Terms	Full Form	Definition	NOTE
		Topic 1) :-IAM and EC2	
Region		is cluster of data center	
IAM	Identity and Access Management		Global view ,Policies are written in json
MFA	Multifactor Authentication		
EC2	Elastic Compute Cloud	An **EC2** instance is nothing but a virtual server in **Amazon Web services** terminology	Application we deploy in EC2 called Instance
AMI	Amazon Machine Image	this is basically operating system launch on server	AMI is region locked and same ID cannot be used across region
Security group		is a set of firewall rules that control the traffic of instance	All inbound traffic is blocked by default ,All outbound traffic is authorized by default . Authorized IPV4 and IPV6. Inbound(from other to instance). Outbound(from insatnce to other)
SSH	Secure shell	It allows to control a remote machine all using command line	
Public IP		machine can be identified on the internet	if we start and stop ec2 instance public ip will change but private ip will remain same
Private IP		machine can only be identified on private network only	
Elastic IP		A fixed (static) IP address that you have allocated in Amazon EC2 and then attached to an instance.	
EC2 Instance Launch Types		EC2 On Demand	pay for what we use, highest cost, no long term commitment
		Reserved Instances	long workload(>1year-3 year),75% discount, ex- x4-large
		Convertible Reserved Instances	54% discount compared to On-Demand, Can change instance type
		Scheduled Reserved Instances	Use when you need (Day, Week, Month) (Eg.: Every Sat-Sun)
		Spot Instances	Short Workload,cheap,can loose instance ,90% discount,
			Instance lost withing 2 mins notification after spot price crosses bid amount
			Typically used for Batch Jobs, Big Data Analysis which are resilient to failures
		Dedicated Hosts	Book entire physical server , 3 years allocation,expensive,Visibility to underlying socket, processor cores, hardware, etc.
		Dedicated Instances	No other customer will share hardware
ENI	Elastic Network Interface		

Terms	Full Form	Definition	NOTE
Topics 2) : ELB AND ASG			
Vertical Scalability		Increase Instance Size	ex: t2.large ex:-database,rds,elasticache
Horizontal Scalability		IncreaseNumber of instance	common for web application, Auto scaling group
High availability		Run instance for same application across multi AZ	Goal : to survive data loss
Load Balancer		server that forward internet traffic to multiple server(ec2 instance)	provide static DNS name we can use in our application
ELB	Elastic Load Balancer	managed load balancer	Cost less but effort more to set up
Types Of Load Balancer	CLB	Classic Load Balancer (v1- old generation)-2009	HTTP and HTTPS(Layer 7),TCP(layer 4). Support only one SSL certificate. Support static DNS(URL)
	ALB	Application Load Balancer(v2- new generation)-2016	HTTP,HTTPS,WebSocket . Fit for microservice and container based application(ex docker and amazon ECS), mutiple target group. Support multiple SSL Certificate. Support static DNS
	NLB	Network Load Balancer(v2- new generation)-2017	TCP,TLS(secure TCP) and UDP. NLB have static ip while ALB and CLB does not have **static IP** but static hostname. NLB support elastic IP. Support multiple SSL Certificate. Support million of connection
Load Balancer Stickness		same client always redirected to same instance behind a load balancer	applicable for CLB and ALB . Enabled at target group level
SNI	Server Name Indication	**Solve problem of loading multiple certificate onto the server**	only worl ALB and NLB
Connection draining		stop sending new request to instance(ec2) which is unhealthy	default is 300 second
ASG	Autoscaling group	scale out(add ec2 instance) to match an increase load	Load balancer and ASG really works hands to hands:Means if asg add new instance then LB will automatically register to target group
		scale in(remove ec2 instance) to match a decrease load	
		IAM role attached to ASG will get assigned to EC2 instance	
ASG Scaling policies		Target Tracking Scaling	asg cpu to stay at 40 %.A target tracking scaling policy assumes that it should scale out your Auto Scaling group when the specified metric is above the target value
		Simple / step scaling	when cloudwatch alarm is triggered(cpu> 70 %) add 2 unit
		Scheduled action	increase min capacity to 10 at 5 pm on Friday
question			
1)		The application load balancer can redirect to different target groups based	Hostname, request path
2)		The Application Load Balancers target groups can be	ec2 instance, ip address, lambda function (LIE)

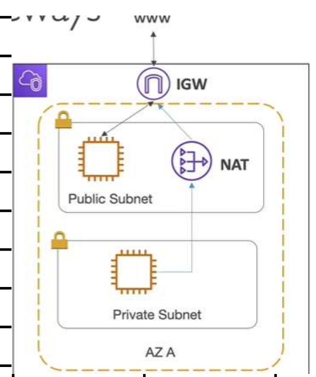
Terms	Full Form	Definition	NOTE
Topics 3) : EC2 Storage EBS and EFS			
EBS Volume	Elastic Block Store	Volume is a network drive , can attach to instances while they run	It’s a network drive
			its locked to an AZ, can attach to only one instance
			increase capacity of drive(GB,IOPS)
		EBS Volume Types(Only GP2 and IO can use as boot volume)	
		GP2(SSD)	general purpose(cheap),low latency,1 Gib-16 Tib, Max IOPS is 16000, System boot volumes, IO incrase if disk size increase
		IOI(SSD)	expensive,critical business application, large d/b workload-mongo db ,cassandra,etc,4 Gib-16Tib,min-100 and max 64000(nito instance) else 32000(other instance)
		STI(HDD)	throughput at low price, big data, apache kafka, cannot be a boot volume,500Gib-16 Tib, max iops is 500
		SCI(HDD)	lowest cost, infequently accessed data, 500 Gib to 16 Tib,cannot be a boot volume , max IOPS is 250
			IOPS :- Input/output operation per second
Instance Store		Physical Disk attached to physical server	Very high IOPS, cannot increase in size, risk of data loss if h/w fails
EFS	Elastic File System	EFS work with EC2 instance in multi A-Z	expensive, pay per use
			use case: content management,seb serving,data sharing, protocol uses: NFSv4.1,linux based API Compatible(not windows),Encryption at rest using KMS
question:			
1)		EBS Volumes are created for a specific AZ. It is possible to migrate them between different AZ through backup and restore	
2)		EFS is a network file system (NFS) and allows to mount the same file system on EC2 instances that are in different AZ	
3)		Instance Store provide the best disk performance	
4)		You are running a high-performance database that requires an IOPS of 210,000 for its underlying filesystem. What do you recommend?	Instance store

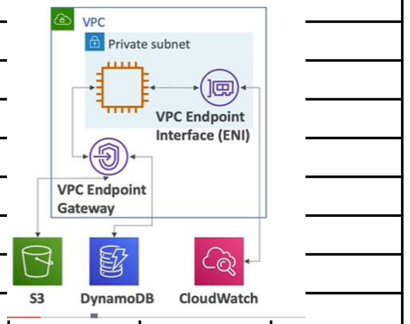
Terms	**Full Form**	**Definition**	**NOTE**
Topics 4) : RDS,Aurora, Elasticache			
RDS	Relational Database Service	allow to create d/b in the cloud that are managed by AWS ex:Aurora,mysql	
		Adavantages of RDS(managesd service)	Read replica, multi AZ setup,scaling capability(vertical and horizontal),daily full backup, DB snapshot
		Read Replica of RDS	Read replica use for select not (update,insert,delete) , can setup for multi AZ
		RDS Security	
		Encryption at rest	is done only when first create DB instance
			unencrypted db-->snapshot--> copy snapshot as encrypted--> create db from snapshot
		User responsibility	check port/IP/Security group inbound rules in DB
			in db user creation and permission manage through IAM
			Ensure paramter group or db is configure to allow SSL connection
		AWS responsibility	no ssh access, no manual db patching, no manual od patching
Aurora DB		is a proprietary technology from AWS (not opened source)	
		Advantages	postgres and mysql both supported as aurora db
			aurora claims 5x and 3x performance of mysql and postgres on RDS resp
			aurora storage automatically grows in increment of 10 gb upto 64 tb
			aurora can have 15 replica (faster)while mysql is 5
			support cross region replication
		Aurora security	similar to RDS, encryption at rest using KMS
			automated backup,snapshot and replica are also encrypted
			encrytion in flight using SSL
			possibility to authenticate using IAM token
			responsible for protecting the instance with security group and can't ssh
		Aurora serverless	automated database instantiation and autoscaling based on actual usage
Elasticache		cache are in memory db with high performance	EX: rsdis and memcached
		points	write scalinng using sharding
			read scaling using read replica
		cache eviction and TTL(Time To Live)	delete item explicitly
			item is evicted because memory is full
			set an item TTL
Question			
1		Read Replicas add new endpoints for databases to read	
2		oracle and mysql use TDE(Transport data encryption) on top of KMS	
3		oracle does not support IAM Authentication	

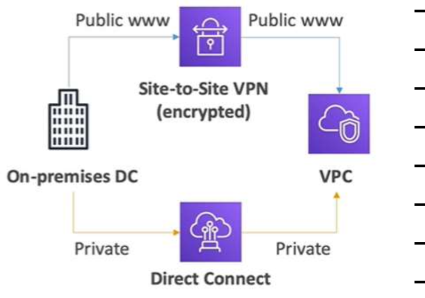
4		Global Aurora allow to have cross region replication	
5		IAM is leveraged to obtain the RDS service token	
6		Lazy Loading would only cache data that is actively requested from the database	
7		Multi AZ keeps the same connection string regardless of which database is up. Read Replicas imply we need to reference them individually in our application as each read replica will have its own DNS name	

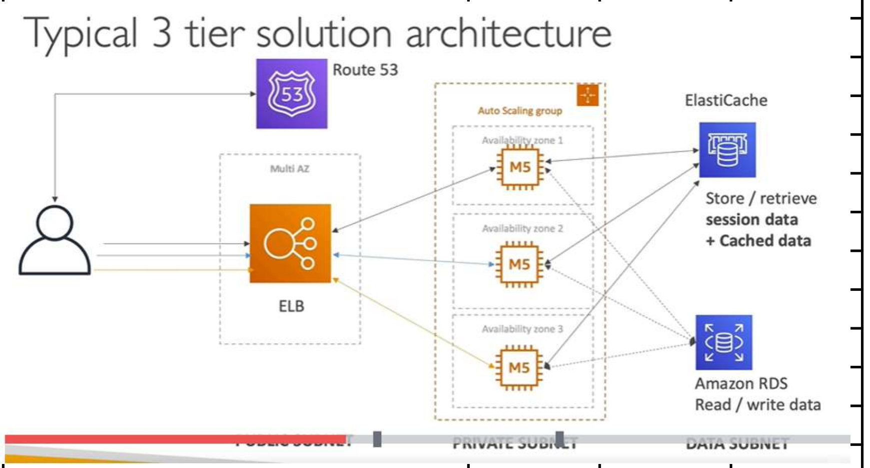
Terms	Full Form	Definition	NOTE
Topics 5) : Route 53			
Route 53		managed DNS(Domain Name System)	DNS IS rule help client how to reach server through domain name. Route 53 will not send traffic to unhealth instance. Can have http, https,tcp health check
		In aws most common record are:	
		A	hostname to IPV4
		AAAA	hostname to IPV6
		CNAME	point hostname to hostname(only for non root domain)
		Alias	hostname to AWS resource
		Features:	health check,load balancing
		TTL IS mandatory for each DNS record	only when ttl is expired then only we get other instance to up for same DNS
			From DNS name we can configure public ip of ec2 in this way it is linked to EC2
		Routing Policy:	
		Simple routing policy	Need to redirect to single resource(from route 53 to ec2 , give public ip of ec2 to route 53)
			can't attach health check
			if multiple value returned a random value is choosen by client
		Weighted Routing Policy	control the % of request that go to specific endpoint
			helpful to split traffic between two region
			can be associated with health check
		Latency Routing policy	Redirect to the server that has least latency
			latency: total round trip time it takes for a data packet to travel.
		Failover Routing Policy	
		Routing policy geolocation	specify traffic will go this specific ip(default will define)
		Multi value routing policy	use when routing traffic to different resources
question			
1		DNS records have a TTL (Time to Live) in order for clients to know for how long to caches these values and not overload the DNS with DNS requests. TTL should be set to strike a balance between how long the value should be cached vs how much pressure should go on the DNS.	
2		Latency will evaluate the latency results and help your users get a DNS response that will minimize their latency (e.g. response time)	

Terms	Full Form	Definition	NOTE
Topics 6) : VPC Fundamentals			
VPC	Virtual Private Cloud	private network to deploy resources(regional resource)	one default VPC per AWS region
Subnet		allow to partition nwtwork inside VPC (AZ resource)	launching ec2 instance, tied to AZ
		public subnet	accessible from the internet
		private subnet	not accessible from the internet
IGW	Internet Gateway	help VPC instance connect with the internet	at VPC level, provide internet access
NAT	Network address translation	private subnet will access by NAT and it connect to IGW	
NACL	Netwok Access Control List	a firewall which control traffic from and to subnet	attached at subnet level
			can have allow and deny rules
			support allow and deny rules
		security group	operate at instance level
			support allow rules only
VPC Peering		connect two VPC privately using AWS network with non overlapping ip ranges	
VPC Endpoint		provide private access to aws service(privately) within VPC	ex:) s3 and dynamo db
			used within VPC
On premises DC		is a group of server that privately own and control	
		lamp stack on ec2: linux -os for ec2 instance, apache : web server that run on linux(ec2), mysql: database on RDS,php :---application logic(runing on ec2)	
			three tier architecture









