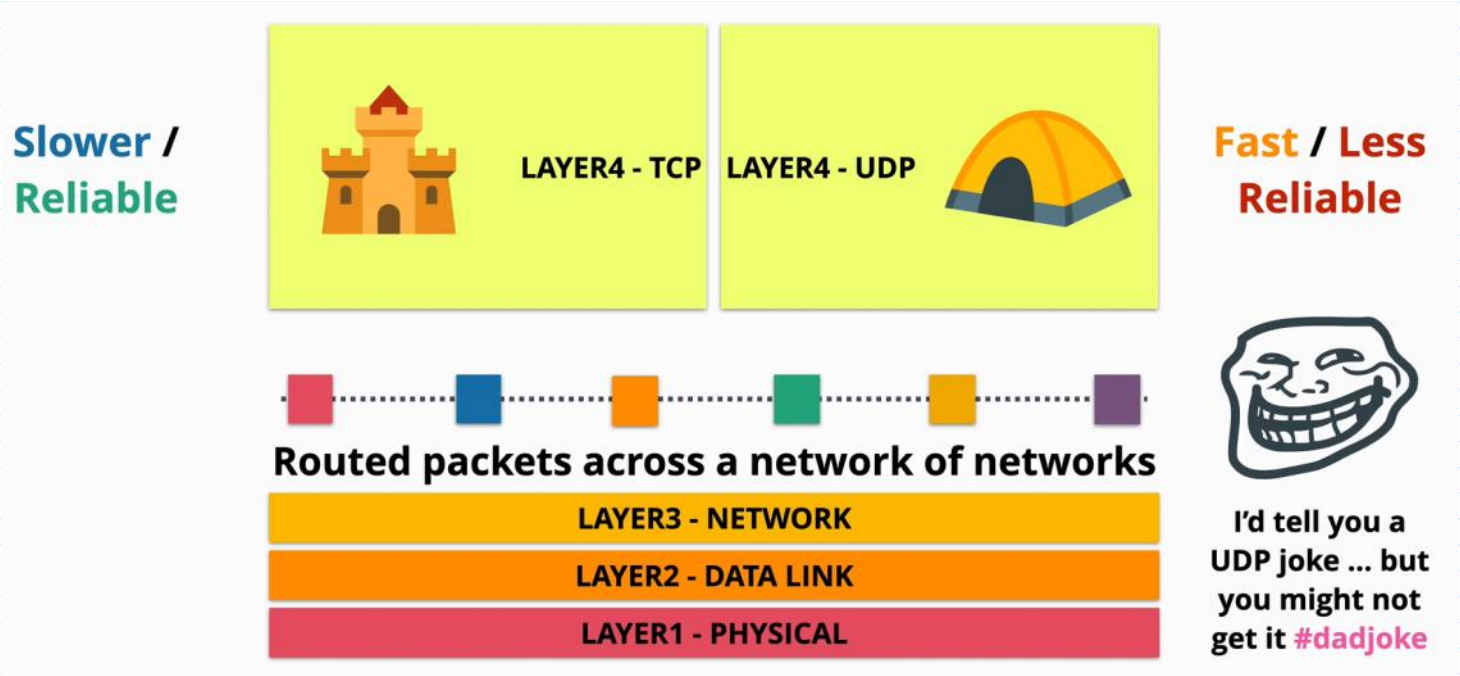
1. **Layer 3 has:**
    a. **routes - where to send the packet**
    b. **route tables**
    c. **ARP - find the mac address for a certain IP**
    d. **IP addresses**
    e. can deliver packets out of order (wrong order)



Layer 4 - Transport Layer

Wednesday, June 9, 2021 1:36 PM



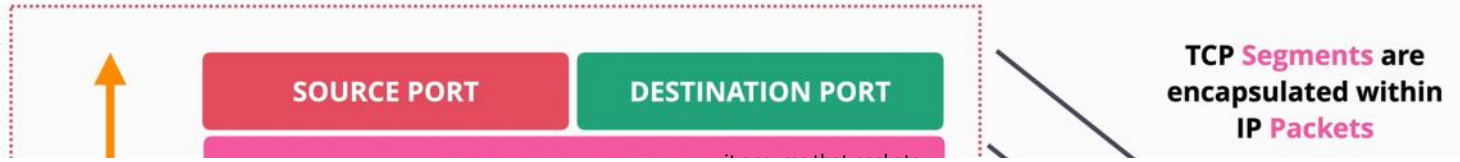


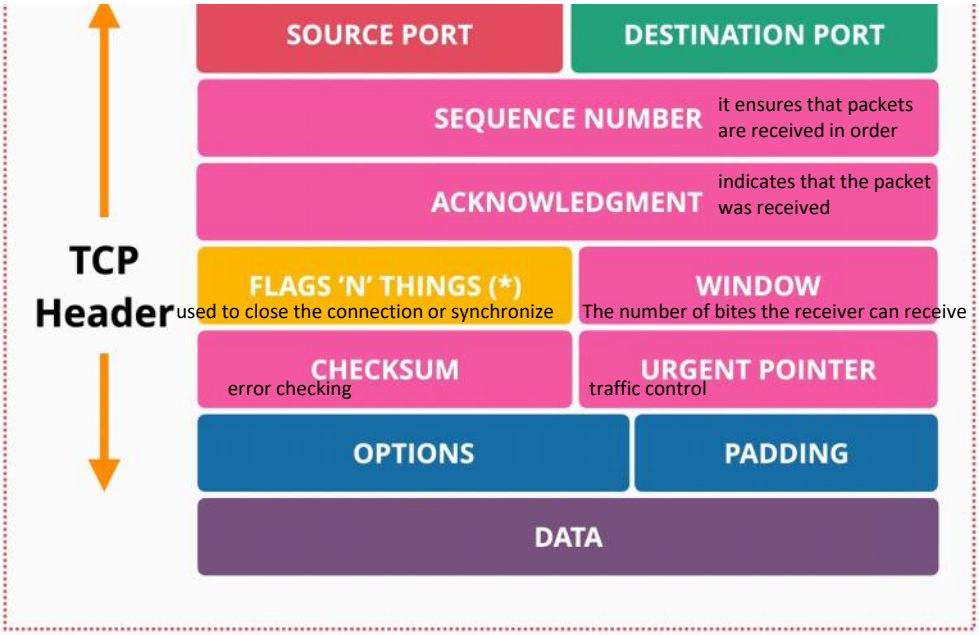
Layer 4 runs over Layer 3, therefore it needs IP addresses.

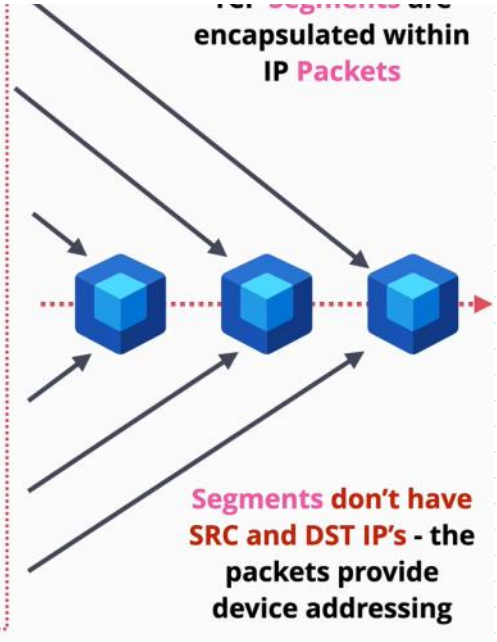
- **TCP -** has a reliable connection
- **UDP -** is al about performance

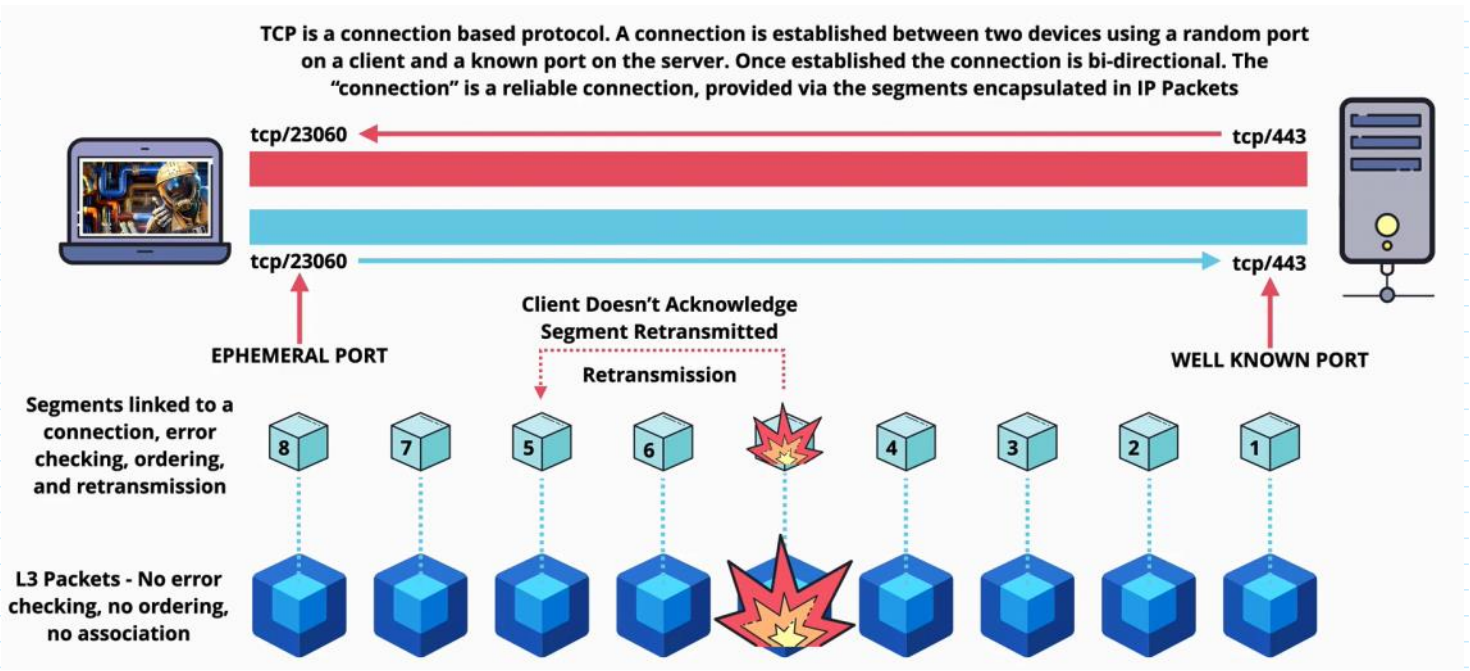
TCP introduces the term of **segments,** which are another type of containers as packets and frames are.

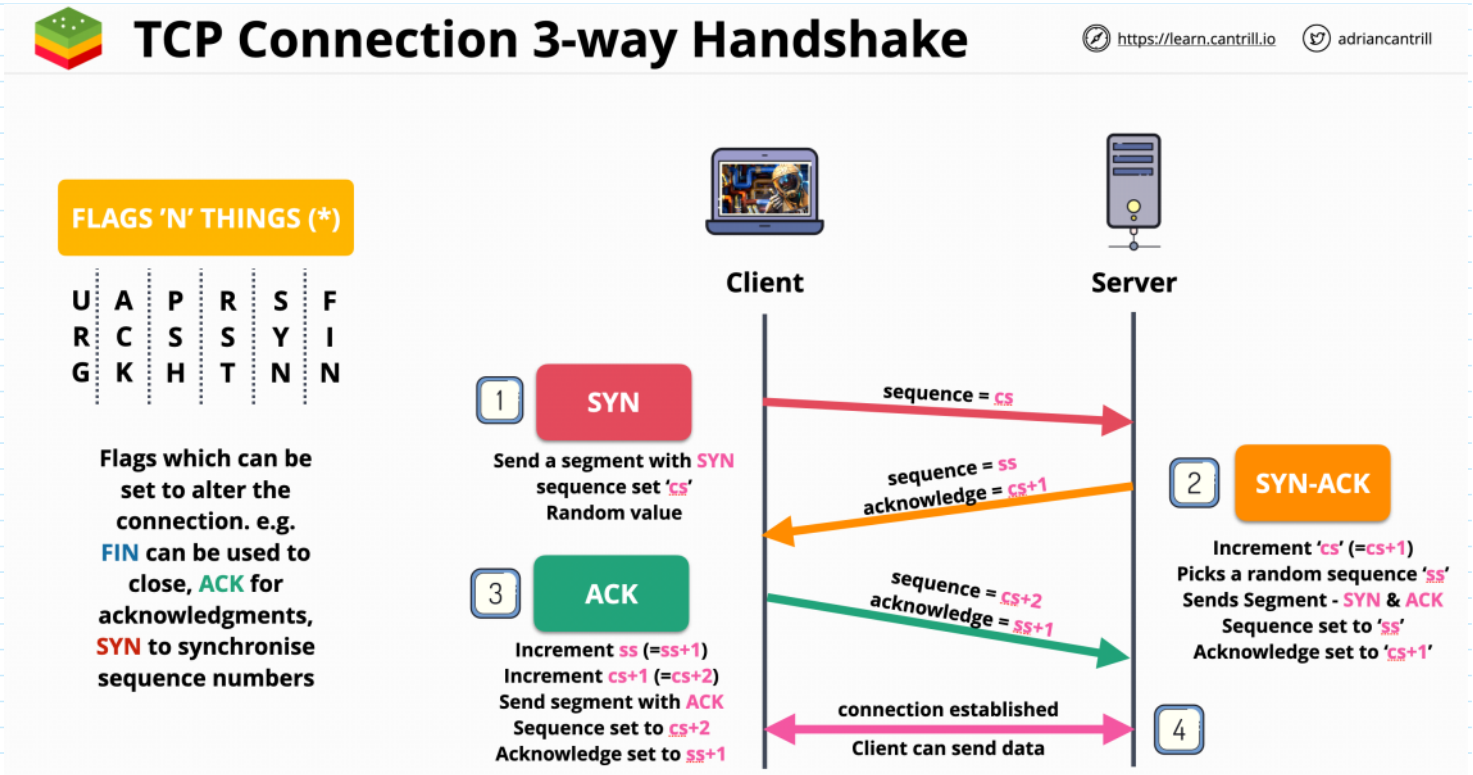
- are encapsulated in packets
- they do not have source and destination information because they the IP packets for the transit

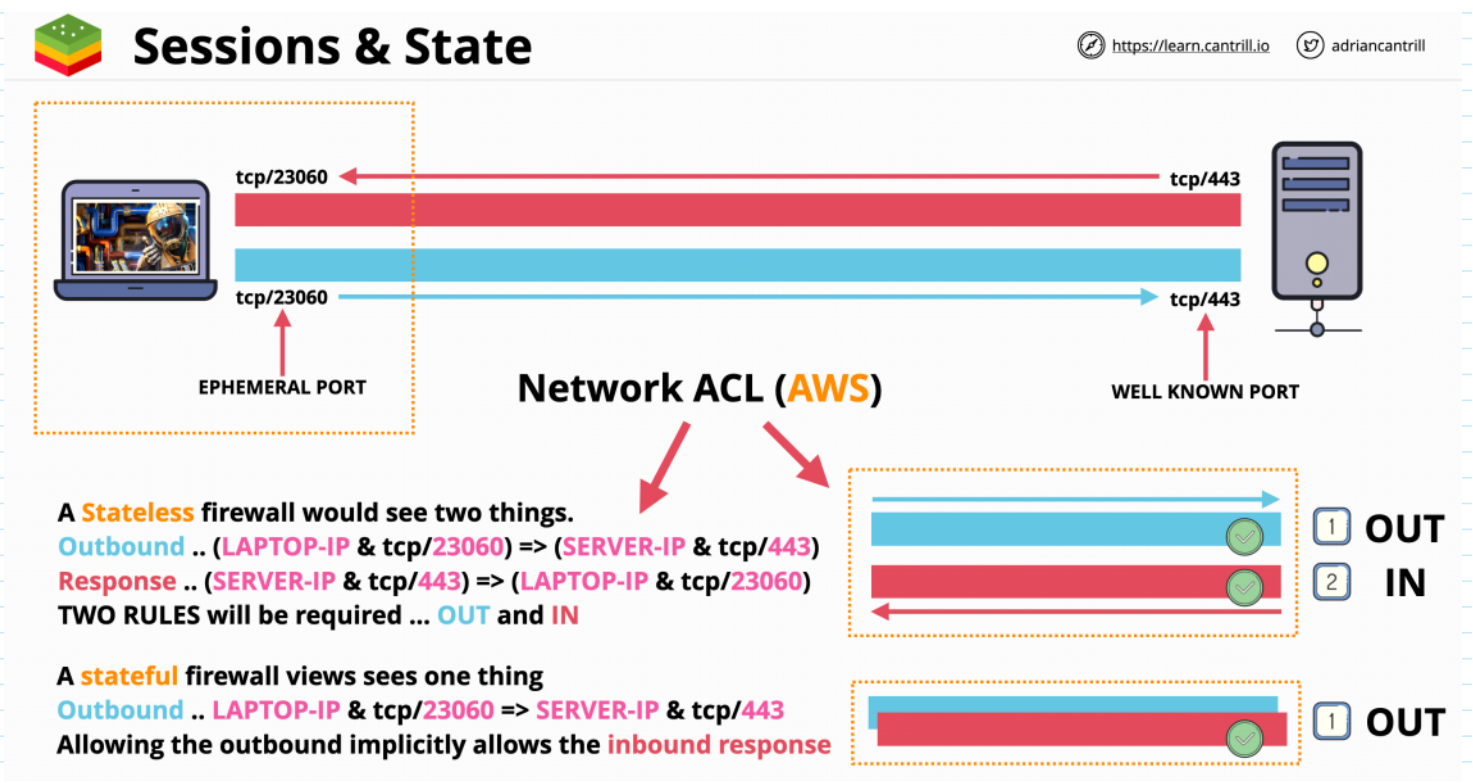



TCP Header
SOURCE PORT
DESTINATION PORT
SEQUENCE NUMBER
it ensures that packets are received in order
ACKNOWLEDGMENT
indicates that the packet was received
FLAGS 'N' THINGS (*)
used to close the connection or synchronize
WINDOW
The number of bites the receiver can receive
CHECKSUM
error checking
URGENT POINTER
traffic control
OPTIONS
PADDING
DATA


encapsulated within IP Packets
Segments don't have SRC and DST IP's - the packets provide device addressing


TCP is a connection based protocol. A connection is established between two devices using a random port on a client and a known port on the server. Once established the connection is bi-directional. The "connection" is a reliable connection, provided via the segments encapsulated in IP Packets
tcp/23060
tcp/443
tcp/23060
tcp/443
EPHEMERAL PORT
Client Doesn't Acknowledge Segment Retransmitted
Retransmission
WELL KNOWN PORT
Segments linked to a connection, error checking, ordering, and retransmission
L3 Packets - No error checking, no ordering, no association
8
7
5
6
4
3
2
1





Stateless firewall does not understand the state of a connection.
Rules

1. Outbound: send -> **OUT**
2. Response: receive **-> IN**

Stateful firewall understands the states of the TCP segments.
It sees the outbound and the response traffic as one, therefore once the connection is established, it automatically allows the response.

TCP Well Known Ports

- tcp/80 - HTTP & tcp/443 HTTPS
- tcp/22 - SSH
- tcp/25 - SMTP (email)
- tcp/21 - TELNET
- tcp/3389 - REMOTE DESKTOP PROTOCOL
- tcp/3306 - MySQL/MariaDB/Aurora

Network Address Translation (NAT)

Wednesday, June 9, 2021 3:51 PM

Layer 5 - Session Layer

Wednesday, June 9, 2021 1:40 PM

Network Address Translation (NAT)

Wednesday, June 9, 2021 4:14 PM

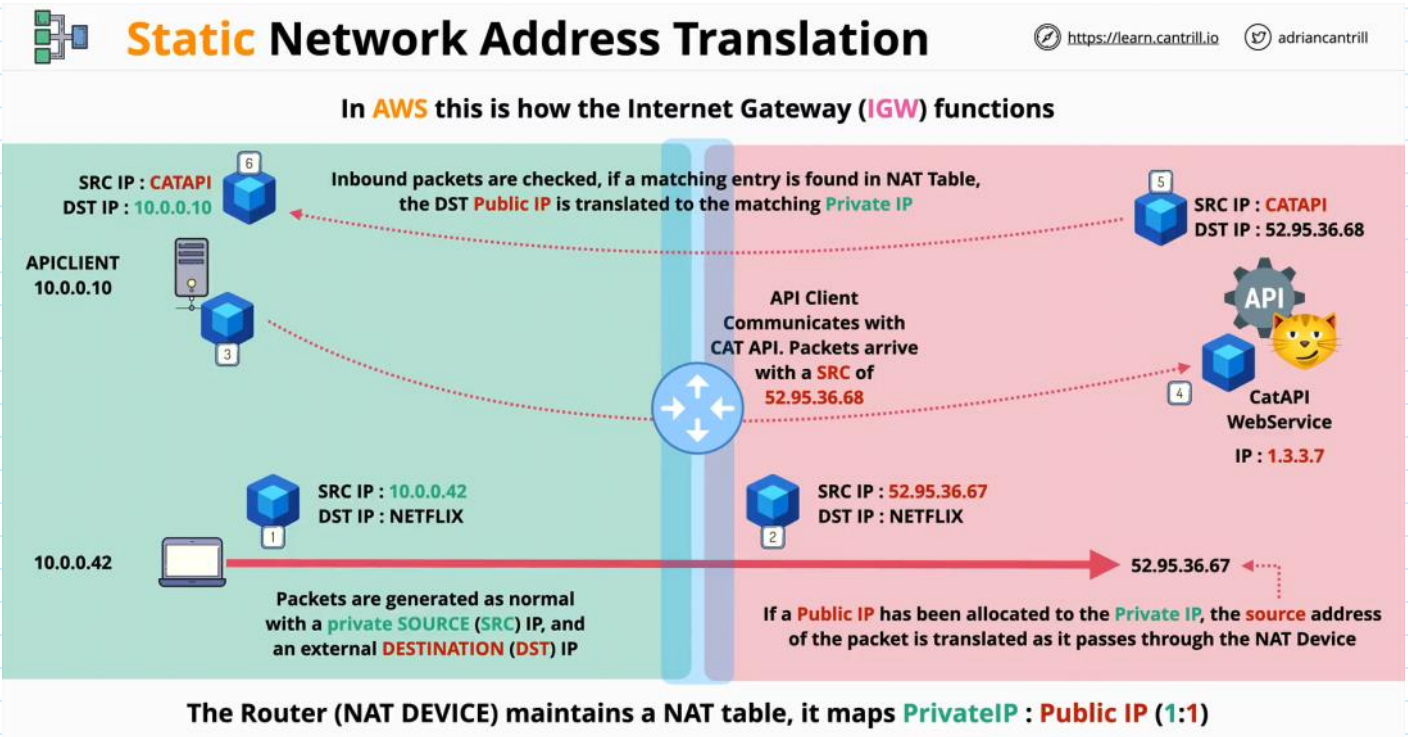
NAT is a process that is designed to address the growing shortage of IPv4 addresses.

