101 Important concepts of

AWS Developer Certification- Associate.

Compiled by: Bilal Majeed.

Table of Contents

MAIN CONCEPTS:.. 3

OTHER AWS SERVICES AND CONCEPTS YOU NEED TO KNOW: 18

Main concepts:

1. **Inline Policy** When you use an inline policy, the permissions in the policy cannot be inadvertently attached to the wrong principal entity.

2. ***DynamoDB*** A primary key can either be a single-attribute partition key or a composite partition-sort key.

3. With *standard queues, SQS* will deliver each message at least once, but cannot guarantee the delivery order. Because each message may be delivered more than once, your application should be idempotent by design.

4. ***RDS*** is not considered to be serverless because it runs on EC2 instances. The following AWS technologies are serverless: ***DynamoDB, API Gateway, SNS, Lambda, Kinesis, and S3.***

5. AWS recommends that you use multipart upload for any files that are greater than 100 MB in size in S3, instead of uploading the object in a single operation.

6. The creation of a **secondary index** will allow you to sort by *userId*. A secondary index is a data structure that contains a subset of attributes from a table, along with an alternate key to support query operations.

7. Messages will be retained in **SQS** queues for up to 14 days.

8. In order to enable *encryption* at *rest* using EC2 and Elastic Block Store, you must configure encryption when creating the EBS volume.

9. **Instance profile** role which is associated with the EC2 instance. An instance profile is a container for an IAM role that you can use to pass role information to an Amazon EC2 instance when the instance starts.

10. **EBS**: Elastic Block Storage:

    It's a virtual disk which let us store data to run an OS, database or install an application. It's for a mission critical workload, it's highly available and automatically replicate within a single AZ. Its very scalable and no down-time.

    GP3: 3k IOPS base line performance max 16k), great for high performance at a low cost.

    io1 - old and legacy

    io2 - new and better performance. - super high performance. like when > 16k IOPS.

    SC1 - Cold HDD**:**

    12 MB per sec for a TB is the baseline throughput (max 80MB per sec for a TB)

    Can't be a boot volume.

11. *Cognito*: A user authenticates with Facebook first. They are then given an ID token by Facebook. An API call, AssumeRoleWithWebIdentity is then used in conjunction with the ID token. A user is then granted temporary security credentials.

12. Web identity federation removes the need for creating individual IAM users. Instead, users can sign in to an identity provider(Google, Facebook) and then obtain temporary security credentials from the AWS Security Token Service.

13. The `STS:AssumeRole` API call returns a set of temporary security credentials which can be used to access AWS resources, including those in a different account. This API can be called by an IAM user or IAM role with existing temporary security credentials.

14. *AWS Secrets Manager* is the current AWS recommended way of securely provide database credentials to Lambda functions. AWS Secrets Manager aids in the managing and rotating the RDS database passwords.

15. *Elastic Beanstalk* has configuration options for before, during, and after environment creation. These files are loaded from configuration files in the. ebextensions folder at the root of the application source bundle.

16. Correct run order for lifecycle hooks for an in-place deployment using *CodeDeploy: ApplicationStop ,BeforeInstall, AfterInstall, ApplicationStart*

17. **VPC**:

- Use Private link to allows EBS app connection for other applications
- define VPC configuration in the aws:ec2:vpc namespace
- VPC endpoint allow lambda to communicate between SQS (outside VPC) and lambda (inside VPC)
- both NAT and Internet gateway is required for public subnet to access internet.

18. EC2 instance that needs to access both the private and public IP address of that instance, GET to access the instance metadata.

19. **SQS**:

- It's a pull-based system. (Send and received message simultaneously)
- Messages kept for 1 minute to 14 days (4 days is a default value)
- Allow us to decouple the components so apps can work independents.
- Supported JSON, Xml and unformatted text
- 256KB max size. it will be guaranteed for at least once delivery
- Batches: send, receive or delete up to 10 messages at once. Can be more cost effective when using batches.
- Encryption using server side (SSE) with KMS. auto decrypt message when send to authorised consumer.
- Dead letter queues (DLQ): When consumer failed to process the message then it will move to DLQ. it should be same type of the SQS. Once remediated (fixed) the issue then message can be move back the main SQS.
- Standard SQS:
  - Same order as they sent but not always.
  - at least once, maybe more than one delivered. Best efforts,
  - Default SQS type.
- FIFO →
  - Restrict perceived, 300 TPS (300 transactions per seconds), with batch this can be 3000 per sec.
  - No duplicates,
  - Maintain an order, Great for banking application.
- *Visibility timeout:*
  - Amount of the time the message is available once they picked up.
  - 30 seconds is the default time, and max it can be 12 hours. message remains hidden
- Short polling: Return response immediately, a lot of empty response even when SQS is empty. Will pay for this response even empty.
- Long polling → consumer periodically pull message, no response until message exists or long poll times out (wait for 20 seconds). Save money. and it's preferable then the short polling.
- SQS Delay and large messages:
  - Postpone delivery of new messages

- default is 0 seconds and the max is 15 seconds
- If large Message (over 256KB to 2FB) use S3,
- Use SQS Extended Client Library for java.
- Can't use AWS CLI, Management console, API or any of the other AWS SDK

20. *A composite* key with UserID as the partition key and the timestamp as the sort key would be best, as this will enable you to find all posts by a particular user based on their UserID, and show the results in chronological order.