Terms	Full Form	Definition	NOTE
Topics 7) : Amazon S3			
S3	Simple Storage Service	object storage service	infinitely scaling storage and global service
			Integrate to many aws service. Explicit DENY in an IAM policy will take precedence over a bucket policy permission
s3 bucket		allow people to store files in bucket	globally unique name (specific region)
			no uppercase and underscore, not an ip,start with lowercase letter or number
s3 object		object(file) have a key	max object size is 5tb(5000 gb)
			if uploading more than 5 gb must use multi upload
			version id (if versioning is enabled)
s3 versioning		version file in amazon s3	enabled at bucket level
			easy roll back to previous version
			any file that is not enabled to previous version will have version null
s3 encryption		**4 method of encrypting object in s3**	
		SSE S3(Server side encryption)	encrytion using key handled and managed by amazon s3
			object is encrypted server side
			AES-256 encrytion type and must set header
		SSE KMS	encryption using keys handled & managed by KMS
			KMS Advantages: user control + audit trail
			Object is encrypted server side. Control rotation policy for the encryptiomn key
			Must set header: “x-amz-server-side-encryption": ”aws:kms"
		SSE-C	server-side encryption using data keys fully managed by the customer outside of AWS
			Amazon S3 does not store the encryption key you provide
			HTTPS must be used
			Encryption key must provided in HTTP headers, for every HTTP request made
		client side encrytion	Clients must encrypt data themselves before sending to S3
			Clients must decrypt data themselves when retrieving from S3
			Customer fully manages the keys and encryption cycle
		Encryption in transit (SSL/TLS)	Amazon S3 exposes: • HTTP endpoint: non encrypted • HTTPS endpoint: encryption in flight
			HTTPS is mandatory for SSE-C
			Encryption in flight is also called SSL / TLS
		s3 security	
		user based	IAM policies - which API calls should be allowed for a specific user from IAM console
		resource based	Bucket Policies - bucket wide rules from the S3 console - allows cross account

		S3 bucket policy	if IAM policy to allow to access bucket but bucket policy doen not allow it then IAM user will not able to access bucket
		JSON based policies	Resources: buckets and objects • Actions: Set of API to Allow or Deny • Effect: Allow / Deny • Principal: The account or user to apply the policy to
		S3 bucket for policy	• Grant public access to the bucket • Force objects to be encrypted at upload • Grant access to another account (Cross Account)
		S3 website	S3 can host static websites and have them accessible on the www
			website URL: <bucket-name>.s3-website-<AWS-region>.amazonaws.com
CORS	Cross origin resource sharing	Web Browser based mechanism to allow requests to other origins while visiting the main origin	Same origin: http://example.com/app1 & http://example.com/app2
			Different origins: http://www.example.com & http://other.example.com
			The requests won't be fulfilled unless the other origin allows for the requests, using CORS Headers (ex: Access-Control-Allow-Origin)
		s3 consistency model	
		Eventual Consistency for DELETES and PUTS of existing objects	If we read an object after updating, we might get the older version
			If we delete an object, we might still be able to retrieve it for a short time
		Read after write consistency for PUTS of new objects	As soon as a new object is written, we can retrieve it
			This is true, except if we did a GET before to see if the object existed

Terms	Full Form	Definition	NOTE
Topics 8) : aws cli,sdk,iam roles, policies			
CLI	Command line interface	how to interact with AWS from cli	never share access key and secret key
			always use IAM roles never put credential on EC2 machine
			IAM role can be attached to many ec2 instance but ec2 instance can be attach to only one IAM at a time
			Iam role use to give permission to EC2 instance so that they can make call
			Behind the scene when we attached role to ec2 instance it get access key id and secret
EC2 Instance			It allows AWS EC2 instances to ”learn about themselves” without using an IAM Role for that purpose.
			The URL is http://169.254.169.254/latest/meta-data
aws cli profiles		to configure multiple aws account from cli we will use profile	
MFA with CLI		To use MFA with the CLI, you must create a temporary session	To do so, you must run the STS GetSessionToken API call : duration 3600
SDK	Software Development Kit		if you want to perform actions on AWS directly from your applications code ? (without using the CLI)
		• Official SDKs are…	java, .net,node.js,php,python,go,ruby
			aws cli uses python SDK
			aws sdk use in lambda function
		good to know	if you don’t specify or configure a default region, then us-east-1 will be chosen by default
exponential backoff		is a standard error-handling strategy for network applications. In this approach, a client periodically retries a failed request with increasing delays between requests.	If you get ThrottlingException intermittently, use exponential backoff
Throttling limit		Throttling is the process of limiting the number of requests	
		AWS Limit(Quotas)	
		API Rate Limit	DescribeInstances API for EC2 has a limit of 100 calls per seconds
			GetObject on S3 has a limit of 5500 GET per second per prefix
			For Intermittent Errors: implement Exponential Backoff •
			For Consistent Errors: request an API throttling limit increase
		The CLI will look for credentials in this order	1. Command line options – --region, --output, and --profile
			2. Environment variables – AWS_ACCESS_KEY_ID,AWS_SECRET_ACCESS_KEY, and AWS_SESSION_TOKEN
			3. CLI credentials file –aws configure ~/.aws/credentials on Linux / Mac & C:\Users\user\.aws\credentials on Windows
			4. CLI configuration file – aws configure ~/.aws/config on Linux / macOS & C:\Users\USERNAME\.aws\config on Windows

			5. Container credentials – for ECS tasks
			6. Instance profile credentials – for EC2 Instance Profiles
		The Java SDK (example) will look for credentials in this order	1. Environment variables – AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY
			2. Java system properties – aws.accessKeyId and aws.secretKey
			3. The default credential profiles file – ex at: ~/.aws/credentials, shared by many SDK
			4. Amazon ECS container credentials – for ECS containers
			5. Instance profile credentials– used on EC2 instances
		Signing AWS API requests	You should sign an AWS HTTP request using Signature v4 (SigV4)
			some requests to Amazon S3 don’t need to be signed
			If you use the SDK or CLI, the HTTP requests are signed for you
question			
1)premise server		An on-**premise server** is a physical, on-site **server** that a company must manage and maintain individually.	you can't attach EC2 IAM roles to on premise servers
2		When I run the CLI on my EC2 Instances, the CLI uses the _______ service to get _____ credentials	meta-data , temporary
3		you can retrieve the role name attached to your EC2 instance using the metadata service but not the policy itself	

Terms	Full Form	Definition	NOTE
Topics 9) : Advanced s3 and athena			
s3 MFA delete			1)only can be done through CLI 2) Only the bucket owner (root account) can enable/disable MFA-Delete 3) MFA-Delete currently can only be enabled using the CLI 4) if we enabled mfa delete from cli we can't delete file version under s3 5) any delete operation is not replicated
		to use MFA delete	enable versioning on S3 bucket
		you will need MFA	permanently delete an object version • suspend versioning on the bucket
		wont need MFA	enabling versioning • listing deleted versions
		default encryption	Bucket Policies are evaluated before "default encryption"
		s3 access logs	always separate application bucket and logging bucket
S3 Replication		Must enable versioning in source and destination	
CRR	Cross Region Replication	Copying is asynchronous,Buckets can be in different accounts	CRR - Use cases: compliance, lower latency access, replication across accounts
SRR	Same Region Replication	Must give proper IAM permissions to S3	SRR – Use cases: log aggregation, live replication between production and test accounts
		S3 Replication – Notes	After activating, only new objects are replicated (not retroactive)
		For DELETE operations:	If you delete without a version ID, it adds a delete marker, not replicated
			If you delete with a version ID, it deletes in the source, not replicated
		There is no "chaining" of replication	If bucket 1 has replication into bucket 2, which has replication into bucket 3
			Then objects created in bucket 1 are not replicated to bucket 3
		S3 presigned url	valid for 3600 second
		s3 storage class	
		1)s3 standard General purpose	Frequently accessed data. High durability (99.999999999%) of objects across multiple AZ
			If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
			99.99% Availability over a given year
			•Use Cases: Big Data analytics, mobile & gaming applications, content distribution…
		2) S3 Standard – Infrequent Access (IA)	Suitable for data that is less frequently accessed, but requires rapid access when needed
			High durability (99.999999999%) of objects across multiple AZs
			Use Cases: As a data store for disaster recovery, backups…
			Low cost compared to Amazon S3 Standard
		3) S3 One Zone - Infrequent Access (IA)	Same as IA but data is stored in a single AZ
			High durability (99.999999999%) of objects in a single AZ; data lost when AZ is destroyed
			Low latency and high throughput performance
			Supports SSL for data at transit and encryption at rest
			Low cost compared to IA (by 20%)
			Use Cases: Storing secondary backup copies of on-premise data, or storing data you can recreate
		4) S3 Intelligent Tiering	Small monthly monitoring and auto-tiering fee

		4) S3 Intelligent Tiering	Automatically moves objects between two access tiers based on changing access patterns
		5) Amazon Glacier	Data is retained for the longer term (10s of years)
			Each item in Glacier is called “Archive” (up to 40TB)
			Archives are stored in ”Vaults”
		Amazon Glacier – 3 retrieval options:	Expedited (1 to 5 minutes) • Standard (3 to 5 hours) • Bulk (5 to 12 hours) • Minimum storage duration of 90 days
		6)Amazon Glacier Deep Archive	for long term storage – cheaper:
			Standard (12 hours) • Bulk (48 hours) • Minimum storage duration of 180 days
		s3 moving between storage classes	
		infrequently accessed data	moved them to standard IA
		archive object don’t need real time	glacier or deep archive
		S3 – Baseline Performance	Your application can achieve at least 3,500 PUT/COPY/POST/DELETE and 5,500 GET/HEAD requests per second per prefix in a bucket
		S3 – KMS Limitation	When you upload, it calls the GenerateDataKey KMS API
			When you download, it calls the Decrypt KMS API
			As of today, you cannot request a quota increase for KMS
		S3 Performance	
		• Multi-Part upload:	recommended for files > 100MB, must use for files > 5GB
			• Can help parallelize uploads (speed up transfers)
		S3 Event Notifications	• S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication…
			If you want to ensure that an event notification is sent for every successful write, you can enable versioning on your bucket.
AWS Athena		Serverless service to perform analytics directly against S3 files	Analyze data directly on S3 => use Athena
		glacier vault lock	object cant be deleted
questions			
1		MFA Delete forces users to use MFA tokens before deleting objects. It's an extra level of security to prevent accidental deletes	
2		S3 Access Logs log all the requests made to buckets, and Athena can then be used to run serverless analytics on top of the logs files	
3		S3 CRR is used to replicate data from an S3 bucket to another one in a different region	
4		Pre-Signed URL are temporary and grant time-limited access to some actions in your S3 bucket.	

5		When a file is over 100 MB, Multi Part upload is recommended as it will upload many parts in parallel, maximizing the throughput of your bandwidth and also allowing for a smaller part to retry in case that part fails.	

Terms	**Full Form**	**Definition**	**NOTE**
Topics 10) : cloufront			
Cloudfront		Content Delivery network(CDN), is for caching globally	Improves read performance, content is cached at the edge
			Can expose external HTTPS and can talk to internal HTTPS backends
			DDoS protection, integration with Shield, AWS Web Application Firewall
		cloudfront origins	
		S3 bucket	For distributing files and caching them at the edge
			Enhanced security with CloudFront Origin Access Identity (OAI)
			CloudFront can be used as an ingress (to upload files to S3)
		Custom Origin (HTTP)	Application Load Balancer • EC2 instance • S3 website • Any HTTP backend you want
		CloudFront:	Global Edge network
			Files are cached for a TTL (maybe a day)
			Great for static content that must be available everywhere
		S3 Cross Region Replication	Must be setup for each region you want replication to happen
			Files are updated in near real-time. Read Only
			Great for dynamic content that needs to be available at low-latency in few regions
OAI	origin access Identity		is used for sharing private content via CloudFront. The OAI is a virtual user **identity** that will be used to give your CF distribution permission to fetch a private object from your **origin** server (e.g. S3 bucket).
			Once cloudfront will create(distribution) then OAI (will create automatically)
CloudFront Caching		Cache based on	Headers, Session Cookies,Query String Parameters
			Control the TTL (0 seconds to 1 year), can be set by the origin using the Cache- Control header, Expires header
			You can invalidate part of the cache using the CreateInvalidation API
		Good to know	Even though we will update file in s3 from cloudfront we will get same due to ttl time
		solution	Now after invalidation anything updated in s3 bucket will update here also
		security	For security we use OAI(origin access identity) and this is used to access to s3 bucket
			S3 bucket “websites” don’t support HTTPS
		CloudFront Signed URL	To Restrict Viewer Access, we can create a CloudFront Signed URL / Cookie
		Signed URL	access to individual files (one signed URL per file)
		Signed Cookies	access to multiple files (one signed cookie for many files)
		CloudFront Signed URL	commonly used to distribute paid content through dynamic CloudFront Signed URL generation.
		S3 CRR (cross region replication)	allows you to replicate the data from one bucket in a region to another bucket in another region
Geo Restriction			. With **Geo Restriction** you can choose the countries where you want Amazon **CloudFront** to deliver your content.

