## Elastic beanstalk

### There is no direct Export and Import option for migrating Elastic Beanstalk configurations.

* You must use saved configurations to migrate an Elastic Beanstalk environment between AWS accounts

### Immutable deployment

* This policy ensures that your new application version is always deployed to new instances, instead of updating existing instances.
* It also has the additional advantage of a quick and safe rollback in case the deployment fails.
* In an immutable update, a second Auto Scaling group is launched in your environment and the new version serves traffic alongside the old version until the new instances pass health checks.
* In case of deployment failure, the new instances are terminated, so the impact is minimal.

### All at once

* The quickest deployment method, but the application may become unavailable to users (or have low availability) for a short time. Also in case of deployment failure, the application sees a downtime

*Rolling*

* This policy avoids downtime and minimizes reduced availability, at a cost of a longer deployment time.
* However in case of deployment failure, the rollback process is via manual redeploy, so it's not as quick as the Immutable deployment.

***Rolling with additional batch***

* This policy avoids any reduced availability, at a cost of an even longer deployment time compared to the Rolling method.
* Suitable if you must maintain the same bandwidth throughout the deployment. However in case of deployment failure, the rollback process is via manual redeploy, so it's not as quick as the Immutable deployment

### Deployment part to Elastic Beanstalk is taking a very long time due to resolving dependencies

* AWS CodeBuild is a fully managed build service. There are no servers to provision and scale, or software to install, configure, and operate.
* A typical application build process includes phases like preparing the environment, updating the configuration, downloading dependencies, running unit tests, and finally, packaging the built artifact.
* Downloading dependencies is a critical phase in the build process. These dependent files can range in size from a few KBs to multiple MBs. Because most of the dependent files do not change frequently between builds, you can noticeably reduce your build time by caching dependencies.
* This will allow the code bundle to be deployed to Elastic Beanstalk to have both the dependencies and the code, hence speeding up the deployment time to Elastic Beanstalk

## API Gateway

### Stage variable

* A name-value pairs that you can define as configuration attributes associated with a deployment stage of a REST API. They act like environment variables and can be used in your API setup and mapping templates
* Using stage variables you can configure an API deployment stage to interact with different backend endpoints.

### Restrict access by using CORS

* Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain.
* When your API's resources receive requests from a domain other than the API's own domain and you want to restrict servicing these requests, you must disable cross-origin resource sharing (CORS) for selected methods on the resource.

### Resource policy

* Resource policies, which are JSON policy documents that you attach to an API to control whether a specified principal (typically an IAM user or role) can invoke the API.
* You can restrict IP address using this, the downside being, an IP address can be changed by the accessing user. So, this is not an optimal solution for the current use case.

### Authentication methods

* Deployment
  + After creating your API, you must deploy it to make it callable by your users. To deploy an API, you create an API deployment and associate it with a stage.
  + A stage is a logical reference to a lifecycle state of your API (for example, dev, prod, beta, v2). API stages are identified by the API ID and stage name.
  + Every time you update an API, you must redeploy the API to an existing stage or to a new stage. Updating an API includes modifying routes, methods, integrations, authorizers, and anything else other than stage settings.
* Permission
  + Access control access to Amazon API Gateway APIs is done with IAM permissions. To call a deployed API or to refresh the API caching, you must grant the API caller permissions to perform required
  + IAM actions supported by the API execution component of API Gateway. In the current scenario, developers are not able to view the changes which do not need permissions on "execution components". But on "management components" of API Gateway that help them to create, deploy, and manage an API. Hence, this statement is an incorrect option
* Lambda authorizer (formerly known as a custom authorizer)
  + An API Gateway feature that uses a Lambda function to control access to your API.

## Serverless Application Model (SAM)

Mandatory Param

## S3

### SSE-C

* Server-side encryption is about protecting data at rest. Server-side encryption encrypts only the object data, not object metadata. Using server-side encryption with customer-provided encryption keys (SSE-C) allows you to set your encryption keys.
* When you upload an object, Amazon S3 uses the encryption key you provide to apply AES-256 encryption to your data and removes the encryption key from memory.
* When you retrieve an object, you must provide the same encryption key as part of your request. Amazon S3 first verifies that the encryption key you provided matches and then decrypts the object before returning the object data to you.