- NAT is designed to overcome **IPv4 shortages**
- ... also provides some **security benefits**
- Translates **Private** IPv4 addresses to **Public**
- **Static NAT** - **1 private** to **1 (fixed) public address** (IGW) **one to one**
- **Dynamic NAT** - **1 private** to **1st available Public** **one to many (from a pool)**
- Port Address Translation (**PAT**) - **many private** to **1 public** (NATGW) **many to one**
- **IPv4 only** ... makes no sense with IPv6

IPv6 does not need NAT, as there is **no** need for **private IP addressed**

Static NAT

Private IP addresses cannot communicate over the internet with the public IP addresses.

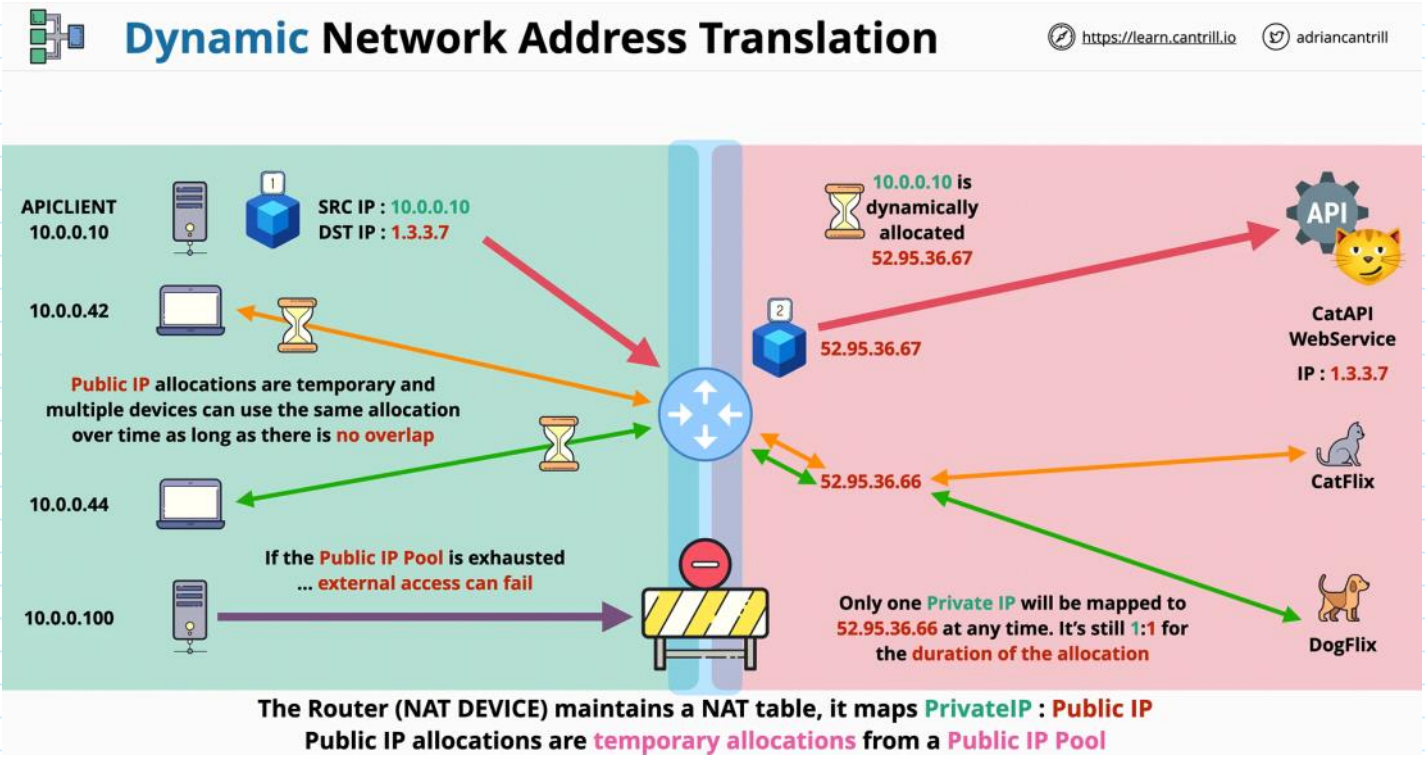


Is important to remember that in all the time, the two private devices, are never assigned any public IP address. The NAT device, is routing the packets to the private devices.

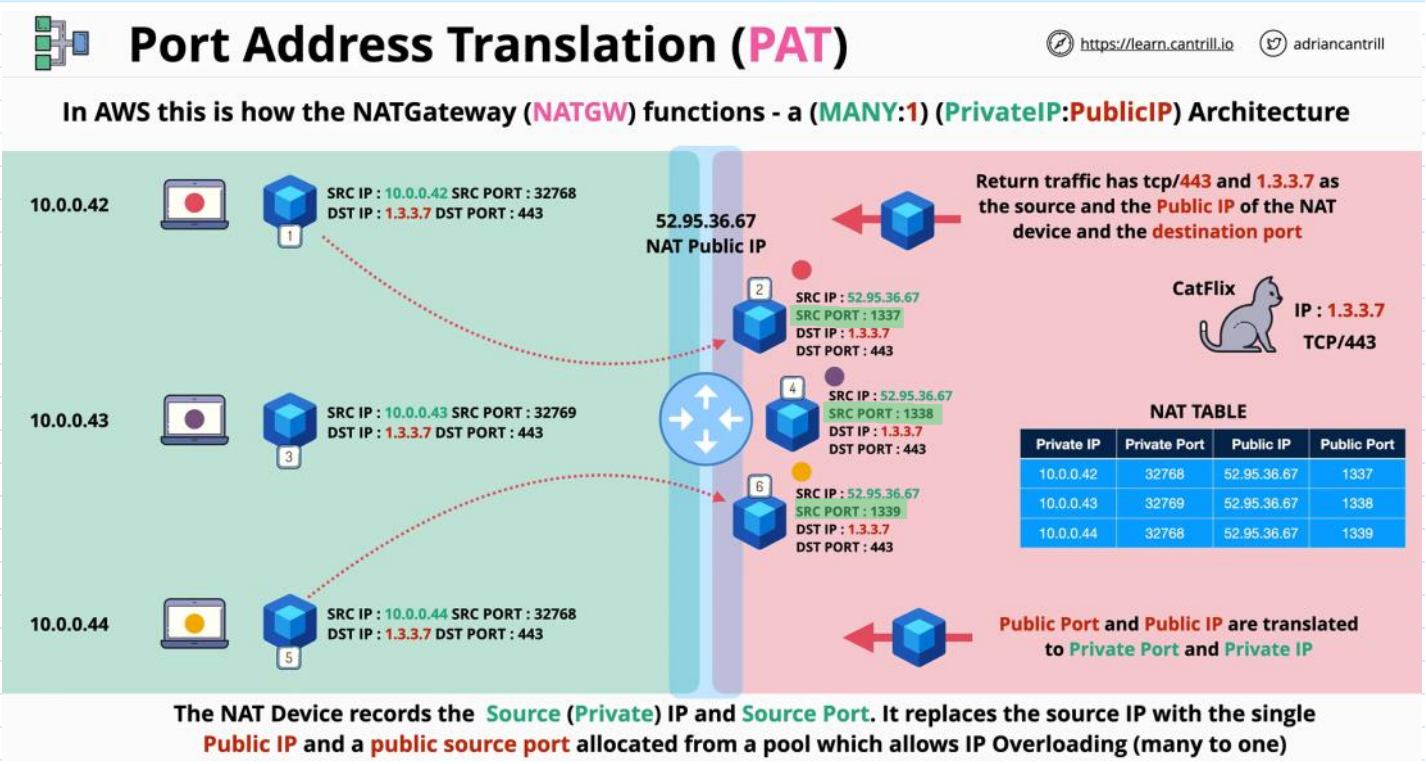
Dynamic NAT

Dynamic NAT is applied when there are not enough public IP addresses for all the private devices,

therefore a pool of public IP address is temporary allocated and used as they are needed.

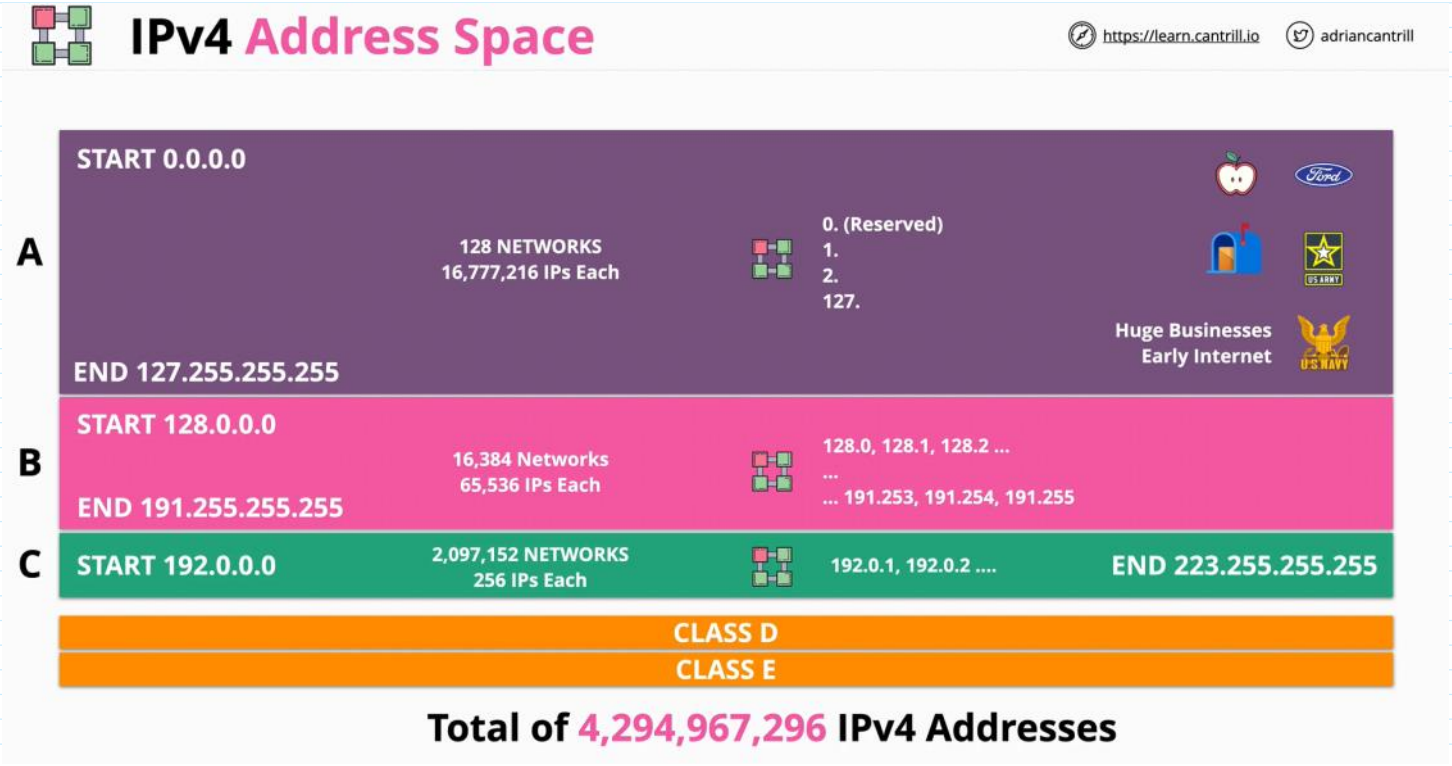


Port Address Translation (PAT)



IPv4 Addressing and Subnetting

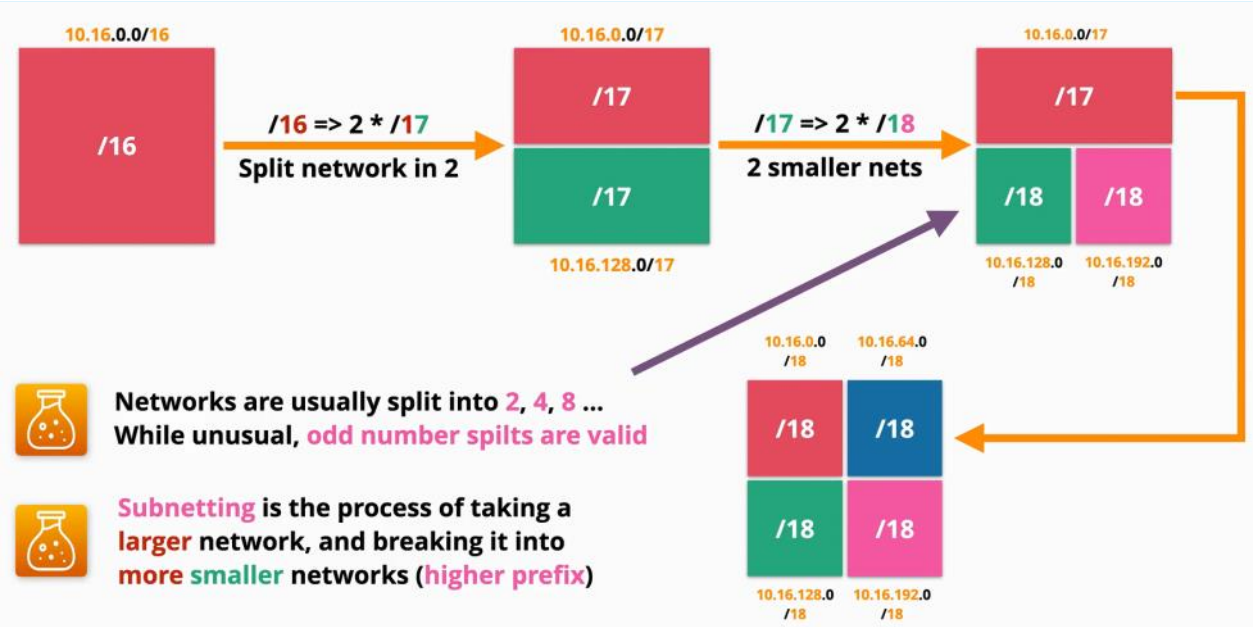
Wednesday, June 9, 2021 5:07 PM



10.16.0.0 /16 (first 16 zeros in binary) - means that the last two octets are available for hosts or subnetting.

1. **starts are 10.16.0.0**
2. **ends at 10.16.255.255**

/0	all internet
/8	class A
/16	class B
/24	class C
/32	a single IP



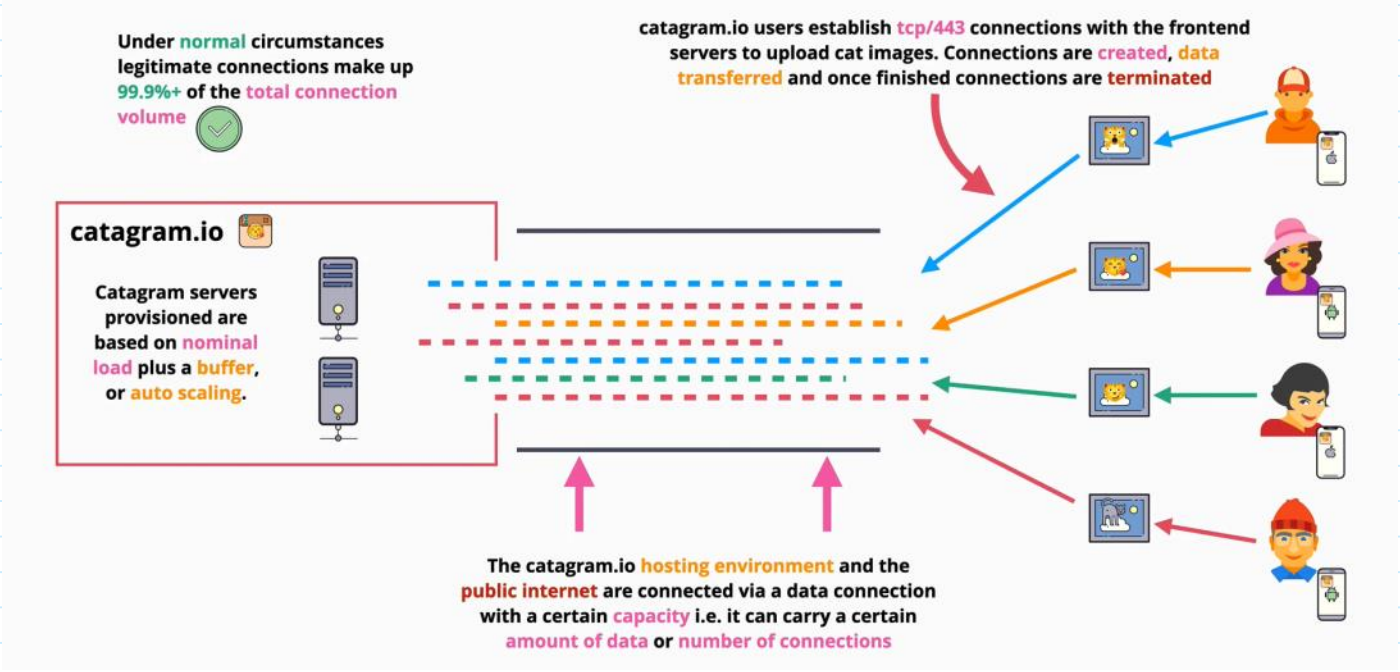
10.16.0.0	**00001010.00010000**.00000000.00000000/**16**	255
	00001010.00010000.00000000.00000000/17	128
	00001010.00010000.00000000.00000000/18	64

1	2	4	8	16	32	64	128	256
0	1	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0

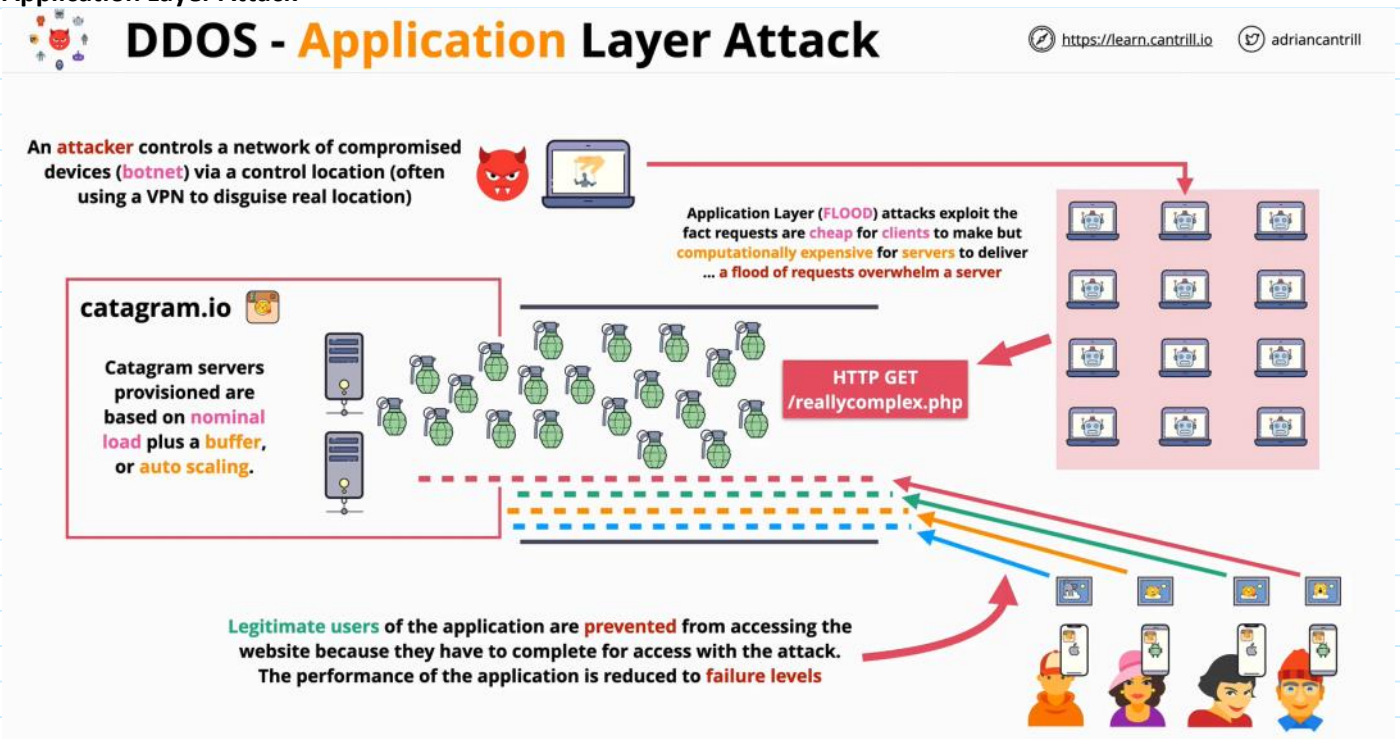
Distributed Denial of Service (DDos)

Thursday, June 10, 2021 1:26 PM

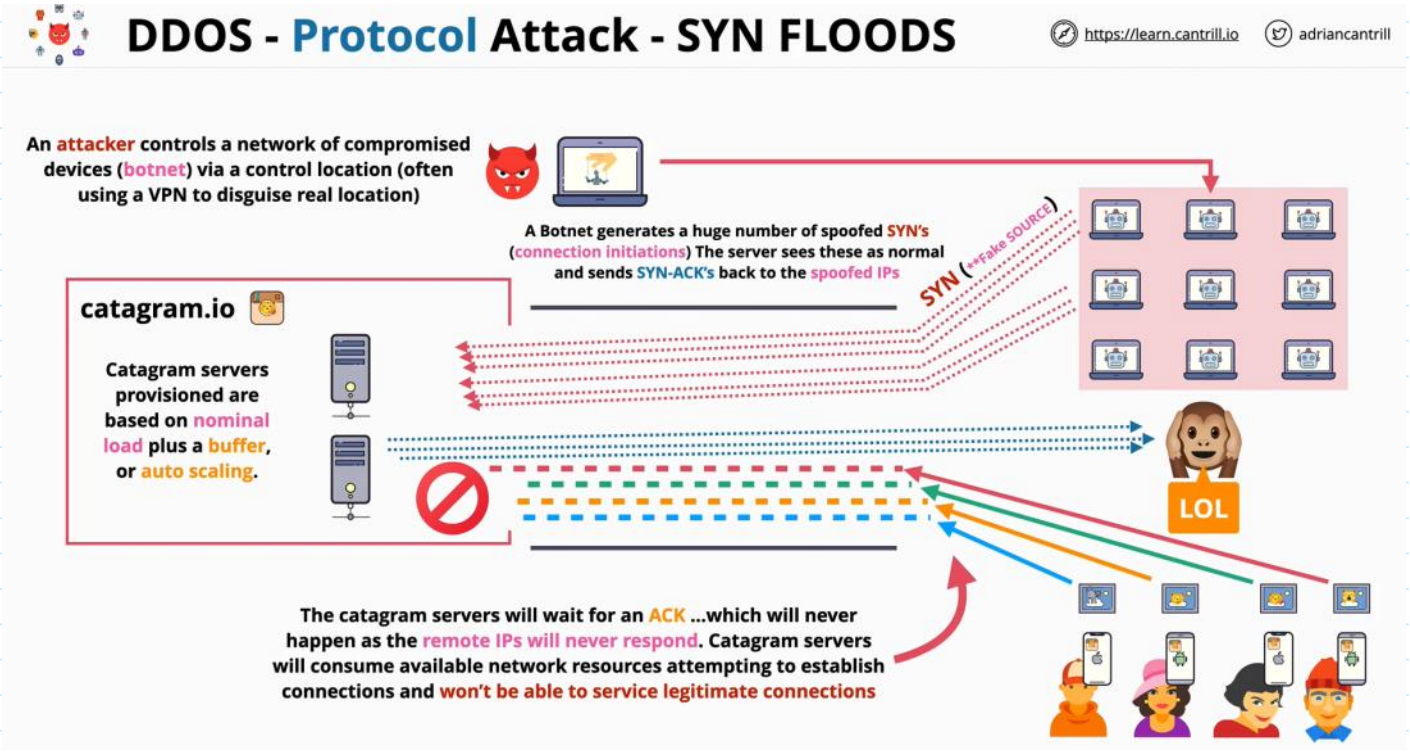
DDoS are attacks designed to **overload** websites.