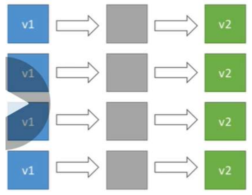
Terms	Full Form	Definition	NOTE
Topics 11) : ecs,ecr,fargate-docker in aws			
Docker		software development platform to deploy apps	Docker images are stored in Docker Repositories
		Docker Containers Management	To manage containers, we need a container management platform
		Three choices:	ECS: Amazon’s own platform
			Fargate: Amazon’s own Serverless platform
EKS	Elastic Kubernetes Service		EKS: Amazon’s managed Kubernetes (open source)
ECS	Elastic Container Service	used to manage docker container	Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.
		ECS Clusters Overview	ECS Clusters are logical grouping of EC2 instances
			EC2 instances run the ECS agent (Docker container)
			The ECS agents registers the instance to the ECS cluster
			The EC2 instances run a special AMI, made specifically for ECS
		ECS hands on	1)First create cluster : ec2 will register in ecs cluster
			2)Then ecs task definition: creating task to run in container
			3)Ecs service : how many task will run (also can use load balancer and auto scaling)
			Configure security group for ec2 and Public ip : 8080
		note	1) While creating ECS due to ASG , ec2 instance will create automatically. AMI id will create by ECS 2) ecs agent will register ec2 to ecs cluster due to autoscaling and task definition : configure container info
		ECS Task Definitions	Tasks definitions are metadata in JSON form to tell ECS how to run a Docker Container
		It contains crucial information around:	Image Name • Port Binding for Container and Host • Memory and CPU required • Environment variables • Networking information • IAM Role • Logging configuration (ex CloudWatch)
		ECS Service	place a task (container) in EC2 and tell how many task should run
ECR	Elastic Container Registry	private Docker image repository	Access is controlled through IAM (permission errors => policy)
		AWS CLI v1 login command	$(aws ecr get-login --no-include-email --region eu-west-1)
		AWS CLI v2 login command	aws ecr get-login-password --region eu-west-1 | docker login --username AWS -- password-stdin 1234567890.dkr.ecr.eu-west-1.amazonaws.com
Fargate		it’s all Serverless	We don’t provision EC2 instances
			We just create task definitions, and AWS will run our containers for us
		NOTE	There is no ec2 container and autoscaling group in fargate but behind the scene aws provide docker container for us serverless manner

		ECS IAM Roles Deep Dive	ecs agent connect with ecs service , cloudwatch logs and ecr service through ec2 instance profile
		EC2 Instance Profile:	Used by the ECS agent
			Makes API calls to ECS service
			Send container logs to CloudWatch Logs
			Pull Docker image from ECR
		ECS Task Role:	Allow each task to have a specific role
			Use different roles for the different ECS Services you run
			Task Role is defined in the task definition
		ECS Tasks Placement	when a service scales in, ECS needs to determine which task to terminate.
			Note: this is only for ECS with EC2, not for Fargate
		ECS Task Placement Strategies	
		1)Binpack	Place tasks based on the least available amount of CPU or memory
			This minimizes the number of instances in use (cost savings)
		2) Random	Place the task randomly
		3) Spread	Place the task evenly based on the specified value
			Example: instanceId, attribute:ecs.availability-zone
		ECS Task Placement Constraint	
		distinctInstance	place each task on a different container instance
		memberOf	places task on instances that satisfy an expression. Uses the Cluster Query Language (advanced)
		ECS – Service Auto Scaling	
			CPU and RAM is tracked in CloudWatch at the ECS service level
		Step Scaling	scale based on CloudWatch alarms
		Scheduled Scaling	based on predictable changes
capacity provider			Capacity provider : give 70 % of cpu if more task create the due to capacity provider more ec2 instance will create
		ECS Other	• ECS does integrate with CloudWatch Logs:
			• You need to setup logging at the task definition level
			• Each container will have a different log stream •
			The EC2 Instance Profile needs to have the correct IAM permissions
			• Use IAM Task Roles for your tasks
			• Task Placement Strategies: binpack, random, spread
			• Service Auto Scaling with target tracking, step scaling, or scheduled
			• Cluster Auto Scaling through Capacity Providers
question			

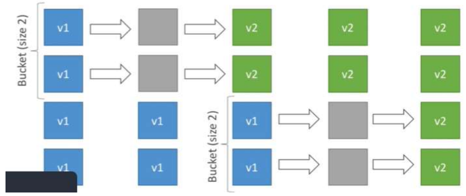
1)		**Which ECS config must you enable in `/etc/ecs/ecs.config` to allow your ECS tasks to endorse IAM roles?**	ECS_ENABLE_TASK_IAM_ROLE
2)			Any permissions issues against ECR is most likely due to IAM policies
3)			To enable random host port, set host port = 0 (or empty), which allows multiple containers of the same type to launch on the same instance
4)		MOST COST EFFICEINT	binpack

Terms	Full Form	Definition	NOTE
Topics 12) : aws elastic beanstalk			
Elastic Beanstalk		AWS Elastic Beanstalk is an easy-to-**use** service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache	
		Elastic Beanstalk Deployment	
		1) All at once	Fastest deployment
			Application has downtime
			Great for quick iterations in development environment
			No additional cost
		2)Rolling	Application is running below capacity
			Can set the bucket size
			Application is running both versions simultaneously
			No additional cost
			Long deployment
		3) Rolling with additional batches	Application is running at capacity
			Can set the bucket size
			Application is running both versions simultaneously
			Small additional cost
			Additional batch is removed at the end of the deployment
			Longer deployment
			Good for prod
		4) Immutable	Zero downtime
			New Code is deployed to new instances on a temporary ASG
			High cost, double capacity
			Longest deployment
			Quick rollback in case of failures (just terminate new ASG)
			Great for prod
		Blue / Green	Not a “direct feature” of Elastic Beanstalk
			Zero downtime and release facility
			Create a new “stage” environment and deploy v2 there
			The new environment (green) can be validated independently and roll back if issues
			Route 53 can be setup using weighted policies to redirect a little bit of traffic to the stage environment

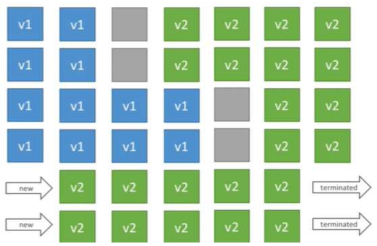
Diagram



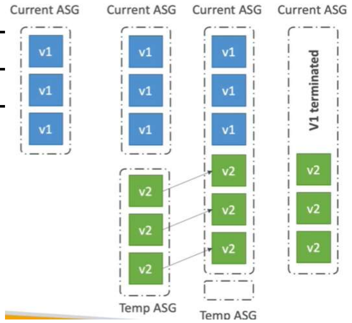
1



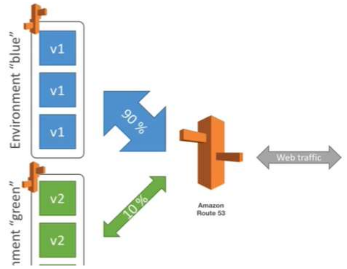
2



3



4



5

			Using Beanstalk, “swap URLs” when done with the environment test
		Elastic Beanstalk Deployment Process	Elastic Beanstalk will deploy the zip on each EC2 instance, resolve dependencies and start the application
		Beanstalk Lifecycle Policy	Elastic Beanstalk can store at most 1000 application versions
		To phase out old application versions, use a lifecycle policy	Based on time (old versions are removed) • Based on space (when you have too many versions)
		Elastic Beanstalk Extensions	A zip file containing our code must be deployed to Elastic Beanstalk
		Requirements:	• in the .ebextensions/ directory in the root of source code
			YAML / JSON format
			.config extensions (example: logging.config)
			Ability to add resources such as RDS, ElastiCache, DynamoDB, etc…
			Resources managed by .ebextensions get deleted if the environment goes away
		use case (good to know)	you can define CloudFormation resources in your .ebextensions to provision ElastiCache, an S3 bucket, anything you want
		Elastic Beanstalk Cloning	Clone an environment with the exact same configuration
			Useful for deploying a “test” version of your application
			After cloning an environment, you can change settings
		Elastic Beanstalk Migration: Load Balancer	After creating an Elastic Beanstalk environment, you cannot change the Elastic Load Balancer type (only the configuration)
		Elastic Beanstalk – Single Docker	Run your application as a single docker container
			Beanstalk in Single Docker Container does not use ECS
		Elastic Beanstalk – Multi Docker Container	• Multi Docker helps run multiple containers per EC2 instance in EB
		This will create for you:	ECS Cluster
			EC2 instances, configured to use the ECS Cluster
			Load Balancer (in high availability mode)
			Task definitions and execution
			Requires a config Dockerrun.aws.json (v2) at the root of source code
			• Dockerrun.aws.json is used to generate the ECS task definition
		Beanstalk with HTTPS	• Idea: Load the SSL certificate onto the Load Balancer
			•.ebextensions/securelistener-alb.config •
			Can be done from the code: .ebextensions/securelistener-alb.config
			SSL Certificate can be provisioned using ACM (AWS Certificate Manager) or CLI
			• Must configure a security group rule to allow incoming port 443 (HTTPS port)

		Custom Image	is to tweak an existing Beanstalk Platform (Python, Node.js, Java…)
		Custom Platform	is to create an entirely new Beanstalk Platform
question			
1		**I would like to customize the runtime of Elastic Beanstalk and include some of my company wide security software. I should**	custom platform
		What service does Elastic Beanstalk use under the hood?	aws clouformation
		How can you remove older versions that are not used by Elastic Beanstalk so that new versions can be created for your applications?	lifecycle policy
		You have created a test environment in Elastic Beanstalk and as part of that environment, you have created an RDS database. How can you make sure the database can be explored after the environment is destroyed?	make a snapshot of db before it get deleted
		You can define periodic tasks in a file cron.yaml	

Terms	Full Form	Definition	NOTE
Topics 13) : cicd commit,pipeline,build,deployment			
AWS Code Commit		private Git repositories	Code only in AWS Cloud account
			Secure (encrypted, access control, etc…)
			Integrated with Jenkins / CodeBuild / other CI tools
		CodeCommit Security	Interactions are done using Git (standard)
		Authentication in Git:	SSH Keys: AWS Users can configure SSH keys in their IAM Console
			HTTPS: Done through the AWS CLI Authentication helper or Generating HTTPS credentials
			MFA (multi factor authentication) can be enabled for extra safety
		Authorization in Git:	IAM Policies manage user / roles rights to repositories
		Encryption:	Repositories are automatically encrypted at rest using KMS
			Encrypted in transit (can only use HTTPS or SSH – both secure)
		Cross Account access:	Do not share your SSH keys
			Do not share your AWS credentials
			Use IAM Role in your AWS Account and use AWS STS (with AssumeRole API)
		CodeCommit Notifications	trigger notifications in CodeCommit using AWS SNS (Simple Notification Service) or AWS Lambda or AWS CloudWatch Event Rules
		Use cases for notifications SNS / AWS Lambda notifications:	Deletion of branches
			Trigger for pushes that happens in master branch
			Trigger AWS Lambda function to perform codebase analysis (maybe credentials got committed in the code?)
		Use cases for CloudWatch Event Rules:	Trigger for pull request updates (created / updated / deleted / commented)
			CloudWatch Event Rules goes into an SNS topic
CodePipeline		Continuous delivery	
		Source:	GitHub / CodeCommit / Amazon S3
		Build:	CodeBuild / Jenkins / etc
		Load Testing:	3rd party tools
		Deploy:	AWS CodeDeploy / Beanstalk / CloudFormation / ECS…
		Made of stages:	Each stage can have sequential actions and / or parallel actions
			Stages examples: Build / Test / Deploy / Load Test / etc
			Manual approval can be defined at any stage
		CodePipeline Troubleshooting	AWS CloudTrail can be used to audit AWS API calls
			If Pipeline can't perform an action, make sure the "IAM Service Role" attached does have enough permissions (IAM Policy)
		NOTE	Code pipeline allow to link sources, build and deploy stages
			Note pipeline do a lot of thing : will talk to s3,code commit, beanstalk
			We need to create service role that have permission to do

