

The Way To The Cloud

By Omar Tamer



Cloud/DevOps Engineer

Table of Contents

• بسم الله الرحمن الرحيم

- Important Terms
- Before Cloud
 - Server Fundamentals
 - Difference Between PC and Servers
 - Servers Types
 - Managing Servers
 - VMware Fundamentals
 - The Problem before VMs
 - The Start of VMs
 - Type's of VMs
 - Network Fundamentals
 - IP Address
 - Dynamic Host Config Protocol(DHCP)
 - DNS
 - DNS Records
 - Proxy Service
 - Security Fundamentals
 - Storage Fundamentals
 - 1. Types of Storage:
 - 2. Networked Storage:
 - 3. Difference Between Back-to-Back and Switching:
 - Databases Fundamentals
- Cloud Foundation & Architect
 - Cloud Concepts
 - Why Cloud
 - Introduction to Cloud Computing
 - 1. Deployment Model
 - 2. Service Model
 - The History of AWS
 - Intro to AWS
 - AWS Cloud Adoption Framework "AWS CAF"
 - Architecting Fundamentals
 - Role of a Cloud Architect
 - General Design Principles
 - AWS Well Architected Framework
 - Best practices for building solutions on AWS
 - Trade Offs

- Loosely Coupled Components
- Disaster Recovery (DR) & DR Approaches in AWS
- AWS Economics and Billing
 - Fundamentals of pricing
 - Exploring AWS Billing and Cost Management
 - Technical Support Plans
- AWS Global Infrastructure
 - AWS Regions
 - Availability Zone
 - AWS Outposts
 - AWS Local Zones
 - AWS Wavelength
 - CloudFront
 - AWS Route 53
 - AWS Global Accelerator
 - Deep Dive AWS Global Infrastructure
 - AWS Regions & AZs Selection
 - Availability, Naming & Edge Locations
- AWS Security
 - AWS IAM
 - IAM Policy
 - IAM User
 - IAM Group
 - IAM Role
 - Security Services
- AWS Networking
 - 1. VPC And Subnets
 - 2. IP Addresses In AWS
 - 3. AWS Gateway's Service
 - 4. Security Layers
 - 5. VPC Connections
 - A) VPC Peering
 - B) VPC Endpoints
 - C) AWS PrivateLink
 - D)AWS VPN
 - E)AWS Direct Connect(DX)
 - G) Transit Gateway
- AWS Networking Deep Dive
 - Why use an Amazon VPC?
 - IP addressing in Amazon VPC
 - Internet gateway, NAT Gateway and NAT instance
 - Securing your System
 - How to connect to managed AWS Services
 - Connecting Amazon VPCs

- VPC Troubleshooting
- AWS Compute Services
 - Amazon EC2
 - EC2 Families & Types
 - Instance Life Cycle
 - Amazon Machine Images (AMIs)
 - EC2 Image Builder
 - AWS Key Pair
 - User Data Script
 - SSH to connect to a Linux EC2 instance
 - EC2 Purchasing/Launch Options
 - 1. On-Demand (most expensive)
 - 2. Reserved Instances (RIs)
 - 3. On-demand Capacity Reservation (no commitment)
 - 4. Spot Instance
 - 5. Saving plans
 - 6. Dedicated hosts
 - 7. Dedicated Instances
 - EC2 Instance Billing
 - EC2 Placement Groups
 - EC2 Status Checks
 - Relevant Pillars
 - Cost Optimization Pillars
 - Pillar One: Right Sizing
 - Pillar Two: Increase Elasticity
 - Pillar Three: Optimal Pricing Model
 - Pillar Four: Optimize Storage Compute
 - Security Pillar
 - Performance Efficiency pillar
- Containers on AWS
 - Amazon Elastic Kubernetes Service(EKS)
 - Amazon Elastic Containers Service(ECS)
 - Amazon Elastic Containers Registry(ECR)
- AWS Batch
- Amazon Load Balancer & ASG
 - 1. Auto Scaling Groups (ASG)
 - 2. Load Balancers
- AWS Serverless Computing
 - Introduction
 - Lambda
 - API Gateway
- Notification, Messaging and Application Integration in AWS
 - Amazon Simple Queue Service (SQS)
 - Amazon Simple Notification Service(SNS)

- AWS Storage Services
 - Amazon Elastic Block store(EBS)
 - Amazon Instance-Store
 - 1. Amazon S3
 - 1. Intro to Amazon S3
 - 2. Characteristics
 - 3. Access Management
 - 4. S3 Storage Class
 - 5. S3 Migration
 - Snow Family
 - 6. Hybrid Cloud with Storage Gateway
 - 7. S3 Pricing
 - 2. Amazon EFS
 - 3. Amazon FSx
 - Storage Service Compared
 - 4. Further Storage Services
 - AWS Database
 - 1. Relational Databases
 - Amazon RDS
 - Amazon Aurora
 - Characteristics
 - Amazon Aurora Serverless
 - Characteristics
 - Amazon Redshift
 - 2. Non-Relational Databases
 - Amazon DynamoDB
 - DynamoDB Features
 - Amazon DocumentDB
 - Amazon Neptune
 - 3. Further Databases Services
 - 1. Amazon EMR
 - 2. Amazon Athena
 - 3. Amazon QuickSight
 - 4. AWS Glue
 - 5. Amazon Kinesis
 - 6. Databases Migration Services
 - AWS Monitoring
 - AWS CloudTrail (Auditing)
 - AWS Network Monitoring
 - What is Monitoring Tools?
 - Amazon CloudWatch
 - Further AWS Services
- Labs & Tasks
- Thanks To

Important Terms

ال Fault Tolerance : هي أن يكون عندي Redundant Component لي switch Mechanism من ال الى failure حصله عندي.

- مثال: لو عندي Disk في data وحصله impact دا كدا ال Fault و دا يحصل في مكان معين و ببغا اسمه Fault Domain يعني ال Failure دا على مستوى ايه؟ في الحالة دي في مستوى ال disk, طب لو فيه حالة server impact حصله server دا على مستوى ال server, طيب عشان الحق ال fault الى حصل دا يحتاج ببغا عندي حاجتين وهما ال Redundancy و Redundancy, ال Disk Mechanism وهو أن ببغا عندي نسخ من ال Disk أقدر Switch عليه في حالة حصل ال Fault دي و ال Hardware هو طريقة ال Smart enough Mechanism هي ال RAID, جزء أولى بيتحققلي ال Tolerance (الطريقة الى تخلص Server شغال حتى لو عندي impact Mechanisms .Mechanism هو Redundancy و Redundancy (Automatic

:Fault Tolerance و High Availability

- ال HA: هي أنه ممكن ببغا في Minimal Downtime ودا وقت الى Recover بشكل سريع, بي Backup system بشكل automatic وبـ switch لـ failure ميحصلش اي
- ال Fault tolerance: هي أنه عندي Zero Downtime على ال Server, يعني ان لو حتى حصل failure يقوم بـ component tolerant على شغال بدل interruption System

ال Replication: لما تشوها أفتكر ال Data, وهي Database عندي Mechanism ويعمل منها تكرار او نسخ بشكل Sync او Sync في مكان مختلف عن ال database بناعتي ال Main Database وقعت ببغا عندي database تانية تقوم مكناها بشكل sync نفس ال Data (في حالة ال sync Data مش هفقد أي Data ولكن في حالة ال sync ممكن فقد بعض Data بسبب أنه ملحق بـ Data جديد).

ال Single Point of Failure (SPF): وهو ممكـن يسبـبـ أن System كلـه يقعـ فيـ حالـةـ آنهـ Fail.

- ال Coupled Architecture: وهو Application كلـهـ فيـ حـتـهـ وـاحـدـهـ لوـ عملـتـ updateـ فيـ جـزـءـ هـيـأـثـرـ فيـ الـ باـقـيـ, بـبـغاـ
- مثال: بـبـغاـ الـ APPـ وـ الـ DBـ معـ بـعـضـ (Monolithic).

:Scaling

- ال Vertical Scaling: هو أني أزود ال Server Resources بـنـاعـتـ الـ CPUـ وـ الـ RAMـ وـ هـكـذاـ).
- ال Horizontal Scaling: هو أني أزود عدد ال Servers ويكون من نفس ال type يعني نفس CPU و RAM و هكذا.

ال Cluster: هي عبارة عن Group of Devices Connected with each other to do a specific function مجموعـةـ منـ Machineـ بـيـتعلـمـهاـ Controlـ وـ بـيـتعلـمـهاـ Taskـ معـيـنةـ اوـ بتـ runـ appـ معـيـنةـ وـ هـكـذاـ.

- ال LB: هو حل مشكلة ال Latency و Downtime عن طريق ال Horizontal scaling و أنه ينقل ال load لـ healthy servers فـاـ ساعـتهاـ هـيـحـقـ أنـ مشـ هـيـقـاـ فيـهـ, الـ LBـ لـيهـ algorithmـ بـبـوزـ بـبـهاـ Loadـ عـلـيـ
- ال HPC: هو عبارة عن مجموعة من servers الغرض منه أني بـ runـ appـ orـ codeـ بـيـشـتـغـلـ عـلـيـ كلـ serversـ دـيـ inـ parallelـ
- ال Provisioning: معناه أن عندي Equipment و بـخـلـيـهاـ Ready~to~useـ
- ال Emphemeral: ليها معنـينـ
- اولـهمـ لـوـ ليـهاـ عـلـاقـةـ بيـ Storageـ معـناـهاـ temporary storageـ اوـ volatileـ

- ثانی حاجه لو نحيت ال random networking معناها .
 - ال Durability : في مجال storage refer لي أقدر أعتمد على ال system عشان يحافظ على ال data بنسبة قد ايه.
 - ال Offload : معناها أنها تنقل ال Load من مكان معين لمكان مخصص او يقدر يستحمل ال Load الي جي عشان احسن ال Failure Application او أنفذ ال performance من Application.
 - على سبيل المثال: استعمل Redundant Hardware (أنقل) ال work بشكل ميلارش على client في حالة ال Fault tolerance.
 - في حاجتين بيعرفوا ال Products او Services بتاعتك وهم:
1. ال Customer يكون satisfied من وقت دخولة ال service لغاية لما يخرج to End to Logical Requirement تكون شغالة و end functionality.
2. ال Application Accepted Latency في ال Non - Functional Requirement .
- ال Testing/Quality Engineers .Performance و Security يعني لما service متاحة. وز بيركز على Available/ Down Service .
- ال Business analysis و Solution Architect و . وال اللي بيهموا بالأمور دي هما ال .Non - Functional Requirement بيركز على:
1. ال Latency
 2. ال High Availability
 3. ال Security
 4. ال Performance
- ال Pattern : هي أني عندي مجموعة من situations هتقابلك وليها Pattern معين تحله بيهها عشان تفادي المشاكل الي ممكن تحصل من ال Situations دي.
- ال Anti-Pattern : هي أني أحل المشكلة بدون معرف ال Root Cause ليهاو بدون معرف الجوانب كلها.
- ال Whitelisting : كل حاجه بتبقa allowed rules لغاية لما تحط .
- ال Blacklisting : كل حاجه بتبقa block rules لغاية لما تتعارض مع permit rule فا تأخذ .
-

Before Cloud

ما لا يسعك جهله قبل Cloud .

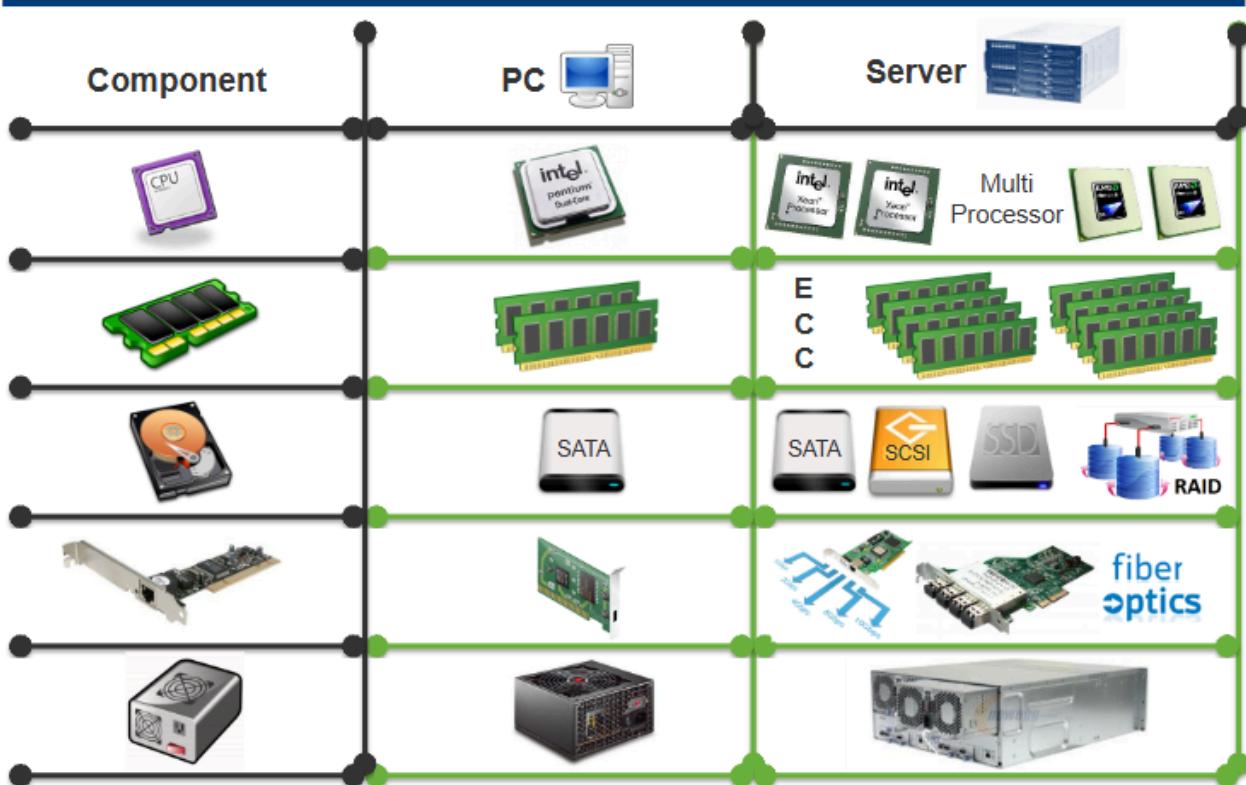
Server Fundamentals

ما لا يسعك جهله عن Servers .

Difference Between PC and Servers

- ال PC : هو معمول لي End User وبيبيقا فيه Limited Resources , لأنة بيبيقا لشخصولي أغراض بسيطة.
- ال Server : هو Powerful pc و معمول أني عن طريقه Deliver Digital Services يعني أني أوصل Services معينه لي Client و من Characteristics أنه يكون Highly available فا هيخليني أحتج Redundant Component عشان يعوض الحاجات الي هتبوظ, هو كمان ب Support مجموعة من Tech مش موجوده غير في Servers .

Key Differences



ال PC عنده One Physical Socket يقدر يحط في ال CPU بعكس ال Server بقدر تحط فيه أكثر من Chip.

ال PC عنده Fixed Number من RAM Slots بعكس ال Servers عنده RAMs ممكן توصل لي 1TB غير أنها بت Support C وهي ال Check Error Correction هل حصل فيه Fault ولا لا قبل ما Load the Application RAM Address في CPU ودا بالتعاون مع OS .

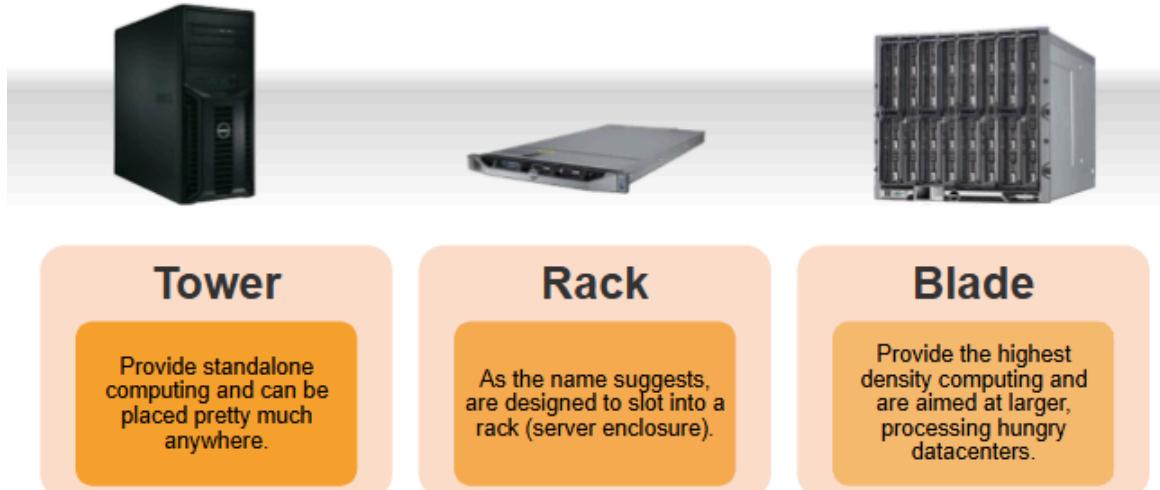
ال PC عنده Fixed Number of Slots دا عشان يستحمل ال Disks غير في أنواع Disks مش بي Support ها زي ال RAID (موجود كا Software ولكن بطى عكس ال Hardware) ودا عشان يعمل Fault tolerance لل Disks وطبعا كل دا موجود في Servers .

ال PC عنده Fixed Numbers من Network Cards ,Servers ال .

Servers Types

Server Form Factor

Servers come in three chassis styles.



Tower

Provide standalone computing and can be placed pretty much anywhere.

Rack

As the name suggests, are designed to slot into a rack (server enclosure).

Blade

Provide the highest density computing and are aimed at larger, processing hungry datacenters.

الTower: جهاز مناسب لي استعمال المكاتب او لي الاستعمال الشخصي لأنك بتقدر تحطه في أي حته.

الRack: بيأخذ مساحة صغيرة وسعره متوسط و cabling متوسط.

الBlade: بيأخذ مساحة صغيرة جدا جدا و سعره غالى و cabling سهل.

Managing Servers

في Technology متوصله بي Motherboard و Networking ومن خلاله تقدر تعمل (Power ON/OFF, Capture Traffic,..etc ,Web interface ..etc) ودي أسمها IPMI عن طريقها تقدر تدي لل OS Connect قبل تثبيت ال Interface . عليها عن طريق زى مانت عايز أكناك قاعد قدامه بالظبط. Servers manage و بت

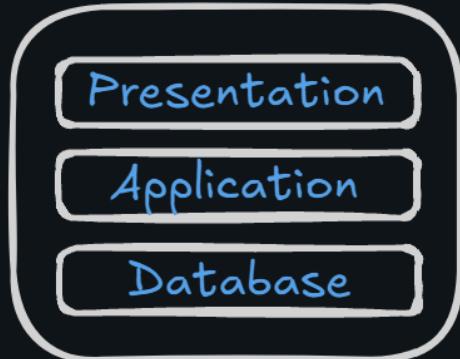
VMware Fundamentals

ما لا يسعك جله عن VMware .

The Problem before VMs

لو حصل Load attack او عالي الي بيتاثر هو server بداعي و ببغا هو ال fault domain ساعتها يحتاج منه Redundant Server Components لـ Fault Tolerance Mechanism . عشان لما يحصل ال fault domain يحول علي Clustering Mechanism . component Servers الي بنفع مع

3 Tier Architecture



- هيتنقسم لي Presentation و app و DB Server ولكن بردو هيبيا لأن لو Server وقع بقيت Servers مش هتعرف تcommunicate مع بعض فا هحتاج يكون فيه Redundant للServer, في حالة أن عندي 10 Servers وعاليز أعمل عمل redundant لي servers على سبيل المثال هعمل من كل service نسخة ثانية يعني هحتاج 60 Server, ودا طبعا هيثير علي Cost, Space, Power and Cooling و هتلافق أن CPU Utilization مش أعلى كفاءة.
- كل اللي اكتب فوق عشان بس عشان يخللي Customer satisfied سواء Functional or Non - Functional .

The Start of VMs

- عشان Business يكون ناجح لازم يطلعك Profit وبالشكل Servers مش بأعلى كفاءة صعب تبدأ Virtualization, وهنا يجي دور ال Business .
- وهو أن Divide the same resources to multiple of resources with the same type Resources .
- بقسم ال resources بتاعتي لي كذا من نفس النوع, أكذك قسمت ال Servers Mini (Virtual Machines) لي عن الثاني من حيث attacks failure و isolationServers عموله هتتأثر وبكدا حسنت ال Server Space و CPU Utilization هتبقا معمولة .
- في اسمها Hypervisor هي اللي بتقسمي ال VMs, وبتعملني Emulation يعني Software بيحاكي ال Network card لي ال VMs عشان توصللي لي Internet .

Type's of VMs

- ال Server ودي في منها نوعين
- ال type1: ودا ال bare metal ودا معناه انه على physical hardware .
- ال type2: ودا ال hosted ودا معناه انه على OS مش على ال Physical hardware .

الي بيحدد أنهي type هو ال CPU Ring ولو لقينا ال 0 CPU Ring فيها -v hypervisor or hyper زى ال 0 CPU Ring ساعتها دا ال bare metal .
نوت: ممكن يكون bare metal بطريقة undirect زى ال windows server 2008 لما تنزلوا بيبقا ال OS هو اللي في 0 ولما بتفعل ال -v Hyper بيطبق تعمل restart وبيكون بعدها هو اللي في 0 Ring .

- ال Storage: هو أني أعمل حاجه اسمها Logical Pool او aggregate و أجمع كل GBs اللي معايا وبعدين بعمل partitioning (LVM) عن طريق ال concept Linux .
- ال Network: زى ال Switch بقسمه لكتا (VLAN) دا بيعتمد على Physical Network Device .
- ال Network Function: كل Device بيكون ليه function معينه بيعملها زى Router, Switch, Firewall, Gateway وهكذا .
- ال OS (Containers): زى ال Server ولكن بيعمل على ال OS كذا OS من أنواع مختلفة و ال OS ال Isolation بتاعه اقل بكثير جدا من Security Server فا بيكون عندي سيرفر واحد فيه اكتر من switch او router او firewall .

Network Fundamentals

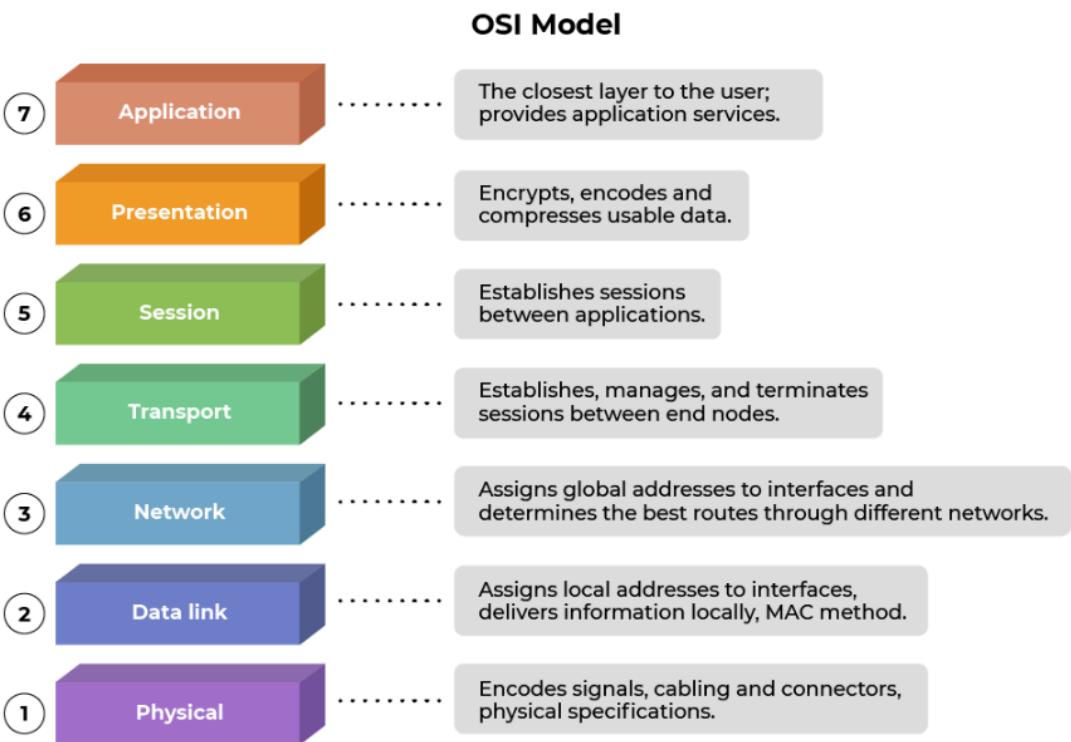
ما لا يسعك جهله عن Network .

- ال MAC Address : ببغا مربوط ب Network Interface Card, وهو طريقة أن ال Devices Communicate مع بعض في Local Network Communication .
- عندي Four type من Addressing communication method :1. ال Unicast : هي أكلم جهاز واحد فقط .2. ال Broadcast : هي أن جهاز واحد يكلم كل الأجهزة الي على ال Network .3. ال Multicast : هي أن جهاز يكلم مجموعة محددة من ال Network .4. ال Anycast : هيSupported في IPV6 و IPV4 ولكن بطريقة مش مباشرة (BGP Protocol) , هو بيكون فيه Multiple Servers عندها نفس ال IP و ال Anycast يشوف ال Nearest server ليه و يعمله Redirect عليه .

حت ال Anycast هتلاقي أستعملتها زي Edge Location

- ال ARP Protocol لما يجي يدور علي MAC ببيعت broadcast msg لكل Devices اللي عنده وبيسأل ال IP دا ال MAC بتاعه ايه و ال IP الصح هبيعت ال Mac address الصح كل دا لو Local , لو في حالة ال Remote هبيعت يشوف ال MAC الخاص بـ destination بدل ما يروح علي gateway وبعدين ال message هو اللي يصل ال message لـ destination .
- ال MAC Address Communicate صعب ولكن بعد كدا لجتنا لي IP وبعدها لي FQDN وهو الشكل الحالي علي سبيل المثال Subnet mask (WWW.XYZ.COM) فـ بقينا محتاجين Resolver يفك ال DNS عشان ناخذ IP وبعدين بيتأكد أن ال IP لو نفس ساعتها هيدور في ال Local ولو لا هيدور Global وبعدين يروح لي مكان ال Switch عشان يصل ال Broadcast message و يجيب بيه ال MAC و يروح عليه .

طيب عشان ال Devices تكلم بعض يحتاج حاجه تنظمهم بي Standards كلهم بيقولوا فاهمنها والي بيطبقي كدا هو ال OSI Model او TCP IP Model



ال Switch هو Device Layer2 و هي الي فيها Mac Address و ببغا عنده Mac table عشان يقدر يبعث ال MSG لي مكانها الصح.

ال Router هو 3 Device Layer و هي الي فيها IP Address و ببغا عنده Route table عشان يقدر ي Map و يجيب MAC من مكانه الصح.

ال data encrypt و authenticate بيعمل secure network layer protocol :internet Protocol Security (IPSec) مابين ال gateway to gateway او host to host او host and gateway packets .VPN في communication

دا سيناريو لي Network يتمشي من جهاز A لي جهاز B:

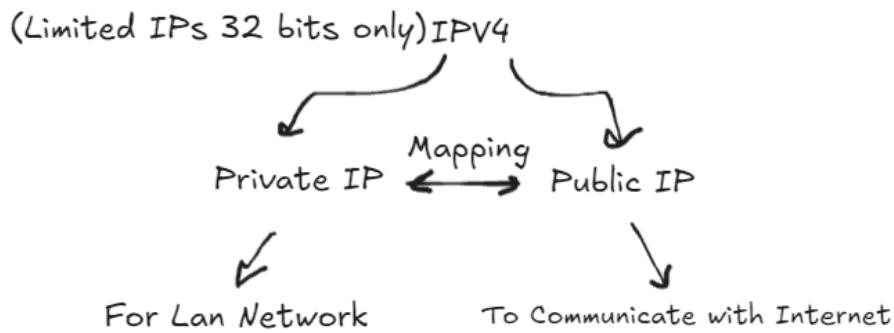
Case 1: Same Local Network

1. Device A resolves `www.xyz.com` to an IP via DNS.
2. Device A checks if the IP is on the same subnet.
3. If yes, Device A sends an **ARP broadcast** to find Device B's MAC address.
4. Device B responds with its MAC.
5. Device A sends data directly to Device B via the switch (using MAC addresses(Table)).

Case 2: Different Network (Remote)

6. Device A resolves `www.xyz.com` to a remote IP via DNS.
2. Device A realizes the IP is not on its local subnet.
3. Device A sends an **ARP request** for the **default gateway's MAC** (router).
4. Device A sends the data to the router's MAC address.
5. The router uses its **routing table** to forward the packet toward the destination network, hopping through other routers if needed.
6. At the destination network, the router uses ARP to find Device B's MAC and delivers the packet.

IP Address



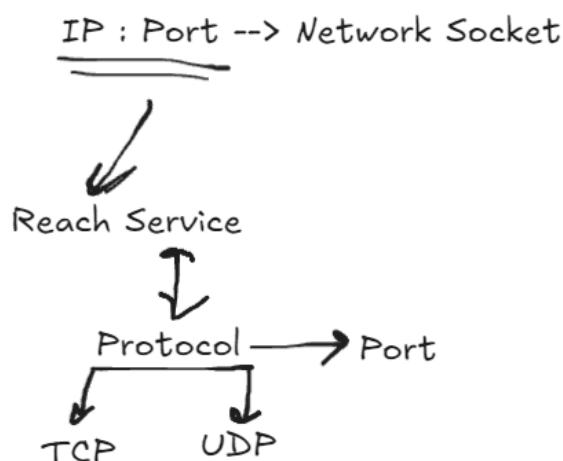
يحصل حاجه اسمها Mapping لـ Private IP و هو اني أدي Public and Private IP لـ Public IP عشان يقدر يكلم ال Internet.

عندى نوعين من ال Mapping :

1. ال NAT: ودا بيدي Private IP العندي بطريقتين (بساوي ال Internet gateway في AWS).
1. ال Static: ودي أنه يثبت Static Public IP for each Private IP (مثل 192.168.1.10 ← 203.0.113.5).
2. ال Dynamic: ودي أنه كل لما يتصل بال Internet بيديه Public IP وبعدين ياخده منه تاني لما يطلع من Internet.
2. ال PAT: بيعمل Port Addressing وهو أنه يستعمل IP واحد لكل ال Devices الي عندي ولكن يزود Port في النهاية.

أمثله على ال PAT:

الجهاز	(Private IP)	المنفذ المصدر (Port)	(Port) بعد PAT	Public IP + Port
	192.168.1.10	5000	203.0.113.1:60001	203.0.113.1:60001
	192.168.1.11	5000	203.0.113.1:60002	203.0.113.1:60002
	192.168.1.12	5000	203.0.113.1:60003	203.0.113.1:60003

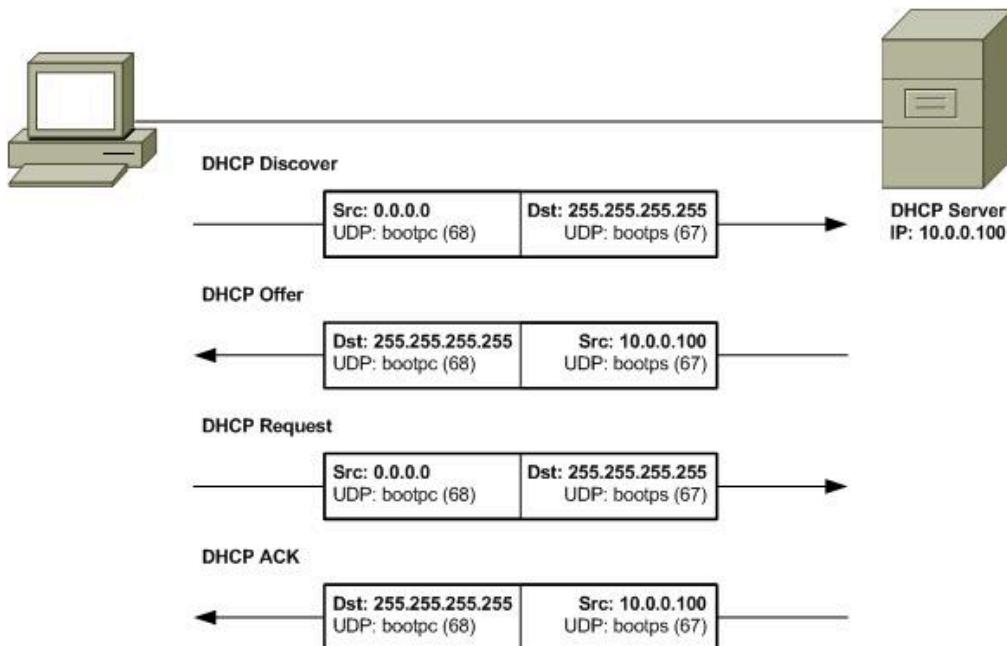


ال دى اسمها Socket و محتاجها عشان أوصل لي Service بتاعتي و ال Service و Protocol هم وجهين لعملة

واحده يعني بتحتاج Service العميل يستعملها وبعدين بتحتاج protocol عشان يعرف يوجه العميل لي Service دي سواء عن طريق TCP او UDP.

Dynamic Host Config Protocol(DHCP)

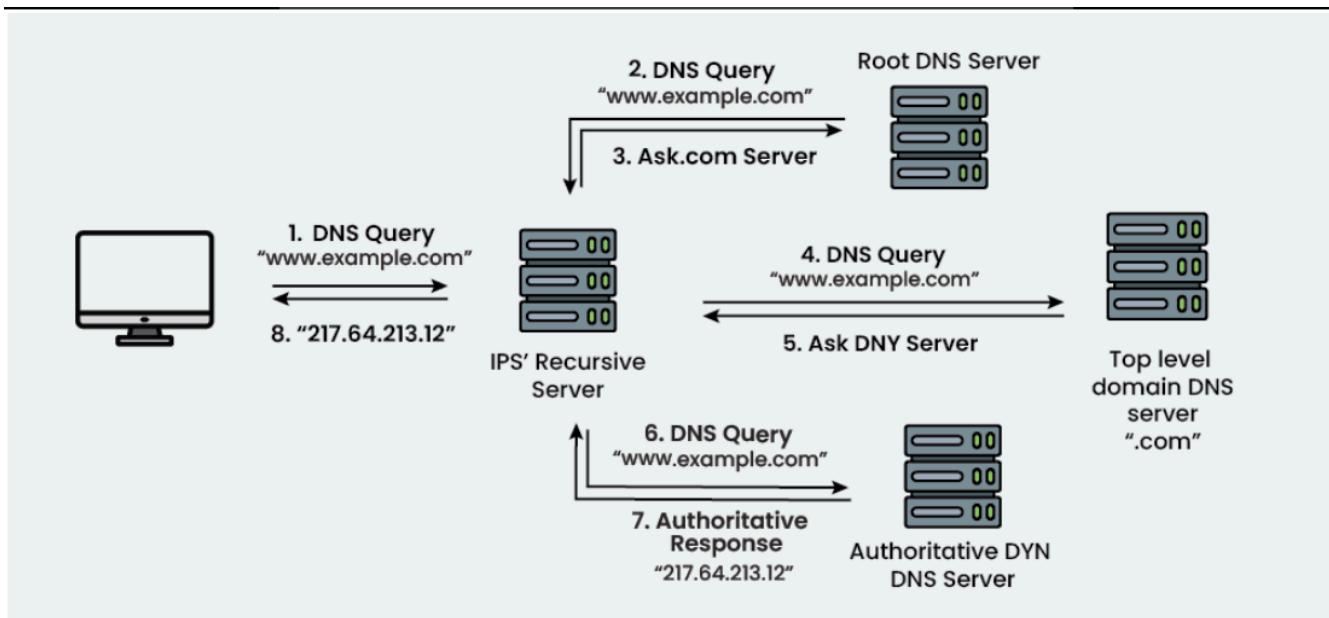
هو بي Assign IP, Subnet mask, Default Gateway من حيث ال Automatic Interfaces Configure بناعتي بشكل من حيث ال Automatic . وهكذا، بدل ما حد يجيي يستعمل الشبكة بناعتي وأقوله حط ال IP كذا و Subnet كذا لا هو بيفقا بشكل Automatic



اول حاجه بيعمل DHCP Discover يعني بيدور على DHCP Server لما ال Client Message تلاقي ال DHCP Server هبيعت DHCP Offer لـ Client لي ال DHCP Request عايز ال IP هتبعدt Client ، ولو ال DHCP Server وبعدين ال DHCP ACK هبيعنله Knowledge Server الي هي طلبها.

معلومات: ال DHCP server لو مش في نفس ال range network ساعتها هبيعت لي Router بشكل broadcast عشان يدور عليه ولكن ال routers بشكل عام بت decline broadcast فـ active يقدر ي AllowBroadcast الي متوجه لي DHCP فقط.

DNS



- أول حاجه في System بناعي ببیقا عندي FQ DN Caches ودا فيه كل ال Hosts أسمه Google زادي , في هلاقیه تحت .etc/hosts/ Linux/Windows/System32/drivers/etc عندي Cache ipconfig /flushdns دا بيمسح أي من DNS.

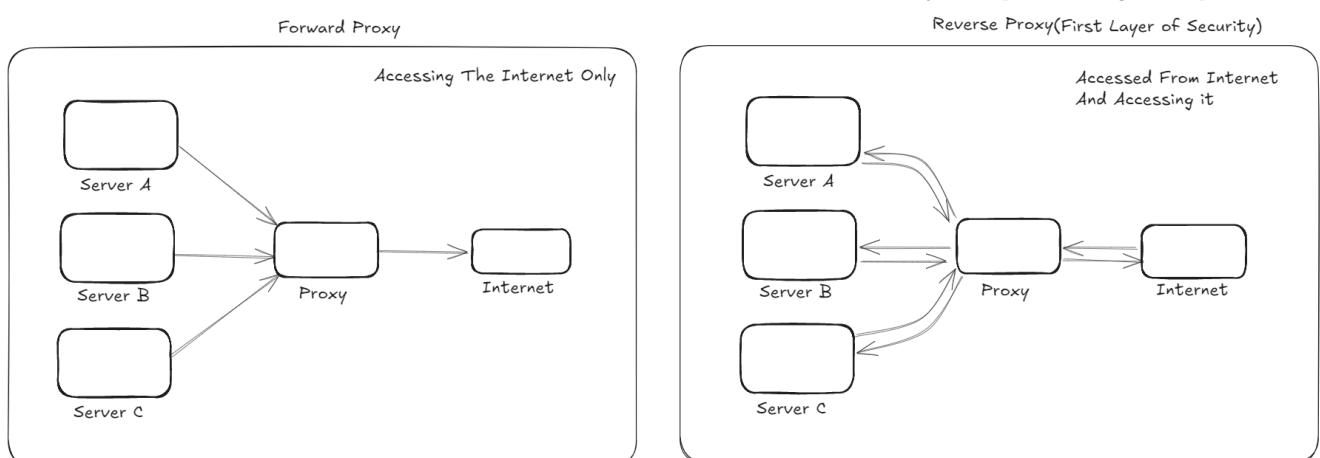
ثاني حاجة هبروح ل DNS Resolver , هو نفسه ببیقا عنده cache ولو ملقوش ببیعت لي أقرب Root DNS Servers عن طريق ال Anycast ويسأله لو يعرف مكان ال FQ DN دا فاييعتلہ روح أصل على سبيل المثال ال ask.com server وهو دا ال TLD الي هو اختصار لي Top level domain server وببیقا فيه com مثلًا او net وهكذا وبعد ما تروله كدا يقولك روح لـ Authoritative server .Cache أسلته ، فاما تبعته وتساله عن IP دا عندي وبيعنه لي Device بناعك ويأخذ منه .Cache

DNS Records

- A Record** - Holds the IPv4 of a domain
- AAAA Record** - Holds the IPv6 of a domain
- CNAME Record** - Domain or subdomain to another domain.

Proxy Service

الفكرة منها أني أعمل Layer 7 Filtering لي



عندي نوعين من Proxy Service

- الـ **Forward Proxy**: وظفته أنه يخرج ال Internet Machines على Network IPs بـ **Network** ما يعرف الـ **Internet** بـ **IPs** بـ **Network**.

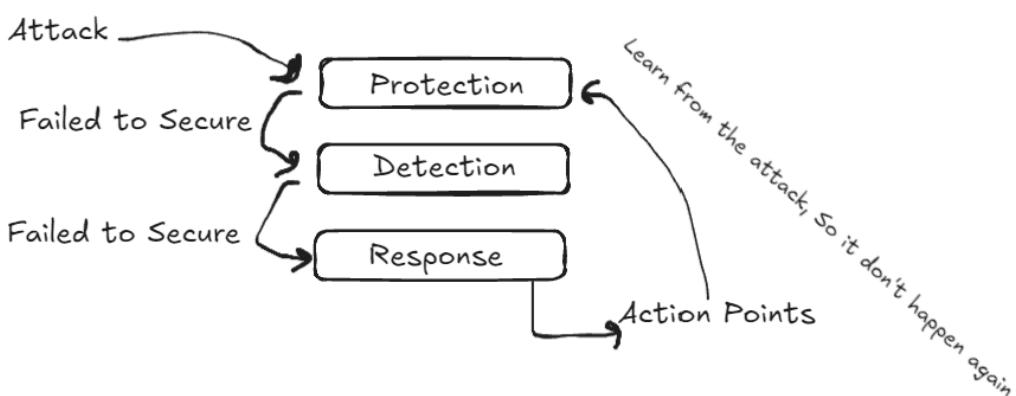
- ال Reverse Proxy: وظفته أنه يخلي ال Internet ي من غير ما يعرف ال Machine Access ال IPs بتاعتهم و ياخذ منهم ويرجعه لي Internet وبيعمل cache بردو.

بتكلم ال Proxy عن طريق Configuration على مستوى ال OS, فا بتقوله وانت خارج/داخل Machines من ال Internet أستخدم ال Proxy .

Security Fundamentals

ما لا يسعك جهله عن Security .

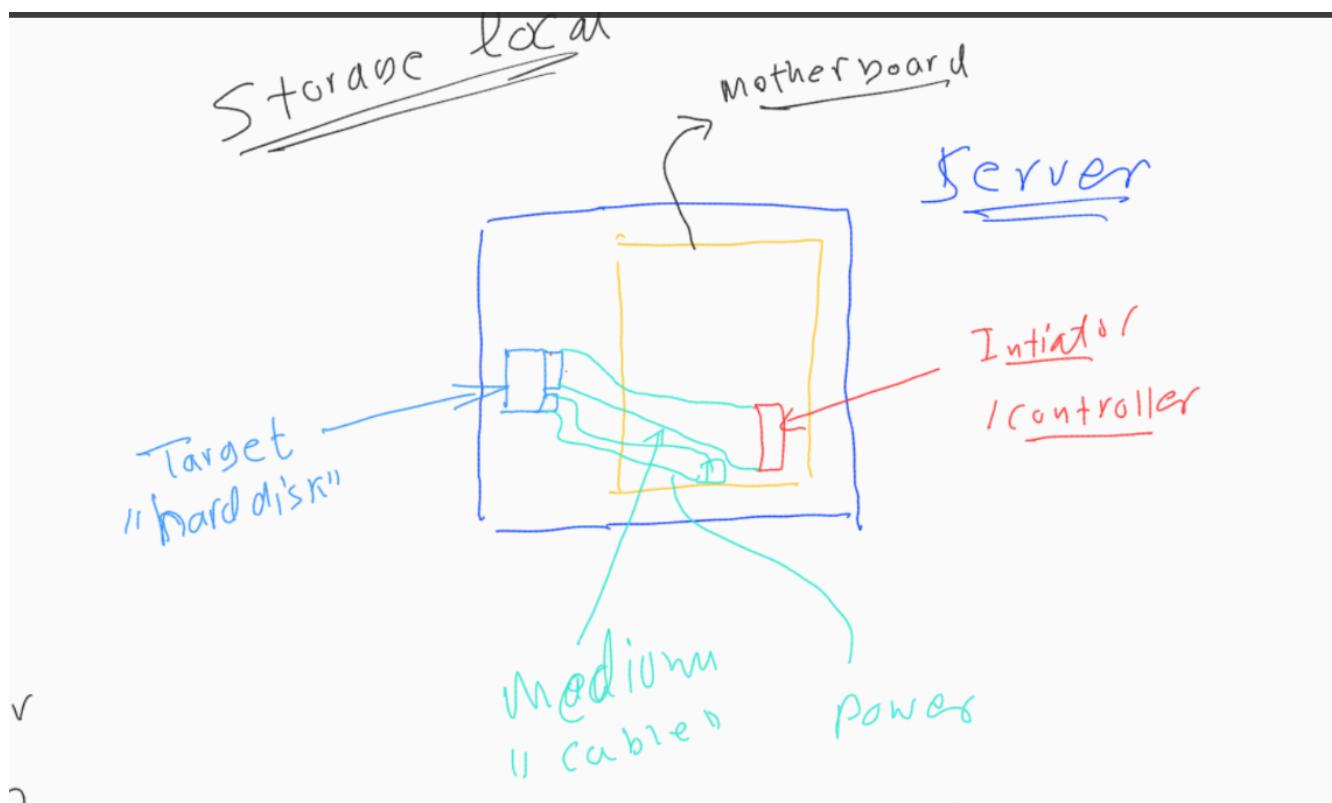
في Layer Security عدي كذا



- ال Protection: هي ال layer الي بتحمي ال Services بتاعتي.
- ال Detection: هي ال layer لو ال Protection System (IPS) احترقت بيبدأ يعملها عن طريق ال Instruction Detection System(IDS).
- ال Response: هي ال layer إللي بتأخذ ال Action عشان تصد ال attack وبعدين تبعنه لي Protection عشان ميحصلش تاني.
- ال Firewall:** ممكن يكون Software او Hardware .
- في حاجه اسمها Security Zones وهي مكان كل Application Component او Service بتاعي يحتاج يكون فيه Firewall بتحدد مين يتعملها Allow او Deny, فا علي سبيل المثال لو عندي App و DB و ال APP مش Allowed أنه يكلمه هيبقا في Firewall يمنعه, فا كل Security zone ببি�قي فيها Firewall ببি�قي فيه مين allowed و مين Denied.
- ال OS عنده Range من Network Ports, ببি�قي عنده من 0 لي 1023 Port registered لي أستخدامات معروفة و من 1024 لي 65535 فاضيين.
- ال Communication ببি�قي عباره عن Request و Response في أي حاجه عامة, ال Request الي جي لي ال App دا اسمه Inbound و ال Response الي رايح دا اسمه Outbound .
- Firewall Types:**

 - ال Packet filtering: ودا يعني بيتأكد من inbound و outbound بنقول عليه stateless(باتحتاج تتأكد رايح جي)
 - ال Packet Inspection: ودا يعني بيتأكد من inbound فقط بنقول عليه statefull(باتحتاج تتأكد من الي جي فقط) ودا لأنه ببقي outbound ensure ال packet info save في RAM فا مش بيحتاج ي
 - ال Application WAF: بتتأكد من مجموعة من ال attacks مش واحد فقط, زي ال WAF.

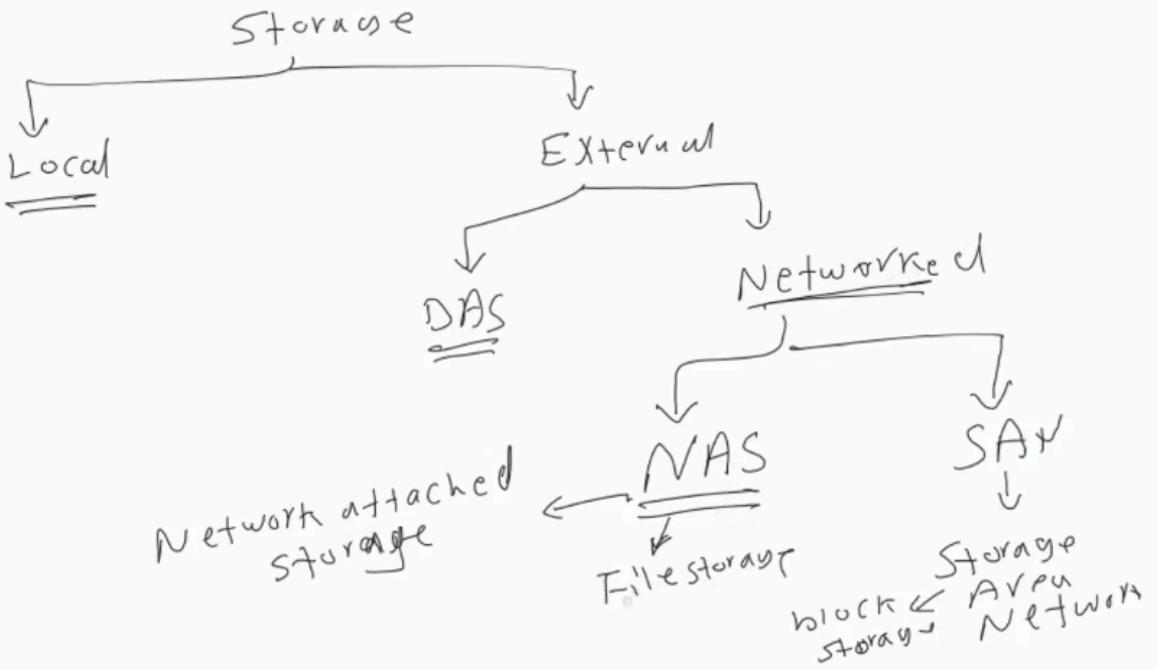
Storage Fundamentals



- ال Server بيبيقا فيه ال Hard disk متصل عن طريق medium(cable) بي ال Initiator(controller) في ال Motherboard يكون متصل بي Power, لو عايز اعمل Commands لـ hard disk (بتبقا ال Read/write) بتبقا عن طريق ال initiator وبعدين لي medium لغاية ما يوصل لـ disk, وعشان ال disk يفهم ال initiator ه يحتاجوا يتكلموا بنفس ال .language=Protocol

- ال Hard disk و Initiator كانوا بيكلموا بي Protocol IDE ودا مع ال PATA(Parallel ATA), وبعد كدا بقا في حاجه باسم AHCI ودي مع SATA(Serial ATA) بت Support AHCI و IDE, ال AHCI بيدعم ال Hot Swapping ال Native Command و هو شغل عكس ال IDE ميقرش يعملها و كمان أبطئ منه ودا لأن ال AHCI فيه Read and write Performance الخاصة بي Queuing (NCQ)

- ال SCSI ودا ليه Protocol مختلف بيستعمل حاجه اسمها SCSI Protocol or Standard Parallel SCSI, ودي نسخة ال SCSI Protocol ولكن version Serial في هي SAS وبيستعمل نفس نوع ال Protocol ولكن version أعلى.



1. Types of Storage:

- الـ **Local Storage**: دي الـ Storage اللي بتكون Direct Connected بالجهاز (مثلاً: HDD أو SSD داخل الكمبيوتر أو السيرفر).
- في حاجه اسمها **DAS**: دي اي HDD or SSD وغيرها بيكون متصل بشكل مباشر مع Device بقىاعي وعيبيها ان مسافتھا محدوده.
- الـ **External Storage**: دي الـ Storage اللي بتكون برا الجهاز، و بتتصل بيها عن طريق كابلات (زي USB أو Thunderbolt).

Note

الـ **External Storage** لو متوصله بجهاز واحد بس بدون شبكة، فھي تعتبر **DAS** برضه.

2. Networked Storage:

- الـ **NAS (Network Attached Storage)**
- ده نوع من الـ **Storage** بيكون متصل بالNetwork ويوفّر Shared Storage لأجهزة كتيرة ودا بيخليها External Storage.
- بيكون **File-Level Storage**، يعني التعامل مع الـ Files (مثلاً: فتح ملف أو حفظه أو أعمل creation).
- الـ **NAS Protocols** المستخدمة مع NAS.
- الـ **NFS (Network File System)** مخصوص لأنظمة Unix و Linux.
- الـ **CIFS/SMB (Common Internet File System/Server Message Block)** مخصوص لأنظمة Windows.
- الـ **SAN (Storage Area Network)**
- ده نوع من الـ **Storage** بيكون متصل بالNetwork بردو، لكن بيكون **Block-Level Storage**، يعني بقدر أعمل External Storage او بيخليها Formatting او partitioning.
- الـ **SAN Protocols** المستخدمة مع SAN.

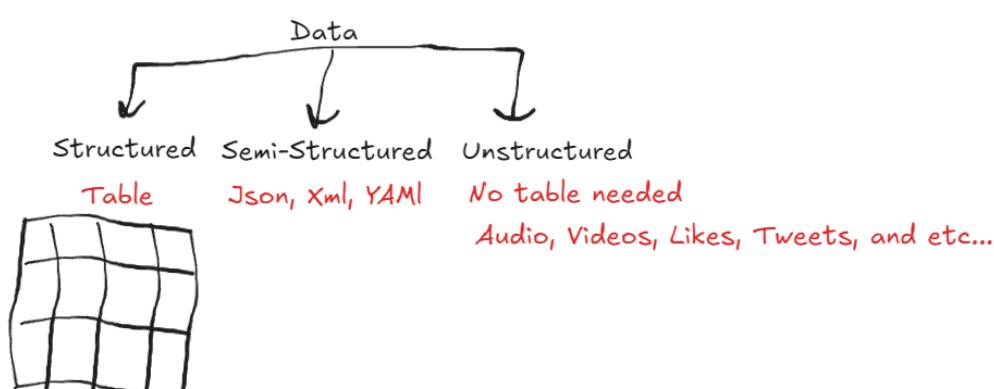
- الـ **iSCSI (Internet Small Computer System Interface)**: يستخدم نفس الـ SCSI لكن عن طريق الـ Ethernet أو الـ Fiber.
 - الـ **Fibre Channel (FC)**: سريع جداً يستخدم الـ Fiber Optic Cables، ومخصص للشبكات الكبيرة.
 - الـ **FCoE (Fibre Channel over Ethernet)**: يسمح باستخدام الـ Fibre Channel Protocol عن طريق Ethernet.
 - الـ **Object Storage**:
 - ده نوع ثالث من الـ Storage (جانب File و Block) ويقدر يشيل 5TB.
 - بيكون عبارة عن Container او بلغة AWS بيقا اسمه Bucket ودا شبيه ب Folders في الـ block storage.
 - الـ Object بيكون الـ data نفسه و يكون عبارة عن ID و metadata، والـ Object ID عن الـ data.
 - زى .size, Last modify, Owner.
 - مشهور في الـ Cloud Storage (مثلًا: Amazon S3 او Google Cloud Storage).
 - الـ Protocols المستخدمة عشان أتعامل مع الـ Object Storage هي HTTP/HTTPS او S3 API.
 - الـ Use Case:
 - مناسب لي الـ distributed architectures.
-

3. Difference Between Back-to-Back and Switching:

- الـ **Back-to-Back**:
 - دي طريقة توصيل مباشرة بين جهازين (مثلًا: سيرفر و Storage) باستخدام Cable واحد (زى .Fiber أو Ethernet).
 - بيكون فيها Direct Connection بدون أي أجهزة وسيطة (مثلًا: Switches).
 - بيكون Limited بعدد الأجهزة اللي ممكن تتصل بيها.
 - الـ **Switching**:
 - دي طريقة توصيل باستخدام **Switches**، علشان Connect أكثر من جهاز مع بعض في شبكة.
 - بتسمح باستخدام أكثر من Cable (زى Ethernet أو Fiber) حسب نوع الـ Switch.
 - بتتوفر Network أكبر في توصيل الأجهزة وتوسيع الـ Flexiable.
-

Databases Fundamentals

ما لا يسعك جعله عن Database.



- ال Data عبارة عن 3 أنواع.
 - Structured ال
 - بتبعا Tabled وهي الى بيقال عليها (Relational Databases(RDMS), زى Oracle DB, MS SQL, Postgres)
 - وهكذا.
 - ال Semi Structured
 - زى ال JSON, YAML, and XML Key-value Databases وتحت فئة ال No-SQL Databases والى هى MongoDB, Redis DB, زى Non-Relational Databases
 - ال Unstructured
 - بتبعا مش محتاجه Table ودى ال
 - في Classifications Data تانية لي
 - ال Online Analytics Processing(OLAP)
 - بيعتمد على Historical Data عشان يقدر يعملها Analytics ودى حاجه زى Data warehouse بجمع ال data فى مكان ولو عايز تجمع data معينة بتعمل queries ويطلع منها Reports, وفي حاجه اسمها data mart ودى بتعمل ال ETL(Extract, Transform, Load) warehouse لكتا فئة وبها بيحصل حاجه اسمها
 - ال Online Transaction Processing(OLTP)
 - دى ال Transactions الى بتبعا بشكل يومي وبنحتاج تعدل عليها, يعني بتعمل بشكل يومي create, read, update, or delete(CRUD)
 - ال Big data لما طلت ظهر حاجه اسمها Data lake وهي حاجه شبيها بال data warehouse, بس هيا بيتخط فيها ال structured و semi unstructured بس مش بتبعا processed
 - ال Data consistency مفهاش Load balancing لوحده لأن لو عندي أتنين بيعدولوا في Entry يعنيه هيكون عندي مشكلة في DBs (أني أحافظ على data المعايا ومحصلش Override), هنا قالوا في حاجه جميلة وهي ال Data Sharding ودى بتتكل ال load balancing وهي لو عندي كذا DB وعندي حاجه هت Load Balance عليهم وعندي table في كل DB فا لو جالي حاجه تتعدل في entry معين هتعدي على hashing algo وهو يوجه مثلاً أنك تروح على DB الأولى في row كذا و DB الثانية تروح في column دى وهكذا, وبكدا هو هيساعد على تحسين الأداء وتقليل مشاكل ال data consistency.
-

Cloud Foundation & Architect

Now that we've covered [the foundational concepts](#), let's explore the core principles of cloud foundations and architecture. This topic will help you understand how AWS designs and manages scalable, secure, and cost-effective cloud environments tailored to support diverse applications.

Cloud Concepts

Before diving into specific AWS services, it's important to understand the core ideas behind cloud computing. In this section, you'll learn what the cloud is, why it matters, and how it changes the way businesses build and manage technology.

Why Cloud

Cloud Computing Characteristics مقارنة ال Cloud عن طريق

-1 Place Resources

الـOn-Premises: هحتاج مكان نشتريه/نأجره عشان نحط في الـResources بشكل عام.