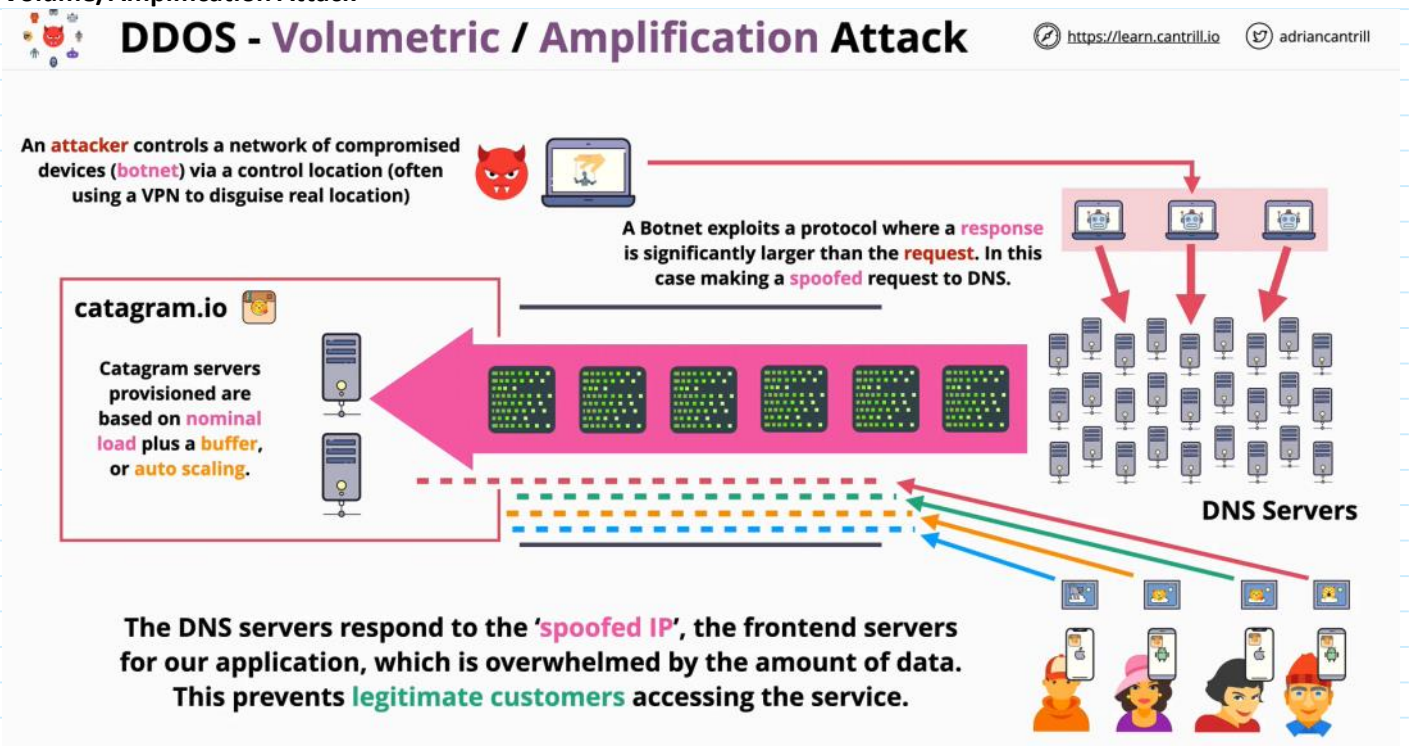
Application Layer Attack



Protocol Attack - SYN Floods



Volume/Amplification Attack



The attacks cannot be avoided with normal network securities measures. Mitigating a DDoS attack, you need to be careful to not block real users!

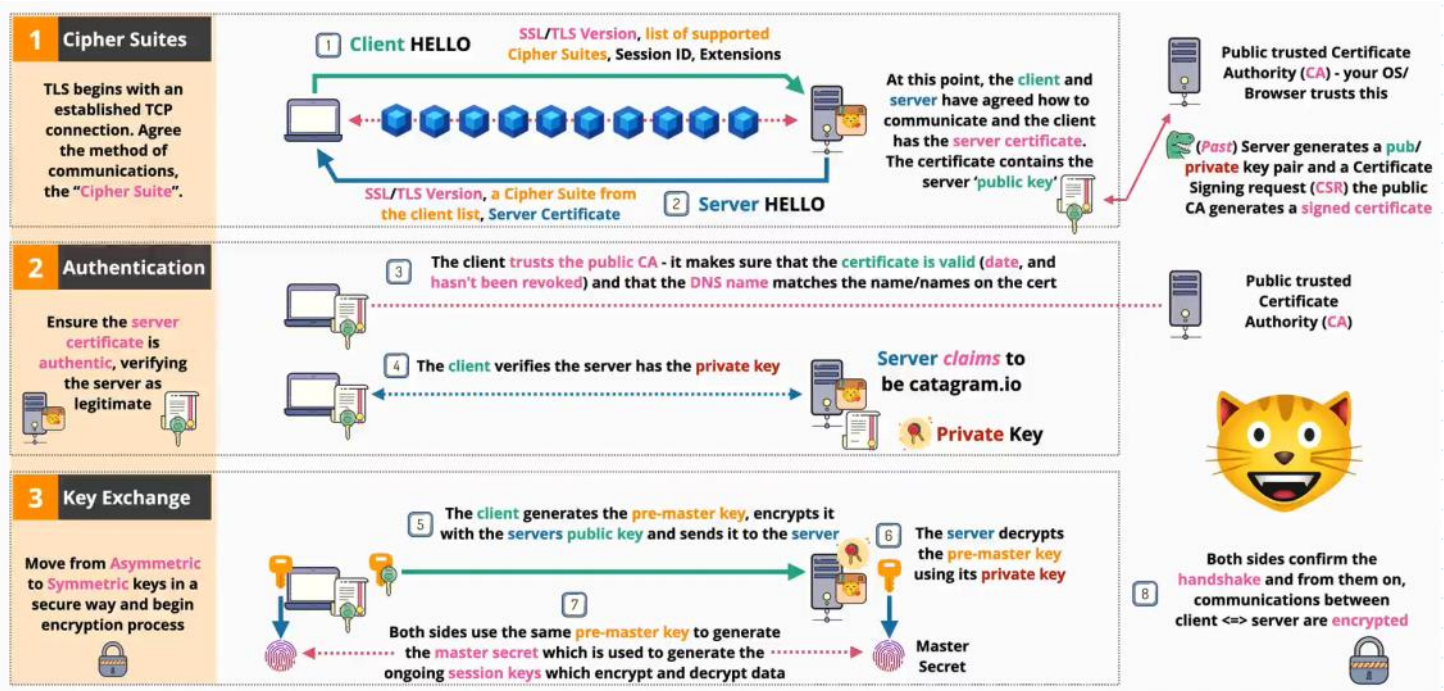
SSL and TLS

Thursday, June 10, 2021 1:52 PM

SSL -> Secure Sockets Layer
TLS -> Transport Layer Security

They both focus on the **privacy** and **data integrity** between the **client** and the **server**.

Privacy - communication are encrypted (asymmetric)
Identity verification - client/server verified
Reliable connection - protection against alteration



Virtual Private Clouds (VPC)

Tuesday, June 15, 2021 3:56 PM

VPC is used to create private networks inside AWS, and is used to connect AWS private networks to the on premises networks when creating a hybrid environment.
VPC is also the service that allows you to connect to other cloud platforms when creating a multi-cloud deployment.

- **VPC** is a virtual network inside AWS.

There are two types of VPC:

1. Default VPC - **maximum** one per region and can be deleted, removed or recreated
   a. created by AWS
   b. they come preconfigured in a very specific way
   c. they are less flexible than custom VPCs
   d. always have the CIDR: **172.31.0.0/16**
   e. **everything placed in the VPC subjects is assigned a public IPv4**
2. Custom VPCs - as many as you want
   a. they can be configured as they need to be
   b. require the configuration, end to end in **detail**
   c. they are **private** by default

- VPCs **cannot communicate** in between them, even if they are in the same AWS account, unless is stated otherwise. Basically, VPCs are by default **entirely private.**

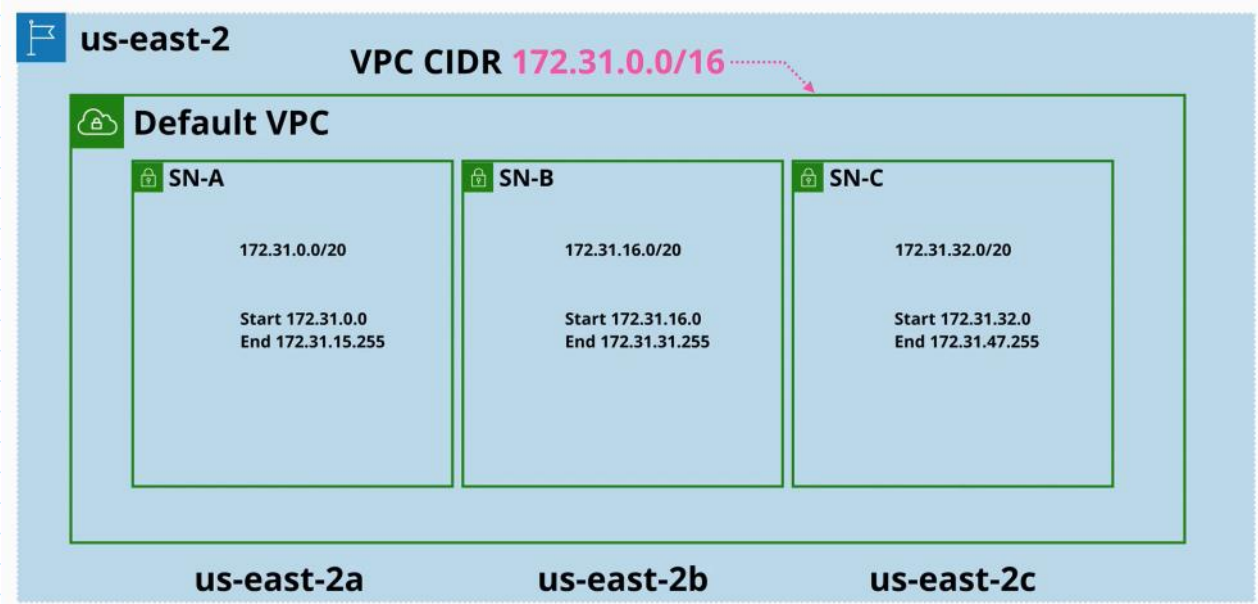
Default VPC

- Every VPC is allocated a range of IP addresses, called **VPC Cider**. Everything inside a VPC, uses the CIDER range of that VPC. Is anything needs to communicate with that VPC (if is allowed) it needs to communicate to that VPC CIDR.

- The default VPC gets only one CIDR: **172.31.0.0/16**

This is one of the strengths, is always configured by default in the same way:

- has one subnet in every availability zone in its region
- each subnet uses a part of the VPC's range of IP addresses
- these cannot be the same or overlap as other subnets in the VPC
- this makes the VPCs **resilient**



Custom VPC

Custom VPCs can have multiple CIDR ranges.

Network IDs and Subject Masks

Tuesday, July 13, 2021 4:49 PM

192.168.1.0/24

/24 is CIDR - tell you how many binary digits are turned on in a sub mask

/**24** - > **11111111.11111111.11111111**.00000000 - > 255.255.255.0
/16 - > 11111111.11111111.00000000.00000000 -> 255.255.0.0
/32 - > 11111111.11111111.11111111.11111111 -> 255.255.255.255

Mask: 255.255.248.0 -> **11111111.11111111.11111**000.00000000 -> **/21**
192.168.40.55 - > what is the network **ID**? can be used for the hosts

Network ID: 192.168.40.0/21
Sub mask: 255.255.248.0

Mask: 255.255.255.192 - > 11111111.11111111.11111111.11000000 -> **/26**
192.128.45.55 -> 11000000.10000000.00101101.00110111
Network ID: 192.128.45.0/26

VPC Sizing and Structure

Tuesday, July 13, 2021 4:17 PM

A **custom VPC** is a private network inside AWS.

Creating a VPC

1. decide the range the VPC will use - **VPC Cider**
    - **more than one can be added -** the IP range needs to be known in **advance**

Considerations

1. What size the VPC should be?
    - how many services can fit into that VPC
        - each service has one or more IPs and those occupy space inside the VPC
2. Are there any other networks that we will **use** or **interact** with
    - overlapping or duplicate ranges will made the network communication difficult
3. What other **ranges** other
    a. VPCs use
    - cloud environments
    - premises networks
    - partners or vendors
    - **avoid ranges that are used by other party ranges use and that you interact with**
4. Try to predict the future
5. Consider the structure of the VPC
    - for a given IP range allocated to a VPC it will need to be broken down further
    - tiers: web tier, application tier, database tier - depend on the VPC architecture
6. Avoid common ranges
7. **A good infrastructure is as much as good design as it is technically implemented**

★ A **VPC** can be the **smallest /28** and the **maximum /16**

Animals4Life example:

When designing a VPC for the business, we cannot use any of this IP addresses spaces.

The Azure IP address is the same as the default VPC address

- that means we cannot use the default VPC for anything in production
- but that is okey, as when is possible we should avoid using the default VPC

A **VPC** can be the **smallest /28** and the **maximum /16**

When creating a VPC:

- **CAUTION**
- **use a 10. IP address**
  - **10.0 NO - as is used by most used**
  - **10.1 NO - as everyone uses this to avoid 10.0**
  - **10.16 YES - would be a good start**
- **when thinking about the number of networks a business will need**
  - we will start at 10.16 and end at 10.128 because that is used by Google
  - the number of regions required can be determinates in how many ranges a business requires
    - in how many regions the business will ever operate in
    - and then add a few as a buffer
  - reserve 2+ networks per region being used per account

VPC Sizing

VPC Size	Netmask	Subnet Size	Hosts/Subnet*	Subnets/VPC	Total IPs*
Micro	/24	/27	27	8	216
Small	/21	/24	251	8	2008
Medium	/19	/22	1019	8	8152
Large	/18	/21	2043	8	16344
Extra Large	/16	/20	4091	16	65456

Important questions

"A subnet, or subnetwork, is a network inside a network. Subnets make networks more efficient. Through subnetting, network traffic can travel a shorter distance without passing through unnecessary routers to reach its destination."

- how many **subnets** will you need?
- how many **IPs total**?
- how many **per subnet**?

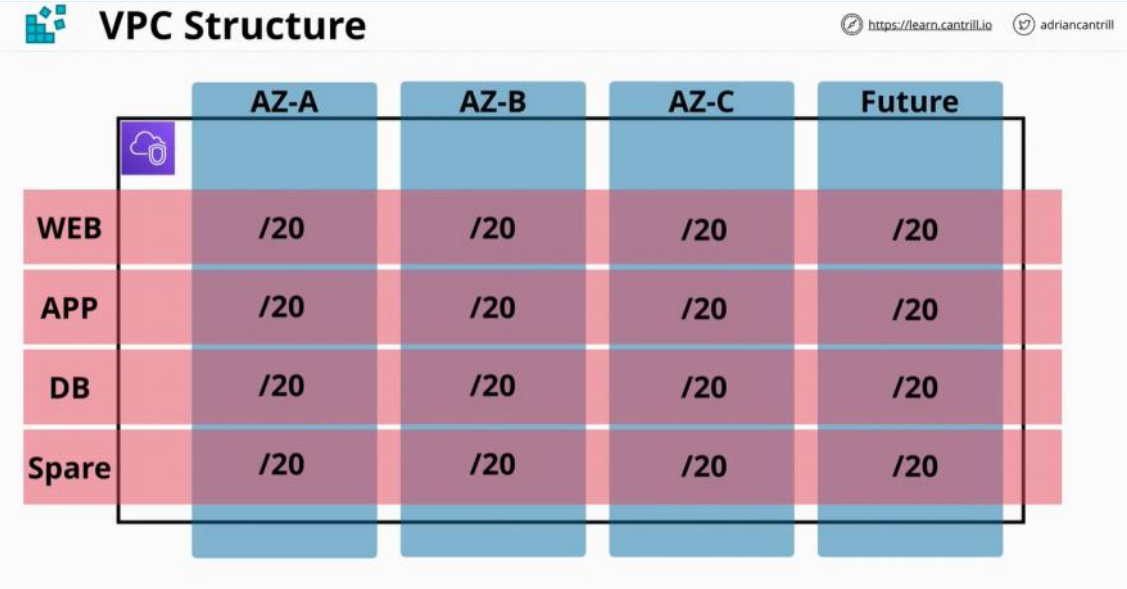
Services run on subnets, not directly from VPC.