			To run this create elastic beanstalk env to link with pipeline
CodeBuild		Fully managed build service	Alternative to other build tools such as Jenkins. SNS Notification
			Build instructions can be defined in code (buildspec.yml file)
			• Output logs to Amazon S3 & AWS CloudWatch Logs
			Use CloudWatch Events to detect failed builds and trigger notifications
		Secure	Integration with KMS for encryption of build artifacts, IAM for build permissions, and VPC for network security, CloudTrail for API calls logging
		CodeBuild BuildSpec	buildspec.yml file must be at the root of your code
		Define environment variables:	Plaintext variables
			Secure secrets: use SSM Parameter store
		Phases (specify commands to run):	Install: install dependencies you may need for your build
			Pre build: final commands to execute before build
			Build: actual build commands
			Post build: finishing touches (zip output for example)
		Artifacts	What to upload to S3 (encrypted with KMS)
		Cache	Files to cache (usually dependencies) to S3 for future build speedup
CodeBuild in VPC			By default, your CodeBuild containers are launched outside your VPC
			Therefore, by default it cannot access resources in a VPC
		You can specify a VPC configuration:	VPC ID • Subnet IDs • Security Group IDs
			Then your build can access resources in your VPC (RDS, ElastiCache, EC2, ALB..)
		Use cases:	integration tests, data query, internal load balancers
AWS CodeDeploy		deploy our application automatically to many EC2 instances • Th	These instances are not managed by Elastic Beanstalk
			several ways to handle deployments using open source tools (Ansible, Terraform, Chef, Puppet, etc…)
			We can use the managed Service AWS CodeDeploy
		AWS CodeDeploy AppSpec	
		Hooks:	set of instructions to do to deploy the new version (hooks can have timeouts).
		The order is:	• ApplicationStop • DownloadBundle • BeforeInstall • AfterInstall • ApplicationStart • ValidateService:
		good to know	Appspec.yml help to understand codedeploy how to deploy application in ec2
		CodeDeploy - roll backks	f a roll back happens, CodeDeploy redeploys the last known good revision as a new deployment.
CodeStar		is an integrated solution that regroups	GitHub, CodeCommit, CodeBuild, CodeDeploy, CloudFormation, CodePipeline, CloudWatch
			• Helps quickly create “CICD-ready” projects for EC2, Lambda, Beanstalk
Question			
1	CICD	Continous Integration and Continous Delivery	
2		You want to send email alerts anytime pull requests are open or comments are added to commits in CodeCommi	AWS Cloudwatch event
3		code commit doesn't support	http public access

4		CodeBuild containers are deleted at the end of their execution (success or failed). You can't SSH into them, even while they're running		
5		CodeBuild can run any commands, so you can use it to run commands including generating a static website and copy your static web files to Amazon S3.		
6		code deploy use for only ec2 instance		
7		Which hook step should be used in appspec.yml file to ensure the application is properly running after being deployed?	Validate service	
8		CodeDeploy is a managed service, there are no security groups to manage!		

Terms	Full Form	Definition	NOTE
Topics 14) : aws cloudformation			
CloudFormation		CloudFormation is a declarative way of outlining your AWS Infrastructure, for any resources (most of them are supported).	
		For example, within a CloudFormation template, you say:	I want a security group. I want an S3 bucket
			I want two EC2 machines using this security group
			I want a load balancer (ELB) in front of these machines
			Then CloudFormation creates those for you, in the right order, with the exact configuration that you specify
		How CloudFormation Works	Templates have to be uploaded in S3 and then referenced in CloudFormation
			To update a template, we can’t edit previous ones. We have to reupload a new version of the template to AWS
			Stacks are identified by a name
			Deleting a stack deletes every single artifact that was created by CloudFormation
		Deploying CloudFormation templates	
		Manual way:	• Editing templates in the CloudFormation Designer
		Automated way:	Using the AWS CLI (Command Line Interface) to deploy the templates
		good to know	even if we didn’t mention order in template cloud formation is intelligent what we need to execute first
resources		They represent the different AWS Components that will be created and configured	
		Resource types identifiers are of the form:	AWS::aws-product-name::data-type-name
parameters			Parameters are a way to provide inputs to your AWS CloudFormation template
		How to Reference a Parameter	The Fn::Ref function can be leveraged to reference parameters
			The shorthand for this in YAML is !Ref
Mappings		Mappings are fixed variables within your CloudFormation Template	Mappings are great when you know in advance all the values
		use Fn::FindInMap	to return a named value from a specific key
		good to know	You can’t delete a CloudFormation Stack if its outputs are being referenced by another CloudFormation stack
			can’t delete the underlying stack until all the references are deleted
		CloudFormation Must Know Intrinsic Functions	
		1) The Fn::Ref function can be leveraged to reference	
		Parameters	returns the value of the parameter
		Resources	returns the physical ID of the underlying resource (ex: EC2 ID)
		2) Fn::GetAtt	Attributes are attached to any resources you create
		example	the AZ of an EC2 machine!
			GetAtt Ec2Instance.AZ

		3) Fn::FindInMap	to return a named value from a specific key
		4) Fn::ImportValue	Import values that are exported in other templates
		5)Fn::Join	• Join values with a delimiter. This creates “a:b:c”
		6) Fn::Sub, or !Sub	is used to substitute variables from a text
			• String must contain ${VariableName} and will substitute them
		CloudFormation Rollbacks	
		Stack Creation Fails:	Default: everything rolls back (gets deleted). We can look at the log
			Option to disable rollback and troubleshoot what happened
		Stack Update Fails:	The stack automatically rolls back to the previous known working state
			• Ability to see in the log what happened and error messages
		Nested stacks	They allow you to isolate repeated patterns in separate stacks and call them from other stacks
		Example:	• Load Balancer configuration that is re-used
			Security Group that is re-used
		CloudFormation - StackSets	Create, update, or delete stacks across multiple accounts and regions with a single operation
			Administrator account to create StackSets
question			
		The `!Ref` function can be used to reference	Parameters and resources
		exported output name must be unique within the region	

Terms	**Full Form**	**Definition**	**NOTE**
		Topics 15) : cloudwatch,xray,cloudtrail	
AWS CloudWatch:		Metrics	Collect and track key metrics
		Logs	Collect, monitor, analyze and store log files
		Events	Send notifications when certain events happen in your AWS
		Alarms	React in real-time to metrics / events
AWS X-Ray			Troubleshooting application performance and errors
			Distributed tracing of microservices
AWS CloudTrail:			Internal monitoring of API calls being made
		AWS CloudWatch Metrics	CloudWatch provides metrics for every services in AWS
			Metric is a variable to monitor (CPUUtilization, NetworkIn…)
			Metrics belong to namespaces
			Dimension is an attribute of a metric (instance id, environment, etc…).
			Can create CloudWatch dashboards of metrics
		AWS CloudWatch Custom Metrics	Use API call PutMetricData
			Use exponential back off in case of throttle errors
		good to know	if we enabled detailed monitoring we will get in every one minute
		AWS CloudWatch Alarms	The new **CloudWatch Alarms** feature allows you to watch **CloudWatch** metrics and to receive notifications when the metrics fall outside of the levels (high or low thresholds) that you configure
			Alarms are used to trigger notifications for any metric
			Alarms can go to Auto Scaling, EC2 Actions, SNS notifications
			Various options (sampling, %, max, min, etc…)
		Alarm States	• OK • INSUFFICIENT_DATA • ALARM
		Period	High resolution custom metrics: can only choose 10 sec or 30 sec
		AWS CloudWatch Logs	Applications can send logs to CloudWatch using the SDK
		CloudWatch can collect log from:	Elastic Beanstalk: collection of logs from application
			ECS: collection from containers
			AWS Lambda: collection from function logs
			VPC Flow Logs: VPC specific logs
			API Gateway
			CloudTrail based on filter
			CloudWatch log agents: for example on EC2 machines
			Route53: Log DNS queries
		CloudWatch Logs can go to	Batch exporter to S3 for archival
			Stream to ElasticSearch cluster for further analytics

		good to know	To send logs to cloudwatch IAM permission should be correct
			Security: encryption of logs using KMS at the Group Leve
		CloudWatch Logs for EC2	By default, no logs from your EC2 machine will go to CloudWatch
			You need to run a CloudWatch agent on EC2 to push the log files you want
			Make sure IAM permissions are correct
on-**premise server**			is a physical, on-site **server** that a company must manage and maintain individually.
		CloudWatch Unified Agent – Metrics	Collected directly on your Linux server / EC2 instance
		CPU	(active, guest, idle, system, user, steal)
		Disk metrics	(free, used, total), Disk IO (writes, reads, bytes, iops)
		RAM	(free, inactive, used, total, cached)
		Netstat	(number ofTCP and UDP connections, net packets, bytes)
		Processes	(total, dead, bloqued, idle, running, sleep)
		Swap Space	(free, used, used %)
		CloudWatch Logs Metric Filter	
		use For example	find a specific IP inside of a log
			count occurrences of “ERROR” in your logs
		good to know	We can create alarm on top of metric filter(filter data in logs)log coming from cloudwatch
			If some alarm raise it will send message to SNS
		AWS CloudWatch Events	
		Schedule	Cron jobs
		Triggers to	Lambda functions, SQS/SNS/Kinesis Messages
			CloudWatch Event creates a small JSON document to give information about the change
event bus		Event buses can be accessed by other AWS accounts	sending and receiving event between aws account
		Default event bus	generated by AWS services (CloudWatch Events)
		Partner event bus:	receive events from SaaS service or applications (Zendesk, DataDog, Segment, Auth0…)
		Custom Event buses	for your own applications. Minimum resolution 1 second
		Rules	how to process the events (similar to CloudWatch Events)
		EventBridge	can analyze the events in your bus and infer the schema
		Schema Registry	allows you to generate code for your application, that will know in advance how data is structured in the event bus
AWS X ray		Visual analysis of our applications	helps developers analyze and debug production, distributed applications, such as those built using a microservices architecture.
		good to know	Error is coming from dynamo db table that we can visualise it using x ray
		AWS X -Ray advantages	Troubleshooting performance (bottlenecks)
			Understand dependencies in a microservice architecture

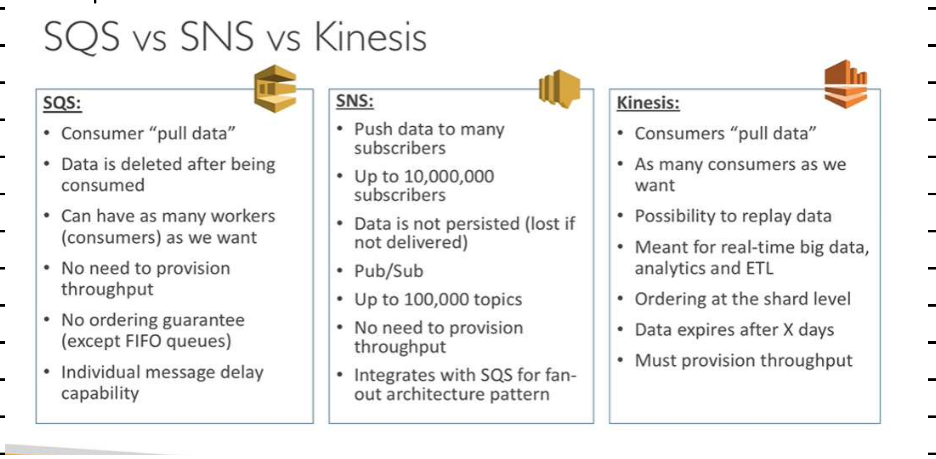
			Find errors and exceptions
		X -Ray compatibility	AWS Lambda • Elastic Beanstalk • ECS • ELB • API Gateway • EC2 Instances or any application server (even on premise)
		X-Ray Security	IAM for authorization
			KMS for encryption at rest
		How to enable it	Install the X-Ray daemon or enable X-Ray AWS Integration
		AWS X -Ray Troubleshooting	
		If X -Ray is not working on EC2	Ensure the EC2 IAM Role has the proper permissions
			Ensure the EC2 instance is running the X-Ray Daemon
		To enable on AWS Lambda	Ensure it has an IAM execution role with proper policy (AWSX-RayWriteOnlyAccess)
			Ensure that X-Ray is imported in the code
		X-Ray Concepts	
		Segments:	each application / service will send them
		Subsegments	if you need more details in your segment
		Trace	segments collected together to form an end-to-end trace
		Sampling	decrease the amount of requests sent to X-Ray, reduce cost
		Annotations	Key Value pairs used to index traces and use with filters
		Metadata	Key Value pairs, not indexed, not used for searching
		good to know	• The X-Ray daemon / agent has a config to send traces cross account:
X ray sampling rules			With sampling rules, you control the amount of data that you record
			By default, the X-Ray SDK records the first request each second, and five percent of any additional requests.
			One request per second is the reservoir, which ensures that at least one trace is recorded each second as long the service is serving requests
			• Five percent is the rate at which additional requests beyond the reservoir size are sampled.
		X-Ray Write APIs (used by the X-Ray daemon)	arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess
		PutTraceSegments:	Uploads segment documents to AWS X-Ray
		PutTelemetryRecords:	Used by the AWS X-Ray daemon to upload telemetry
			SegmentsReceivedCount, SegmentsRejectedCounts, BackendConnectionErrors…
		GetSamplingRules:	Retrieve all sampling rules (to know what/when to send)
			GetSamplingTargets & GetSamplingStatisticSummaries: advanced
			The X-Ray daemon needs to have an IAM policy authorizing the correct API calls to function correctly
		X-Ray Read APIs	
		GetServiceGraph:	main graph
		BatchGetTraces	Retrieves a list of traces specified by ID. Each trace is a collection of segment documents that originates from a single request