الـCloud: وهي المكان مش هضرط اشتري ولا مكان لأني بعتمد علي الـregions اللي موجود فيها resources الخاصه بكل provider يعني هوصل لي Customers أنحاء العالم منغير ما أحتج DC.

2- الـSizing

الـOn-Premises: هحتاج احسب الـSizing وهو أحسب هو الـApplication بناعي متوقع هيجله كام user فا بيتحسب عن طريق نوعين، الـAvg وهو متوسط كام users يوميا او Peak وهو أعلى نسبة users بيشفوا الموقع كانت كام.

في حالة الـavg لو أفترضنا أن كان في خصومات و الـusers زادوا عن المتوقع ساعتها الـresources بتاعتي هتبقا overutilized ودي ببساطة الـapplication مش هتشتعل كويس و الـuser experience هتقابل مشاكل مختلفة.

الـCloud: الـsizing احنا بنأخذ الـavg ساعتها مش هتواجهنا نفس المشكلة! عشان الكلاود عنده ميزة الـElasticity وهو التكيف و المرونة، يعني لو بقا في load على على الـresources بتاعتي هقدر أخلي الـresources تتكيف معاه عن طريق أني الـscale out ولو الـscale in مبقتش ساعتها اعمل في use resources.

لو جه **Peak**، الموارد هتزيد (Auto-scaling)، ومش هتدفع تكفة استخدم كامل لأنك هتستخدم الـResources فقط خلال فترة الـPeak، وبعدين هتقلّهم تاني (Pay-as-you-go).

3- الـCosts

الـOn-Premises: لما بحتاج resources بتشتريها و بتدفع بنظام Capex وهي معناها أني أدفع المبلغ كله upfront (مقدما).

الـCloud: لم بحتاج resources بتلاقيها متوفرة و بتدفع بنظام Variable Expenses وهي معناها أني أدفع علي قد أستخدامي للـResources دي.

4- الـTime to go Global

الـOn-Premises: الوقت إللي هستناه عشان الـresources دي هنلاقيه في حدود الـ3-6 شهور غير الوقت إللي لسه هجهز في الـEnvironment عشان تستقبل الـapplication كدا ممكن توصل لي 9 شهور عشان اجهز الـuser بوصول للـresources maintenances.

الـCloud: الوقت إللي هحتاجه عشان استعمل الـresources دي، وهو بيكون دلائق معدودة فعليا (Speed and agility).

5- الـControl Over The Resources

الـCloud: هتحكم في الـresources دي كلها على الـcloud وأفعالها في دلائق عن طريق الـInfrastructure as code وهو اني أكتب كل الـresources إللي محتاجها في code وبسيطة اعمل apply او destroy الـresources commands.

Introduction to Cloud Computing

في عندي **Two Models** لازم يكونوا في حساباتي لأنهم بيأثروا على **Cost, Performance, Security** و الـ **Two Models** هما:

1. Deployment Model

Private Cloud - On Premises

هو عبارة عن أن الشركة بتكون عندها الـAPP اللي بتعمل عليه host لي الـPhysical Hardware بتاعه او أي يكن وممكن ت بتاعتهم لي customers Services تانين.

Public Cloud - Cloud

هو عبارة عن أن الشركة تكون حظه كل الـ load بناء على الـ Cloud من حيث الـ VMs و Networking و Storage وكل حاجه من الآخر بتبنا على الـ Cloud.

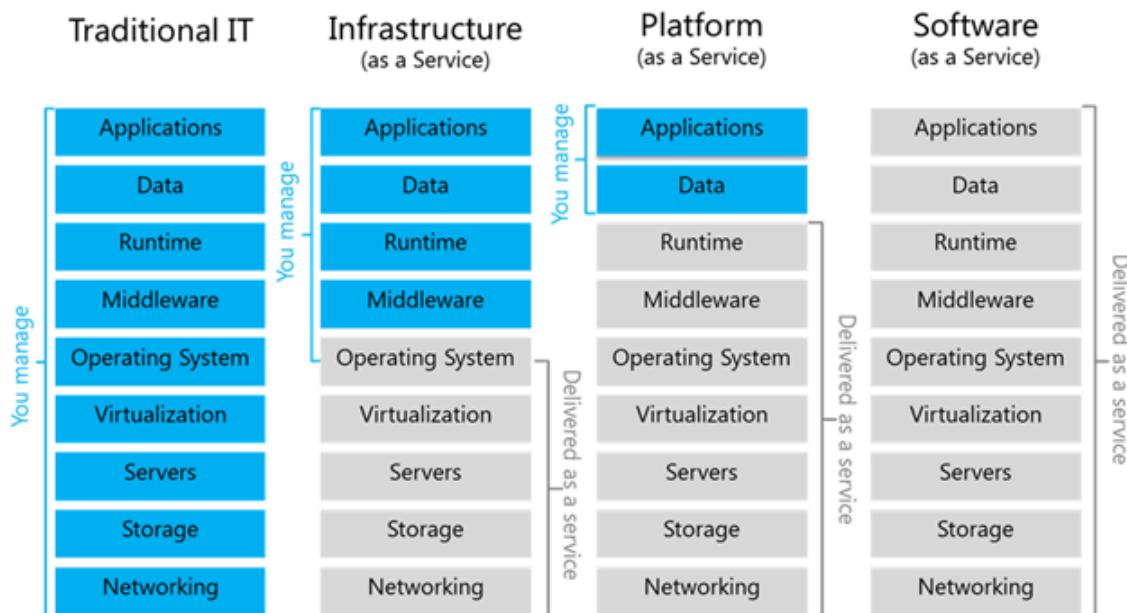
Hybrid Cloud

هو عبارة عن Mix مابين الـ Public Cloud و الـ Private Cloud, بمعنى انك مثلاً تستعمل الـ VMs بناء على الـ Cloud ولكن متمسك بي Databases الي عندك في On premises.

Community Cloud

هو عبارة عن أن Data Center مشتركة في خاصه بيهم وكل شركة بتـ consume الـ Resources من الـ Multiple Company DC.

2. Service Model



بعض يا معلم، الـ Cloud في الـ Service Models متقسمة 3 مستويات رئيسية:

1. IaaS – Infrastructure as a Service

دي كأنك واحد سيرفر فاضي وانت اللي هتركب عليه كل حاجتك.

- انت مسؤول عن: OS, التحديثات, Applications, Security.
- الـ Provider بس بيديك الـ hardware والـ Network.

أمثلة:

- زى AWS EC2
- زى Azure Virtual Machines
- زى Google Compute Engine

2. PaaS – Platform as a Service

دي كأنك واحد سيرفر جاهز معنول Setup وانت كل اللي عليك تركب Application بناعنك وخلاص.

- هـ ما مسؤولين عن OS, التحديثات, Runtime.
- انت مسؤول عن الكود بناعنك بـ s.

أمثلة:

- زى AWS Elastic Beanstalk

- زی Google App Engine
- زی Azure App Service
- SaaS – Software as a Service .3**
- دی انت مش شایل هم ای حاجة خالص.
- مجرد تستخد ال Application الجاهزة.
- البروگایدر بیشیل هم کل حاجة من تحت.

أمثلة:

- زی Gmail
- زی Google Docs
- زی Salesforce

Note

Distributed applications مابین data management و Communication هو الی بیسمح ب Middleware

The History of AWS

The Beginning of Amazon(2000)

کانت Amazon بتحاول تعمل Ecommerce Service الخاص بیها و ت Build Online Shopping sites

Early 2000s

کانت طورت ال Well-Document APIs Based ونجحت فیها.

Amazon Web Services(2006)

کانت بدایه انهم بیبعوا ال Services الی طورها و ال Experience Services الی مردا بیها، وکانت اول Three Services هما S3 و EC2 و SQS.

Intro to AWS

فا زی مقولنا AWS هي بتقدم Services كتيرة ولكن اختيارك لی Service ب depends on your business goals and technology

تقدر تتعامل مع AWS بأشکال كتيره عن طريق سواء Code او Configurations او Console او interact فا ای AWS بیبقا عباره عن API Requests و بيرجطي بی Response، وعندی 3 طريق عشان AWS Interact :

1. عن طريق AWS CLI
2. عن طريق AWS Console
3. عن طريق Programming Languages (يعني SDKs)

AWS Cloud Adoption Framework "AWS CAF"

لبيها 6 :perspectives

• Business Capabilities

1. الـ business needs: لازم تبقا فاهم أن الـ Business يكون بي Aligned أو ينسق مع IT Needs.
2. الـ people needs: لازم يكون فيه Prioritize Training, Staffing, and organizational changes عشان تبني agile organization.
3. الـ governance needs: شبـهـ الـ Business في أنها بتحاول Align IT Strategy و الـ goal من حيث Business Strategy.

• Technical Capabilities

1. الـ platform needs: محتاج أبـقا فاهم طبيعة الـ Cloud System ولما يروح على الـ IT Architecture هيـشتغل ازاي من حيث Compute, Network, Storage, Database, Systems and Solution arch, and Application (Development).
2. الـ security needs: لازم تتأكد من Organization Needs تكون تحقق الـ Security Objectives من حيث (control, Infra Security, Data Protection and Incident response).
3. الـ operations needs: هوـ الـ Resource management أو Monitoring, من حيث الـ Day-to-day operations او disaster recovery و analytics و Release management.

Architecting Fundamentals

Before building solutions in the cloud, it's important to understand the fundamentals of cloud architecture. In this section, we'll explore key design principles and best practices that help create scalable, resilient, and cost-effective architectures using AWS services.

Role of a Cloud Architect

1. Plan

- بـتخطيطـ ليـ Technical Strategyـ بـنـتـاعـتكـ الـ Businessـ الـيـ بـتـخـدمـ الـ
- بـتشـوفـ الـ Businessـ عـاـيـزةـ اـيـهـ وـ طـلـبـاتـهـ عـشـانـ تـعـملـواـ .Solution

2. Research

- بـتشـوفـ الـ Servicesـ الـيـ عـنـدـكـ بـتـعـمـقـ .
- بـتشـوفـ الـ Workloadـ الـحـالـيـ مـاـشـيـ اـزـايـ وـ تـرـاجـعـةـ .
- بـتـعـملـ الـ Designـ اوـ تـصـورـ لـيـ Protـotypeـ بـنـتـاعـكـ وـ تـشـوفـهـ هـيـمـشـيـ اـزـايـ .

3. Build

- بعدـ مـتـظـبـطـ الـ Prototypeـ هـتـبـدـأـ تـعـملـ خـطـةـ اـزـايـ تـعـملـ transformationـ للـ شـكـلـ الـجـدـيدـ بـدـونـ مـأـثـرـ عـلـيـ Usersـ .

General Design Principles

الـ Auto Scalingـ فـيـ Cloudـ اـنـتـ بـتـعـتمـدـ عـلـيـ الـ Cloudـ فـاـ مشـ مـحـتـاجـ تـشـوفـ Peakـ وـ Resource Capacityـ وـ تـحـطـ الـ Minimum Workloadـ عـلـيـهـ .

Production Scale Environment: Test Systems at production scale
العشرات من الأجهزة المترابطة التي تعمل في بيئة الإنتاج.

Manual processes: Automate to make architectural experimentation easier
العمليات اليدوية: تجعل التجارب المعمارية أسهل.

Fixed: Allow for evolutionary architectures
النماذج الثابتة: تتيح التطور التدريجي.

Drive architectures using data: Metrics
الهيكلية المدعومة بالبيانات: مetrics.

Improve through game days: Disaster attacks Failures Simulate
تحسين من خلال أيام اللعب: الكوارث الهجمات الأعطال.

AWS Well Architected Framework

الفكرة منه لو انا عايز أعمل Design او Solution ببقا عايز أعرف ال Best Practices, بتساعد أي solution architect بيهارع يعمل Prototype.

Operation excellence

- هو بيتأكد ان كل Process شغالة تمام و قيمة Business Monitor و عشان Systems لي Run و يقدر User لي Monitor.

أعملي كذا محتاج:

- Automate Changes

أي Changes تحصل على System تكون Automated.

- Responding to events

أني أبقا Ready على System Events أي React على System.

- Defining standards to manage daily operations

بيقا عندي Rules بعملها بشكل يومي عشان أتأكد ان ال Operation شغالة.

الDesign Principles الخاصة بي Pillar.

- Perform operations as code

نفس ال Automate Concept أي حاجه ينفع أعملها Automation.

- Learn from all operations events and failures

أي حاجه تحصل لي System failure سواء او غيره حتى دماغك أنها فرصة أنك تتعلم حاجه جديدة.

- anticipate failure

ت Design System بيتحقق اي Failure, يعني على سبيل المثال تحاول تمنع اي Single point of failure او customer trust على Downtime impact Reduce.

- refine operations procedures frequently

خلي ال Operations Adapts مع new technologies.

بيقا عندك Docs على طول بترجمتها في حالة outage في system.

- make frequent, small, reversible changes

تعمل updates بشكل سريع و بسيط وأنك تقدر ترجع Version القديم لو الحال في مشكلة.

الملخص

1. **Automation is king** - If you do it more than twice, script it
 2. **Small steps win** - Think "evolution" not "big bang" changes
 3. **Document everything** - Especially lessons from failures
 4. **Assume breakdowns** - And plan how to handle them
 5. **Never waste a crisis** - Every problem teaches you something
-

Security

- هي أني أ Protect و أ Monitor ال info و system بتعاي وعشان أطبقا محتاج:
 - Protecting confidentiality and integrity of data
 - Identifying and managing who can do what
 - Protecting systems
 - ال events System بتعاي وأشوف اي شئ غريب: Establishing controls to detect security events
 - بيحصل عليها سواء Software او Hardware
 - Pillar الخاصة بي Design Principles
 - ال identity foundation عرف identity اي حد علي system ولو ملهاوش لازمة اطلعه
 - Enable traceability
 - automate Security best practices
 - apply security at all layers
 - protect data in transit and at rest
 - keep people away from data
-

Reliability

- هي أ Recovery procedure (إجراءات الرجوع من المشكلة) عشان أخلي ال business مكمل وعشان احقق دا محتاج:
 - Designing distributed systems
 - اوزع ال Load بتعاي على أكثر من Resource عشان ميقاتش في Single points of failure توقة
 - Recovery planning
 - بيقا عندي خطط احتياط عشان لو حصل مشاكل في System أقدر ارجعة في وقت قصير
 - ال Outage (يعني توقف الخدمة) بتحصل بشكل Random مش بتحصل بشكل Updates فا محتاج يكون عندي System behavior و القدرة أني ارجع لي Version الى كان شغال, وأني اشوف ال Deployment strategies Real-time
 - Pillar الخاصة بي Design Principles
 - Recover from failure
 - dynamically acquire computing resources to meet demand
 - test recovery procedures (Very important)
 - scale horizontally to increase aggregate workload availability
-

Performance Efficiency

- أخلي ال Resources بـ efficiency بـ بـ بناعتي في أقصي , وعشان احق دا محتاج:
 - Selecting the right resource types and size
 - monitoring performance
 - making informed decision to maintain efficiency
 - ال Design Principles Pillar الخاصة بي :
 - Go global in minutes
 - Experiment more often
 - use serverless architectures
-

Cost Optimization

- أ Deliver business value عشان اخد أقل price ولكن مش على حساب ال performance, الغي او بعد عن ال unnecessary costs وعشان احق دا محتاج:
 - Understanding and controlling where money is being spent
 - selecting the most appropriate the right number of resource types
 - analyzing spend over time
 - scaling to meeting business needs without overspending
 - ال Design Principles Pillar الخاصة بي :
 - Adopt A consumption model
 - Stop Spending Money on undifferentiated heavy lifting
 - Analyze and attribute expenditure
-

Sustainability

- هي أني ال Minimize environmental impacts الخاص بي running workloads
 - عشان cloud في AWS بـ يستعمل 100% من renewable impact على ال Env عكس ال on-premises env impacts فـ تكون حريص على أـنـكـ مـعـمـلـشـ بـ يستعمل renewable 100% ومـمـكـنـ يـعـمـلـ impact الـ envـ بـ Pillar الخاصة بي :
 - understand your impact
 - maximize utilization
 - reduce the downstream impact of workload
 - establish sustainability goals
-

AWS Well-Architected Tool

هي Tool بـ تساعدك انك تبني Design يكون Secure, High-performing, resilient, and efficient cloud architectures . ودا بيكون عن طريق أسلة وانت تجوب حسب situation workload compare وبيعمله AWS Best practices .

Best practices for building solutions on AWS

Trade Offs

- **Design trade-offs**

- لما بتDesign Solution ببغا مهم أنك تحط في الاعتبار أنك هتضطر تضحي بحاجة (trade-off) مقابل تحسين fast durability او consistency او efficiency او cost. يعني انت ممكن تضحي بي عشان تأخذ speed او ممكن تهتم بالcost saving على ال deployment.
- فا لازم ال trade-offs دي تكون بناء على empirical data يعني ارقام تجريبية عشان تأخذ قرار سليم.

- **AWS Best practices for solution design and common mistakes(anti-patterns) to avoid**

- **Implementing Scalability**

- هحتاج تتأكد ان الArchitecture بناعك بيقدر يhandle workload changes في الـ.
 - أنك تقدر لـ scaling implement في كل layer دا هيساعدك تمنع أي مشاكل ليها علاقة بالcapacity, فا انت تقدر تخلي ASG لـ demand وما يبغا في system trigger لي CloudWatch Monitor يـ.

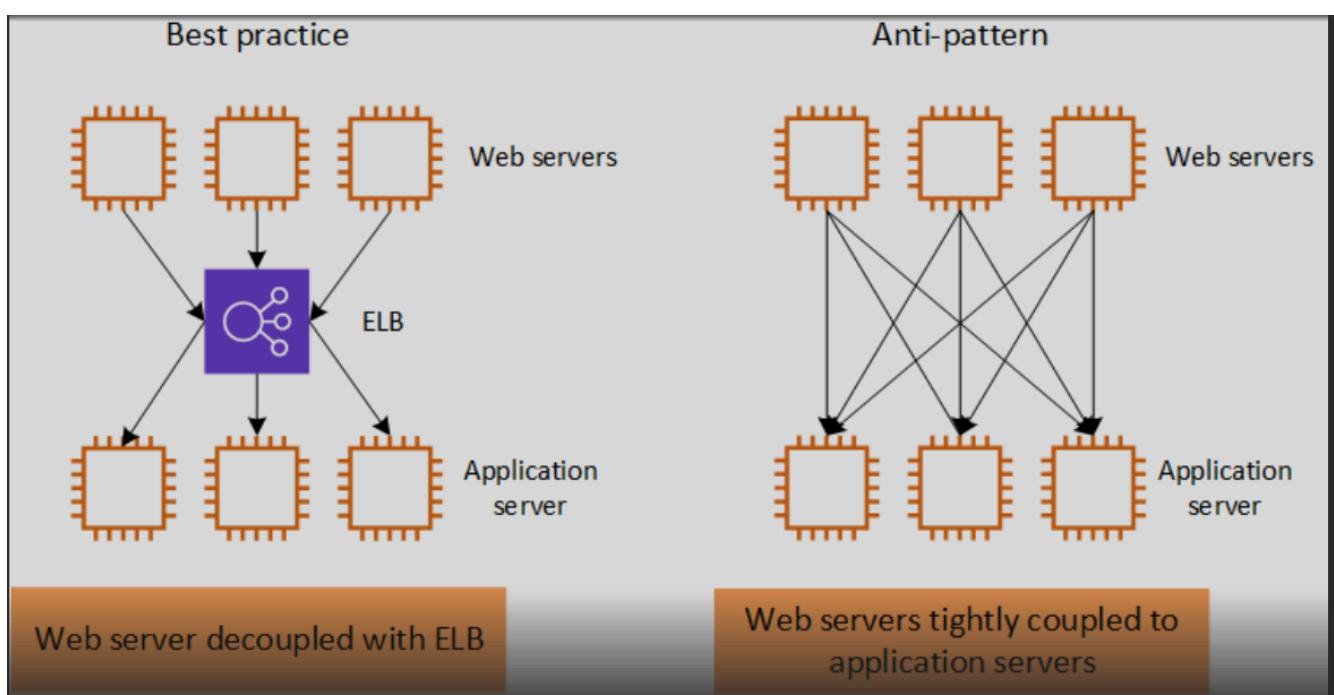
- **Automating your environment**

- أنك تخلي الـ provisioning و termination و configuration لـ automated resources.
 - أـ AWS عندـها Build-in monitoring and automation tool لـ infra تساعدـك عـشـان تـاخـذ قـرـارات سـريـعة بنـاء علىـ healthy instance زـي ASG و detect changes لـ CloudWatch.

- **Using IaC**

- أنك تتحكمـ في resources بـ Cloudformation او Terraform زـي Code.
 - دـ لأنـ الـ IaC هـتخـليـ الـ Human error أقلـ و Speed أعلىـ و More control.

Loosely Coupled Components



- **Using Loosely Coupled**

- تعملـ الـ Independent Components Architectureـ علىـ فيهـ مـيـاـزـ عنـ التـانـيـةـ، عـشـانـ لـوـ حـصـلـ مشـكـلةـ فـيـ Componentـ.
- بـيـخـنـ منـ Scalabilityـ وـ System reliabilityـ.
- بـتـحـقـقـ دـاـ عـنـ طـرـيقـ إـنـكـ تـسـتـعـمـلـ Intermediariesـ عـلـيـ انـهـ بـكـونـواـ Message queuesـ وـ Load balancersـ وـ system layersـ مـايـبـينـ.

- **Designing services, not servers**

- يعني أنك تختار تستعمل Serverless solutions عن traditional EC2 على رغم ان EC2 بطيء Flexibility اكتر بس أنت بتختار علي حسب use case بقاعدتك.
- الـ Services زي Lambda, SQS, DynamoDB, and ELB يقدموا ليك فرصة احسن أنك تـ Scale و تـ لـ ax.
- Cost efficiency و Performance احسن.

• Choosing the right database solution

- Match technology to the workload, not the other way around
- Read and write needs
- Total storage requirements
- Typical object size and nature of access to these objects
- Durability requirements
- Latency requirements
- Maximum concurrent users to support
- Nature of queries
- Required strength of integrity controls

• Avoiding Single points of failure

- متخليش بتاعك ملهاوش backup في على سبيل المثال لما تعمل database خلي ليها replica في حته تانية عشان لو الـ db وقعت تبدل على النسخة الثانية.

• Optimizing for cost

- Are my resources the right size and type for the job?
- Which metrics should I monitor?
- How do I turn off resources that are not in use?
- How often will I need to use this resource?
- Can I replace any of my servers with managed services?

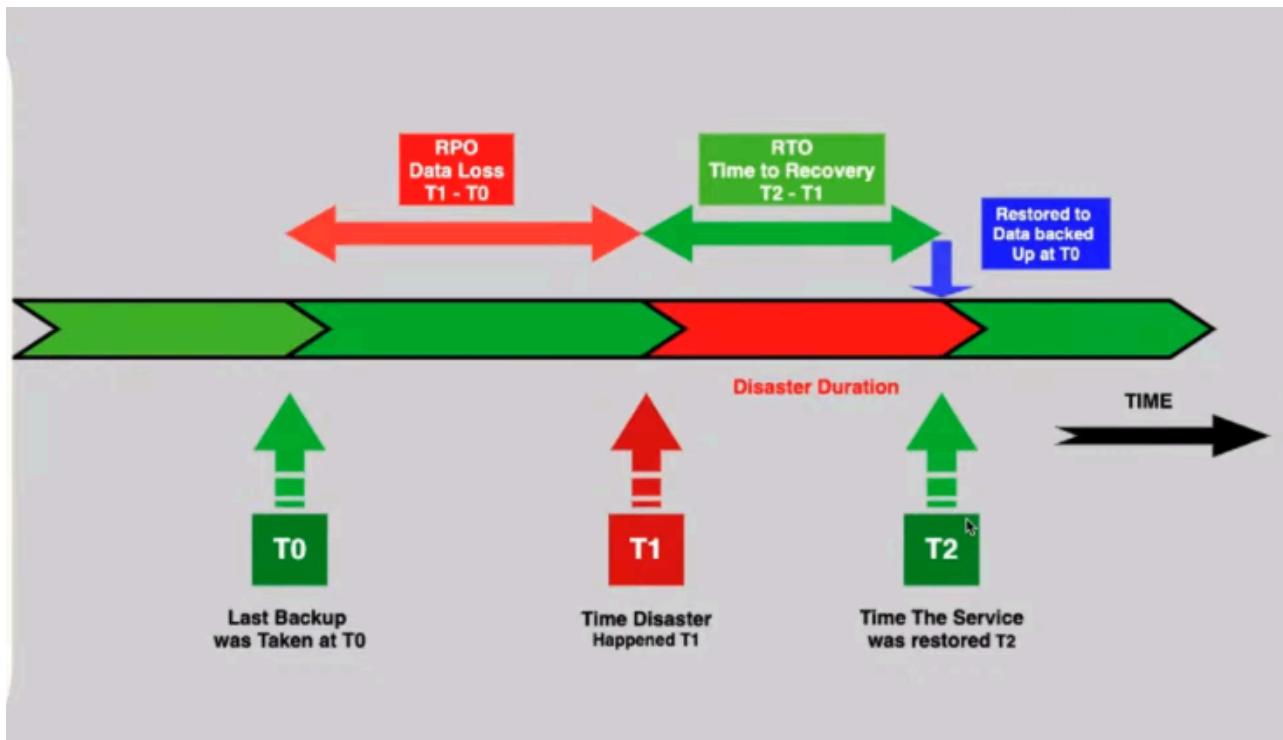
• Securing your entire infrastructure

- في كل System Layer في كل حط فيها Security وشد عليها جداً.
- أستعمل الـ AWS Managed Services لأن الـ AWS بتبقا على .resources لـ Log access.
- أتأكد انك فاصل اجزاء معينه من الـ infra بتاعتك عشان لو جزء اتعرض لـ attack الباقي بيقا في أمان.
- انك تـ encrypt الـ data سواء at rest او in transit .Principle of least privilege
- أستعمل MFA

Disaster Recovery (DR) & DR Approaches in AWS

What is Disaster Recovery?

- اولا Disaster event هي حصلت عملن مشكلة ووقفت أستمارارية الـ business.
- ثانيا Disaster Recovery هي خطة بتبقا ready to use عشان لو حصلت المشاكل بيقا في خطة عشان الـ business يـ استمر، ولازم أتأكد منها بـأني اعملها .test
- DR Strategy بـتعتمد على أني اعمل analysis عشان أعرف تـأثير الـ disaster على الـ business.



اي Strategy Plan لازم تعرفك الأنطين دول:

- ال (RTO) : الوقت الي هاخده عشان ارجع ال business process
- ال (RPO) : ال data loss المقبوله خلال ال disaster

AWS Economics and Billing

it's time to focus on the economic side of the cloud. This section will guide you through AWS's pricing models, cost optimization strategies, and billing practices, helping you manage your cloud expenses efficiently while making the most out of your AWS investments.

Fundamentals of pricing

ا AWS عندها Pricing Li Fundamentals و هما 3 حاجات:

1. Compute

.Windows with SQL و دا في Instance و Windows و Linux و حسب نوع ال Per Hour/Sec بتحاسب

2. Storage

Month Per GB في بتحاسب

3. Data transfer

عندی نوعين من data transfer و هما ال inbound(ingress) و outbound(egress)

ال AWS Inbound (البيانات اللي داخلة لـ)

• في الغالب مجاناً. أي بيانات بتجي من الإنترن特 أو من مكان تاني داخل AWS مش بتتكلف شيئاً.

ال Outbound إلى الإنترن特 (الإنترنت الخارجي)

• فيها تكلفة فعلاً، ومقسمة حسب المنطقة والحجم (tiered pricing)

AWS (Inter-Region) Outbound بين مناطق

- فيه تكلفة، غالباً بحسب حسب المنطقة المصدر فقط.
 - البيانات الداخلة للمناطق الثانية (destination) غالباً ما عليهاش تكلفة.
- مثلاً: لو نقلت بيانات من Region A إلى Region B، هيظهر عندك في الفاتورة سطر "RegionA-Out" والتكلفة بتحتس علىه، لكن بيظهر لكن بدون تكلفة "RegionB-In".

How do you pay for AWS

Pricing Models بتقدم 3 من AWS

1. **Pay for what you use**: بتدفع مقابل الاستخدام الفعلي، ودا يوفر agility scale demands.
2. **Pay less when you reserve**: بتتجز ال Instances بدل ما كل لما تحتاجها تطلبها ودا هيتوفر لي من 66% لـ 72% عن On-demand.
3. **Pay less by using more and AWS Growth**: تخفيضات بناءً على ال Business grows بتعاك و AWS Growth هي كمان.

AWS Pricing Calculator

ال AWS Pricing Calculator هو أداة مجانية على AWS بتخليلك تعمل Cost estimating بناءً على السيناريوهات الخاصة بك.

- ال Model Solutions : أعمل Model Estimation قبل تنفيذها.
- ال Explore Price Points : استكشف أسعار ال Service.
- ال Plan Spend : تحطيط النفقات.
- ال Find Savings Opportunities : اكتشاف فرص توفير ال Cost.

Exploring AWS Billing and Cost Management

Billing Dashboard

The screenshot shows the AWS Billing and Cost Management home page. The left sidebar includes links for Billing and Cost Management, Choose billing view (Primary view), Home (Getting Started, Billing and Payments, Bills, Payments, Credits, Purchase Orders), Cost and Usage Analysis (Cost Explorer, Cost Explorer Saved Reports, Cost Anomaly Detection, Free Tier, Data Exports, Customer Carbon Footprint Tool), and Cost Organization (Cost Categories, Cost Allocation Tags).

The main content area has several sections:

- Cost summary**: Shows Month-to-date cost (\$0.00), Last month's cost for same time period (\$1.62), Total forecasted cost for current month (\$0.11), and Last month's total cost (\$1.62). It also includes a "View bill" button.
- Cost monitor**: Displays Budgets status (OK, 1 active budget(s)), Cost anomalies status (MTD) (None detected, 1 monitor(s) active).
- Cost breakdown**: Allows grouping costs by Service, showing a bar chart with costs ranging from 0.00 to 2.00.
- Recommended actions (1)**: A callout box with "Getting started" and "Add an additional billing contact. Update billing contact."

- الـ **AWS Bills Page** : تعرض الـ Bill بقاعة كل شهر. لو الشهر لسه مش اتفعل، هتظهر التقديرات الحالية بناءً على الاستخدام اللي تم قياسه لحد دلوقتي

Cost Allocation Tags

The screenshot shows the 'Cost allocation tags' page in the AWS Billing and Cost Management console. On the left, there's a navigation sidebar with various links like 'Payments', 'Credits', 'Purchase Orders', etc., and a section for 'Cost and Usage Analysis' which includes 'Cost Explorer', 'Customer Carbon Footprint Tool', and 'Cost Organization'. Under 'Cost Organization', the 'Cost Allocation Tags' link is highlighted with a red box. The main content area has tabs for 'User-defined cost allocation tags' (which is selected) and 'AWS generated cost allocation tags'. Below these tabs is a search bar and a table with the following data:

Tag key	Status	Last updated date	Last used month
ASG	Inactive	-	December 2024
env	Inactive	-	November 2024
Enviroment	Inactive	-	November 2024
Environment	Inactive	-	February 2025
local	Inactive	-	November 2024
name	Inactive	-	October 2024
Name	Inactive	-	January 2025
Owner	Inactive	-	February 2025

- الـ **Tags for Cost Tracking** : بعمل TAG معين لكل group of resource, مثلا عندي Prod Env فا هحطها Tags أنها Prod وأبدأ أعرف Cost بـ Prod قد ايه شهريا.
- الـ **AWS Generated Tags** : بتتطبق آلياً على الـ Resources اللي بتتشهد، وبتبدأ بـ `aws` زي `aws:createdBy`.
- الـ **User-Defined Tags** : بتتحدد من المستخدم، وبتبدأ بـ `user`.

Cost and Usage Reports

The screenshot shows the 'Cost and Usage Reports' page in the AWS Billing and Cost Management console. On the left, there's a navigation sidebar with sections for 'Reservations', 'Preferences and Settings', and 'Legacy Pages'. The 'Cost and Usage Reports' link is highlighted with a red box. The main content area has a prominent message about the legacy page being deprecated: 'The Cost and Usage Reports legacy page will be deprecated. Data exports is the new experience to export your cost and usage reports and more. As a result, the Cost and Usage Reports legacy page is pending depreciation. You'll continue to have access to Cost and Usage Reports, the customer carbon footprint tool, and the AWS usage report from data exports.' Below this message, there's a 'Create report' button. To the right, there are three main sections: 'Cost and Usage Reports' (with a description of AWS Cost and Usage reports), 'Analyze your cost and usage' (with a description of AWS Cost Explorer), 'Monitor your Reserved Instance (RI)' (with a description of RI utilization), and 'AWS Usage Report' (with a description of dynamically generated AWS usage reports).

- الـ **Detailed Insights** : بتقدم لك بيانات شاملة عن التكاليف والاستخدام، بما في ذلك بيانات إضافية عن الخدمات، الأسعار، والاحتياطيات زي Reserved Instances.
- الـ **Integration Options** : ممكن تدمج التقارير دي مع أدوات زي QuickSight، Athena، Redshift عشان تعمل Analysis.

Cost Explorer

The screenshot shows the AWS Cost Explorer interface. On the left sidebar, under 'Cost and Usage Analysis', the 'Cost Explorer' option is highlighted with a red box. The main area displays a 'Cost and usage graph' showing monthly costs from August 2024 to January 2025. The total cost is \$4.90, average monthly cost is \$0.82, and service count is 19. The graph is a stacked bar chart with categories including VPC, Route 53, EC2-Other, Tax, Relational Database Service, Secrets Manager, DynamoDB, S3, EC2-Instances, and Others. Below the graph is a 'Cost and usage breakdown' section with a search bar and a 'Download as CSV' button.

- **الـ Visualize Costs** : كشف بـشكل graph عن التكاليف والاتجاهات بناءً على الاستخدام.
- **الـ Custom Reports** : بـتعمل Reports مخصصة عـشـانـت البيانات.
- **الـ Forecast Usage** : بـنتـوقـعـ الاستـخدـام لـمـدة تـصلـ لـ 12 شـهـرـ بنـاءـ عـلـىـ الـبـيـانـاتـ التـارـيـخـيةـ

AWS Budgets

The screenshot shows the AWS Budgets interface. On the left sidebar, under 'Budgets and Planning', the 'Budgets' option is selected. The main area displays a table titled 'Budgets (1)' with one entry: 'Billing Alert'. The table columns include Name, Thresholds, Budget, Amount used, Forecasted amount, Current vs. budgeted, and Forecast. The 'Billing Alert' row shows 'OK' for Thresholds, '\$1.00' for Budget, '\$0.00' for Amount used, '\$0.11' for Forecasted amount, '0.00%' for Current vs. budgeted, and '0.00%' for Forecast.

- **الـ Budget Creation** : اعمل Cost معـينـهـ مشـ عـاـيزـ عـديـهـ ولوـ عـديـهـ يـعـتـنـيـ alert.
- **الـ Budget Types** :

- Usage
- Cost
- Reservation
- Savings Plans

- **الـ Notifications** : تقدر تضـيفـ لـحدـ 5ـ تـنبـيهـاتـ SNSـ لـكلـ مـيزـانـيـةـ.
- **الـ Filters** : تقدر تـرـشـحـ حـسـبـ الخـدـمـةـ،ـ الحـسـابـ المـرـتـبـطـ،ـ التـسـمـيـاتـ،ـ نـوـعـ الشـرـاءـ،ـ النـوـعـ،ـ الـمـنـطـقـةـ،ـ وـغـيـرـهـاـ
- **الـ Pricing** : أولـ مـيزـانـيـةـ يـوـمـيـاـ two budgets freeـ،ـ وبـعـدـ كـدـهـ 0.02ـ دـولـارـ لـكـلـ مـيزـانـيـةـ يومـيـاـ

AWS Cost Anomaly Detection

- **الـ Machine Learning Monitoring** : استـخدمـ MLـ عـشـانـ تـراـقبـ الـCostـ وـالـاستـخدـامـ بـشـكـلـ مـسـتـمرـ.

- الـ **Anomaly Detection** : الـ Algo بتفهم انواع مختلفه من Historical Cost Data عشان تكشف أي Costs زيادات مؤقتة أو مستمرة

- الـ **Notifications** : اختر بين تنبيهات فردية أو ملخصات يومية/أسبوعية عبر SNS
-

AWS Service Quotas

- الـ **Quota Alerts** : AWS Service Quotas بتبلغك لما تكون قريب من حدود الـ Service Quotas Console على الـ CloudWatch Alarms
 - الـ **Example** : Lambda Concurrent Executions.
 - الـ **Quota Increase** : اطلب زيادة في الحدود أو أوقف الـ Resources قبل ما توصل للحد
-

Technical Support Plans

اـ Amazon Support بقدملك في حالة أن وجهاـك أي مشاكل في استعمال Services بتاعتـهم و فيه Support Plans مختلفة على حسب اـ حتياـجـك.

:Support Plans أنواع الـ

.1. الـ **Basic Support**

- دي الـ Default Accounts بنـاع كل customer service 24/7 لـي الـ access
- موجود الـ AWS Docs white papers
- الـ Personal Health dashboard
- ليـك Trusted Advisor لـي Limited Check

.2. الـ **Developer Support**

- كل حاجـه فيـ Basic Support
- تقدـرـ email customer support
- ردـ فيـ أقل من 12 ساعـة
- مفـيدـ ليـ Testing and staging

.3. الـ **Business Support**

- ليـك Trusted advisor لـي Full set
- ليـك Direct phone or web or chat
- لو عندـك مشـكلـه هـيرـدوا عـلـيكـ خـلـالـ 4 ساعـات
- ولو prod down هـيرـدوا عـلـيكـ فيـ أقل من ساعـة
- بتـبقـا Prod tier recommended فيـ حالة الـ

.4. الـ **Enterprise ON-Ramp**

- فيهاـ كل حاجـه من previous plans
- الـ business critical workloads هـيـقاـ أقل من 30 لو عندـك response
- ليـك technical account managers(TAMs) coordinate guidance

.5. الـ **Enterprise Support**

- فيهاـ كل حاجـه من previous plans
- الـ Response فيـ أقل من 15 دقـيقـة
- ليـك TAM مـخـصـصـ ليـك

AWS Global Infrastructure

let's explore how AWS's global infrastructure powers the cloud, providing scalable and reliable services to customers worldwide.

AWS Regions

وهي زي موقع جغرافي زي مثلا North Virginia بيكون فيه مجموعة من ال Availability Zones والي هنتشرح لاحقا .

- ال Characteristic
 - ال Region Minimum of three AZs في .Region
 - كل Region بي Communicate مع بقية Regions عن طريق AWS Network Backbone .
 - كل AZ بتبناها Independent من حيث كل شئ و توصيلين ب Ultra low latency networks .
- ال Region Selection Criteria
 - بختار ال compliance and regulations based on (قوانين امشي عليها)
 - بختار ال proximity region قریب من user (عنوان احسن Latency)
 - بختار علي حسب ال services availability (عنوان مش كل services مش بتتبناها موجوده علي regions الجديدة)
 - بختار علي حسب Cost (كل Region ليه cost مختلف).

Availability Zone

ال AZ هو مكان بحط فيها أكثر من Data Center عشان يكون Highly Available يعني معنديش Single Point of failure عشان لو وقع بيقا في حاجة تانية تعطي عليه Component

طلب لو حصل مشكلة في AZ دي نفسها؟ ساعتها AWS عملت أكثر من AZ وربطتهم بي High fiber networking عشان ينقل بسرعه عالية جداً (الAZ بتبناها أقل حاجة 3 في Region).

ال Recommended من AWS أنك وانت بتبني Service توزعها علي كذا AZ .

- ال Characteristic of AZ
 - بيبيها أكثر من DC منفردين
 - فيها Redundant power, Networking and connectivity
 - بنقسم ال Application في أكثر من AZ عشان ال isolate و احميها من أي Issues .
 - ال AZs دي Physically separated within 100kms

AWS Outposts

الOutposts هي Fully managed Solutions بمتطلبات AWS Datacenter لـ Racks بمتطلبات AWS. هي بمتطلبات Outposts بمتطلبات Power فقط.

هي بمتطلبات في أنني AWS Services على A Run لـ Extend. On-premises على ال Low latency access Support. محتاجها عشان ال Ultra low latency Apps التي بتحتاج.

AWS Local Zones

هي قريبه من Customer Zones عشان نقل ال Latency محتاجها عشان ال Apps التي بتحتاج. هي بمتطلبات AWS Local Zones لـ Region, زى ما أكلمنا عن infra extend لـ Outposts هي بمتطلبات AWS بس على .region بس هي بمتطلبات local or on-premises.

⚠ Warning

الفرق بينها وبين ال Outposts ان Outpost بيعت Rack مخصص ليك ولكن Local zone بيمتلكها AWS نفسها.

:Characterstic ال

- هي Extension of AWS Region
- أقدر ا Run عليها ال Latency sensitive applications
- أقدر منها أطلع على ال Direct connection support internet و بت
- ال User Resources بمتطلبات أقرب لل

AWS Wavelength

محتاجها عشان ال Ultra low latency Apps التي بتحتاج. AWS Infra هي موجودة في AWS Wavelength ال. Communication service Providers (CSP) 5G network هي موجودة في AWS Infra. لو انا عندي user وبيعملوا access من خلال 5G Network لـ infra موجوده جوا شركة الاتصالات بمتاعتي ال telecommunications network من App traffic الى بيبقا شغال على wavelength zone مش بيخرجني من wavelength zone فـ 5G Devices.

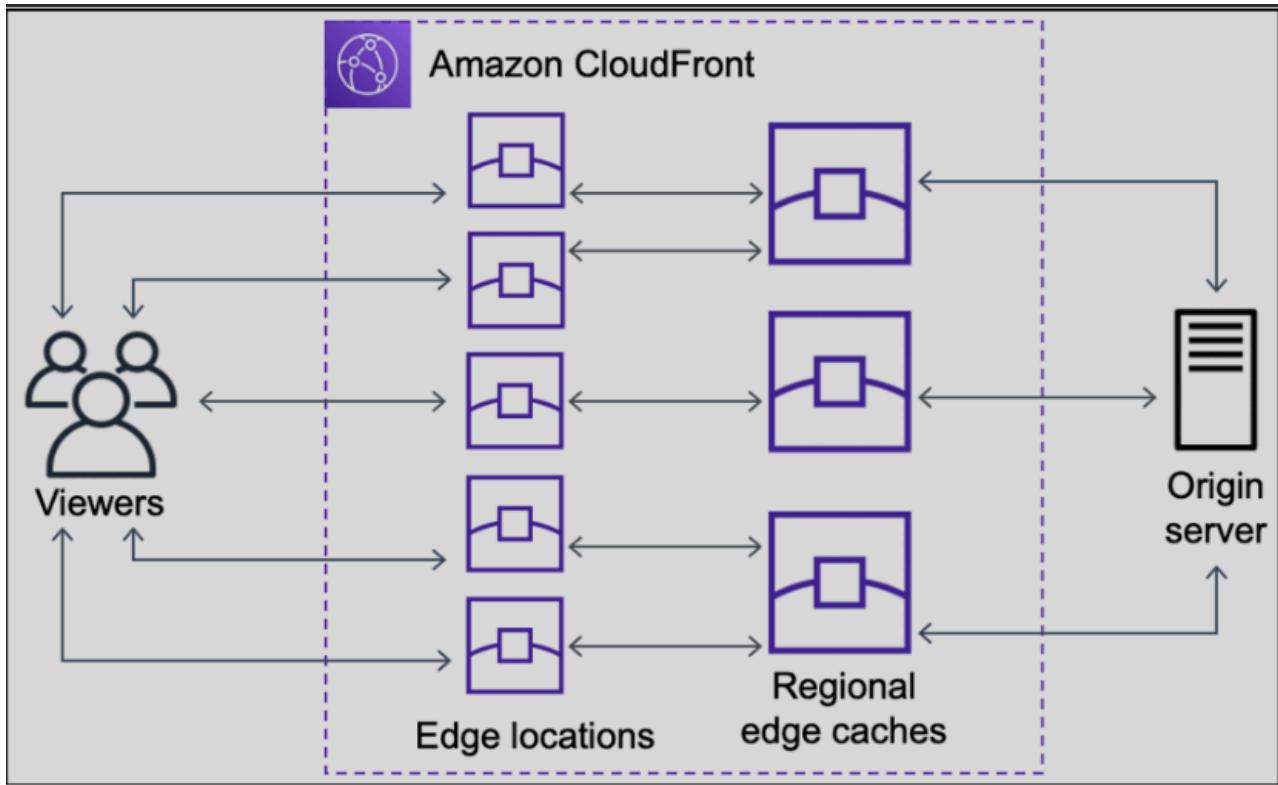
CloudFront

هي Content Delivery Networks(CDN) ولو هنختصر هي بتعمل ايه فـ Caching. AWS Points of Presence ودا هو ال Edge Locations التي موجود فيها caches بمتاعتي وتكزن فربية من Customer او User.

السيناريو الي بيحصل لما ال User يطلب Request Edge Location وبعدين ي Check ال Caches عنده عن محتوي Request ملقوش بيقا دا اسمه Cache Miss وبعدين بيعتوا لي مكان ال Storage الخاص بي الـ Request يمشي في AWS Backbone Network وبعد ما يروح لي Edge Location هيرجع لي .miss بقى بتعته وبعدين أرجع لي User عشان لو أطلب من أي حد تاني نفس Request ساعتها يعمل Cache hit بدل ما بيقا

Edge Locations

- هي عبارة عن datacenters موجودة في كل مكان و بت serve as endpoint يعني نقطة وصول لي AWS Services، بس هي معمولة مخصوص عشان ال caching و content delivery .
- و دي Service من AWS الي كانت مشروحة و اسمها Cloudfront و بتقدم .content delivery network(CDN)
- الغرض منها:
- أنها تكون فيه static and dynamic content cache copies زي web pages و videos
- .retrieve most used ودا بيقل وقت ال .retrive يكونوا



ف عبارة عن CloudFront و Edge locations و Regional edge caches ف ايه الي بيحصل عشان عشان ال Content يوصل لي User ؟

1. Edge location معين و ال Request دي بتروح لأقرب Viewers.
2. الـ Request يبيشوف الـ Content موجود ولا لا، لو موجود هيتعت Edge location لي viewers ولو لا بيروح يعملRegional edge cache لي.
3. الـ Regional edge cache بتكون قريبة من Edge location و بتتحفظ بالـ content الي طلب عليه بيقا قليل، لو المحتوى مش موجود فيها بتتعت Origin server request لي.

ال Characteristic

- الـ Caching Speeds up distribution عن طريق الـ .
- الـ Request بتـ route لي closest edge location والـ i هو route لـ i.
- الـ AWS Backbone network Routing each user.
- لو Content موجود الـ Cloudfront بتـ retrieves من الـ Origin immediately (Retrieves).
- لو Content عندي ساعتها هيرجعها (Retrieves).
- بسـ t عمل Encryption Perfect Forward Secrecy(PFS) ودا بيـ t عمل configuartion Up to 25 origins per distribution.
- أقدر يكون عندي distribution معناها وحدة الـ .

- أقدر اعمل Origin Group ويكون فيه Primary origin ودا الي شغال فيه ال Application بتاعي و تاني Failover لـه ويحول عليه لو ال Primary وقع.
- تقدر تعمل Time To Live(TTL) وهو أنه يستني قد ايه على cached objects قبل ما يحدثها من ال Origin لو 0
- معناها اي Request هيروج ال CDN هيروح يشوف Origin يتأكد ال Cache انه updated Cache
- ال Cache Behavior وهو ازاي هي Request لـي Path pattern يعني يوصل ال Path لـي Request الصح.
- في حاجه اسمها Origin Access Identity(OAI) وهي بشيل ال Origin Access Identity من على S3(هنجلها بعدين وه يكون مشروح فيها الموضوع S3) ونخليها تنت access عن طريق CloudFront , ودا بيكون Special IAM Policy بتحط في S3 Bucket Policies