- a subnet is allocated in one availability zone
  - how many availability zone the VPC will use?
    - this decision impacts availability an resiliency and it depends somehow on the region the VPC is in
    - is good to start with 3 availability zones
      - is a good practice as it will work in almost any region
      - is good to add a spare as things grow as time passes
      - => **this means 4 availability zones**
    - divide the VPC in 4 smaller networks: 4 subnets
      - if we started at /16 we would have four /18 subnets
      - each tier has its own subnet in its availability zone

Remember:

Each time a prefix is increased, subnets are created

- from 16 to 17 it creates 2 networks
- from 16 to 18 it creates 4 networks
- from 16 to 19 it creates 8 networks
- from 16 to 20 it creates 16 networks



Animals4lifeIPAddressingPlan

10.X Network	Usage Level 1					
10.1						
10.2						
10.3						
10.4	COMMON					
10.5	RANGES					
10.6	AVOID					
10.7						
10.8						
10.9						
10.10						
10.11						
10.12						
10.13						
10.14						
10.15						
10.16	ANIMALS4LIFE	USREGION1	GENERAL ACC	VPC1		
10.17	ANIMALS4LIFE	USREGION1	GENERAL ACC	VPC2		
10.18	ANIMALS4LIFE	USREGION1	GENERAL ACC	VPC3		
10.19	ANIMALS4LIFE	USREGION1	GENERAL ACC	VPC4		
10.20	ANIMALS4LIFE	USREGION1	PROD ACC	VPC1		
10.21	ANIMALS4LIFE	USREGION1	PROD ACC	VPC2		
10.22	ANIMALS4LIFE	USREGION1	PROD ACC	VPC3		
10.23	ANIMALS4LIFE	USREGION1	PROD ACC	VPC4		
10.24	ANIMALS4LIFE	USREGION1	DEV ACC	VPC1		
10.25	ANIMALS4LIFE	USREGION1	DEV ACC	VPC2		
10.26	ANIMALS4LIFE	USREGION1	DEV ACC	VPC3		
10.27	ANIMALS4LIFE	USREGION1	DEV ACC	VPC4		
10.28	ANIMALS4LIFE	USREGION1	RESERVED	VPC1		
10.29	ANIMALS4LIFE	USREGION1	RESERVED	VPC2		
10.30	ANIMALS4LIFE	USREGION1	RESERVED	VPC3		
10.31	ANIMALS4LIFE	USREGION1	RESERVED	VPC4		
10.32	ANIMALS4LIFE	USREGION2	GENERAL ACC	VPC1		
10.33	ANIMALS4LIFE	USREGION2	GENERAL ACC	VPC2		
10.34	ANIMALS4LIFE	USREGION2	GENERAL ACC	VPC3		
10.35	ANIMALS4LIFE	USREGION2	GENERAL ACC	VPC4		
10.36	ANIMALS4LIFE	USREGION2	PROD ACC	VPC1		
10.37	ANIMALS4LIFE	USREGION2	PROD ACC	VPC2		
10.38	ANIMALS4LIFE	USREGION2	PROD ACC	VPC3		
10.39	ANIMALS4LIFE	USREGION2	PROD ACC	VPC4		
10.40	ANIMALS4LIFE	USREGION2	DEV ACC	VPC1		
10.41	ANIMALS4LIFE	USREGION2	DEV ACC	VPC2		
10.42	ANIMALS4LIFE	USREGION2	DEV ACC	VPC3		
10.43	ANIMALS4LIFE	USREGION2	DEV ACC	VPC4		
10.44	ANIMALS4LIFE	USREGION2	RESERVED	VPC1		
10.45	ANIMALS4LIFE	USREGION2	RESERVED	VPC2		
10.46	ANIMALS4LIFE	USREGION2	RESERVED	VPC3		
10.47	ANIMALS4LIFE	USREGION2	RESERVED	VPC4		
10.48	ANIMALS4LIFE	USREGION3	GENERAL ACC	VPC1		

10.45	ANIMALS4LIFE	USREGION2	RESERVED	VPC2		
10.46	ANIMALS4LIFE	USREGION2	RESERVED	VPC3		
10.47	ANIMALS4LIFE	USREGION2	RESERVED	VPC4		
10.48	ANIMALS4LIFE	USREGION3	GENERAL ACC	VPC1		

10.X Network	Usage Level 1					
10.49	ANIMALS4LIFE	USREGION3	GENERAL ACC	VPC2		
10.50	ANIMALS4LIFE	USREGION3	GENERAL ACC	VPC3		
10.51	ANIMALS4LIFE	USREGION3	GENERAL ACC	VPC4		
10.52	ANIMALS4LIFE	USREGION3	PROD ACC	VPC1		
10.53	ANIMALS4LIFE	USREGION3	PROD ACC	VPC2		
10.54	ANIMALS4LIFE	USREGION3	PROD ACC	VPC3		
10.55	ANIMALS4LIFE	USREGION3	PROD ACC	VPC4		
10.56	ANIMALS4LIFE	USREGION3	DEV ACC	VPC1		
10.57	ANIMALS4LIFE	USREGION3	DEV ACC	VPC2		
10.58	ANIMALS4LIFE	USREGION3	DEV ACC	VPC3		
10.59	ANIMALS4LIFE	USREGION3	DEV ACC	VPC4		
10.60	ANIMALS4LIFE	USREGION3	RESERVED	VPC1		
10.61	ANIMALS4LIFE	USREGION3	RESERVED	VPC2		
10.62	ANIMALS4LIFE	USREGION3	RESERVED	VPC3		
10.63	ANIMALS4LIFE	USREGION3	RESERVED	VPC4		
10.64	ANIMALS4LIFE	EUROPE	GENERAL ACC	VPC1		
10.65	ANIMALS4LIFE	EUROPE	GENERAL ACC	VPC2		
10.66	ANIMALS4LIFE	EUROPE	GENERAL ACC	VPC3		
10.67	ANIMALS4LIFE	EUROPE	GENERAL ACC	VPC4		
10.68	ANIMALS4LIFE	EUROPE	PROD ACC	VPC1		
10.69	ANIMALS4LIFE	EUROPE	PROD ACC	VPC2		
10.70	ANIMALS4LIFE	EUROPE	PROD ACC	VPC3		
10.71	ANIMALS4LIFE	EUROPE	PROD ACC	VPC4		
10.72	ANIMALS4LIFE	EUROPE	DEV ACC	VPC1		
10.73	ANIMALS4LIFE	EUROPE	DEV ACC	VPC2		
10.74	ANIMALS4LIFE	EUROPE	DEV ACC	VPC3		
10.75	ANIMALS4LIFE	EUROPE	DEV ACC	VPC4		
10.76	ANIMALS4LIFE	EUROPE	RESERVED	VPC1		
10.77	ANIMALS4LIFE	EUROPE	RESERVED	VPC2		
10.78	ANIMALS4LIFE	EUROPE	RESERVED	VPC3		
10.79	ANIMALS4LIFE	EUROPE	RESERVED	VPC4		
10.80	ANIMALS4LIFE	AUSTRALIA	GENERAL ACC	VPC1		
10.81	ANIMALS4LIFE	AUSTRALIA	GENERAL ACC	VPC2		
10.82	ANIMALS4LIFE	AUSTRALIA	GENERAL ACC	VPC3		
10.83	ANIMALS4LIFE	AUSTRALIA	GENERAL ACC	VPC4		
10.84	ANIMALS4LIFE	AUSTRALIA	PROD ACC	VPC1		
10.85	ANIMALS4LIFE	AUSTRALIA	PROD ACC	VPC2		
10.86	ANIMALS4LIFE	AUSTRALIA	PROD ACC	VPC3		
10.87	ANIMALS4LIFE	AUSTRALIA	PROD ACC	VPC4		
10.88	ANIMALS4LIFE	AUSTRALIA	DEV ACC	VPC1		
10.89	ANIMALS4LIFE	AUSTRALIA	DEV ACC	VPC2		
10.90	ANIMALS4LIFE	AUSTRALIA	DEV ACC	VPC3		
10.91	ANIMALS4LIFE	AUSTRALIA	DEV ACC	VPC4		
10.92	ANIMALS4LIFE	AUSTRALIA	RESERVED	VPC1		
10.93	ANIMALS4LIFE	AUSTRALIA	RESERVED	VPC2		
10.94	ANIMALS4LIFE	AUSTRALIA	RESERVED	VPC3		
10.95	ANIMALS4LIFE	AUSTRALIA	RESERVED	VPC4		
10.96	ANIMALS4LIFE	UNUSED				
10.97	ANIMALS4LIFE	UNUSED				

10.X Network	Usage Level 1					
10.98	ANIMALS4LIFE	UNUSED				
10.99	ANIMALS4LIFE	UNUSED				
10.100	ANIMALS4LIFE	UNUSED				
10.101	ANIMALS4LIFE	UNUSED				
10.102	ANIMALS4LIFE	UNUSED				
10.103	ANIMALS4LIFE	UNUSED				
10.104	ANIMALS4LIFE	UNUSED				
10.105	ANIMALS4LIFE	UNUSED				
10.106	ANIMALS4LIFE	UNUSED				
10.107	ANIMALS4LIFE	UNUSED				
10.108	ANIMALS4LIFE	UNUSED				
10.109	ANIMALS4LIFE	UNUSED				
10.110	ANIMALS4LIFE	UNUSED				
10.111	ANIMALS4LIFE	UNUSED				
10.112	ANIMALS4LIFE	UNUSED				
10.113	ANIMALS4LIFE	UNUSED				
10.114	ANIMALS4LIFE	UNUSED				
10.115	ANIMALS4LIFE	UNUSED				
10.116	ANIMALS4LIFE	UNUSED				
10.117	ANIMALS4LIFE	UNUSED				
10.118	ANIMALS4LIFE	UNUSED				
10.119	ANIMALS4LIFE	UNUSED				
10.120	ANIMALS4LIFE	UNUSED				
10.121	ANIMALS4LIFE	UNUSED				
10.122	ANIMALS4LIFE	UNUSED				
10.123	ANIMALS4LIFE	UNUSED				
10.124	ANIMALS4LIFE	UNUSED				
10.125	ANIMALS4LIFE	UNUSED				
10.126	ANIMALS4LIFE	UNUSED				
10.127	ANIMALS4LIFE	UNUSED				
10.128	GOOGLE					
10.129	GOOGLE					
10.130	GOOGLE					
10.131	GOOGLE					
10.132	GOOGLE					
10.133	GOOGLE					
10.134	GOOGLE					
10.135	GOOGLE					
10.136	GOOGLE					
10.137	GOOGLE					
10.138	GOOGLE					
10.139	GOOGLE					
10.140	GOOGLE					
10.141	GOOGLE					
10.142	GOOGLE					
10.143	GOOGLE					
10.144	GOOGLE					
10.145	GOOGLE					
10.146	GOOGLE					

10.X Network	Usage Level 1					
10.147	GOOGLE					
10.148	GOOGLE					
10.149	GOOGLE					
10.150	GOOGLE					
10.151	GOOGLE					
10.152	GOOGLE					
10.153	GOOGLE					
10.154	GOOGLE					
10.155	GOOGLE					
10.156	GOOGLE					
10.157	GOOGLE					
10.158	GOOGLE					
10.159	GOOGLE					
10.160	GOOGLE					
10.161	GOOGLE					
10.162	GOOGLE					
10.163	GOOGLE					
10.164	GOOGLE					
10.165	GOOGLE					
10.166	GOOGLE					
10.167	GOOGLE					
10.168	GOOGLE					
10.169	GOOGLE					
10.170	GOOGLE					
10.171	GOOGLE					
10.172	GOOGLE					
10.173	GOOGLE					
10.174	GOOGLE					
10.175	GOOGLE					
10.176	GOOGLE					
10.177	GOOGLE					
10.178	GOOGLE					
10.179	GOOGLE					
10.180	GOOGLE					
10.181	GOOGLE					
10.182	GOOGLE					
10.183	GOOGLE					
10.184	GOOGLE					
10.185	GOOGLE					
10.186	GOOGLE					
10.187	GOOGLE					
10.188	GOOGLE					
10.189	GOOGLE					
10.190	GOOGLE					
10.191	GOOGLE					
10.192	GOOGLE					
10.193	GOOGLE					
10.194	GOOGLE					
10.195	GOOGLE					

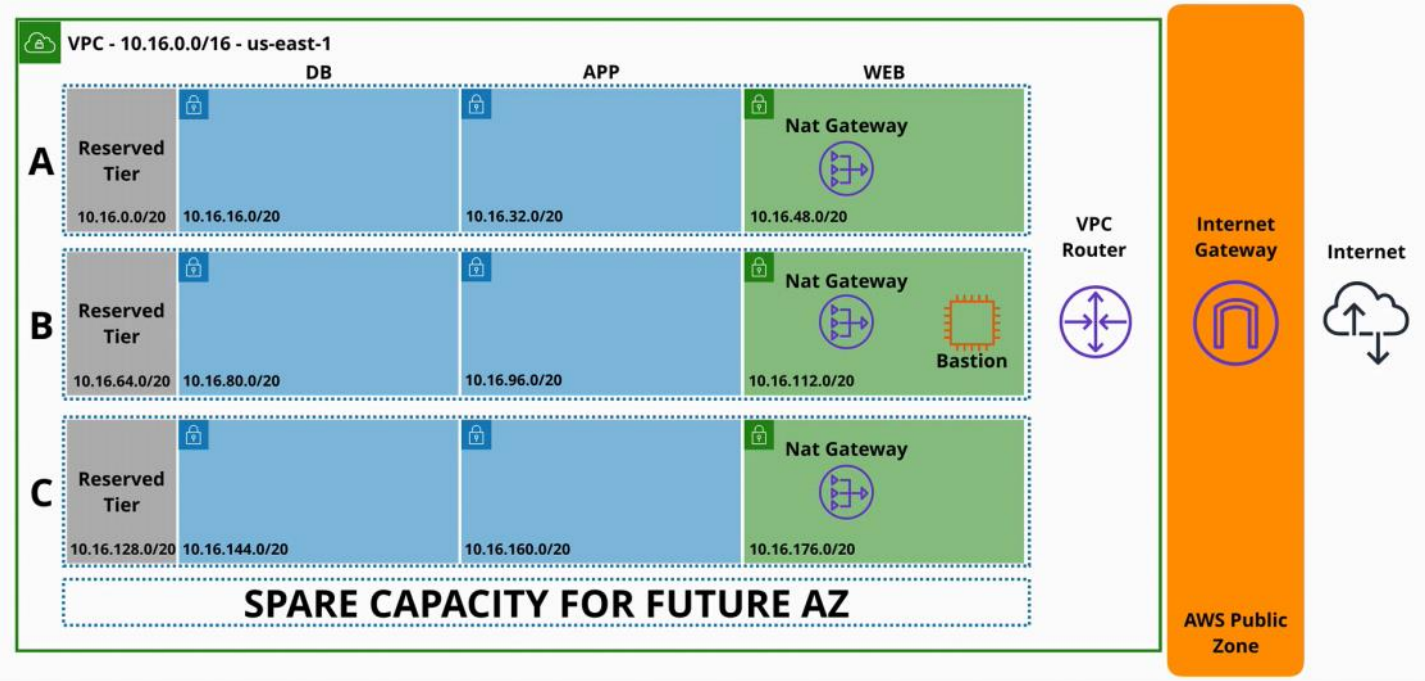
10.X Network	Usage Level 1					
10.196	GOOGLE					
10.197	GOOGLE					
10.198	GOOGLE					
10.199	GOOGLE					
10.200	GOOGLE					
10.201	GOOGLE					
10.202	GOOGLE					
10.203	GOOGLE					
10.204	GOOGLE					
10.205	GOOGLE					
10.206	GOOGLE					
10.207	GOOGLE					
10.208	GOOGLE					
10.209	GOOGLE					
10.210	GOOGLE					
10.211	GOOGLE					
10.212	GOOGLE					
10.213	GOOGLE					
10.214	GOOGLE					
10.215	GOOGLE					
10.216	GOOGLE					
10.217	GOOGLE					
10.218	GOOGLE					
10.219	GOOGLE					
10.220	GOOGLE					
10.221	GOOGLE					
10.222	GOOGLE					
10.223	GOOGLE					
10.224	GOOGLE					
10.225	GOOGLE					
10.226	GOOGLE					
10.227	GOOGLE					
10.228	GOOGLE					
10.229	GOOGLE					
10.230	GOOGLE					
10.231	GOOGLE					
10.232	GOOGLE					
10.233	GOOGLE					
10.234	GOOGLE					
10.235	GOOGLE					
10.236	GOOGLE					
10.237	GOOGLE					
10.238	GOOGLE					
10.239	GOOGLE					
10.240	GOOGLE					
10.241	GOOGLE					
10.242	GOOGLE					
10.243	GOOGLE					
10.244	GOOGLE					

10.X Network	Usage Level 1					
10.245	GOOGLE					
10.246	GOOGLE					
10.247	GOOGLE					
10.248	GOOGLE					
10.249	GOOGLE					
10.250	GOOGLE					
10.251	GOOGLE					
10.252	GOOGLE					
10.253	GOOGLE					
10.254	GOOGLE					
10.255	GOOGLE					

Custom VPCs

Wednesday, July 14, 2021 3:27 PM

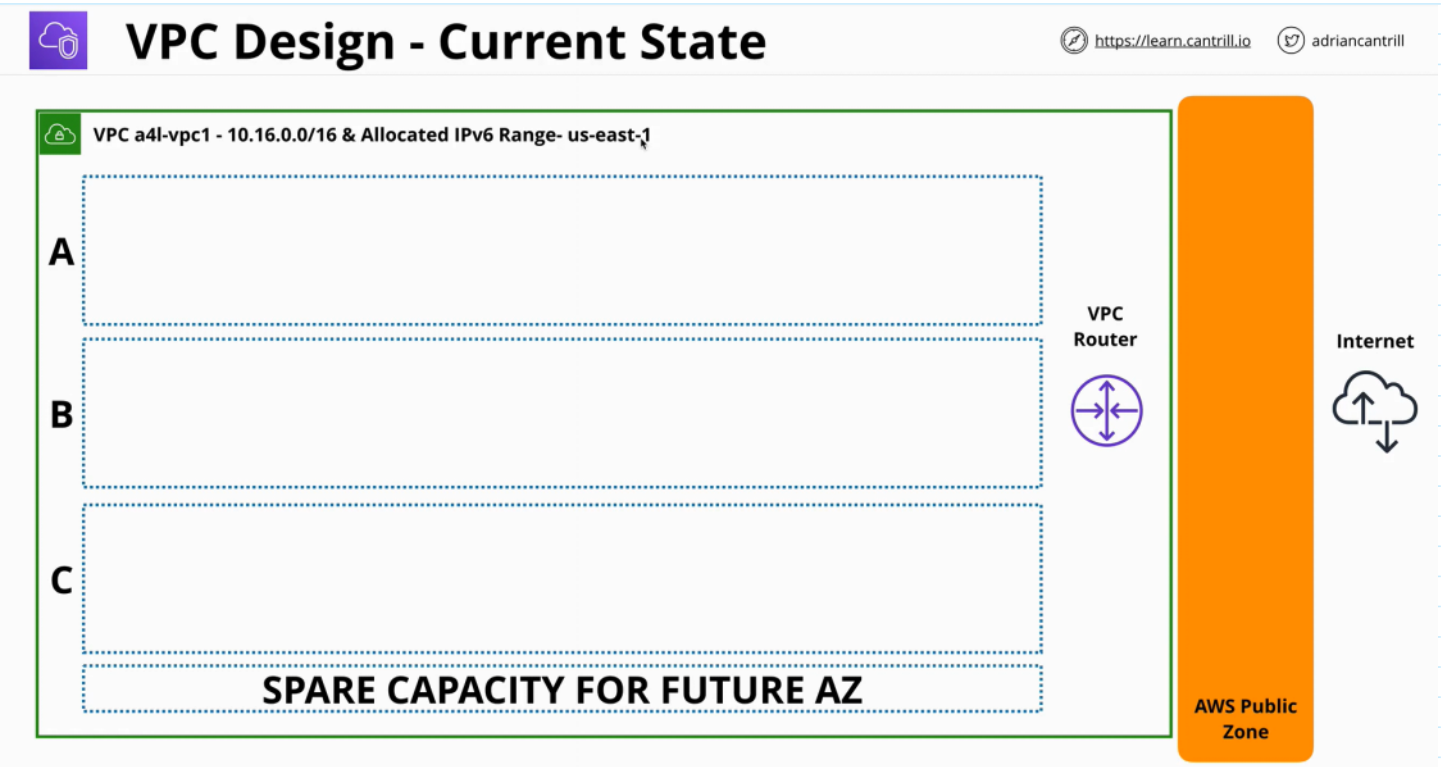
One of the benefits of VPCs is that you can start up simple, and layer components piece by piece.



VPCs are regionally isolated and resilient service.

- ★ - is created in a region and it operates from all availability zones from that region
- ★ - it allows the creation of isolated networks in AWS
    - even in a single region within one account there can be multiple isolated networks
- ★ - nothing is allowed **in** or **out** of a VPC without **explicit configuration**
- ★ - custom VPC has flexible configuration - simple and multi-tier
    - they also support hybrid networking - other cloud platforms and on-premises networks
    - have default or dedicated tenancy - if resources inside a VPC are having shared or dedicated hardware
        - **DEFAULT** - you can choose on a per resource basis later on when you provision resources as whether it goes on dedicated or shared hardware
        - **DEDICATED TENANCY** - it is locked in, any resources that are created inside the VPC have to be on dedicated hardware
- ★ - are using IPv4 private CIDR Blocks and public IPs
    - the private CIDR block is the main communication channel with the VPC
        - by default 1 primary private IPv4 CIDR block is allocated to a VPC
            - this has two restrictions:
                1. at its smallest it can be /28 prefix (the entire VPC has 16 IP addresses)
                2. at its largest it can be /16 prefix (65,536 IP addresses)
        - other optional secondary IPv4 blocks can be added after creation
    - public IPs are used when resources need to be made public
- ★ - it can be configured to use IPv6 by assigning /56 CIDR block to the VPC
    - they do not have the concept of private or public - are all publicly routable by default
        - when used, explicit allow connectivity to and from the public internet is required
- have fully featured DNS
    - provided by Route53
    - is available by the base IP address of the VPC + 2 (if the address 10.0.0.0 the DNS will be 10.0.0.2)
    - Two options for the DNS:
        1. **enableDnsHostNames** - this indicates weather instances with public IP addresses in a VPC are given public DNS host names
            - if is set to TREU, then instances do get a public DNS name
        2. **enableDnsSupport** - it indicates weather the DNS is enabled or disabled in the VPC

- if enabled then instances in the VPC can use the DNS IP address (VPC +2 IP)



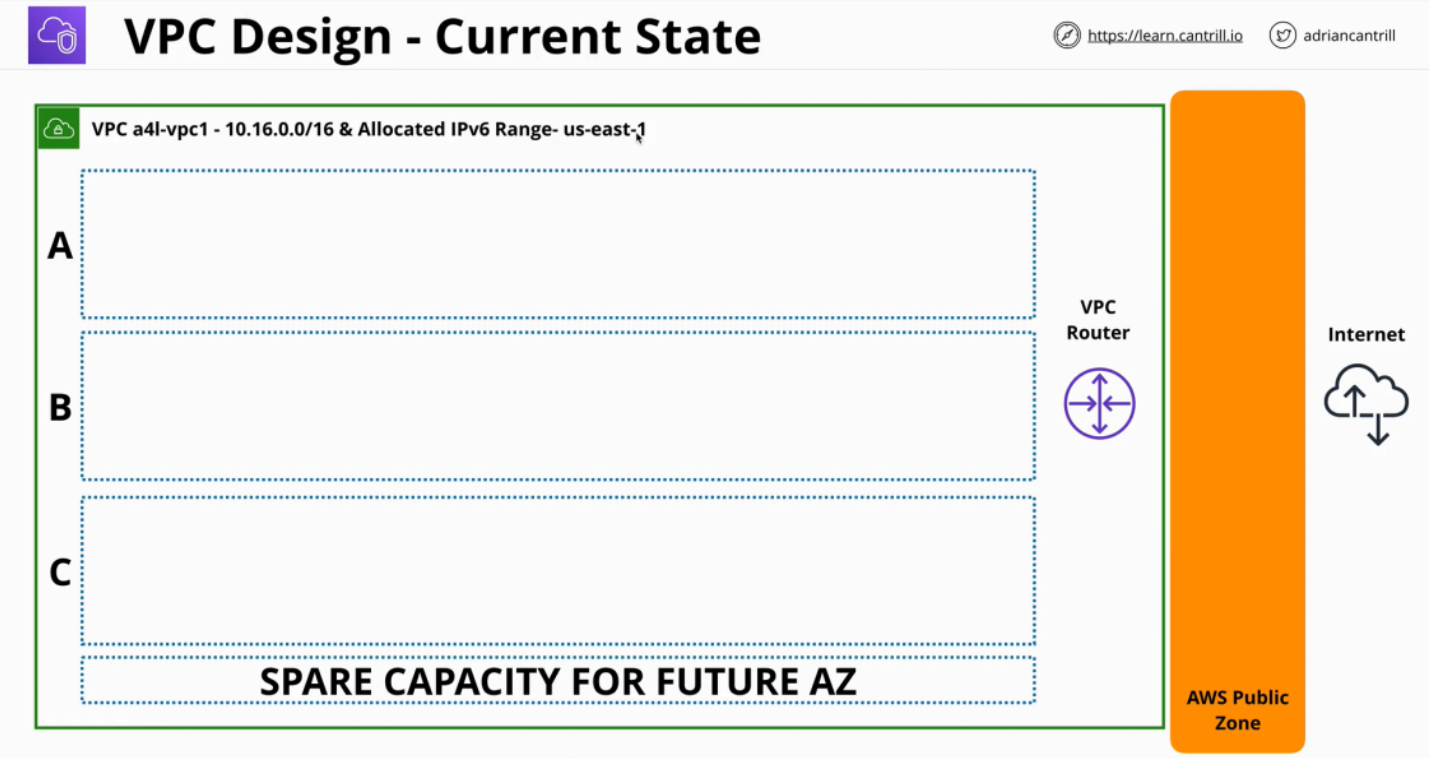
VPC Subnets

Thursday, July 15, 2021 11:23 AM

Subnets are what services run from inside VPC

[Subnet Calculator | IP Subnet Mask Calculator for IPv4 - Site24x7]

- how add structure, functionality and resilience to VPCs



This is the structure that was left off, since last lecture. (We only created a VPC). Today we will create an internet structure using subnets.