		GetTraceSummaries	Retrieves IDs and annotations for traces available for a specified time frame using an optional filter. To get the full traces, pass the trace IDs to BatchGetTraces
		GetTraceGraph:	Retrieves a service graph for one or more specific trace IDs
		X-Ray with Elastic Beanstalk	You can run the daemon by setting an option in the Elastic Beanstalk console or with a configuration file (in .ebextensions/xray-daemon.config)
AWS CloudTrail		we can track any api call that is done by anyone	• CloudTrail is enabled by default!
		Get an history of events / API calls made within your AWS Account by	• Console • SDK • CLI • AWS Services
			Can put logs from CloudTrail into CloudWatch Logs
		CloudTrail	Audit API calls made by users / services / AWS console
			Useful to detect unauthorized calls or root cause of changes
		CloudWatch	CloudWatch Metrics over time for monitoring
			CloudWatch Logs for storing application log
			CloudWatch Alarms to send notifications in case of unexpected metrics
		X-Ray	Automated Trace Analysis & Central Service Map Visualization
			Latency, Errors and Fault analysis
			Request tracking across distributed systems
question			
1			the alarm will remain in alarm state and never decrease number of instance in ASG
2		cloudwatch logs never expire by default	
3		CloudWatch Logs expiration policy should be defined at which level	Log Groups

Terms	Full Form	Definition	NOTE
Topics 16) : messaging sqs,sns,kinesis			
		synchronous communication	applicaion to application. directly talking with each other
		Asynchronous / Event based	application to queue to application
		NOTE	deploy multiple application they need to communicate
		synchronous application problem	sudden spikes of traffic
		in that case	it’s better to decouple(separate) your applications,
		using SQS	queue model
		using SNS:	pub/sub model
		using Kinesis:	real-time streaming model
Queue			producers send message to queue and consumers polls message from queue
SQS	Simple Queue Service	is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications	With **serverless** computing, your **application** still runs on servers, but all the server management is done by **AWS .sqs store message until microservice and serverless application process them**
		decoupled application	A decoupled application architecture allows each component to perform its tasks independently
		SQS Producing Message	Oldest offering (over 10 years old)
			Produced to SQS using the SDK (SendMessage API)
			The message is persisted in SQS until a consumer deletes it
			Message retention: default 4 days, up to 14 days
			Limitation of 256KB per message sent
			• Example: send an order to be processed • Order id • Customer id
			SQS standard: unlimited throughput
			Can have duplicate messages
		SQS – Consuming Messages	Consumers (running on EC2 instances, servers, or AWS Lambda)…
			Poll SQS for messages (receive up to 10 messages at a time)
			Process the messages (example: insert the message into an RDS database)
			Delete the messages using the DeleteMessage API
		SQS to decouple between application tiers	
		Requirement need to process video	To frontend request will come and it pass to sqs then backened will process and insert it into s3 (here application are independent with each other)
SQS - Security		Encryption	In-flight encryption using HTTPS API
			At-rest encryption using KMS keys
			Client-side encryption if the client wants to perform encryption/decryption itself
		Access Controls	IAM policies to regulate access to the SQS API
			(similar to S3 bucket policies)

		SQS Access Policies	Useful for cross-account access to SQS queues
			Useful for allowing other services (SNS, S3…) to write to an SQS queue
		good to know	purge will delete all message in queue
		SQS – Message Visibility Timeout	When message is processed by consumer other consumer cant process it during visibility time period if message is not deleted then after visibility time period consumer can process it
			• By default, the “message visibility timeout” is 30 seconds. minimum is 0 seconds. The maximum is 12 hours.
			if a time taking by consumer is more to consume message(not able to process within visibility timeout) then it will call api ChangeMessageVisibility to get more time
			If visibility timeout is too low (seconds), we may get duplicates
			we need to read message in visibility time period only
DLQ		SQS – Dead Letter Queue	If a consumer fails to process a message within the Visibility Timeout… the message goes back to the queue. Useful for debugging
		Make sure to process the messages in the DLQ before they expire	Good to set a retention of 14 days in the DLQ
		Delay Queue	Delay a message (consumers don’t see it immediately) up to 15 minutes
			Default is 0 seconds (message is available right away)
			• Can set a default at queue level
			Can override the default on send using the DelaySeconds parameter
		SQS - Long Polling	When a consumer requests messages from the queue, it can optionally “wait” for messages to arrive if there are none in the queue
			LongPolling decreases the number of API calls made to SQS while increasing the efficiency and latency of your application
			The wait time can be between 1 sec to 20 sec (20 sec preferable)
			Long polling can be enabled at the queue level or at the API level using WaitTimeSeconds
		SQS Extended Client	Message size limit is 256KB, how to send large messages, e.g. 1GB?
			SQS will tell to consumer hey go and retrieve bigger message from s3
		SQS – Must know API	
		CreateQueue (MessageRetentionPeriod), DeleteQueue	
		PurgeQueue	delete all the messages in queue
		SendMessage (DelaySeconds), ReceiveMessage, DeleteMessage	
		ReceiveMessageWaitTimeSeconds:	Long Polling
		ChangeMessageVisibility	change the message timeout
		Batch APIs for SendMessage, DeleteMessage, ChangeMessageVisibilit	helps decrease your costs

		SQS – FIFO Queue	removing duplicates , maintain ordering of message
		Limited throughput	300 msg/s without batching, 3000 msg/s with
		Deduplication of message	if we send same message twice to SQS within 5 minutes the second message will refuse.
		Content-based deduplication	will do a SHA-256 hash of the message body
		SQS FIFO – Message Grouping	• If you specify the same value of MessageGroupID in an SQS FIFO queue, you can only have one consumer, and all the messages are in order
SNS	Simple Notification Service	SNS uses the publish/sub model for push delivery of messages	• The "event producer" only sends message to one SNS topic
			Each subscriber to the topic will get all the messages
			Up to 10,000,000 subscriptions per topic
			100,000 topics limit
		Subscribers can be	• SQS • HTTP / HTTPS • Lambda • Emails • SMS messages • Mobile Notifications
		AWS SNS – How to publish	
		• Topic Publish (using the SDK)	Create a topic • Create a subscription (or many) • Publish to the topic
SNS – Security		Similar to SQS	
		SNS + SQS: Fan Out	• Push once in SNS, receive in all SQS queues that are subscribers
			Ability to add more SQS subscribers over time
			Make sure your SQS queue access policy allows for SNS to write
			SNS cannot send messages to SQS FIFO queues (AWS limitation)
		Application: S3 Events to multiple queues	If you want to send the same S3 event to many SQS queues, use fan-out
Kinesis		use for computation of real time data arrived through stream	Kinesis is a managed alternative to Apache Kafka
			Great for application logs, metrics, IoT, clickstreams
			Great for "real-time" big data
			Great for streaming processing frameworks (Spark, NiFi, etc…)
			Data is automatically replicated to 3 AZ
		Kinesis Streams	collect and store data
		Kinesis Analytics:	perform real-time analytics on streams using SQL/ process and deliver data
		Kinesis Firehose:	load streams into S3, Redshift, ElasticSearch/ analyze streaming data
		note	Data produces into kinesis stream and kinesis analytic want to processing the data(computation in real time data) and store data in kinesis firehouse(s3,database)
		Difference between sqs and kinesis	In sqs : once data is consumed data is gone but while in kinesis data is still there and it will expire after sometime
			In sqs no order but in kinesis record are going to be in order per shards
		Kinesis Streams Overview	Streams are divided in ordered Shards / Partitions
			Data retention is 1 day by default, can go up to 7 days
			Multiple applications can consume the same stream
			Once data is inserted in Kinesis, it can't be deleted (immutability)

		Resharding	througput increases more and we increase shard while someday throughput decreases then merging of shard occur(shard decrease)\
		Kinesis Streams Shards	One stream is made of many different shards
			1MB/s or 1000 messages/s at write PER SHARD
			2MB/s at read PER SHARD
			The number of shards can evolve over time (reshard / merge)
			• Records are ordered per shard
		good to know	Key is hashed to determine shard id
			hot partition:--message are going in same shard and it will be overloadeds
		AWS Kinesis API – Exceptions	
		ProvisionedThroughputExceeded Exceptions	Happens when sending more data (exceeding MB/s or TPS for any shard)
			Make sure you don't have a hot shard
		Solution	• Retries with backoff • Increase shards (scaling) • Ensure your partition key is a good one
KCL		Kinesis Client Library	use to consume message from kinesis efficiently
		Kinesis KCL in Depth	Kinesis Client Library (KCL) is Java library that helps read record from a Kinesis Streams with distributed applications sharing the read workload
			Rule: each shard is be read by only one KCL instance
			KCL can run on EC2, Elastic Beanstalk, on Premise Application
			Records are read in order at the shard level
		Kinesis Security	Control access / authorization using IAM policies
			Encryption in flight using HTTPS endpoints
			Encryption at rest using KMS
			Possibility to encrypt / decrypt data client side (harder)
			VPC Endpoints available for Kinesis to access within VPC
		SQS vs SNS vs Kinesis	

SQS vs SNS vs Kinesis

SQS:

- Consumer "pull data"
- Data is deleted after being consumed
- Can have as many workers (consumers) as we want
- No need to provision throughput
- No ordering guarantee (except FIFO queues)
- Individual message delay capability

SNS:

- Push data to many subscribers
- Up to 10,000,000 subscribers
- Data is not persisted (lost if not delivered)
- Pub/Sub
- Up to 100,000 topics
- No need to provision throughput
- Integrates with SQS for fan-out architecture pattern

Kinesis:

- Consumers "pull data"
- As many consumers as we want
- Possibility to replay data
- Meant for real-time big data, analytics and ETL
- Ordering at the shard level
- Data expires after X days
- Must provision throughput

Question			
1		SQS scale automatically	
2		In KCL, you can have a maximum of EC2 instances running in parallel equal to the number of shards in your Kinesis Stream.	
3		you can have as many consumers as GroupID for your FIFO queues	

Terms	Full Form	Definition	NOTE
			Topics 17): serverless lambda
serverless		Serverless does not mean there are no servers… it means you just don't manage / provision / see them	developers don't have to manage servers anymore…
			They just deploy code and functions
Cognito		where user identity is stored	User will get static content from s3 and from cognito identity will match then it invoked api gateway and from there lambda function and it will invoke dynamo db
		Serverless in AWS	• AWS Lambda • DynamoDB • AWS Cognito • AWS API Gateway • Amazon S3 • AWS SNS & SQS • AWS Kinesis Data Firehose • Aurora Serverless • Step Functions • Fargate
Lambda function		Lambda function cab be integrate with many service	is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you
		lambda function features	Virtual functions – no servers to manage
			Limited by time - short executions
			Run on-demand
			Scaling is automated. How much time lambda is running we need to pay for that time only
		Benefits of AWS Lambda	Pay per request and compute time
			1,000,000 AWS Lambda requests and 400,000 GBs of compute time
			• Integrated with many programming languages
			Easy monitoring through AWS CloudWatch
			Easy to get more resources per functions (up to 3GB of RAM!)
		important	Docker is not for AWS Lambda, it's for ECS / Fargate
		aws lambda integration main ones	api gateway, kinesis,dynamo db ,s3,cloudfront,sns,sqs,cognito,cloudwatch(cw) logs & cw events EventBridge
		Lambda – Synchronous Invocations	Results is returned right away means direct invocation you wait for result
			Error handling must happen client side (retries, exponential backoff, etc…)
			. With synchronous invocation, you wait for the function to process the event and return a response.
		Lambda - Synchronous Invocations - Services	
		User Invoked:	Elastic Load Balancing (Application Load Balancer) • Amazon API Gateway • Amazon CloudFront (Lambda@Edge) • Amazon S3 Batch
		Service Invoked:	Amazon Cognito • AWS Step Functions
		Lambda Asynchronous	With asynchronous invocation, Lambda queues the event for processing and returns a response immediately
		trigger	. A trigger is a Lambda resource or a resource in another service that you configure to invoke your function in response to lifecycle events
		Lambda Integration with ALB	Requirement : to invoke lambda function from ALB
			To expose a Lambda function as an HTTP(S) endpoint…
			• You can use the Application Load Balancer (or an API Gateway)
			The Lambda function must be registered in a target group
		note	First create lambda then ALB and linked lambda to ALB in target group. ALB can support multi header values (ALB setting)