Amazon S3 will reject any requests made over HTTP when using SSE-C. For security considerations, AWS recommends that you consider any key you send erroneously using HTTP to be compromised.

## Dynamo DB

### ProjectedExpression

* A projection expression is a string that identifies the attributes you want. To retrieve a single attribute, specify its name. For multiple attributes, the names must be comma-separated.

### Filter Expression

* If you need to further refine the Query results, you can optionally provide a filter expression. A filter expression determines which items within the Query results should be returned to you. All of the other results are discarded.
* A filter expression is applied after Query finishes, but before the results are returned.
* Therefore, a Query consumes the same amount of read capacity, regardless of whether a filter expression is present. A Query operation can retrieve a maximum of 1 MB of data. This limit applies before the filter expression is evaluated.

### Scan

* Scan operation in Amazon DynamoDB reads every item in a table or a secondary index. By default, a Scan operation returns all of the data attributes for every item in the table or index.
* You can also use the ProjectionExpression parameter so that Scan only returns some of the attributes, rather than all of them.

### DynamoDB Accelerator (DAX)

 If your traffic is read-heavy, consider using a caching service such as DynamoDB Accelerator (DAX). DAX is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10x performance improvement—from milliseconds to microseconds—even at millions of requests per second.

## Lambda

* To use the X-Ray SDK on Lambda, bundle it with your function code each time you create a new version

### Lambda Versions

* Versions are immutable and cannot be updated over time

### Alias

* A Lambda alias is like a pointer to a specific Lambda function version. Users can access the function version using the alias ARN.
* Lambda Aliases allow you to create a "mutable" Lambda version that points to whatever version you want in the backend. This allows you to have a "dev", "test", prod" Lambda alias that can remain stable over time.
* aws lamda custom interceptor / authorizer

### Provisioned concurrency

* Due to a spike in traffic, when Lambda functions scale, this causes the portion of requests that are served by new instances to have higher latency than the rest. To enable your function to scale without fluctuations in latency, use provisioned concurrency.
* By allocating provisioned concurrency before an increase in invocations, you can ensure that all requests are served by initialized instances with very low latency.
* You can configure Application Auto Scaling to manage provisioned concurrency on a schedule or based on utilization. Use scheduled scaling to increase provisioned concurrency in anticipation of peak traffic.
* To increase provisioned concurrency automatically as needed, use the Application Auto Scaling API to register a target and create a scaling policy.

### Reserved concurrency

* To ensure that a function can always reach a certain level of concurrency, you can configure the function with reserved concurrency. When a function has reserved concurrency, no other function can use that concurrency.
* More importantly, reserved concurrency also limits the maximum concurrency for the function, and applies to the function as a whole, including versions and aliases.
* You cannot configure Application Auto Scaling to manage Lambda reserved concurrency on a schedule.

### Environment variables

* The Lambda runtime makes environment variables available to your code and sets additional environment variables that contain information about the function and invocation request.
* The total size of all environment variables doesn't exceed 4 KB. There is no limit defined on the number of variables that can be used.

## ECS

### Terminate a container instance while it is in the STOPPED state

* If you terminate a container instance while it is in the STOPPED state, that container instance isn't automatically removed from the cluster.
* You will need to deregister your container instance in the STOPPED state by using the Amazon ECS console or AWS Command Line Interface.
* Once deregistered, the container instance will no longer appear as a resource in your Amazon ECS cluster.

### Terminate a container instance in the RUNNING state

* The container instance is automatically removed, or deregistered, from the cluster.

## ELB

ALB access logs

Can we have ELB without ASG

* Amazon EC2 Auto Scaling works with Application Load Balancers and Network Load Balancers including their health check feature
* EC2 Auto Scaling groups are regional constructs. They can span Availability Zones, but not AWS regions
* **The ping path field of the Load Balancer** is configured incorrectly - Ping path is a health check configuration field of Elastic Load Balancer. If the ping path is configured wrong, ELB will not be able to reach the instance and hence will consider the instance unhealthy. However, this would then apply to all instances, not just once instance.