21. SNS:

    Pub/Sub model and it could use SMS, HTTP, SQS and email. Can push a message to multiple endpoints

    Trigger a lambda function.

    Consumer must subscribe to the topic

22. ***Kinesis*** gives you the ability to consume records according to a sequence number applied when data is written to the Kinesis shard.

23. ***DAX*** is the recommended approach to reducing response times for read-intensive applications, applications which read a small number of items frequently and also applications which perform repeated reads against a large set of data.

24. ***DynamoDB Read***: *Your application needs 100 strongly consistent reads on items that are 9 KB in size every second. How many units of read capacity units should you provision? DynamoDB read* —> 300 is the correct answer. The calculation is as follows: 9 KB rounds up to 12 KB. 12 KB / 4 KB = 3 strongly consistent read capacity units each. 3 * 100 = 300 strongly consistent read capacity units.

25. ***DynamoDB writing***: Writing to the database every six seconds, there are 10 writes/minute/vehicle. There are 60 vehicles in the fleet, so there are 600 writes/minute overall. 600/60 seconds = 10 writes/second.

26. When you instrument your application, the X-Ray SDK records information about incoming and outgoing requests, the AWS resources used, and the application itself. You can add other information to the segment document as annotations and metadata. Annotations are simple

key-value pairs that are indexed for use with filter expressions. Use annotations to record data that you want to use to group traces in the console.

27. You are exceeding an individual partition's throughput capacity, even if you're not exceeding the overall table throughput capacity. DynamoDB distributes capacity evenly across all available partitions. If a given partition is consuming more than its share of throughput, this error will be raised.

28. **IAM Policies**:

    - Managed Policies - created and maintain by AWS and mainly used for most common- Can be assigned to multiple users or groups or roles.
    - Customer managed: We create ourself (copy from the existing policy) - Recommended when managed policy is not a good fit.
    - Inline policy: Can't attach to multiple - its a one specific user, group or a role. One to one relationship

29. ***API Gateway:***

    - Import using definition file (open API or swagger)
    - Transform XML response to JSON
    - Configure as a SOAP passthrough or its simply passthrough or convert to JSON
    - Caching:
        - reduce the number of the calls to backend or origin
        - improve performance and reduce the latency
        - TTL (time to live) - default time is 5 minutes or 300 seconds
        - Throttling: too many requests, default limit is 10k per seconds, or 5k concurrent request per seconds. 10k request in the first ms, then 5k request services.
        - 429 error - too many requests volume - when it reaches the steady-state request rate and bursting limit.
        - API gateway throttles your request to protect backend API service. when requests start to throttled then it API gateway throw 429 code.
        - We can use query string for cache key. for a separate cache response.

30. Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud and are the lowest cost option for on-demand and short-term capacity requirements. As the HPC cluster nodes store the data in S3 the termination of Spot instances will not impact the data processing. Both on-demand and dedicated instances are more expensive than Spot instances, and reserved instances are for long running applications (1 to 3 years) so are not suitable for this HPC cluster scenario.

31. ***AWS Systems Manager Parameter Store*** provides secure, hierarchical storage for configuration data management and secrets management. You can store data such as passwords, database strings, and license codes as parameter values.

32. *To encrypt RDS instance in rest:* Create a snapshot of the EC2 volume. Then create a copy of the snapshot, checking the box to enable encryption. Create an AMI of the copied snapshot and then redeploy the EC2 instance using the encrypted AMI. Delete the old EC2 instance.

33. The Transform section specifies one or more macros that AWS CloudFormation uses to process your template. The Transform section builds on the simple, declarative language of AWS CloudFormation with a powerful macro system. The declaration Transform: AWS::Serverless-2016-10-31 is required for AWS SAM template files. Transforms is used to reference code located in S3 and also for specifying the use of the Serverless Application Model (SAM) for Lambda deployments.

34. ***CodeArtifact*** is a fully managed artifact repository service that makes it easy for organizations of any size to securely store, publish, and share software packages used in their software development process. This includes everything needed to build your application, including libraries, deployable packages, compiled applications, and documentation relating to your application.

35. ***Continuous Integration (CI)*** *i*s the practice of merging all developers' working copies to a shared repository frequently, preferably several times a day. One of the key benefits of integrating regularly is that you can detect errors quickly and locate them more easily. As each change introduced is typically small, pinpointing the specific change that introduced a defect can be done quickly.