By default subnets are **private** and they require some configuration to make it **public**.

A **subnet** is a feature of VPC that is **AZ resilient** - a sub-network of the VPC.

- **a is hosted in a AZ, if that AZ fails, that the subnet itself fails**
- to make a system highly-available, we put different components of the system into different AZ , to make sure that if one AZ fails, the whole system doesn't fail
  - the way we do this, is to put different components in different subnets, where each are located in a specific AZ
- once a AZ has been chosen for a subnet
  - it can never be changed
  - and cannot spread over multiple AZs
  - ★ **one subnet is in one AZ**
- ★ **an AZ can have zero to lots of subnets**
- by default a subnet uses a IPv4 and is allocated an IP-version-four CIDR
  - the CIDR is a subset of the VPC CIDR block
  - has to be in the range of IPs allocated to the VPC
  - ★ the CIDR used by a subnet cannot overlap other subnets
- optionally IPv6 CIDR block can be allocated to a subnet
  - as long as the VPC is also enabled for IPv6
  - /64
- subnets from a VPC can communicate with other subnets from the same VPC
  - the isolation happens at the permitter of the VPC, not inside it
  - internally there is free communication between subnets by default
- some IP addresses inside subnets are reserved - hence they cannot be used

there are 5 in total	example **10.16.16.0/20** (10.16.16.0 -> 10.16.31.255)
1. first IP address	**10.16.16.0** -> represents the starting address of the network
2. network + 1	**10.16.16.1** -> it is used by the VPC router

	• the logical network device which moves data between subnets and **in** and **out** of the VPC if is allowed
3. network + 2	**10.16.16.2** -> is used for DNS
4. network + 3	**10.16.16.3** -> reserved for future requirements
5. last IP address	**10.16.16.255** -> used for broadcast • broadcast is not supported inside a VPC

- A VPC has a configuration object applied to it called a **DHCP option set**
  - stands for **Dynamic Host Configuration Protocol**
  - is how computing devices receive IP address automatically
  - controls the DNS servers, NTP servers, NET bile
  - **you can create option sets, but you CANNOT edit them**
- can be set two IP allocation options
  1. controls if resources are allocated a public IPv4 address in addition to the private subnet
  2. controls if resources deployed into the subnet are also given a IPv6 address
     - both option are defined at a subnet level and flow on any resources inside that's subnet

VPC Routing, Internet Gateway & Bastion Hosts

Friday, July 16, 2021 12:04 PM

VPC Router

A **VPC router** is a highly available device

- is present in every VPC (default or custom)
- moves traffic from point A to point B
- it runs in all AZ that the VPC uses
- has a network interface in every subnet in the VPC - **network + 1**
- **without any configuration, the router simply routes traffic between subnets**
- can create route tables which influence what to do with the traffic when it leaves a subnet
    - each subnet has it own route table
- a VPC is created with a **main route table**
    - when a route table is associated with a subnet, then the main rout table is disassociated
    - a subnet can **only have one route table** associated with it!
    - a route table can be associated with **many subnets**

The process

When traffic leaves the subnet that the route table is associated with

- the VPC router reviews the IP packets
    - packets they have a **source address** and a **destination address** along with the data that is transmitted
- the VPC router looks at the destination address of all packets leaving the subnet
- **checks the route table for a matching destination address**
    - does this my checking the destination field of a route
        - could match a specific IP
        - could match a specific network
        - could match a default route(0.0.0.0/0)
        - if there are multiple matches, **then the prefix is used as a priority**
            - the higher the prefix value, the more specific the routers and the higher priority that route has
    - when a single route is selected, then the VPC router forwards that traffic through its destination which is determined by the **target** field on the route (local or AWS gateway)
        - local - in the VPC it self
        - local routes always take priority

Internet Gateway (IGW)

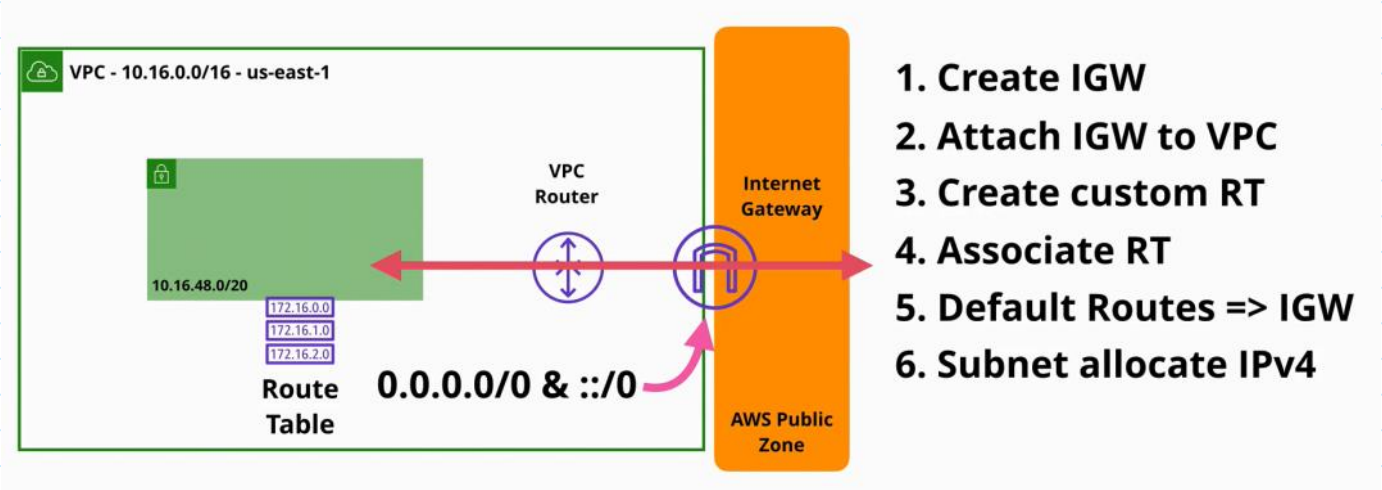
Internet Gateway is one of the most important add-on features available within a VPC.

- it is a **region resilient** gateway that can be attached to a VPC
- there is a one to one relationship between a VPC and gateways
    - a VPC can have one internet gateway, or zero
    - an internet gateway can have one VPC, or zero
- it is at the boarder or the AWS Public Zone
- allows traffic between the VPC and the Internet or the AWS public zone
- it is a manages service
- AWS handles the performance

The process

1. we attach an internet gateway to a VPC
    - it means that is available to be used for use inside the VPC
    - we can use it as a target within the route tables

2. we add the default route to the route's route table with its target being the internet gateway
3. we configure the subnets to allocate IPv4 and optionally IPv6
4. the subnet is classified now as being a public subnet
    - any services inside that subnet with public IP addresses can communicate with the internet (and vice-versa)
    - they also can communicate with the AWS public zone (as long as there are no other security limitations)

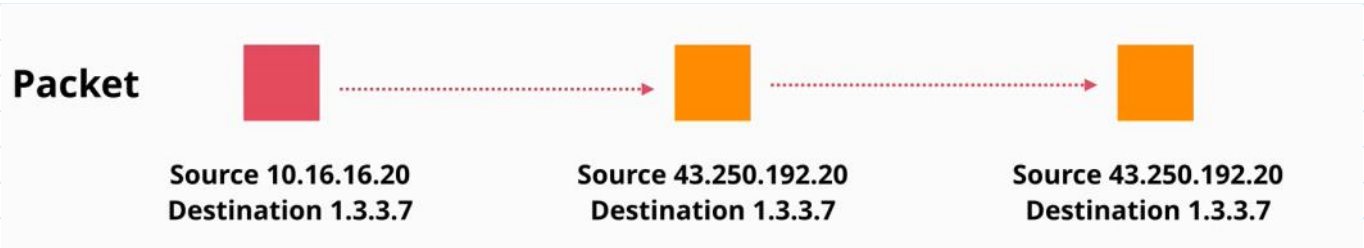


IPv4 addresses with Internet Gateway

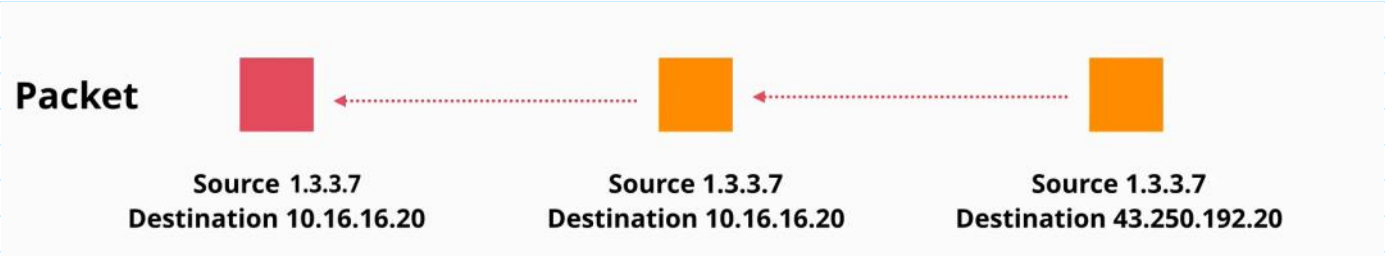
A public IPv4 (43.250.192.20 in our example) never touches the actual services inside a VPC.

- a record of the address is created which the internet gateway maintains
    - it links the private IP to its allocated public IP
    - the instance itself, is not allocated with the public IP - the gateway is

The instance communicating with the Linux server:



The Linux server communicates with the Instance:



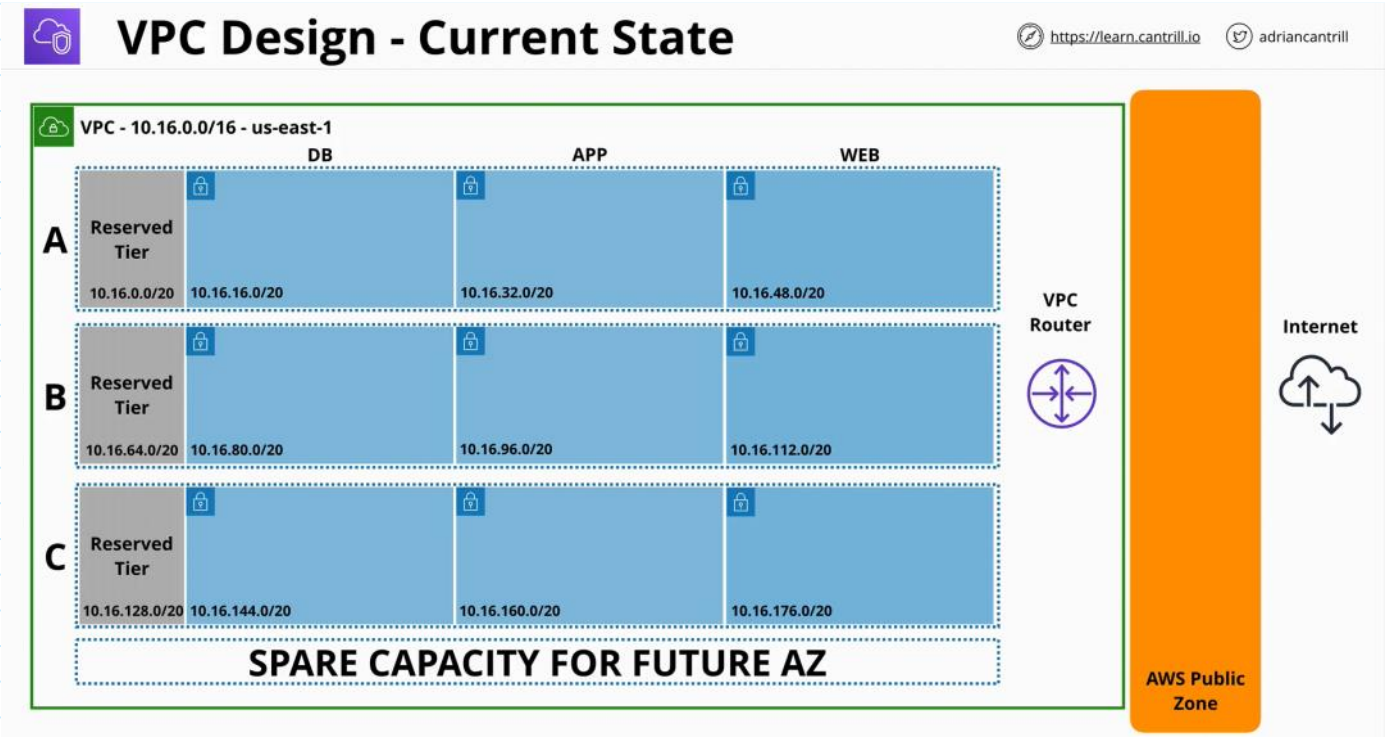
- At any point, the OS on the EC2 instance is aware of the public IP address - the internet gateway is!

- With IPv6 all IP address are natively publicly routable! - the OS does have the IPv6 address configure on it, and all the internet gateway does is to pass traffic from an instance to the internet server and back again

Bastion Host / Jumpbox

Bastion Host or Jumpbox is an instance in a public subnet

- they are used to allow incoming management connection
- when connected you can access internal-only VPC resources
- are generally used as:
    - management point
    - entry point for private only VPC

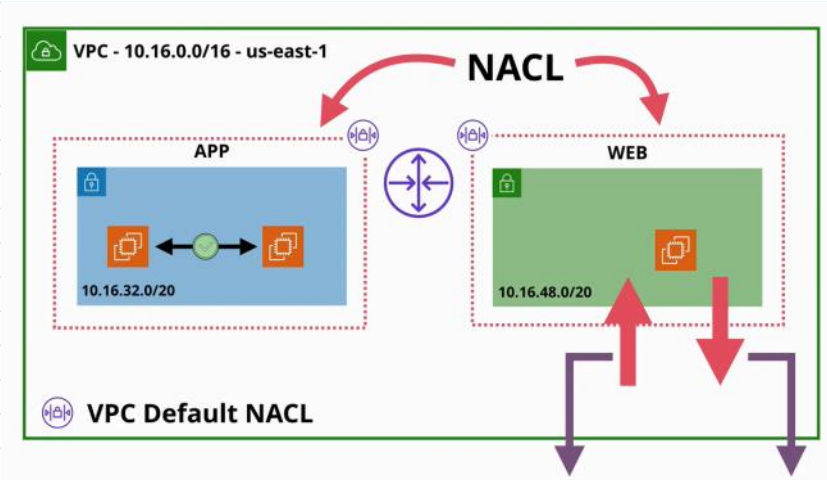


Network Access Control Lists (NACLs)

Saturday, July 17, 2021 11:47 AM

Network Access Control Lists (NACLs) are a security feature of AWS that is seen as a firewall which surrounds VPC subnets

- all VPCs are created with a default network ACL associated with all subnets in that VPC **by default**
- NACLs are used when traffic enters or leaves a subnet
    - because they are associated with subnets (and not services) they are used when data crosses the subnet boundary
    - are formed of two sets of rules
        - **inbound**
        - **outbound**
    - **the rules are processed as:**
        1. in order - lowest rule first
        2. when a rule is matched an action is taken and the process has stopped
            - the rule can **allow** or **deny** the traffic to cross the subnet permitter
        - each rule has a number of fields:
            - type, protocol, port range -> are concerned with the type of traffic going in or out (SSH, HTTP, HTTPS, and others)
            - for inbound rules there is the source field - where the traffic is coming from
            - for outbound rules there is the destination field - where the traffic is going to
        - if all fields match, the first rule matched it determinates which action is taken: to allow or deny the traffic
        - if no rule matches the traffic, then the traffic is denied
            - **there is always an implicit deny**
            - the Asterix rule can never be removed, or edited
            - processed if nothing else matches
        - by default there is also a rule number 100 - that can be edited or removed
            - is an explicit catch-all allow - **never block any traffic**



- Processed in order
- Low Rule # First
- Stops if matched

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Inbound

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Outbound

These NACLs are the most difficult security feature inside a VPC

- networks ACLs are stateless - they are unable to assess the state of communication between two entities on a network
    - initiation and response are seen intendent

    - they need two different rules: for the request and for the response
- network ACLs only impact traffic which **crosses a subnet border**
    - data that enters or leaves a subnet
- can be used to explicitly allow and explicitly deny traffic
- only support using IPs , networks, ports and protocols
- they do not have any visibility of AWS logical resources
- they can be used in conjunction with security groups
- one subnet **is only associated with one** network ACL
    - if one subnet has not added a network ACL, they default to the default network ACL of the VPC
    - it needs a network ACL at all times: the custom or the default one
- rules are processed in order, and the Asterix (the explicit deny) is the last, and it applies if anything else does not match
    - processing stops at the first matching rule

Security Groups

Saturday, July 17, 2021 12:21 PM

Security groups are somehow similar to NACLs, but hey operate at a higher level of the networking stack, and integrate with AWS product and services in a much closer way.

- it is attached to a resource/service and not the subnet
- they have two sets of rules
    1. **inbound**
    2. **outbound**
- are stateful (opposite of stateless)
    - the request and the response are viewed as part of the same communication
    - only one rule is required for an inbound communication - where the return traffic is automatically included in this allow
- they match traffic almost the same as the ACL
    - type of traffic, protocol, port range and source
- they **understand logical resources**
    - they are not limited to IP and network
    - they can references services and resources
    - they can reference other security groups and even themselves
        - having a rule for itself, it means that it matches anything associated with itself
- they have a **hidden implicit deny**
    - what is not stated as allowed is implicitly denied
    - they cannot explicitly deny - if is not allowed then is denied

- **NACLs** are usually used if products do not support security groups (NAT gateways)
    - are also used when you want to add **explicit denies**
    - subnet protection
- **security groups should be used for anything else**

Network Address Translation (NAT) & NAT Gateway

Tuesday, July 20, 2021 10:55 AM

Network Address Translation (NAT)

- is a set of different processes which can adjust IP packets by changing their source or destination IP address
- **IP masquerading** provides a whole private range of IP addresses outgoing only access to the public internet and the public AWS public zone **(incoming access does not work)**
  - ★ many **private IPv4** addresses to **one public IP address**
  - this is important, because IPv4 addresses do run out

The internet gateway actually performs a type of NAT known as **static NAT**

- how a resource is allocated with a public IPv4 address

AWS has two ways to implement NAT services:

1. via EC2 instances - configured to provide the NAT gateway
   - the **NAT** gateway makes a record of the packet sent from an instance
     - destination, source address and other details which help the NAT gateway identify the specific communication in the future - **all this is recorded into a translation table**
     - this is needed as multiple instances could be communicating at once, and each instance would have multiple communication with different public internet hosts
   - after the packet details had been stored in the translation table, then the packet is adjusted by changing the source packet to match the NAT Gateway IP address (with the public routable IP address)
   - the packet is now moved from the **NAT Gateway** to the **internet gateway** by the VPC router

★ **If you need to give an instance its own public IPv4 address then the internet gateway is required**

★ **If you want to give multiple private instances, outgoing access to the internet and the AWS public zone services, the both a NAT gateway and the internet gateway are needed.**

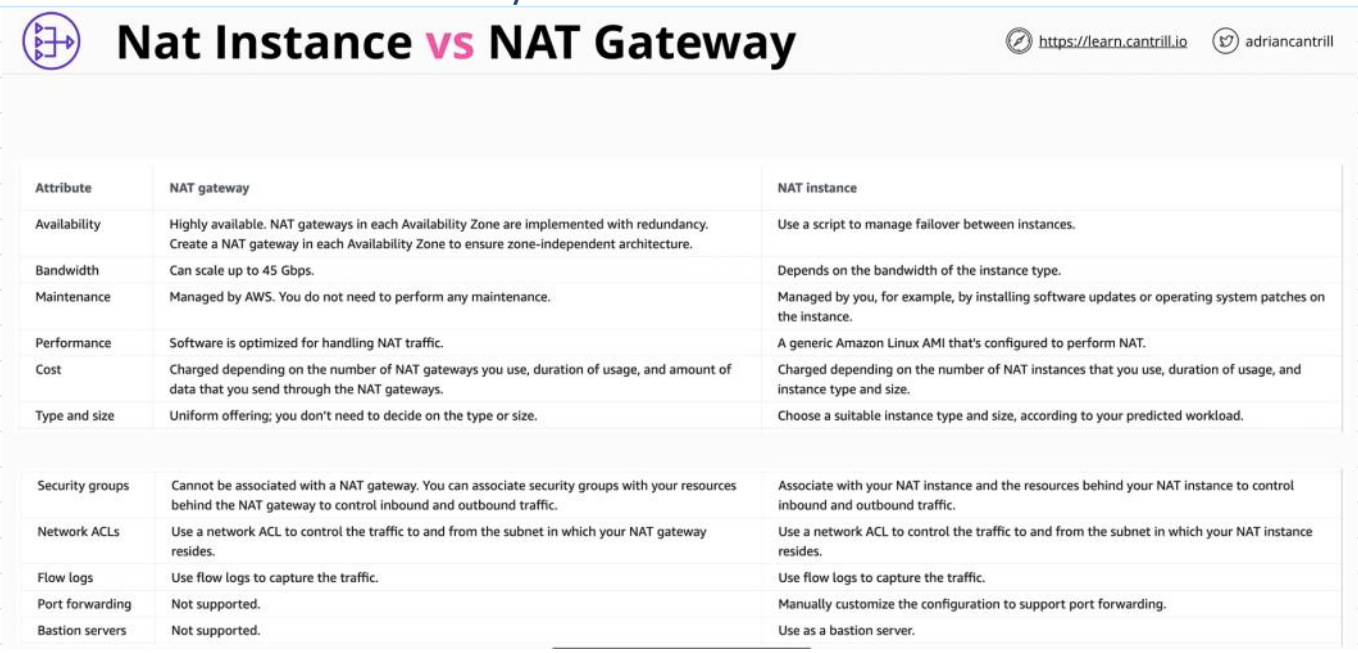
- the **NAT gateway** is needed to do the many to one translation
- the **internet gateway** is needed to translate from the NAT gateway IP address to a real public IPv4 address

NAT Gateway Product

- it needs to run from a public subnet as it needs to be assigned a public IPv4 address
- it uses a special type of IPv4 address called **Elastic IP - static IPv4 Public**
  - **IPs do not change**
  - **they are allocated ina region and the be used for any purpose**
- ★ **are AZ resilient service**
  - to make it mirror the internet gateway, then a NAT gateway is needed in each AZ that is used by a VPC and then have a route table for the private subnets in that AZ pointing at the NAT gateway in the AZ
  - so for every AZ used, a NAT gateway is needed and one route table pointing at that NAT gateway
  - **for maximum availability a NAT gateway needs is needed in every AZ**
- ★ **they are a managed service**
  - after they are deployed AWS handles everything else
  - they scale up to 45 GB/second in bandwidth
  - more gateways can be created
  - you cannot connect to its OS
- they are billed by the number that you have
  - there is an hourly charge for running a NAT gateway
  - partial hours are billed as full hours
  - is billed also per gigabyte of processed data with the same charge (as per hour)

NAT Instance vs. NAT Gateway

NAT Instance vs. NAT Gateway



Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances.
Bandwidth	Can scale up to 45 Gbps.	Depends on the bandwidth of the instance type.
Maintenance	Managed by AWS. You do not need to perform any maintenance.	Managed by you, for example, by installing software updates or operating system patches on the instance.
Performance	Software is optimized for handling NAT traffic.	A generic Amazon Linux AMI that's configured to perform NAT.
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
Type and size	Uniform offering; you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload.
Security groups	Cannot be associated with a NAT gateway. You can associate security groups with your resources behind the NAT gateway to control inbound and outbound traffic.	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.
Flow logs	Use flow logs to capture the traffic.	Use flow logs to capture the traffic.
Port forwarding	Not supported.	Manually customize the configuration to support port forwarding.
Bastion servers	Not supported.	Use as a bastion server.