		Lambda@Edge	is a feature of Amazon CloudFront that lets you run code closer to users of your application, which improves performance and reduces latency. ... With Lambda@Edge, you can enrich your web applications by making them globally distributed and improving their performance — all with zero server administration
		Lambda@Edge: Use Cases	Website Security and Privacy.Dynamic Web Application at the Edge.• User Authentication and Authorization
		Lambda – Asynchronous Invocations	Idempotent : in case of retries result will be same. use asynchronous if we don't need to wait for the result
			if the function is retried, you will see duplicate logs entries in CloudWatch Logs
			Can define a DLQ (dead-letter queue) – SNS or SQS – for failed processing (need correct IAM permissions)
			Asynchronous invocations allow you to speed up the processing if you don't need to wait for the result
		Lambda - Asynchronous Invocations - Services	s3,SNS,Amazon CloudWatch Events / EventBridge,CloudWatch Logs (log processing),CloudFormation,SES
			AWS CodeCommit (CodeCommit Trigger: new branch, new tag, new push)
			AWS CodePipeline (invoke a Lambda function during the pipeline, Lambda must callback)
		NOTE Requirement	to invoke a function asynchronous we will not wait for result
			If exception is there it will not show ,**will integrate with DLQ and send exception message there**
		scenario	Run synchronously execution get failed but if we run asynchronously we will not get to know about it
		S3 Events Notifications	event-amazon s3-async(lambda)-DLQ-SQS
			S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication
			• If you want to ensure that an event notification is sent for every successful write, you can enable versioning on your bucket
		Lambda – Event Source Mapping	Lambda read data from kinesis then internally Event Source mapping is created which is responsible to poll data and getting the result back from kinesis then it will invoke lambda
		Streams & Lambda – Error Handling	By default, if your function returns an error, the entire batch is reprocessed until the function succeeds, or the items in the batch expire. Note: DLQ for lambda is only work for asynchronous
		Lambda Event Mapper Scaling	
		Kinesis Data Streams & DynamoDB Streams:	One Lambda invocation per stream shard
			If you use parallelization, up to 10 batches processed per shard simultaneously
		SQS Standard	Lambda adds 60 more instances per minute to scale up
			Up to 1000 batches of messages processed simultaneously
		SQS FIFO	Messages with the same GroupID will be processed in order
			The Lambda function scales to the number of active message groups
		Lambda event source mapping hands on (SQS)	first create lambda then service(SQS) attach service with lambda(by trigger and attach IAM to lambda) do something in service then check using cloudwatch lambda function invoke or not
		Lambda Destination	it very tough for asynchronous call to see whether it succeed or not so idea is that to send result of asynchronous to destination. destination allow both successful and failure while DLQ allow only failure
		Asynchronous invocations	can define destinations for successful and failed event
			Amazon SQS • Amazon SNS • AWS Lambda • Amazon EventBridge bus
		Lambda Execution Role (IAM Role)	Grants the Lambda function permissions to AWS services / resources

		Sample managed policies for Lambda:	
		AWSLambdaBasicExecutionRole	Upload logs to CloudWatch
		AWSLambdaKinesisExecutionRole	Read from Kinesis
		AWSLambdaDynamoDBExecutionRole	Read from DynamoDB Streams
		AWSLambdaSQSQueueExecutionRole –	Read from SQS
		AWSLambdaVPCAccessExecutionRole	Deploy Lambda function in VPC
		AWSXRayDaemonWriteAccess	Upload trace data to X-Ray
		Lambda Resource Based Policies	Use resource-based policies to give other accounts and AWS services permission to use your Lambda resources
			When an AWS service like Amazon S3 calls your Lambda function, the resource-based policy gives it access
			Lambda invoking SQS so it is not a resource based policy
		Lambda Environment Variables	Environment variable = key / value pair in "String" form
			Adjust the function behavior without updating code
			Helpful to store secrets (encrypted by KMS)
			Secrets can be encrypted by the Lambda service key, or your own CMK
		Lambda Logging & Monitoring	
		• CloudWatch Logs:	AWS Lambda execution logs are stored in AWS CloudWatch Logs
			Make sure your AWS Lambda function has an execution role with an IAM policy that authorizes writes to CloudWatch Logs
		Lambda Tracing with X-Ray	Enable in Lambda configuration (Active Tracing) • Runs the X-Ray daemon for you • Use AWS X-Ray SDK in Code
			• Ensure Lambda Function has a correct IAM Execution Role • The managed policy is AWSXRayDaemonWriteAccess
		Lambda by default	• By default, your Lambda function is launched outside your own VPC (in an AWS -owned VPC)
			Therefore it cannot access resources in your VPC (RDS, ElastiCache, internal ELB…)
		Lambda in VPC	You must define the VPC ID, the Subnets and the Security Groups
			Lambda will create an ENI (Elastic Network Interface) in your subnets
			AWSLambdaVPCAccessExecutionRole
		Lambda in VPC – Internet Access	Deploying a Lambda function in a public subnet does not give it internet access or a public IP
			Deploying a Lambda function in a private subnet gives it internet access if you have a NAT Gateway / Instance.
			You can use VPC endpoints to privately access AWS services without a NAT
		good to know	lambda function is launched outside VPC. Solution: we can deploy lambda in VPC
			to access RDS lambda will go through ENI(elastic network interface) which will created internally
		Lambda Function Configuration	
		RAM	From 128MB to 3,008MB in 64MB increments
			At 1,792 MB, a function has the equivalent of one full vCPU
			If your application is CPU-bound (computation heavy), increase RAM
		Timeout	default 3 seconds, maximum is 900 seconds (15 minutes)
		Lambda Execution Context	The execution context is a temporary runtime environment that initializes any external dependencies of your lambda code
			Great for database connections, HTTP clients, SDK clients

			The execution context includes the /tmp directory
		Lambda Functions /tmp space	If your Lambda function needs to download a big file to work
			You can use the /tmp directory. Max size is 512MB
			• For permanent persistence of object (non temporary), use S3
		Lambda Concurrency and Throttling	for each lambda function we can set limit(means upto this lambda function can scale) if it exceed it will throw a throttle.Concurrency limit: up to 1000 concurrent executions.If one function goes over limit other function can get throttle
		Throttle behavior	
		If synchronous invocation	return ThrottleError - 429
		• If asynchronous invocation	retry automatically and then go to DLQ
		cold start	**AWS** can drop the container after a period of inactivity, and your function becomes inactive or **cold**. A **cold start** happens when you execute an inactive Lambda function. The execution of an inactive Lamda function happens when there are no available containers, and the function needs to **start** up a new one.
		Concurrency	**Concurrency** is the number of requests that your function is serving at any given time
		concurrency limit in **Lambda**	Each account has a concurrency limit in **Lambda**. This limit specifies the number of **function** invocations that can be running at the same time. When the concurrency limit is hit, **Lambda** will not invoke a **function** and will **throttle** it instead
		Lambda and CloudFormation – through S3	You must store the Lambda zip in S3
			• You must refer the S3 zip location in the CloudFormation code • S3Bucket • S3Key: full path to zip • S3ObjectVersion: if versioned bucket
			• If you update the code in S3, but don't update S3Bucket, S3Key or S3ObjectVersion, CloudFormation won't update your function
		layer	A **layer** is a ZIP archive that contains libraries, a custom runtime, or other dependencies. With **layers**, you can use libraries in your function without needing to include them in your deployment package. **Layers** let you keep your deployment package small, which makes development easier.
		AWS Lambda Versions	When you work on a Lambda function, we work on $LATEST. everything in version is immutable can change only in latest
			When we're ready to publish a Lambda function, we create a version
			Versions are immutable. Versions have increasing version numbers
			Versions get their own ARN (Amazon Resource Name)
			Version = code + configuration (nothing can be changed - immutable)
			Each version of the lambda function can be accessed
		AWS Lambda Aliases	Aliases are "pointers" to Lambda function versions
			We can define a "dev", "test", "prod" aliases and have them point at different lambda versions
			Aliases are mutable. Aliases have their own ARNs . Alias cannot reference alias
			Aliases enable Blue / Green deployment by assigning weights to lambda functions
			Aliases enable stable configuration of our event triggers / destinations

		Lambda & CodeDeploy	CodeDeploy can help you automate traffic shift for Lambda aliases	
			• Feature is integrated within the SAM framework	
			• Linear: grow traffic every N minutes until 100%	
			• Canary: try X percent then 100%	
			• AllAtOnce: immediate	
		AWS Lambda Limits to Know - per region		
		Execution:		
		Memory allocation	128 MB – 3008 MB (64 MB increments)	
		Maximum execution time:	900 seconds (15 minutes)	
		• Environment variables	(4 KB)	
		Disk capacity in the “function container” (in /tmp):	512 MB	
		Concurrency executions	1000 (can be increased)	
		Deployment:		
		Lambda function deployment size (compressed .zip)	50 MB	
		Size of uncompressed deployment (code + dependencies):	250 MB	
Questions				
1		**Which of the following service does NOT require an event source mapping?**	SNS	

Terms	Full Form	Definition	NOTE
Topics 18:) Dynamodb			
NoSQL databases		NoSQL databases are non-relational databases and are distributed	NoSQL databases include MongoDB, DynamoDB, etc
			NoSQL databases do not support join
			All the data that is needed for a query is present in one row
			NoSQL databases don’t perform aggregations such as “SUM”
			NoSQL databases scale horizontally
DynamoDB		It is nosql serverless database and manage by aws.NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph.	Fully Managed, Highly available with replication across 3 AZ
			Millions of requests per seconds, trillions of row, 100s of TB of storage
			Integrated with IAM for security, authorization and administration
			Low cost and auto scaling capabilities
		DynamoDB - Basics	DynamoDB is made of tables
			Each table has a primary key (must be decided at creation time)
			Each table can have an infinite number of items (= rows)
			Each item has attributes (can be added over time – can be null)
			Maximum size of a item is 400KB
		Data types supported are:	Scalar Types: String, Number, Binary, Boolean, Null
			Document Types: List, Map • Set Types: String Set, Number Set, Binary Set
		DynamoDB – Primary Keys	
		Option 1: Partition key only (HASH)	Partition key must be unique for each item • Example: user_id for a users table
		Option 2: Partition key + Sort Key	The combination must be unique. Data is grouped by partition key
			Example: users-games table • user_id for the partition key • game_id for the sort key
		good to know	don’t need to create database that will be manage by AWS. need to create table
		DynamoDB – Provisioned Throughput	Table must have provisioned read and write capacity units
			Read Capacity Units (RCU): throughput for reads
			Write Capacity Units (WCU): throughput for writes
			• If burst credit are empty, you’ll get a “ProvisionedThroughputException”
			• It’s then advised to do an exponential back-off retry
		DynamoDB – Write Capacity Units	forrmula)Find wcu ----per second----one wcu===1 kb
			One write capacity unit represents one write per second for an item up to 1 KB in size
			If the items are larger than 1 KB, more WCU are consumed
		Example 1: we write 10 objects per seconds of 2 KB each.	We need 2 * 10 = 20 WCU
		Example 2: we write 6 objects per second of 4.5 KB each	We need 6 * 5 = 30 WCU (4.5 gets rounded to the upper KB)
		Example 3: we write 120 objects per minute of 2 KB each	We need 120 / 60 * 2 = 4 WCU
		Eventually Consistent Read:	If we read just after a write, it’s possible we’ll get unexpected response because of replication

		Strongly Consistent Read	: If we read just after a write, we will get the correct data
		By default	: DynamoDB uses Eventually Consistent Reads
		Formula for RCU	1)per second / 4kb in size 2)Strongly consistent read---1 3) Eventually consistent read---2. round off
		DynamoDB – Read Capacity Units	One read capacity unit represents one strongly consistent read per second, or two eventually consistent reads per second, for an item up to 4 KB in size
			If the items are larger than 4 KB, more RCU are consumed
		• Example 1: 10 strongly consistent reads per seconds of 4 KB each	We need 10 * 4 KB / 4 KB = 10 RCU
		Example 2: 16 eventually consistent reads per seconds of 12 KB each	We need (16 / 2) * (12 / 4) = 24 RCU
		• Example 3: 10 strongly consistent reads per seconds of 6 KB each	We need 10 * 8 KB / 4 = 20 RCU (we have to round up 6 KB to 8 KB)
		DynamoDB – Partitions Internal	Data is divided in partitions
			Partition keys go through a hashing algorithm to know to which partition they go to
		To compute the number of partitions:	• By capacity: (TOTAL RCU / 3000) + (TOTAL WCU / 1000) • By size: Total Size / 10 GB •
			Total partitions = CEILING(MAX(Capacity, Size))
		Amazon **DynamoDB** Accelerator (**DAX**)	is a fully managed, highly available, in-memory cache for Amazon **DynamoDB** that delivers up to a 10 times performance improvement—from milliseconds to microseconds—even at millions of requests per second.
			Writes go through DAX to DynamoDB • Micro second latency for cached reads & queries • Solves the Hot Key problem (too many reads)
			5 minutes TTL for cache by default • Up to 10 nodes in the cluster • Multi AZ (3 nodes minimum recommended for production) • Secure (Encryption at rest with KMS, VPC, IAM,
		DynamoDB -Throttling	• If we exceed our RCU or WCU, we get ProvisionedThroughputExceededExceptions
		Reasons:	Hot keys: one partition key is being read too many times (popular item for ex)
			Hot partitions:
			Very large items: remember RCU and WCU depends on size of items
		Solutions:	Exponential back-off when exception is encountered (already in SDK)
			Distribute partition keys as much as possible
			If RCU issue, we can use DynamoDB Accelerator (DAX)
		DynamoDB – Writing Data	
		PutItem	Write data to DynamoDB (create data or full replace) • Consumes WCU
		UpdateItem	Update data in DynamoDB (partial update of attributes) • Possibility to use Atomic Counters and increase them
		Conditional Writes	Accept a write / update only if conditions are respected, otherwise reject • Helps with concurrent access to items
		DynamoDB – Deleting Data	
		DeleteItem	Delete an individual row • Ability to perform a conditional delete