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForCloudFrontPrivateContent",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::cloudfont:user/CloudFront Origin Access Identity EH1HDMB1FH2TC"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::originserverbucket/*"
    }
  ]
}
```


This OAI is provided when
The OAI is created in
The CloudFront Web
Distribution Configuration

CloudFront - Geo Restrictions (Blocking or Filtering)

ال Geo Restriction feature بحصر دخول موقع على فئه معينه سواء بعمل Whitelist or blacklist

CloudFront - Monitoring and Access Logs

بقدر ليها علي كل CloudFrontLogs بتروح علي S3 تكون انا مختارها.

CloudFront - Price Classes

كل ال بتكون معايا Edge Location فا بتحتاج تعرف معظم User بتوعك فين عشان تحدد تعمل Select لجزء من Edge Location ومدفعش علي كله.

AWS Route 53

ال Route 53 الفكرة هو Fully managed DNS Server

DNS هو اختصار لي Domain Name Server/System, لما بكتب في Browser واكتب اسم الموقع الأسم بيروح لي DNS Server ويعين هو بيعتزا لي authoritative server ويرد عليا بي IP Address اللي محتاجة. فال Route 53 هو AWS authoritative server بتاع.

بيقدر يعمل حاجه اسمها Global Server Loadbalancing او VSLP او traffic request او traffic بيوديه لأقرب app بتاعي موجود فيها

Route 53 - Supported DNS Record Types

ال A Record : ودا بسيط بيقيا عندي Domain ربطه بي IP

ال AAAA Record : ودا هو A Record وكل بيكون IPV6

ال CNAME Record : ودا بي ال Sub domains على Domain واحد بيكون URL لـي URL

ال NS Record : ودا ال nameservers الي فيهem ال records بتاعتك لو واحد وقع يكون في سيرفرات تانيه تعرف ال Records بتاعتك AWS: دا Record internal مش بيتشاف برا AWS, متقدرش تعمله create لـ Alias Records. تقدر تستعمله انه ي route DNS Queries لـ AWS Services وهم:

- ال ELB
- ال CloudFront
- ال S3 Static Websites
- ال Elastic Beanstalk
- ال API Gateway
- ال VPC Endpoint
- ال Global Accelerator
- و أي existing record في نفس ال Route 53 hosted zone
- ال Zone Apex على Alias دى

Info

.Subdomain هو نفس اسم ال Domain من غير اي Zone Apex

يعني:

- ال Zone Apex ← ده اسمه example.com
- (subdomain ده مش Zone Apex) ← ده مش www.example.com

ال Charge بـ CNAME اما Free بتكون Alias Records مع Route 53 Queries

Characteristic

- هدفه أنه ي Route Internet traffic لـ resources بتاعت ال Domain بتاعي.
- بيعمل Check على ال Health بتاع ال Resources بتاعتني وساعتها لو تمام هيبيعت ال IP بتاعها
- أقدر اعمل Register Domain Name أعمله
- أقدر ابعت ال User لـ Application بتاعي على أقرب location ليه
- يقدر يحدد علي أساس ال latency الأسرع ليه
- بيعمل Checks health لـ Relevant resources

Route 53 - Routing Policies

ال Simple Routing Policy: ودا بسيط لو عندك Resource واحد تستعمله. مش بيدعم ال Health Checks

ال Failover Routing Policy: ودا من اسمه بيعمل failover لـ active Resource تكون قليله, بيعتبر active/active وبيدعم .checks

ال Latency-based Routing Policy: لما يكون عند multiple sources عايز تتأكد ان ال user يروح لأقرب مكان ليه عشان latency .health checks

ال Weighted Routing Policies: لما يكون عندي multiple resources وبكون عايز احدد weight معين يخش في كل active/active failover policy مثلـ resource one 70% في و resource two 30% في

.health checks يدعم ال

• ال Use Cases لـ Load Balancing and testing new releases of app لـ

ال Geo-Location Routing Policies بتـ traffic على حسب geolocation بتـ user, مفید لو في compliance و active/active data distribution rights خاصه بـ دولة معينه, ويـعتبر

ال Geo-Proximity Routing Policies بتـ traffic على أساس Location بتـ user و resource, وتـقدر تـعمل shift resources ثانية في مكان معينه لي resources لـ

ال Multi Value Answer Routing Policy دا نـسخـه المـحسـنـه من simple routing, ولكن هو بي support health checks

Route 53 - Resolver

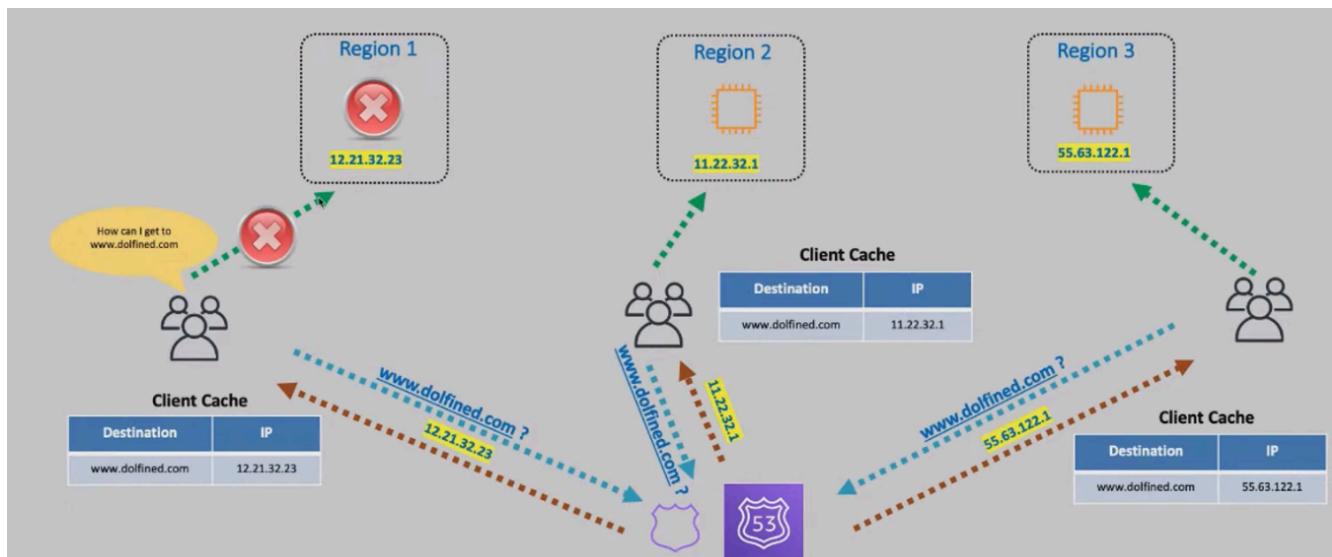
لو ال Application بتـ انتـي بتـكلـم DB عن طـريق Resolver هو ال IP Hostname فـا مش IP اللي بـيفـكـها ويـحـولـها لـ, بيـكون Intra-VPC DNS Queries resolve لـ VPC عـشـانـي by default في

وـتنـفـعـ بـرـدوـ لـ Hybrid Cloud Environment لـ DNS Queries جـيـ علىـ الـ On-premisesـ Resـolveـ بـدـلـ ماـ Resـolveـ فيهـ لـاهـيفـكـهاـ عنـ طـريقـ الـ Route 53ـ Resـolverـ انهـ بـيرـميـ ENIـ فـيـ VPCـ بـتـاعـتكـ علىـ AWSـ فـاـ تـتحـسـبـ انـهـاـ تـبعـ Resourcesـ وـيـعـملـهاـ

خدـ بالـكـ عـشـانـ يـعـملـ Resourceـ اـنـتـ هـتـتـحـاجـ تـعـملـ Direct Connect Connectionـ اوـ VPNـ تـرـبـطـهـاـ بـيـ Inbound endpointـ عـشـانـ .destinationـ وـهـتـتـحـاجـ on-premisesـ عـشـانـ outbound endpointـ يـوصـلـ لـ destinationـ

AWS Global Accelerator

Without Global Accelerator - The Problem



عـنـدـ 3ـ EC2ـ وـكـلـ وـاحـدـهـ فـيـ Regionـ يـدـعـهـ Route 53ـ لـ distributeـ trafficـ حـسـبـ latencyـ بتـاعتـ userـ مـثـلاـ (بسـتعـملـ Route 53ـ لـ Latency Policyـ كـمـثـلـ) فـاـ كلـ وـاحـدـ منـ Userـ بـيـاخـدـ الـ IPـ بتـاعـ EC2ـ وـيـروحـ عـلـيـهاـ Cacheـ عنـدهـ عـلـيـ (Browserـ, فالـ لوـ EC2ـ وـقـعـتـ وـحـاـلـ يـخـشـ عـلـيـهاـ مـشـ هـيـعـجـهـ لـأنـهـ مـشـ بـيـنـقـلـهـ عـلـيـ التـائـيـةـ هوـ بـيـسـيـهـ معـ أـقـرـبـ EC2ـ لـيهـ عـشـانـ Cacheـ فيـ) TTLـ معـيـنـ مـمـكـنـ دـقـايـقـ اوـ سـاعـاتـ وـالـ Userـ مـشـ هـيـعـجـهـ كـلامـ دـاـ لـوـ هـيـخـشـ عـلـيـ Websiteـ هوـ عـاـيـزـهـ يـكـونـ عـلـيـ طـولـ متـاحـ, وـديـ باـضـبـطـ نقطـةـ الـضـعـفـ الأسـاسـيـهـ فيـ DNS-based routingـ

فاـ عندـكـ حلـ انـكـ تـعـملـ ELBـ وـ الـ Route 53ـ يـبـعـتـ trafficـ لـ EC2ـ وـلكـنـ دـاـ حلـ Regionalـ يـعـنيـ مـصـوـصـ لـكـلـ Regionـ تـعـملـهاـ .Global Acceleratorـ فـاـ الحلـ انـكـ تـسـعـملـ

Global Accelerator - What is it?

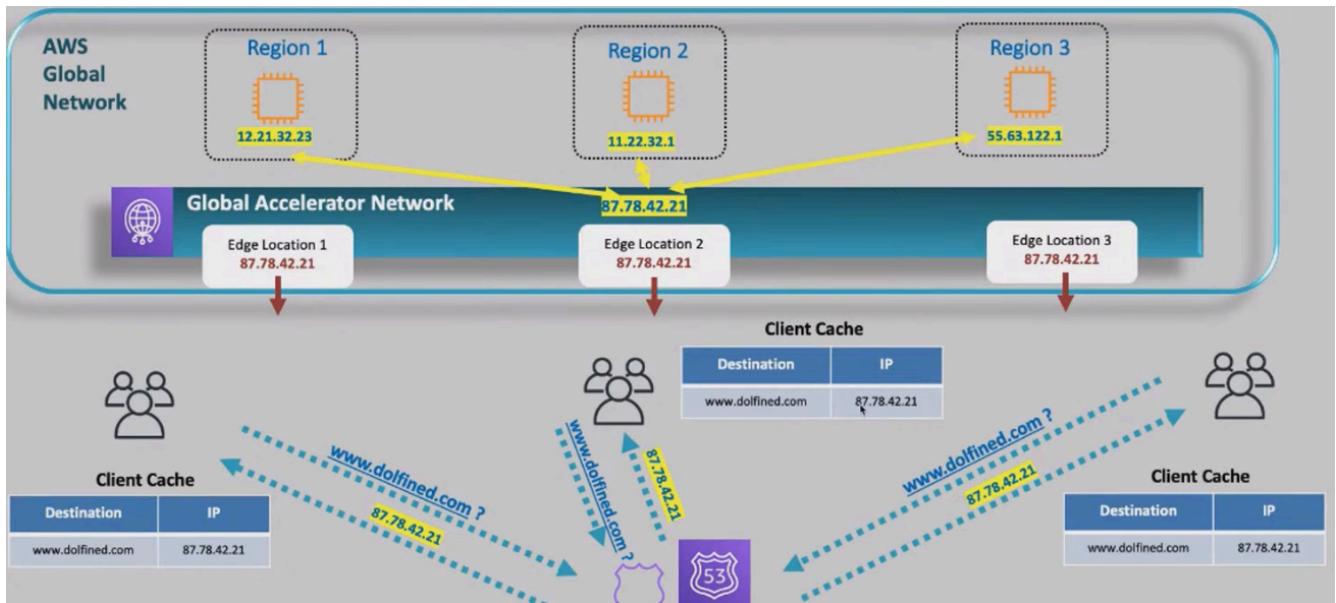
هي عبارة عن Network Layer, بتكون Network Layer لـ Fronting Layer,Highly available و distributed applications يعني هي الـ traffic وتبدأ توزعه وهي توزعه بنكاء يعني هي هتخفي الـ IP Address الى IP التي تكون هو الـ IP الى مع كل وبالتالي لو اتعلمه cache مش مشكلة لأنه مش هيقع وه يكون Static. بيعمل كل دا عن طريق Anycast IP يعني هيوديك لأقرب Network Fundamentals (User مسروح في_endpoint بالنسبة لـ)

والـ Accelerator بتستعمل Edge Location وكل Traffic لي Route لي Two Static IPs بيطلع Accelerator Health and Performance.

Note

لما تعمل AWS بيديك Global Accelerator، Static Anycast IPs 2 معناها: نفس الـ IP بيعلن (advertised) من كذا مكان (كل Edge Locations). الكلمة Anycast هيوجه لأقرب request على الـ IP ده، الـ routing ليه (حسب الـ IP في الإنترن特) النتيجة؟ أي يوزر يعمل على الـ IP ده، الـ request هيتوجه لأقرب Edge Location.

With Global Accelerator - The Solution



فا الحل للمشكلة انك تحط Accelerator لكـ Region فيه الـ EC2 والـ IP واحد في Static IP هيديك (الـ IP واحد بـ IP). ومنها هو الـ Route 53 ويعدين تحطه في الـ Traffic distribute لـ Traffic. وبـ الـ Traffic موجود في الـ Client Cache هو الـ IP Accelerator وهو الـ IP يوزع بعد كـ الـ Health Check على الـ User. الـ Accelerator هو الـ IP بتـ الـ IP بتـ الـ User. بتـ الـ IP بتـ الـ User بتـ الـ Website بتـ الـ User.

فـ الـ Global Accelerator من الـ Enhanced fault tolerance يعني لو حاجه وقعت هيخش على الـ User والـ User مش هيس بـ Delay.

- أنـ عنده Enhanced fault tolerance يعني لو حاجه وقعت هيخش على الـ User والـ User مش هيس بـ Delay.
- الـ UDP and TCP مدعاومين فيه.
- الـ Endpoint Health checks وـ الـ Dead Endpoint يعرف الـ User عليها ومـ الـ User مـ الـ User.
- ـ الـ Backbone traffic علىـ Better Performance وـ الـ AWS Shield سـ الـ AWS Shield.
- ـ الـ DDoS Protection وـ الـ AWS Shield منـ Integrate.
- ـ الـ Client(Source) بتـ الـ IP بتـ الـ Client(Source).

Use Cases

Applications that require whitelisting of a small number of IPs:

- Autonomous vehicles.
- Payment/retail transactions.
- Healthcare.
- IoT.

Multi-region applications:

- Financial services.
- DR/Failover scenarios.

UDP traffic applications

- Gaming.
- Voice over IP.
- DNS.

Live Video Ingestion for media applications:

- Latency sensitive applications.

Global Accelerator Vs CloudFront

CloudFront

- Uses the AWS global network and its edge locations.
- Improves performance for static and dynamic content through **caching**.
- Is primarily for HTTP and HTTPS traffic.
- Integrates with AWS Shield for DDoS protection (explained later).

Global Accelerator

- Uses the AWS global network and its edge locations.
- Improves performance and availability of **internet facing applications**.
- Can be used for TCP and UDP (HTTP and non-HTTP traffic).
- Integrates with AWS Shield for DDoS protection (explained later).

الـ Global Accelerator يستهدف الـ application layer وـ Cloudfront يستهدف الـ network layer

Deep Dive AWS Global Infrastructure

AWS Global Infrastructure Map

The AWS Cloud spans 114 Availability Zones within 36 geographic regions, with announced plans for 12 more Availability Zones and four more AWS Regions in New Zealand, the Kingdom of Saudi Arabia, Taiwan, and the AWS European Sovereign Cloud.



AWS Regions & AZs Selection

• Selecting a region:

- بنتشوف أقرب Region لي Users عشان تأخذ Lowest Network latency وتحسن ال experience الي user بيكون متوقعها منك.
- ال Pricing يختلف من Region لـ الثانية، بتحاول تختار ال Region الي مناسبة لي Budget.
- ال Compilance، طبعا كل شركة وبالذات البنوك بيفقا عندهم قوانين معينة مفروضه عليهم تتبع الدولة مثل او الحاجات الحكومية فـ لازم تتأكد ان ال Region دي في منطقة آمنة بالنسبة لـ دولة الي انت فيها او هي في دولة الي انت فيها اصلا.
- لازم تتأكد ان Service الي عايزة من AWS موجودة في Region دا، لأن مش كل Services متاحه في اي Region.

• Selecting Availability Zones:

- احنا عارفين ان كل AZ فيها اكتر من Data centers، فـ هي Fault isolation لـ Designed وـ فيها Interconnected مع ال AZs الثانيين بي Clients Disaster لو حصل أي High speed private links ميتاثروش.
- اـ AWS بترشحلك اـ انك ت App Replicate بـ AZs مختلفه عشان تحقق ال HA وـ resiliency.

Availability, Naming & Edge Locations

• Availability:

- ال AWS في معناه قدرت High availability System أنه يفضل Accessible وـ minimal Operational accessible مع :Mechanisms downtime.
- ال AZs مـ Multiple AZs.
- ال Load balancing: وهي أني اوزع traffic على اكتر من AZs فـ لو حصل failure في واحده الباقي هيكمـ.
- ال Auto Scaling: لو workload زاد او بـقا في unhealthy machine هـيغيرها او هـيزود غيرها.
- ال Fault tolerance: S3, RDS, and dynamodb services بـ built in زـي Downtime.
- مع كل ال Mechanisms دي هـقدر أخلي ال Downtime قـليل اوـي.

• Naming:

- بيكون كـاـk مـثال --> us-east-2a وـدا منقسم لـ 3 أجزاء:
- أول جــزء وهو us وـدا ال area الي موجودـة فيه

- ثاني جزء وهو `us-east` مع بعض ودا ال Region name
- ثالث جزء وهو `us-east-2a` مع بعض كله ودا ال AZ name

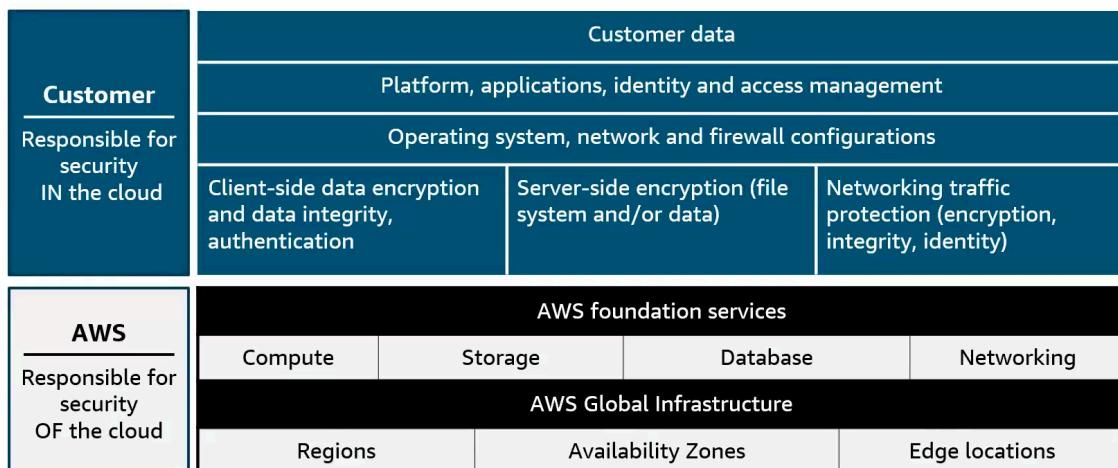
- Edge Locations

AWS Security

Now that you understand [the basics of security](#), let's explore how AWS keeps your cloud environment secure using various security services and best practices.

AWS Shared Responsibility Model

AWS shared responsibility model



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8

الـ AWS Shared responsibility model بيقولك ايه الي عليك تعمله في Cloud وأيه الي علي الـ Cloud provider

الي عليك تعمله:

الـ Customer data: يعني تحتاج تحمي data الـ client بتاعك في كل احولها زي في حالة `data in transit` تتأكد ان بياناته وهي بتتنقل محميه زي عن طريق الـ HTTPS/SSL Certificate او في حالة `data at rest` وهي بياناته عندك في database مثلًا انك تتأكد انه `database queries` ومحميه ومحدش يعرف يوصلها, او في حالة `data in use` هي اني data بستعملها لغرض معين زي `encrypted real-time process` وهكذا.

الـ platform, app : بتحكم في مين بيقا عنده قدره يـ access لي VMs و Platform بشكل عام.

الـ Firewall و Security : بتحكم ببردو في user access بتاع port زي ليه معينه ومين يخش ومين لا علي سبيل عايز اـ ban منقطه كامله وهكذا.

الـ AWS Responsibility: هي مسؤولة عن أي حاجه Physical, وان الخدمه تكون متاحه 24/7

الـ IaaS: الـ Customer هو الي مسؤول عن configuring Network و Security و access control, الباقى علي AWS.

الـ PaaS: الـ Customer هو الي مسؤول عن Code و data, الباقى علي AWS.

ال SaaS Customer بيعملها فقط مش مسؤول عن أي حاجة.

Design Principles for The Security Pillar

1. ال Least privilege تطبق مفهوم ال Security Implement a strong identity foundation و هو أني لى ال Team member مثلًا أقل صالحيات هو محتاجها عشان يعمل الي عايز يعملوا ودا بيساعد في أنه يقلل مخاطر ان حد يعمل حاجة بصلاحية او permission المفروض متكونش معاه.
2. ال Encryption , tokenization Protect data in transit and at rest .and access control
3. ال Security at all layers زى ما هنأخذ بعدين انت تقدر تعمل لكل Layer عندك خاص وتسمح مين يخش ومين لا فا حاول تطبق ال Security فىهم كلهم.
4. ال mechanics Keep People away from data مع ال Users او People عامة.
5. ال AWS is audit Maintain traceability عشان اعرف اعمل ايه (who do what). وعشان اي حاجة علي API فا دا بيخليني اقدر اعمل Audit على Service ومين استعملها وكذا.
6. ال testing Prepare for security events على ال App بناعي في حالة ال عشان Attacks اشوف ال App و Security تمام ولا محتاجه restrictions اكتر.
7. ال Automation Automate Security best practices فا حاول ت Automate اي حاجة علي قد متقدر.

Amazon Resource Name(ARN)

دا ال resources لأي عنده ، بيكون بشكل دا:

```
arn:partition:service:region:account:resource
```

Examples:

- An ARN for an AWS Account has the syntax - > **arn:aws:iam::account-id:root**
 - An ARN for an IAM user -> **arn:aws:iam::account-id:user/user-name-with-path**
 - An ARN for an IAM Group -> **arn:aws:iam::account-id:group/group-name-with-path**
 - An ARN for an IAM Role -> **arn:aws:iam::account-id:role/role-name-with-path**
- We can use * as a wildcard mask for example :user/* in a user ARN means all users

دي شوية عن arn عشان تفهمها اكتر Examples

AWS IAM

What should i know before IAM

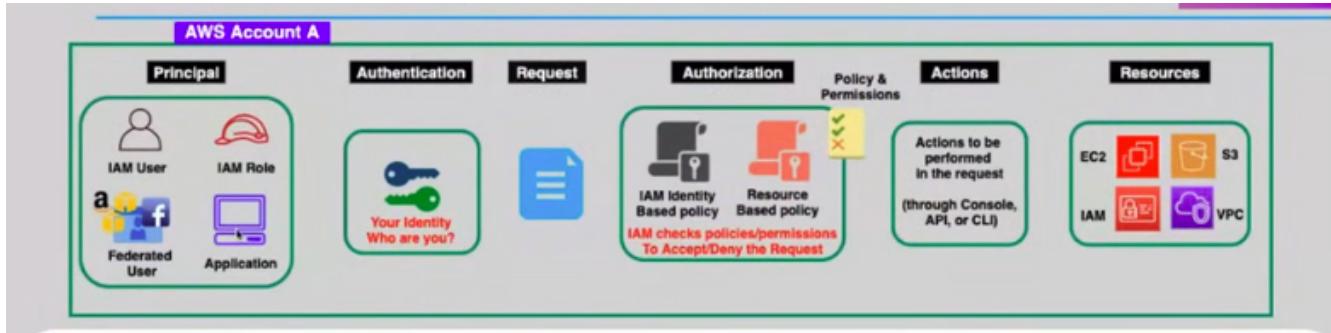
ال Authentication هي هل الشخص دا عنده بيانات عندي ولا لا؟(انا مين)

ال Authorization: طب ايه صالحيات الي يقدر يعمله الشخص دا.(انا مسموطي أعمل ايه؟)

الي بيحصل لما Client بيحاول يخش علي Site معين أنه بيعت ال Authentication بباتعاته عشان يثبت أنه ليه صلاحية، بعدين لو هي صح ال او Service provider بيرد عليه بي Authorized يعني ال privilege الى على Site مطبوبة، الحته دي اسمها Identity Provider(IDP) & Service Provider(SP)

ال IDP وظفته يعمل عملية ال Authentication
ال SP وظفته أنه يعمل ال Authorization

عادة بيبقو مرتبطين بعض بس في cases تانية بيحصل Separation/Federation و بتسمى Federated Auth/Users/Sites. مثل ساعات Trusted Login as Google او Linkedin او SP يبعلوه أنه Federated Auth عندهم فا تسجل على الموقع عادي وهو دا ال.



What is the Identity and Access Management(IAM)

هي service بتساعدك أذك الفرد او جروب يكونوا عندهم control over AWS Resources حسب permissions .Regions على مستوى كل Global Service بتتفا.

IAM Features

- Multi-Factor Authentication(MFA)**
 - الـ MFA هي عبارة عن Extra Security Layer بتنقل من أحتمال unauthorized access .
 - عبارة عن code مربوط بي الـ Account مش هينفع ت Login غير و Code معاك.
- Identity Federation**
- Identify information logs for audit and compliance purposes**
 - بيبيقا في logs على كل تحركات الـ users عشان أعرف لو حصلت مشكلة بسبب مين
- Integrated with many AWS services**
 - بتقدر تستعملها مع Services عشان تعرف تخلي Services يكلموا بعض.
- AWS STS (Security Token Service)**
 - الـ AWS Security Token Service (STS) بتقدم temporary, Limited- privilege Service او لو بتتكلم مع حد وذكرهم تكون فاهمهم: Expiring time لـ users وببيقا ليها credentials

IAM Terminology

ولكن مع IAM هنحتاج نفهم شوية Terminology عشان تتفا فاهم Docs او لو بتتكلم مع حد وذكرهم تكون فاهمهم:

1. IAM Resource

- الـ Term دا مش مقصود بي AWS Resources عامة لا مقصود بي الـ Service اي جوا IAM نفسها كا Resources و تقدر تتحكم فيها كـ API Target يعني تقدر تعمل iam:GetRole , iam:DeleteUser , الخ.

2. IAM Entity

- دي الـ Role او Users الي بيقدروا يـ login او يعملوا Assume Role يعني اصح الي يقدر يعمل Authentication

3. IAM Identity

- دول اللي تقدر تديهم Permissions عشان يتحكموا في AWS resources
- نشمل: Group , User , Role

4. Principal

- ده الشخص او الـ Service اللي بيعت AWS Request
- سواء كان:
- الـ User داخل manually

- Lambda function
- EC2 instance شغالة برول

IAM Policy

ال IAM Policy هي اللي بتحدد ال Permissions، وبنحطها عادة على Identity (User, Role, Group)، أو ممكن تنت attach .resource-based policies في حالة S3 Bucket Resource زي

- ليها Three Types
- AWS Managed permissions: بتكون معموله على الاكونت من AWS
- .customize permissions: بتقا inline policies
- والدي بتحكم فيها بنفسك: Customer Managed Policies

هي عباره عن document مكتوب فيها ايه ال Allows or Denies permissions .JSON و مكتوبه بال .

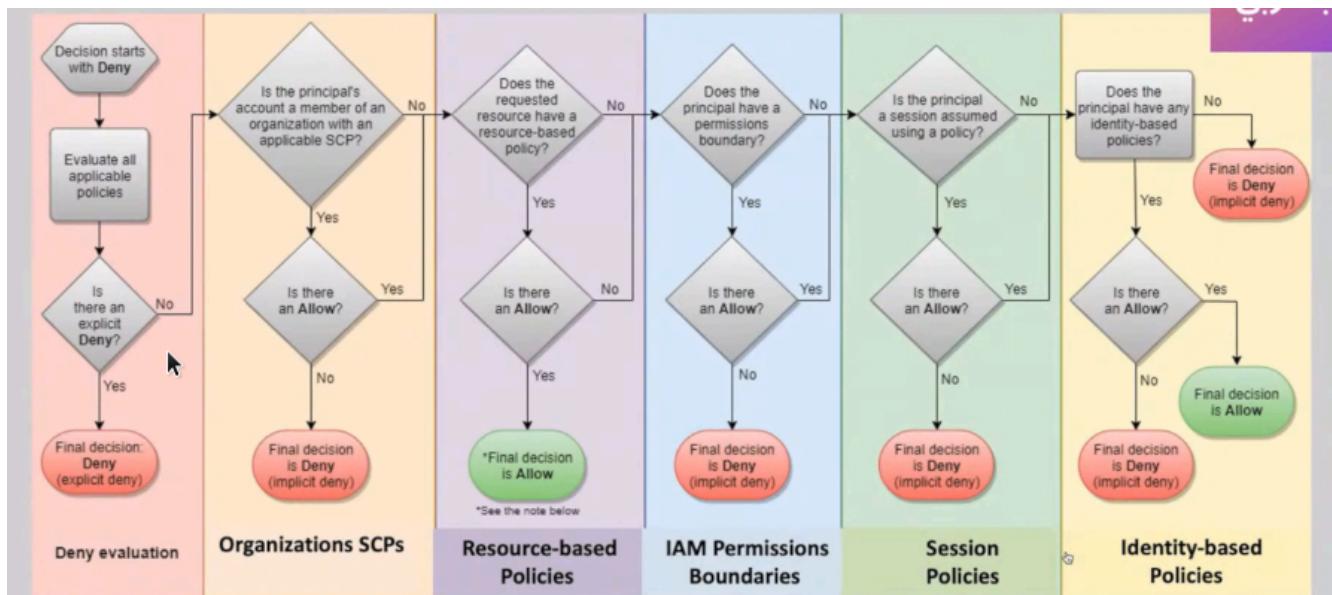
كل حاجه By default بتقا implicit deny (دا اسمه deny) لو حطيت بنفسك ال هيكون explicit deny.

عندى نوعين من Policies الفرق ما بينهم هو فين بيتعلمهـ Attach وهـ محتاج أكتب الـ Principal ولا لا؟:

- الـ Identity-based by default: بعمل attach لـ Principal او user او role او group، وجواها مش بتحتاج تحدد الـ attach لأنـه يكون implicitly معروف لي شخص الي بيتعلمواـ attach
- الـ Resource-based: بعمل attach لـ Principal، اما هنا فا لازم تحدد الـ resource عشان يقول مين الي هـ يستعمل resources دـي.

IAM Permissions Boundary

ال IAM Permissions Boundary هي طريقة تحط فيها Identity limit اللي يقدر يستخدمها، حتى لو عنده IAM Policy بتسمح بأكتر من كده. هي مش بتدي صلاحيات، لا دي بتقيـد الموجـود فعلـاً.



دا ال Policy evaluation logic وهي بتتكلم عن ترتيب اللي هي execute بيهـ الـ Policy

IAM User

ال IAM User هي زي Person account لي التيم يعني بيكون ملك لشخص واحد فقط ويكون authenticate مع AWS . ومن الـ Best practice أنني أدي لكل user بيفا معاه IAM User عشان أعرف اراقب او اعمل auditing على الـ User.

- Type of access
 - Programmatic: access via Access key id and Secret access key (Provides AWS CLI and AWS SDK access)
 - Aws Management Console: access via Account ID, Username, and Password
-

IAM Group

ال IAM Group هي Group من ال IAM Users عندهم نفس ال IAM Policy أو نفس ال Authorization بمعنى أصح.

✓ Best Practice

من الـ Best practice أنك تحط ال Policies بتاعت مثلا ال IT في Group وأي حد في IT يتحط في Group ومنها بدل ما كل لسه هنختارله Policies اللي تحتاجها User.

IAM Role

ال IAM Role دي أني أعمل temporary permissions لي ال service او User او Application و بتبقا معاه بشكل مؤقت، ممكن تعتبرها زي مندوب (Delegate) :

مش موجود دايماً، بس وقت ما حد يحتاجه بيروح ياخذ إذن (AssumeRole) من ال STS عشان ي Generate临时 token . وقد استعمل ال temporary Permissions بشكل ودي ويدأ ينفذ بيها المطلوب منه.

تقى تستخدمها مع User زى مقولنا بس ازاي؟ ، اول حاجه ه create ال IAM Role واحظ ال Policy المطلوبة ، وبعدين اعمل Temporaray Permissions واربطها بي IAM User بتاعه وبكدا هبيقا معاه AssumeRole .

ممكن استعملها فى Cross-account access for IAM User .

Security Services

AWS Organizations

ال AWS Organizations هي Account management بتجمع فيه حسابات AWS في Organization واحدة.

- الـ Centralized Management : الحساب الرئيسي (Master Account) بيكون هو اللي بيدير كل الحسابات الثانية.
 - Cost Benefits : طريقة دفع واحدة لكل الحسابات.
 - Consolidated Billing : تخفيضات على الخدمات زى EC2 و S3 لما تجمع الاستخدام.
 - Volume Discounts : توفير أكبر باستخدام Reserved Instances Reserved Instance Pooling مشترك.
 - الـ API Automation : أتمتة إنشاء الحسابات باستخدام API.
-

Organizations Components

1. The Root
2. The Organizational Units(OUs)
3. Accounts

Organizational Units (OUs)

ده بيخلي **Group** واحد مع بعض في **Department** هي طريقة لتجميع الحسابات تحت **Organizational Units (OUs)**.
الـ **streamlined administration** أسهل و **Policies** easier to manage.

الـ **Policies** يتكون على مستوى OUs او Accounts عادي

Service Control Policies (SCPs)

الـ **SCPs** هي يستخدم عشان تحدد (Whitelist/Blacklist) الإجراءات (Actions) اللي مسموح فيها أو مش مسموح فيها، ويوزع فيها Policies من حيث أن OU اللي فيها ACCOUNTS دي هيتعلها inherit policies لي بتاعتتها.

- الـ **Application Level** : يطبق على مستوى OUs أو الـ Account، ما عدا .Master Account
- الـ **Impact** : يتأثر على كل المستخدمين والـ Roles، حتى الـ User.
- الـ **exceptions** : مش يتأثر على الـ Roles المرتبطة بـ .(Service-Linked Roles)
- الـ **Explicit Allow Rules** : لازم يكون فيه قواعد "Allow" واضحة.
- الـ **Use Cases** : تحديد الوصول لـ Service معينة أو فرض معايير مثل PCI Compliance

Consolidated Billing in AWS Organizations

لما تشغله AWS Organizations، هتحصل على ميزة **Consolidated Billing**

- الـ **Aggregated Usage** : تجميع الاستخدام عبر كل الـ Accounts عشان تستفيد من .(Volume Discounts, Savings Plans, Instances).
- الـ **Single Invoice** : فاتورة واحدة لكل الـ Accounts.
- الـ **Discount Control** : يقدر يتحكم في Management Account الخصومات في Reserved Instances Sharing.

AWS Control Tower

الـ **AWS Control Tower** هو حل مبسط عشان تعمل Create و Manage لي (Multi-Account Environment) بطريقة Best Practices ومتواقة مع Secured.

- الـ **Benefits** :
- الـ **Automated Setup** : اـ automate إعداد البيئة بأقل خطوات.
- الـ **Policy Enforcement** : اـ automate إنفاذ السياسات باستخدام Guardrails.
- الـ **Automatic Remediation** : تصحيح تقائي لأي انتهك للسياسات.
- الـ **Interactive Dashboard** : لوحة تحكم لمراقبة الـ Compliance.
- الـ **Use Cases** : تحديد الوصول لخدمات معينة أو فرض معايير مثل PCI Compliance.

Amazon KMS And Amazon Certificate Manager (ACM)

ال Data Encryption Two Service دول بستعملهم في Data Encryption, احنا عندنا نوع من

1. ال Data Encryption In-Transit: وهو أبعد حاجه على ال internet عشان يكون أمن ليا و Client Server فا ال.

بيبعث ال Certificate Trust ببناعته لي Client عشان يأكـدـ الـ .

2. ال Data Encryption @Rest: هنا مش بتتحرك زي أنها موجوده على DB او S3 مثلـاـ فـاـ بيـتـعـلـمـها Encrypt رغم عدم تنقلـهاـ من مـكانـ لأـخـرـ .

ال AWS Certificate Manager Generate SSL/TLS Certificate بـنـاعـتـيـ لـيـ Encrypt عـشـانـ اـعـمـلـ Certificateـ .HTTPSـ .

- ال KMS هي Key Management Service هي بتـCreateـ keysـ لـيـ manageـ بـنـاعـيـ الـ keysـ لـيـ .Decryptionـ وـ الـ .

- أقدر احـطـ Policiesـ مـينـ يـسـتـعـمـلـ انهـيـ keyـ وـ هـكـذـاـ .

- الـ Keysـ بـتـكـونـ HAـ وـ Highly durableـ .

- بتـكونـ Cloudtrailـ معـ integrateـ عـشـانـ اـرـاقـبـ الـ activityـ بنـاعـ كلـ حـسـابـ ولوـ حاجـهـ باـظـتـ مـينـ بـوـظـهـاـ .

- لوـ عنـديـ keysـ عـلـيـ localـ اوـ on-premisesـ بـقـدرـ انـفـقـهـمـ جـواـ .KMSـ .

- بـيـبـقـاـ فيـ Key rotationـ اـنـيـ اـغـيـرـ الـ keyـ كـلـ فـقـرـةـ .

- هيـ Regional Serviceـ .

- ال CMKsـ هيـ Customer master keysـ الـ Keysـ بـتـخـلـقـهـاـ فـيـ KMSـ بـيـبـقـاـ الـ keyـ وـ بـيـنـقـسـمـ لـيـ أـنـتـينـ .

- لوـ عنـديـ مـثـلاـ EC2ـ مـحـتـاجـهـ تـعـمـلـ KEYـ عـشـانـ عـنـدـهـ DATAـ عـاـيـزـهـ تـعـمـلـهاـ Encryptionـ سـاعـتـهاـ هـدـيـهـاـ IAM Roleـ عـشـانـ تـعـمـلـ مـعـ .

- لوـ عنـديـ CMKـ وـ بعدـينـ الـ EC2ـ هـبـتـعـتـ الـ API Requestـ اـنـهـاـ عـاـيـزـهـ data keyـ (وـ دـيـ بـتـكـونـ temporary keysـ), هـتـرـوـحـ لـيـ CMKـ وـ تـعـمـلـهاـ الـ KEYـ وـ مـمـكـنـ تـبـعـتـ الـ keyـ تـفـسـهـ Encryptedـ الـ CMKـ وـ مـشـ Encryptedـ الـ CMKـ وـ هـسـتـعـمـلـهـ عـادـيـ .Encryption/decryptionـ PLAIN KEYـ وـ اـعـمـلـ .

- واحدـ AWS MANAGEDـ هيـ الـ keyـ createـ وـ تـحـكـمـ فـيـ كلـ حاجـهـ وـ بـيـكـونـ فـيـهـ Minimal controlـ .

- واحدـ Customer Managedـ بـتـدـيـكـ الـ data keyـ وـ تـخـلـيـكـ تـعـمـلـ الـ policy rotationـ وـ rotationـ .policyـ .

Amazon Cognito

الـ Cognitoـ هيـ Login Serviceـ وـ Sign Upـ بـتـبـقـاـ AWSـ بـدـلـ ماـ أـكـتـبـ الـ Codeـ, سـوـاءـ عـلـيـ webـ اوـ mobile appـ بـشـكـلـ سـرـيعـ وـ بـسـيـطـ .

بـتسـاعـدـ فـيـ user sign-up and sign-inـ .

بـتـسـاعـدـ فـيـ Millions of usersـ لـيـ Scalesـ .

AWS Config

الـ AWS Configـ هيـ بتـ Auditـ اوـ اـ Relationـ Resourcesـ لـيـ Configurationsـ الخـاصـةـ بـيـ Resourcesـ, أـقـدرـ أـعـرـفـ الـ Relationsـ مـاـ بـيـنـ الـ Resourcesـ وـ بـعـضـهاـ .

وـ تـقـدـرـ تـ Testـ عـلـيـهاـ لـوـ Resourceـ دـيـ بـتـمـشـيـ عـلـيـ Configurationsـ مـظـبـطـ وـ لـاـ لاـ (Compliance or Non-compliance)ـ .

AWS Artifact



The screenshot shows the AWS Artifact homepage. On the left, there's a sidebar with links to 'AWS Artifact' (selected), 'Agreements', 'Reports', 'Notification', 'Documentation', 'FAQ', and 'Forum'. The main content area has a dark background with white text. It features the 'AWS Artifact' logo and the tagline 'Compliance and security in the AWS Cloud'. Below this, it says 'No cost, self-service portal for on-demand access to compliance reports and for entering into select online agreements.' To the right, there's a 'Get started with AWS Artifact' section with 'View reports' and 'View agreements' buttons. Another section shows 'Pricing' as 'Free' for 'AWS Artifact'. At the bottom, there's a video thumbnail titled 'Get to know AWS Artifact' with a 'watch on YouTube' link.

الـ AWS Artifact هي المكان الي فيه compliance-related information الي مهمه
هي بتقدملك on-demand access لـ security و reports

AWS Shield

الـ AWS Shield وفكرتها أنها تحميني من attack زى denial of service (DDoS).

طب أزاي بي secure الـ server من الـ DDoS؟

هو أولاً كان في الأول denial of service فقط ببغا عباره عن one user مع many request ببغيت single IP لما الـ server يقع فـ IP limit لي معين أنه يقدر بيعت request يومياً.

وبعدين اتطور لي يكون Distributed denial of service فـ بقا عباره عن أكثر من IP مختلف فـ مبقاش ينفع الـ limit دا يوقفه عشان كدا AWS Shield جـه يمنع الـ DDoS attack.

الـ AWS Shield عندـه Two levels من Protection.

1. الـ Shield Standard: بيدافع ضد أغلب الـ DDoS attack الي ببغا على الـ Network and Transport layer الي بتسهدف الموقع بتاعك، منغير ميسبك عليه.

2. الـ Shield Advanced: بيحمايك من الـ advanced attacks الي في layer 7 و بيديك الـ access لي ATTACKs دى. دا نيم بيساعدك انك تنهـي الـ Response team (SRT).

بستعلـها عشـان الـ Minimize Application Downtime و Latency.

AWS Trusted Advisor

الـ Service consultant دي زي يقترح عليك تدبر الـ Resources بشكل أفضل ازاي

- هي بتراقب الـ Resources بتاعتك بشكل دائم بتديك real time -guidance.
- هي بتـ Categories اللي Best Practices معين في كذا Recommendations.
- من حيث Cost Optimization او Security او Performance او Fault Tolerance او Service Limits او ازاي Run Operations بشكل أحسن.

AWS Service Catalog

لو عندك مستخدمين جدد في AWS، غالباً هييفوا مشوشين بسبب الخيارات الكثيرة، وده ممكن بسبب لو عندك مستخدمين جدد في AWS، غالباً هييفوا مشوشين بسبب الخيارات الكثيرة، وده ممكن بسبب non-compliant .

- الـ **Solution** : الـ IT Services Catalogs (Self-Service Portal) للوصول إلى AWS Service Catalog الخاص بي .
- الـ **Available Resources** : آنـا VMs (VMs)، قواعد بيانات، خيارات تخزين، وغيرها.

Amazon GuardDuty

GuardDuty هي Reactive monitoring بتنأك من logs و الـ account بتناعي عشان تتأكد من malicious activity . Threat detection analyzes events and logs عشان تكتشف الحاجات الي مش معناده و تعمل هي ببساطه بتعمل . هي تكون على مستوى VPC Flow logs و route 53 و Cloudtrail و network traffic .

Amazon Inspector

Inspector هي automated vulnerability management service . يعني أن الـ service دي بتـ scans software vulnerabilities سواء في الـ EC2 او الـ Lambda Functions اوContainers او EC2 او Lambda Functions اوContainers او EC2 او الـ . update و تقولي عشان اعمله .

Application security compliance هي .

Service	GuardDuty	Inspector
Intended use	Find if an attack or a threat exists by analyzing logs (detection of exploits). Is more like an IDS, it looks for events based on logs.	Analyzes to ensure that there is no risk even if an EC2 instance (or application) is attacked (Detects vulnerabilities even if it was not exploited).
Scope	Account scope.	EC2 instance or group of EC2 instances scope.

Amazon Detective

Detective بتعمل analyze و visualize لي issues عشان اقدر اعمل investigate الـ security data بشكل اسرع و اعرف الـ root cause .

AWS Secrets Manager

Secrets Manager دي بتـ generate credentials بدل ما أكتبـه في الـ code نفسه و بتـ change credentials كل فترة ، و rotates and audits credentials . Encrypt data in transit and at rest using KMS keys ،

Amazon Macie

ال Macie هي service fully managed data security and data privacy Scan وتحقق دا عن طريق pattern matching machine learning Protect personally sensitive data discover اشنان pattern matching و machine learning AWS identifiable information(PII).

AWS Web Application Firewall (WAF)

ال WAF هي firewall بتساعدك monitor ال HTTP(S) Requests.

طب ما انا عندي ال NACL و Security Group ليه يحتاج ال WAF؟

- لأن ال NACL و SG هي بتحمي او بتعمل Rules على Layer 3 و 4 وهما ال Network Layer و Transport Layer.
- ولكن لو في attacks على ال Application layer 7 هي ال NACL and SG ساعتها مش هيقدر يحمي ال App بناعي.
- واسعتها هيجي دور ال WAF.
- بيحمي ال app من malicious traffics زى SQL Injection - XSS "Cross Site scripting" - ال "Cross Site scripting".

Service	CloudFront*	Route 53	WAF
Meaning of the Geo feature	Filter out (block) traffic from certain regions or countries.	Confines access to the traffic to a certain region . Not meant to be a filtering or blocking service.	Filter out (block) traffic from certain regions, or countries.

AWS Elastic Beanstalk

هي Easy to use service بتعمل بيهها servers familiar ، بيكون لي deploy, managing, and scaling web applications infrastructure ، دي تستعملها لو مش عايز تعرف ال شغالة with apache, nginx, passenger, docker, tomcat, and MS IIS ازاي.

بندعم ال GO, Ruby, Python, Java, Nodejs, Dotnet and PHP

مبيكلاش حاجه ولكن بتدفع علي ال infrastructure الشغالة.

AWS Systems Manager

لية أستخدامه – Why

لو عندك سيرفرات على AWS أو حتى سيرفرات On-Premises أو في Cloud تاني، وتحتاج تتحكم فيها من مكان واحد:

- تدخل عليها من غير ما تفتح SSH أو RDP.
- تعمل Automation لناسكات مملة زي التحديثات أو الباتشات.
- تدبر ال Secrets بتاعتك بشكل آمن.
- وتعرف مين compliant ومين لا.

هنا بقى بيجي دور SSM.

بيشتغل إزاي – How

- كل Machine عنده EC2 أو سيرفر (On-Prem) عليها SSM Agent بتسطع عليه AWS Agent ده بيكلم خدمة SSM على AWS.
- أي أمر تبعته من AWS CLI أو AWS Console بيتفذ على السيرفر ويرجع لك النتيجة.
- والـ CloudWatch Logs ممكن تترفع على S3 أو AWS Lambda.

إمكانياته – What

1. Session Manager

تدخل على AWS EC2 من غير ما تفتح Port 22 أو 3389 SSH/RDP آمن وسهل، وكله بيتسجل.

2. Run Command

عايز تشغل Script أو Command على 10 أو 100 سيرفر مرة واحدة؟ زي مثلاً Run Command أو restart yum update بيعملها في ثانية.

3. Patch Manager

عايز السيرفرات بتاعتك تبقى Up to Date؟! Automation Patching يقدر يعمل SSDM يحدلك جدول للتحديثات.

4. Automation (Documents)

عبارة عن خطوات جاهزة (أو تكتبها بنفسك) عشان تنفذ تاسكات متكررة زي:

- Backup لـ AMI
- Restart خدمات
- Cleanup لـ CloudWatch Logs

5. Parameter Store

مخزن آمن تحفظ فيه Variables أو Secrets زي:

- Passwords
- API Keys

وتحكم مين يقرأهم عن طريق IAM.

6. Inventory & Compliance

يقولك كل سيرفر فيه إيه:

- نسخة AWS OS
- الباكيجات اللي متسقطة
- مين compliant مع السياسات بتاعتك ومين لا.

أمثلة بسيطة

- عايز تدخل على سيرفر: افتح Session Manager، مفيش SSH ولا Ports مفتوحة.
- تحديثات جماعية: Run Command وتشغل Script على كل AWS Instances مرة واحدة.
- إدارة Secrets: تخزنهم في Parameter Store، وسيب التطبيق يقرأهم بأمان.

المميزات

- أمان: مفيش Ports مفتوحة.
- تكلفة: الخدمة نفسها Free تقريباً، بتدفع بس على AWS CloudWatch Logs لو رفعتها في S3 أو AWS Lambda.

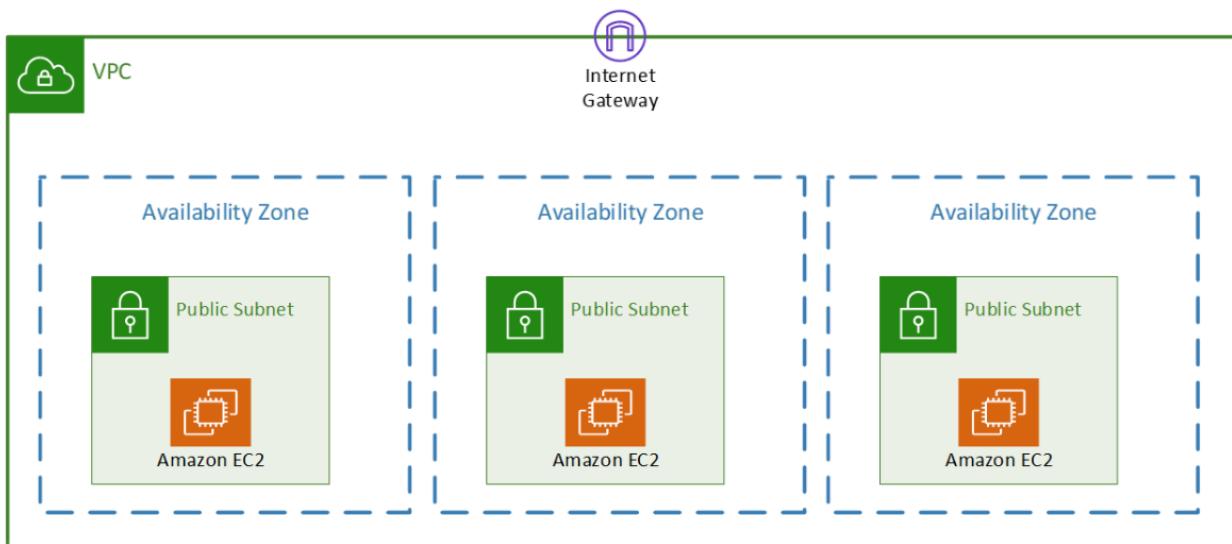
- **Integration** جامد: شغال مع AWS Ecosystem وكل اد، IAM، CloudWatch، S3 و كل اد.

	SSM Parameter Store	AWS Secrets Manager
Storing plain/encrypted text data	Both are supported. Encrypted uses KMS keys.	Only encrypted data using KMS keys.
Use case	Configuration data management including secrets.	Secrets management only.
Audit and IAM Policies for access control	Supported	Supported
Storing values under a name or key	Supported	Supported
Referencing from CloudFormation templates	Supported	Supported
Cost	Free for standard (limit on parameters total number), chargeable for advanced.	Chargeable per secret per month, and for each 10K API calls.
Secret and DB credential rotation and LifeCycle management	Not directly supported but Parameter Policies can do a similar job (parameter expiration, TTL).	Full DB credential rotation on a schedule with RDS (& Aurora). Custom Lambda function can be used for other services.
Maximum size	Max parameter value size is 4KB for standard, 8KB for advanced tier.	Maximum length of a secret is 64KB.
Tracking changes and versions	Tracks parameter history (changes) and can have up to 100 versions of a parameter.	Integrates with CloudTrail where API calls (including changes) can be logged.

AWS Networking

Now that you know *the basics of networking*, let's explore how AWS uses networking to connect, protect, and organize cloud resources..

1. VPC And Subnets



الـ **VPC**: ده Logical network AWS دايرنقدر تقدر Launch Services (DB، EC2، زى حاجة) فيها أي حاجة .Datacenter Isolated زى الـ Traditional network في أي زى الـ .

- يبيكون في region (زى "eu-west-2" أو "us-east-1") محدده .
- آخرك تعمل 5 VPC في الأكانت الواحد بس دا Soft limit يعني تقدر تغيره مع Support .

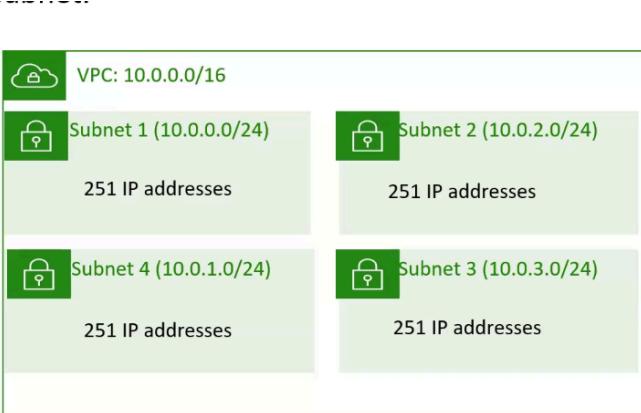
الـ **Subnets**

- بقى الـ VPC لي عدد من subnets عشان اخد IP Range of IP , الـ Subnet آخرك تعملها لـ AZ واحد .
- آخرك تعمل 200 VPC في Subnets .

- فيه نوعين من الـ subnets
- الـ Internet gateway: أي instance هنا يقدر يصل له من الإنترن特 ودا بيكون عن طريق الـ Public Subnet
- الـ Internet gateway: بيبقا معزول عن الإنترن特(مش مرتبطة بي Private Subnet)

الـ CIDR Block

- الـ MAX بيبدأ من 16 addresses .addresses 28 من Min بيبدأ من
- متقدرش تعمل Overlapping يعني تكرر ال CIDR في كذا Subnet عشان ميتوافق فيه IPs متشابه.



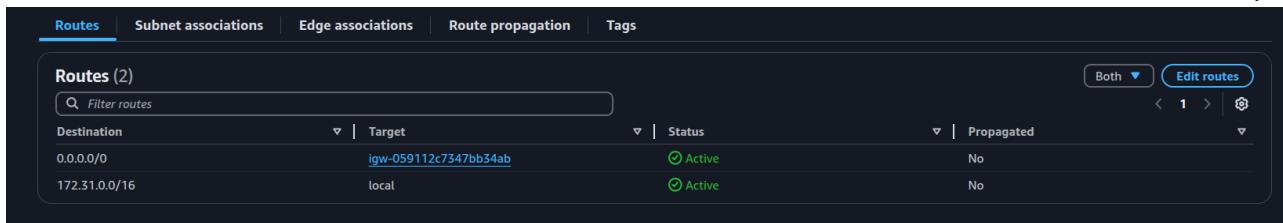
IP Addresses for CIDR block 10.0.0.0/24	Reserved for
10.0.0.0	Network address
10.0.0.1	Internal communication
10.0.0.2	Domain Name System (DNS) resolution
10.0.0.3	Future use
10.0.0.255	Network broadcast address

aws

أ Network address Reserve IPs اكتر من ال Network الطبيعية (في Reserve Networkips بـ AWS) و .(Broadcast

الـ Routing: يستعمل route table عشان احدد ال Network traffic في subnet او gateway هيتوجه لغين.

- عنده حاجه اسمها route propagation يعني أنه routes الجديدة بـ automatic route tables يتعلم لي routes الجديدة بـ manual transit او vpn او بـ direct connect.



فـ على الصورة الي فوق عندنا route table فيه حاجتين ال target وهو بيكون ال gateway او local network او target, فـ المثال الي عندنا هنا فيه internet gateway عشان بشوف النت, فـ يعني كـ ان اي subnet attach لـ route table attach لـ public subnet وتبـقا internet.

الـ Route table association: لازم كل subnet route table يكون مرتبط بي one route table فـ كل subnet اقدر اربطها بأكـتر من gateways بعض.

كل VPC يكون فيها Main route table عشان توجه ال internal traffic بـ CIDR Block عـشان يخـلي ال resources يكمـلوا مع بعض.

من الـ best practice انه كل نوع subnet يستعمل route table different.

لو مفيش اي route table عملتها تلقائي هيستعمل ال main route table.

2. IP Addresses In AWS

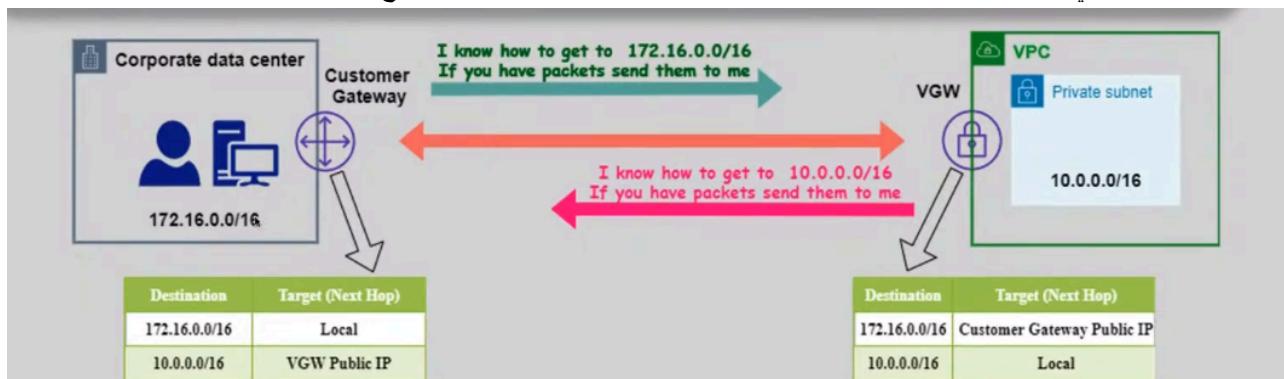
الـ Public IPv4

- دا IP بيخلي أي حد يقدر يوصل لي .internet
 - بيتغير كل ما تعمل لـ .instance restart
 - **:Private IPv4**
 - دا IP للجهاز فقط بي map بيه جوا ال network بناعتي (مثلا 10.0.0.1).
 - بيثبت حتى لو عملت لـ .instance restart
 - مجاني ومش بيتحاسب عليه.
 - **:Elastic IP**
 - عنوان IPv4 وثابت تقدر تربطه بأي .instance
 - آخرك تعمل 5 EIP في الاكانت و Region بس دا soft limit تقدر تغيره لما تكلم ال support
 - **:IPv6**
 - كل عنوانين IPv6 في AWS تكون Public (مفيش private IPv6)
 - الاستخدام مجاني، لكن مش كل الخدمات تدعمه.
-

3. AWS Gateway's Service

ال :Internet Gateway (IGW)

- ده ال Router الي بيبقا في VPC بيسمح له instances بأنها تشووف ال internet فا هو بيقوم بمهمه ال router بظبط.
- بيقوم بمهمة ال NAT (مش هي هي ال Service الي في AWS) وأنها بت MAP ال Public IP مع ال Private IP.
- لو وصلت ال IGW في اي routetable هيخلي اي subnet public جوا access internet
- آخرك منه واحد فقط في ال VPC، AWS بتوفرك أنه يكون redundant و HA فا مش يحتاج تعمل أكثر من واحد.



ال :Virtual Private Gateway (VGW) هي gateway في ال VPC عشان يكون زي ال bridge ما بين Amazon VPC و another network. بيكون زي bridge و بشكل secure.

ال :Customer Gateway Device (CGWD) هو جهاز أو برنامج في شركتك (مثل router).

ال :Customer Gateway هو service AWS Create و Configure بعمله عن ال On-premises gateway.

ال :VGW و VGW هي الخطوات التي تحتاجها عشان تعمل ال VPC.

- حاجه ال ROUTE Table بناع كل واحد فيهم هيشارو علي الثاني.

ال :NAT Gateway/NAT Instance هي في الحقيقة PAT Service لأنها بتساعد كل instances على ال Internet بـ IP واحد.

- دول بيساعدوا instances في ال private subnet تتصل بالإنترنت، لكن من غير ما الإنترت يوصل لهم.
- الفرق بينهم:
- ال NAT Gateway عن طريق AWS managed (سرع و Secure من AWS).

- الـ **NAT Instance**: انت اللي تتحكم فيه (زي أي EC2 عادي و Security بتكون عليك)، لكن تحتاج إداره وصيانة.

الـ :Elastic network interface(ENI)

- دا ببيفا instances لي Virtual network interface بتاعتي عشان تقدر توصلها بالInternet بعد كدا (بسحب IP من Subnet اللي هيا فيها)

الـ **Endpoints و Gateways** هي طريقة بربط ببها الـ network Isolated network environment بـ Gateway هي طريقة على سبيل المثال، اني استعمل الـ VPC بتاعتي عندي internet gateway عشان اخلي الـ Endpoint, أما الـ Internet هي طريقة بربط الـ AWS Services في AWS بـ IGW او NAT.

4. Security Layers

تحتاج تبقا rules جدا في كل layer. restricted

الـ :Network ACL (NACL)

- بيكون على مستوى الـ subnet (أي instance في الـ subnet) بيعنى على الـ Rules بتاعتة).
- بيتطبق في implied router level فـ أي subnet عندي هناخد الـ rules بتاعتـه فـ وانت بتحطـ الـ rules لازم تتأكدـ أنها تتفـع للـ كلـ .
- يقدر يسمح أو يمنع traffic بناء على IP معين (Allow and deny).
- لكلـ rules ليها sequence و الـ rules بتشتغلـ من أقلـ sequence لي أعلاـهمـ، ولو لقتـ denied هـنطلعـ مشـ هـتكـملـ .
- هو (stateless) يعني بيتأكدـ من الـ Inbound و الـ Outbound ، فـ لازم تحددـ الـ Rules من اتجاهـينـ (& & outbound).

- آخرـ NACL لي كلـ Subnet

- الـ Default NACL بتاعـها بيكون denied

الـ :Security Groups (SG)

- بيكون على مستوى الـ ENI بتاعتـ الـ instance.
- يقدر يسمحـ بـ rules فقطـ بيكونـ اسمـها permitـ rules يعنيـ permitـ rules.
- أقدرـ أستعملـ الـ SG IDs كـ Destination او Source فيـ الـ rules.
- هو (stateful) يعنيـ بيتأكدـ منـ الـ Inbound فقطـ مشـ بيحتاجـ يتأكدـ منـ Outbound، فـ لو دخلـ request منـ بـرهـ، هـيرـدـ عليهـ منـ غيرـ ما تضـيفـ rule لـ outbound.
- لـو معـملـشـ SG لي Createـ Resource لـوـحدـهـ وـهـيـفـا all traffic Denied by default.

الـ :Firewall

- بيـقاـ علىـ مستـويـ الـ OSـ، وـيـعملـ حاجـهـ اسمـهاـ hardeningـ وـظـفـتهاـ أنهاـتـ disableـ لـي unusedـ portsـ.

5. VPC Connections

الـ VPC مشـ منـعزلـ! فيهـ طـرقـ كـثـيرـ توـصلـهـ بـ networksـ ثـانـيـةـ فيـ AWSـ أوـ بـ servicesـ ثـانـيـةـ بـ برـةـ. هـنـتكلـ علىـ أشهرـهاـ:

A) VPC Peering

إـيهـ هوـ **VPC Peering**؟: هيـ توـصـيلـ Two VPCs بـ بعضـ عنـ طـريقـ peering connection وـدا عـشـانـ الـ traffic مـابـينـ الـ VPCـيـ عـلـيـ الـ الآـتـيـنـ VPCـيـوصـلـ عـادـيـ وـبـيـكـنـ Private connection لأنـهـ مـطـلـعـشـ بـراـ الـ resourcesـ Isolated networkـ environmentـ.

- هو يحيط الـ IP بتاتع كل VPC جو Route table بتاتع كل VPC
- الـ IP ranges (CIDR) بتاتعة الـ VPCs لازم تكون مختلفة عشان ميحصلش (overlapping).
- الـ Peering Connection مش transitive: يعني لو VPC A متصل بـ B، و VPC B متصل بـ C، يبقى A متصل بـ C.

مشكلتها في الـ scaling limitations يعني مش هقدر اوصل اكتر من Two VPCs بنفس الـ peering connection.

B) VPC Endpoints

في العادي منغير ما نستعمل الـ VPC Endpoint عشان تكلم AWS عن طريق تعدى على internet ودا مش Secured Service.

بيانات مهمة

- إيه هي الـ VPC Endpoints؟: طريقة Secure توصل الـ VPC بـ services AWS (زي S3، DynamoDB)
- على الإنترت.

أنواعه:

- الـ Gateway Endpoint: لـ S3 و DynamoDB بس.
- الـ Interface Endpoint: لـ باقي services مثل Lambda، EC2 API

C) AWS PrivateLink

- إيه هو؟: طريقةربط الـ services لـ آلاف الـ VPCs.
- لو عايز الـ Customers يكلموا الـ Service عن طريق الـ AWS Internal network.
- تعمل الـ Network Load Balancer (NLB) في الـ VPC اللي فيه الـ Service وعشان تربط الـ Services هتسعمل VPC PrivateLink و الرابط دا بيكون هو الـ Endpoint Interface.
- المميزات:
- مناسب للشركات الكبيرة اللي عايزة توزع خدماتها بشكل آمن.

D) AWS VPN

إيه هو Site-to-Site VPN؟: توصيل بين شبكة AWS والـ Public internet والـ on-premises Traffic.

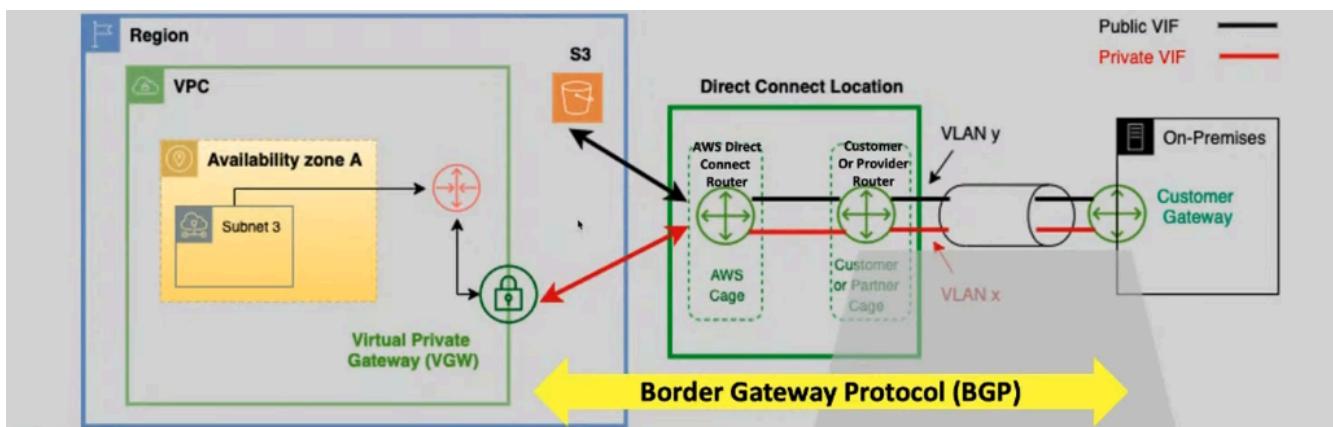
- المتطلبات عشان يستغل:
- الـ Customer Gateway Device (CGWD): جهاز أو برنامج في شركتك (مثل router).
- الـ Customer Gateway: هو Create resourceConfigure بعمله AWS يعبر عن الـ On-premises gateway.
- الـ Virtual Private Gateway (VGW):
- بتEncrypt Data ودي بـ استعمال VPN.
- مشكلة انه بطئ في نقل عشان يبعدي على الـ public internet.
- مشكلة انه مش بي support الـ IPv6 Traffic على الـ VGW.
- تقدر تعمل monitoring عن طريق الـ cloudwatch هيشوف الـ VPN Tunnel.
- بتحاسب data transfer out وبختلف من region لثانية، و per hour.

ایه هو Client VPN؟ هي مابنية على OpenVPN Technology, هي managed service, بتخليك تقدر ت access resources على AWS و secure On-premises network بشكل client VPN, بي بتقدر Access resources على AWS على location openVPN-based.

- اللهم بتتحكم فيهم لاما يشغل الـ Client VPN Connection.
- الـ Client VPC endpoint بتعملها create و configure في AWS, ومنها تتتحكم أنهي Network و resources.
- الـ VPN Client Application اللي بتستعملوا عشان تخلي Client connect مع الـ endpoint .endpoint

E) AWS Direct Connect(DX)

هو Physical Connection بين AWS On-premises و Physical Location طب ازاي؟ عن طريق أنه بيعمل connection يعني متصل بي Customer Cage و دا اللي بياخد ال AWS Cage هما ال Cages من on-premises و بيوصلها لي VGW يكون فيه AWS VPC Network و بحتاج في ال AWS VPC Network برو عشان هو ال bridge اللي بيعدي منه كل حاجه بشكل secure.



عنه 3 حاجات مهمه:

- الـ AWS Network متصل بي Private VIF (Virtual Interface) معينة في AWS.
- الـ AWS Service متصل بي كل Public VIF (Virtual Interface).
- الـ Transit Gateway متصل بـ Direct Connect gateway.

يكون Dynamic routing.

الـ Data مش تكون Public VIF و هي on transit فا حلك تعمل IPsec encrypted فوق.

الـ Characteristic

- الـ Connection هو fast و Secure.
- الـ Traffic بيعدى على Private network.

الـ عيوب: مكلف و يأخذ وقت (شهر على الأقل) عشان يتنفذ.

G) Transit Gateway

ایه هو Transit Gateway؟: Hub يوصل أكثر من VPC و شبكات on-premises مع بعض وهو Regional Service.

أزاي؟: عن طريق ال Transit Routing Table ببيقا فيه كل VPC CIDR ويوصلهم ببعض و في نحية ال VPCs الثانية بحط ال Route table في GW الخاص باكل VPC.

مثال: لو عندك 10 VPCs و 5 on-premises networks، تقدر توصلهم كلهم بـ Transit Gateway بدل ما تعمل peering بين كل واحد.

AWS Networking Deep Dive

شرح Networking Services foundation بشكل بعمق عن ال

Why use an Amazon VPC?

لأنك تقدر ت Create Logical environment في دفانق على الـ data center كانت موجود على Cloud.
ولأنك على الـ Cloud بتدفع على الـ Cloud بتستعمله فقط من الـ resources فـ هي More cost-effective.
تقدر ت migrate Cloud services بسهولة وتكون secure, scalable, and reliable.
تقدر تشغليها مع third-party services.
تقدر تعمل كذا VPC و ت create test environments قبل متطلع على production.

IP addressing in Amazon VPC

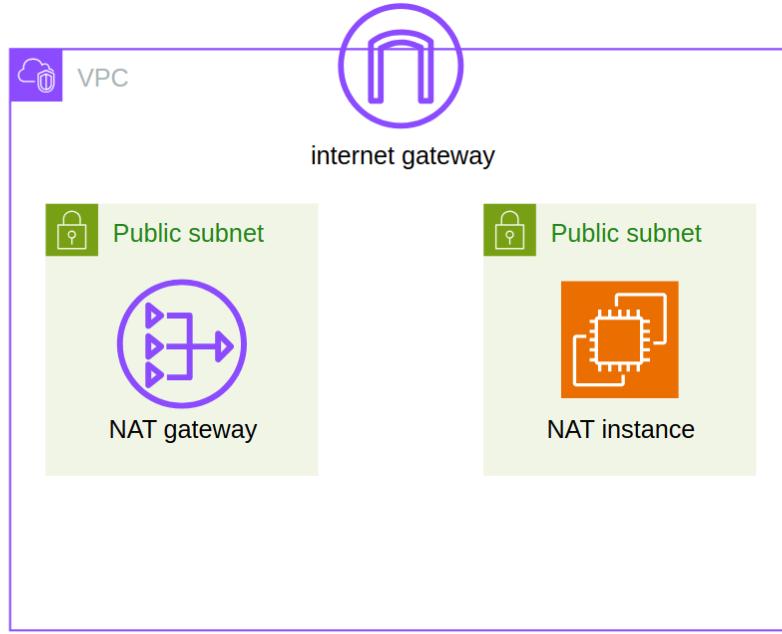
لما بعمل ال CIDR محتاج اخد في الأعتبار ال IP Range الي ه يكون في VPC او Subnet .
فا على سبيل المثال لو عملت 16 block /16 هتاخد 65,536 IP addresses .

RFC 1918 range	Example Amazon VPC CIDR block
10.0.0.0–10.255.255.255	10.0.0.0/16
172.16.0.0–172.31.255.255	172.31.0.0/16
192.168.0.0–192.168.255.255	192.168.0.0/16

والـ Private ranges الي بحددها يفضل تكون تبع RFC 1918 range .

الـ IPv6 Egress-only gateway هو Internet gateway بيبيل IPv6 فقط و بقدر اديه لي Private subnet انه تكلم الـ internet زيها زي ال EC2 في Private subnet بيخلي ال IGW + nat statful retrieve internet تكلم الـ internet المعلومات او الحاجة بس محدث هيعرف يصلها .

Internet gateway, NAT Gateway and NAT instance



Highly available Router بظبط بيساعدك عشان تقدر تتكلم مع Internet من جوا ال VPC، بيكون **Internet Gateway** ال عشان يناسب كم ال Traffic الي داخلة و الي طالع منه.

VPC > Internet gateways > Create internet gateway

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource.
[Add new tag](#)
You can add 50 more tags.

[Cancel](#) [Create internet gateway](#)

عشان تعمل Internet gateway مش بتحتاج غير أنك تسمية وتعملوا attach في VPC الي عايزها.

هي Service بتخلي ال Private subnets توصل لـ Internet منغير ما أعملExpose لي ال IP بتاعها, بتشتعل فقط مبياخدش Elastic IP بي.

ب تكون instance انت عاملها Security Configure وقافل فيها ال Src/dest check عشان هي مجرد وسيط هي مش بت support ال Subnet ولكن ال NACL ممكن تحطها علي مستوى ال Security group ما حماية ليها.

و عشان ال redundancy اعمل deploy NAT gateway/instance لي اكتر من AZs ولو واحد وقع يكون معاك غيرة وتحقق ال HA.

عشان تعمل NAT Instance Steps

Instances (1/1) Info

Last updated 2 minutes ago [C](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive)

All states [▼](#)

Instance ID = i-01ba75cc194044e13	X	Clear filters						
<input checked="" type="checkbox"/> Name ↗	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ..
<input checked="" type="checkbox"/> nat-instance	i-01ba75cc194044e13	Running Q Q	m5.large	Initializing View alarms +	eu-central-1b	ec2-18-195-169-223.eu...	18.195.169.2...	

تعمل instance NAT تكون AMI Configured as NAT او تعملها انت Configuration

Instances (1/1) [Info](#)

Last updated 3 minutes ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[Name](#) [Instance ID](#) | [Instance state](#) [Instance type](#) [Status check](#) | [Alarm status](#) [Availability Zone](#)

nat-instance i-01ba75cc194044e13 Running [Q](#) [Q](#) m5.large Initializing

[Attach network interface](#) [Detach network interface](#) [Connect RDS database](#) [Disaster recovery for your instances](#) [Change source/destination check](#) [Disassociate Elastic IP address](#) [Manage IP addresses](#) [Manage ENA Express](#) [Manage bandwidth](#)

i-01ba75cc194044e13 (nat-instance)

و دي اهم خطوه وهي انك تقلل الSrc/dest check و تختار Networking Action و بعدين Action

Instances (1/1) [Info](#)

Last updated 4 minutes ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[Name](#) [Instance ID](#) | [Instance state](#) [Instance type](#) [Status check](#) | [Alarm status](#) | [Availability Zone](#) | [Public IPv4 DNS](#) | [Public IPv4](#)

nat-instance i-01ba75cc194044e13 Running

Change Source / destination check

The source / destination check ensures that the instance is the source or destination of all the traffic it sends and receives. Each EC2 instance performs source and destination checks by default. [Learn more](#)

Instance ID: [i-01ba75cc194044e13 \(nat-instance\)](#)

Network interface: [eni-0326f09726682baa3](#)

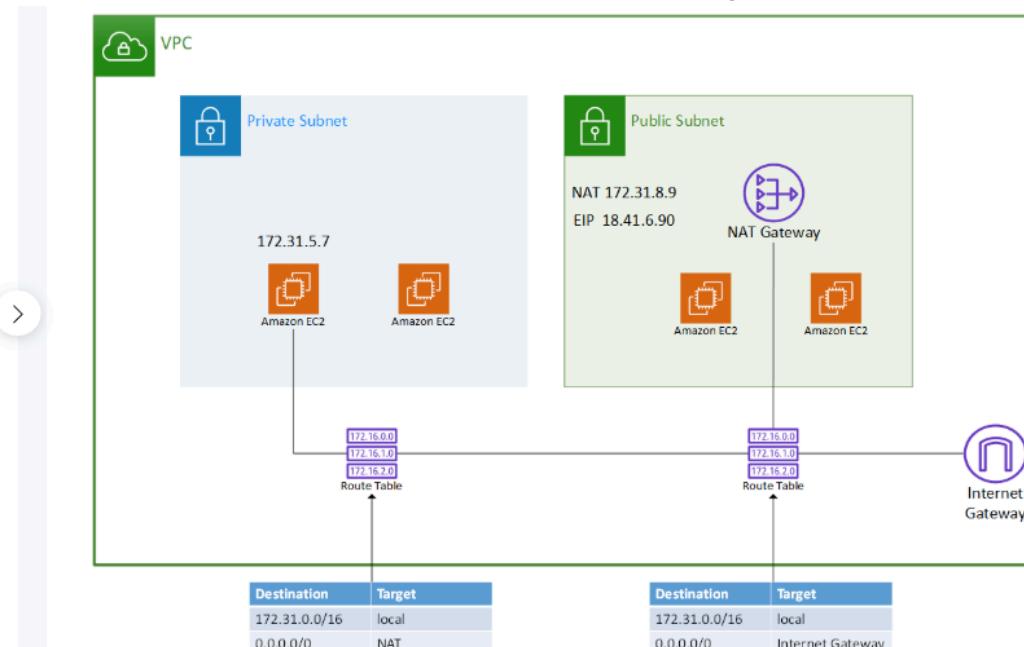
Source / destination checking:

Stop to allow your instance to send and receive traffic when the source or destination is not itself.

Stop

[Cancel](#) [Save](#)

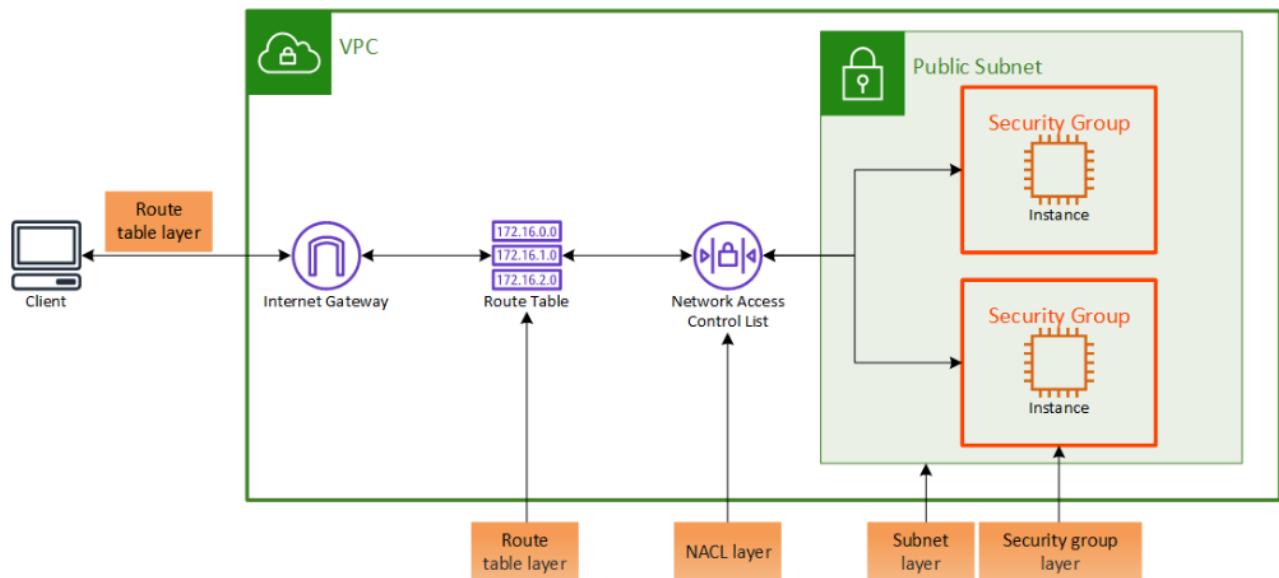
ال NAT Server شغال انت محتاج توقفه عشان يشتعل لك Src/dest check by default



بعدين تعمل تحط ال public subnet في NAT Gateway/instance و ال private subnet تكون في internet gateway متصل بي route table attached

Securing your System

Security layers of defense



الـ Public subnet مش بتقبا secure كفایا بتحتاج عشان تتحاول تمنع اکبر قدر من ال attack الي ممکن يحصل عليك.

الـ best practice اني استعمل كذا layer of defense عشان احمي ال system.

استعمل Encrypt TLS/HTTPS عشان اـ .data in transit.

الـ Security layers الي عندي:

- الـ Route table layer ودا عشان هو الي ببيوجه ال traffic الي جي والي رايح علي internet gateway
- الـ NACL layer عشان traffic Filter قبل ميخش علي subnet
- الـ subnet layer دي الي فيها resources تكون بمثابة firewall علي مستوى ال instance
- الـ security group layer

Inbound rules (3)

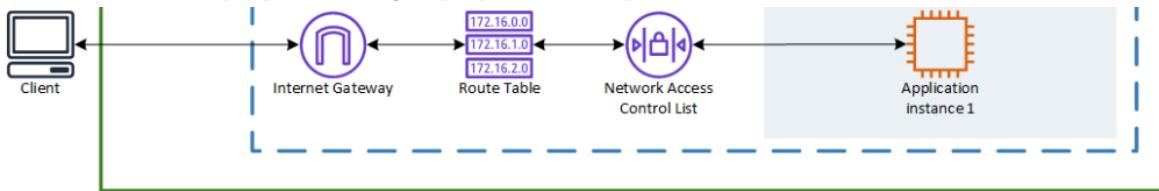
[Edit](#) [Manage t](#)

<input type="checkbox"/>	Name	Security group rule ID	Type	Protocol	Port range	Source
<input type="checkbox"/>	-	sgr-0ecd53095efb5bab1	HTTP	TCP	80	0.0.0.0/0
<input type="checkbox"/>	-	sgr-0574643ab220a1b8b	SSH	TCP	22	0.0.0.0/0
<input type="checkbox"/>	-	sgr-005c3a71371a4221f	HTTPS	TCP	443	0.0.0.0/0

دا مثال علي Security group

- بيطلب منك فا علي سبيل المثال عند web server HTTP/HTTPS علي حسب معاك ولا . Certificate
- بيطلب الـ port الي عايز تفتحها وتقدر تحط range لي ports زى 49156-65545
- بيطلب الـ source وهو الـ IP الي داخلك دا range ايه، طبعا في الـ SSH متعلمهاش 0.0.0.0/0 عشان لو اي حد معاه ال KEY هيخشن عندك عادي لا حدد انه يكون تبع Network الشركة مثلـ.

- مش هتحتاج تطبيق outbound rules لأنها كدا stateful يعني مش بيتأكد على الـ traffic قبل من الي جي فقط.



Inbound rules (3)

Edit inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
90	All traffic	All	All	12.588.205/32	<input checked="" type="checkbox"/> Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Deny

Outbound rules (2)

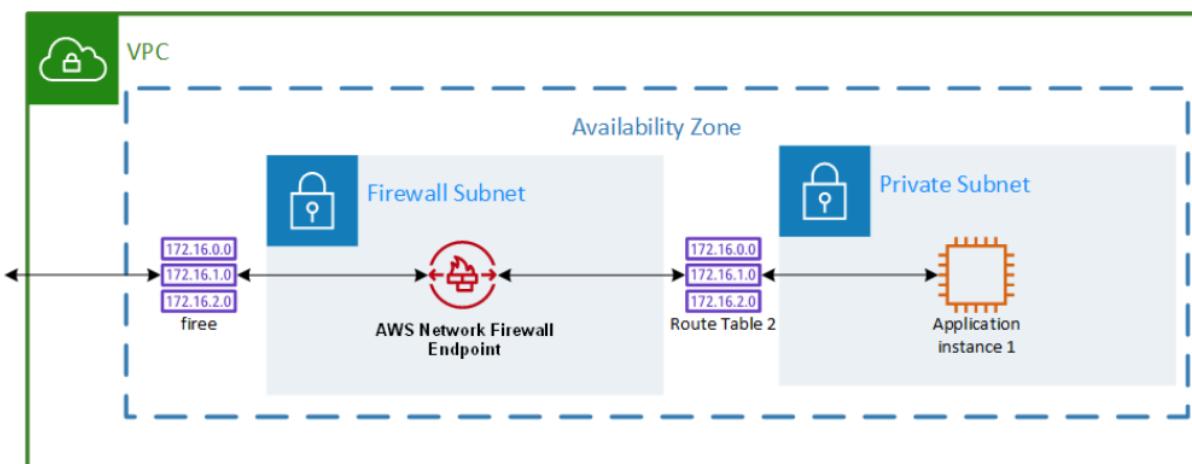
Edit outbound rules

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Deny

دا مثال على ال NACL

- كل Subnet لازم ترتبط بي NACL لو معمليش واحد هو بي Assign لوحده لي VPC's Default NACL وبيكون Allow فيها كل حاجه اما ال .deny-all custom تكون بتكون explicit deny عشان لو معمليش واحد فيها هيكون deny يعني هيرفض كل حاجه
- بتحتاج تعمل ال inbound and outbound rules عشان لو معمليش واحد فيها هيكون denied عشان لو كل الفرق منفعش ي deny فوراً.
- دايماً بيتهي بال asterisks denied وبيكون denied عشان لو كل الفرق منفعش ي deny فوراً.

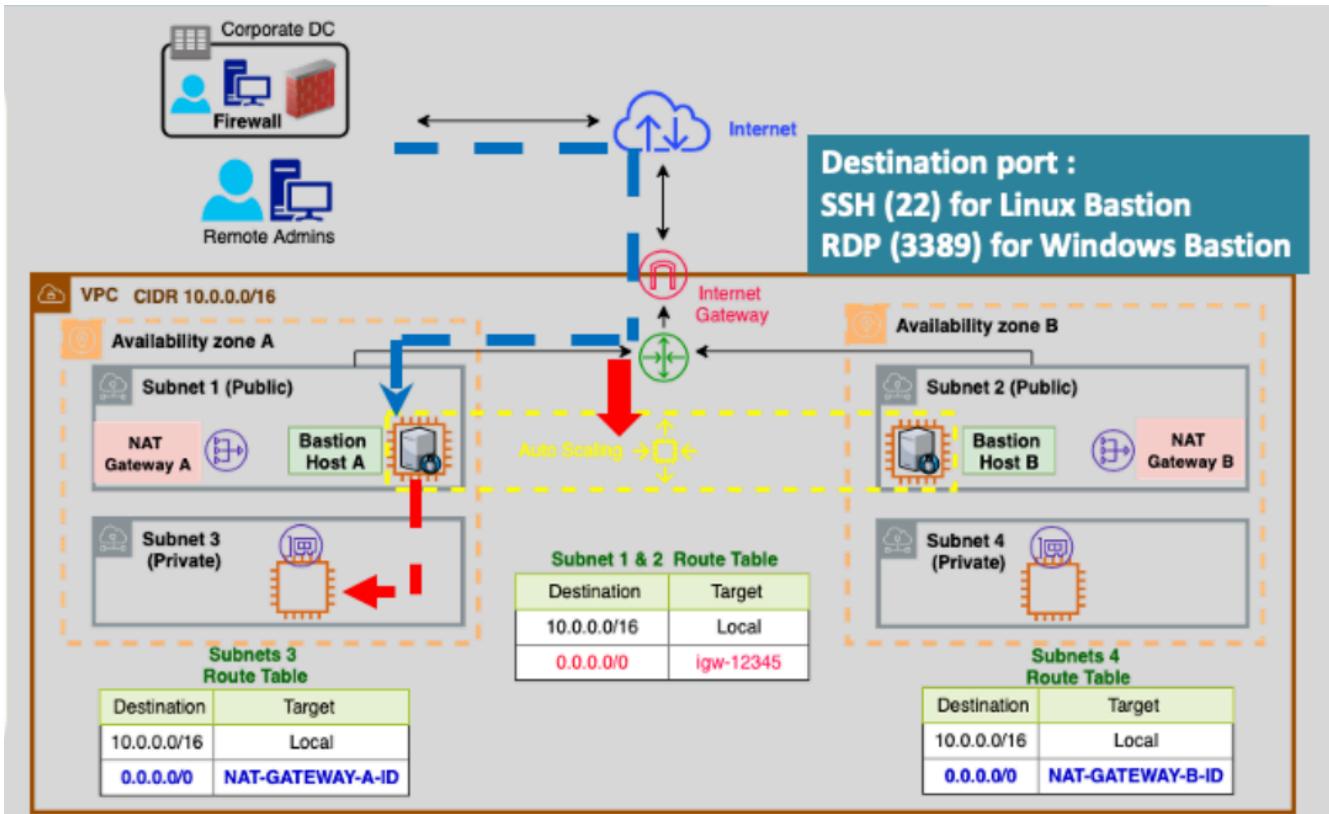
,VPC resources هو AWS Network firewall قبل ما تخش على filtering traffic stateful, managed firewall وبتعمل مابين application subnets internet gateway زى external sources قبل traffic قبل Deploy .NACL و SG بجانب ال extra layer هو



بتعملها Deploy مابين application subnets internet gateway زى external sources فالـ traffic قبل من الـ subnet لميوصل لي.

Route table configuration

- اول route table الى متصل بي internet و firewall
- ثانى route table الى متصل من firewall الى subnets



الـ **Bastion host** هو أني عمل EC2 في Public Subnets عشان اقدر ا access الـ machines اللي فيه private subnet. هحتاج secure, lock source IPs ويكون بتاع الشركة بتاعتي بس عشان محدش يقدر ي attack.

يفضل استعمل معاه Elastic IP عشان أعرفه في firewall الشركة.

اعمل lock لـ src IPs .Minimum required

How to connect to managed AWS Services

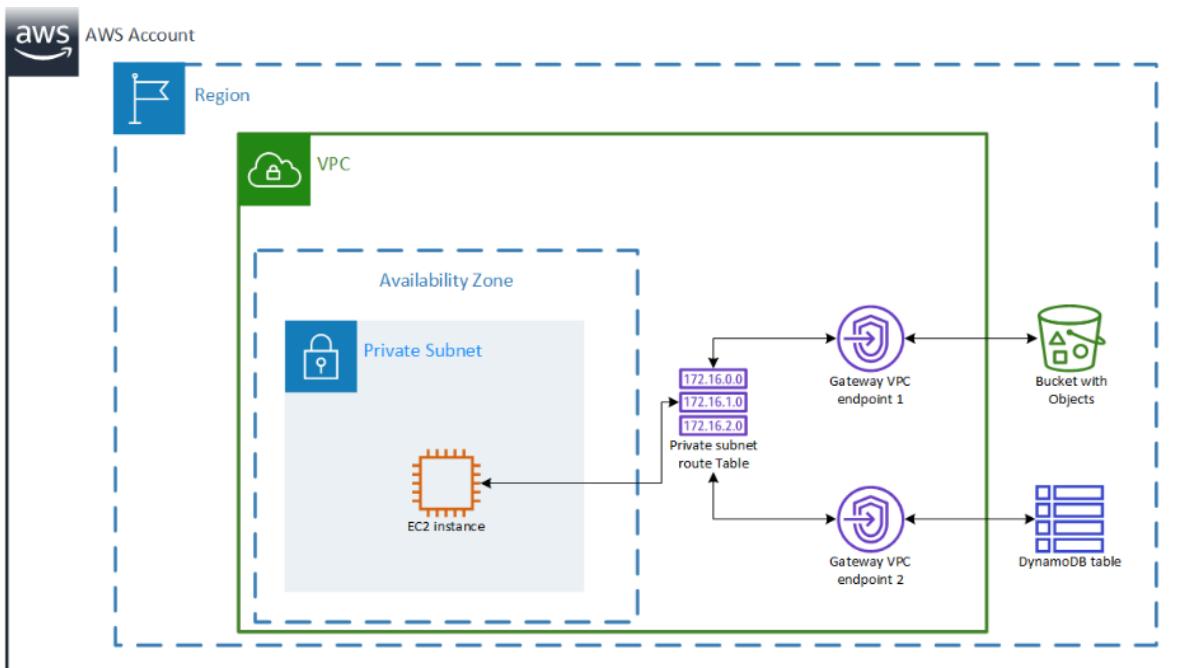


دلوقتی انت عندك EC2 في Private subnet تحتاجه S3 Bucket access و ال S3 Bucket access أحنا عارفين انه مش بيرتبط بي Network معينه فالو سبته كدا انا هحضر أمشي عن طريق ال internet gateway عشان اوصله يعني هزود NAT Gateway كمان.

فالک سبب کل دا بقا عندی حل احسن و هو VPC Endpoint

ایه هي ال VPC Endpoint؟ کنا قايلين في Cloud foundation Services ان ال Endpoint بربط ال Services بعض منغير معمل على ال internet يعني privacy عاليه.

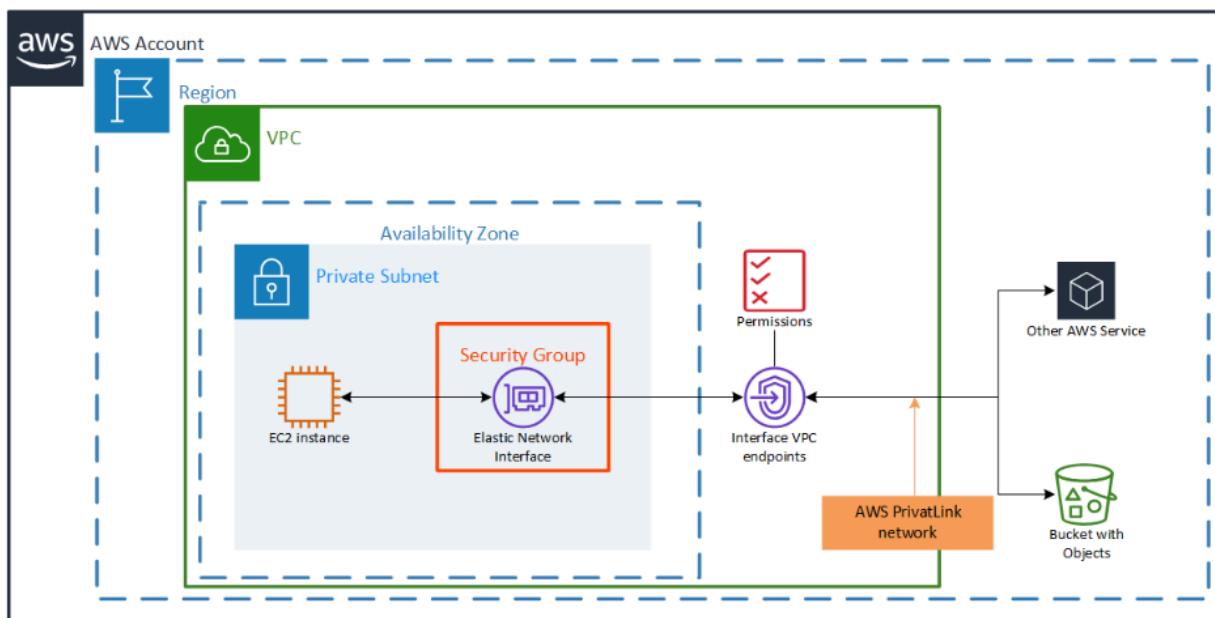
- **الGateway Endpoint:** دا بيكون recommended لـ S3, عموما هو بي Route ال traffic direct بشكل Route من خلال ال AWS Backbone ويبتتجنب ال internet.
 - **الInterface Endpoint:** دا معتمد على ENI (ENI-based), لأنه بيعمل Create ENI في Private subnet لـ ENI ودا بيخلية يزود عليك شوية costs بالذات لو هتحط ENI في كل AZ عشان ال HA.



Private subnet route table	
Destination	Target
Prefix list 1 id	Gateway VPC endpoint 1
Prefix list 2 id	Gateway VPC endpoint 2

لو هستعمل S3 هيكون RECOMMENDED عشان هو مش بيعتاج ENI يعني cost أقل وبتحطه في aws backbone لأنه بيوجه traffic خلال route table فقط.

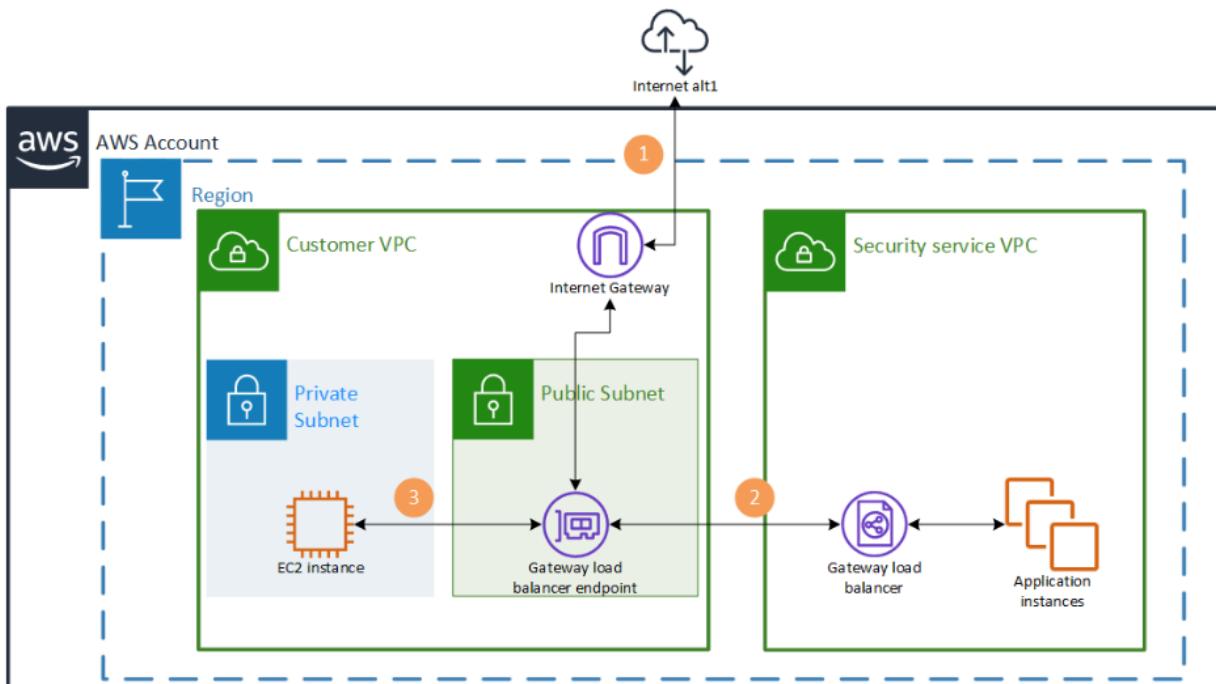
محتاج فقط أحيط الي Destination و target بتوع VPC Endpoint Gateway .Route table في الـ



لو هستعمل ال interface فا هو هيعملني IAM Policies مع Private IP subnet جو ال subnet يكون لي access على services الي هياخد منها والي هيوصلها, بس استحمل بقا ال charges الي هتحصل بسبب ال .(hourly and data processing) .

Best practice: استعمل ال gateway endpoint مع ال S3 عشان تقلل التكلفة و تحسن ال security, واستعمل ال private IP-based access الي محتاجه services عشان تربط ال endpoint .

الـ **Gateway Load balancer endpoint** ي يقدم private connectivity عشان يحمي ال appliances مابين VPCs الي عندي, يستعمل عشان يوجه ال traffic مابين 1 و 2 VPC, بنستعلمه inspect عشان او نتابع ال traffic مع applications .



الTraffic flow

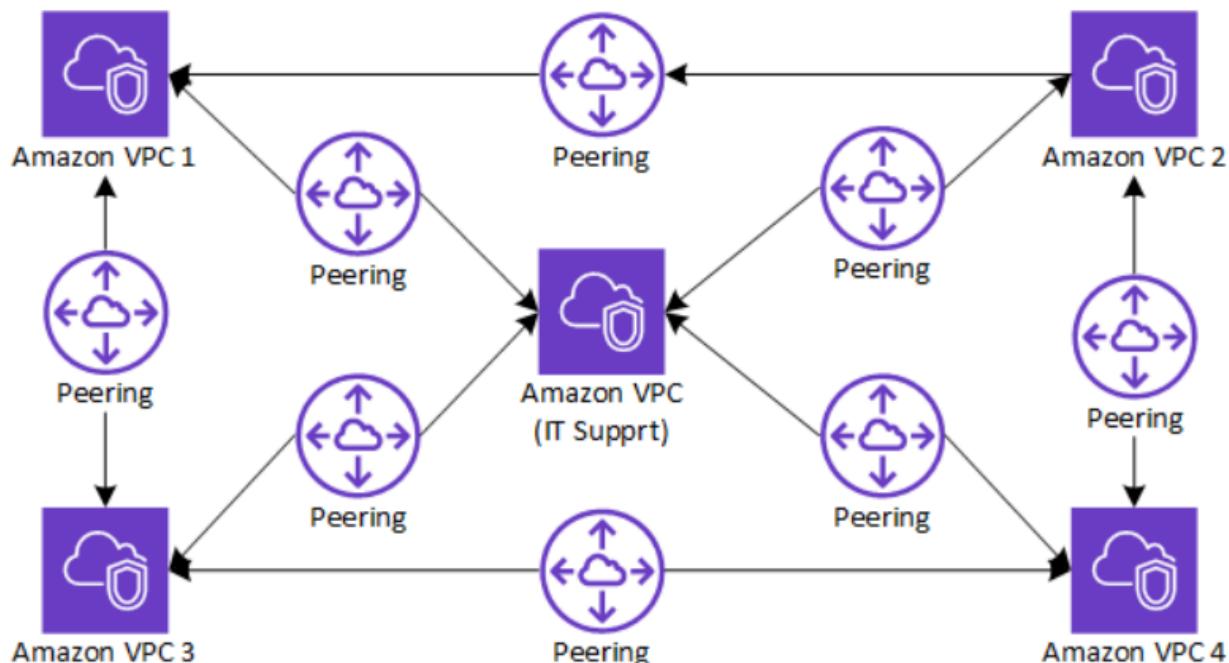
1. بيخش على VPC من خلال الـ Internet gateway ENI، بيروح على public subnet الي فيها GW Loadbalancer ENI
2. بيعت ال Traffic لـ GW Load balancer في 2 VPC (والـ 2 هي Security service في المثال دا).
3. بيتأكد من traffic وبعدين بيعته لـ EC2 الي في 1 VPC وـ Private subnet

Connecting Amazon VPCs

VPC Peering هي أني افتح Connection مابين two VPCs ودا هيخليني اقدر a route traffic مابينهم بشكل private.

- هي HA: يعني بيكون في Peering connection Redundant component ولو ال connection وقعت هيطلك غيره، يعني مش هحتاج تحط اكتر من VPC Peering عشان لو واحد وقع بيقا في connection لسه شغال.
- بيعدي على AWS Backbone عشان كدا ال communication تكون private و secure.
- تقدر تشغلو على Regions مختلفة (inter-region-peering).
- ال Peering connection مش قادر ت extend edge-to-edge routing او VPC endpoints او NAT VPNs, Direct connect, IGW.

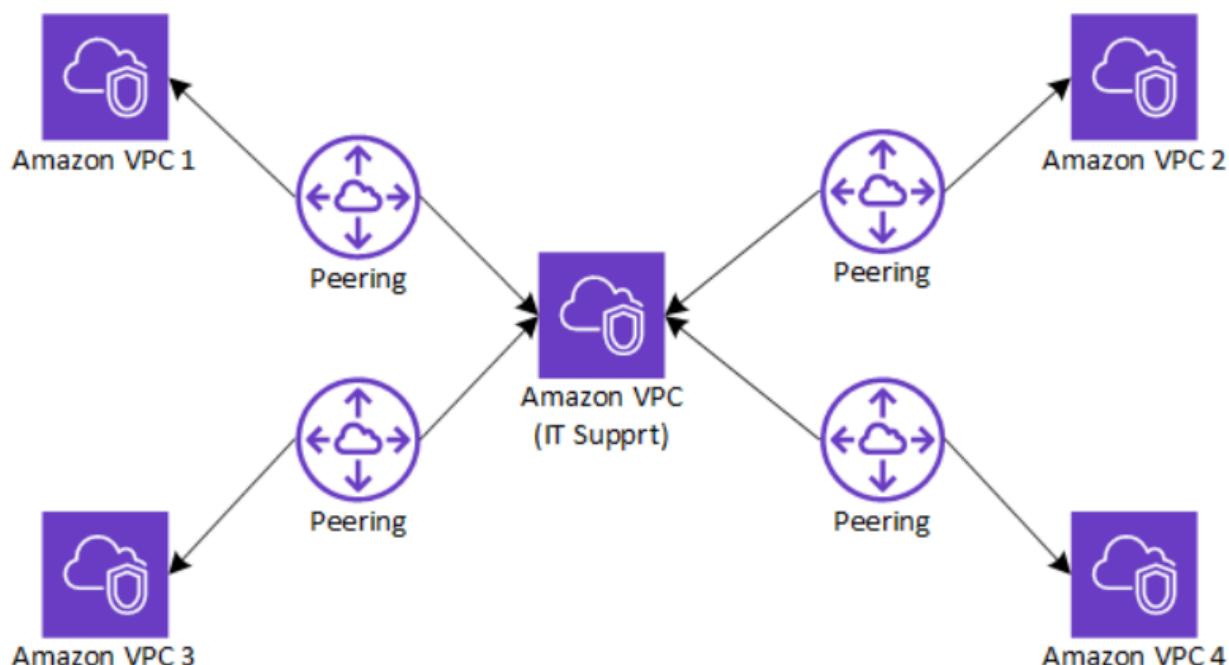
Scenario 1: Full sharing of resources between all VPCs



دا ال peering connection عندی کل VPCs کا، بیتکد ان کل عامله fully meshed VPC Peering

الحل الأفضل هنا اني استعمل VPC Transit Gateway کا Hub بیجمع کل

Scenario 2: Partial sharing of centralized resources



دا ال hub-and-spoke VPC Peering، مش بی connect کل VPCs کا، بیتعامل علی اساس ان فی واحدہ منهم

Peering pricing: کل ال peering connection علی ال data transfer علی ال بیلاش
in-region data transfer ال AZ بفلاوس بمعیار ال AZ data transfer crosses ال

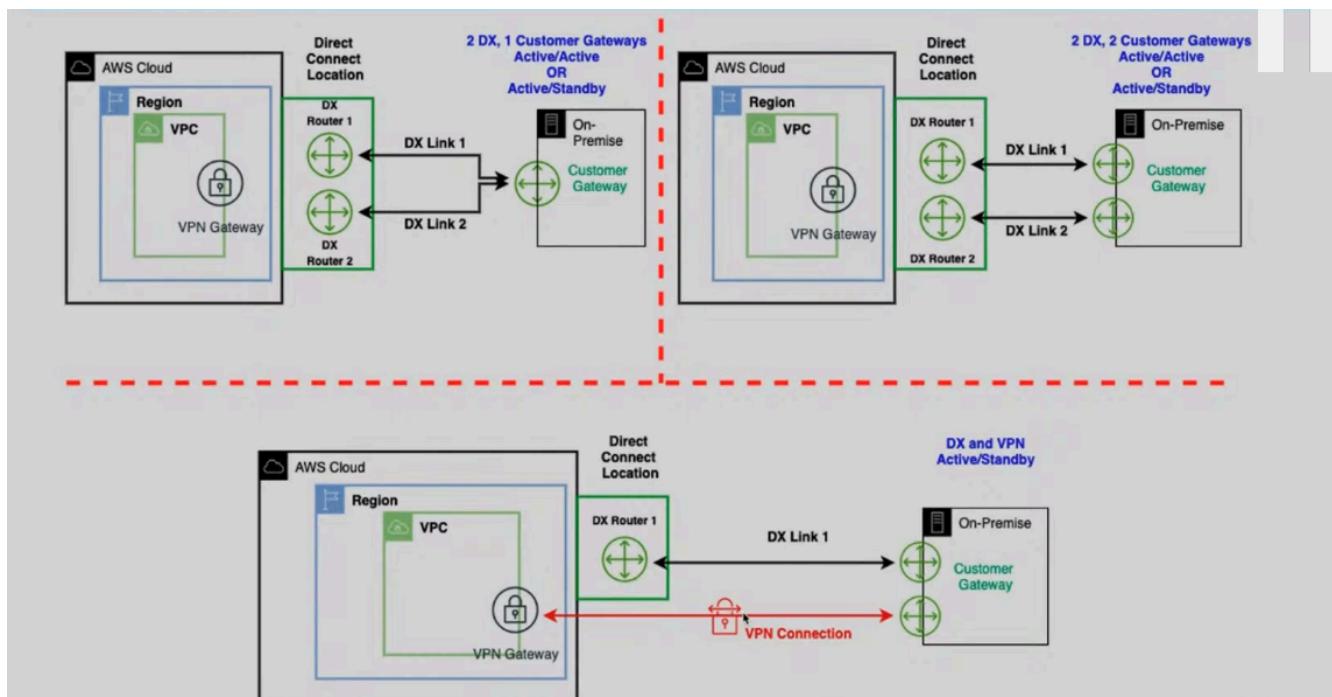
الـ Peering Connection بیتم ازای؟:

- اول حاجه هتروح ت Create VPC Peering
 - هنختار ال Two VPC الي عايز تعمل مابينهم Peering
 - وبعد مت create هتروح علي route table الخاص بي كل VPC من العندي و احط ال target routes و الي هو connection

Transit Gateway هو centralized routing يعني سيكون ز Yi ال hub بيجمع كل ال VPCs, سيكون Highly scalable عشان يقدر ي support large-scale MTU of 8500 bytes و VPCs من hybrid networks بيدعم.

- بيعتبر **inter-region Peering**, يعني تقدر ت connect اكتر من VPCs في AWS مختلفة بستعمل **Regions**.
 - (يربط ال Region transit GW في كل عن طريق ال Transit GW Peering connection) **Backbone**.
 - يكون **hybird integration**, يعني اقدر استعمل direct connect او S2T VPN عشان اربطه مع On-premises بكل بساطة.
 - طب الي تحتاج اعمله عشان اربط ال VPCs بي **Transit GW**؟
 - اول حاجة ال **Attachments**: هي زي ورقة عقد مابين ال Service الى هربطها بيها وي ال transit.
 - ثاني حاجة ال **Transit GW Routetable** بربطوا بي **Attachments** بناعه و ال Routes و تحط كل VPCs.
 - ال **Associations**: هي زي الخطيب الي بيوصل ال attachment بي route table كل attachment بيرتبط ب Route Table واحدة بس، بس Route Table ممكن يرتبط بيها اكتر من attachment.
 - ثالث حاجة الجانب من كل **VPC** تعمل route table و توجه ال routes لـ **transit gw** وبعدين تعمل associate لكل subnet تحتاجه توصل بيه.
 - **Pricing**: لكل **transit gateway** 0.1 GB of processed data 0.02\$ ولكل attachment 0.05\$ في ساعة.

AWS Direct Connect & VPN - HA and Fault tolerance

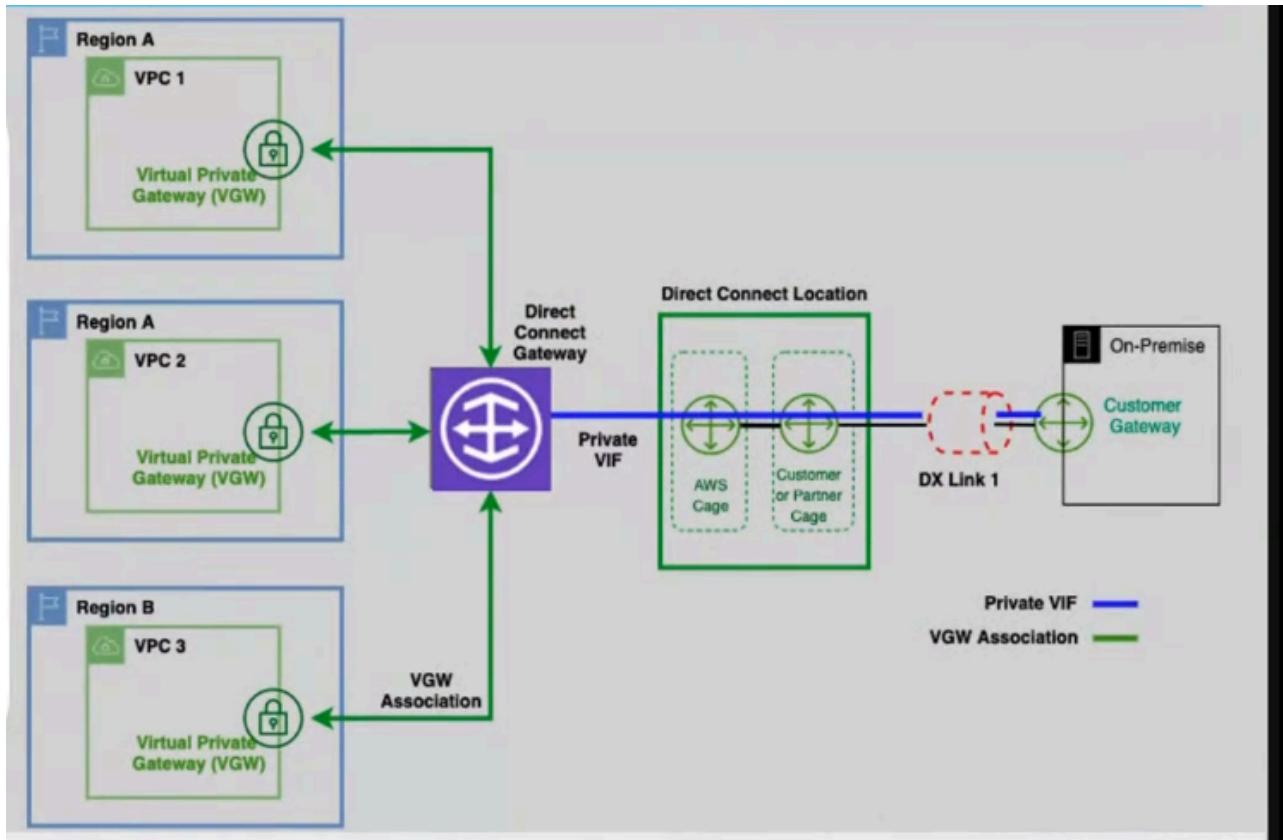


عندی 3 حلول عشان أضمن ان ال Direct connect في Connection مش هتفق:

1. ان يكون عندي VPN Connection وبكدا هيكون عندي اتنين Customer Gateway, عشان لو ال Direct Connect وقعت يكون عندي غيرها.

2. ان يكون عندي Two Direct Connect مربوطين بي واحد ودا معرض انه يكون Single point of Customer Gateway .failure

3. ان يكون عندي Two Direct Connect مربوطين بي اتنين Customer Gateway ودا أضمن من حل رقم أتنين. الـ **Link Aggregation Groups(LAGS)**: هو ببغا عندهم كذا Link بسرعة مختلفة وانا ببغا عايز سرعة معينة فا بيدبني السرعة الي عايزه من Links المختلفة دي علي شكل LAGS.



الـ **Direct Connect Gateway** دا زي hub بيعدي من عليه كل ال VGW (يجمع ال VGW ويعملها بيه) من المختلفة لي AWS وبيوصلها لـ Customer Gateway الي في On-premises Regions .

الـ **DX Gateway** بيخلی ال VPCs تتكلم مع ال On-premises ولكن مش بيخلی ال VPCs يكلموا مع بعض زي ال transit gateway .transit gateway بي Direct connect gateway اقدر اوصل ال .

VPC Troubleshooting

عشان تعمل troubleshooting مظبوط على ال VPC بنتاعتك لو فيها مشكلة: Reachability محتاج تتأكد من

- تشفو ال resources يقدروا ي connect ولا لا .
 - تمسي network traffic step-by-step المفروض يمشي ازاي .
 - تشفو ال route table, SGs, and NACLs زي ال blocking components .
 - تقدر تشفو ال flow logs لو فاتحه عشان تعرف المشكلة فين بالضبط .
- علي سبيل المثال تعمل SSH على EC2 من خلال IGW

AWS Compute Services

Now that we've covered the essentials of [Server Fundamentals](#) and [VMware Fundamentals](#), let's dive into AWS Compute Services. In this section, we'll explore how AWS provides scalable, flexible compute power to run applications and workloads in the cloud, from virtual machines to serverless solutions.

Amazon EC2

هي عبارة عن physical resources من حيث الـ Virtual machine managed by AWS ولكن

الـ EC2 بتقدمResizable compute capacity و secure

أقدر انزل على الـ EC2 أنواع مختلفة من الـ OS زى Mac او Linux او Windows

. support ليها soft limit بيفقد بي 20 EC2 Instance في الاكانت تقدر تغيره بس هحتاج تتكلم مع الـ

وانت بتعمل الـ EC2 فكر ايه الي ت interact مع EC2 عشان اعمل Attach لـ IAM Role لـ Service

EC2 Families & Types

كل نوع من instance يقدم أنواع مختلفة من memory و balance of compute و network و storage resources

General Purpose	Compute Optimized	Memory Optimized	Storage Optimized	Accelerated Computing
<p>Instance types include: A, M, T</p> <p>Balanced memory and CPU.</p> <p>Use cases include web servers, small & medium DBs, distributed data stores, among others.</p> <p>Includes burstable bandwidth instances</p>	<p>Instance types include: C</p> <p>More CPU than memory</p> <p>Use cases include batch processing, media transcoding, HPC, scientific modeling, gaming and ad serving engines.</p>	<p>Instance types include: R, U, X, Z</p> <p>More RAM/memory</p> <p>Use cases include high performance DBs, web scale distributed caches, In-memory DBs, Hadoop, Spark, HPC, Electronic Design Automation (EDA), big data.</p>	<p>Instance types include: D, H, I</p> <p>High I/O, Low Latency</p> <p>Use cases include massive parallel processing, data warehouses, log or data processing, MapReduce and Hadoop distributed computing, NoSQL DBs, cache for in-memory DBs</p>	<p>Instance types include: F, P, G</p> <p>Graphics Optimized</p> <p>Use cases include genomics, financial analysis, real-time video processing, big data analysis, game streaming, 3D applications and security workloads among others.</p>

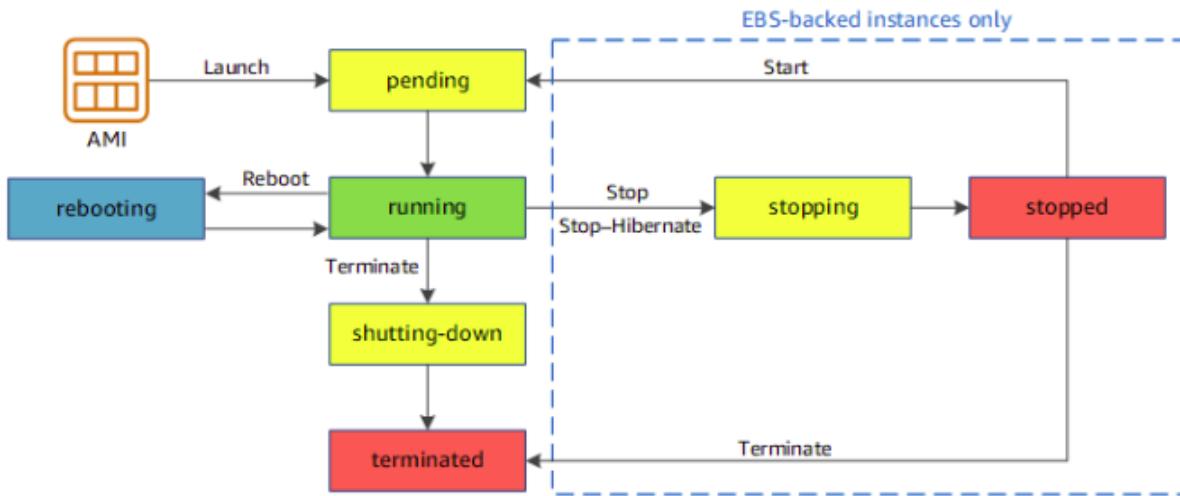
1. الـ General Purpose : مناسب لـ workloads بـ balance resources (CPU, Memory and Networking resources)

2. الـ Compute optimized : مناسب لـ workloads بـ high-performance processors (compute-intensive)

3. الـ Storage Optimized : مناسب لـ workloads بـ storage-intensive tasks يعني الـ workloads يعتمد على large data sets على local storage لـ sequential read and write access

4. الـ Memory Optimized : مناسب لـ workloads بـ large datasets processing على memory

Instance Life Cycle

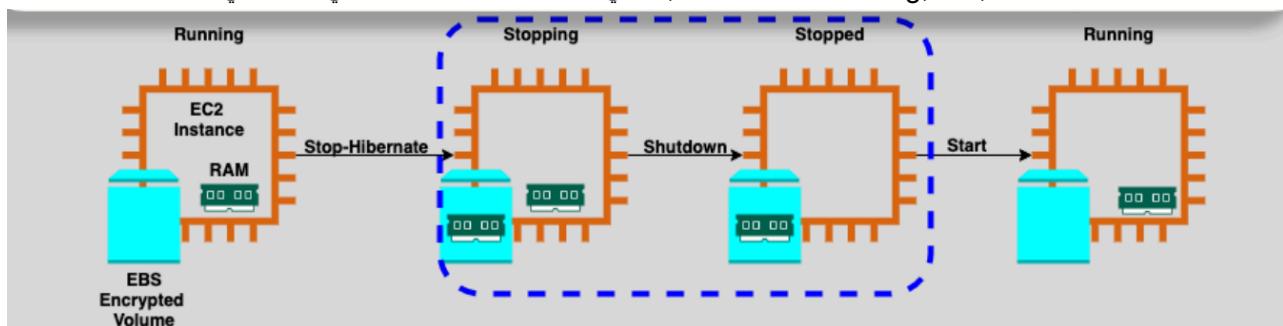


الـ EC2 بتمر بمراحل علي حسب هي بتشتغل لسه او عايز تشغليها او توقفها او تممسحها خالص.

غالبا الـ cycles كلها واحد بس الي غريبة شوية هو stop-hibernate هو وايده دا؟

- الـ .instance type دا option مش موجود في كل الـ

الفكرة منه انه بيكون عندي EC2 up and running وعليها app شغال او حاجه بتحمل وبحتاج اوقفها او انقلها اي كان ال RAM Content, instance ID لي كل دول من اول وجديد, انا بعمل Save في كل دلول RAM من اول تشغيل(EC2 ID, Processes running, IPs, and EBS root and data)

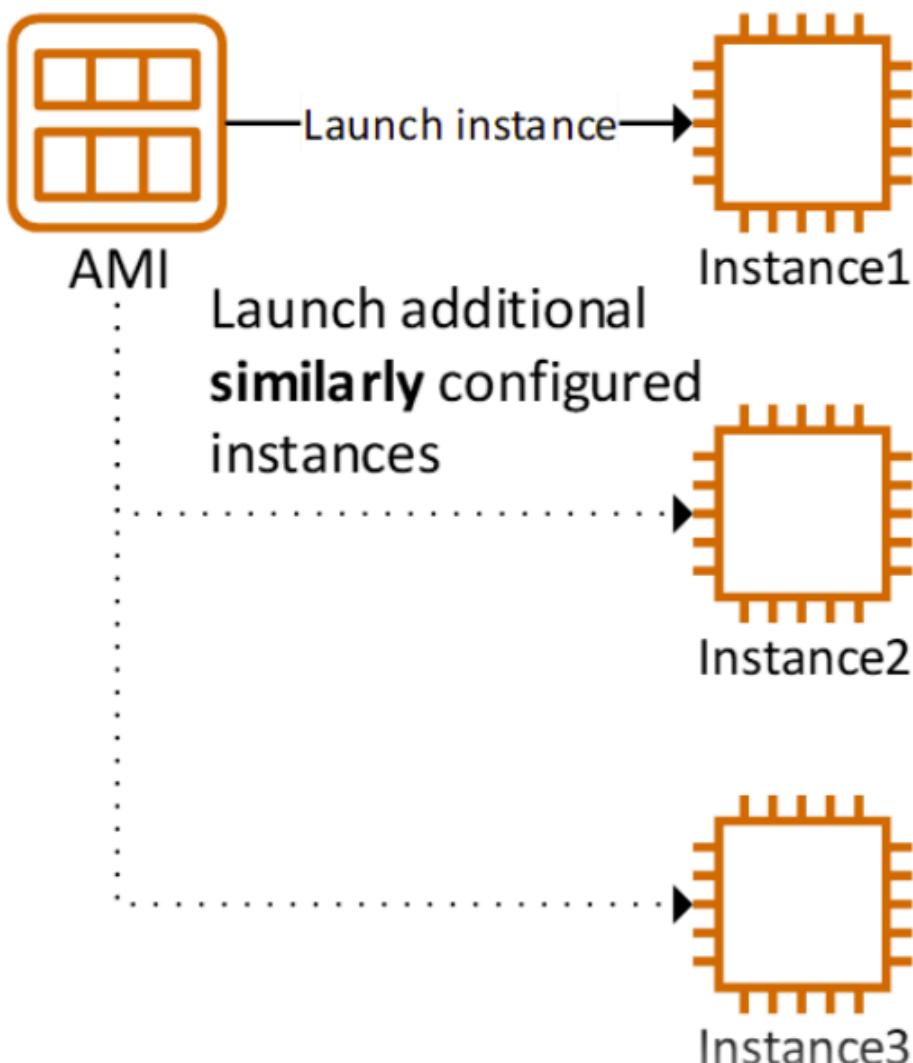


Amazon Machine Images (AMIs)

هي عشان ت Create Template بيهـا EC2 instance مع الـ OS, البرامج, الـ configurations وحاجات تانية.
أنواع الـ : **AMIs**

- الـ Quick start AMIs: بتبقا جاهزة من AWS زي Ubuntu, Windows Server
- الـ My AMIs: اللي انت عاملها مخصوص لـ application معين.

- ال Marketplace AMIs: بتبقا بتدفع عليها فلوس أو مجانية من ناس تانين.



الAMI بندى ال information الي محتاجها ال EC2 عشان تشغالها زي:

- يكون فيه OS فيها Image عشان يتحط على Root volume .
- ال AMI Access permissions مين يقدر ي

AMI Benefits

- ال Repeatability انه هيكونه عندي AMI واحد ويشغل منها عدد من Instance زي ما انا عايز.
- ال Reusability instance الى هعملها launch الى AMI من نفس ال Configuration هكون متآك من ال
- ال Recoverability تقدر تعمل AMI من Configured instance backup لي EC2 في حالة ال failure.

ال Ami تقدر تعمل منها Copy لي Regions ثانية.

⚠ Warning

لما تعمل Copy لي AMI بتاخد Snapshot بشكل Automatic لـ Region ثانية

لو ال Option مش شغال بيقا دا بسبب ان ال Instance مش EBS-Backed instance

Hands on

هتروج علی ال direction دا

Instances (1/1) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

All states

Name: AMI-Handon | Instance ID: i-0cb11ebc56d9dfcc2 | Instance state: Running | Instance type: t2.micro | Status check: Initializing | Alarm status: View alarms | Availability Zone: us-east-1b | Public IPv4 DNS: ec2-3-87-40-130.comp... | Public IPv4 IP: 3.87.40.130

i-0cb11ebc56d9dfcc2 (AMI-Handon)

Details Status and alarms Monitoring Security Networking Storage Tags

يكون عندك instance الي عايز تاخذ منها image

Instances (1/1) Info

Last updated 1 minute ago

Connect Instance state Actions Launch instances

All states

Name: AMI-Handon | Instance ID: i-0cb11ebc56d9dfcc2 | Instance state: Running | Instance type: t2.micro | Status check: Initializing | Alarm status: View alarms | Availability Zone: us-east-1b

Actions

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Image and templates**
- Monitor and troubleshoot

Create image
Create template from instance
Launch more like this

تختار ال instance وتختر ال create Image and templates وبعدين

⚠️ Important

لو ال Option انك تعمل instance مش شغال ييقا دا بسبب ان ال instance مش EBS-Backed instance

Instance ID
I-0cb11ebc56d9dfcc2 (AMI-Handon)

Image name
Custom-AMI

Maximum 127 characters. Can't be modified after creation.

Image description - optional
Image description

Maximum 255 characters

Reboot instance
When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/xv...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	3000		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag Image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

[Cancel](#) [Create image](#)

Amazon Machine Images (AMIs) (1/1) Info									
Owned by me		Find AMI by attribute or tag							
<input checked="" type="checkbox"/>	Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date	Platform
<input checked="" type="checkbox"/>	Custom-AMI	ami-0409300ae2321d93a	022499038054/Custom-AMI	022499038054	Private	<input type="radio"/> Pending	Q Q	2025/05/08 08:17 GMT+3	Linux/UNIX

هنا انت عملت ال AMI

Note

ال AMIs بتنبع نفس قوانين ال EBS في Sharing هنلقيها في

ال AMIs الي عليها share billing product codes مقدرش اعملها copied غير لو عملت instance من AMI دا و ساعتها اقدر اعمله

EC2 Image Builder

ال AMIs بيتاكل من Image BuilderCreation, Maintenance, validation, sharing, and deployment الخاص بي او Linux انه يكون سلس وميحصلش مشاكل زي بيعدي على Windows images Pipeline كدا.

الفائدة من:

- بياخد الحاجات المهمة في software و دا بيقلل ال security risks setting
- بيعمل custom tests AWS Tests لي بيعمل automated validation images قبل ما ينزل على .Production
- Version Control

Tip: Use EC2 Image Builder for automation and manage AMIs across accounts with AWS Organizations.

AWS Key Pair

هنشرح حاجه اسمها Encryption قبل ما أشرح ال Key pair

ال Encryption هو عبارة عن Algorithm بيغير شكل ال Message عشان تكون متشفرة والي يقدر يشففها هو الشخص الي معتوله ال Authentication (ال Message Key) وهو بيكون معاه Key عشان يقدر يعمل لي ال Decrypt Message ويشفف المحتوي .Message

Key Pair ال

وهنا هنلاقي أن في Two types من ال Key Pair

1. ال Symmetric: دا بيكون نفس ال Key الي معا ال Owner بيكون مع ال Client
2. ال Asymmetric: دا بيكون فيه Two Keys واحد Private (بيكونوا مختلفين عن بعض) و واحد Public

Amazon Keys في

بيتفا ال Client معاه ال Private Key هي اللي معاه ال Machine

User Data Script

تقدر تجهز ال machine بال Configurations الي عايزها عن طريق ال user data وهي أني اديها Script ولما ت create environment机器上就有一个叫 machine 的环境变量

ال data scripts بيستعمل بيشتغل بي root privileges او cloud-init directives او shell commands قبل ما ال network تشتغل على ال .machine

ال instance metadata دي معلومات عن instance بتاعتك.

- تقدر تعرفها من جوا ال Instance بتاعتك بي ال URL دا `/http://169.254.169.254/latest/meta-data`
- فيها ال DATA زى Instance ID او IPs او غيرها
- ال User data بتكون مترافقه في ال URL دا: `/http://169.254.169.254/latest/user-data`

لما يكون عندك Instance شغالة وعايز تعدل حاجه في user data تعمل ايه؟

1. تقلل ال instance
2. تخس على AWS Console في instance settings>Edit user data
3. امسح ال config_scripts_user file على ال Connect من خلال SSH او SSM
4. شغل ال Command دا