- NAT instance, has a feature called **source and destination checks**, which drops any traffic that does not match the source or the destination network card of that

  NAT instances and gateways are similar as:
  - they both need a public IP address
  - they both need to run a public subnet
  - they both need a functional internet gateway

- in most situations, a **NAT gateway** is the bets choice when a NAT is needed. However, there are a few configurations where using an **EC2 based NAT instance** should be considered :
  1. if availability, bandwidth, low level of maintenance and high performance are valued then **NAT gateway** should be used
     - A NAT gateway offers high end performance, it scales and its custom designed to perform network address translation
     - A NAT instance is limited by the capabilities of the instance it is running on.
     - A NAT instance is general purpose, so it won't offer the same level of custom design performance
     - A NAT instance is a single EC2 instance running inside a AZ, so if the EC2 hardware or its AZ fails, the NAT instance will fail
     - **Benefits** if NAT instances:
       - inside an AZ is highly available
       - it will automatically recover
       - it will automatically scale
       - you can connect to them like to any other EC2 instance
       - **NAT instances are only EC2 instances, therefore you can filter the traffic using ACLs on the subnet the instance is in or security groups directly associated with that instance**
       - **NAT gateways do not support security groups - you can only use NACLs with it**
     - NAT instance is cheaper especially with high volumes of data

IP version 6

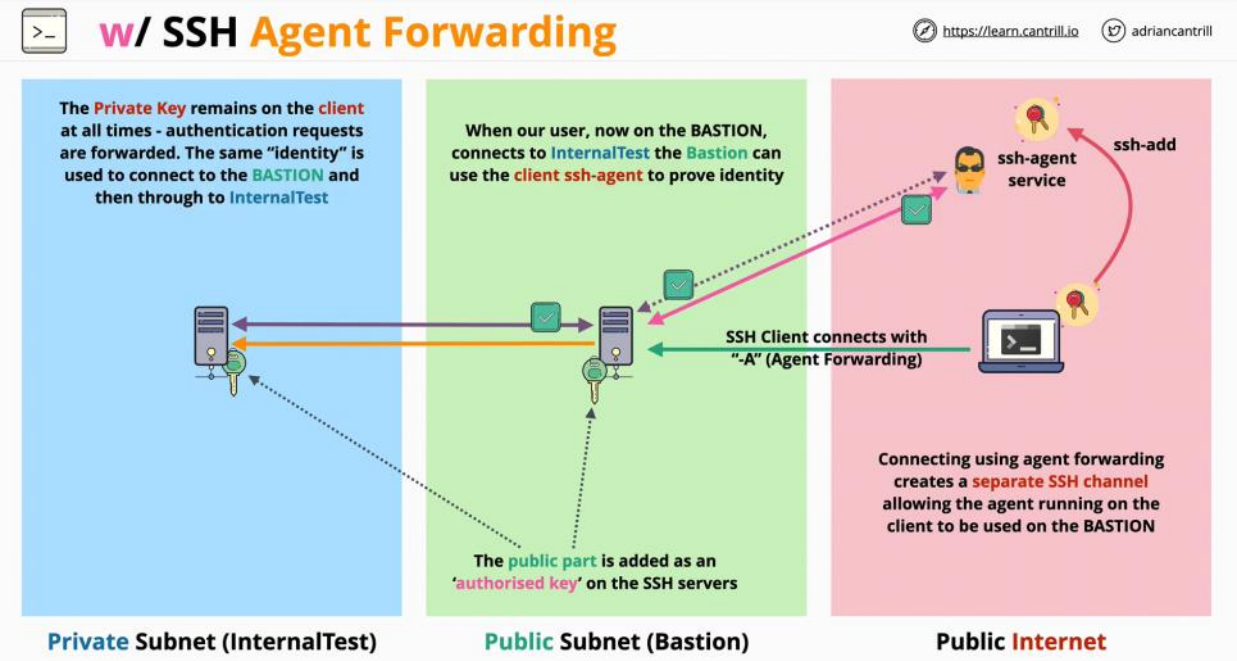
- **NAT gateway** is not required and they do not function with IPv6 because inside AWS all IPv6

addresses are publicly routable

- ::/0 in the internet gateway will allow bidirectional network access to that instance

SSH Forwarding

1. We have the private client on the public internet, the bastion in the middle and the private subnet on the right
2. a client private key is stored on the laptop which is used to connect to the bastion
3. a public key is stored on the bastion and the private server
4. on the client we have the SSH agent (which usually is present by default)
    - the command **ssh-add** and this adds our private key into the agent
    - it is stores securely and guarded carefully - but is still a copy of the key inside the ssh agent
5. when connecting to the bastion, using the special command ssh -A
    - this creates a separate channel of communication between the client and the bastion and allows the bastion to connect to the SSH agent running on the client laptop for authentication
6. then when we create an interaction between the bastion and the private server
7. the private server asks the bastion to prove the identity of the user connecting (which normally will need a copy of the private key from the bastion)
8. the bastion forwards the request to the ssh agent, as it can prove the identity as it has a copy of the private key
    - the private key never leaves the client machine



Elastic Compute Cloud - EC2

Tuesday, June 15, 2021 6:02 PM

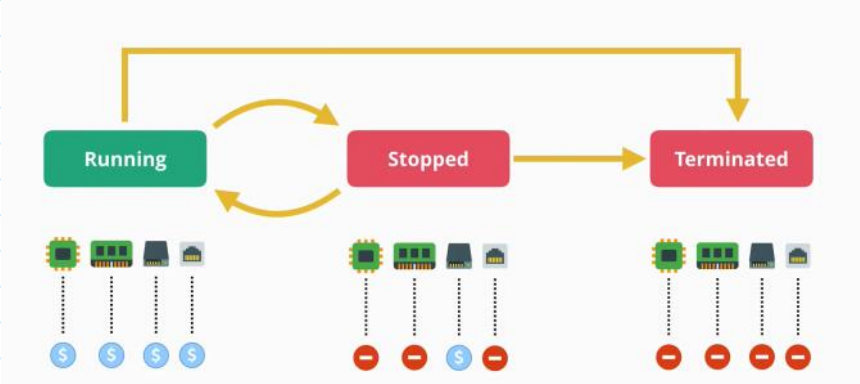
EC2 provides access to virtual machines known as instances.

EC2 is **IAAS** (infrastructure as a service)

- provides access to virtual machines (EC2 instances)
- an instance is an OS configured in a certain way with a certain set of allocated resources
- is a private AWS service by default
- for public access to an EC2 instance than the VPC running needs to support that public access
- is AZ resilient
- the developer managers the OS and everything up from the infrastructure stack
- is an on-demand billing by second or hour
- instance charge are present such as:
  - running the instance
  - storage
  - commercial software
- storage is of 2 types
  - on host storage - where the instance runs
  - EBS Elastic Block Store - network storage made available for the instance

ECS has **few** states

- **running**
- **stopped**
- **terminated** - one time action, that deletes the allocated storage



AMAZON MACHINE IMAGE (AMI)

- **is an image of an EC2 instance**
- is used to create an EC2 instance
- can be created from an EC2 instance
- contains attached permissions that control which accounts can or can't use the AMI
- can be public
- can be private (owner)
  - explicit permissions can be granted from the owner to other users
- contains the root volume (partition) - the driver that boots the OS
- can contains other volumes (other drivers)
- contains block device mapping
  - a configuration which link the volumes that the AMI has, and how they are linked withing the OS

Windows **Linux**

- a configuration which link the volumes that the AMI has, and how they are linked withing the OS

Windows

- **remote desktop control port 3389**

Linux

- **SHH protocol port 22**

What Permissions options does an AMI have?
Public access, Owner Only, Specific AWS accounts

Virtualization 101

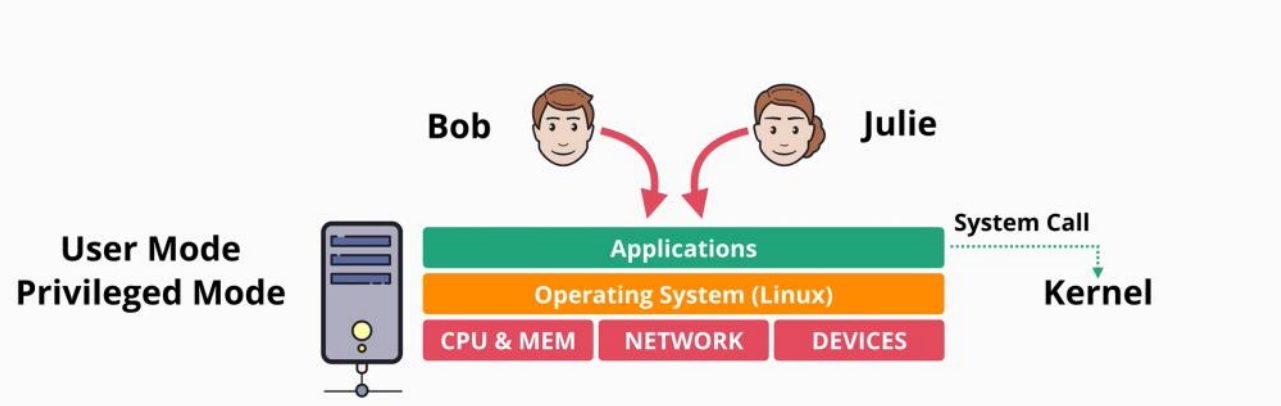
Wednesday, July 21, 2021 12:00 PM

EC2 provides virtualization as a service - IAAS

Virtualization is the process of running more than one Operating System on a piece of physical hardware

In traditional OS:

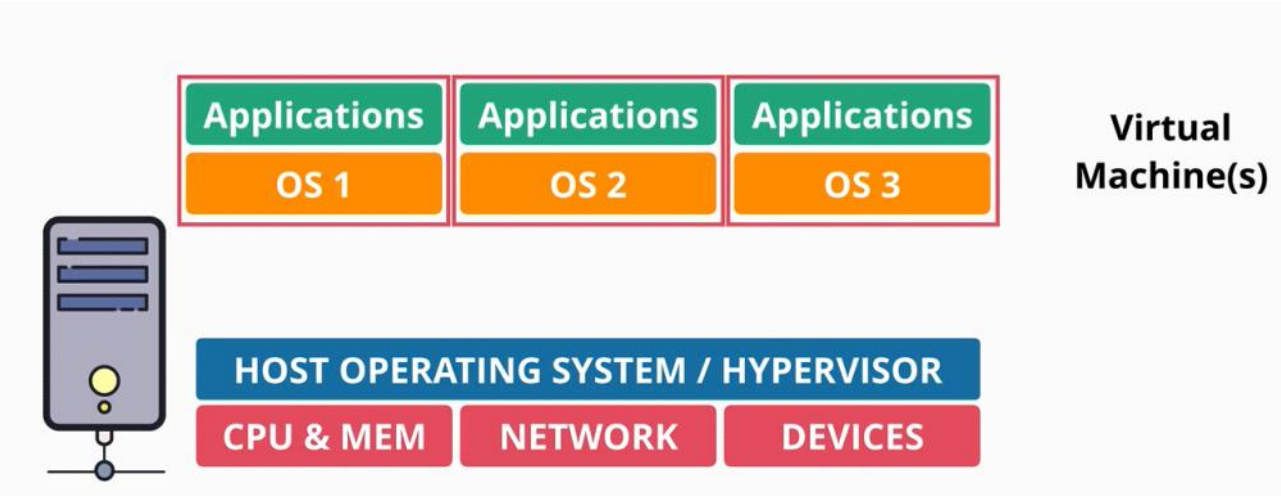
- the **kernel** from the OS is the only peace of software that is able to interact with the hardware
  - also, it runs in **privileged mode**
- on top of the OS are **application** which are running in **user mode**
  - they cannot interact with the hardware, they need to go through the OS



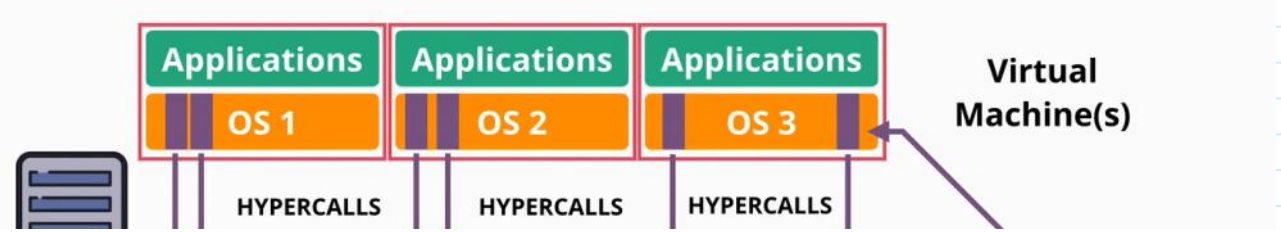
With **virtualization:**

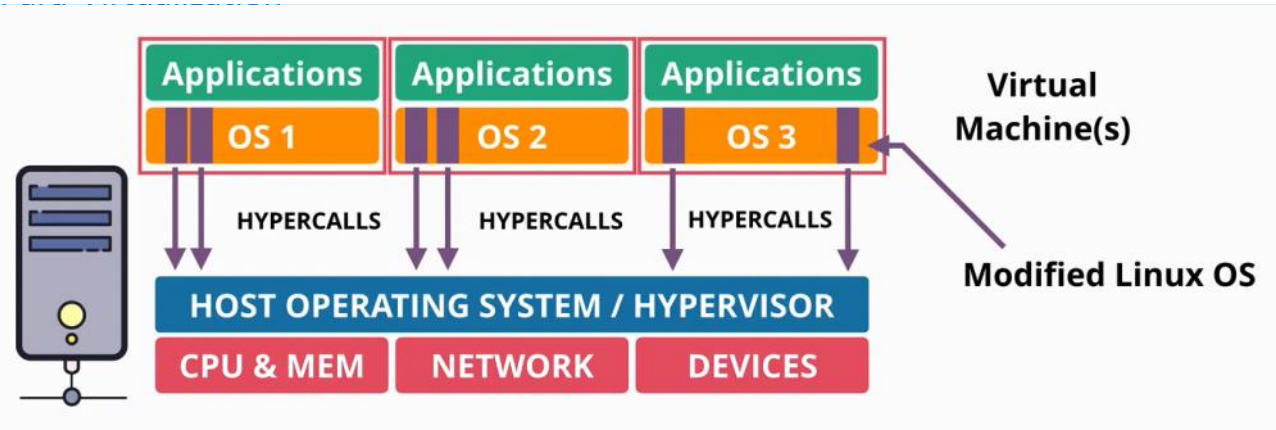
- a single piece of hardware running multiple Operating Systems
  - each OS is separate
  - each OS runs its own applications

Virtual Machines



Para-Virtualization

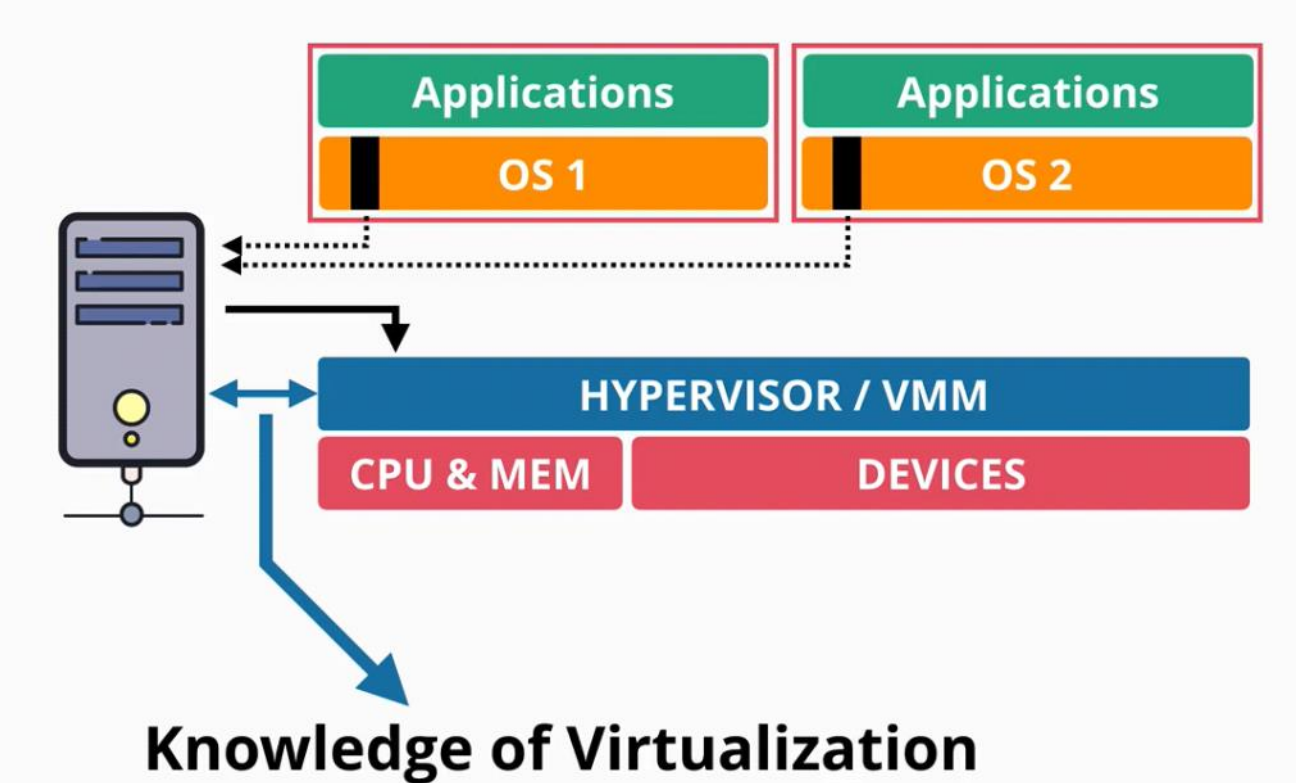




The guest OS makes calls to the hypervisor, not to the hardware

Hardware Assisted Virtualization

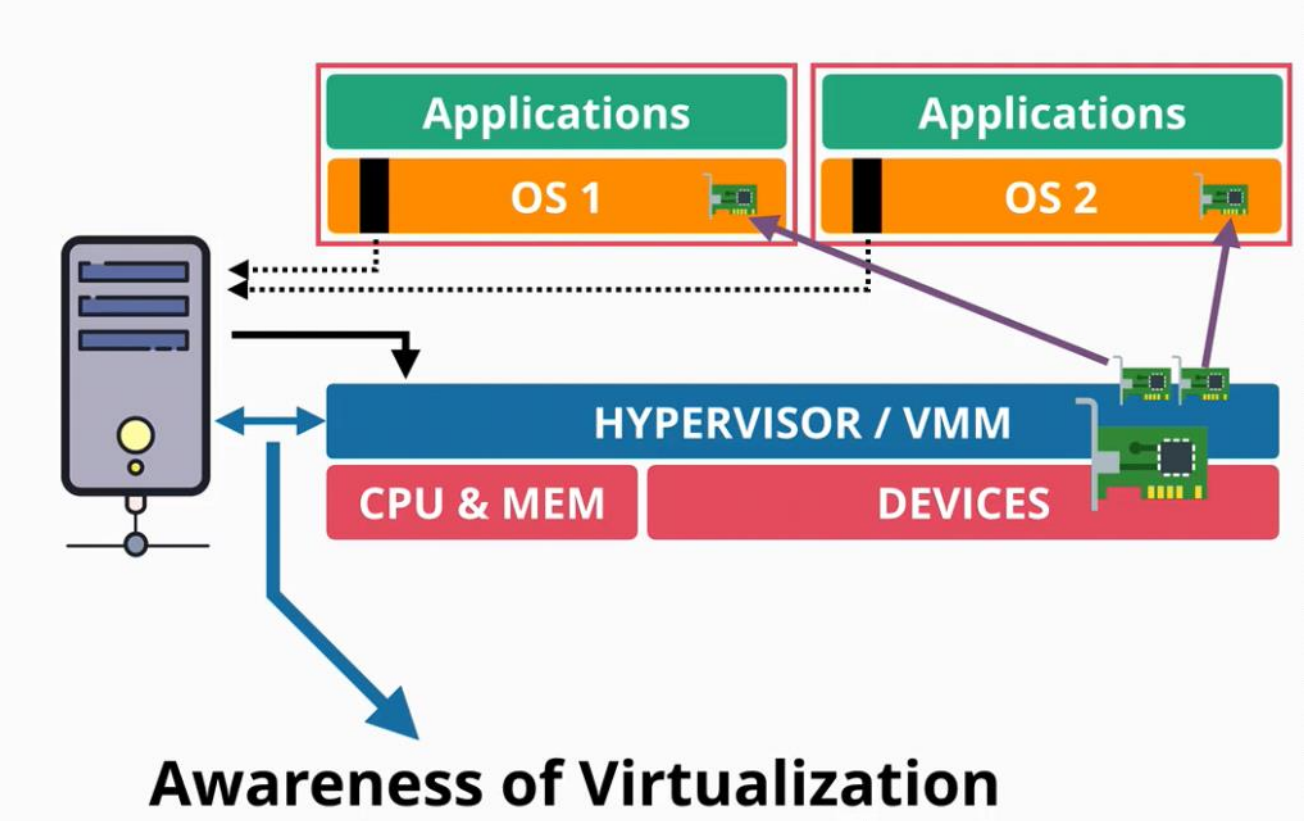
- the hardware became virtualization aware
  - the CPU is aware that it's performing virtualization
  - when guest OS attempt to run any privileged instructions they're trapped by the CPU which knows to expect them from the guest OS
  - the guest OS still believe they are running directly on the hardware



Even this was a big improvement, for IO tasks, there are still obstacles to run smoothly as there is always software getting in the way of the guest OS, which impacts performance and consumes a lot of CPU cycles on the host OS.