		DeleteTable	Delete a whole table and all its items
		DynamoDB – Batching Writes	
		BatchWriteItem	Up to 25 PutItem and / or DeleteItem in one call
			Up to 16 MB of data written
			Up to 400 KB of data per item
			Operations are done in parallel for better efficiency. Reduce number of API call
			It's possible for part of a batch to fail, in which case we have the try the failed items (using exponential back-off algorithm)
exponential backoff			is to use progressively longer waits between retries for consecutive error responses. You should implement a maximum delay interval, as well as a maximum number of retries
		DynamoDB – Reading Data	
		GetItem:	Read based on Primary key • Primary Key = HASH or HASH-RANGE
			Eventually consistent read by default
			Option to use strongly consistent reads (more RCU - might take longer)
			ProjectionExpression can be specified to include only certain attributes
		BatchGetItem	Up to 100 items • Up to 16 MB of data • Items are retrieved in parallel to minimize latency
		DynamoDB – Query	
		• Query returns items based on:	PartitionKey value (must be = operator) • SortKey value (=, <=, >, >=, Between, Begin) – optional • FilterExpression to further filter (client side filtering)
		Returns:	Up to 1 MB of data • Or number of items specified in Limit
			Can query table, a local secondary index, or a global secondary index
		DynamoDB - Scan	Scan the entire table and then filter out data (inefficient)
			• Can use a ProjectionExpression + FilterExpression (no change to RCU)
		DynamoDB – LSI (Local Secondary Index)	has the same partition key as the primary key (index), but a different range key. The way to think about an **LSI** is that its the same data as the primary index (key), just ordered by a different attribute
			Alternate range key for your table, local to the hash key
			Up to five local secondary indexes per table
			LSI must be defined at table creation time
		DynamoDB – GSI (Global Secondary Index)	whole new table. To speed up queries on non-key attributes, use a Global Secondary Index
			GSI = partition key + optional sort key
		The index is a new "table" and we can project attributes on it	• The partition key and sort key of the original table are always projected (KEYS_ONLY)
			Can specify extra attributes to project (INCLUDE
			Can use all attributes from main table (ALL)
		important point	Possibility to add / modify GSI (not LSI)

		DynamoDB Indexes and Throttling	
		GSI:	• If the writes are throttled on the GSI, then the main table will be throttled!
			Choose your GSI partition key carefully
		LSI:	Uses the WCU and RCU of the main table • No special throttling considerations
		DynamoDB Concurrency	DynamoDB has a feature called "Conditional Update / Delete"
		DynamoDB Streams	is a powerful service that you can combine with other AWS services to solve many similar problems. When enabled, **DynamoDB Streams** captures a time-ordered sequence of item-level modifications in a **DynamoDB** table and durably stores the information for up to 24 hours.
			DynamoDB stream represent change log of things happen in table(read,update,delete)
			This stream can be read by AWS Lambda & EC2 instances, and we can then do: • React to changes in real time (welcome email to new users)
			Could implement cross region replication using Streams
			DynamoDB Streams are made of shards, just like Kinesis Data Streams
		good to know	An **event source mapping** is an AWS Lambda resource that reads from an **event source** and invokes a Lambda function
		DynamoDB Streams & Lambda	You need to define an Event Source Mapping to read from a DynamoDB Streams
			You need to ensure the Lambda function has the appropriate permissions
			Your Lambda function is invoked synchronously
		DynamoDB -TTL (Time to Live)	TTL = automatically delete an item after an expiry date / time
			TTL is enabled per row (you define a TTL column, and add a date there)
			DynamoDB typically deletes expired items within 48 hours of expiration
			Deleted items due to TTL are also deleted in GSI / LSI
			DynamoDB Streams can help recover expired items
		DynamoDB Transactions	Transaction = Ability to Create / Update / Delete multiple rows in different tables at the same time
			Write Modes: Standard, Transactional
			Read Modes: Eventual Consistency, Strong Consistency, Transactional
			Consume 2x of WCU / RCU
		DynamoDB as Session State Cache	It's common to use DynamoDB to store session state
		vs ElastiCache	ElastiCache is in-memory, but DynamoDB is serverless • Both are key/value stores
		vs EFS	EFS must be attached to EC2 instances as a network drive
		vs EBS & Instance Store	EBS & Instance Store can only be used for local caching, not shared caching
		vs S3	S3 is higher latency, and not meant for small objects
		DynamoDB Operations	Copy dynamodb table) copy into s3 and back put it into dynamodb table
		Table Cleanup	
		Option 1: Scan + Delete	very slow, expensive, consumes RCU & WCU

		Option 2: Drop Table + Recreate table	fast, cheap, efficient
		Copying a DynamoDB Table:	
		• Option 1:	Use AWS DataPipeline (uses EMR)
		Option 2	Create a backup and restore the backup into a new table name (can take some time)
		Option 3	Scan + Write => write own code
		DynamoDB – Security & Other Features	
		Security	VPC Endpoints available to access DynamoDB without internet • Access fully controlled by IAM • Encryption at rest using KMS • Encryption in transit using SSL / TLS
		Backup and Restore feature available	Point in time restore like RDS • No performance impact
		Global Tables	Multi region, fully replicated, high performance
			Amazon DMS can be used to migrate to DynamoDB (from Mongo, Oracle, MySQL, S3, etc…)
			You can launch a local DynamoDB on your computer for development purposes
Questions			
1		optimistic locking cab be implemented with dynamo db	
2		conditional write allow optimistic locking	

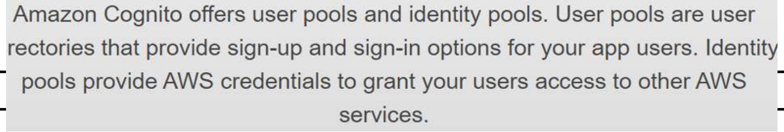
Terms	Full Form	Definition	NOTE
Topics 19): serverless api gateway			
API Gateway		API Gateway is in building **serverless** HTTP **APIs**	Api -gateway connect aws lambda connect dynamo db
		client can invoke lambda function in multiple ways	Client can invoke directly to lambda function
			Client can invoke using ALB where lambda function is configured
			Client will talk to API gateway and it will connect to lambda function
		AWS API Gateway	Support for the WebSocket Protocol • Handle API versioning (v1, v2…) • Handle different environments (dev, test, prod)
			Handle security (Authentication and Authorization) • Create API keys, handle request throttling • Swagger / Open API import to quickly define APIs
			Transform and validate requests and responses • Generate SDK and API specifications • Cache API responses
		API Gateway - Endpoint Types	
		Edge-Optimized (default):	For global clients
			Requests are routed through the CloudFront Edge locations (improves latency)
			The API Gateway still lives in only one region
		Regional	For clients within the same region
			Could manually combine with CloudFront (more control over the caching strategies and the distribution)
		Private	Can only be accessed from your VPC using an interface VPC endpoint (ENI)
			Use a resource policy to define access
		API Gateway – Deployment Stages	Making changes in the API Gateway does not mean they're effective
			You need to make a "deployment" for them to be in effect
			Changes are deployed to "Stages" (as many as you want) • Use the naming you like for stages (dev, test, prod)
			Each stage has its own configuration parameters • Stages can be rolled back as a history of deployments is kept
		API Gateway – Stage Variables	Stage variables are like environment variables for API Gateway
		They can be used in:	Lambda function ARN • HTTP Endpoint • Parameter mapping templates
		Use cases:	Configure HTTP endpoints your stages talk to (dev, test, prod…)
			Pass configuration parameters to AWS Lambda through mapping templates
			Stage variables are passed to the "context" object in AWS Lambda
		API Gateway Stage Variables & Lambda Aliases	We create a stage variable to indicate the corresponding Lambda alias
			Our API gateway will automatically invoke the right Lambda function!
		API Gateway – Canary Deployment	Possibility to enable canary deployments for any stage (usually prod)
			Choose the % of traffic the canary channel receives
			This is blue / green deployment with AWS Lambda & API Gateway
		API Gateway - Integration Types	
		Integration Type MOCK	API Gateway returns a response without sending the request to the backend

		Integration Type HTTP / AWS (Lambda & AWS Services)	Setup data mapping using mapping templates for the request & response
		Integration Type AWS_PROXY (Lambda Proxy):	incoming request from the client is the input to Lambda
			The function is responsible for the logic of request / response
			No mapping template, headers, query string parameters… are passed as arguments
		Integration Type HTTP_PROXY	No mapping template(client-api gateway-ALB)
			The HTTP request is passed to the backend
			The HTTP response from the backend is forwarded by API Gateway
		Mapping Templates (AWS & HTTP Integration)	Mapping templates can be used to modify request / responses
			Rename / Modify query string parameters • Modify body content • Add headers
			Filter output results (remove unnecessary data)
		Mapping Example: JSON to XML with SOAP	• SOAP API are XML based, whereas REST API are JSON based
			• In this case, API Gateway should: • Extract data from the request: either path, payload or header
		Caching API responses	how to enable caching in API Gateway
			Caching reduces the number of calls made to the backend
			Default TTL (time to live) is 300 seconds (min: 0s, max: 3600s)
			Caches are defined per stage
			Cache capacity between 0.5GB to 237GB
			Cache is expensive, makes sense in production, may not make sense in dev / test
		API Gateway Cache Invalidation	Able to flush the entire cache (invalidate it) immediately
			Clients can invalidate the cache with header: Cache- Control: max-age=0 (with proper IAM authorization)
			If you don't impose an InvalidateCache policy (or choose the Require authorization check box in the console), any client can invalidate the API cache
Throttling limits		define the maximum number of requests per second available to each key	
Quota limits		define the number of requests each API key is allowed to make over a period.	
		API Gateway – Usage Plans & API Keys	
		Usage Plan:	who can access one or more deployed API stages and methods
			how much and how fast they can access them
			uses API keys to identify API clients and meter access
			configure throttling limits and quota limits that are enforced on individual client
		API Keys	alphanumeric string values to distribute to your customers • Ex: WBjHxNtoAb4WPKBC7cGm64CBibIb24b4jt8jJHo9
			Can use with usage plans to control access
			Throttling limits are applied to the API keys
			Quotas limits is the overall number of maximum requests

		API Gateway – Correct Order for API keys	Create one or more APIs, configure the methods to require an API key, and deploy the APIs to stages.
			Generate or import API keys to distribute to application developers (your customers) who will be using your API
			reate the usage plan with the desired throttle and quota limits.
			Associate API stages and API keys with the usage plan.
		API Gateway – Logging & Tracing	
		CloudWatch Logs:	Enable CloudWatch logging at the Stage level (with Log Level)
			Can override settings on a per API basis (ex: ERROR, DEBUG, INFO)
			Log contains information about request / response body
		X-Ray	Enable tracing to get extra information about requests in API Gateway
			X-Ray API Gateway + AWS Lambda gives you the full picture
		Integration latency	how much time backend take to reply to response
		NOTE	latency is going to be higher than integration latency since in latency we check some other stuff also like authorization and authentication time.Maximum time for api : 29 second if it is over them timeout we will get
		API Gateway – CloudWatch Metrics	
			Metrics are by stage, Possibility to enable detailed metrics
		CacheHitCount & CacheMissCount	efficiency of the cache
		Count	The total number API requests in a given period
		IntegrationLatency	The time between when API Gateway relays a request to the backend and when it receives a response from the backend
		Latency	The time between when API Gateway receives a request from a client and when it returns a response to the client. The latency includes the integration latency and other API Gateway overhead
			4XXError (client-side) & 5XXError (server-side)
		API Gateway Throttling	
		Account Limit	API Gateway throttles requests at10000 rps across all API • Soft limit that can be increased upon request
		• In case of throttling	429 Too Many Requests (retriable error)
			Can set Stage limit & Method limits to improve performance • Or can define Usage Plans to throttle per customer
			Just like Lambda Concurrency, one API that is overloaded, if not limited, can cause the other APIs to be throttled
		API Gateway - Errors	
		• 4xx means Client errors	400: Bad Request • 403: Access Denied, WAF filtered • 429: Quota exceeded, Throttle
		5xx means Server errors	
		502	Bad Gateway Exception, usually for an incompatible output returned from a Lambda proxy integration backend and occasionally for out-of-order invocations due to heavy loads.
		503	: Service Unavailable Exception
		504	Integration Failure – ex Endpoint Request Timed-out Exception API Gateway requests time out after 29 second maximum

		AWS API Gateway - CORS	CORS must be enabled when you receive API calls from another domain.
		API Gateway – Security	
		NOTE	user will send sig v4 to api gateway and it will check from IAM if it is authorized then it will talk to lambda function
		API Gateway – Resource Policies	Resource policies (similar to Lambda Resource Policy)
			Allow for Cross Account Access (combined with IAM Security)
			Allow for a specific source IP address
			Allow for a VPC Endpoint
		Cognito	database of users
		API Gateway – Security – Summary	
		IAM	Great for users / roles already within your AWS account, + resource policy for cross account
			Handle authentication + authorization
			Leverages Signature v4
		Custom Authorizer:	Great for 3rd party tokens
			Handle Authentication verification + Authorization in the Lambda function
			• Pay per Lambda invocation, results are cached
		Cognito User Pool:	You manage your own user pool (can be backed by Facebook, Google login etc…)
			No need to write any custom code • Must implement authorization in the backend
		note	first user will get token from cognito and api gateway match token with cognito
			If it match then it will allow to access lambda function
		API Gateway – WebSocket API – Overview	WebSocket APIs are often used in real-time applications such as chat applications, collaboration platforms, multiplayer games, and financial trading platforms.