```
sudo rm /var/lib/cloud/instances/*sem/config_scripts_user
```

5. وبعدها يا تعمل Restart لي ال instance Re-run لي ال script يا تكتب دا

```
/var/lib/cloud/instance/scripts/part-001
```

عشان تتأكد ال data بي execute صح.

لو عايز تعرف ال output من user data script هنلاقيه جوا ال machine في ال path دا:

```
sudo cat /var/log/cloud-init-output.log
```

SSH to connect to a Linux EC2 instance

- تقدر ت connect مع ال EC2 بباتعاتك عن طريق ال SSH Protocol بس بتحتاج:
 - تعملوا download و تعملوا Key pair وتنديله .permission
 - تكون فاتح في ال Security group (حدد ال IP الي هيسعمل ال SSH) عشان ميحصلش attack عليك حتى لو لقدر الله و ال key اتسرق) بشكل دا:

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0b799df2607c4d164	SSH	TCP	22	My IP	156.218.38.179/32

Add rule Cancel Preview changes Save rules

النتيجة:

```
→ devops ssh -i "forgtech-keypair.pem" ec2-user@ec2-44-211-131-11.compute-1.amazonaws.com
The authenticity of host 'ec2-44-211-131-11.compute-1.amazonaws.com (44.211.131.11)' can't be established.
ED25519 key fingerprint is SHA256:+IYb4m/eH9u2/F20Tn5r4zgZenbZ0l9bi72U92L7HX8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-211-131-11.compute-1.amazonaws.com' (ED25519) to the list of known hosts