Single Route IO Virtualization (SR-IOV)

SR-IOV allows any add on card (like networking card) to present it self as several mini cards. Because of this, there is no translation required by the hypervisor. The guest OS can directly use the its card whenever it wants.
The physical card that supports this feature, handles the process end to end, making sure that when the guest OS use their logical mini network cards that they have physical access to the physical network connection when required.



In EC2 this feature is called **Enhanced Networking**, offering

- faster speed
- lower latency
- lower latency at high loads

EC2 Architecture and Resilience

Wednesday, July 21, 2021 12:47 PM

EC2 instances are **virtual machines** (OS + Resources). It is the default compute service in AWS

- **EC2** is great when you have a traditional OS and Application Compute need. SO if you have an application that requires to run on a certain OS with a certain runtime and configuration.
- It is also great for long running compute needs - **to run 24/7, 365**
- They are also perfect for applications or services that need burst requirements or steady-state requirements
- Is great to use EC2 if you have a monolithic application - that uses a stack, database, middleware or other run-time based components, and is needs a Traditional OS
- EC2 is very useful for application workloads or disaster recovery
  - they run on **EC2 hosts**
    - are physical hardware that are managed by AWS
    - these hosts can be
      1. **Shared Hosts**
         - these are shared between different AWS customers
         - even resources are shared, each customer is isolated from other customers
           - there is no visibility, interaction between customers
         - you do not get ownership of the hardware
         - you pay for the individual instances based on how long you run them for and what resources they have allocated
         - are the default option
      2. **Dedicated Hosts**
         - you are paying for the entire host not for the instances that run on it
         - you do no share it with other customers
  - is an **Availability Zone resilient service**
    - if the AZ fails, the instance fails
    - the service is very reliant on AZ, as the host and most of its dependencies need to be in the same AZ
  - have some local hardware:
    - CPU
    - memory
    - local storage - instance store - is temporary
      - is not portable
    - storage networking
      - can make use of remote storage
      - can connect to the Elastic Block Store, which is known as EBS
        - EBS also runs inside a AZ
        - they cannot be accessed crossed zone
        - allows the creation and allocation of volumes - persistent storage
    - data networking
      - when instances are provisioned into a specific subnet, within a VPC, a Primary Elastic Network Interface is provisioned in a subnet (which are also AZ resilient) which maps to the physical hardware on the EC2 host
      - instances can have multiple network interfaces even in different subnets as long as **they are in the same AZ**
  - **instances** run on a specific host, and if they are restarted they will stay on the same host
    - they stay on a host until two things happened:
      1. the host fails or is taken down for maintenance by AWS
      2. if an instance is **stopped** and the **started**

        - **if any of this two conditions happens, then the instance will be moved to another host, but on the same Availability Zone**
- instances cannot natively move between AZ
    - there are ways to do a migration but on the basics, it means that a copy of the instance is moved to another AZ
- **you can never** connect an instance to EBS or network interfaces in **another AZ**
    - you cannot cross AZ with EBS or EBS volumes
- instances running on a host, share the resources of that host
    - instances of different sizes can share a host but generally instances of the same type and generation will occupy the same host

Instance Types

Wednesday, July 21, 2021 1:56 PM

Instance types will influence different things:

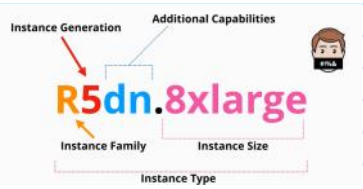
- the raw amount of resource that you get for that instance: virtual CPU, memory, local storage capacity, and the type of the storage
  - resource ratios
  - the pay/minute in influenced by the raw amount of resources
- influences the amount of network bandwidth for storage and data networking capability you get
- influences the architecture of the hardware that instances run on and the vendor (ARM, x86)
- influences any additional features and capabilities

EC2 are grouped in 5 big categories:

1. **General Purpose** - should always be the starting point
   - designed for steady work loads
   - have even ratios for resources
   - should be the first choice, and you should move away from that if there is a specific workload requirement
2. **Compute Optimized** - designed for media processing, high performance computing, scientific modeling, gaming and machine learning
   - they provide access to the latest high performance CPU
   - offer a ratio where more CPU is offered for the memory
3. **Memory Optimized -** is the inverse of the **compute optimized**
   - offers large memory allocations for a given CPU amount
   - it is ideal for application which need to work with large in memory data sets, memory cashing, etc
4. **Accelerated Computing** - additional capabilities come into play
   - dedicated CPUS for high scale parallel processing
   - custom programmable hardware such as FPGAs
5. **Storage Optimized** - they provide large amounts of super-fast local storage
   - designed for high sequential transfer rates
   - designed to provide massive amounts of IO operation per second
   - is great for application with serios demands on sequential and IO: databases, warehousing, elastic search, analytics workloads

R5dn.8xlarge - is a name of a type of instance

- **R** - is the instance family
- **5** - is the generation (always use the last generation)
- **8xlarge** - the instance size
- **dn** - additional capabilities (might exist in the name or not)



EC2 Instance Types

https://learn.cantrill.io adriancantrill

Categories	Type	Details / Notes
General Purpose	A1, M6g	Graviton (A1) Graviton 2 (M6g) ARM based processors. Efficient.
	T3, T3a	Burst Pool - Cheaper assuming nominal low levels of usage, with occasional Peaks.
	M5, M5a, M5n	Steady state workload alternative to T3/3a - Intel / AMD Architecture
Compute Optimized	C5, C5n	Media encoding, Scientific Modelling, Gaming Servers, General Machine learning
Memory Optimized	R5, R5a	Real time analytics, in-memory caches, certain DB applications (in-memory operations)
	X1, X1e	Large scale in-memory applications .. lowest $ per GiB memory in AWS
	High Memory (u-Xtb1)	Highest memory of all AWS instances
	z1d	Large memory and CPU - with directly attached NVMe
Accelerated Computing	P3	GPU instances (Tesla v100 GPUs) - parallel processing & machine learning
	G4	GPU Instances (NVIDIA T4 Tensor) - Machine learning inference and Graphics Intensive
	F1	Field Programmable Gate Arrays (FPGA) - Genomics, Financial Analysis, Big Data
	Inf1	Machine Learning - recommendation, forecasting, analysis, voice, conversation
Storage Optimized	I3/I3en	Local high performance SSD (NVMe) - NoSQL Databases, warehousing, analytics
	D2	Dense Storage (HDD) - data warehousing, HADOOP, Distributed File Systems, Data processing - lowest price disk throughput
	H1	High Throughput, balance CPU/Mem. HDFS, MAPR-FS, File systems, Apache Kafka, Big data

Storage Refresher

Wednesday, July 21, 2021 4:15 PM

Key Storage Terms

- **Direct** (local) attached Storage
  - which are physical disks connected directly to a device
  - for EC2 they are directly connected to the EC2 host and is called **Instance Store**
  - it is super fast as it is attached directly to the hardware
  - if the disk fails the storage can be lost
  - if the hardware fails the storage can be lost
  - if the instance moves between hosts the storage can be lost
- **Network** attached Storage
  - here is where volumes are created and attached to a device over the network
  - in AWS it uses a product called Elastic Block Store (EBS)
  - it is highly resilient
  - it is separate from the instance hardware
    - can survive issues which impact the EC2 host
- **Ephemeral** Storage
  - temporary storage - storage that does not exist long term
  - cannot rely on to be persistent
  - an example is: instance storage
- **Persistent** Storage
  - storage which exists as its own thing
  - it lives longer than the device that is attached to
  - an example is: network attached storage delivered by EBS

3 Main Categories of Storage Available

Describe how the storage is presented - to you or to a server - and what it can be used for

1. **Block Storage**
   - you can create a volume (eg. inside EBS)
   - a block storage has a number of addressable blocks
   - the block can have a huge or small number of blocks depending on the size of the volume
   - **the blocks are presented logically as a volume, or a hard drive**
   - they come in the form of **spinning hard disks or SSD** - physical media or delivered as a **logical volume** which is itself backed by different types of physical storage
     - network attached storage systems provide block storage
   - has no inbuild structure - it is a collection of uniquely addressable blocks
     - is up to the OS to create the file system and to mount that file system
   - you **can boot** from a block storage, and can save files to it
2. **File Storage**
   - it is provided by a file server
   - you can access, request, traverse, search a file storage
   - you cannot boot from a file storage
   - can be mounted
3. **Object Storage**
   - used to store objects
   - there is no structure is a flat collections of objects
   - an object can be anything (key:value pair)
   - not bootable
   - not mountable
   - it is scalable - can be accessed by millions of people simultaneously

Storage Performance

IO (Block) Size	Input-Output operations per second - IOPS	Throughput (amount of data transferred at a given second - MB/second)
the size of the blocks of data that can be written on the disk - it can range from pretty small sizes (KB) to large ones (MB)	measures the IO operations the storage system can support in a second. How many reads/writes the storage system can accommodate in a second	is the rate of data a storage system can store on a particular piece of storage generally expressed in MB/second

Elastic Block Store (EBS) Service Architecture

Sunday, July 25, 2021 12:26 PM

EBS is a service that provides block storage (storage that can be addressed using block IDs).

- raw disk allocations - volume which can be encrypted using KMS
- volumes when attached to EC2 instances they are seen as raw storage and can be used to create a file system in top of it (ext3, ext3, XFS and more)
- they appear as any other storage device to an EC2 storage
- storage **is provisioned in one AZ** (is separate and isolated in one AZ)
    - they can be backed up to an S3 bucket in a form of a snapshot **which are regional resilient**
        - very useful to migrate data between availability zones
        - can be **replicated to other regions**
- they are created and attached to an EC2 instance **over a storage network**
    - it can be attached to multiple EC2 instances at the same time, using a featured **called multi attached**
    - needs to be managed to prevent over-writing at the same time from different instances
- they can be de-attached from one instance, and the re-attached to another
    - are not linked to the EC2 instance life cycle of one instance
- it is persistent until **the volume is deleted**
- **can provision data based on different physical storage types** (SSD based, high performance SSD and volumes based on HDD)
    - can also provision different sizes of volumes, different performance
- they are billed on GB/month metric, and in some cases on performance characteristics
- **volumes cannot be attached to instances from other AZ!**

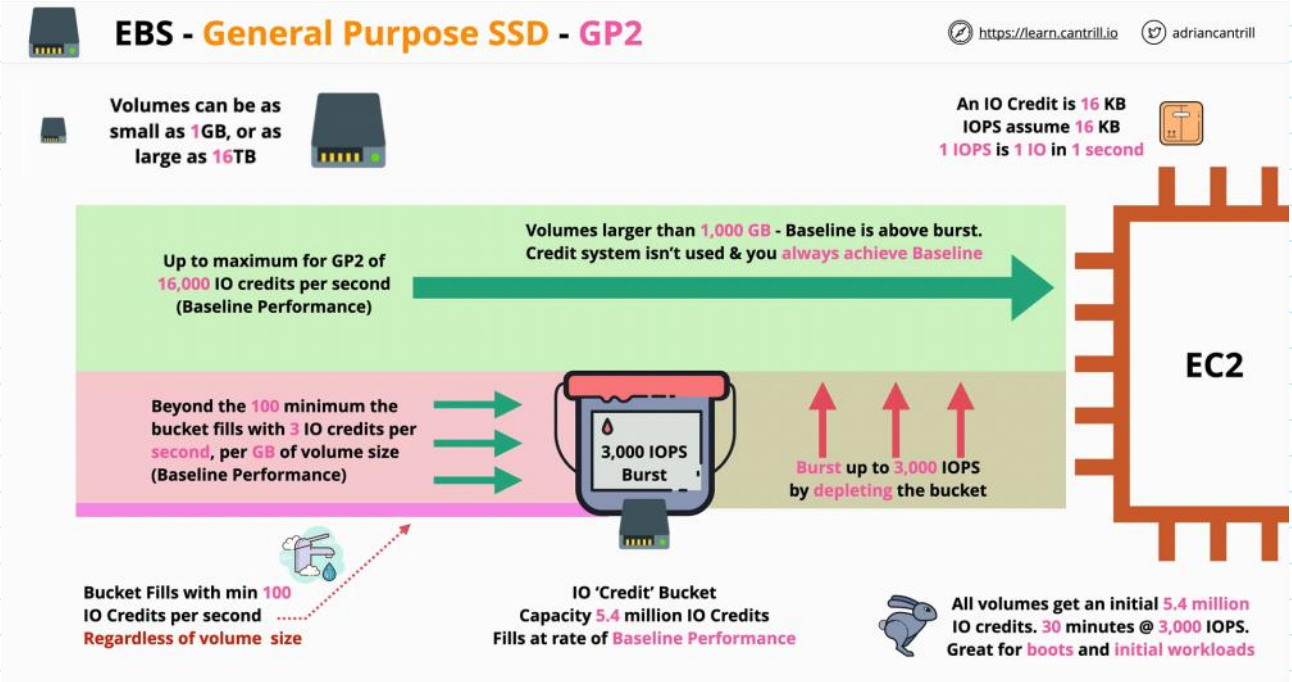
EBS Volume Types - General Purpose

Sunday, July 25, 2021 12:47 PM

EBS Volume Types

General Purpose SSD - GP2

- is the default general purpose SSD based storage
- is a high performance storage for a low price
- has a throughput of 250 MB/second
  When is created:
  - minimum storage is **1GB** and the maximum storage is **16 GB**
  - created with an IO credit allocation (16 KB data)
    - has a capacity of 5.4 million IO credits and it fills at the baseline performance rate of the volume
    - if you consume more IO credits than the rate at which the bucket is filling, then you are using up the resources of the bucket
    - the maximum IO/second is 16,000



  - is **great for boot volumes, low latency interactive apps** and for **dev and test environments**
    - it is currently the default

General Purpose SSD - GP3

- is also SSD based but it **removes the credit bucket architecture of GPS2**
- starts with a standard of 3000 IOPS (3,016 KB operation/second)
  - can transfer 125 MB/second
  - is standard regardless of the volume size - 1200MB/second
- can safely swap between GP2 to GP3 at any point
  - ★ just remember that over 3000/IOPS the performance does not get added automatically
  - ★ with GP3 the extra IOPS need to be added manually which come at an extra cost

EBS Volume Types - Provisioned IOPS

Sunday, July 25, 2021 1:15 PM

There are **3 types of provisioned IOPS SSD:**

- what they have in common is that they can be configured independently of the size of the volume
- they are designed for super high performance situations
  - consistent low latency and jitter
  - ★ IO1 and IO2:
    - 64,000 IOPS/volume
    - 1000 MB/s throughput
    - volume size range from 4GB to 16TB
  - ★ IO2 block express
    - 256,000 IOPs/volume
    - 4,000 MB/s throughput
    - volume size range from 4GB to 64TB
- there is a maximum performance which can be achieved between the EBS service and a single EC2 instance
  - this can be influenced by:
    - the type of volume
    - the type of the instance
    - the size of the instance
- **should be used for anything that needs really low latency, or sub millisecond latency, consistent latency and high levels of performance**
  - **when you need smaller volumes but super high performance**
  -

IO1

- **50 IOPS/GB** of volume size
- per instance performance 260,000 IOPS/instance and a throughput 7,500 MB/s
  - ★ you need 4 volumes to achieve this per instance limit

IO2

- **500 IOPS/GB** of volume size
- per instance performance 160,000 IOPS/instance and throughput of 4,750MB/s

IO2 - Block Express

- **1000 IOPS/GB** of volume size
- per instance performance 260,000 IOPs/instance and throughput 7,500 MB/s

EBS Volume Types - HDD-Based

Sunday, July 25, 2021 1:31 PM

HDD based means that the storage has moving parts such a spinning platers (disks)

- therefore they are slower

There are two types of HDD storage within EBS

Throughput Optimized - ST1	Cold HDD - SC1
fast but not agile	is cold
is cheaper than SDD volumes which makes it ideal for any larger volumes of data	is even cheaper but comes with significant trade-offs - is the lowest cost EBS storage available
is designed for data which is sequentially accessed - is not great at random access - designed for data that needs to be written or read in a sequential way - where economy is more important that IOPS and extreme performance	
volumes ca be from 125GB to 16TB in size	volumes ca be from 125GB to 16TB in size
maximum of 500 IOPS (on 1MB block) - maximum 500MB/s throughput	maximum of 250 IOPS (on 1MB block) - maximum 250MB/s throughput
they work with a credit bucket with MB/S - base line performance 40MB/S for every 1TB of volume size - burst maximum of 250 MB/S for every 1TB volume size	they work with a credit bucket with MB/S - baseline performance 12MB/S for every 1 TB of volume size - burst maximum of 80MB/S for every 1TB volume size
is designed: - for when cost is a concern but you need frequent access storage for throughput intensive sequential workloads - big data, data warehouses and log processing	is designed: - for infrequent workloads - if it geared for maximum economy when you just want to store lots of data and do not care about performance - archives, colder data, anything that requires less than a few loads or scan/day

Instance Store Volumes - Architecture

Sunday, July 25, 2021 1:58 PM

Instance Store Volumes provide block storage devices.

- raw volumes which can be attached to an instance presented to the OS on that instance and used as the basis for a file system which can then in turn be used by application
- they are local, **not over the network**
  - they are physically connected to the EC2 host
  - they are isolated to that one particular host
- they offer the highest storage performance available on AWS
  - more than IBS can provide
- included in the price of any instance
  - difference instance types come with different selections of instance store volumes
- they need to be attached at launch time
  - they cannot be attached afterwards
- each instance can have a collection of volumes which are backed by physical devices on the EC2 host
- if an instance moves between hosts, then any data was present on the instance store volume is lost
  - if they are moved to a new host, the instances are given a new blank ephemeral volumes
    - the data on the previous volume is lost, as they are whipped before they are re-assigned
  - instances can move hosts for many reasons
    - when they are stopped and started
    - if the instance is resized
    - if the host is in maintenance
    - changing an instance type
    - if the volume fails
- they are temporary data holders
  - should not be used for persistent data
- some instance types do not support store volumes
- as the instance increases in time, it has a large number of instance store volumes
- **advantages**
  - performance
  - high level of throughput
  - more IOPS
- instance store volume **can be added only when the instance is launched**

Cloud Formation

Thursday, June 17, 2021 12:30 PM

Cloud formation is a tool that allows you to create, update and delete infrastructure in a AWS account.
Instead of deleting and creating resources manually, a template can be created.
The **CloudFormation** automates infrastructure

A cloud formation template is written in **YAML** or **JSON**.