### X-Forwarded-For

* The X-Forwarded-For request header helps you identify the IP address of a client when you use an HTTP or HTTPS load balancer. Because load balancers intercept traffic between clients and servers, your server access logs contain only the IP address of the load balancer. To see the IP address of the client, use the X-Forwarded-For request header.

### X-Forwarded-Proto

* The X-Forwarded-Proto request header helps you identify the protocol (HTTP or HTTPS) that a client used to connect to your load balancer. Your server access logs contain only the protocol used between the server and the load balancer;
* They contain no information about the protocol used between the client and the load balancer. To determine the protocol used between the client and the load balancer, use the X-Forwarded-Proto request header.

### X-Forwarded-Port

The X-Forwarded-Port request header helps you identify the destination port that the client used to connect to the load balancer.

## ASG

* By default, the health check configuration of your Auto Scaling group is set as an EC2 type that performs a status check of EC2 instances.
* To automate the replacement of unhealthy EC2 instances, you must change the health check type of your instance's Auto Scaling group from EC2 to ELB by using a configuration file.

Target tracking scalling policy

clouldtrail logs

alb request tracing

## Kinesis:

KPL vs KCL

## CloudFormation

### **!GetAtt**

### The Fn::GetAtt intrinsic function returns the value of an attribute from a resource in the template

cloudfront usage plans

aws billing usage plans

AWS system manager / secret manager / certificate manager /ssm parameter

access control list

SSH keypairs accross regions

AWS database engines can be configured with IAM Database Authentication?

## Auto Scaling group:

* Auto Scaling groups cannot span across multiple Regions
* Auto scaling groups can span across the availability Zones of a Region.
* When one Availability Zone becomes unhealthy or unavailable, Auto Scaling launches new instances in an unaffected Availability Zone.
* When the unhealthy Availability Zone returns to a healthy state, Auto Scaling automatically redistributes the application instances evenly across all of the designated Availability Zones.
* An Auto Scaling group can contain EC2 instances in one or more Availability Zones within the same Region

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Cognito User pools:

* A user pool is a user directory in Amazon Cognito. With a user pool, your users can sign in to your web or mobile app through Amazon Cognito, or federate through a third-party identity provider (IdP).
* Cognito User Pools cannot be used to obtain temporary AWS credentials to access AWS services, such as Amazon S3 and DynamoDB

## Cognito Identiy pools:

* Amazon Cognito identity pools (federated identities) enable you to create unique identities for your users and federate them with identity providers.
* With an identity pool, you can obtain temporary, limited-privilege AWS credentials to access other AWS services. Amazon Cognito identity pools support the following identity providers:
* Public providers: Login with Amazon (Identity Pools), Facebook (Identity Pools), Google (Identity Pools), Sign in with Apple (Identity Pools).
* Amazon Cognito User Pools, Open ID Connect Providers (Identity Pools),SAML Identity Providers (Identity Pools), Developer Authenticated Identities (Identity Pools)

## Amazon Kinesis

* Data Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift,
* Amazon Elasticsearch Service (Amazon ES), and Splunk. With Kinesis Data Firehose, you don't need to write applications or manage resources.
* You configure your data producers to send data to Kinesis Data Firehose, and it automatically delivers the data to the destination that you specified.

Amazon ElastiCache with Amazon S3 as backup -

* Amazon ElastiCache is a fully managed in-memory data store, compatible with Redis or Memcached. ElastiCache is NOT a supported destination for Amazon Kinesis Data Firehose.
* Amazon Elastic Search is a supported destination type for Kinesis Firehose. Streaming data is delivered to your Amazon ES cluster, and can optionally be backed up to your S3 bucket concurrently.
* Amazon S3 destinations, streaming data is delivered to your S3 bucket. If data transformation is enabled, you can optionally back up source data to another Amazon S3 bucket.
* Amazon Redshift destinations, streaming data is delivered to your S3 bucket first. Kinesis Data Firehose then issues an Amazon Redshift COPY command to load data from your S3 bucket to your Amazon Redshift cluster. If data transformation is enabled, you can optionally back up source data to another Amazon S3 bucket.