A newer release of "Amazon Linux" is available.
Version 2023.5.20240916:
Version 2023.5.20241001:
Version 2023.6.20241010:
Version 2023.6.20241028:
Version 2023.6.20241031:
Version 2023.6.20241111:
Version 2023.6.20241121:
Version 2023.6.20241212:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   _#
  /_###_
 /_####\      Amazon Linux 2023
 \##|_
 \#/  __->  https://aws.amazon.com/linux/amazon-linux-2023
  \_/
  /_/
 _/m/'
```

EC2 Purchasing/Launch Options

كل Model إلي هيتشرح علي حسب ال requirements و budget بباتعات الشركة.

1. On-Demand (most expensive)

الفكرة منه أنه Pay-as-you-go منغير أي commitments

مناسب لي short-term project و Unpredictable workloads

يقدم flexibility في أنك ت_start instances end لما تحتاجها من غير أي upfront او long-term contracts.

2. Reserved Instances (RIs)

يتحجّز نوع معين من Instance في كل الأوقات في region أو في Zone محددة بي commitments من سنة إلى 3 سنوات.

on-demand مقارنة بي ال خصم 72%

مناسِب لِي ال applications إلَي المستقرة أو ال predictable workloads.

في منها نوعين:

- ال RIs: ال Discount Standard family/type size up to 72% وتقدير تعدل ال فقط بس ال ثابت مش بيتغيروا.
 - ال RIs: ال Convertible convertible up to 54% وتقدير تعدل و تغير ال

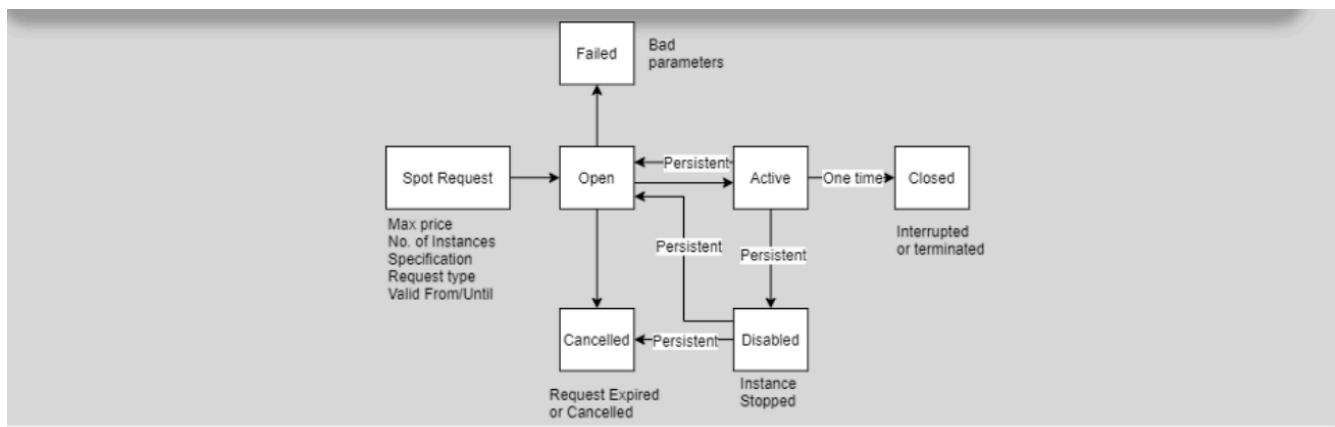
3. On-demand Capacity Reservation (no commitment)

يتحجز نوع معين من on-demand instance في AZ معينة.

4. Spot Instance

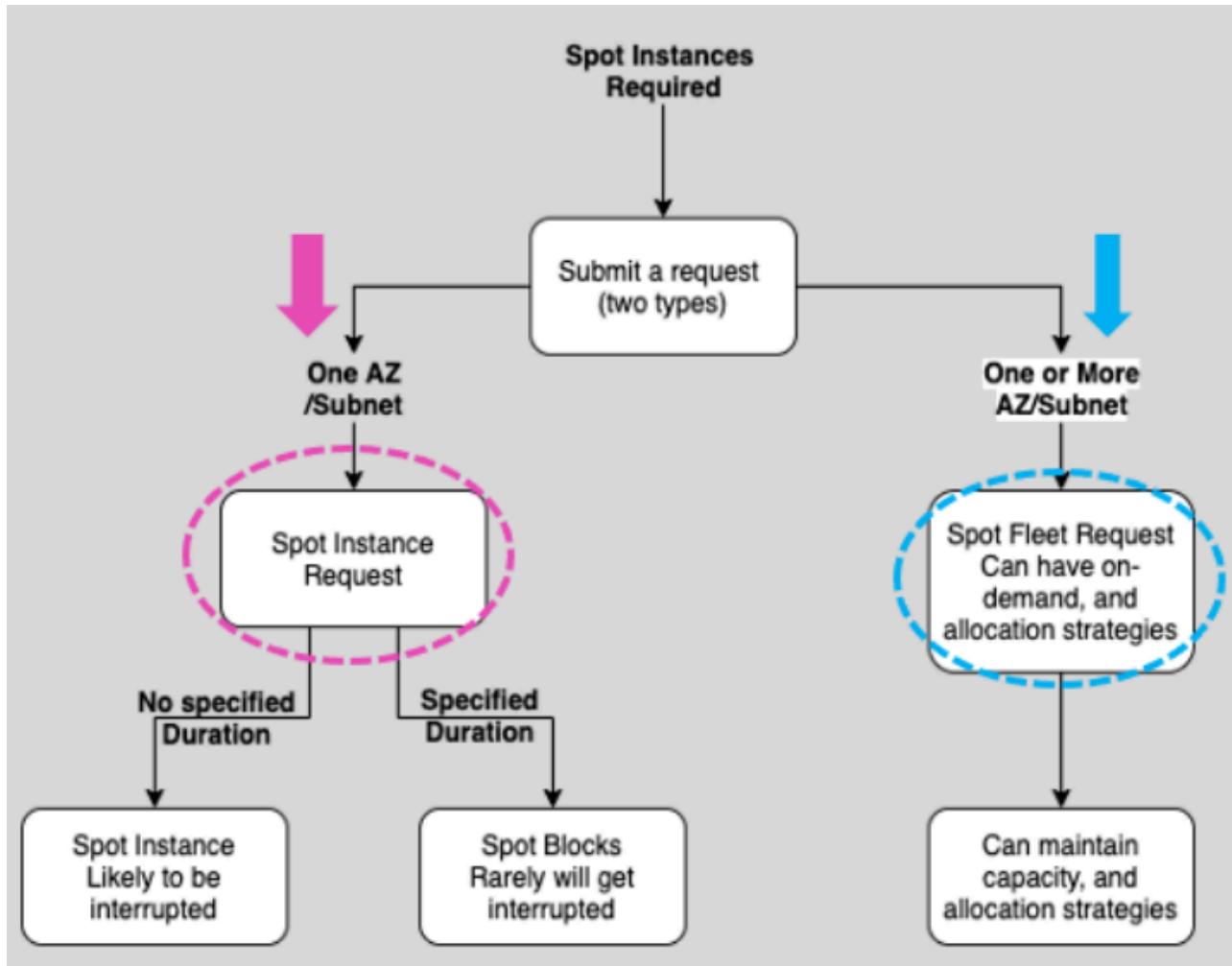
بقدر تستعمل ال unused EC2 capacity، مما يسمح أني استعملها بسعر قليل .
تقىم ممك يوصل لي 90 % مقارنة بي on-demand significant cost savings مشكلتها هي،:

- اي حد ممکن ياخده منك لو احتاجها (يعملها interrupted بس AWS بتعملك NOTICE قبلها).
 - مش بتضمن ال availability لما تحتاجها.



فی نوعین من ال:spot requests

- الـ One Time request: ان الـ spot instance يحاول بـ request بـ expire نفسه بـ terminate، بعد ما تشتغل الـ instance، ومش بـ request بـ terminate instance تاني لو الـ instance موجود لـ expire او يـ cancel او يـ terminate.
 - الـ Persistent request: لو الـ instance بـ terminate تانية تلقائياً، طالما الـ instance still active لـ request بـ terminate، لـ expire او cancelled.



عندی نوعین من ال Request :

1. ال **Spot Instance Request**: ودا في حالة ان عندی One AZ/Subnet وبيعدها بيتفرع منه:

1.1 ال **No Specified Duration**: ودي اني اعمل Request منغير فترة محددة وبيكون سهل انه يتعمل لـ interrupt instance في الفترة دي.

2. ال **Specified Duration**: ودي اني اعمل Request بفترة محددة وبيكون اسمها Spot block وبيكون من الصعب/نادر ان .spot instance يتعمل لـ interrupt.

2. ال **Spot Fleet Request**: ودا في حالة ان عندی Multiple AZ/Subnet او طلب fleet او كمية من ال spot instance mix ما بين ال spot instances (on-demand و optional) • بقدر تحدد من spot instances target capacity يعني لو عندی 8 spot instance الواحدة قفلت بيدور انه يعوضها ويخليةم 8 ثانی.

5. Saving plans

محددة بي commitments من سنة لي 3 سنين عشان استعمل كمية محددة من ال usage واحد يكون 72% Discount.

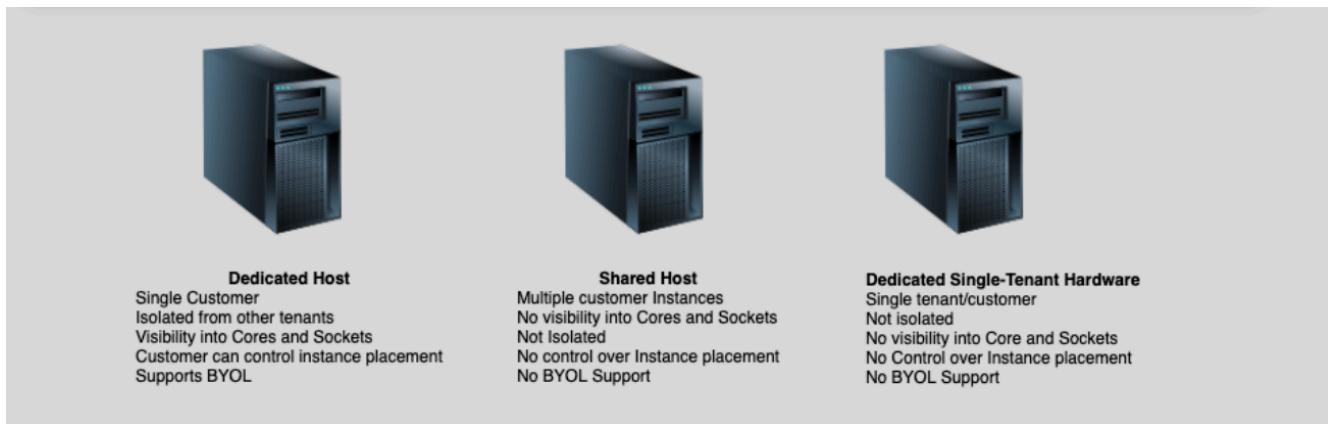
6. Dedicated hosts

customer بيأجر ال instance placement بشكل كامل ودا بيقدم isolation و control على ال physical servers customer بيلجي ليه في حالة انه معاه License وعليز يستعملها (BYOL)

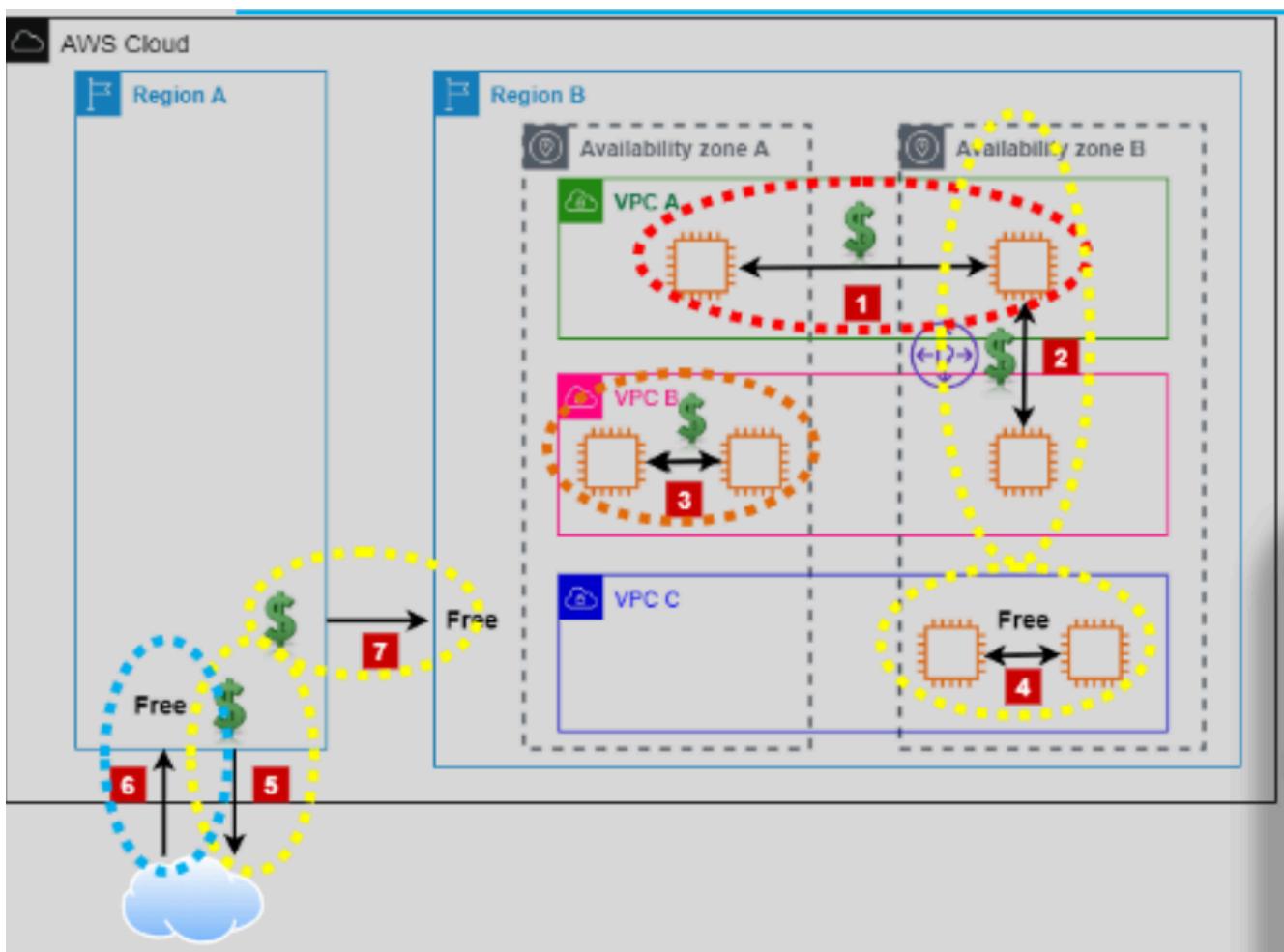
Dedicated host يكون Billing Reserved على ال on-demand billing by default commitment

7. Dedicated Instances

نفس كلام ال dedicated hosts مع اختلاف أنها تستهدف instances كامل يكون server كلها ليك, ال instances دي حد يقدر يستعمله في حالة انه بيستعمل اكونتاك.



EC2 Instance Billing



لو Data بتنتقل من region a لي region b زي رقم 7 في الصورة, الـ inbound traffic او الي داخل li region b مش هيحاسب عليه بس الـ traffic الي طالع منها هيالي region a

الـ traffic مابين two AZ مابين VPC مابين نفس الـ VPC بدفع في الاتجاهين زي رقم 1 في الصورة, نفس الشئ لو نفس AZ بس في VPC مخلفة زي رقم 2 في الصورة لأنه بيستعمل Peering connection.

لو VPC traffic واحد ونفس الـ AZ وبمستعمل Public IP او Elastic IP بدفع عليه في النحنيتين زي رقم 3 في الصورة.

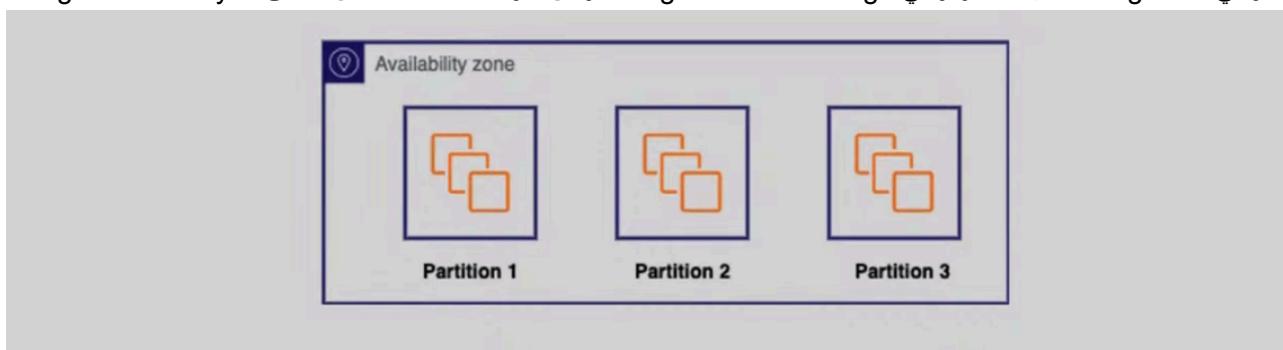
لو VPC traffic في نفس الـ VPC ونفس الـ AZ وبمستعمل Private IP مش بدفع اي حاجه في الاتجاهين زي رقم 4 في الصورة.

EC2 Placement Groups

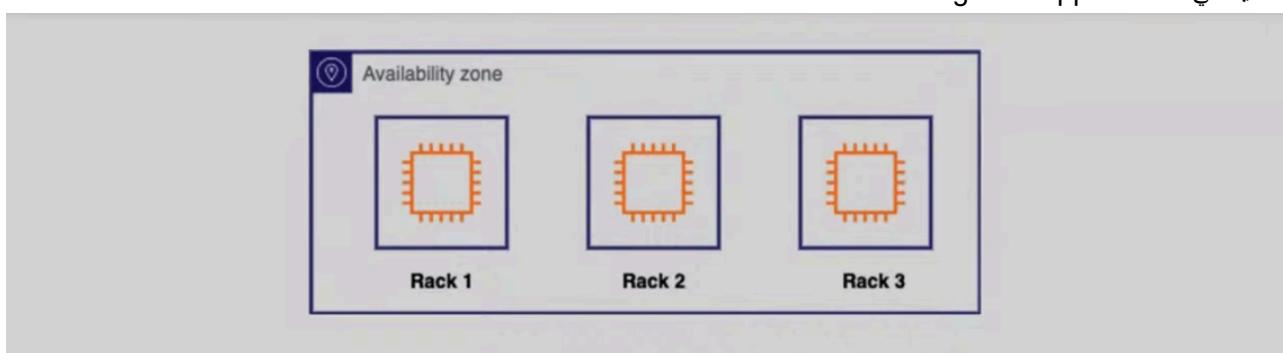
هي طريقة بتحطط بيها ال EC2 بشكل معين سواء بشكل قريب جدا من بعد (نفس AZ) او بعد عن بعض (AZ مختلفة) وهكذا. في العادي لما بتعمل launch لـ EC2 بتبقا Spread يعني متفرقة عن الثانية عشان تتجنب ال impact الي ممكن يحصل لو حصل failure لما يتحطوا بوضع معين ه يكون اي التأثير على ال Performance وغيرها بتساعدك عشان تعرف ال Instances لما يتحطوا بوضع معين ه يكون اي التأثير على ال Placement Group على حسب workload عندنا 3 طرق مختلفة عشان launch placement group.



1. ال Cluster: بتحطط ال Instances في نفس ال AZ, ال instance بتكون في نفس ال rack او مختلف بس نفس ال AZ ودا بيساعد في:
 - لو بتحتاج low latency
 - او High performance.High availability دا هيساعد بردو في a huge cost advantage



2. ال Partition: هو بيقولك ان rack دا ه يكون Logical Partition و ال instances هيعملها launches مختلفة (بتكون up to 7 per AZ), يعني لو عندك 20 instances هيطلوك كل 7 في AZ واحدة وجوا ال AZ دي ه يكون عندي partitions متقسمة racks ويتحط اكتر من instances جوا ال partitions دي .big data applications مفيدة لـ



3. ال Group: بيحط كل instance في racks مختلف، بتبقا اكتر من AZ واحدة، واخرى 7 INSTANCES في AZ الواحدة في كل Group

EC2 Status Checks

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
		i-049bc248699ed59fb	Running	t2.micro	2/2 checks passed	View alarms +
	dev-bastion-h...	i-01dc36cf7ab510e91	Running	t2.micro	2/2 checks passed	View alarms +

و دي ليها Two Responsibilities: و دا الي في صورة لو مبقاش 2/2 و passed بيقا المشكلة على AWS ممكن تكون Host في مشكلة او System Status Check: و دا الي في صورة لو مبقاش 2/2 و passed بيقا المشكلة على AWS ممكن تكون Instance status check في مشكلة او Instance status check: و دا الي في صورة لو مبقاش 2/2 و passed بيقا المشكلة على AWS ممكن تكون host في مشكلة او Instance status check.

- في حالة impaired/failed بقدر اعمل recover (دي الحالة الوحيدة) misconfigured startup: و دي مشكلة حصلت و يتطلب من customer انه يحلها, ممكن تكون cloudWatch configurations وغيرها, بتتبع لـ cloudWatch عشان تحلها انت و تحلها.

Relevant Pillars

Cost Optimization Pillars

Pillar One: Right Sizing

If you can't measure it you can't control it.

ممكن تبدأ بي Small size او Large size و تعمل Monitoring و ت Shawf CPU و RAM الأنسيلك.

لما تطبق الـ Right sizing خلاص تبدأ تعملوا Reserved عشان توفر.

Pillar Two: Increase Elasticity

.In Active Storages او Unused EBS Storages توقف الـ Auto Scaling.

تقدر تفعيل الـ Auto Scaling.

Pillar Three: Optimal Pricing Model

لما يكون الـ Workload بتاعك Variable متغير او غير معروف حالياً تقدر تستعمل الـ On-Demand او On-Spot.

لو عارف الـ Workload بتاعك خد الـ Reserved عشان تأخذ اللي تحتاجه وتستفيد بخصم.

خد فلأعتبر الـ Serverless architecture لأنه أقل Cost.

Pillar Four: Optimize Storage Compute

أمسح الـ Snapshots الـ Unused.

تخثار الـ Storage Priority حسب الـ Priority.

Security Pillar

Best practices:

Automate Compute protection

حاول ت automate الى تقدر عليه عشان تحمي ال compute, لأن ال human error بيفعل ال automation ويبخلي ال Security rules تطبق بشكل افضل على مستوى واحد بدل ما تنسى rule معينة في SG مثلاً وهكذا.

فا عندك ال EC2 Image builder دا بيساعد انه test EC2 وتعرف ال Vulnerabilities وبالتالي هيقول ال Security risks عندهك user data scripts لما تعمل instance launching لـ automate commands عشان ت اعمل لـ launching.

Control traffic at all layers

حاول تبقا بخبل او في rules في كل layer في Network بتاعتك وكل route table وكل subnet وكل sg وغيرها. وت monitor traffic عشان لو في حاجه unusual تقدر تحمي نفسك منها.

Performance Efficiency pillar

Best practices:

Scale the best compute options for your workload

اخثار ال compute المناسب لي workload بتاعك مش الأحسن عامة لا الأنسب عشان تتجنب ال infrastructure costs الى ملهاش داعي عشان تأخذ efficient performance.

Configure and right-size compute resources

حاول تختار ال size المناسب لي workload's performance requirements under/over utilized resourced وتنجنب ال workload's performance requirements.

Containers on AWS

قبل شرح ال Services فكرة ال Containers ببساطة عبارة عن OS Virtualization ودا في حاجتين:

- أول حاجه وهي Isolation عن طريق Namespaces ودي أنه بيكون عندي مجموعه من processes ويتكون خاصه بحاجات زمي نيتورك و storage وهكذا، وعن طريق ال kernel resources (يعني ال kernel resources manage) فـ containers يقدر يكلم مع ال kernel resources ويكون فيه isolation.
- والحاجه الثانية هي resource limits التي بيتم عن طريق cgroup التي بيحدي ال resources أخرى فيها قد ايه.

ال Container هي أني بجمع او packaging لي Code Environment في صندوق واحد وأقدر أنتقل بيها من أي Device لي الثاني منغير متوجهني مشاكل ال Dependencies و من أشهر ال Tools Docker التي بتعمل كدا هي.

ال Images عبارة عن Layers زي ال Minimal OS و Code وهكذا.

وعشان أقدر اعمل Orchestration لـ Containers التي عندي في حالة Production عشان كمية ال Containers بتبقا كبيرة ساعتها Kubernetes زمي Tool هاجي له.

Amazon Elastic Kubernetes Service(EKS)

دي خدمة AWS من Managed Kubernetes لو عندي:

- الـ AWS Migration عشان أستفيد بالـ **Scalability** اللي عايز عمل On-premises Containers لـ AWS.
- أو عايز عمل Kubernetes Cluster عموماً على الـ Cloud.
- أو حتى عايز AWS تساعدني أدير الـ on-premises clusters عندي.

في أول حالتين، هحتاج أعمل Deploy على EC2 أو Fargate

- الـ EC2 بيكون Nodes عادي أنت مسؤول عن إدارتها وصيانتها.
- أما الـ Fargate فهو Containers لـ Serverless يعني AWS بتدبر البنية التحتية كلها، وأنا كل اللي عليّ أشغل الـ Pods وخلاصن.

الفرق هنا زي الفرق بين إنك تشغّل Instances بنفسك أو تسيب AWS تشغّل Lambda، لكن خلي بالك: Lambda بتشغّل Functions قصيرة العمر.
الـ Fargate بيتشغيل Containers كاملة وتفضل شغالة طالما أنت محتاجها.

اما في الحالة الثالثة، فممكن استخدم EKS Anywhere أو أعمل Integration مع EKS Connector عشان أقدر أتابع وأدير الـ AWS Clusters اللي برة AWS وكأنني شافيهما من جوة الـ AWS Console.

Amazon Elastic Containers Service(ECS)

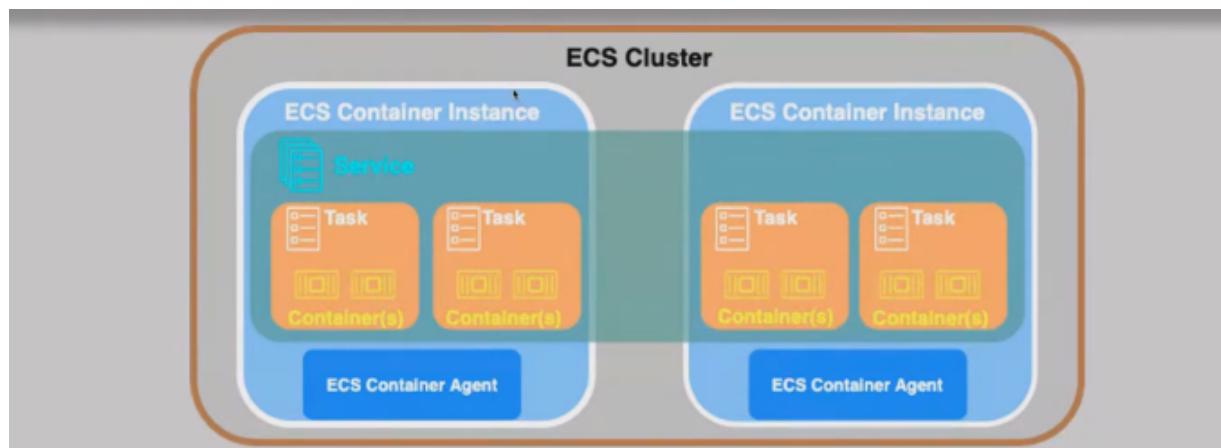
دي من AWS بتكون Fully Managed Service ، بحتاج منك يكون في Container Image جاهزة عشان تستعملها.

فا هي بتأخذ Container image دي وتحطّها جوا نوع من النوعين اللي هي بتتوفر لهم وهما:

- الـ EC2 Instance + ECS Agent حواها ECS Instances.
- الـ Fargate ودي عبارة عن Serverless-based containers.

في حاجه اسمها ECS Cluster ودي عبارة عن Grouping لي مجموعه من ECS Instances او Faragte.

Task Definitions and ECS Services



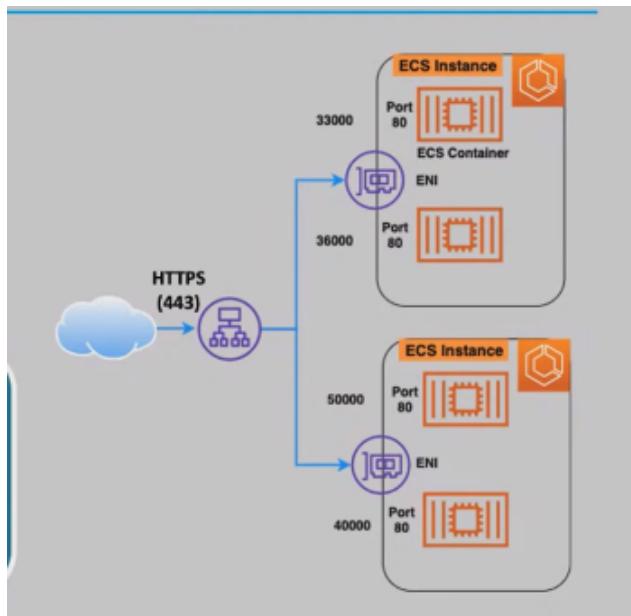
الـ Task Definition هو الـ Template او Configuration اللي هتقوم بيه الـ Container زي عددهم والـ Image دي فبن سواء على Fargate او ECS Instances وتحتاجه قد ايه في memory per container ونوعها زي ECR or Dockerhub. تقدر تقوم لي 10 من containers من Task Definition.

الـ ECS Service هي الـ mechanism اللي بيشغل ECS Task ويعملها تحت الـ ECS Cluster اللي انت محددهالوا.

الـ ECS Auto scaling في منه نوعين:

- الـ EC2 Auto scaling Group ECS Instance على مستوى الـ Auto scaling.

2. ال Application auto scaling (service auto scaling) ودا علي مستوي Task او container حسب demand and utilization.



ال ALB: تخيل معايا عندنا ECS Cluster فيها Two ECS Service بتشغله Two ECS Instance تحتها على ECS and ALB وليكن على Two EC2 Instance فا انا كدا عندي Two ECS Instance وحطتهم ALB ي Load مابينهم على Ports مختلف عادي. طب لما يخش جوا ال ECS Instance دي هيلاقى في Containers موجوده بنفس Ports على سبيل مثال كلها 80 ؟ ساعتها متلقش لان في حاجه اسمها Dynamic host port mapping ودي بتوزع Ports مختلفة على containers من نفس ports وهي الي هيكون شايفها ALB فا يوزع Load هو مرتاح. في حالة انها كانت Fargate مكتنش هتبقا مشكلة لأن Fargate بتعامل بي ان كل container ليه ENI واحده ودا بيسهل كثير على ال ALB.

⚠ Warning

مع Dynamic host port mapping وانت بتعمل ECS Instance خليها بي 0 عشان يقدر يعمل Hostport لـ EC2 بناءً على Configure ALB مع mapping

Amazon Elastic Containers Registry(ECR)

وهي المكان الي بـ Store على AWS Containers Images

ولو شغال على Docker Hub بقدر Integrate معهاها

AWS Batch

بيبيقا عندي processes يحتاج اشغلها في وقت معين وبتحتاج compute عالي او instances كتيره.

ال AWS Batch هي Fully managed regional service ,di محتاجه tasks,batch processing jobs ,بتبسيط ال Region .multiple AZ في نفس ال resources كبيرة, بتنشغل في اكتر من

:characteristics

- ال Plan: يحدده ال Job المطلوبة لي كل Resources

- الـ **Schedule**: يتحدد أتى تشغيل الـ **Jobs**
- الـ **Tasks** على **compute environment** executes مناسبة

:Use cases الـ

- في الـ Big data
- في الـ images transfer videos او simulation
- في الـ model training

Amazon Load Balancer & ASG

1. Auto Scaling Groups (ASG)

- ايه هو الـ **ASG**؟
- هي Service بتحكم في عدد الـ EC2 instances بناءً على الـ Workload (زي الـ Request أو الـ CPU).
- ازاي بيشتغل؟:
 - الـ Horizontal scaling
 - الـ Scale Out: يضيف instances جديدة لو الـ Workload زاد (مثال: لو الـ CPU وصل لـ 80%).
 - الـ Scale In: يقلل الـ instances لو الـ Workload قل (مثال: الـ CPU نزل لـ 30%).
 - الـ Vertical Scaling
 - بغير نوع الـ Instance بي Type أعلى بي CPU أعلى و غيرها
 - الـ Health Checks: بيعتبر الـ instances الي مش Healthy بوحدة تانية سليمة.
- الـ Characteristics
 - الـ Cost Optimization: بتدفع بس ع لى الـ instances اللي تحتاجها.
 - الـ ELB مع Integration: أي instance جديد بيتسجل في الـ Load Balancer Automatic
 - الـ High Availability: بيوزع الـ instances على أكثر من AZ.
- Use Cases
 - لو عندك Workload مش ثابت (Zig-Zag) (daily or weekly variations)
 - الـ Cyclical traffic patterns
 - الـ on and off traffic patterns
 - الـ variable traffic patterns

Application Auto Scaling

- ECS Services
- Spot Fleet requests
- EMR Clusters
- AppStream 2.0 fleets
- Aurora Replicas
- DynamoDB Read and write capacity units
- Sagemaker endpoints
- Amazon comprehend

EC2 ASG - Components

هو عبارة عن Logical grouping لـ managed instance

يحتاج (Template recommended من aws من recommended launch configuration or template) انك تستعمل ال

Scaling Policy ودي اعتبرها الخطه او ال Plan .when and how to scale .ويبي اعتبره ايها يعمل ايه, يعني

Info

الـ EC2 Integration health checks by default بتشوف الـ ELB مع ELB شافت ان في disabled by default EC2 terminate Healthy EC2 مش

بتقدر تحدد الـ Min و الـ Max , ببغا عندك Current State بيقارنها بالـ Min و ميعلاش عن الـ Max بيكون دول الـ Range بتاعك, ولو معنديش scaling policies ساعتها الـ Min == Max وساعتها مش هيكون ليهم لازمة, الـ Desired مش هو ولكن بيكون مابين الـ Min و الـ Max.

EC2 ASG - Features

هو Regional Service وانت الي بتحدد الـ AZs الي هيكون فيها.

الـ ASG بيحاول يـ Balance على مستوى الـ AZs الي معاه

الـ ASG بتـ Integrate مع الـ ELB , Cloudwatch, and Cloudtrail

EC2 ASG - Launch Templates & Scaling Policies

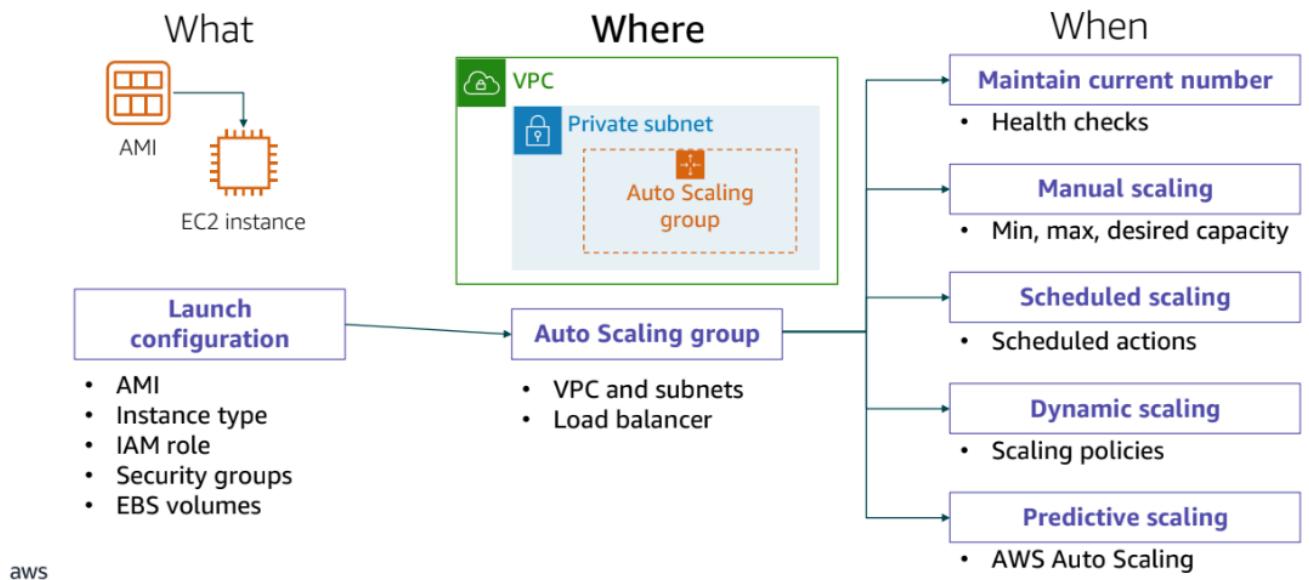
الـ Launch Templates هي هي Launch Templates ولكن لو عندك اختيار مابين الأنتين اختار الـ Launch configuration ليه؟

1. لأن Templates في فيها different version . فا يكون عندي من نفس template version مختلف واستعمله.

2. اقدر استعمل أنواع مختلفة من instances type و اقدر استعمل on-demand و spot instances في نفس ASG , دا هيساعد اننا .desired scale, cost , and performance

3. في اقدر اعمل template group ليها زي ما كان مشروع في Placement Groups

How Amazon EC2 Auto Scaling works



ال Dynamic Scaling يعني لو حصل event معين ال scaling يشتعل.

• بيشتعل عن طريق alarm معين و cloudwatch generates هي اللي بتعملوا وعلي اساسه هي scale out/in

• عند 3 أنواع:

1. ال **Simple scaling**: يعني لو حصل rule واحد زي أن لو ال cpu instances زود %70

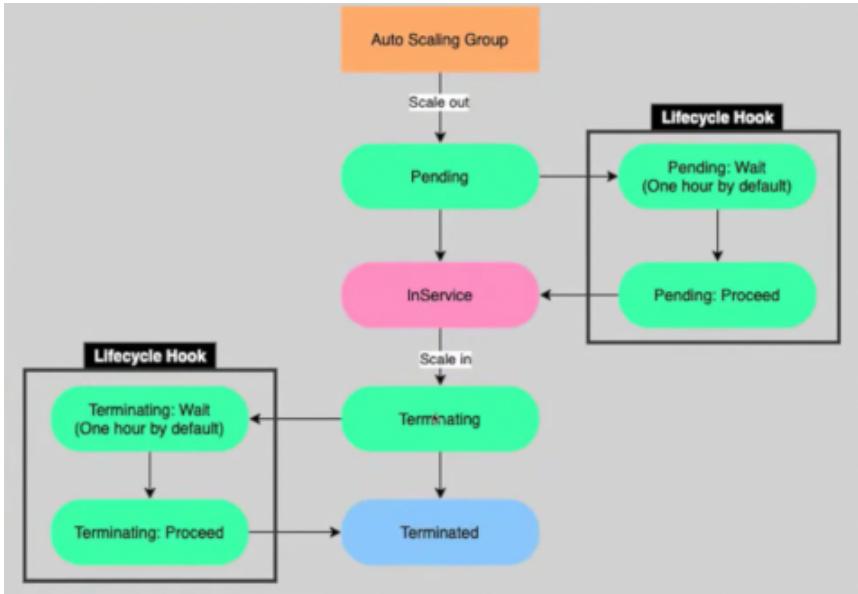
2. ال **Step scaling**: يعني لو حصل alarms حسب ال multiple steps/adjustments two instance ووصل %70 زود لو instances ووصل %80 زود two instance وهكذا.

3. ال **Target Tracking Scaling**: يعني طريقة بتخلي AWS يراقب متوسط استخدام CPU لكل instances ، وإن كنت بتحددله Target معين زي مثلاً 60% ، وهو بيبدأ يزود أو يقل عدد instances علشان يحافظ على المتوسط ده حوالي ال 60%. ال **Scheduling** يعني في وقت معين بيحصل ال scaling . ال **predictive scaling** دا بيشتعل بناء على توقع معين باستعمال AI.

Amazon ASG - Lifecycle Hooks

ال **Lifecycle Hooks** بتخليك توقف ال EC2 instance أو مؤقتاً في مرحلة ال launching أو terminating ، علشان تنفذ عمليات معينة زي إنك تنزل ال logs قبل ما ال instance terminate . Group

بتشغل عن طريق إن AWS بتحول حالة ال instance لـ **Pending:Wait** أو **Terminating:Wait** ، وبيسنن لحد ما إنت تبعته signal (مثلاً من Lambda أو SQS) تقول إنك خلصت. ولو ما بعدتش signal في وقت معين، AWS بيكمel العملية تلقائياً حسب ما انت



Amazon ASG - Cooldown and Warm-up periods

الCooldown (دا بيكون علي مستوى الASG): دي الوقت الي هيستناء ال auto scaling بعد ما بيحصل scale out/in عشان يعمل scale out/in تاني.

ال instance warm-up period (دا بيكون علي مستوى INSTANCE): دي الوقت ال محتاجه قبل ما تخش في instance metrics .cloudWatch

Amazon ASG - Scale-in Termination Protection

دا option بيخليك تحمي ال Instances من automatically terminated instances scale in, يعني وقت scale in مش هيخلقي ال instance تتشال.

ال instance دا مش بيحمي من option:

- Manual termination
- replacement if it become unhealthy
- spot instance interruption

ال Termination Policy: لو عندي واحد فيهم عندها عدد من instances select كبير هي عملها launch templates ويعمل منهها launch configuration terminates instances مابين terminate , لو عندي ال instances terminate launch configuration best practice , لو كلها كدا مش من configuration oldest first , لأنها كانت اولى .best practice .Billing same في كل حاجه ساعتها هي terminate الى قربت فيهم على أنها تكمل ساعة عشان متحسبش على .Billing

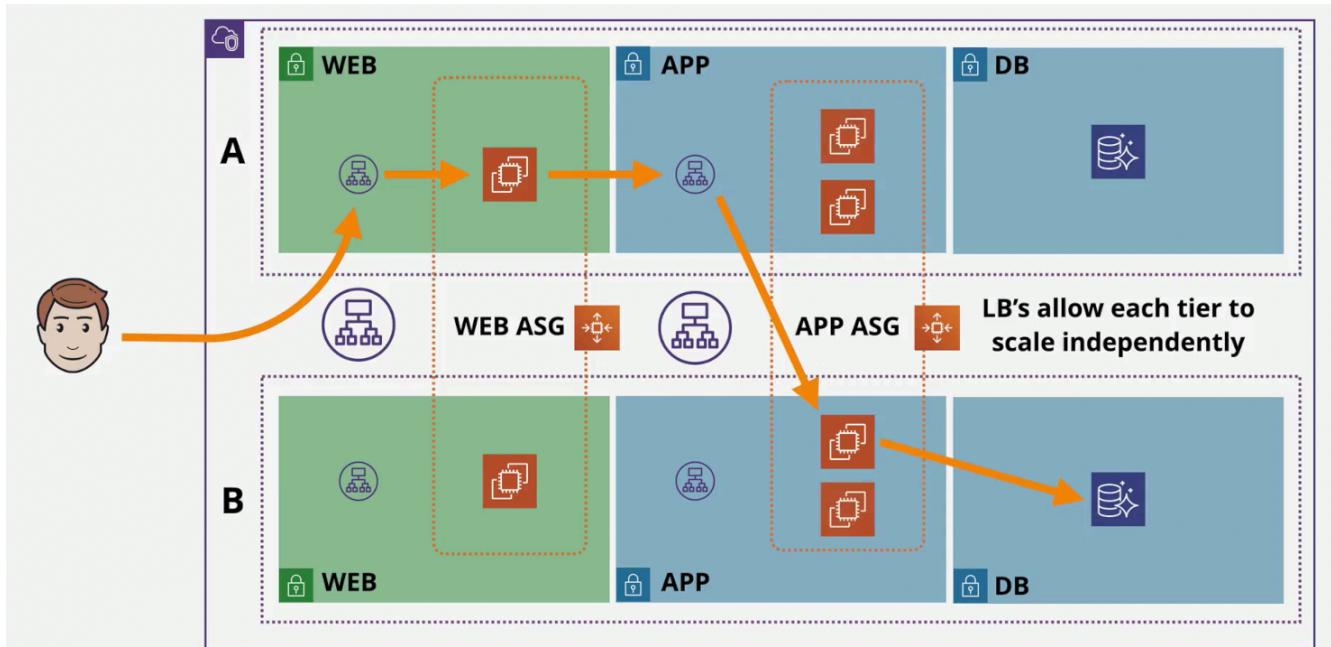
2. Load Balancers

- ايه هو ال Load Balancer ?
- هي ال traffic بتوزع بين Service مجموعة من instances EC2 على واحد. وهي Regional Service
- ال Characteristics

- الـ DNS (عن طريق Single point of access).
- يعمل health checks على الـ instances traffic لـ Instances السليمة وعليها load أقل.
- بي SSL/TLS (HTTPS) support الـ High Availability عن طريق تشغيله في أكثر من AZ.
- يوزع الـ traffic عن طريق أن هو بيقا فيListeners وهو بيعت عليهم الـ Health check.
- **أنواع الـ AWS في Load Balancers**

النوع	الاستخدام	(Layer)	ملاحظات
Application Load Balancer (ALB)	HTTP/HTTPS (تطبيقات)	Layer 7	يتتعامل مع target resources عن طريق الـ target group ويقدر يكون عنده اكتر من target group
Network Load Balancer (NLB)	TCP/UDP (أداء عالي)، gaming, streaming	Layer 4	نفس كلام الي في ALB، مناسب له Millions Request في الثانية يعني بيديني More Speed في حالة انك اخرك layer 4.
Gateway Load Balancer	Third-party appliances (زي firewalls)	Layer 3	.VPC appliances traffic

ELB Architecture



الـ User يجي عن طريق Load Balancer DNS ويروح لي (Loadbalancer Node) بتكون ENI من الـ ENI التي انت عرفتها في configuration, وبعددين يوزع الـ traffic على target الي عندي وفي الصورة الـ target هو الـ Autoscaling group.

وبيكون فيه نوعين من ELB:
الـ Internet-facing: ودا لو جيلك traffic من الـ internet
الـ internal: لو بكلم مع resources منغير ميطلع لي internet

Recommended

من AWS Recommendations أك تعرف nodes بنا عن AZs و subnets مختلفه،
وال subnets يكون فيه +8 IPs زياده عشان يعرف يscale

ELB Target Groups

هي يعني لما create target group هنفياً متعارفه جوا ال Region بنا عنها فقط.
ال ALB/NLB register مع Target هو Endpoint معموله
ال resources that VPC Peered, On premises servers في instances target أو IP Target
ال ممكن استخدمه عشان أ can be addressed by IP and port
ال ALB/NLB بي اكتر من Route traffic لي
ال اقدر اعمله target group مع registered target
.different ports اكتر من مره بأسعهمال

ELB Listeners

بنقسمه نقطتين:

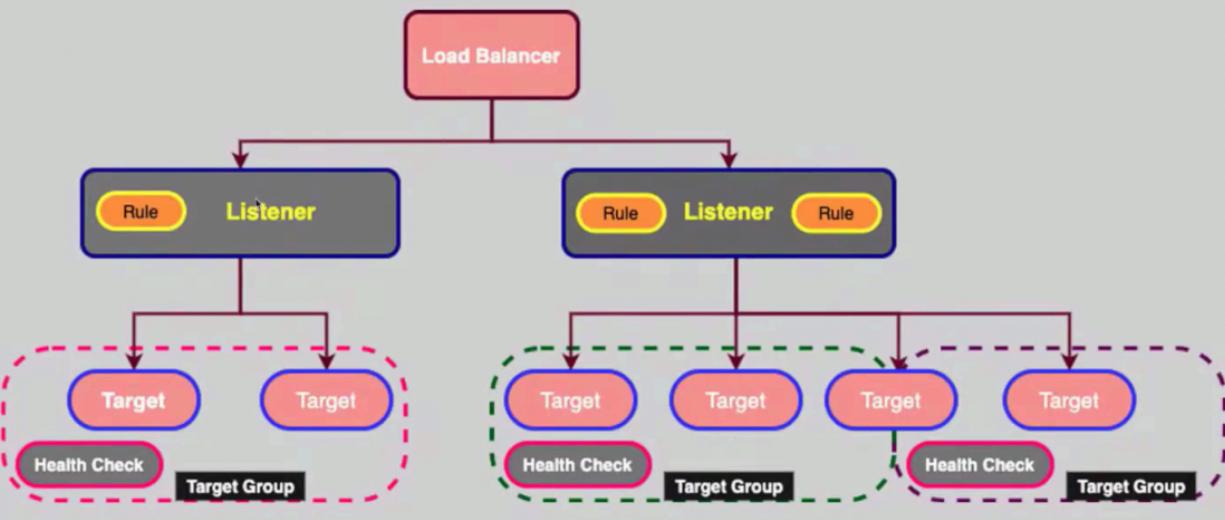
- ال Frontend: دا الجانب الخاص بي ال users
- ال Backend: دا الجانب الخاص بي ال resources

ال listeners هي process بتتأكد من connection request الى رايحة على ال ELB Node وبتحتاج تعرف تتأكد من أنهي port وأقدر
اعرف اكتر من listeners .

ال ELB بيعت Health checks عشان يتأكد من resources الي هيعت عليها ال request شغاله ولا لا, يتأكد من ports و thresholds

ال ELB يتحقق ال DECOUPLES وهي أني اعزل ال failure عن بعضوا عن طريق انه يبعث health checks لو مش شغال مش هيعت
ال resources request وال الثانية مش هتصدر.

Target Groups & Rules



ال listeners و target groups بيشتغلوا مع بعض ازاي؟
ال listeners بنتيجي ب rules وعندما request يجي تبعته لي rule default و هي اي target group و تقدر تغيرها وتعرف rules مختلفة او multiple rules.

Note

لو كل الي عندي مش matched ساعتها هروح لي default rule

في حاجه اسمها content routing path وهي اني عندي نفس ال IP و Targets ولكن هيديني علي نفس path مختلف.

ELB and Cross-Zone LB

:Cross-zone قبل ال traffic من internet جي من ال instances و يبدأ يتوزع على ال ENIs او ELB Nodes ، وبعدين يشوف ال instances الي عندي في نفس ال AZ و يبدأ يبعث ال Traffic عليهم، طب لو عندي instance واحد فقط في AZ ؟ هيبعد كل ال Traffic عليها.

عيوب الكلام دا أن الميزة الي كنت بتحاول تستغل ELB عشانها وهي ال High available مبقاش ليها لازمة

:Cross-zone بعد ال traffic من internet و يبدأ يتوزع على instances الي عندك حتى لو في AZ مختلفة المهم انه يكون شايفها ويقدر ي route علىها.

Info

ال traffic يبيغا Cross-zone active by default لما Application LB create لاما need to have active by default across zone على ال traffic الي راجح

فما ال traffic يخلي ELB Node بتحاول تخلي ال traffic الي جيلك يتوزع بشكل متساوي علي ال instance الي نفس ال AZ ومع cross-zone traffic يتوسع حتى على AZ المختلفة.

ELB and Connection Draining

لو عندي instance من ELB الي تحطسيطرة ال traffic او اي troubleshooting عليها عايز اعملها علها delay time بس عشان يبدأ يعملها deregistration او يطلعها من انها تكون عشان متسقبلش ال sessions الحالية فا تقدر تستعملها براحتك.

ال deregistartion delay.seconds by default 300 بببا

ELB - Subnets & Design For High Availability and Scalability

عشان تعمل HA يكون Scalable need requires (Two public Subnets) في Two AZs عشان مختلفين عشان HA.

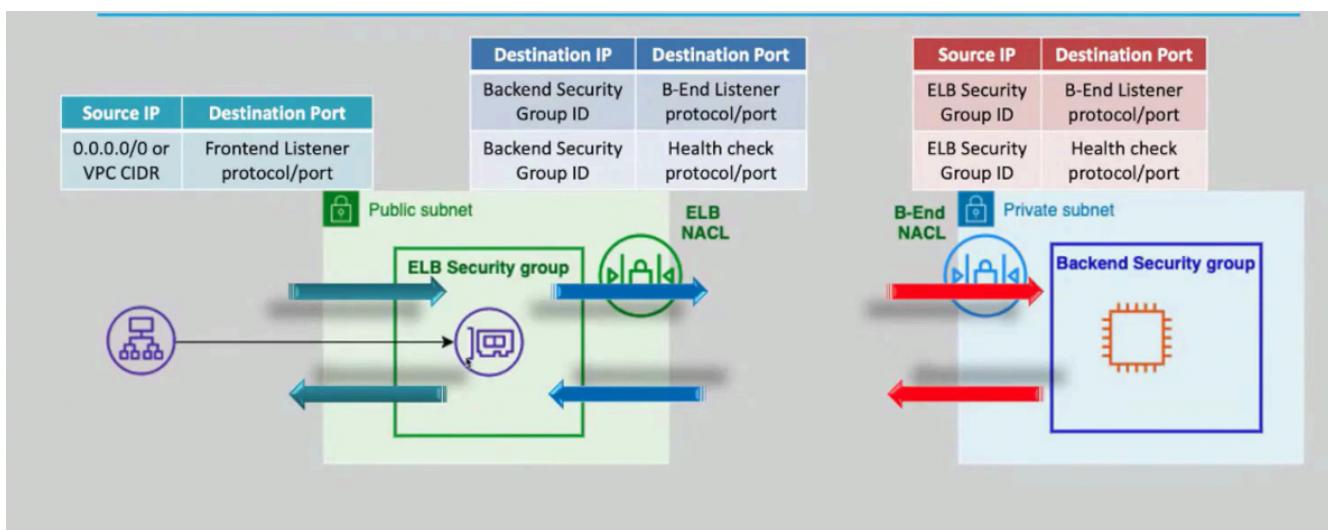
لو عايزه يكون Secured يبقا تحط ال LB Nodes عن طريق application private subnets وتكلم مع app بتاعك في .NAT Gateway احتاج ال application بتاعك انه ي communicate مع internet عن طريق Private IP.

ELB - Security Groups

ال Application Load balancer لازم اعمله security group واشوفه هيستقبل ال traffic على انهي port مسبهاش Open, وهاد ال security group rule بناعت ال LB احطها في instances وعلى ال port الي هو بي listen على عشان user request و يصل ال response.

⚠ Warning

ال Network Load Balancer ملحوظ انك تعمله Security group handle IP Nodes NLB تحتاج تحط ال APP/Backend Security group في Subnet CIDR-Block.



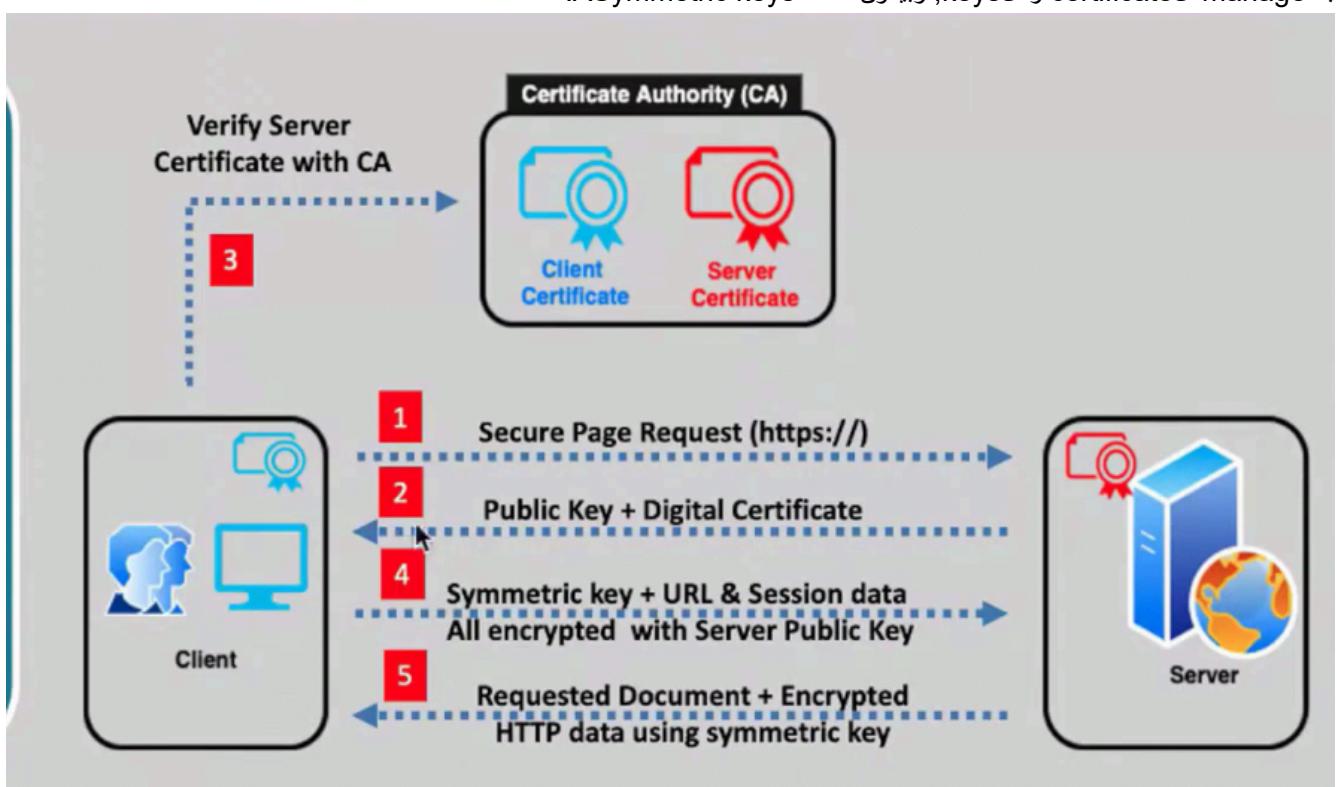
يبقا ال ALB source IP يكون 0.0.0.0/0 لو هيكل مع VPC Internal security group او VPC-CIDR مع تحديد ال Listener Port طب دا كدا لما traffic يعدي علي Destination port الخاصه بي LB.

لما يطلع منها هيبيت لي two destinations, لي Destination Application/Backend security group بتاع ال two destinations وهو ال Destination port health check port (دا في حالة ان health check port هتكون في Destination port مختلفه).

وال APP/Backend security group بتاعها source IP ه تكون ال LB security group ID بدل متعامل بي IPs.

ELB and SSL Certificates

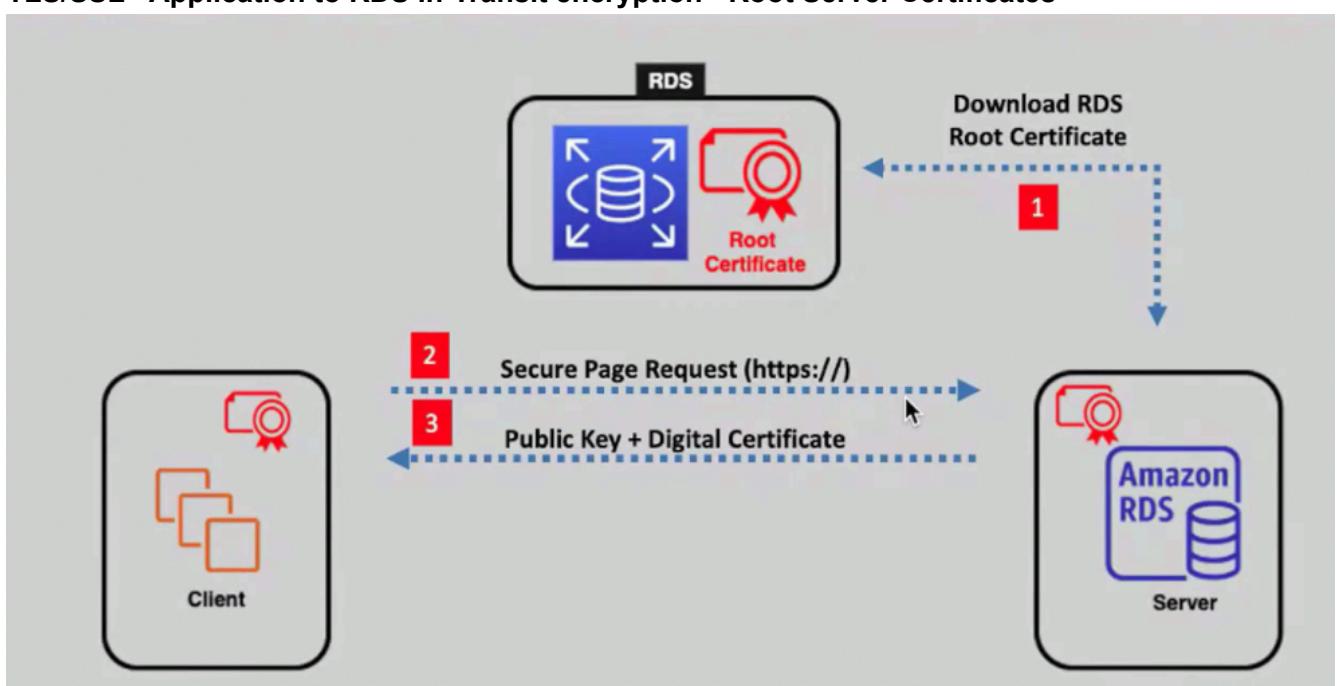
Right Company او Digital ID عشان ال User لما يسجل بینتوأ يتأكد أنه بيكلم مع AWS Certificate manger عندها خدمة زيها اسمها AWS Certificate manager . ASymmetric keys و certificate manage بتكون معها keys .



الي بيحصل انك بتطلب Secure Page request من ال server فا بيعننك Public key + digital certificate ولو هي عندك (ساعات تكون متخزن في server عندك مين ال authority) لو مش عندك هيروح علي certificate authority ويتتأكد منها ولو تمام هتنبع . و سيرفر هيرد عليك بال data الي محتاجها .

ال (PFS) Perfect forward secrecy وهي انها تغير ال Key كل فترة (ephemeral session key) عشان لو اتسرق او حصله application يبقا ال unauthorized third parties compromised من امان بتعاي في

TLS/SSL - Application to RDS In-Transit encryption - Root Server Certificates

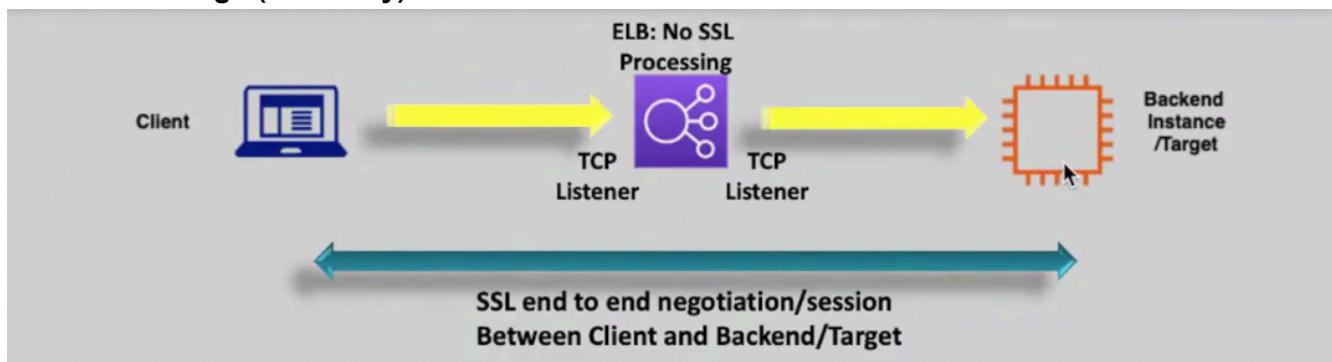


لو عايز اعمل Certificate مابين RDS و ال Application, فلأول لو انا عرفت ال RDS انها هتشتغل بي SSL Certificate ساعتها هينزل مع RDS Service من Root Certificate ال RDS.

SSL/TLS Offloading

يعني أن ال SSL/TLS Handshake تحصل عند ال Loadbalancer وبعدين يبعثها لي instance. الميزة انك هتلحد performance احسن على instance لأنك قلللت ال operations, ال content routing لأنها هيملك ال HTTP .path-based routing header.

TCP Passthrough (NLB Only)



وهي اني اخلي الي نفسها الي فيها ال SSL وتفك ال request من Client, يعني ال Listener Loadbalancer هيكون TCP instance. و من لي Target Listener هيكون TCP, و Session هتفك لهاها وبعدين يبدأ SSL يتفك عن طريق ال Instance.

Multiple Certificates using Server name Indication (SNI)

هو عبارة عن أني اقدر احط اكتر من digital certificate مع ELB وعن طريق ال server name indication () والي هي موجوده بقاعدت ال request هبيعت ال DNS الي مطلوب headers.

فا علي سبيل المثال عندي اتنين CERTIFICATE لي www.example.com و www.cloudjourney.com و الآلترين علي ال LB وقولت LB انا عايز اخش على www.cloudjourney.com هبيبدأ LB يبعطي ال certificate بتعمدوا ومنها هفتح ال session على الموقع دا وهكذا مع الموقع الثاني.

ELB & Client (Source) IP Address - Proxy Protocol & X-Forwarded-For

في العادي ال Loadbalancer بيكلم مع ال application عن طريق ال Private IP . لو عايز اتعامل مع Client IP واعرف ال ALB مع X-Forwarded-For Source IP . وهي ALB enabled by default.

Info

لو انت layer 4 وشغال بي NLB ومعرف ال target instance id في الحاله دي هبيعت ال Client IP بشكل Automatic ولو شغال علي ip ساحتها تحتاج ال proxy protocol.

عنوان يعرف يقرأها من Instance ولازم ال application يكون بيعرف يفك ال payload دي ويقرأها عشان يستعمل ال IP (ليها). علاقه بي developer.

ELB Monitoring Logging

في Cloudwatch metrics: بيعت cloudwatch كل دقيقه لو في traffic بيحصل على ال ELB Node كل دا (by default).
في CloudTrail: كل API Calls الي بتحصل لي ELB API هتنبع.
في Access Logs: معناها ان ال ALB هيدأ بيعت علي ال actual traffic الي رايحه لي ال application (بستعمله لو عايز data على user والوقت و ip).

ELB - Session Affinity (Sticky Sessions)

ال affinity هي أن ال session الي جاية من user تكون مع target على طول ولو target مش بي (يعني share data ساعتها stateful).
لو instance وقعت فا ال session هتقع معاها ، ولو instance يعني كل share data ساعتها هقدر يروح لي instance تانية في حالة الأولى وقعت.

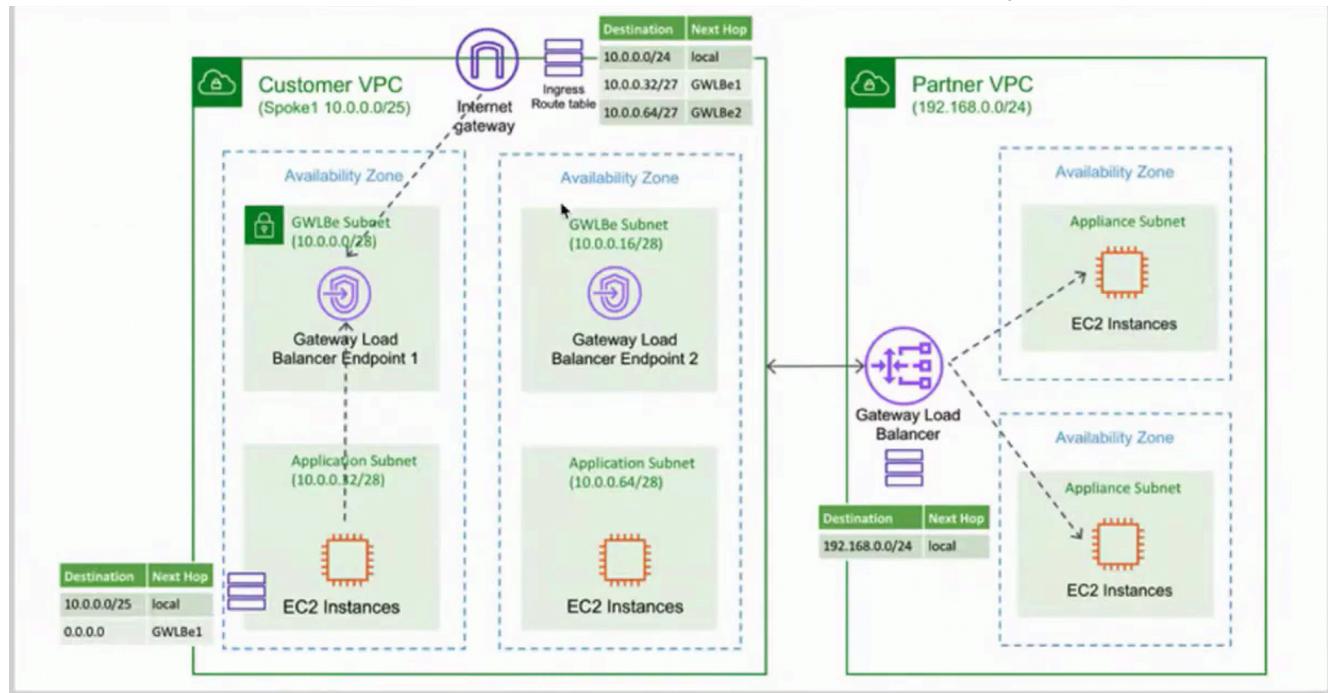
ال ALB بيدعم:

- تقدر تضيف عليه ال AWS WAF
- ال Web Sockets by default
- ال least outstanding requests algorithm و ال round robin algorithm (default) الفاضية.
- ال Path & Host-based routing
- ال path وهي نفس ال domain بس ال path مختلف يعني ال domain دا <- www.cloudjourney.com يوديني على two target groups مختلفين على حسب ال path يعني لو اتبعد كدا <- www.cloudjourney.com/image يروح على target group 1 ولو كدا <- www.cloudjourney.com/video وانا بستعمل نفس target group 1 ALB
- ال host نفس كلام بس هغير ال www يعني هيكون في target group توديني على sales.cloudjourney و target group توديني <- www.cloudjourneye ALB
- ال Slow start mode: وهي supported في ال ALB فقط، وهي انها تستوي ال healthy Instance تكون healthyInstance تأخذ target group configuration وهي على مستوى ال requests enabled by default users ولكن ال access logs, and delete protection disabled by default

ال NLB بيدعم:

- ال ELBs مابين كل ال lowest latency و highest connections per second
- ال TCP & UDP (هو الوحيد الي بيدعم UDP)
- تقدر تحطله Elastic IP في كل AZ (دا Optional)
- مش بتعلموا Security Group
- ال Delete protection disabled by default كل دول Access Logs, Cross-zone LB, and Delete protection دي).
- بيدعم ال VPC Peering, AWS VPNs and 3rd party VPNs زى client connections

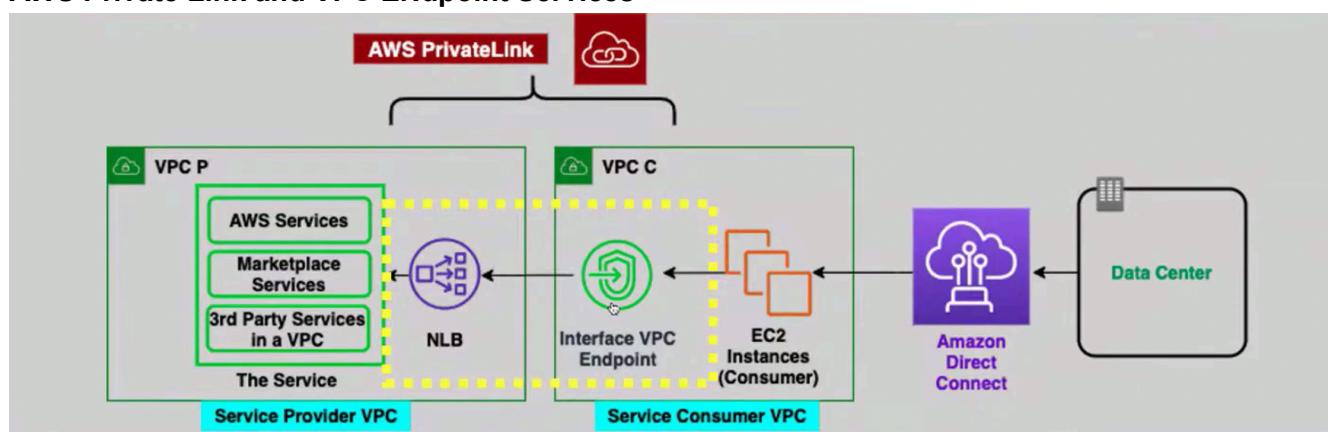
intrusion firewalls و third party appliances زى ال Service بتعامل مع : وهي GWLB (Gateway Load Balancer) ال security detection وغيرها من أمور ال



طب بيشتغل ازاي؟

1. لما يجي traffic هيعدي على GWLB
2. ال GWLB هيعتولى VPC الثانية او المكان الي فيه Third party inspection الي هيعمل عليه
3. هيعمل ال inspection لو تمام هيرجعه لي EC2 , لو اترفض هيعمله DROP/Reject حسب policy .appliance

AWS Private Link and VPC Endpoint Services



لو عندي service provider باخدها من VPC P زي صورة وعايز اوصلها لي Consumer والي هوا VPC C ويكون Secure ساعتها هعمل:

- ال NLB في Provider
- ال (NLB endpoint service (Private Link) واربطها بي
- في Private link service name VPC Endpoint consumer هعمل تشاور علی وهخليةها

AWS Serverless Computing

Having explored the basics of [Storage](#), let's now turn our attention to AWS Storage Services. This section will cover the different storage options AWS provides, from object storage to block and file storage, designed to meet diverse business needs with high durability, availability, and scalability.

Introduction

قبل شرح ال Serverless هنكلم عن EC2 زى

- ال Self-managed .
• مثلاً EC2 ببیقا فيها OS الى اختارته عن طريق ال AMI وبيبیقا تحتاج اعمله Patching كل فترة عشان ال Vulnerability وبحتاج احسب الخاصل بي APP Code لي ال RAM and CPU لي ال Rightsizing بتعاعي، وكان ال Pricing هو ال On-Demand as long as instance running .
- ال Serverless :
• الفكرة منه ببیقا تحتاج اعمل app code لي trigger دا و اقول ايه الي بيعمله عشان يشتغل، ميزته هو اني مش بحتاج اعمل App code بت AWS و Servers لي ال Provision Assign محتاجه، بتحاسب علي كل ./msec Execute و دا بال Trigger و الوقت الي أخده ال Code عشان يتعمله

Lambda

- ال Characterstics .
• ب Run Code منغير اي Provisioning او .Managing infra
• ال Save Costs باني بدفع فقط لـ Compute time و دا بي .Per-millisecond
• هي Event - driven architecture
• هي Highly available بدون تدخل منك
• لو عندك Traffic في بيقدر ي handle عكس ال Auto Scaling مش هيقدر يتفاعل مع spike مره واحده
• ال Lambda function can scale dynamically مع بعض request مع بعض بحد اقصي 1000 by default و بقدروا يتعدوا عادي ؛ وفي انواع:
• ال Reserved Concurrency: و دا انك تقول تحتاج desired concurrency قد ايه لي Lambda function عشان لما طلبوا يكونوا موجودين.
• ال Provision Concurrency: و دا بيخللي ال Lambda Pre-warm او جاهزة في اي وقت انها لما تطلب تشتعل فوراً و دا هيقل Latency (لأن ال Lambda عنده فلاؤل ال Cold start وبعدين لما تشتعل بتكون Warm فا دا هيخليلها pre-warm يعني جزئياً جاهزة انها تشتعل فا هيقل ال process المطلوبة فا هنكون اسرع)
• ال Application Auto Scaling: تقدر ستخدمها مع Target scaling policy عشان ت scale لي Provisioned .concurrency
• ال Lambda Function Configuration:
 - ال Function code
 - ال Dependencies
 - ال Execution role
- ال Lambda Quotas:
 - اخرها من 3 ثواني لي 900 ثانية (يعني 15 دقيقة) ويعمل timeout
 - تقدر ت run container .10GB package size و اخره

عشان ال Lambda تقدر ت Communicate مع Service في VPC معينه، لما بتعمل Configure لي Lambda بي VPC و SG و Subnets اللي بيحصل ان AWS Lambda بتعمل Launch لي ENI ف ال VPC دي وبيكون ال Communication من خاللها.

طب ولو عايز اطلعها على Internet؟ لو حطتها في VPC بشكل اللي اشرح فوق يبقيا هتتحاج تعرف NAT Gateway (في حالة انك حاطت ال Private subnet في VPC) عشان تخليها تشفوف internet لو مش حطتها في VPC فا هي هتكلم مع Internet عادي.

AWS Lambda - Execution Models

ال execution Model دا بيستي ال Response يجي من Lambda function عشان تخلص ال .Function Response بيكون فيه.

ال Model دا مش بيستي ال Response هو مجرد بيسلموا ال event وبعدين لما يخلص بقا يخلص بس مش بيستي.

ال Model دا بيشفوف لو حصل أي changes في Service الي معاها ولو حصل يبدأ يشتغل.

AWS Lambda - Dead Letter Queues

دي طريقة ان لو حصل timeout او execution مشتغلش انه بيعتها على Event Bridge او SNS او SQS عشان تقدر تشفوف Letter Queue(DLQ) وتعرف ليه فشلت انها تشتغل.

AWS Lambda - Environment Variables

تقدر تستخدم Lambda في أنك ت pass لي Code بتابعك عن طريق Enviornment values.

AWS Lambda - Pricing

ال Customers بيدفعوا علي حسب ل Request و Code Execution.

Request هو الي كل مره بيحصل لما executes function يحصلها event او response to a notification عشان ت عملها function.

AWS Lambda - CloudFront and Lambda@Edge

دا Option بيخليلك تعمل Processing على Edge بتابع CloudFront على Origin عشان تعملها ، دا هيخليه يقول ال User experience Improve و ي Latency.

عشان تعملها لازم يكون في cloudfront trigger لي Lambda ، ال lambda نفسها تكون software function code على حسب processing لي ايها.

AWS Step Functions

ال AWS Step Functions هي visual workflows و التحكم في them تكون Lambda Functions.

API Gateway

What is Application Programming Interface(API)

يتخلي Two Application Servers عن طريق Request و Response ، ال APIs بتتعمل في Clients أما ال Developers ويكلموا معاه.

أي حاجة في AWS Services بتتكلم عن طريق ال APIs حتى في AWS Console

ال Client Requests ويباخد Responses من ال API.

API Types

في نوعين لي API

1. ال Public ودا مفتوح للكل
2. ال Internal او Private ودا مغلق لي access معينه

في Standard او Architecture معين لي API عشان ال App يفهموا يعني من الآخر طريقة Communication.

1. ال HTTP APIs: دا بيكون اسرع ولكن مفهوش Features كتيره
2. ال RESTful APIs: بيكون فيه Features كتيره

ال API Standards في النوعين API Gateway

REST API - Anatomy or a Request or API Call

ال API Request في ال API دا معناه انك عايز تجيب Data معينه من Target عدك ودا بيكون علي شكل URL or Endpoints.

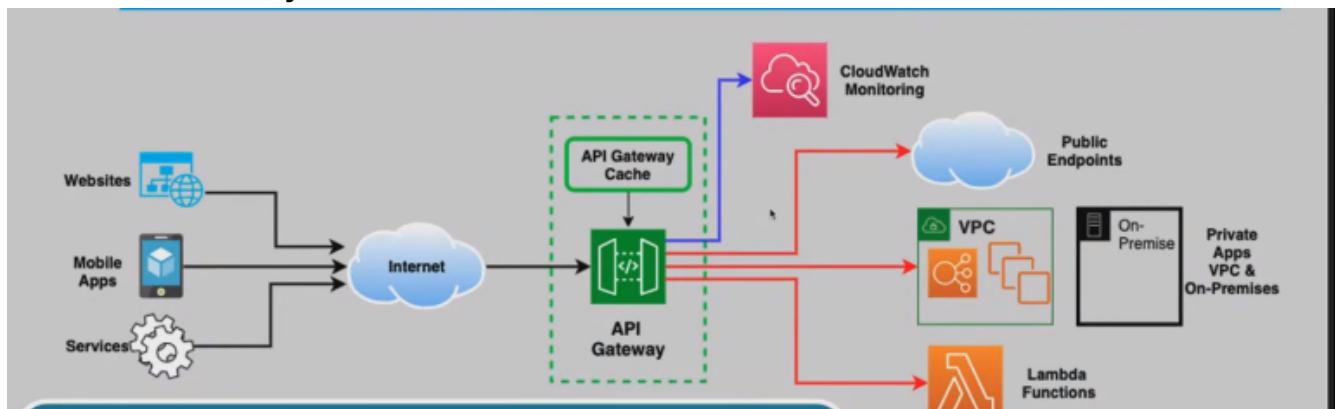
ال نوع ال Request زى بيكون GET, PUT, POST, PATCH, DELETE وغیرها حسب طلب ال Client.

ال Response بيكون فيه Data الي عايزها ال Client.

⚠ Warning

حاجه مهمه او ي لازم مابين ال HTTPS او TLS/SSL Certification يكون فيه Client/Server معني اصح عشان يكون Data encrypted in Transit.

Amazon API Gateway



ال API Gateway هي Service Managed بتكون عن طريق AWS و بتكون Secure و Highly Available.

يقدر يعمل Requests لـ Caching و دا بيساعد على Response.

Using API Gateway we can create:

- RESTful APIs, which include:
 - REST APIs,
 - Offer API proxy functionality and advanced features such as usage plans, API Keys, publishing and monetizing APIs.
 - HTTP APIs
 - Optimized for APIs that proxy to Lambda and HTTP backends.
- WebSocket APIs

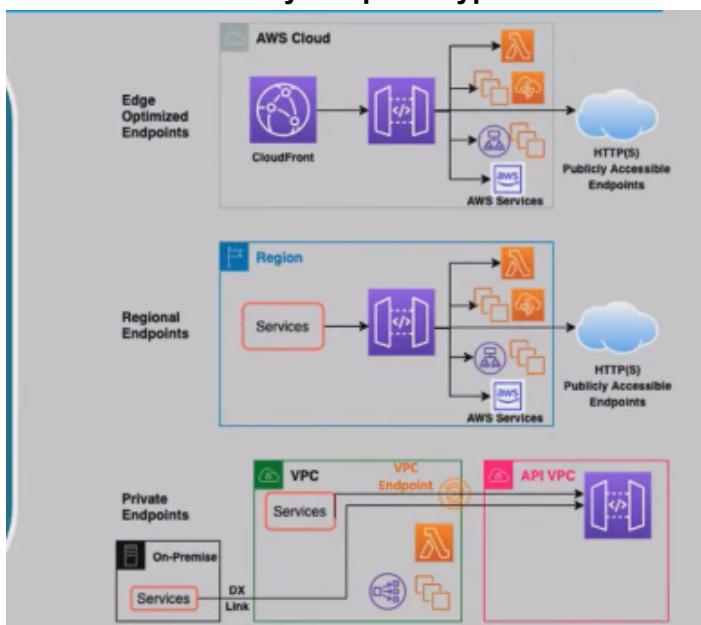
Usage plans:

Can be used to create access plans for certain APIs, define quota limits, associate APIs with keys, and general usage and billing documents.

دي الحاجات الي يقدمها APIs في API Gateway ، هنلاقي نوع جديد اسمه WebSocket APIs ودا بيحصل من Communication Server/Client بيكون.

الUsage Plans هي زي بتحدد عدد معين او Quota معينه لي API Requests دا وهكذا

Amazon API Gateway Endpoint Types



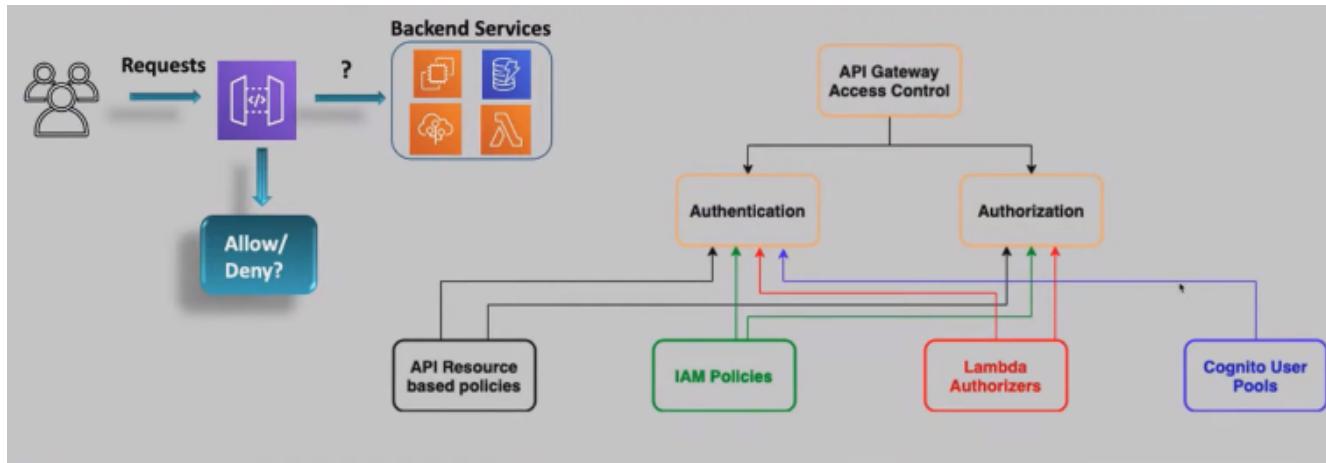
عندی 3 أنواع

1. الـ Edge Optimized Gateway دا بيكون عن طريق CloudFront وبيخدم الـ Public APIs ، وبيكون هو الـ Default بتاع API

2. الـ Regional API Endpoint بتكون لـ CloudFront بعنها ، دي معندهش CloudFront ولكن ممكن تحط بنفسك .CloudFront عن طريق Public APIs بتخدم الـ Public APIs

3. الـ Private API Endpoint بيستخدم الـ Private APIs وبنكون في حدود الـ VPC من خلال VPC

Amazon API Gateway Endpoint - Authentication and Authorization



عشان تعمل Request محتاج تحدد لي API Gateway مين مسموح ليه ومين لا ، ودا بيحصل عن طريق ال IAM Policies وبنقولها مين مسموح ليه يكلم منخلالك وهكذا وممكن من خلال Cognito user pool او Lambda Authorizers .

الLambda Authorizers: دا بيخلி Client لى API Request + Token بيعت ال API Gateway ، ويروح ببها على Lambda Validate ، فا لو تمام هبيعت لي Identity provider وفياها تتأكد من Token Validate وسليم ، فا لو تمام هبيعت لي API GatewayLambda .Deny ي Allow ولا ي Permit

الAPI Gateway: بيدى Client لى Token لما يعوز يعمل Request ، وبعددين يروح بروج Client باعت لي API Resources .Cognito User Pools يعملها Validate و ال API Request + Token مع Cognito

Notification, Messaging and Application Integration in AWS

Amazon Simple Queue Service (SQS)

Message Queues - Why?

الفكرة اني اخلي Services تبقا independent عن بعض ، يعني لو عندي Application بتابع طلبات فا انا كا User لما اجي اطلب واحد confirmation email بناعتي وادفع لازم يجي لي email عشان ارتاح واتاكد اني كله تمام ، فا مينفعش لما اطلب ميجليش ال email ساعتها هحس في حاجه غلط اتنصب عليا مثلا ولا حاجه وهو ممكن تكون مشكلة بسببه ان في طلبات كتيره في نفس الوقت فا هنا بحاول اخلي الخدمه بتاعتي تحمل ويكون في زي طابور وكل لما خدمه تقدر انها ت process لطلب تبدا ت process ليه وتبعه ال email .

فا ال Queue عباره عن وسيط مابين حاجتين بتخفف الضغط الي جي من Producer لي Consumer وتحملوا هي ويكون Async ويبينفع في Microservices architecture .

فا ال Queue بت:

- بت Enhance App performance
- بتخليك تقدر تعتمد علي Service بتاعت

Amazon SQS

ال SQS هي عباره عن Waiting Queue او Buffer service في حاله وجود app بس تعملها في حاله وجود orders منغير ما نوقف وبدل ما أعمل load على services الي بتعمل shipping service لي shipping service دي واحاول اسر عها، انا هحط SQS Queue عشان بيقا فيها كل حاجه، و shipping هي الي تعمل poll وقت الي تبقا فاضية فيه منغير pressure حتى لو Shipping وقعت ال Orders متروحش مني

ال Orders Producer و ال Shipping هنقول عليها Consumer

طبعاً هيحتاج Policy عشان يكلم مع الحاجه الثانية لو علي EC2 او ECS وغيرها من Services تانية

ال Region بي multiple AZ message في نفس SQS بي ال Queues

مش مطالب انك ت manage cluster ولا اي حاجه هو بيبقى برا VPC اصلاً ، كل الي عليك انك تكتب URI بـنـاعـه بـس فـي كـوـد و تـدـيلـه permission انه يكلـمـها و بـسـ.

هي عباره عن poll-based مش بتعمل push لأي message لا هي بيتسحب منها فقط.

• ال Characteristics :

- بي Receive و store ويبيعـتـ ال message مـابـينـ compo~nentsـ اليـ عـاـيزـهـمـ بـيـتوـاـصلـوـاـ
- لو Consumer مش متوفـرـ هيـفـضـلـ ال queue شـغـالـ
- بـيـسـاعـدـ عـلـيـ تـطـبـيقـ / de-coupling وـهـيـ أـفـرـقـ الـ compo~nentsـ عـنـ بـعـضـ

ال message size unlimited بـيـكـونـ اـخـرـهـ 256kbـ لوـ اـكـتـرـ اـعـلـمـ batchـ وـقـسـمـهـ Message Numberـ صـغـيرـةـ.

تقـدرـ تـقـدـرـ السـلـكـيـهـ EC2ـ اليـ فـيـ ASGـ عـلـيـ حـسـبـ Metricـ الـ Number of message of the queueـ عنـ طـرـيـقـ Cloudwatchـ قـاـدـهـ اـكـتـرـ مـنـ وقتـ معـيـنـ

:SQS Queueـ فيـ نوعـيـنـ

1. الـ Standardـ : بـيـكـونـ لـيـكـ Unlimited throughputـ ، بـيـضـمـنـكـ انـ messageـ توـصـلـ عـلـيـ الـأـقـلـ مـرـهـ يـعـنـيـ هيـ مـمـكـنـ تـظـهـرـ اـكـتـرـ مـرـهـ فـاـ هـيـخـلـيـكـ تـزـوـدـ فـيـ كـوـدـيـنـجـ انـكـ تـتـأـكـدـ ، بـيـحاـولـ يـعـمـلـ best effortـ عـشـانـ يـحـافـظـ عـلـيـ تـرـتـيـبـ

2. الـ FIFOـ : بـيـكـونـ messageـ exactly once processingـ يعنيـ مـفـيشـ first in first outـ duplicatesـ ، والـ FIFOـ SQS Queueـ . وتـقـدرـ تـعـدـلـهـاـ اـنـهـ تـاخـدـ 3000ـ Message per secondsـ 300ـ FIFOـ اـخـرـهـ وـدـاـ هـيـزـوـدـ

الـ pollingـ فيـ منـهـاـ نوعـيـنـ

- الـ Short pollingـ : بـيـقـرـاـ اـكـتـرـ مـنـغـيـرـ ماـ يـسـتـنـيـ لـيـ Queueـ
- مشـكـلـتـهـاـ لـوـ mes~sagesـ بـتـاعـ rateـ مشـ كـبـيرـ فـاـ دـاـ مـعـنـاهـاـ اـنـيـ بـضـيـعـ requestsـ وـدـيـ بـتـحـاسـبـ عـلـيـهاـ.
- الـ Long pollingـ : بـيـديـ مـهـلـهـ لـيـ 20ـ queueـ ثـانـيـهـ قـبـلـ مـيـرـجـ عـقبـاـلـ ماـ يـقـرـأـ مـنـهـمـ وـتـقـدرـ تـعـدـلـهـاـ.
- مـعـنـاهـاـ اـقـلـ فـاـ costـ requestsـ اـقـلـ

الـ Retention periodـ : يـقـدـرـ يـخـزـنـ الـ messageـ لـمـدـدـ 4ـ ايـامـ كـاـ defaultـ ولكنـ تـقـدرـ تـخـلـيـهاـ لـيـ 14ـ يومـ وـبـعـدـيـنـ تـخـفـيـ وـبـتـكـونـ حـسـبـ Requirementsـ

الـ Visibility timeoutـ : بـقـولـ لـيـ messageـ لـمـاـ تـتـاـخـدـ اـنـهـ تـخـفـيـ عـنـ consumerـ لـمـدـهـ مـحـدـدهـ وـلـوـ خـلـصـتـ تـتـمـسـحـ بـقاـ اوـ تـنـعـادـ مـنـ الـ firstـ processingـ . علىـ حـسـبـ حـصـلـهـاـ اـيـهـ جـواـ الـ

الـ Delay Queueـ : انـكـ بـتـخـلـيـ انـ فيـ delayـ لـيـ new messageـ قـبـلـ مـتـظـهـرـ فيـ الـ Default~Queuesـ بـتـكـونـ 0ـ Secondsـ الـ Maximumـ 15ـ دقـيقـةـ .

الـ Dead Letter Queues(DLQs)ـ : لوـ messageـ رـاحـتـ لـيـ processingـ وـمـتـعـلـهـاـ deleteـ وـفـضـلـتـ تـرـجـعـ لـيـ queueـ . troubleshootingـ عـشـانـ نـعـمـلـ عـلـيـهاـ DLQـ

الـ Encryption in transit TLS/HTTPSـ : نـسـتـعـمـلـ IAM Policiesـ عـشـانـ نـقـولـ مـيـنـ يـكـلـمـ مـعـاـهـ ، بـتـدـعـمـ الـ SQS Securityـ . SEE With KMS keysـ بـسـتـعـمـلـ الـ encryption at restـ

الـ Group Message In FIFO Queuesـ : وهيـ اـرـتـبـطـ الـ Sequenceـ معـنـهـ جـيـ منـ Sequenceـ IDـ حـسـبـ Queuesـ . messageـ وـكـلـ group idـ producerـ بنـعـرـفـ انهـيـ تـتـبعـ لـيـهـ وـهـكـذاـ .

Amazon Simple Notification Service(SNS)

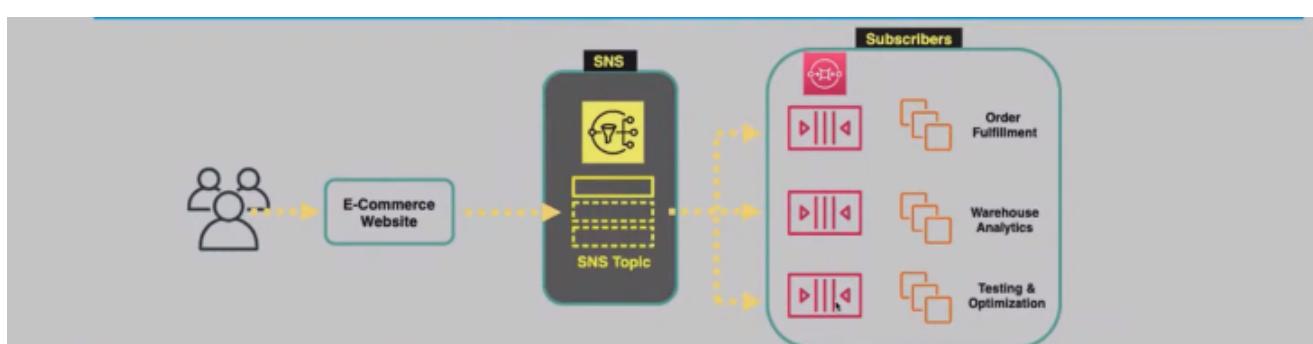
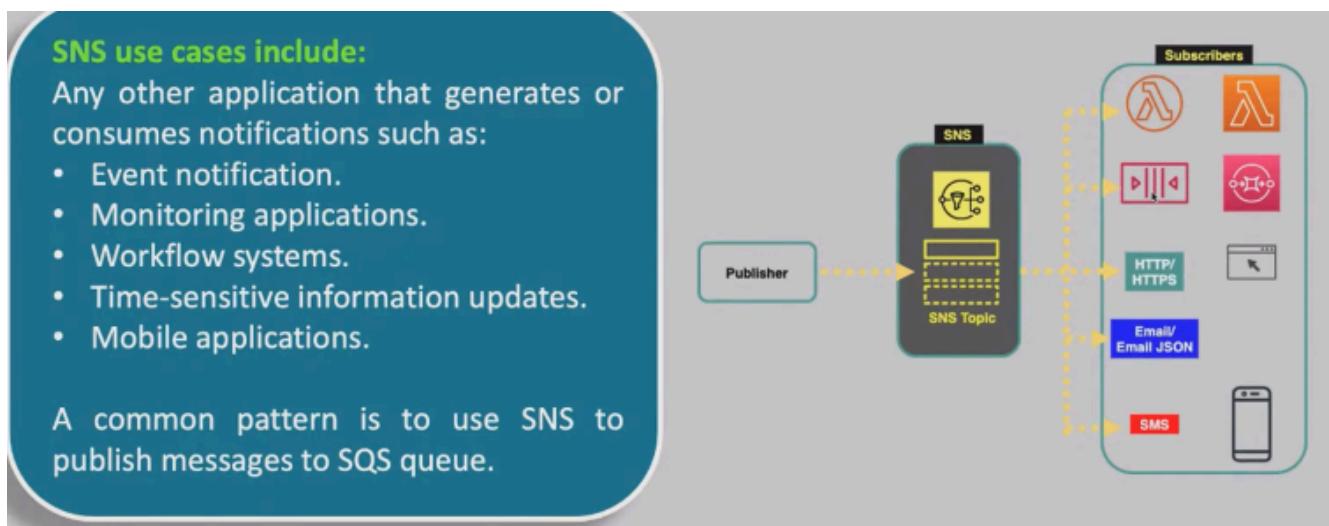
ال Service دی بتشغل بطريقة ال Pub/Sub و عندي ال SNS Topic حاجه اسمها create ب Publisher . وبيفا في subscribers بيشرك في topic دی ولما publisher يعمل message لي topic بتروح بشكل immediate لـ subscribers.

ممكن استعملها على SNS عن طريق ال Email/SMS زى لو عايز أكد ال order فا عن طريق ال

ال SNS هي Push service يعني بتاخذ ال message وتوصلها على طول مش بتستني حد يعملها حد poll زى

.Encryption at rest و HTTPS in Transit

Use Cases:



في حاجه اسمها Fanout ودي بتسعمل SNS كا وسيط عشان ابعث نفس message لـ multiple SQS Queues ، ودي بعمل فيها Message Async processing .

ال Message filtering وهي اني ممكن اعمل filter لـ queue معينه مش لازم messages كل queues بباتاعتي وهكذا.

فيها زى DLQs نفس ال FIFO SQS

AWS Storage Services

Having explored the basics of Storage, let's now turn our attention to AWS Storage Services. This section will cover the different storage options AWS provides, from object storage to block and file storage, designed to meet diverse business needs with high durability, availability, and scalability.

Amazon Elastic Block store(EBS)

الـ EBS هي Block storage وبنكون Networked External Storage من نوع EBS ودا لأنه مش في نفس المكان الي متواجد فيه الـ EC2 ومن نوع SAN لأنها Block Storage.

- بيكون unformatted في حالة انه مش Root Volume, يعني بيعتاج منك شوية steps زيادة عشان ترتبط بي EC2 باتاعتك.
- بتكون persistent يعني الـ data بتفضل في disk حتى بعد متوقف الـ instance, terminated/deleted لغاية لما تعملها على instance بتغير الـ data على disk مش هتروج.
- لازم كل EBS تكون في نفس الـ AZ الخاصة بالـ EC2 الي عايز اعملها Attach.
- الـ EC2 مع الـ EBS Root volume بنقول عليها EBS-Backed EC2 instance.
- بقدر أخد منها Backup بشكل自动的に snapshots ويحطها في S3.

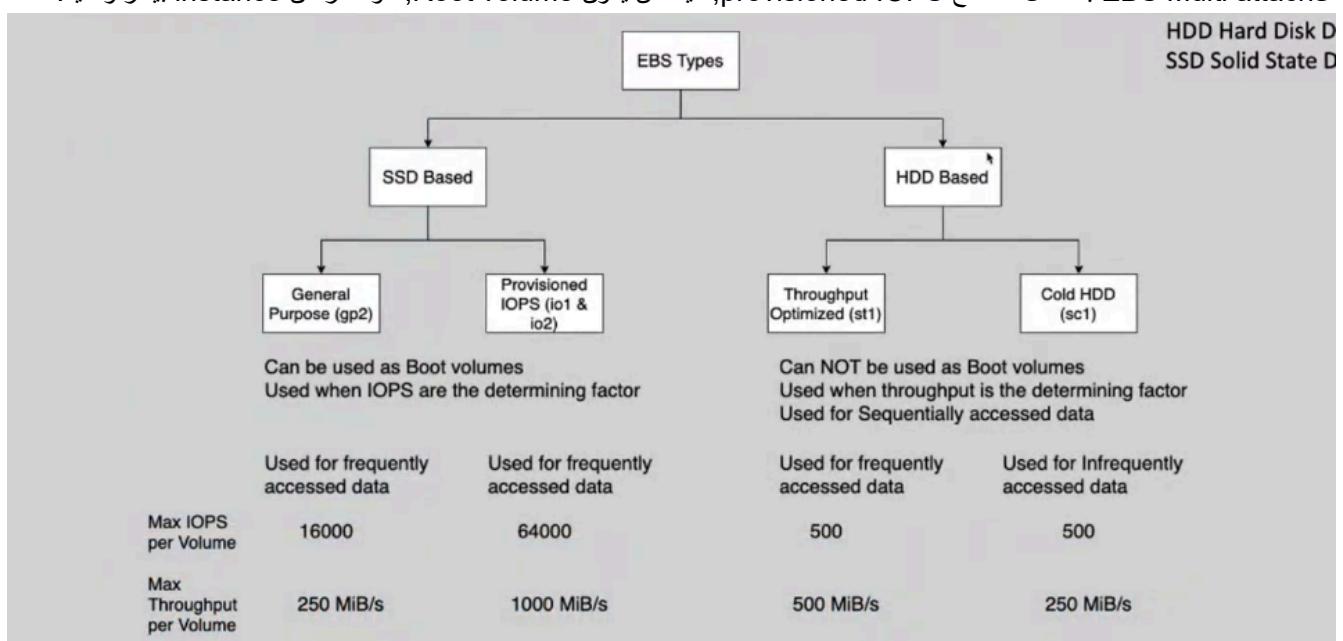
بيكون Primary storage durable يعني يعتمد عليه, عشان data متروحش وتقدر تعملوا attach لـ EC2 Running, هو بيكون الـ data device عشان مش بيضيع الـ data, ويكون recommended على الـ DB run لو.

- بيكون Durable في نفس الـ AZ عشان كدا هو Replicated multiple server على الـ instance.

الـ IOPS دا معناة عدد العمليات الـ read writes per second عن الـ EBS وبتكون أقل في instance-store.

الـ Throughput دا حجم البيانات الـ data per second بتتنقل.

الـ EBS Multi attach بتحصل فقط مع instance, provisioned IOPS يكون Root volume, هو اكتر من instance بيخرزنا فيه.



	General Purpose (gp2)	Provisioned IOPS (io1)	Throughput Optimized (st1)	Cold HDD (sc1)
Use case	<ul style="list-style-type: none"> • General workloads • Small Databases • Dev/Test environments • Virtual Desktops • Workloads performing small, random I/O 	<ul style="list-style-type: none"> • Large IOPS intensive workloads that require consistent performance • Large production databases 	<ul style="list-style-type: none"> • large, sequential I/O workloads such as Amazon EMR, Big Data, ETL, data warehouses, and log processing. • Streaming workloads requiring consistent, fast throughput at a low price. 	<ul style="list-style-type: none"> • large, sequential cold-data workloads. • Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important
Cost	Higher	Highest	Low Cost	Lowest Cost
Orientation	IOPS	IOPS	Throughput	Throughput

دي أنواع الـ EBS و use case

لو عندك Random I/O و SSD-based EBS performance ثابت
ولو عندك HDD-based EBS فكر في Large sequential I/Os, throughput based

• Pricing

- أي Volume مستقل عن EC2(Root volume) هنتحاسب عليه
- هنتحاسب علي حسب نوع ال IOOPS.

- هنتحاسب علي S3 بي Snapshots لـ Per GB-month of data stored
- هنتحاسب علي outbound data transfer لو هتبقا

Modifying EBS Volume

تقدر تعدل علي ال EBS وهي شغاله منغير ما تعمل Detache, تقدر تعدل ال size ال type performance



تقدر ت increase size لي اس مش هتقدر تعمل increase decrease

EBS Snapshots

في منه نوعين:

- الـ Data Lifecycle policies: وهي Snapshot بتعمل بناء على Automated/Scheduled policies
- الـ Regions: ويدبيها على S3 على نفس Region manager(DLM) وتقدر تعاملها Copy على نفس Region

الـ Manual: هي الـ Automated بس الفرق انك بتعملها بنفسك وبتروح علي نفس ال S3 في نفس Region

Snapshots Lifecycle Policies Using DLM

الـ Amazon Data Lifecycle management هي بتتوفر لك انك تعمل automated snapshot management لي اس snapshots و هي احاط policies معينة واربطها بي tags الي على ال EBSs يعني اي TAGS عليها نفس Policy هيطبق علي ها ال snapshot expiration backup حسب policies الي حدتها, ولازم اعمل tags عشان مدفعتش مبلغ كبير علي حجم المتر acumulated snapshots.

Copying EBS Snapshots Between AWS Regions

الـ Use cases الي بحتاج فيها لـ Copy لـ snapshot

- الـ Geographic expansion يعني عايزة data تكون علي region تانية عشان الشركة بتتوسع او اقرب لـ user
- لو عندي disaster recovery plan
- لو هـ migrate لـ region تاني
- لو audit requirements او compliance مطلوبة يكون في مكان بعيد عن region الأساسية

EBS Encryption

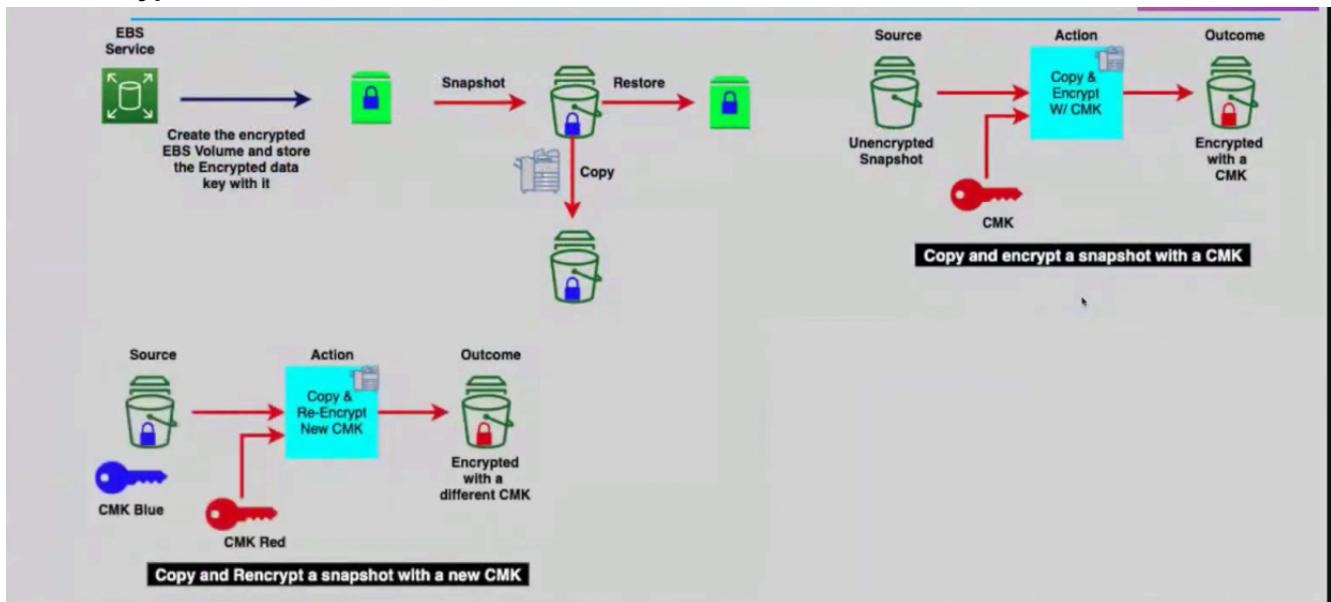
بتستعمل ال CMKs عشان تعمل generate لـ data encryption و ي encrypt data عشان ي decrypt data على EBS.

الـ Data encryption بيحصل الـ encryption/decryption تحت مستوى الـ OS

- الـ AWS بتحطه في Storage Subsystem (EBS, S3, Redshift Managed Storage)
- الـ App يعني مشحتاج تغير حاجة في الكود او تسطب agent عشان تشغله encryption

فمعني كلام دا أن ال EBS نفسها بتكون Encrypted عن طريق ال CMKs يعني كدا ال data at rest بتكون encrypted وال data على host نفسه قبل متروح لـ EBS بتكون Encrypted فلما تتبع لـ EBS بتتبع و هي Encrypted يعني data in-transit أمان

EBS Encryption Status



لو عندك EBS معموله encrypted snapshot هيطلعلك EBS Snapshot encrypted وعملت منه copy ولو هعمل من EBS Snapshot restore ه تكون النسخة دي بردو encrypted لو هعمل copy لي region تكون تانية ه تكون by default encrypted key option أني أغير ال

لو عندي unencrypted snapshot وجيست اعملها copy مع CMK ساعتها هيفا معايا ENCRYPTED EBS Copied

لو عندي unencrypted snapshot وعيز اخليها create لـ volume ساعتها هعمل encrypted snapshot لـ EBS وكمان واحد من Encrypted EBS restore واعمل snapshot CMK

لو عايز تحول ال EBS لـ Unencrypted encrypted attach لـ EBS تانية مش على نفس ال host وتنقل ال data عليه، فـا الي هيحصل ان ال EBS على نفس ال Host ويعتها على ال EBS وبـكـا حولتها (يعني مـبيحصلـش غير بشـكل مـلـتوـي مش طـبـيعـي).

Sharing EBS Snapshots

- ال unencrypted snapshots
 - بقدر اخلي ال snapshot دي لـ AWS Community all في حالة ال unencrypted بـس يحتاج اعدل ال permissions public لـ
- ال encrypted snapshots
 - مـبـقـرـشـ اـخـلـيـهاـ sharedـ لـ accountsـ فقطـ فيـ الـ publicـ snapshotsـ ،ـ تـقـدـرـ تـكـونـ sharedـ لـ accountsـ اليـ مـخـتـارـهاـ اليـ مـعـاهـهاـ .ـ CMKـ لـ permissionsـ
 - لو بيـعـلـمـ اـعـلـمـهاـ مشـ هـيـنـعـ اـعـلـمـهـ shareـ لـ permissionـ defaultـ CMKـ لـ

Default Amazon EBS Encryption

دا بتقدر من خالله أن أي EBS هـتـ تكونـ by default encryptedـ علىـ مستوىـ الـ regionـ كلـهاـ.

Redundant Array of independent Disks (RAID) and EBS Volumes

الفـكـرهـ منهـ فيـ الـ Physical~Serversـ اـنـيـ اـحـطـ الـ Disksـ بـنـاعـتـيـ فيـ Arrayـ وـدـاـ هـسـتـفـيدـ منهـ انـ Dataـ هـتـكـونـ معـ كلـ Disksـ فالـ رـاحـ واحدـ منـ الـ data~Disksـ هـنـفـضـلـ اوـ عـشـانـ أـحـسـنـ الـ Performanceـ .

عشـانـ تـقـدـرـ تـأـخـذـ الـ EC2ـ بـنـاعـتـهـ increaseـ IOPSـ اـقـدرـ أـعـملـ:

- استعمل EBS-Optimized
- استعمل RAID array الخاص بي EBS، الـ RAID Typesـ بـتـدـعـ كلـ OSـ

Not Recommended

.Root/boot volume کا RAID Volumes من Recommended AWS من استعمل اني

أنواع ال RAID:

RAID 0 :

Highest IOPS performance among all RAID types.

Resulting IOPS is the sum of individual IOPS for all volumes.

No redundancy/mirroring. Failure of any volume means failure of the entire array.

RAID 1:

NO IOPS performance enhancement.

Redundant since the same data is written to all volumes.

RAID 10:

Combines the benefits of RAID 0 and RAID 1. Provides redundancy and performance enhancements.

Amazon Instance-Store

هي بتكون temporary/Ephemeral storage بيقوم مع ال EC2 عشان كدا بتكون not persistent, و بتكون في نفس ال Physical Server .EC2 مع ال

- هي مناسبة لي Caching Buffer و
- بيتقال ليها AWS instance-store backed EC2 instance لما نربطها بي .
- بيتتسخ automatic لما تعمل stop او instance terminate لي ال، ودا يخلينك متقدرش تغير نوع ال data instance لأن ال هتضييع كدا.

بستعملها في حالة ال distributed system/architecture, يعني لو data بتتعاي بتتوزع على اكتر من Instance, نكون استفينا بسرعة وانه ينقل من system معين لغيرة.

1. Amazon S3

هنتكلم عن أشهر Storage service في AWS وهي Amazon S3، وإزاي تستخدمها، وإيه المميزات والخصائص اللي تميزها.

1. Intro to Amazon S3

إيه هو S3؟: هي container store objects، يكون عندك bucket دا اسمه container، يكون عندك unlimited storage لأنك عندك scalability EBS العادي زي Storage.

ال Object يكون عبارة عن data و metadata و دي object زمي عن ال owner last modify size.

الاسم لازم يكون **Unique** (مش أي حد يقدر يعمل نفس الاسم). مرتبط Region معينة في AWS. ممنوع يستخدم أحرف كبيرة أو .
مثل: my-unique-bucket-123.underscores

مش بترتبط بي أي Network

• مناسب له:

• Backups

• استضافة Static Websites

• ال Big Data Analytics

• ال distributed architectures

• المميزات:

• ال Durability: توصل ل 99.999999999% (11 تسعة) مش هتضيع, يعني أحتمالية فقدان ال data قليلة جداً لأنها بتعمل Stores لي داتا أقل حاجه في 3 AZs مختلف.

• ال Availability: توصل ل 99.99% Availability

• رخيص

• مش ممسوك بي limitation

Understanding Resource Ownership

ال owner هو ال اللي عمل ال Bucket ودا بيكون معاه full control .Objects and buckets

.upload الى owner بناء على عمله ,bucket by default

2. Characteristics

S3 - Data Consistency Model

ال S3 عشان تتكلم معها بتكلم عن طريق ال API فا دا بيعتبر HTTP Request وبيكون فيه GET, PUT, and DELETE فا دي طريقة التعامل مع S3

• ال GET : نقرأ .object

• ال PUT : نرفع او نكتب .object

• ال DELETE : نمسح .object

ال Data consistency مقصود لما بعمل PUT / DELETE هل بترجع data على طول ولا بيعتاج يستوي شوية عقبال ما يتحدى؟ يعني هل ال data اللي لسه اتحطت او اتغيرت، ظهرت فعلاً لما جيت أقرأها؟ فا على حسب عندك نوعين من Operation :

• ال **Read after write** هي مقصود بيه ال GET (يعني Read) بعد ال PUT (يعني Write) على طول ال new object writes or upload لـ S3 ومعناه ان ال S3 لو عملته Read وجيـت بعدها تعمل Update على طول بالـ update ودا اسمه consistency

• لو حصل overwrite او Deletes لـ object معين بيكون eventual consistency دا معناه انه مش هحصل بشكل فوري ولكن بعد شوية هيفقا .Updated

S3 - Static Website Hosting

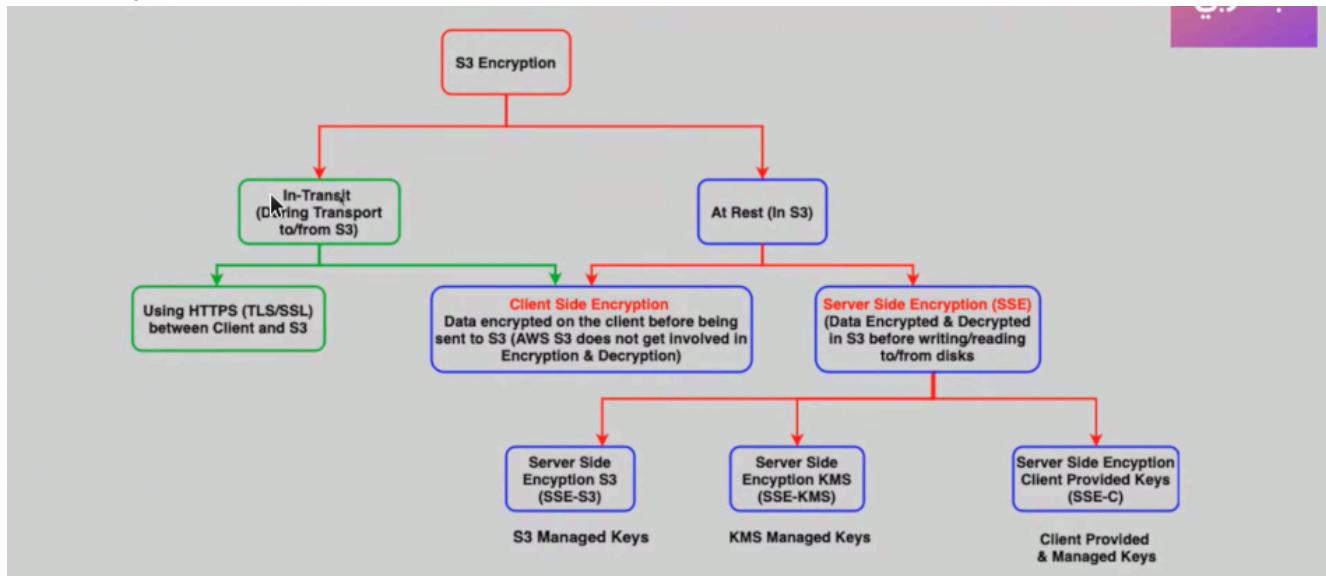
تقدر ت Host لـ Static website في S3 من غير سيرفر.

• اللينك بيكون بالشكل: . http://bucket-name.s3-website-aws-region.amazonaws.com

ال S3 Bucket العادي يستعمل HTTPS ومقدرش استعملها في redirection وبتدعم ال bucket operations زي .List, GET, PUT وغیرها ، يستعمل REST API endpoint سواء لـ Public or private content .XML formatted response

الـ S3 Static website مبيده عمش الـ HTTPS لو عايز تعملها هحتاج تربطه بي Cloudfront دا في حالة HTTPS مهمه ليك، بيرجع على مستوى redirects ، بتدعم public content , object and bucket ، كل لازم يكون HTML Documents .s3 hosted websites مع domain name

S3 - Encryption



في S3 عنده نوعين من Encryption

1. الـ In-transit: ودا في وقت أستعمال HTTPS مابين الـ Client and S3 عشان يكون Secure.
2. الـ At Rest: وهي أني Data لي الـ Storage بناعتي وهي في الـ Secure.

في الـ Client side encryption: دي أني اعمل data encrypted على الـ client قبل ما ابعتها لي S3 ودي الـ S3 مش بيتدخل في Encryption&Decryption بناعتها.

في الـ Server Side Encryption(SSE) ودا لـ Data بتكون Encrypted & Decrypted على الـ S3 قبل الـ Writing/Reading او على Disks ولليها 3 أنواع :Encryption

1. الـ Server side encryption S3 (SSE-S3): بتكون S3 هي اللي بتعملها Managed S3.
2. الـ Server side encryption KMS (SSE-KMS): بتكون عن طريق KMS managed ، لما بتعملها فا انت بتدفع على Requests KMS Service دا هتتدفع مبلغ كبير عليها فالـ Cost استعمل S3 Requests.
3. الـ Server side encryption Client provided keys (SSE-C): بكون client هو اللي مقدمها وبيعملها managed.

⚠ Warning

كل اللي فوق دا Object Level encryption يعني بتعمل على مستوى object فقط يعني ممكن تلاقي S3 Bucket Default encryption وتنانى لا. ولو عايز تعمله على مستوى bucket هحتاج ت enable لي SSE-KMS او SSE-S3 و دا بيعملك Encryption على مستوى الـ bucket.

الـ S3 Bucket Default Encryption: دا بيعملك Encryption على مستوى الـ bucket، بتعمل فيه

S3 - Versioning

عامل زی Git و هي أنه بيحفظ بال Object عشان لو حصل اي حاجه بشكل غير مقصود زي overwrite او اتمسح من غير قصد وعايز ارجعه فا أرجعه بسهولة.

مشكلة أن كدا ممكن يحصل مشكلة أن ال Storage يزيد مني ودا يعلم معناه cost زيادة ، ممكن هنا اعمل حاجه كويسيه وهي أنيأشغل كل حاجه ولكن استعمل Lifecycle policy وهي أن بعد مده معينه امسحني versions القيمة

تقدر تعمل MFA Delete ودي another layer of security عشان تأكيد ان الـ delete هو الـ bucket owner

لما تعمل Delete لـ Object مش بيمسحوا ولكن بيحطوا DELETE Marker عليه ولو حبيت ترجعه تاني هتعملوا delete تاني وبكدا هيرجع ال last version او current version بمعنى اصح.

S3 - Replication

وهي أني اعمل نسخ من Bucket بمتاعي يا في نفس Region يا Region مختلفة، ال TLS/SSL بستعمله عشان ي Encrypt لي replication data in-transit.

لازم Versioning يكون شغال في الـ bucket الأساسي والـ Destination bucket عشان تعمل .Replication

ويمكن احدد ال objects الى تتلخذ كلها ولا يبي tag او prefix معين

أنواع الـReplication:

- الـ **Cross-Region Replication (CRR)**: نسخ الـ Data bucket لـ Region ثانية.
 - الـ **Same-Region Replication (SRR)**: نسخ الـ Data bucket لـ Region في نفسها.

Encryption

1. لو عامل Replicate SSE-S3/SSE-KMS تقدر تعملوا
 2. لو عاماً SSE-C مش هقدر اعمله Replicate

S3 - Bucket Policies

هی دی ال Resource based - Policy و هي انك بتقول مين هيقدر ي Manage او يعمل Operations على S3 بناعتي مذكورة هنا : IAM Policy

مثال اقدر استعملها في آيه : تعمل Bucket public عشان تستضيف موقع ويب.

S3 - Multipart Uploads

لما تعمل upload لـ Object يكون اكتر من 5mb ودا لأنه بي improve throughput يعني بيقسم ال uploaded parts ويداً يعملها upload in parallel ودا بيعحسن ال throughput لـ object

S3 - Object Lock

دی Feature بتخلیک متقدرش تعمل operations زی update فا ال object بیکون (WORM) ودا
لستعملو ه عشان compliance معنه.

دي بتتحمي objects من أنها تتمسح لي وقت معين او لي الأبد, لو الـ object نقلته سواء CRR or SRR ه يكون في Protected destination bucket.

Note

خد بالك الـ Object Lock by default بيشغل versioning حتى لو انت قافله بيفتحوا

عندى طريقيت عشان اشغل الـ object lock

- الـ Retention Period: دا بحط lock لي مده معينه, قد يليها Two modes
- الـ Governance mode: دي بحدد ناس معينه يقدروا يتحكموا في object حتى لو عملوا lock.
- الـ Compliance mode: دي مش بخلி حد نهائي حتى الـ root user ميقدرش يغير حاجه في object.
- الـ Legal Hold: دا بحط lock لغاية لما حد معين يشيله

S3 - Presigned URLs (or Query String) Authentication

هي أني ادي لي Users عاديين ملهمش علاقة بي AWS شوية URL يقدروا من خلالها يعرفوا حاجات Private خاصة بشركة مثلاً. الـ Pre-signed URLs بتكون بنكoon permission على request عشان تقدر تعمل على Private By default. الـ object specific URLs ب تكون بنكoon object يعني يعنيه.

S3 - Transfer Acceleration

دي على مستوى الـ Bucket Feature بتخليلك تعمل files من أي مكان بعيد في العالم لي الـ S3 بتاعك.

هي معمولة مخصوص عشان تسرّع نقل الملفات خصوصاً لما تكون المسافة بين الـ Client و الـ Bucket كبيرة (long-distance).

طيب إزاي بتشتغل؟

الـ acceleration دي بتستغل الـ CloudFront edge locations (يعني السيرفرات اللي موزعة في العالم):

- الـ Client بيتعت الـ data لأول Edge Location قريبة منه.
- ومن هناك، بتنتقل الـ data على طول من خلال AWS optimized backbone network (مش عبر الإنترن特 العادي).
- . وأخيراً توصل لـ S3 Bucket بتاعك في الـ Region.

لما تشغلي الـ endpoint بتاعتك هتنغير لي : `bucketname.s3-accelerate.amazonaws.com`

ولكن كل حاجه ليها تكلفتها وأكيد خدمه زي دي هتكلفك زيادة حسب حجم GB و عدد الـ Requests

Tip

في AWS من AWS Transfer Family هي Fully managed، هي دي اللي بتستعملها AWS Transfer Service اسمها .FTPS, FTP, SSH, SFTP, and AS2 او الـ bucket في وبرا الـ EFS/NFS Files

لو جه سؤال بيقول:

A junior scientist working with the Deep Space Research Laboratory at NASA is trying to upload a high-

resolution image of a nebula into Amazon S3. The image size is approximately 3 gigabytes. The junior scientist is using Amazon S3 Transfer Acceleration (Amazon S3TA) for faster image upload. It turns out that Amazon S3TA did not result in an accelerated transfer.

يعني لو عندك S3 وشغلت فيها Transfer accelerator وجالك حد عايز يعرف ملف حوالي 3 جيجا ولكن الـ Acceleration محصلش ساعتها مش هتحاسب علي مصاريف الـ Acceleration لانه محصلش ولكن هتحاسب عادي علي

S3 - Requester pay

في S3 انت بتتحاسب علي Data transfer ولكن بي الـ feature دي انت بتخلி الي بيعمل transfer دا هو الي يحاسب عليه ، استعمالاتها زي انك عندك مشروع open source مثلاً واي حد عايز data يدفع تمنها بس انت مسؤول عن تخزنها بس وهكذا

S3 - Cross-Origin Resource Sharing (CORS)

الـ CORS هو طريقة بتخلی الـ Web Application اللي جایة من Domain معين تقدر تتفاعل مع Resources موجودة في Domain ثاني كأنه Authenticated Domain من الـ الأولاني.

في حالتنا هنا: تقدر تخلی Web Application تشووف الـ S3 Bucket اللي في files اللي في S3 بناءً على Web App جایة من Domain مختلف.

مثال: لما تبقا حاطط مثلاً JSON file أو Image في S3، وبتستخدمهم في موقع مستضاف على Domain ثاني، هنا المتصفح لازم يعرف هل S3 يسمح بالـ Access ده ولا لا؟ لو مفيش CORS configuration، المتصفح هيمعن الـ Request

S3 - Batch Operations

بستعملها لو تحتاج اعمل operation على Objects كل execute على Large-scale operation list of objects بيقدر يعمل job لـ اكتر من billions of objects و exabytes of data يمكن تستعمله لي:

- الـ copy objects
 - الـ set object tag and ACLs
 - اعمل initiate object retrieval/restores from glacier
-

S3 - SELECT

هي Feature بتخليني أعمل Simple SQL Statements بـ Filter على S3 Object Contents عشان اعمل Enhance Application Performance بتـ .Required data

عن طريق انها بتقلل Latency و CPU و cost و بتوصل لي %400 .cost reduction %80 مش في كل الحالات يعني و performance boost

S3 - Performance Recommendations

الـ Retries and timeout: خلي الـ Application بـ Retries Request عشان لو واحده اتعمله drop او الـ channel

queue احسن تكون retries لها توصل لما تعمل chances فا

الـ parallelization for high throughput لـ higher throughput GET or PUT connection استعمل multipart request بـ شكل multiple : عمل اكتر من request لـ

Transfer Acceleration

الـ Byte range fetches تقدر ت download parts من object بـ كل بدل مره واحد object

S3 - Server Access Logging

بـ ينcluded أنه يسجلك الـ Records خاصـه بي requests التي حصلـت على Bucket بـ تـابعـك على bucket ثـاني يكون لي .disabled by default ،

لـ ما تـعملـه enable بـ تحتاج تـحدـد source التي هيـجـب منه Logs وـ دـا الـ Logs هـتـبعـلـوا.

مش هـتـدفع cost زـيـادة عـلـيـه بـس هـتـدفع عـلـي storage .

S3 - Monitoring and Event Notifications

الـ S3 بـ يعمل مع CloudWatch Integration (Logs error وـ فيه ولا لا مـثـلا (و) CloudTrail) دـا عـشـان يقولـك مـين عمل get ، put وـ هـكـذا)

الـ Event Notifications بـ يحصلـ أـمـا معـين بـ يحصلـ زيـ Event Notifications

الـ S3 Bucket مـمـكن تـعملـه Configure event notification automatically انه بنـاء عـلـي event notification يـشـغل بـ شـكـل

- الـ SNS Topic
- الـ SQS Queue
- الـ Lambda Function

3. Access Management

الـ goal الأساسـي هنا إنـك تحـكم بـ شـكـل دقـيق جـداـ (granular) مـين يـقدـرـ :

- يـقـرأ أو يـكـتب (Objects)
- يـعـدـل أو يـمـسـح (Files أو Buckets)
- يـدـخـلـ مـنـين وـيـسـتـخـدمـ إـيـه

AWS بـ تـقـمـ أـكـترـ من Layer أو طـرـيـفـةـ تـقدـرـ تحـكمـ بـيـهاـ، بـسـ مشـ كـلـهـ بـقـوـةـ بـعـضـ، وـكـلـ وـاحـدةـ لـيـهاـ مـكـانـهاـ وـسـيـنـارـيوـ اـسـتـخـامـهاـ.

IAM Policies (Best Practice)

ديـ أـفـضلـ وـأـقـوىـ وـسـيـلـةـ تـحـكمـ بـيـهاـ فيـ صـلـاحـيـاتـ الوـصـولـ لـ S3 لأنـكـ بـتـاخـدـ تحـكمـ:

- بـتـديـكـ تحـكمـ دقـيقـ اوـ نـفـصـيلـيـ جـداـ
- تـقدـرـ تـربـطـ الـ Permissions ديـ بـ IAM User أوـ Service IAM Role
- بـتـكـتـبـ بـصـيـغـةـ JSONـ وـاضـحةـ
- بـتـطـبـقـ عـلـيـهـ AWS Resourceـ، مشـ بـسـ S3

- الـ Centralized Policy: بتنقر تكتب واحدة تطبقها على أكثر من user أو service.
- الـ Controlled & Auditable: تقدر تتبع مين بيعمل ايه.
- الـ AWS services Compatible مع باقى AWS services بسهولة.