Templates have:

- a list of **resources**
  - **the only mandatory part**
- description
  - lets the author to add a description
  - it is usually used to describe what the template does, its resources, cost of the template etc.
  - the description needs to follow directly after the AWS template format version!
- metadata
  - controls how different components are presented through the console UI
  - force how the UI presents the template
- parameters
  - fields that prompt the user for more information
- mappings
  - allows to create lookup tables
- conditions
  - actions that occur only if a condition is bet
- outputs
  - once the template is finished it will present outputs of what had been created, updated or deleted

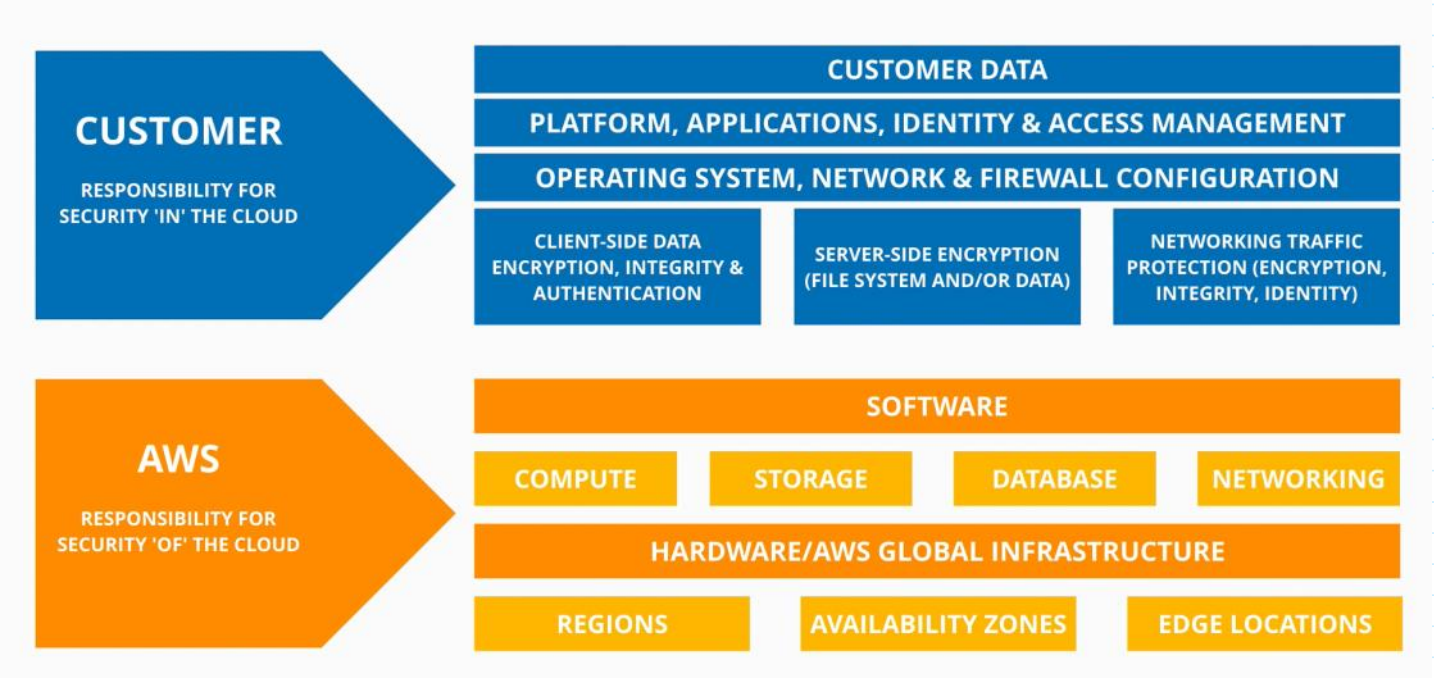
A template contains a logical stack of resources. When a template is executed, the Cloud formation creates a corresponding physical resource.

A CloudFormation Physical resource is a physical resource by creating a CloudFormation stack

Shared Responsibility Model

Tuesday, June 22, 2021 1:59 PM

Shared Responsibility Model



High-Availability vs Fault-Tolerance vs Disaster Recovery

Tuesday, June 22, 2021 3:06 PM

High-Availability (HA)

Maximize uptime

- aims to ensure an **agreed level of operational performance** usually uptime, for a higher than normal period
- -designed to be online as much as possible
- is not about the user experience
- is about maximizing the online time as much as possible
- this required backup (redundant) system to replace of a outbreak system
    - this action has a small outbreak time, and this is okey

Fault-Tolerance (FT)

Operating through failure

- is the property that enables a system to continue operating properly in the event of the failure of some or more components
- if a system has faults, it should continue properly until the faults are fixed
- system are designed to work through failure without disruption

Disaster Recovery (DR)

Keep the crucial and non-replaceable parts of the system safe

- **what we do when high-availability and fault-tolerance DO NO WORK**
- a set of policies, tools and procedures to enable the recovery or the continuation of vital technology infrastructure and systems, following a natural or human-induced disaster
- plan for an action plan when disaster occur
- is a multiple steps processes:
    - what happens before (pre-planning)
        - backup premises
        - virtual backup systems
        - offsite backup storage
        - DR testing
    - what happens after (disaster recovery process)

DNS

Tuesday, June 22, 2021 3:40 PM

DNS is a **discovery service**, it helps users to discover system on the internet.
DNS is resilient, distributed and scalable.

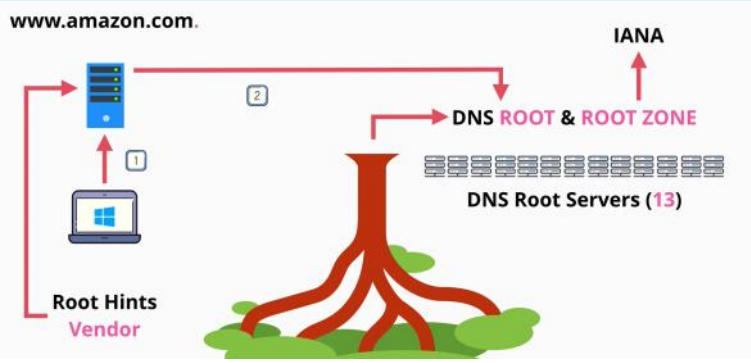
- it translates information machine into human and vice-versa
- translate a domain name to an IP address
- is huge, resilient and hard to distribute
- is a database, distributed and global
- each server has a ZONE file, that links a domain name to the correct IP address
    - that zone file is located anywhere on potentially one or two of millions of DNS name services.
    - the DNS resolver is sitting either on the internet router, or the provider

- **DNS Client** => your laptop, phone, tablet, PC
- **Resolver** => software on your device, or a server which queries DNS on your behalf
- **Zone** => A part of the DNS database (e.g. **amazon.com**)
- **Zonefile** => physical database for a zone
- **Nameserver** => where zonefiles are hosted

DNS is distributed, the data is stored in zone files, and the zone files are stored on name servers globally. The DNS needs a start point.

DNS flow: The DNS resolver needs to locate the correct name server for a given zone, query that name server and retrieve the information it needs, and then pass it back to the DNS client.

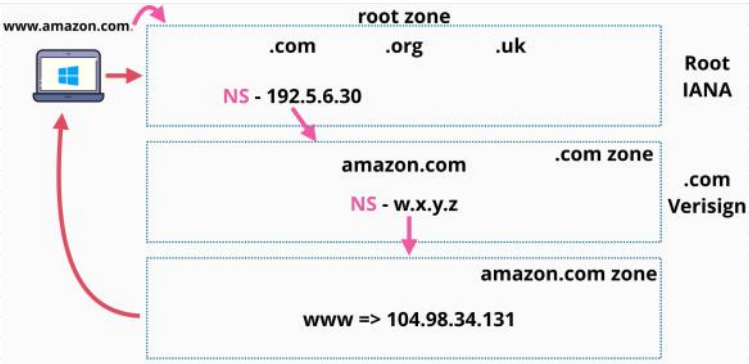
The DNS is structured as a tree. The DNS root (domain name), is stored on 13 special servicer known as **DNS root servers**. This servers are the entry point.



1. The DNS client asks a DNS resolver of given DNS name
2. using the root hints file, the DNS resolver communicated with one or more root servers to access the root zone and begin the process of finding the IP address

A local system (laptop, desktop, etc.). The local system will use a **DNS resolver server** which is located on the local router or internet provider. The vendor of the OS (Microsoft for Windows, OS maintaining group for Linux, etc.) supply the root hints file. The **root hints file** is a pointer to the DNS root servers.

Top level domains are in the root zone (.com, .org, .uk, etc.). This root zone database is managed the IANA. These root zones point to other servers that manages country codes zones.



The image above is how the chain of trust works in DNS.

- **Root Hints** => config points at the root servers IPs and addresses
- **Root Server** => hosts the DNS root zone
- **Root Zone** => points at TLD authoritative servers
- **gTLD** => generic top level domain (.com .org)
- **ccTLD** => country-code top level domain (.uk .eu etc)

Route 53

Wednesday, June 23, 2021 2:53 PM

Route 53 provides to main services:

1. Register domains
2. Host zone files and manage nameservers

Route53 is a global service with a single database, and it is globally resilient.

Register domains

Route 53 allows the registration of domains, and to do that it has relationships with major domain registers.
When registering a domain Route 53 follows the following steps:

1. registers a zone file (dora.org)
2. allocates a name service for that zone
3. puts the zone file in 4 name servers
4. communicates to .org registry to add the new domain into the zone file for the.org top level domain
    a. using name server records

Host zone

(DNS as a service)
This service allows hosting, creating and managing zone files.

Zone files are hosted on 4 managed name servers.
When a hosted zone is created , a number of server are allocated and linked to the hosted zone.
A hosted zone can be public (accessible on the public internet) or private which means that they are linked to one or more VPCs.

DNS Record Types

Wednesday, June 23, 2021 4:02 PM

Which type of organisation maintains the zones for a TLD

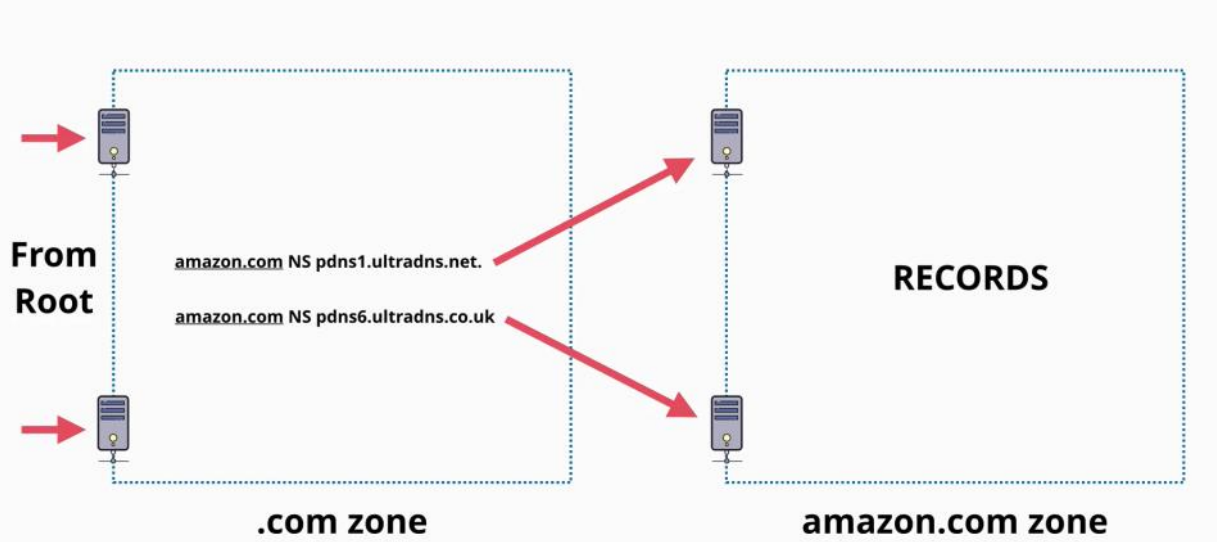
Registry

Which type of organisation has relationships with the .org TLD zone manager allowing domain registration?

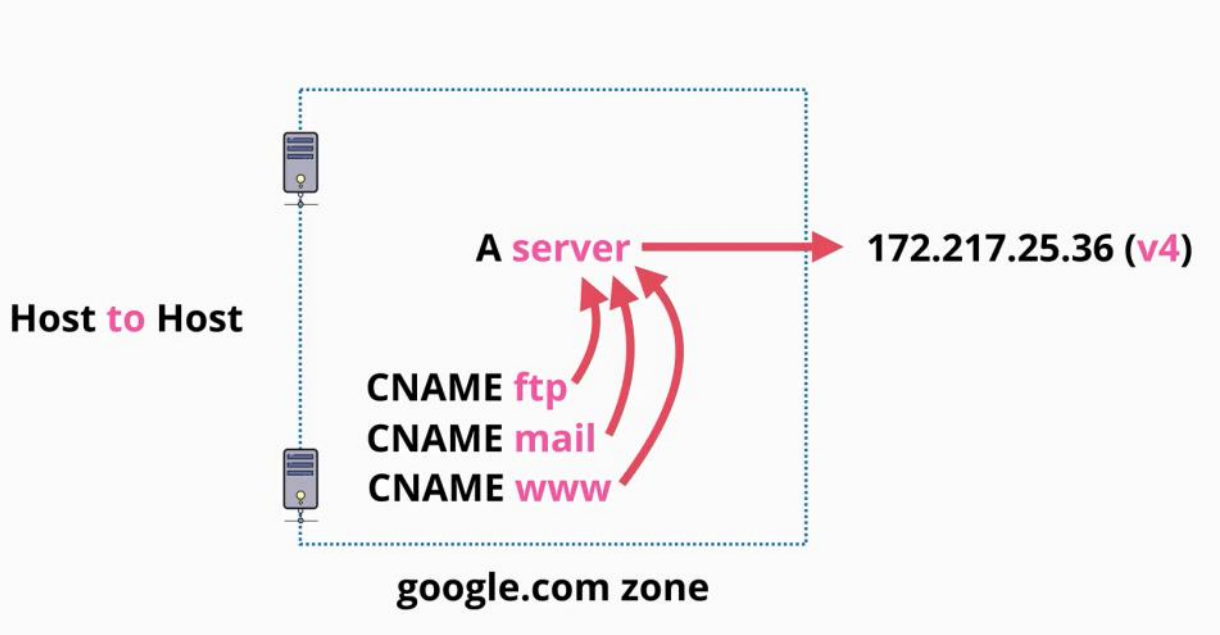
Registrar

Nameserver (NS)

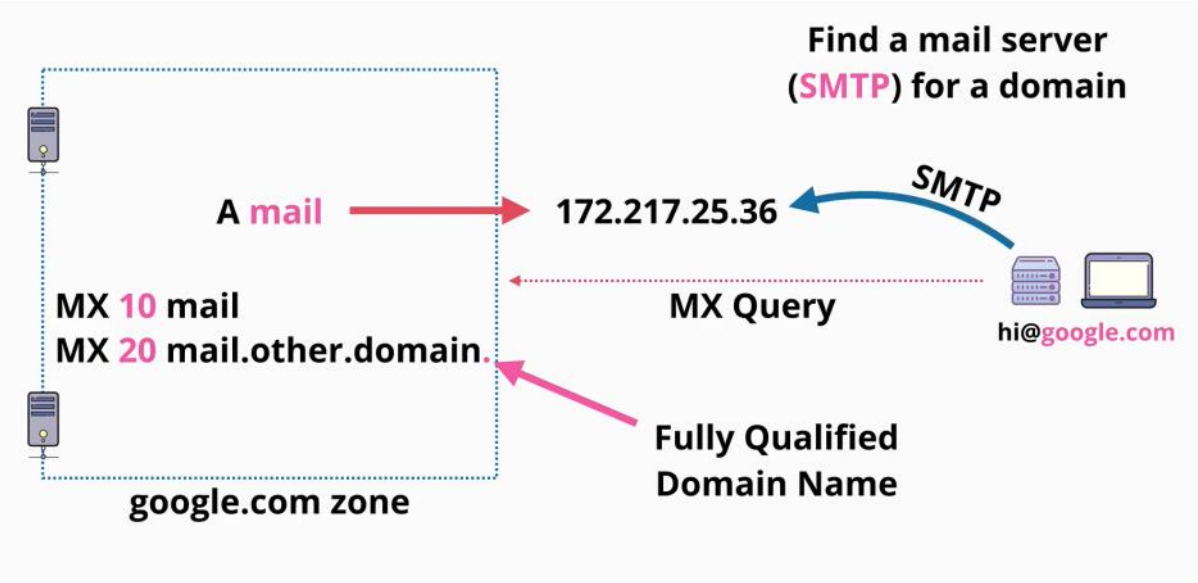
Nameservers are record types, which allow delegation to occur in DNS



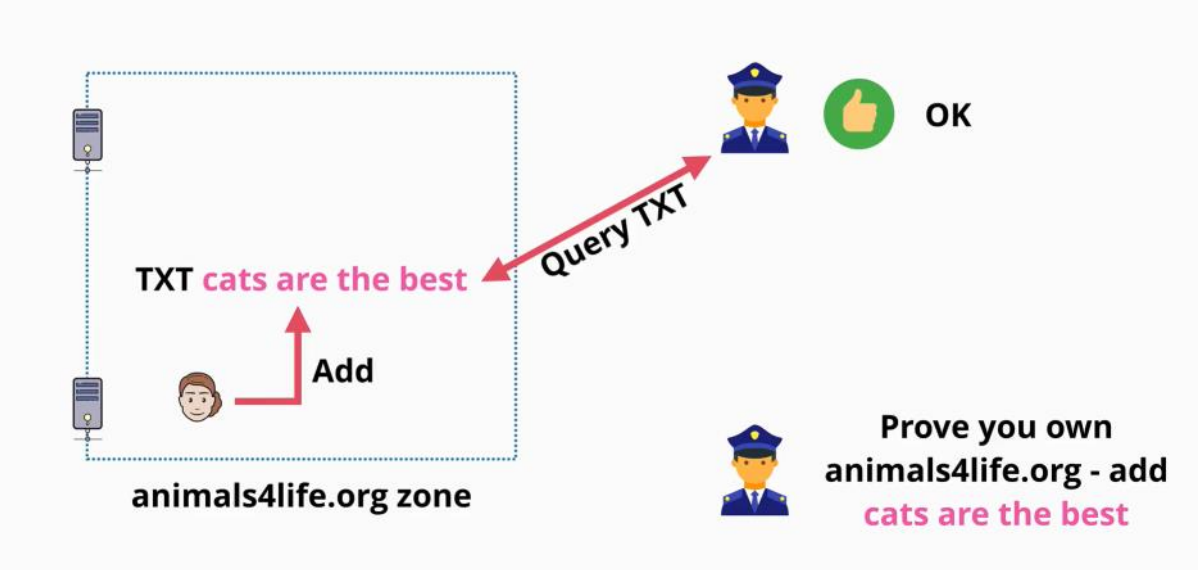
NS record - delegates control of .org to the .org registery
A record - > IPv4 addresses
AAAA record -> IPv6 addresses
CNAME record -> Cnames cannot point directly at an IP address, only at other names.



MX Records - are used as apart of a process (mail)



TXT Records - add arbitrary text to a domain - is used to prove ownership



DNS TTL - Time to live

ATTL value is set to DNS record, is a numeric value in seconds.
Walking the tree takes time, to talk with the root, and all the following levels to get the result needed. It is a lengthy processes.

The TTL specifies for how long the response values can be cached. This is set by the administrator. Once a request had been responded to, and the result is cached on the resolver server, other clients do not need to wait too much for a response.

Authoritative and non-authoritative responses are the same, so in this case TTL matters when things change so that responses are not delayed (if TTL is high).
When things change, the TTL must be changed to a low value, or have a low value of TTL all the time.

Exam guide

Thursday, June 3, 2021 12:25 PM

aws training and certification

AWS Certified Developer – Associate (DVA-C01) Exam Guide

Introduction

This AWS Certified Developer - Associate (DVA-C01) examination is intended for individuals who perform a developer role.

It validates an examinee's ability to:

- Demonstrate an understanding of core AWS services, uses, and basic AWS architecture best practices.
- Demonstrate proficiency in developing, deploying, and debugging cloud-based applications using AWS.

Recommended AWS Knowledge

- 1 or more years of hands-on experience developing and maintaining an AWS based application
- In-depth knowledge of at least one high-level programming language
- Understanding of core AWS services, uses, and basic AWS architecture best practices
- Proficiency in developing, deploying, and debugging cloud-based applications using AWS
- Ability to use the AWS service APIs, AWS CLI, and SDKs to write applications
- Ability to identify key features of AWS services
- Understanding of the AWS shared responsibility model
- Understanding of application lifecycle management
- Ability to use a CI/CD pipeline to deploy applications on AWS
- Ability to use or interact with AWS services
- Ability to apply a basic understanding of cloud-native applications to write code
- Ability to write code using AWS security best practices (e.g., not using secret and access keys in the code, instead using IAM roles)
- Ability to author, maintain, and debug code modules on AWS
- Proficiency writing code for serverless applications
- Understanding of the use of containers in the development process

Exam Content

Response Types

There are two types of questions on the examination:

- **Multiple choice:** Has one correct response and three incorrect responses (distractors).
- **Multiple response:** Has two or more correct responses out of five or more options.

Select one or more responses that best complete the statement or answer the question. Distractors, or incorrect answers, are response options that an examinee with incomplete knowledge or skill would likely choose. However, they are generally plausible responses that fit in the content area defined by the test objective.

Unanswered questions are scored as incorrect; there is no penalty for guessing.

Unscored Content
Your examination may include non-scored questions that are placed on the test to gather statistical information. These questions are not identified on the form, and do not affect your score.

Exam Results
The AWS Certified Developer - Associate (DVA-C01) examination is a pass or fail exam. The examination is scored against a minimum standard established by AWS professionals guided by certification industry best practices and guidelines.

Your results for the examination are reported as a score from 100–1,000, with a minimum passing score of 720. Your score shows how you performed on the examination as a whole and whether or not you passed. Scaled scoring models are used to equate scores across multiple exam forms that may have slightly different difficulty levels.

Your score report contains a table of classifications of your performance at each section level. This information is designed to provide general feedback concerning your examination performance. The examination uses a compensatory scoring model, which means that you do not need to "pass" the individual sections, only the overall examination. Each section of the examination has a specific weighting, so some sections have more questions than others. The table contains general information, highlighting your strengths and weaknesses. Exercise caution when interpreting section-level feedback.

Content Outline
This exam guide includes weightings, test domains, and objectives only. It is not a comprehensive listing of the content on this examination. The table below lists the main content domains and their weightings.

Domain	% of Examination
Domain 1: Deployment	22%
Domain 2: Security	26%
Domain 3: Development with AWS Services	30%
Domain 4: Refactoring	10%
Domain 5: Monitoring and Troubleshooting	12%
TOTAL	**100%**

Domain 1: Deployment
- 1.1 Deploy written code in AWS using existing CI/CD pipelines, processes, and patterns
- 1.2 Deploy applications using Elastic Beanstalk
- 1.3 Prepare the application deployment package to be deployed to AWS
- 1.4 Deploy serverless applications

Domain 2: Security
- 2.1 Make authenticated calls to AWS services
- 2.2 Implement encryption using AWS services
- 2.3 Implement application authentication, and authorization

Domain 3: Development with AWS Services
- 3.1 Write code for serverless applications
- 3.2 Translate functional requirements into application design
- 3.3 Implement application design into application code
- 3.4 Write code that interacts with AWS services by using APIs, SDKs, and AWS CLI

Domain 4: Refactoring
- 4.1 Optimize application to best use AWS services and features
- 4.2 Migrate existing application code to run on AWS

Domain 5: Monitoring and Troubleshooting

5.1 Write code that can be monitored

5.2 Perform root cause analysis on faults found in testing or production