## AWS X-Ray

* AWS X-Ray helps developers analyze and debug production, distributed applications, such as those built using a microservices architecture.
* With X-Ray, you can understand how your application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors. X-Ray provides an end-to-end view of requests as they travel through your application, and shows a map of your application’s underlying components.
* To use the X-Ray SDK on Lambda, bundle it with your function code each time you create a new version.

## EC2

### Zonal Reserved Instance

* When you purchase a Reserved Instance for a specific Availability Zone, it's referred to as a Zonal Reserved Instance. Zonal Reserved Instances provide capacity reservations as well as discounts.
* A zonal Reserved Instance provides a capacity reservation in the specified Availability Zone. Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage Capacity Reservations independently from the billing discounts offered by Savings Plans or regional Reserved Instances.

### Regional Reserved Instances

* When you purchase a Reserved Instance for a Region, it's referred to as a regional Reserved Instance. A regional Reserved Instance does not provide a capacity reservation.
* Enable detailed monitoring for an existing instance
  + aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
* Enable detailed monitoring when launching an instance from AWS CLI.
  + aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true

### ESB

* When you create an EBS volume, it is automatically replicated within its Availability Zone to prevent data loss due to the failure of any single hardware component.
* You can attach an EBS volume to an EC2 instance in the same Availability Zone. EBS volumes are AZ locked
* Encryption by default is a Region-specific setting. If you enable it for a Region, you cannot disable it for individual volumes or snapshots in that Region.

## IAM

* IAM database authentication works with MySQL and PostgreSQL.

## SQS

* There are no message limits for storing in SQS, but 'in-flight messages' do have limits. Make sure to delete messages after you have processed them.
* There can be a maximum of approximately 120,000 inflight messages (received from a queue by a consumer, but not yet deleted from the queue).

## Amazon ElastiCache

* Amazon ElastiCache is an ideal front-end for data stores such as Amazon RDS, providing a high-performance middle tier for applications with extremely high request rates and/or low latency requirements.
* The best part of caching is that it’s minimally invasive to implement and by doing so, your application performance regarding both scale and speed is dramatically improved

## AWS CloudTrail

* AWS CloudTrail increases visibility into your user and resource activity by recording AWS Management Console actions and API calls.
* You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.
* With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure.
* CloudTrail provides an event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command-line tools, and other AWS services.

## AWS X-Ray

* AWS X-Ray helps developers analyze and debug production, distributed applications, such as those built using a microservices architecture.
* With X-Ray, you can understand how your application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors.
* X-Ray is a very important tool in troubleshooting but is not useful in logging user activity.

## Amazon Inspector

* Amazon Inspector - Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS.
* Amazon Inspector security assessments help you check for unintended network accessibility of your Amazon EC2 instances and for vulnerabilities on those EC2 instances. This does not log User activity at the account level.

## ElastiCache

* To provide a shared data storage for sessions that can be accessed from any individual web server, you can abstract the HTTP sessions from the web servers themselves.
* A common solution for this is to leverage an ElastiCache service offering which is an In-Memory Key/Value store such as Redis and Memcached.
* Amazon ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads (such as social networking, gaming, media sharing, and Q&A portals) or compute-intensive workloads (such as a recommendation engine) by allowing you to store the objects that are often read in the cache.

## RDS

* RDS automatic backups You can enable automatic backups but as of 2020, the retention period is 0 to 35 days.

## AWS Systems Manager Parameter Store

* Systems Manager Parameter Store, you can create SecureString parameters, which are parameters that have a plaintext parameter name and an encrypted parameter value.
* Parameter Store uses AWS KMS to encrypt and decrypt the parameter values of Secure String parameters. Also, if you are using customer-managed CMKs, you can use IAM policies and key policies to manage to encrypt and decrypt permissions. To retrieve the decrypted value you only need to do one API call.