```
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::my-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "s3:x-amz-server-side-encryption": "AES256"
    }
  }
}
```

كده أي حد يحاول يرفع حاجة مش معهولة ليها encryption Identity-Based policy بتحط لي User/Role.

S3 - ACLs (Access Control Lists)

لأنها AWS من Best PracticesDeprecated غير لحالات نادرة.

الـ ACLs هي طريقة قديمة من AWS عشان تدي Permissions على مستوى:

- الـ Bucket (Bucket ACL)
- الـ Object (Object ACL)

بتسخدم امتى؟

- لو بتترفع Objects من Account تانى غير صاحب الـ Bucket.
- أو لو System قيم مش بيدعم IAM policies (rare cases).
- عيوبها:
- لأنها Limited جداً، متقدرش تتحكم في تفاصيل زي الـ Conditions أو Tag-based access.
- لأنها Cross-account environments أو Large systems Confusing في Large systems.
- ميفعش تعمل Auditing عليها بسهولة.

S3 - Enforcing Encryption (Bucket Policy)

لو عايز تتأكد إن كل Object داخل الـ Bucket لازم يبقى Encrypted (مثلاً بالـ KMS) ليه دا مهم؟

- الـ Security Best Practice، خصوصاً مع Sensitive Data.
- يساعدك تلتزم بيـ Compliance (PCI, HIPAA, GDPR).

- يمنع أي خطأ بشرى (زي رفع data unencrypted)