Terms	Full Form	Definition	NOTE
Topics 20) : Serverless Application Model			
AWS SAM	Serverless Application Mode	allow to deploy application into AWS	SAM) is an open-source framework for building **serverless applications**. It provides shorthand syntax to express functions, APIs, databases, and event source mappings. With just a few lines per resource, you can define the **application** you want and **model** it using YAML
			Framework for developing and deploying serverless applications
			All the configuration is YAML code
			Generate complex CloudFormation from simple SAM YAML file
			Supports anything from CloudFormation: Outputs, Mappings, Parameters, Resources
			SAM can use CodeDeploy to deploy Lambda functions
			SAM can help you to run Lambda, API Gateway, DynamoDB locally
		AWS SAM – Recipe	Function : Amazon lambda **Api : API Gateway SimpleTable : DynamoDb**
		Transform Header indicates it's SAM template	Transform: 'AWS::Serverless-2016-10 -31
		Write Code	AWS::Serverless::Function • AWS::Serverless::Api • AWS::Serverless::SimpleTable
		Package & Deploy	aws cloudformation package / sam package • aws cloudformation deploy / sam deploy
		NOTE	generated template will have the reference of application code into S3
			Stack could be dynamodb , api gateway . Transform : indicating we are using a SAM template
		SAM Policy Templates	• List of templates to apply permissions to your Lambda Functions
		Important examples	
		S3ReadPolicy	Gives read only permissions to object in S3
		SQSPollerPolicy	Allows to poll an SQS queue
		DynamoDBCrudPolicy:	CRUD = create read update delete
		SAM – Exam Summary	SAM is built on CloudFormation
			SAM requires the Transform and Resources sections
		Commands to know	
		sam build:	fetch dependencies and create local deployment artifacts
		sam package	package and upload to Amazon S3, generate CF template
		sam deploy	deploy to CloudFormation
			SAM Policy templates for easy IAM policy definition
			• SAM is integrated with CodeDeploy to do deploy to Lambda aliases
		note	api gateway talk to lambda function and lambda function talk to dynamo db and also lambda function IAM Policy is there
			Cloudformation will deploy from code deploy to lambda function
Questions			
1)		**two commands to run to upload Lambda functions and CloudFormation templates to AWS**	cloudformation package and cloudformation deploy

Terms	Full Form	Definition	NOTE
topic 21) Cognito-cognito user pool, cognito identity pool and cognito sync			
Cognito		: give an identity to users to access application outside the cloud (like mobile application)	 Amazon Cognito offers user pools and identity pools. User pools are user rectories that provide sign-up and sign-in options for your app users. Identity pools provide AWS credentials to grant your users access to other AWS services.
User pools		are for authentication (identify verification).	
Identity pool		are for authorization (access control).	
		Amazon Cognito	We want to give our users an identity so that they can interact with our application.
		Cognito User Pools	Sign in functionality for app users
			Integrate with API Gateway & Application Load Balancer
		Cognito Identity Pools (Federated Identity):	Provide AWS credentials to users so they can access AWS resources directly
			Integrate with Cognito User Pools as an identity provider
		Cognito Sync	Synchronize data from device to Cognito
			is deprecated and replaced by AppSync
		Cognito vs IAM:	“hundreds of users”, ”mobile users”, “authenticate with SAML
		Cognito User Pools (CUP) – User Features	When user login or register to any other website
			Create a serverless database of user for your web & mobile apps • Simple login: Username (or email) / password combination • Password reset • Email & Phone Number Verification • Multi-factor authentication (MFA) • Federated Identities: users from Facebook, Google, SAML… • Feature: block users if their credentials are compromised elsewhere • Login sends back a JSON Web Token (JWT)
		NOTE	Whenever user login or sign up these are the events happening in user pool and we may
			want To invoke lambda function to react to these events. We can set up lambda function with these events
		Cognito User Pools – Lambda Triggers	
		1)Authentication Events	
		Pre Authentication Lambda Trigger	Custom validation to accept or deny the sign-in request
		Post Authentication Lambda Trigger	Event logging for custom analytics
		Pre Token Generation Lambda Trigger	Augment or suppress token claims
		2)Sign-Up	
		Pre Sign-up Lambda Trigger	Custom validation to accept or deny the sign-up request
		Post Confirmation Lambda Trigger	Custom welcome messages or event logging for custom analytics
		Migrate User Lambda Trigger	Migrate a user from an existing user directory to user pools
		3)Messages	
		Custom Message Lambda Trigger	Advanced customization and localization of messages
		4) Token Creation	

		Pre Token Generation Lambda Trigger	Add or remove attributes in Id tokens
		Cognito User Pools – Hosted Authentication UI	Cognito has a hosted authentication UI that you can add to your app to handle signup and sign-in workflows
			Using the hosted UI, you have a foundation for integration with social logins, OIDC or SAML
			• Can customize with a custom logo and custom CSS
		Cognito Identity Pools (Federated Identities)	Outside users want access to aws environment (ex dynamodb table or s3)
			We don't provide IAM to these users because so many are there and we cant trust them so we allow to access aws through cognito user pool
			Cognito Identity Pools allow for unauthenticated (guest) access
AWS STS	Security Token Service		is a web service that enables you to request temporary, limited-privilege credentials for **AWS** Identity and Access Management (IAM) users or for users that you authenticate (federated users).
		Cognito Identity Pools – IAM Roles	Default IAM roles for authenticated and guest user
			IAM credentials are obtained by Cognito Identity Pools through STS
			The roles must have a "trust" policy of Cognito Identity Pools
		Cognito User Pools vs Identity Pools	
		• Cognito User Pools:	• Database of users for your web and mobile application
			Allows to federate logins through Public Social, OIDC, SAML
			• Can customize the hosted UI for authentication (including the logo)]
			Has triggers with AWS Lambda during the authentication flow
		Cognito Identity Pools:	Obtain AWS credentials for your users
			Users can login through Public Social, OIDC, SAML & Cognito User Pools
			Users can be unauthenticated (guests)
			Users are mapped to IAM roles & policies, can leverage policy variables
		NOTE	CUP + CIP = manage user / password + access AWS services
		Cognito Sync	Deprecated – use AWS AppSync now
			• Offline capability (synchronization when back online)
			Store data in datasets (up to 1MB), up to 20 datasets to synchronize
			• Push Sync: silently notify across all devices when identity data changes
			• Cognito Stream: stream data from Cognito into Kinesis
			Cognito Events: execute Lambda functions in response to events

Terms	Full Form	Definition	NOTE
		Topics 22) Other serverless - step function and app sync	
step function overview			AWS Step Functions lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Using Step Functions, you can design and run workflows that stitch together services such as AWS Lambda and Amazon ECS into feature-rich applications. Workflows are made up of a series of steps, with the output of one step acting as input into the next. Application development is simpler and more intuitive using Step Functions, because it translates your workflow into a state machine diagram that is easy to understand, easy to explain to others, and easy to change. You can monitor each step of execution as it happens, which means you can identify and fix problems quickly. Step Functions automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected.
		AWS Step Functions	Build serverless visual workflow to orchestrate your Lambda functions
			Can also integrate with EC2, ECS, On premise servers, API Gateway
			Maximum execution time of 1 year
			Use cases: • Order fulfillment • Data processing • Web applications • Any workflow
		Step Functions – Error Handling	By default, when a state reports an error, AWS Step Functions causes the execution to fail entirely
			Retrying failures - Retry: IntervalSeconds, MaxAttempts, BackoffRate
			Moving on - Catch: ErrorEquals, Next
		Step Functions – Standard vs Express	
		Standard Workflows	
		Maximum duration	1 year
		Supported execution start rate	Over 2,000 per second
		Supported state transition rate	Over 4,000 per second per account
		Execution semantics	Exactly-once workflow execution.
		Express Workflows	
		Maximum duration	5 minutes
		Supported execution start rate	Over 100,000 per second
		Supported state transition rate	Nearly unlimited
		Execution semantics	At-least-once workflow execution
		AWS AppSync - Overview	**AppSync** is a managed GraphQL service that makes it easy to build mobile and web applications. The power of **AppSync** is that it allows you to build, mange and synchronize real-time subscriptions while also allowing you to have access to **app** data when mobile devices are offline
			AppSync is a managed service that uses GraphQL
			GraphQL makes it easy for applications to get exactly the data they need.
		This includes combining data from one or more source	NoSQL data stores, Relational databases, HTTP APIs.
			Integrates with DynamoDB, Aurora, Elasticsearch & others

			Custom sources with AWS Lambda
		AppSync – Security	There are four ways you can authorize applications to interact with your AWS AppSync GraphQL API:
			API_KEY
			AWS_IAM: IAM users / roles / cross-account access
			OPENID_CONNECT: OpenID Connect provider / JSON Web Token
			AMAZON_COGNITO_USER_POOLS
question			
		Which of the following does NOT allow for a real-time WebSocket API?	DynamoDB on its own does not push changes to the users and does not have a two-way communication. It's just a request/response database

Terms	Full Form	Definition	NOTE
			Topics 23) : Advanced Identity
STS	Security Token Service	allow to get temporary access	
		AWS STS – Security Token Service	
			Allows to grant limited and temporary access to AWS resources (up to 1 hour)
		AssumeRole	Assume roles within your account or cross account
		AssumeRoleWithSAML	return credentials for users logged with SAML
		AssumeRoleWithWebIdentity	return creds for users logged with an IdP (Facebook Login, Google Login, OIDC compatible…)
			AWS recommends against using this, and using Cognito Identity Pools instead
		GetSessionToken	for MFA, from a user or AWS account root user
		GetFederationToken:	obtain temporary creds for a federated user
		GetCallerIdentity	return details about the IAM user or role used in the API call
		DecodeAuthorizationMessage	decode error message when an AWS API is denied
		Using STS to Assume a Role	Temporary credentials can be valid between 15 minutes to 1 hour
		STS with MFA	
			Use GetSessionToken from STS
			• Appropriate IAM policy using IAM Conditions
			aws:MultiFactorAuthPresent:true
			Reminder, GetSessionToken returns: • Access ID • Secret Key • Session Token • Expiration date
		IAM Best Practices – General	• Never use Root Credentials, enable MFA for Root Account
		Grant Least Privilege	Each Group / User / Role should only have the minimum level of permission it needs
			Never grant a policy with "*" access to a service
			Monitor API calls made by a user in CloudTrail (especially Denied ones)
			Never ever ever store IAM key credentials on any machine but a personal computer or on-premise server
			On premise server best practice is to call STS to obtain temporary security credentials
			• EC2 machines should have their own roles • Lambda functions should have their own roles • ECS Tasks should have their own roles (ECS_ENABLE_TASK_IAM_ROLE=true) • CodeBuild should have its own service role • Create a least-privileged role for any service that requires it • Create a role per application / lambda function (do not reuse roles)
		good to know	explicit deny has higher policy then explicit allow
		IAM Policies & S3 Bucket Policies	IAM Policies are attached to users, roles, groups
			S3 Bucket Policies are attached to buckets
			When evaluating if an IAM Principal can perform an operation X on a bucket, the union of its assigned IAM Policies and S3 Bucket Policies will be evaluated.
		Example 1	

		IAM Role attached to EC2 instance, authorizes RW to “my_bucket”	EC2 instance can read and write to “my_bucket”
		• No S3 Bucket Policy attached	
		Example 2	
		IAM Role attached to EC2 instance, authorizes RW to “my_bucket”	EC2 instance cannot read and write to “my_bucket
		S3 Bucket Policy attached, explicit deny to the IAM Role	
		Example 3	
		IAM Role attached to EC2 instance, no S3 bucket permissions	EC2 instance can read and write to “my_bucket”
		S3 Bucket Policy attached, explicit RW allow to the IAM Role	
		Example 4	
		IAM Role attached to EC2 instance, explicit deny S3 bucket permissions	EC2 instance cannot read and write to “my_bucket”
		S3 Bucket Policy attached, explicit RW allow to the IAM Role	
		Inline vs Managed Policies	
		AWS Managed Policy	Maintained by AWS • Good for power users and administrators • Updated in case of new services / new APIs
		Customer Managed Policy	Best Practice, re-usable, can be applied to many principals
			Version Controlled + rollback, central change management
		Inline	Strict one-to-one relationship between policy and principal
			Policy is deleted if you delete the IAM principal
		Granting a User Permissions to Pass a Role to an AWS Service	To configure many AWS services, you must pass an IAM role to the service (this happens only once during setup)
		• Example of passing a role	To an EC2 instance • To a Lambda function • To an ECS task • To CodePipeline to allow it to invoke other services
			• For this, you need the IAM permission iam:PassRole
		Can a role be passed to any service	No: Roles can only be passed to what their trust allows
		note	**to pass a role we need to create correct trust relationship**
		Directory Service Overview	**Active Directory** stores information about objects on the network and makes this information easy for administrators and users to find and **use**. **Active Directory uses** a structured data store as the basis for a logical, hierarchical organization of **directory** information.
		AWS Directory Services	
		AWS Managed Microsoft AD	Create your own AD in AWS, manage users locally, supports MFA

		AWS Managed Microsoft AD	Establish “trust” connections with your on- premise AD
		• AD Connector	Directory Gateway (proxy) to redirect to on- premise AD
			Users are managed on the on-premise AD
		Simple AD	AD-compatible managed directory on AWS
			Cannot be joined with on-premise AD