```
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::secure-bucket/*",
  "Condition": {
    "StringNotEquals": {
      "s3:x-amz-server-side-encryption": "aws:kms"
    }
  }
}
```

دي يتمتع أي حد يرفع File مفهوم (Resource-based policy) KMS Encryption لأنها بتحظ على S3

Other Access Control Layers

Bucket Ownership Settings (ObjectOwnership)

- يتمتع إن الـ Object يبقى Owned من حساب غير صاحب الـ Bucket
- في Cross-account scenarios، ممكن تحصل مشكلة إن الـ Object يتربع لكن مينفعش يتحكم فيه.
- الحل: خلي الـ Bucket ownership = BucketOwnerEnforced

Block Public Access

يتمتع أي حد يحاول يخلي الـ Bucket Public حتى لو كتب Policy غلط!

و دي بقى حاجة خطيرة جداً:

- بتحميك من أخطاء الـ junior engineers أو developers
- لو أي حد حاول يكتب AWS Policy تفتح الـ Bucket، AWS تمنعها مباشرةً

فعل كل الخيارات اللي في Block Public Access

4. S3 Storage Class

كل الـ S3 Classes replicated على أكثر من 3 AZs في one-zone IA

الـ Frequent/General Access: دي العادة اللي بخشها بشكل Daily ومش ببيقا في charges على الـ Requests او Retrieval

الـ S3 Standard

- الـ S3 Standard، وهي أني بخشن أخذ داتا من s3 بشكل يومي.
- زي مثلاً: cloud applications, dynamic websites, content distribution

كل يوم access

most cost-effective access tier لـ data لـ **S3 Lifecycle** او **S3 Intelligent Tiering**: يعمل بشكل Automatic, بينما كل data لي على حسب الـ access frequency.

- بدفع small charges كل شهر عشان مراقبة كل Object.

الـ **Infrequent Access**: مش مستعملها بشكل Daily وبدفع علي Retrieval

• الـ **S3 Standard-IA**

- بيبقا موجود فيها الـ Data اللي بتبقا less frequently, يعني الداتا مش بيخسلها ناس كتيره.
- بتبقا lower storage price بس بيبقا extra retrieval fee.
- بستعمل لي Backup و Disaster recovery storage.
- الـ availability three nine's تكون فيها.

• الـ **S3 One Zone-IA**

- الـ data بتبقا في Single AZ.
- بتبقا أقل سعر من Standard-IA ولكن لازم مستعملها بحد عشان لو AZ وقعت خلاص راحت عليك.

الـ **Archive**: معمولة للداتا اللي مش بنحتاج نفتحها كتير، بس تحتاجين نخزنها لفترة طويلة بأرخص تكلفة ممكنة. في كذا نوع:

1. الـ **S3 Glacier Instant Retrieval**

- دي أرخص Storage class ينفع تستخدمها لو عندك داتا مش بتتفتح كتير، بس أول ما تحتاجها، تحتاج توصلها في millisecond.
- مناسب لي الـ media assets او Medical images .user-generated content archives.

2. الـ **S3 Glacier Flexible Retrieval**

(الاسم القديم كان: S3 Glacier): دي مناسبة أكثر للداتا اللي مش هنفتحها خالص، بس لو احتجتها ممكن تستنها شوية.

- مناسبة لحالات زي:
- Backups
- Disaster Recovery
- Offsite Storage

- لو يحتاج الداتا 1 أو 2 مرة في السنة، دي الأنساب.
- عندها حاجه اسمها Expedited retrieval في Glacier: دا نوع من أنواع الـ Retrieval في Glacier و هي انك بتحط Options يعني يقول داتا دي عايزة ترجعلي من دقيقة لي 5 ودا أغلى حاجه.
- الـ Retrieval بياخد وقت شوية (minutes - hours)

3. الـ **S3 Glacier Deep Archive**

(دي أرخص Storage class في S3 كلها معمولة للداتا اللي غالباً مش هنفتحها خالص، أو يمكن مرة في السنة).

- الـ Retrieval بياخد وقت أطول
- مناسبة لحالات زي Archive طولية المدى (7-10 سنين)
- شركات في مجالات زي:
 - البنوك
 - الصحة
 - الحكومة
- اللي عندهم متطلبات قانونية للاحفاظ بالداتا.

4. الـ **S3 on Outposts**

(On-premises environment): بتخلي عندك S3 على أجهزة AWS Outposts جوه الـ On-premises environment بتناعك.

- يعني تقدر:

- تستخدم نفس الـ S3 API
- تخزن وتقرأ البيانات محليًا على الـ Outposts
- تحكم في الـ access, tags, reports locally
- مناسبة جدًا لو عندك requirements ان البيانات تفضل في نفس المكان وتحتاج Performance عالي لأن التطبيقات شغالة محلي.

Note

(وهي في منها Service وعشان تتوافق معها تحتاج API أو SDKs أو S3 Policies) أو S3 Glacier

Breakdown

S3 Storage Classes Comparison

	S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)					
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved

هنا لاحظ ان All storage class معمولين انهم:

- يكونوا Durable لي 11 nines
- يكونوا available لي 99.9 مع كل One zone-IA Classes تكون 99.5 %
- يكونوا available service-level agreement(SLA) بي 99% : دي الـ "ضمان رسمي" من AWS. يعني يقولوا لك رسميًا: "إحنا بنوعدك إن الخدمة تشتعل بنسبة 99% أو أكثر شهريًا، ولو حصل أقل من كده، ممكن نرجعلك فلوس في صورة Service Credit".
- الـ Millisecond latency for first byte: يعني معناها إن أول بآيت من البيانات اللي بتحاول تقرأها من S3 هيوصللك بسرعة جدًا، في خلال milliseconds.
- يكونوا في Lifecycle transitions.

S3 Lifecycle Policies

أنت لما تخزن data على S3، عندك كذا Storage Class، كل واحدة منهم ليها سعر معين، و Use Case، ومدة تخزين محددة (يعني أقل وقت تدفع عليه حتى لو مسحت الـ Object).

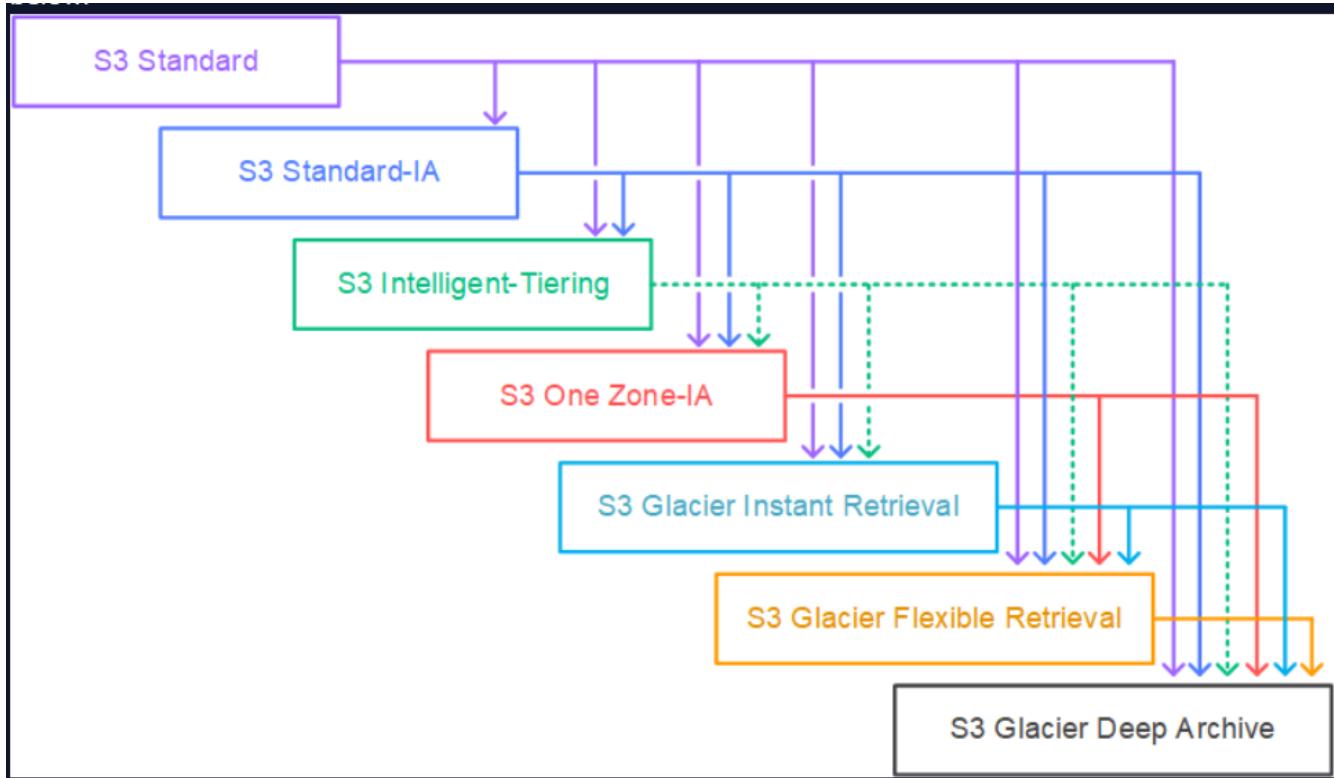
و ساعات بتحب تنقل الـ object من Class لـ Class تانية حسب الـ Access Pattern بتعالك، ودي بتعملها باستخدام Rules.

بستعملها عشان تعمل نواعين من Actions

1. الـ Transition actions: وهي نقل objects مابين classes بعد وقت محدد.

2. الـ Expiration actions: وهي أمسح الـ Objects اللي ملهاش لازمة بعد وقت معين.

كل دا بيحصل علي bucket بشكل عام او علي objects معينه زي اني احدد .prefix or tag



1. كل Storage Class ليه Minimum Storage Duration

يعني لو حطيت object في Class معينة ومسحته قبل المدة دي، هتفتح تمن المدة كاملة

2. الـ Transition دايماً بيتم بعد 30 يوم على الأقل

يعني AWS مش هتنقلك object من Storage Class للثانية غير لما يعدي عليه 30 يوم (أو أكثر)، حسب الـ Rule اللي انت

حددها.

5. S3 Migration

Snow Family

هي بسistema عشان ي transport data من/الي AWS S3 Secure storage system

بتقدر ت لي transport massive amount of data من/الي AWS

الـ Snow Devices بتكون on-premises و بتكون DATA In transit protected by KMS يحمي DATA قبل portection device ودا بيزود متروح علي device

AWS Snowcone

هي ببساطة Device صغير، Rugged يقدر يستحمل الدرجات الحرارة العالية او المية و Secured وبيقدم Edge Computing يعني المكان اللي مفهوش نت أقدر استعمل فيه الـ Snowcone عشان أنقل الـ Data

وفيه منه نوعين

1. ال Snowcone : وهو عبارة عن 2vCPUs , 4GB Mem, 8TB HDD Storage

2. ال Snowcone SSD : وهو عبارة عن 2vCPUs , 4GB Mem, 14TB SSD Storage

AWS Snowball Edge

وهو في منه نوعين

1. ال Snowball Edge Storage Optimized : ودا عبارة عن 80TB OF usable HDD او 210TB of Usable NVME

Storage

2. ال Snowball Edge Compute Optimized : ودا مهم لي Connectivity يعني مكان مفهوش Edge Compute فا هو بيحافظ .28TB of Usable NVME Storage او 42TB OF usable HDD عليها ونشحنه تاني في AWS ودا عبارة عن 1000 Device .petabytes

AWS Snowmobile

هو Device بيكون عبارة عن huge storage truck وفيه customer site. بيتوصلى عن طريقها ينقل لي 1000 .petabytes

Data Units and Data Transfer Rates

- 1 Exabytes = 1000 PB (PetaBytes).
- 1 Petabytes = 1000 TB (TeraBytes).
- 1 TBytes = 1000 GB (MegaBytes).
- 1 GByte = 1000 MB
- 1 Byte = 8 bits

As a rule of thumb and to avoid calculations, remember:

- 100 Mbps (M bits per second) link -> can transfer 45 Giga Bytes per hour.
- Or 45 GB/hr x 24 hr/day = 1.08 TB/day
- 1 Gbps (G bits per second) link -> can transfer 10 x 45G Bytes or 450 GB per hour.
- Or 450 GB/hr x 24 hr/day = 10.8 TB/day
- Pay close attention to units. Per second, per hour, bytes or bits make a big difference.

ودي طريقة تحسب بيه.

6. Hybrid Cloud with Storage Gateway

Background & Introduction

ال Local Disk .Local Disk وتعامل معاه كأنه Storage عن طريق الشبكة iSCSI (Internet Small Computer Systems Interface) بروتوكول بيخليك متصل للسيرفر Storage يعني لو أنت عندك Server On-Premises وعايز تصيف له Storage جديد من خلال الشبكة، iSCSI بيخلطي الموضوع بسيط كأنك وصلته بـ SATA .SATA

ال iSCSI بيستعمل over IP يعني بيشتغل بي IP

ال Initiator هي سيرفرات الي تحتاجه متصل على Target Disk وهي ال Disks نفسها وبيتوصل بي SAN, NAS .TAPE, LUNs

الحل بقى فين؟ – AWS Storage Gateway

بعض يا سيدى، الفكرة بدأت من إن في شركات عندها Data موجودة On-Premises ومش عايزة يسيبوا خالص، وفي نفس الوقت عايزة

يستفيدوا من الـ Cloud سواء في الـ Archiving أو Backup أو حتى عايزين يوسعوا مساحة التخزين من غير ما يجيروا هاردات جديدة. هنا ظهر حل اسمه AWS Storage Gateway.

هو ببساطة عبارة عن جسر (Bridge) يربط الـ AWS Cloud On-Premises Environment بـ AWS Cloud بحيث تقدر تخزن، تدير، وتنترجم الـ Data بطريقة Seamless وكأنها شغالة محلياً عندك.

وهو كمان طريقة تخلي الـ S3 بيان لي File or Block Storage انها Applications على رغم من انه Object Storage عادي فـا هي بتعتبر S3 أكمنها محول من Object لي Block لي الـ App.

يبكون Encrypted In transit و comprised at rest و comprised قبل ميتبعت و comprised بعد ميتبعت عشان يتبعت بسرعه على S3 او العكس

AWS Storage Gateway Configurations

الـ File Gateway: بيخلطي السيرفرات عندك تتعامل مع S3 كأنه File System عادي. وبيعمل caching لـ S3 ، ويبكون NFS/SMB & Comprised لما يتبعن وتقدر توصل او ت mount عادي ، ببستعمل HTTPS & Comprised

الـ Volume Gateway: بيقدم لك iSCSI Volumes (Block-level Storage) . نوعين:

1. الـ Cached Volumes: هو كل الداتا في S3 .

2. الـ Stored Volumes: نسخة كاملة AWS Backup + Local Backup + Snapshots في AWS Disaster Recovery مثلـا.

الـ Tape Gateway: بيحاكـي Tape Library (Virtual Tape Library) في S3 أو Glacier . يخزن Backup في S3

7. S3 Pricing

الـ S3 بتندفع فيه على قد ما بتستخدم، يعني مفيش minimum fee، وبتحاسب حسب كمية الـ Data، وطريقة استخدامها، ونوع الـ Storage.

بتندفع على ايـ؟ حجم الـ Storage Objects (GB) في الـ Bucket، ودا بيتحسب شهرياً، والأسعار بتختلف حسب:

- الـ Region
- الـ Storage class (Standard, IA, Glacier ... الخ)
- الـ Requests اللي بتحط أو تنقل بيها Data زي:

 - زي PUT , COPY , POST , LIST
 - استخدام storage classes عشان تنقل data بين Lifecycle rules

الـ S3 Intelligent-Tiering لو بتستخدم Monitoring fees

• بتندفع شوية زيادة عشان AWS تراقب الاستخدام وتنقل الـ objects بين الـ Tiers حسب الـ access pattern.

مش بتحاسب على ايـ transfer في الـ S3 او ايـ Cloud front ليـ Cloud front او EC2 في نفس الـ region

بتحاسب على Transfer out سواء ليـ internet او regional internet

مفيش charges على الحاجات دي : (Free)

1. أول 100GB خارج من S3 ليـ internet (شهرياً).

2. ايـ data دخلة لـ S3 من الإنترنـت (Inbound traffic).

3. نقل البيانات بين Buckets جوه نفس الـ Region

2. Amazon EFS

What is the Elastic File System(EFS)

إيه هو EFS؟ هي NFS Storage و بتسعمل مع Linux يعني بنستعلها زى ما أنتشح في Storage .Fundamentals

بي ال File Level Target بتنقدر mount على أكثر من instance في نفس الوقت. ولكن لازم تكون Linux Distro وبيشتغل .multiple AZs

How to Use EFS

الـ EFS بي Mount ENI بقاعد كل AZ وكل Instance هت Attach ليها وبكدا Network interface دي زي Switch الحاجات ليبعض. لما تعمله Mount متناسش تفتح TCP Port 2049 ليه عشان هي دي port بتاعته وتقدر تعمله mount منغير مشاكل

لما تكون عايز تعمل EFS على Instance تحتاج تعمله manual mount او تكتب script عموما ولكن تحتاج تنزيل efs-utils لما من بعد 2012 windows فا مش هيديك NFS Native ولكن مش windows على mount .best performance

EFS - Features

تقدر ت Share ويكون فيه concurrent file access يعني كذا حد يعمل access في نفس الوقت.

يكون Durable ، و Automatic Scaling and consistent performance يعني بي scale لوحده وعنده اداء ثابت.

تقدر تعمل مش زي ال S3 actual hierarchical directory

عنه EFS مع EC2 high throughput (strong Read-after-write) يعني data consistency فورية ، يعني لما حد يكتب في تقدر ت Shawfها عندك خلال وقت قليل او ي ومكان فورا.

تقدر تتحكم في Permission مين ي access ومين لا عن طريق IAM and POSIX و فيه File locking

EFS Use Cases

- Big data analytics.
- Media processing workflows.
- Content Management and web serving
- Home Directories
- Cloud bursting

EFS - Classes & Lifecycle Management

الـ Standard : ودا الطبيعي الي بيكون فيه files Frequent access لي

• الـ EFS Infrequent Access (EFS-IA)

• نوع تخزين أرخص (وصل لـ 92% توفير) للملفات اللي مش بتتفتح كثير.

- تقدر تعمل سياسة(lifecycle policy) إن الملفات اللي مفتوحة من 60 يوم تتنقل لـ IA أوتوماتيك.
 - بيكون أقل سعرا ولكن بتدفع على .data retrieval
-

EFS - Data Encryption, Backup & Restore

ال EFS يدعم Encryption :

• في Transit

• في Rest ولكن يستعمل .KMS Keys

ال Backup و Restore :

- تقدر تعمل backup من EFS لي AWS DataSync بيعمل copy لي same Data مابين الأتنين سواء كانوا في same region لو مختلفة حتى.
 - في AWS Backup دي بتعمل backup solution creation, scheduling, restoration
-

3. Amazon FSx

ال FSx هي Managed Service بتعمل نفس الـ EFS ولكن لي Windows و Lustre

FSx For Windows File Server

هو Shared File server لي Windows عشان لو تحتاج access لـ objects text files او documents

عشان تستعملوا لازم يكون عندك Microsoft Active Directory او AWS Self managed ، بيستعمل SMB Protocol كـ network communication protocol

بيستعمل CIFS (Common internet file system) على Linux machines عشان يشتغل كمان

لو Clients الي جاية لي FSx جايin من On-premises او Direct Connect(DX)

ال FSx لازم يتحط في VPC

لـ Data Encrypt سواء في .in-transit and at rest

نقدر نعمله Deploy على Single AZ او Multi-AZ في ENI create لي FSx على حدتهاوا سواء .Primary Filesystem Standby لي Two ENIs az ، لو az two ENIs مع بعض و هيكون Sync

لو عايز تعمل Migrate من windows filesystem On-premises AWS DataSync

FSx For Lustre

ال Lustre هي عبارة عن Cluster + Linux ، هو لو انت تحتاج big data machines كتيره جدا تقوم in parallel و تدلك million of iops تحتاجها

بيكون Single AZ فقط.

ال Use Cases :

- HPC

- Machine Learning
- Electronic Design Automation
- Video Processing/rendering
- Big data and financial analytics

هو لي Short term لو هحتاج compute لي وقت قصير ومحاج intensive workloads

هو مش هنقدر تخزن عليه حاجه لي Long-term فا هحتاج durable storage system سواء S3 او On-premises

في نوعين من Lustre

1. ال Scratch: هو اقل سعر وبيكون لي short term ، مش هيضمن ال data بقىتك لان مفيش backup

2. ال Persistent: هو بس مش معناه انه هيفضل على طول ، بيكون highly available ، وال data volume بتكون daily automated backups في AZ نفسها ، وبيدعم ال replicated

Storage Service Compared

	S3	EFS	EBS	FSx for Windows	FSx for Lustre
Data Stored	Objects	Files (File system)	Block data	Files (File system)	Files (File system)
Interface Exposed	S3 Simple APIs	NFS	APIs	SMB for Windows (CIFS for Linux)	NFS
Latency	Low	Low / Consistent	Low / Consistent	Sub millisecond latencies/ Consistent	Sub millisecond latencies/ Consistent
Data Availability and Durability	Stored redundantly in multiple AZs	Stored redundantly in multiple AZs	Stored redundantly in a single AZ	Can be a single AZ, or Multi AZ (Primary /standby file systems)	Single AZ, with or without replication (Scratch and Persistent)
Access	One to millions over the web. Concurrent access needs to be built into the application.	Concurrent. One to thousands of EC2 instances or on-premises servers. Linux only.	One EC2 instance in the same AZ (or multiple Instances in case of Multi-Attach)	Concurrent. Thousands of EC2 instances and on-premises Windows or Linux clients.	Concurrent. Thousands of Linux clients in AWS and/or on-premises. (Must have Lustre client installed)
File Locking	No, has to be built into the code	Yes			Yes

4. Further Storage Services

AWS Backup

هي AWS Services backup activities لي كل Automate Service في بتعملك ، Fully managed backup service هي حتى On-premises

يدعم زى Services. و بي integrate مع Cloudtrail و SNS

تقدر تعملها بيحصل backup job سواء Automatically او Manually

ال Backups Encrypted at rest using SSE-KMS بتكون

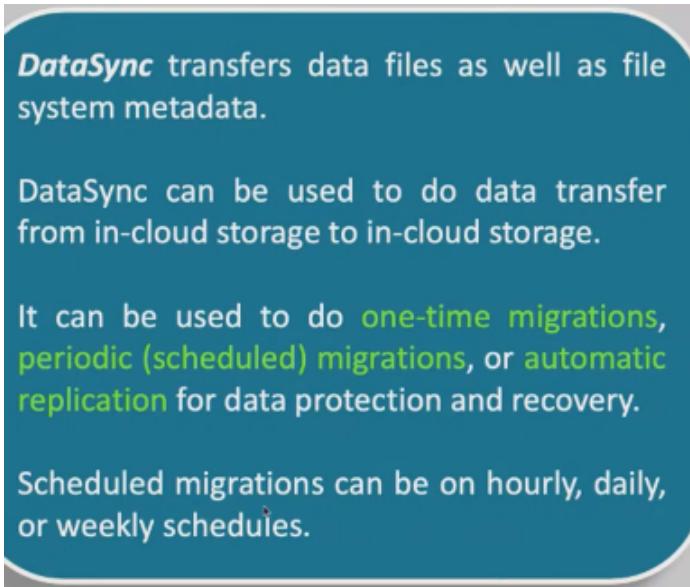
Restoring من backup باستعمال Service دى هيحصل على new resource مش الشغال حاليا.

AWS DataSync

هي Storage gateway Server عکس ال مش Virtual Machine Client

فا هو بي copy process Automates, accelerates, and simplifies cloud on-premises مابين cloud وما بين لـ

عادی cloud



فا هو VM Client بيكون علي Cloud عن طريق internet او Direct connect وبيأخذ Large amount of data On-premises

ممكن يكون 10x اسرع من available tools ، و بيستعمل TLS عشان ي Encrypt data in-transit

AWS DataSync vs Snowfamily

The Snow Family is used for offline data transfer.

- Use the Snow Family when the communication link cannot serve due to high utilization or low bandwidth, or for massive data sizes.

DataSync is used to transfer data online 10x faster than other tools, and with minimal overhead.

- Use DataSync to move data that changes frequently.

AWS Datasync vs Storage Gateway file gateway

DataSync

- Behaves like an NFS or SMB client and connects to an existing SMB file servers or NFS file system.
- We can use it to automate and accelerate online data transfers to S3, or to move data that existed before deploying storage gateway.
- Use it to transfer data to S3, EFS, or FSx for Windows File Server, and between in-cloud storage systems.
- DataSync only moves/copies data but does not provide access to data.

Storage Gateway File Gateway

- Plays a file system or file server role (supports SMB and NFS interfaces) where servers and clients mount it.
- Storage gateway transfers data to S3 only.
- Use it as NFS/SMB server to transfer on-going updates and to provide low latency access to data in S3.

AWS Database

Now that we've covered [Databases Fundamentals](#), let's explore AWS Database services. In this section, we'll delve into the fully managed, scalable, and secure database solutions AWS offers, enabling you to store, manage, and analyze data with ease across a variety of use cases.

1. Relational Databases

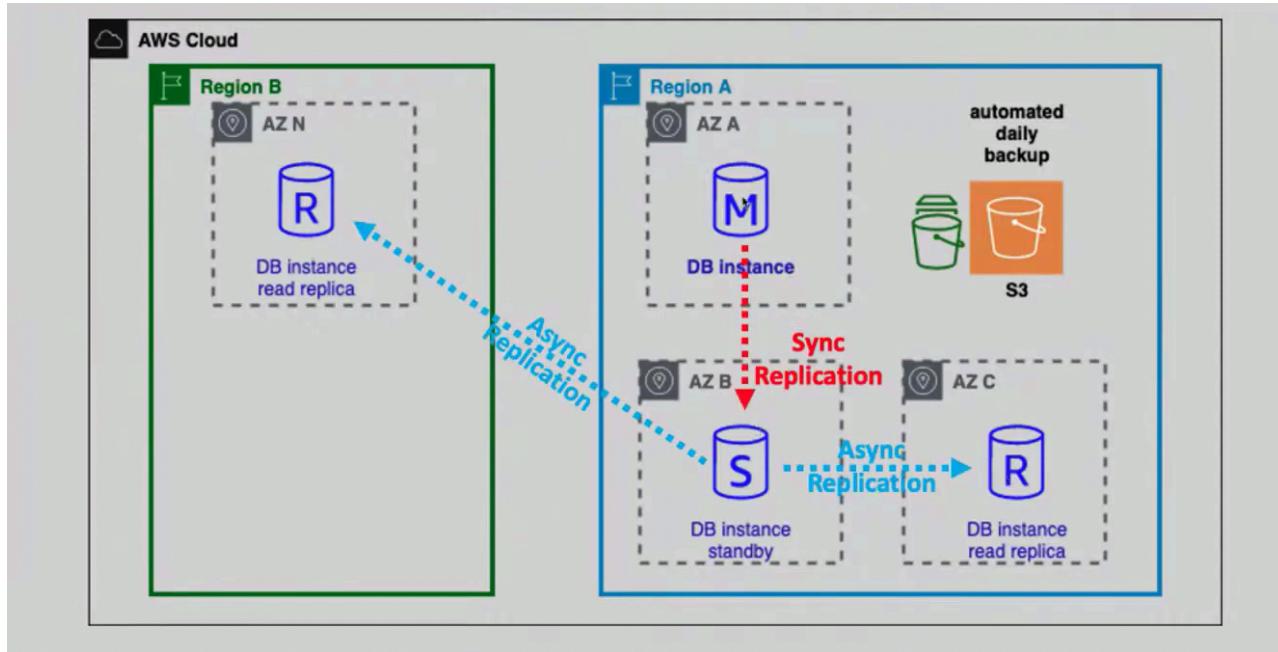
شرح معناها في DB

لما بت Scale بـ Vertical Scale

Amazon RDS

- هي Managed Service لـ AWS من Relational Databases (زي MySQL، PostgreSQL، Oracle).
- هي OLTP DB، ويتقدر تختار نوع ال database engine براحتك.
- هي **Fully managed**: يعني انت مسؤول بس عن انك تستعملها و Security الـ Security التي عليها مين يعدي ومين لا.
- الـ **best practice**: أنك تحطها في private subnet.
- الـ **Auto Scaling**: AWS بت Manage الـ DBs scales.

- هي بت Launch VPC جو ال (لازم).



- **Read Replicas:** (بتبقا Copy بتعمل عليها Read queries) لتوزيع ال Read Load في حالة ال Queries Load. فا بيحصل Offload يعني بيواند ال Load مابين Primary RDS و بتكون **Read Replica**.
- **ASynchronous:** انت تقدر تحدد مكان مكانها يكون فيه، ممكن تحطتها في Region مختلفه مثلاً، ممكن تكون primary instance class او storage type مختلفه عن standby او.
- ال AZ وقعت، فيه نسخة تانية تشتعل بشكل Automatic Standby وأنها تحول لي النسخة ال Standby الى **Multi-AZ DB Instance**: مع ال Primary RDS بس ال standby مش هيستعملها غير لو ال Primary وقعت. ودي مش بقدر اعمل عليها اي لغاية لما ال primary تقع وي هي تشتعل.
- ال Two readable standby DB instance Primary DB Cluster: بتعمل DB Cluster فيها DB instance و كل instance بيبكون لها AZ مختلفة دا بيقملك serve read workload و High availability لو عالي وهكذا.
- ال RDS بت Integrate SNS مع عشان لو حصل مشكلة يتبعنك.
- ال Multi-Region: نسخ بين Regions Disaster Recovery من أي حاجه ممكن تحصل لل Region.
- ال Pricing: بناعها يتحسب زي ال EC2، بيحسبك على ال backup size ولكن مش بيحاسبك على عملية ال backup بتكون بشكل Outbound Data transfer, automatic.
- العيوب: مفيش وصول لا ssh Instance (مش هنقدر تعدل على السيرفر بنفسك).

Multi-AZ deployments	Multi-Region deployments	Read replicas
Main purpose is high availability	Main purpose is disaster recovery and local performance	Main purpose is scalability
Non-Aurora: synchronous replication; Aurora: asynchronous replication	Asynchronous replication	Asynchronous replication
Non-Aurora: only the primary instance is active; Aurora: all instances are active	All regions are accessible and can be used for reads	All read replicas are accessible and can be used for readscaling
Non-Aurora: automated backups are taken from standby; Aurora: automated backups are taken from shared storage layer	Automated backups can be taken in each region	No backups configured by default
Always span at least two Availability Zones within a single region	Each region can have a Multi-AZ deployment	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Non-Aurora: database engine version upgrades happen on primary; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent in each region; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent from source instance; Aurora: all instances are updated together
Automatic failover to standby (non-Aurora) or read replica (Aurora) when a problem is detected	Aurora allows promotion of a secondary region to be the master	Can be manually promoted to a standalone database instance (non-Aurora) or to be the primary instance (Aurora)

RDS Multi -AZ - Failover

ال Failover بيشتغل لما:

- ال Primary AZ or DB .failure يحصلهم
- يحصل primary DB loss of network connectivity لـ
- ال Compute او storage حصل فيهـ .failure
- ال DB Instance OS لـ Patching
- لو بغير ال .instance type
- لو حصل manual failover (يعني عملـ .primary reboot with failover على الـ)

⚠ Warning

لما تربط الـ App بيـ RDS اديـ الـ IP addresses عـشان لو حصل failover يستعمل نفسـ الـ DNS hostname متضـطـرـش تغيـروا بـأـيدـكـ تـانـيـ.

RDS Automated Backup

الـ Automated Backup بـيـحصل يومـياـ وـبـنـطـحـ فيـ S3 كلـ دـا enabled by default , الـ S3 Managed by AWS , وفيـ Point in time recovery , وـديـ AWS بـتـعـمل Backup بـشكلـ مستـمرـ لـ transaction logs كلـ 5 دقـائقـ , الـ backup دـا بـيـكونـ فـيـ الـ S3 لـمـدةـ 7 أيامـ وـتقـدرـ تـعـدـلـهـ لـ 35 يومـ فقطـ وـمـكـنـ تـنـقـلـهـ لـ S3 تكونـ اـنتـ الـ Primary RDS عملـها عنـ طـرـيقـ انـكـ تـعـملـهاـ .export to s3

الـ Multi-AZ هناـ بـيـفـيدـ جـداـ , وـداـ لأنـ الـ Backup مـمـكـنـ يـائـرـ عـلـىـ سـرـعـهـ الـ Read/write الـ Primary RDS فيهـ backup , فـاـ الـ RDS معـ .Primary RDS متـاثـرـ وكـداـ الـ standby sync هيـ multi-az بـيـحصلـ معـ

RDS Manual Snapshots

هيـ هيـ Automated Backup ولكنـ مشـ بـيـحصلـ point in time recovery بعدـ ماـ عملـهاـ transaction backup ولكنـ مشـ هـيـتسـجـلـ فيـ manual snapshot وـطـبـعاـ مشـAutomated

⚠ Warning

الـ Multi-AZ بـيـأـنـرـ عـلـىـ primary RDS بـتـاعـ performance حتىـ لوـ مشـغـلـ الـ Manual Snapshot

RDS Restoring from Backups

هـيـعمـلـكـ RDS DB جـديـدةـ لـماـ تـعـملـ Restore لـ backup وـتقـدرـ تـغـيـرـ الـ DB Storage type اوـ .Backup

RDS - Copying/Sharing Database snapshots

لوـ عملـتـ الـ Copy لـيهاـ سـوـاءـ فـيـ نفسـ الـ Region اوـ Region مختلفـ فـاـ هيـ بـقـتـ Manual snapshot وـاناـ لـيـ بـتـحـكمـ فيهاـ مشـAutomated Backup وـمشـ هـيـطبـقـ عـلـيـهاـ ايـ حاجـهـ لـيهـ عـلـاقـهـ بيـ

الـ automated backup سـوـاءـ فـاـ اـقـدرـ اـعـملـهاـ share encrypted or not Manual snapshots مشـ هـقـدرـ اـعـملـهاـ accounts share تـانـيـةـ , اـماـ لوـ

RDS - Vertical Scaling

دي RDS Feature في أنها بتغير ال storage instance type او RDS الخاص بنفس storage بيكونوا decoupled يعني مفصولين عن بعض.

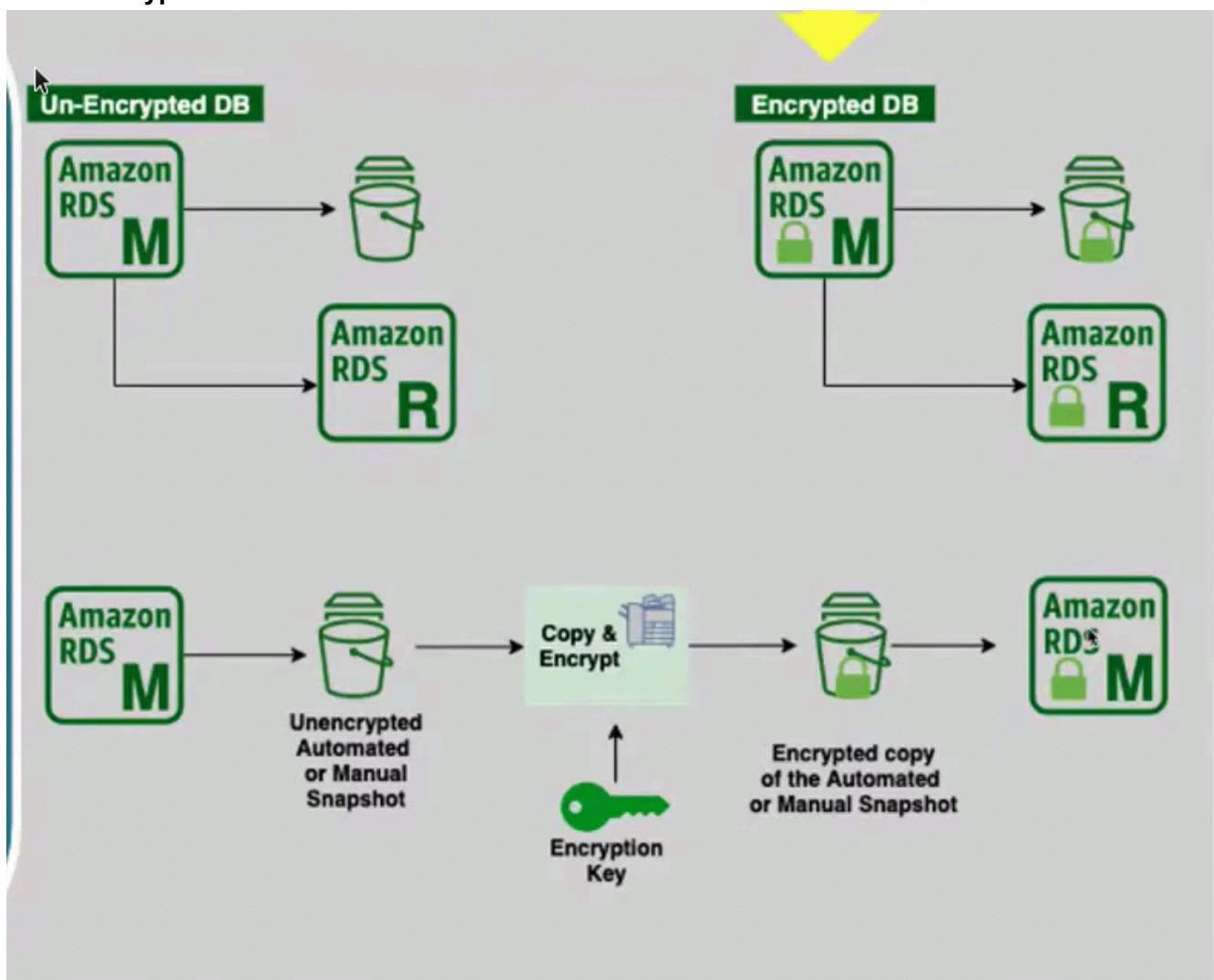
RDS - Storage Auto Scaling

دي RDS Feature أن ال storage بتاعته لو ال storage بتاعته قربت تخلص وبشكل Automated ومتغير. بتقدر ت enable ليها لو هي existing او هتعمل واحده جديدة. Downtime

Info

ومش كل ال DB Engine الي في RDS بي Support نقطة دي.

RDS - Encryption



نفس حكایة ال EBS

ال RDS DB و RDS Client بقدر تكون Communication TLS/SSL encrypted. لو عايز اكلم مع RDS Client (RDS Client app هو ال Application (خ بالك هنا بكلم علي app هو ال Application

- فبتحط AWS Docs Client Root certificate من Client Certificate لـ DB
- ال Client بيسعمل Certificate دي عشان يصل لـ DB , فا لما بيروح لـ DB بيفرلها وربني certificate (app) الي بيشوف ال Operation Certificate بقى مطابقة هيدا يعمل ال Operation ويشتغل ولو لا ساعتها مش هيكمل ال Operation Certificate

Transparent Data Encryption (TDE)

دي موجوده في Oracle و MS SQL فقط، ودي قبل ما APP يكتب ال data على DB اعملها Encryption قبل متكتب على DB.

IAM DB Authentication for MySQL and PostgreSQL

دا اني محطش Password لي RDS DB بتعاتي وبدلها استعمل ال IAM Authenticate لي DB ، وال AWS RDS Authentication دا بيتعمل عن طريق Token بيكون مدته لي 15 دقيقة، ودا بيطلب من

ال IAM User or Role لازم يكون معموله في DB بنفس الأسمو يعني لو عندك user اسمه omar-dev ، لازم تعمل IAM User or Role جوه الـ MySQL/PostgreSQL بنفس الاسم omar-dev .

لازم يكون في login .token حتى لو معاك TLS connection ، لو مش عامل token ، مش هتقدر تعمل Client to DB SSL Encrypted .

RDS Proxy

هي Service بتقعد بين Application و RDS DB ، تخيلها ك middle layer بتعمل:

- الـ connection pooling (تحسين الأداء)
- الـ credential management (أمان وأسهل إدارة)
- الـ failover handling (سرعة وiability)

Security Enhances

بيشتعل ازاي؟

1. IAM Authentication

- الـ App بيروح لي token ياخد IAM role عشان يستعملها مع RDS Proxy واخرها 15 دقيقة.

الـ App بيوصل لي RDS proxy بيستخدم ال token بدل ال password .

الـ Secret Manager يجيء من RDS Proxy بعد ما بيستعمل token مع RDS Proxy ، بيروح ال Credential retrieval username/password

الـ Proxy بعدها بيستخدم ال credentials دي عشان يصل لي DB بنیابه عن ال app .

الفكرة هنا مفيش hardcoded passwords على password rotation كل حاجه ، بيحصل Manage ، الـ IAM هو اللي بي

Hands-on

Connectivity & security		
Endpoint & port	Networking	Security
Endpoint database-1.cj089x8av8xn.us-west-1.rds.amazonaws.com	Availability zone us-west-1c	VPC security groups default (sg-def2fcbb9) (active)
Port 5432	VPC vpc-fa90249e	Public accessibility No
	Subnet group default-vpc-fa90249e	Certificate authority rds-ca-2019
	Subnets subnet-e157a7bd subnet-d22568b6	Certificate authority date Aug 22nd, 2024

- في Endpoint دي اللي بتربطها بال application بتاعك .
- في Public accessibility ودي انت بتتشوف عايزة تبقا على النت ولا لا .

- في encryption عشان وانت بتتكلم معاها بيعملك certificate و بتتفا مع ال RDS.
 - في العادي بيعملك KMS Key عن طريق ال encryption .
-

Amazon Aurora

• ايه هي ال Aurora؟: نسخة Enhanced من RDS مع توافق مع MySQL و PostgreSQL .

Characteristics

أسرع 5 مرات من MySQL العادي و 3 مرات من PostgreSQL .

في RDS ال EBS الى بتتفا على instance بتخزن فيها Aurora Data فا ال Cluster volume بتخزن two .Replicas لانهم كلهم بيقولوا شایفين نفس Replicas و دا بيعمل Sync مع كل volume .

ال Automatic Storage Scaling (من 10GB لحد 128TB) .

ال Read Scaling بيدعم Aurora Replica 15 في نفس Region و AZs مختلفه .

ال Aurora Auto Scaling بي Scale لي Aurora Replicas على حسب ال need سواء up او down (لازم يكون على الأقل عندي واحد). Aurora Replica .

ال Instance Scaling بـ modifying aurora instance سواء up or down لي scale .type/size وهي أني اعمل بأني اعمل up او down .

ال Support بت Save Redo Log files قبل Restart/Crash عشان لما ترجع يعمل ال Transactions دى .

ال Cons: أغلى من RDS بـ 20% .

Connection Management & Endpoints

عندنا 3 أنواع لي Aurora مع ال Endpoints :

1. ال Cluster Endpoint Connect على Primary Aurora Instance : دى لو عايز ا
2. ال Reader Endpoint Connect على Aurora Replicas : لو عايز ا Read only واعمل
3. ال Instance Endpoint Connect على Specific Aurora replica : لو عايز ا

Aurora Replicas & Backup

ال Aurora Replicas بتقدر ترجع ال data مكتوبه في اقل 100msec , و دا بسبب ال Cluster Volume .

عشان Availability increase لي ال Aurora Replicas تكون failover target ، يعني لو حصل و primary instance promote هت تكون aurora replica fails ال primary instance fails .

ال Aurora database Backup بتدعم كل كلام الي في RDS الخاص بي .

Aurora - Infrastructure Security and DB Authentication

ال Aurora بتتحط في VPC ولازم يكون فيها Two subnets في AZ مختلفة .

بنقدر تحط عليها Aurora وتنقول مين يكلم مع Aurora ومين لا
الـ TLS/SSL بيدعم الـ Aurora Endpoint في Connection
الـ IAM DB authentication نفس الكلام اللي قولناه في RDS
ممكن تتكلم مع Aurora API's في نفس VPC بأسعمال VPC Interface endpoints Service نفسها بدل ما تروح
.NAT Gateway او تستعمل Public Network تعديها على

Aurora - Encryption

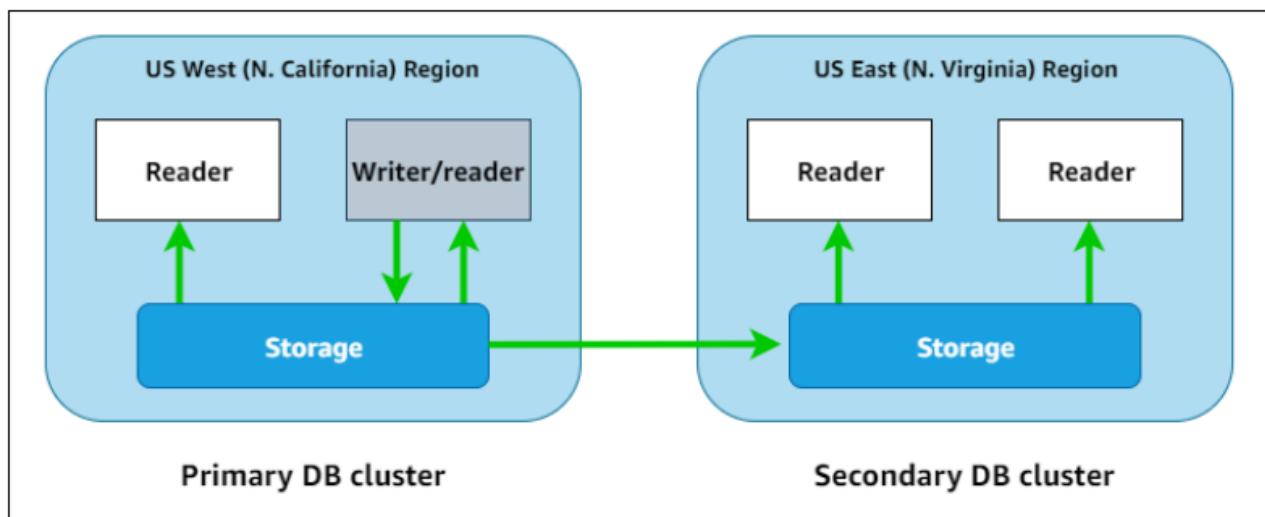
الـ Aurora Encryption زي بالظبط الـ RDS Encryption اما بتكون Enable لي Cluster Encryption و الـ snapshots, backups, replicas كله من الآخر.

متقدرش تغير الـ encryption status بتاعت cluster لي restore في encryption status ولكن نقدر ن restore الـ unencrypted Aurora DB Snapshot .encrypted aurora DB Cluster

Aurora - Global Database

هي Feature من Amazon Aurora بتسمح إنك تعمل up to 10 Regions في Secondary Clusters والـ Clusters دي بتكون:

- Read-only
- بتعمل Replication من الـ Primary Region بـ Latency قليلة جداً (عادة أقل من ثانية)



الـ Secondary DB Clusters بيكون بيعمل synchronization مع الـ Primary Cluster، بس خد بالك:

⚠ Warning

الكلمة "synchronizes" معاناها بيعمل Replication اللي بيحصل في الـ Primary، بس مش معاناها إن كل عملية Write هتنسجل فوراً في الـ Secondary Cluster لأنـ Asynchronous Replication هنا بس سريع جداً (Lag) بيكون أقل 1s غالباً، عشان كده بيفolloوا عليها "synchronizes" بالمعنى العام مش بالمعنى التقني بناءً Sync Replication. الـ Secondary بنكون Readonly خد بالك

الفایدة لو حد طلب منك System يكون:

- عند Recovery Point Objective (RPO) أقل من ثانية، يعني مش هتسخر أي Data تقربياً
- و Recovery Time Objective (RTO) أقل من دقيقة، يعني تقدر ترجع تشغيل الـ DB في Region ثانية بسرعة لو حصل Disaster ، دا لأن Aurora replicas بتكون على latency دا يدخلني الـ dedicated infrastructure تحت الثانية

هناك في حاجة اسمها Aurora MySQL Cross Region Replication ولكن تحتاج وقت Global Database ودي هي الـ data transfer charges عكس الـ global تكون مدخلين ليك الـ charge مع سعرها.

Aurora - Backtracking

الـ feature دي بتخليلك ترجع الـ DB Cluster بتابعك لوقت معين (زي ما يكون rewind)، من غير ما تعمل Restore من Backup أو Snapshot.

Info

الـ Backup مش بديل للـ Backtrack

هو feature تساعدك تراجع عن غلطة حصلت بسرعة ومن غير Downtime كبير.

مش موجوده في RDS

Aurora - Multi-Master Cluster

دي بت Enable لي يكون عندي كذا Primary ، يعني عندي كذا aurora instance فيها اكتر من read and write. دي بستعملها لو عندي app ميتحملش الـ DB اللي ممكن يقع فيها.

Amazon Aurora Serverless

الـ Aurora Serverless ده نسخة Automated Aurora اللي بتدعم MySQL و PostgreSQL، لكن مع ميزة إنها تضبط نفسها بنفسها حسب الاستخدام. يعني مش تحتاج الـ Capacity Manage ولا حتى تفك في Scaling.

طب ازاي بيشتغل؟ بيكون عند AWS شوية Aurora Instance بيكونوا Warm pool يعني جاهزة علي طول انها تشتعل، فا موضوع مش هياخد ثواني علي بعض.

يدعم الـ Client/app connections في TLS/SSL

Characteristics

1. Auto-Scaling

- لو الحمل زاد فجأة (مثلاً: تطبيقك اتعمل عليه هجوم طلبات)، الـ Aurora Serverless هيزيد الـ Resources أوتوماتيك.
- لو الحمل قل، هينقص الموارد علشان توفر التكلفة.

2. الـ Pay-Per-Second: حسب الوقت الي اشتغلته هدفع عليه بدلاً ما ادفع علي service شغاله علي طول.

3. مناسب لـ Workloads الي بتبقا بشكل Separated

Amazon Redshift

الـ **Redshift** هي Databases مخصصة (OLAP) وهي queries علشان تعمل على large size of data store لـ historical data فقط، يعني هو يقدر يقدر بـ petabytes. الـ **Analytics** هي بـ **OLAP**، يعني هي بـ **Historical data** فقط، يعني هو يقدر بـ **large size of data**. الـ **Real-time analytic** هي بـ **real time analytic**.

Characteristics:

أسرع لأنّه قادر على العمل **Parallel processing**.

(Data analysis) **Tableau** و **Quick Sight** لـ BI زى Support.

الـ **Cost**: تدفع حسب الاستخدام (Pay-as-you-go).

Redshift Architecture

- الـ **Cluster** = مجموعة من **nodes**.
- الـ **Leader Node**: يأخذ queries من الـ clients، ويقسمها.
- الـ **Compute Nodes**: يسيروا البيانات ويعملوا على execution queries.
- كل الـ cluster دا بيقى جوه **AZ** واحدة بس.
 - AWS مش بتوزع نفس الـ cluster على أكثر من AZ.
 - السبب: Redshift OLAP heavy latency.
- بيـ store لـ data على شكل columns وـ transactions SQL DB من 10x أسرع دا اللي بيـ خليـها مخصوصـه لـ OLAP.
- لو بتستخدم **Single-Node Cluster**.
 - عندك node واحدة فقط → منـش HA.
- لو بتستخدم **Multi-Node Cluster** (Leader + 2+ Compute).
 - البيانات بتتوزع across compute nodes.
- لو وقعت node، Redshift قادر على data redistribution/rebuild من داخل الـ cluster.
- بـ ... كلـه لـسه في نفس الـ AZ.
- في **Redshift RA3 instances**.
 - البيانات نفسها بتتخزن في **Redshift Managed Storage** (على S3 + EC2 mix).
 - حتى لو وقعت البيانات مش بتضيع، بيـعمل replacement بـسرعة.

Redshift Concurrency Scaling

دي Feature بتـ support المـulti queries منـ غير مـيـاثـر على Performance وـدا لأنـه يـزوـد في cluster.

Redshift - Backup & Restore

بيـاخـد Backups بشكل Automaticـly كلـ 8 ساعات، ولـما اعمل Restoring منـ Backup هـيـعمل New Cluster، ومـمـكن تكون في نفس الـ AZ او Different AZ.

Manual backups

الـ snapshots مـمـكن تكون across region عـادـي وـدا يـنـفع في disaster recovery.

Redshift - Data Sources & security

المصادر الي ممكن ياخد منها الـ Data زى الـ S3 وبيكون فيها Parallel reads او .EMR, EC2, Dynamodb, data pipeline (S3 OR Dynamodb) or Database migration service او Security support لـ keys manage (لازم تشغله بنفسك), تقدر ت manage data at rest بتو عك عن طريق encrypted data in transit و بتكون Snapshot او KMS او HSM او

Redshift - Workload Management(WLM) & Enhanced VPC Routing

الـ WLM دي طريقة عشان ارتق انهي Queries تبدأ ويكون عندي Queue هي الي بترتب دنيا متخليش دنيا عشوائية وميكونتش فيه عشوائية و Queues لا نهائية

الـ Enhanced VPC traffic دي بتجبر الـ redshift cluster انه يكون مابين s3 و redshift cluster عن طريق VPC Endpoint

Redshift Serverless

نسخة Automated Resource Manage حسب الـ Load من غير ما تـ Reports او business reports مثلـ Manage لـ Reports او Dashboards مثلـ.

بتقدر تعمل Queries على Redshift Spectrum Feature أساسها S3 بـ Scanned number of bytes بـ SSE encrypts data in transit and at rest وهو بيكون عبارـ VPC Cluster بـ VPC شغال فيه Queries بـ Multiple nodes فـ لازم يكون Redshift جواـ VPC، بتـ

2. Non-Relational Databases

DB شرحت معناها في AWS لما بتـ Scale بـ Scale بشكل Non Relational في

Amazon DynamoDB

الـ DynamoDB من نوع Sem-structured and unstructured data Data هي بـنـدـعـمـ الـ OLTP database شغالـةـ عن طـرـيقـ Key-Value او Document وـ هيـ سـرـيـعـةـ جـدـاـ وـ بـتـسـتـحـمـلـ الـ High Load لأنـهاـ بتـ scale علىـ حـسـبـ scale وـ كـمـانـ هيـ Serverlessـ.

الـ Replicas بـتـنـوـزـ عـلـيـ الـ AZsـ

عـنـهـاـFlexible schemaـ يعنيـ معـنـدـهـاـشـ Schema-lessـ مـحـدـدـهـ مشـ هـنـفـرـقـ معـاـهاـ

بـتـسـتـعـمـلـ HTTPSـ فيـ transportـ

عـنـهـاـunlimited scalingـ وـ high performanceـ downtimeـ بـأـيـ workloadـ مـيـتـأـثـرـشـ بـأـيـ single-digitـ scaleـ علىـ أيـ millisecond latencyـ

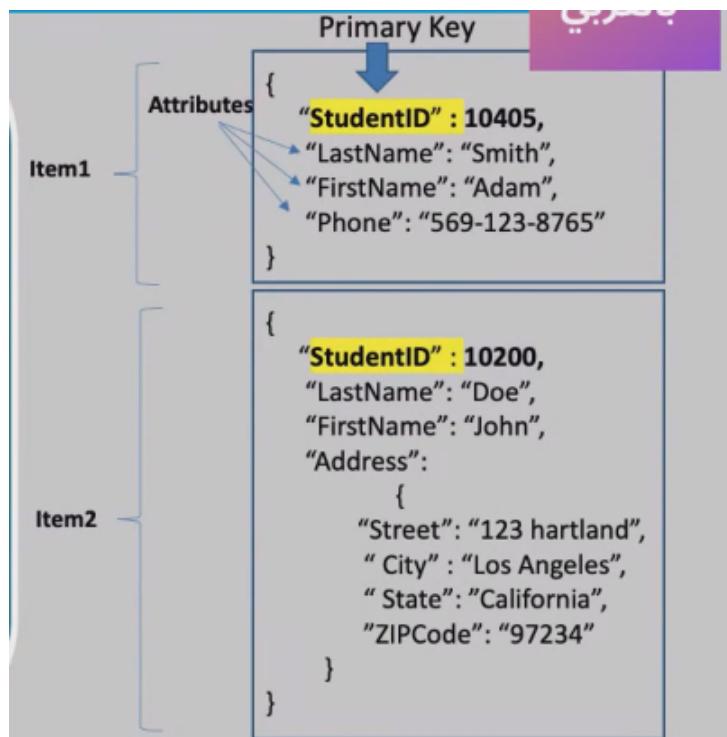
عـنـهـاـFeature multi masterـ وـ multi-regionـ

⚠ Warning

خد بالك لو فالك NoSQL او مش multi master or multi region structured dynamodb ساعتها هختار و لكن لو قال Aurora او SQL معقدة هتبقا Queries

هي durable database عندها built-in backup and restore أسرع 10x Reads بتعمل (Cache) بتخلي الـ DAX و عندها الـ Cache.

مش بتدعم complex joins or queries



الـ DynamoDB يخزن Data على شكل Table, كل table عبارة عن مجموعة من data items , كل item مinfinity و بيكون attributes group of attributes في relational DBs كل attribute دى زي columns في item .
يعدى 400KB كل item هو .
فا هو زي مقولنا عباره عن Key and value or set of values ليه .

بتعتمد على Single key في حالة ال Partition/Primary key و فيه بقىت ال items Unique و بيكون Item و دا أول جزء في ال Sort key حاجة اسمها Compound key ولو هيقيا في حاجة اسمها Attributes .

DynamoDB - Data Consistency Model

ال Eventual Consistency reads

- مش بيقرأ الحاجات الي لسه كاتبه حالا في نفس الوقت :Strong Consistency Reads
- بيرجع آخر حاجة data لسه دخله DB بتاعتي.

ال dynamodb بتدعم الاثنين ولكن ال Eventual consistency هي default لو عايز ال Strong consistency تقدر تفعلها عادي.

DynamoDB - Capacity Units

دي ال Unit الي بيفرأ فيها و يكتب فيها و هما:

- Read Capacity Unit(RCU)

- لو بستعمل Two item 4kb وآخر item 8kb في ال One RCU ساعتها هاخد Strong Consistency ؟ ساعتها هاخد .one RCU مع الأتنين eventual read بياخدوا 8kb يعني 4 لي ال eventual read الواحده ودا هيكون RCU .1kb item لي كل one WCU في write بيحسب Write Capacity Unit(WCU) .
 - لو ال throughput Exceeds Read or write requests حدتها ال DynamoDB هيأخذ اللي انت محدده ويرمي الباقي
-

DynamoDB Features

DynamoDB - Auto Scaling

هو مش بيعمل على مستوى ال instance ببناعته لا هو بيعملها على مستوى ال application يعني بيزود ال RCU كل لما يلاقي أن App محتاج اكتر، زي اي Auto Scaling بيعتاج بدایة وحد معين. فا هو شغال على on-demand, بيحاسبك على unpredictable workloads، مناسبه لي Request

DynamoDB - On demand Backup and Restore

زي ال Manual Snapshots في RDS

- بتعمل backup لي كلها وبيكون tables long-term retention and archival
 - مش بياثر على ال Primary DynamoDB
 - بتفصله persist لغاية لما امسحها Manually
 - بتكون في نفس Region بناع Source
-

DynamoDB - Point in Time Recovery(PITR)

زي ال Automated Backup في RDS

- بتحذلوك ال backups بشكل مستمر و incremental.
 - بتحمي DB ببناعتك من انها تتمسح منغير قصد لأنها تقدر ترجع لك DB لو اتمسحت (هيرجعلك لغاية اخر 5دقائق)
 - تقدر ت ال new table Restore سواء في نفس region او غيرها
 - ال ال retention period هي 35 يوم ثابت
-

DynamoDB - Time To Live(TTL)

دي Items ببناعت Feature لـ DynamoDB هو بشكل Automatic ويسمحها هو بشكل Flows، مناسبه لي temporary data اي

DynamoDB - Accelerator (DAX)

هو Cache مخصص لـ DynamoDB بيخلي ال Reads أسرع

بيخلي ال reads تكون بال microseconds latency

ال DAX بيكون Eventually consistent reads

✓ AWS Best Practices

ال Availability محتاجه تكون موجوده على الأقل في 3 AZ مختلفين في 3 nodes Production

Amazon ElastiCashe (it is not a DynamoDB feature but i have to Explain it before i compare it with DynamoDB Caching(DAX))

هو caching layer عشان تحسن ال DB Performance .
يقلل الحمل على Databases الرئيسية عن طريق Data Cache لل المكرر استخدامها.
مناسب لـ Apps إللي فيها Read كثيرة (زي Social Media).

بت Store data Sessions لي عشان ال Data متراوح مني و تكون Sync مابين ال Instances .
بنبص على data دي بحيث لو واحده يحولها لي تانية

ال ElastiCache بندعم نوعين:

- ال Redis: يقدم ميزات إضافية زي client persistence, pub/sub, failover, snapshot, security . execute commands قبل ميعمل auth token
- ال Memcached: أبسط وخفي، ملوش replication أو persistence مش بيكون Encrypted

Amazon ElastiCache Engines – Use Cases

Use cases for Redis include:

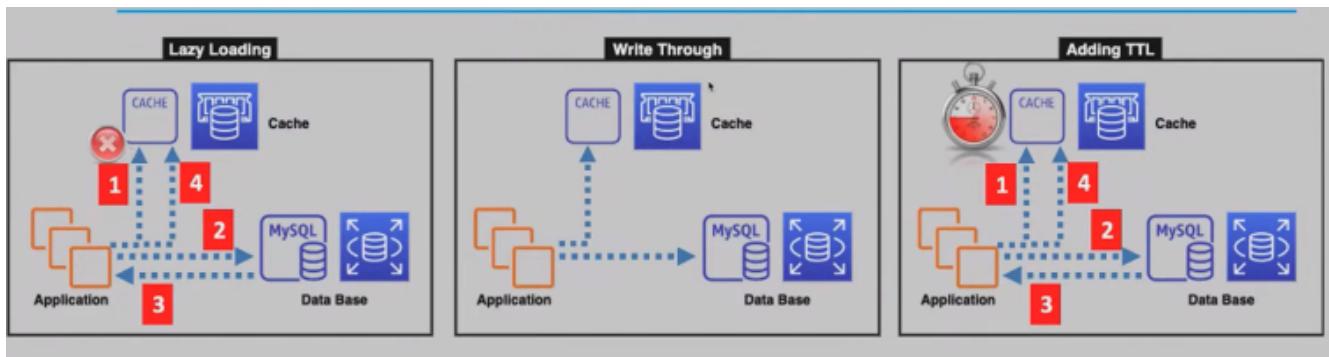
- Web.
- Mobile Apps.
- Healthcare Apps.
- Financial Apps.
- Gaming.
- Ad-Tech, and
- IoT.

Use cases for Memcached include:

- Cache contents of a DB.
- Cache data from dynamically generated webpages.
- Transient session data, and
- High frequency counters for admission control in high volume web Apps.

بتحتاج تغير Logic عشان تستعمل ال Caching دا.

بنقدر تعمل ElastiCashe Cluster ال Nodes في EC2 Instances عباره عن VPCs وبتكون محميه بي Security Group .
ال APP عشان ي connect معاهما ه يحتاج ال Endpoints .
مش بيكون accessible من الأنترنت فا دا حمايه قوية ليه .
ال nodes بتكون Reserved او on-demand تكون Spot لأنها بتتسحب منك في أي وقت
.automatic failed nodes بيعبر ال ElastiCache بشكل



عدها :Caching Strategies

- الLazy Loading: هو بيتأكد ان data موجوده في cache ولا لا لو ملقهاش هبيعدt cache miss ودا معانها مش موجوده وهيجها من DB ويديها لي App وبعدين احطها في caching ويفضل يخزن في data ومش بيمسحها
 - الWrite Through: هعمل Query لي Cache و DB في نفس الوقت لو ملقهاش هيروح يمليها في cache بتاعتي عشان تكون اسرع مشكلتها بردو في حتن تخزين هيأخذ مساحه
 - الAdding TTL: زى Lazy loading ولكن بيمسح caching بعد وقت معين يعني بيعملها Expiration data لو محدث استعملها فيه تنتمسح.

الفرق بين ElastiCache و DAX

- الـ DAX مخصص فقط لـ ElastiCache، أما DynamoDB ينفع مع أي DB.
 - الـ DAX يوفر latency أقل.

DynamoDB - Stream

ممكن تستعملوا كا trigger لو حصل changes في item tableFeature بترافق ال real-time order وتبعد بتاعتكم، ال data changes item بتاعتك، ال rest order بتاعتك.

DynamoDB - Global Table

هو أني اعمل نفس Table في Regions مختلفة ودا موجود في Aurora بردو ولكن ميزة DynamoDB أنك تقدر تعمل Multi-Master وعنه قدره انه ي user base global عشان يخدم massively scaled application.

ای change بیحصل فی one region automatic وفی خلال ثانیه بیسمع فی بقیت

Amazon DocumentDB

MongoDB مع Compatible سکون Document-Based Databases هم عباره عن

عند نسخه Aurora بالظبط نفس الـ Architecture ونفس طرقه Cluster Volume Replicas 15 ، Primary

- تخزين JSON بشكل مرن (مش محتاج Schema ثابت).
 - التخزين بيزيد بشكل Automatic.
 - مناسب لـ Apps إلى، فيها Catalogs Complex Data (مثل Complex Data).

Amazon Neptune

هو Databases Graph عشان Data Manage ال الي ليها علاقة ببعض. عنده نفس ال Cluster Volume Aurora بالضبط نفس ال Primary و Replicas 15 . سريعة في أنك تربط ال Relations ببعض.

مش بيستعمل SQL Queries و Gremlin SQL language .Social Media او Recommendation زي: نظام

3. Further Databases Services

شوية DB Services زيادة تبقة عارفهم علي الأقل

1. Amazon EMR

قبل منكلم علي EMR ناخذ خطوة لورا

What is Hadoop

data-intensive ، Distributed computing و Scalable و Reliable ، هو Open Source java framework دا يعني بيقدر يعمل processing لي كمية كبيرة من data على large cluster of computing .large datasets process لي large datasets بشكل سريع . بيسعمل MAP Reduce Model وده بيقدر يعمل

Amazon Elastic Map Reduce (EMR)

هو Managed Cluster Service بستخدمها عشان AWS Run جوا Hadoop Process . يعني: بيعمل بـ Anaylsis او Process . أدوات مفتوحة المصدر زي Spark و Hadoop . ميدرش يشتغل علي real-time data ingestion

ال EMR Clusters بتشتغل في single AWS AZ . Cost effective processing ودا لانه بيعمل معين وبعدين يقل تاني فا دا بيخليها EMR Cluster تقدر تنسعمل On-demand او Spot instances و بتدعم auto-scaling . المميزات:

- بيسمح لك تشغيل Cluster من السيرفرات بسرعة عشان تعالج بيانات ضخمة.
- مناسب لـ تحويل البيانات، Machine Learning، أو إنشاء Data Lakes .
- دعم لـ HBase، Flink، Presto .
- مثال: شركة بتجمع بيانات من مواقع كثيرة وتعملها Process عشان تطلع تقارير شهرية.

Comparison Between EMR and Redshift Spectrum

	EMR with Apache Hive	Redshift Spectrum
Compute	Cluster-Server based	Serverless
Use case	SQL based queries - Great for scale-out processing like scans, filters, and aggregates.	SQL based queries - Great fit as it can scale out to thousands of nodes to pull data, filter, project, aggregate, group, and sort.
Ingest the entire data from S3 into the service to process it?	Not required.	Not required.
Complex querying and joins use cases? (very critical for analytics).	It gets very slow as the data size and number of nodes increases.	Very efficient.
Billing	Pay for the compute.	Pay for the data scanned.

2. Amazon Athena

ايه هو؟: خدمة **Serverless** بـ**Tech** تعمل **SQL Queries** مباشرة على ملفات في **S3**

يستخدم حاجه اسمها **schema-on-read** ، وهي انها مبتاخدش **schema** معينه ولكن هي بتعمل **queries** على ال **files** موجوده بشكل معين عشان تستخرج ال **data** الي تحتاجها.

يدعم **.JSON**، **CSV**، **Parquet** في **query unstructured, semi-structured, and structured data**

الCost هي: **5\$** لكل **TB** من البيانات اللي اتعمل عليها **.scan**

مثال: تعمل **analysis** لي اي **services Logs** المخزنة في **S3** بدون ما تحملها على قاعدة بيانات.

لأنها **Serverless** دا هيخلية **result** في **s3** على شكل **CSV Format** وتفضل 45 يوم فقط الا لو عملتها **store** في مكان ثاني بأيدك.

ال**query result** ممكن يكون **encrypted in s3** او في **query**

Comparison Between Athena Vs Redshift spectrum

	Athena	RedShift Spectrum
Compute (Serverless?)	A completely serverless service.	Redshift spectrum itself is serverless. One or more RedShift clusters are required (higher cost).
Complex Joins, Queries and Aggregations.	Not meant for this use case.	Ideal for this use case.
Ad-hoc SQL queries.	Ideal for this case.	Not meant for this case.
Can query data in S3 without loading it.	Yes	Yes
Large data lake users that want to run concurrent BI and reporting workloads.	Not meant for this use case.	Perfect fit for this use case.

3. Amazon QuickSight

فكرتها إنها بتخليك تعمل **dashboards** و **visualization** على الـ **.data**.

خطواتها:

1. بتحدد **Data Source** (ممكن يكون Excel sheet, أو حتى S3, Redshift, Athena, RDS).
2. بعدين بتعمل **Dataset** (بتجهز الـ **data** وتفلترها/تنصفها).
3. تعمل **Analysis** (queries + exploration).
4. تطلع **Visuals** (charts, graphs, KPIs).
5. تجمعهم في **Dashboard** وتشاركها مع النايم أو الـ **stakeholders**.

- **Use cases:**

- Business analytics.
- Monitoring KPIs.
- Visualizing streaming data (واحدة من Kinesis → S3 → Athena → QuickSight).

4. AWS Glue

قبل AWS Glue، نرجع خطوة ونفهم يعني فيه **ETL**:

- الـ **Extract**: تسحب البيانات من مصدرها (...Database, API, Files).
- الـ **Transform**: تنظفها وتعيد تنظيمها أو تغير الـ **Schema**.
- الـ **Load**: ترفعها لمكان تقدر حللها فيه (Data Lake أو Data Warehouse).

AWS Glue

هي خدمة ETL Serverless Managed بالكامل، بتساعدك تجهز البيانات للتحليل أو الـ **Machine Learning**.

يعني: بدل ما تكتب Scripts معقدة وتشغلها على سيرفراتك، Glue بتعملها أتوماتيك وتدبر الجدولة والتنفيذ.

المميزات:

- يكتشف Schema للبيانات أتوماتيك وتعمل **Data Catalog**.
- متكاملة مع **S3**, **Redshift**, **RDS**, وغيرها.
- تدعم Jobs بالـ **Python** و **Spark**.
- transit و rest في data encrypted.

مثال: شركة بتجمع بيانات العملاء من CRM وبيانات الطلبات من ERP، وبدمجهم في Data Warehouse موحد للتحليلات.

الـ **sequence** الصبح بيكون:

- عندك **data** في **S3** (أو أي source ثاني).
- الـ **Data Catalog** AWS Glue Crawler بيقراها ويعمل لك **ETL** (AWS Glue Job).
- النتيجة تتحط في **S3** أو **Redshift** أو غيره، وتكون جاهزة للـ **Athena** أو **QuickSight**.

Use Cases

- Run serverless queries against an S3 data lake.
- Build a data warehouse from different, disparate, data sources.
- Create event-driven ETL pipelines with AWS Glue and Lambda.
- Understand stored data assets.

Comparison Between AWS Glue and EMR

	Glue	EMR
	Is a fully managed, pay as you go, ETL tool for big data. It can transform the data and make it ready for analytics purposes.	Is a managed big data platform known for its speed and ease of data conversions. It also supports ETL jobs.
Platform	Serverless. Based on Hadoop Framework. Runs on top of Hadoop Spark.	Server-based. Based on Hadoop framework. Supports many of the Hadoop services including Spark, Hive, and Pig among others.
Cost	More expensive.	Less expensive.
ETL Operations – Performance and flexibility	Higher.	Lower compared to Glue.

5. Amazon Kinesis

Streaming Data

هي data بتجيلك وبنكون حجمها صغير (KBs or MBs) و نتيجي من كذا sources بشكل مستمر

Kinesis

Big data Analytics في AWS ، وهي بتسخدم مع IoT و AWS Managed Real-time streaming data service هي

Kinesis Data Streams

ال Data في Kinesis data streams انه تستقبل data في real time .milliseconds

ال Data بنكون موجوده لمدة 24 ساعه by default ولكن ممكن تتغير لي 7 days

Data durability High availability مع replicas data sync يعمل Kinesis data على 3 AZs مختلفه و دا هيديلك

ال Use cases

- Accelerating log and data feed intakes
- Real time metric and reporting analytics
- complex stream processing

ال set of shards هو عباره عن Kinesis Data Stream .records كل shard ده بنعتبره زي pipeline صغيرة بتسقبل

إنت ك architect أو consumer بتحدد عدد ال shards بنفسك على حسب:

- حجم ال data اللي متوقع بيجي من ال producers
- معدل الكتابة (write throughput).

- معدل القراءة (read throughput).

كل record جاي من producer لازم يكون معاه **Partition Key**. هو اللي بيحدد أي record هيتخزن في أي shard. يعني Partition Key اللي ليها نفس الـ records هتدخل كلها على نفس الـ shard.

خد بالاك:

- كل shard ليه حد أقصى لكتابه: 1MB في الثانية أو 1000 record في الثانية.
- وليه كمان حد أقصى للقراءة: 2MB في الثانية.
- لو زودت عن كده، هتبدأ تشفوف throttling (بيوقفك).

علشان كده الـ service بناء Architecture بيبقى:

1. تحط Partition Key منطقى (زي orderId أو userId) علشان توزيع الحمل يبقى متوازن.
2. تزود أو تقل الشارذات (shards scaling) على حسب الحمل.

الـ shards تقدر merge ليها وتقدر تعملها split ودا هيكون مهم في حته Costs.

Comparison Kinesis Data Stream Vs SQS

	Kinesis Data Streams	SQS
Intended use	Real-time ingestion and processing of streaming big data.	Reliable, highly scalable hosted queue for storing messages as they travel between computers.
Ordering	Ordering of records, and ability to read/replay records in the same order by several Kinesis applications.	SQS FIFO queues can guarantee message ordering.
Use when your requirements are any of the following:	<ul style="list-style-type: none"> Routing related records to the same record processor (consumer). When we need ordering of records (Important in case we need to keep the order of logs messages the same at the consumer as they arrived from the producers). When we need multiple applications to consume the records concurrently. The ability to consume the same records few hours or couple of days later. 	<ul style="list-style-type: none"> Messaging semantics (ack/fail) and visibility timeout are required. You need the queue to scale transparently without pre-provisioning shards.

Kinesis Data Firehose

خدمة Data Streams أو تحدد زي provisioning، ومش تحتاج تعملها **fully managed**، real-time

إنت بس بتحدد destination (S3, Redshift, Elasticsearch/OpenSearch, Splunk) وهى بتأخذ الـ data وتوصلها هناك بشكل أوتوماتيك.

بتعمل auto-scaling حسب حجم الـ data ، مفيش بقى قصة AWS Manage بـ throughput limits أو AWS shards هي AWS بتـ كلـ.

بس خد بالاك: مش معمولة علشان تحفظ الـ data فترة طويلة... هي مجرد pipeline للنقل (buffering بسيط بالثواني أو الدقائق)، وبعدها تكتبها على الـ destination.

• Use cases:

- IoT analytics
 - Log analytics
 - Clickstream analytics

- Security monitoring
- وكمان ممکن source بيقى Kinesis Data Stream ليها وتاخد ال Data منه.

Kinesis Data Analytics

Real streaming data analysis processing Firehose. فكرتها إنها بتخليل تعمل layer Streams أو Firehose في analysis processing لـ streams. time

- إزاي؟ عن طريق SQL code أو بتركيب Data Analytics Application.
- مثل: الـ Data Stream جاية clickstream من موقعك.
- بدل ما تخزن كلـ raw على S3، ممكن بالـ Analytics عمل aggregation (زي: كام click في الدقيقة لكل user)، وتبعـ .Lambda أو Firehose النتيجة على S3 أو Lambda.
- دي مناسبة لو عايز insights real-time من غير ما تستـ ETL بعدين.

Kinesis Service

- الـ raw pipe الأساسي، إنت بتديره، تحدد shards، تتحكم في retention (افتراضي 24 ساعة، وممكن تزود لـ 7 أيام أو أكثر).
 - الـ Kinesis Firehose scalable destinations، مفيش data وتدبيها Pipe managed لـ shards، بتاخد الـ data وتدبيها بشكل destinations.
 - الـ Kinesis Data Analytics الـ processing layer لـ analysis SQL وتعمل SQL تكتب تكتـ Kinesis Data Streams.
-

6.Databases Migration Services

هذا هترف إزاي تنقل الـ Data بـ AWS بـ AWS بـ سهولة، وإيه اللي عليك وإيه اللي على AWS في حماية البيانات.

What is DMS?

- خدمة بـ سهـلـك عملية نقل الـ Databases من أي مكان (Cloud On-premises) أو AWS، أو حتى من لأـ AWS لأـ Destination ثاني.
- بـ شـغلـ مع معظم أنواع قواعد البيانات (S3, SQL, NoSQL, Data Warehouses).

Migration Types

• Homogeneous Migration:

- نقل بين Databases من نفس النوع (مثال: Oracle → Oracle on AWS).
- بيكون سهل لأنـ Schema مش يحتاج يتغير.

• Heterogeneous Migration:

- نقل بين Databases مختلفة (مثال: SQL Server → Amazon Aurora).
- هنا بنحتاج AWS Schema Conversion Tool (SCT). والـ SCT بيـ Target DB (Stored Procedures) علشـن يـنـاسبـ الـ Schema والـ code (زي Target DB).
- لو في حاجـاتـ مشـ قـابلـةـ للـ تحـويلـ، بيـحدـدـكـ تـعـملـهاـ manual.
- كـمانـ بيـشـيكـ علىـ الـ applicationـ الليـ فيـ الـ AWSـ وـيـحـسـنـهـ لـ AWSـ SQLـ codeـ.

Key Characteristics

- الـ **Continuous Replication** data بتنقل بشكل مستمر لحد ما على الـ target.
 - الـ **Wide Support** بيدعم مصادر وأهداف كثير (... MySQL, PostgreSQL, MongoDB, Oracle, SQL Server, S3).
 - الـ **Security** data دائمًا encrypted أثناء النقل.
-

AWS Monitoring

Once your applications are running in the cloud, it's important to keep an eye on their health and performance. In this section, you'll learn how AWS monitoring tools help you track usage, detect issues, and maintain reliability.

AWS CloudTrail (Auditing)

بختلخص في الـ Auditing معناها Who Did What, When, and How فا هي بترافق كل User Activity . Troubleshoot و Analyze و Filter Logs ممكن تعمل.

أنواعها:

1. Management events

- بختلني اشوف الـ operations الي بتحصل على resources و بتبقا enabled by default.

2. Data Events

- بختلني اشوف الـ operations الي بتحصل جوا الـ resources و بتبقا disabled by default.

3. Insights Events

- لو حصل unusual API activities على الاكانت و بتبقا disabled by default.

الـ auditing دى بتضمنلك ان حد لعب في data الي جوا S3 ال الخاص بالـ Log file integrity validation.

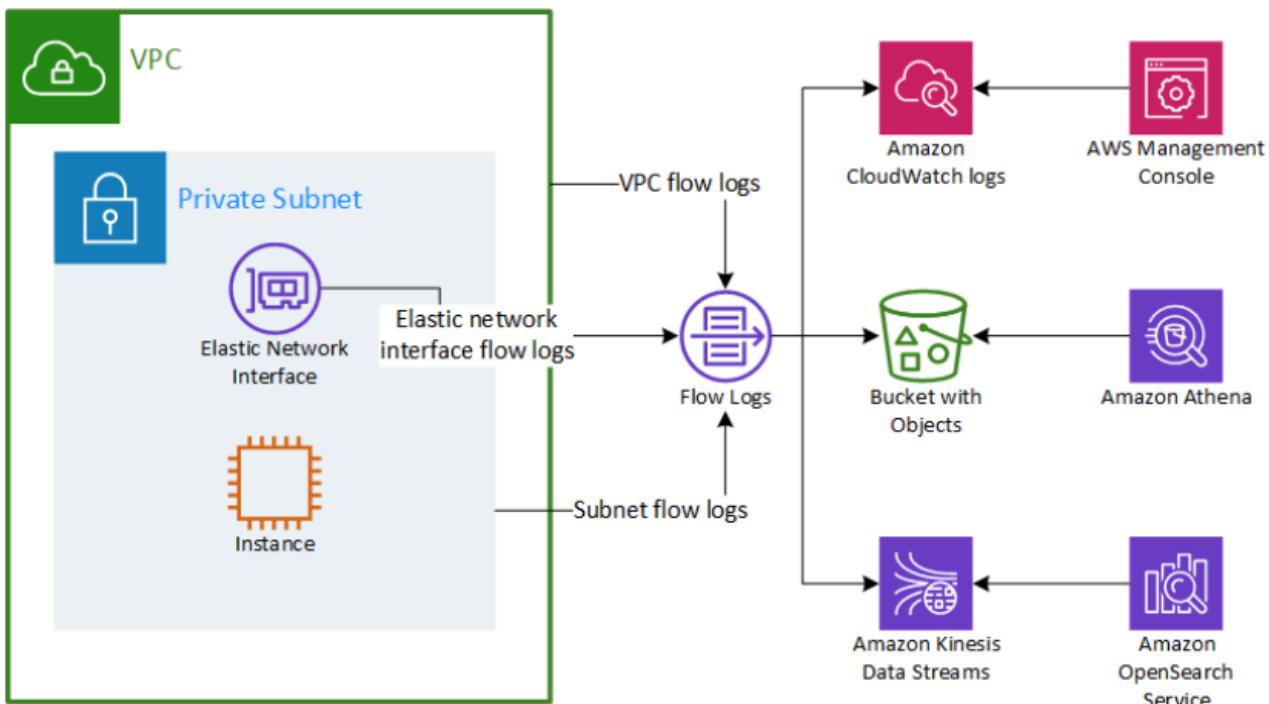
الـ Cloudtrail ليه علاقة فقط بي API Calls بتاعت الـ service.

هيRegional Service by default تقدر تختليها configuration Multi-region بس لما تغير الـ Network.

AWS Network Monitoring

الكلام ده عن Monitoring Services الخاصة بـ AWS في Network، وازاي تقدر تتبع أداء وشغل الـ VPC بتاعك.

What is Monitoring Tools?



الـ **VPC Flow Logs**: تسجل كل Traffic الى بتدخل و تخرج من الـ (IPs والـ ports المستخدمة).

- بتساعدك تكتشف:

- لو فيه subnet معين مش متصل بالإنترنت.

- لو فيه traffic مش طبيعي بين subnets.

- لو في حد بيحاول يدخل من الإنترت على subnet معينة.

- الـ **VPC Flow Logs**: بتوريكLogs على مستوى الـ VPC وأي حاجة جواها فيها Logs.

- الـ **Subnet Flow Logs**

- نفس فكرة الـ VPC Flow Logs، لكن تكون مرکزة على subnet واحد (مثل public subnet).

- الـ **Elastic Network Interface (ENI) Flow Logs**

- تراقب الـ traffic على مستوى الـ network interface معين (زي لو عندك EC2 وحابب تعرف مين اللي بيوصل له).

تقرن توصلها بي Cloudwatch logs عشان تشوف logs و تعملها filter amazon athena query logs مع S3 Storage عشان لو عايز تعمل عليها Network latency or performance

هحتاج تحط لي IAM Policy بتناعنك service زي كدا:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteFlowLogs",
        "ec2:CreateFlowLogs",
        "ec2:DescribeFlowLogs"
      ],
    }
  ]
}
```

```

        "Resource": "*"
    }
]
}

```

Default flow log record example 1

Field	Field description	Example value
version	VPC Flow Logs version	2
account-id	Network owner AWS account	123456789
interface-id	Traffic network interface	eni-123456789abc
srcaddr	Source address for incoming traffic, or the network address interface for outgoing traffic	172.31.18.205
dstaddr	Destination address for outgoing traffic, or the network interface address for incoming traffic	172.31.18.102
srcport	Traffic source port	10530
dstport	Traffic destination port	22
protocol	Traffic IANA protocol number	6 (TCP)

records تكون شكلها كذا.

Problem you will face

- الـ **Public subnet** في public subnet instance مش قادره توصل للنت، الـ traffic هتظهر لو الـ (REJECT) اتم رفضه.
- لو فيه تواصل بين subnets ومتش عارف ليه، الـ logs هتسجل كل التفاصيل.
- اتصال **subnets** ببعض:
- هجمات أو **traffic** مش طبيعي:
- لو لاقيت IP معين بييعت آلاف الطلبات في وقت قصير، ممكن يكون هجوم (زي DDoS).

Where do the flow logs stores

تقدر تختار Data Store في واحدة من التلات أماكن دول:

- الـ **Amazon S3**
- مناسب لو عايز تخزن الـ logs لفترة طويلة وبتكلفة قليلة.

- تقدر تستخدم أدوات مثل Athena عشان تبحث فيها.
 - الـ CloudWatch Logs
 - لو عايز تراقب الـ logs في الوقت الحقيقي وتعمل عليها تحليل سريع.
 - الـ Kinesis Data Firehose
 - لو عايز تـ stream logs لخدمات ثانية (زي Elasticsearch أو Redshift) لتحليل متقدم.
-

Amazon CloudWatch

أول حاجه هي بت Collects Metrics and Logs

- الـ Metrics هي بتراقب الـ CPU , RAM وهكذا.
- الـ Logs هي بتراقب الـ Software و الـ Output بقاعة.

ثاني حاجه أقدر ا Monitor Services AWS بناء على الـ CloudWatch Agent لأنني أنزل On-Prem Resources على الـ Prem.

ثالث حاجه أقدر ا Configures Service لـ Alerts و Actions تحصل بشكل Automatic لما رسالة معينة او تحصل مشكلة في Service تكون عرفتها من Metrics and logs.

الـ CloudWatch بتبع:

- الـ CPU Utilization
- الـ Network Utilization
- الـ Disk Performance
- الـ Disk Read Writes

Further AWS Services

Beyond the core AWS services, there are many additional tools and services that help you build smarter, more scalable, and flexible applications. In this section, we'll look at some of these services like messaging (SNS & SQS), machine learning tools, and how to find and deploy third-party software through the AWS Marketplace.

Elasticsearch - Background

- هي Open-source, near real-time, scalable search and analytics engine
- يتضمن الـ data processing visualize تخزنها وتخليك تعملها
- عادةً بنستعملها مع Elastic Stack (ELK)

- Logstash → ingestion & processing.
- Elasticsearch → indexing & search.
- Kibana → visualization.

Use Cases:

- Log analytics.
 - Clickstream analytics.
 - Real-time application monitoring.
-

Amazon Elasticsearch Service - ES

هي AWS من Fully managed service Elasticsearch بتدیک نفس قدرات لكن بشكل:

- .Secure (IAM, VPC endpoints, encryption)
 - .Scalable
 - الـ Cost-effective (إنت بتدفع بس على usage).
- Features:
 - Built-in alerting.
 - SQL queries for integration مع BI tools.
- الـ VPC Endpoints: الخدمة نفسها مش جوه الـ VPC، لكن تقدر تدخل عليها securely عن طريق الـ .endpoint

Amazon Elastic Transcoder

هي AWS Service managed من بتخليك تعمل media transcoding من غير ما تشيل هم الـ .infrastructure

- تقدر convert videos & audios لاي end-user device يناسب الـ format (مثل: iPhone, Android, Web)
- الـ Auto Scales حسب حجم الملفات.
- الـ Pricing → بتدفع بس على الـ transcoding اللي استعملته.

Amazon AppSync

هي Service بتخلي الـ clients (mobile أو web) يقدروا يعملوا:

- Query
- Mutation (change)
- Subscription
 - بتشتغل باستخدام GraphQL
 - بتسخدمها في data sync ما بين الـ backend وـ apps
- Use cases: real-time chat apps, collaboration apps, dashboards.

Amazon WorkSpaces

- عبارة عن Virtual Desktops (DaaS) fully managed
- الـ client بيقدر بـ access desktop environment securely من أي مكان.
- إنت كـ admin بتحدد له:
 - الـ applications المتاحة.
 - الـ data اللي يقدر يشوفها.

- الميزة: بيشتغل كأنه desktop عادي، بس كلـه hosted على AWS (أكتر secure data leak)، مفيش hosted على AWS.

Amazon WorkDocs

- It is a fully managed, secure enterprise storage and collaboration service.
- Can integrate with existing corporate directories.
- Users can preview and comment on different supported file types.
- Deleted folders and files can be recovered for up to 30 days after deletion.
- Each user account comes with 1TB storage; administrators can add or limit storage per user.

Amazon X-Ray

بساعدك تعرف تعمل Requests tracking لي وتعرف الـ latency في App بناتوك موجوده فـين وهي مناسبـه ولا لا ، ومنها هتحسن بنـاءـك performance بنـاءـك App بنـاءـك.

خدمة بتساعدك تعمل trace و debug للـ applications لـ distributed apps (microservices) بنـاءـك، خصوصـاـ الـ applications.

- بتديك Service Map يوضحـك كلـ الـ requests ماشيـة إزاـي بينـ الـ services.
- تقدر تـ identify errors & bugs بـ سهـولة.
- كمان ممكن تبني عليها custom analysis & visualization apps.

Security

- الـ X-Ray دايـماـ بيـعمل encryption لـ traces والـ data at rest والـ KMS keys باستخدام.

Integration

- بيـقدر يـتعامل مع أغلـبـ الـ AWS services زيـ:
- EC2
 - ECS
 - Lambda
 - Elastic Beanstalk
 - API Gateway
 - ELB

بس خلي بالـك: محتاج X-Ray Daemon يـشتـغل عـلـشـان يـقدر يـجمـع الـ traces منـ الخـدـمـاتـ ديـ.

Machine Learning Services

- الـ Amazon Recognition services هيـ بـتـعـملـ image و video لـ Recognition analysis.
- الـ Image Analysis
- الـ Object Detection: بيـتـعـرفـ الحاجـهـ ديـ ايـهـ اليـ فيـ صـورـةـ.

- ال Facial Analysis: بتحل الصورة وتديك details اكتر.
 - ال Text Detection: بتقرأ ال text في الصورة.
 - ال Unsafe Content: بتعرف المحتوي الي غير لائق وتقدر تحذفه.
 - ال Amazon Polly: هي بتحول ال Text-to-Speech
 - ال Amazon Transcribe: هي بتحول ال Speech to text
 - ال Use Cases
 - .Customer Conversation Insights
 - .Clinical Documentation
 - ال Amazon Translate: هي Delivers fast Neural machine translation service بت language translation بت customize ليها وشغلها الأساسي بت .(conversational interface or chatbot)
 - ال Amazon Lex: دي service بتكلم معها (Amazon Comprehend
 - ال Amazon Comprehend: هي بتسعمل ال Natural language processing (NLP) عشان اعمل extract من insights
 - ال Document data او :Use Case
 - .Sentiment analysis
 - .Document Classification
 - .Language Detection
 - .Entity recognition
 - ال Amazon SageMaker: هي Data scientists and developers لي fully managed service train, and deploy
 - هي بتأخذ dataset ومنها ب train ال model ومنها هطور ال model دا.
 - ال Amazon Kendra: هي advanced machine learning algo بتسعمل ال document search service عشان ترجع معينه من ال Doc answer
 - ال Amazon Textract: هي بتساعد تعمل extract من صورة مكتوبه بي handwriting
 - ال Amazon Personalize: بيقدم recommendations system
-

AWS Market Place

ال Market Place هو Software third-place تقدر تستخدموا مع AWS Services بنايعت سواء تدور عليه او تشتريه او تعلمو Deploy

Labs & Tasks

مجمع لكل الحاجات الي ممكن تجرب فيها إللي أتعلمنته في PDF دا، ويستحسن خدهم بالترتيب الي مكتوب.

- Cloud Quest
 - DevOps Kitchen Workshop كل المعلومات هتلقيها في ال README)
 - AWS Project By Lama & Rizk
 - AWS Project By Momen Mohamed
-

Thanks To

حابب اشكر بشهادة محمد عراقي وبشهادة محمد بلينج علي تأسيس وفهم الكلاود بشكل جميل وممتع.

وحابب اشكر بشهادة عيسى ابو شريف علي مجهوده في كورس SAA ومعلومات القيمة والخبرة التي قدمها فيه.

وحابب اشكر بشهادة يوسف شوقي علي مجهوده في مراجعه الملخص او الكتيب جدا.

Reference

- All Fundamentals (شرح بشهادة محمد بلينج)
- AWS Cloud (شرح بشهادة محمد بلينج و بشهادة عيسى ابو شريف)
- AWS Documentation
- [Manara Academy](#)
- [Cloud Simplified By Rafik Soliman](#) (AWS CLF Prep)
- AWS Cloud Solution Architect Associate (شرح بشهادة عيسى ابو شريف)