36. *Outputs* is used to output user-defined data relating to the resources you have built. It can also use as input for another ***CloudFormation stack.***

37. ***CodeDeploy*** is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

38. You can add a connection between a CodeArtifact and an external, public repository, so that when developers request a package from the CodeArtifact repository that's not already present in the repository, the package can be fetched from the external connection. This makes it possible to consume open-source dependencies used by your application.

39. AppSpec files on an EC2/on-premises compute platform must be a YAML-formatted file named appspec.yml and it must be placed in the root of the directory structure of an application's source code. Otherwise, deployments fail.

40. The hooks section specifies the location (as relative paths starting from the root of the revision bundle) of the scripts to run during each phase of the deployment. Each phase of a deployment is called a deployment lifecycle event.

41. ***Elastic Cache:*** In order to address scalability and 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 to this is to leverage an in-memory key/value store such as Redis and Memcached, and in AWS, the service to use is ElastiCache. And it could be used for storing session state.

42. **Elastic Cache** is and in-memory cache designed to improve read performance for read heavy databases.

    Memcached: in-memory key value data store, object cache is primary goal, keep things simple, don't need multi availability zones, don't need to support advance data types or sortings.

    Redis: in-memory key value data store, performing data sorting and linking such as gaming leader boards, advance datatypes such as list and hashes, need persistent data persistence and multi availability zones.

43. ***x-amz-server-side-encryption*** to request server-side encryption using the object creation REST APIs, provide the ***x-amz-server-side-encryption*** request header.

44. Using the AWS CLI to list all the files in an S3 bucket containing thousands of files can cause your API call to exceed the maximum allowed time for the AWS CLI, and generate a "timed out" error. To avoid this, you can use the --page-size option to specify that the AWS CLI request a smaller number of items from each call to the AWS service.

45. All **DynamoDB** tables are encrypted at rest with an AWS-owned key by default.

46. The policy must be attached to the ECS task execution role to allow the application running in the container to access SQS. A task execution role grants the Amazon ECS container (and Fargate agents) permission to make AWS API calls on your behalf.

47. *CodeBuild* uses the BuildSpec file as a specification of build commands and settings. “secrets-manager” syntax can be used to retrieve API keys stored in AWS Secrets Manager.

48. Create your own *AWS Elastic Beanstalk platform using Packer*. Use this platform for your application.AWS Elastic Beanstalk supports custom platforms, which lets you develop an entirely new platform from scratch and customize the operating system, additional software, and scripts that Elastic Beanstalk runs on platform instances. This flexibility enables you to build a platform for an application that uses a language or other infrastructure software for which Elastic Beanstalk doesn't provide a managed platform. In addition, with custom platforms, you have the advantage of using an automated, scripted method to create and maintain your customization.

49. AWS CloudFormation provides the cfn-init Python helper script that can be used to enable/disable and start/stop services. AWS Cloud Formation provides the cfn-init Python helper script that can be used to fetch and parse metadata from Cloud-Formation.

50. The application requirement states support for path-based routing. This means that we must use an Application Load Balancer, as a Network Load Balancer does not have this feature. It is best practice to deploy the SSL certificates on the Load Balancer. This implements SSL termination on the load balancer and off-loads this task from the application, thus reducing the load on EC2 instances. Additionally, it removes the requirement of distributing the certificate to all target EC2 instances.

51. ***Platform hooks*** can be used to define custom scripts or executables that run at various stages when EC2 instances are provisioned. They are supported for Amazon Linux 2 environments.

52. ***A projection expression*** is an Amazon ***DynamoDB*** string that identifies the attributes that you want and can be used to limit the attributes returned by operations such as GetItem, Query, or Scan. Thus, this can be used to reduce the size of the payload returned by a read operation.

53. **CloudWatch**: If you set the alarm on a high-resolution metric, you can specify a high-resolution alarm with a period of 10 seconds or 30 seconds, or you can set a regular alarm with a period of any multiple of 60 seconds

54. You can create different environment variables in the Lambda function that can be used to point to the different services.

55. Each **AWS SDK** implements automatic retry logic. If you're not using an AWS SDK, you should retry original requests that receive server (5xx) or throttling errors. However, client errors (4xx) indicate that you need to revise the request to correct the problem before trying again. In addition to simple retries, each AWS SDK implements an exponential backoff algorithm for better flow control. The idea behind 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.

56. **Visibility Timeout** is a period during which Amazon ***SQS*** prevents other consumers from receiving and processing the message. In the above requirement, the application takes 45 secs to process each message. So, Visibility Timeout should be greater than 45 seconds. Hence, setting visibility timeout as 60 seconds is the correct option.

57. Batching Message Actions To reduce costs, batch your message actions: • To send, receive, and delete messages, and to change the message visibility timeout for multiple messages with a single action, use the Amazon SQS batch API actions. • To combine client-side buffering with request batching, use long polling together with the buffered asynchronous client included with the AWS SDK for Java.

58. Amazon Cognito supports authentication with identity providers through Security Assertion Markup Language 2.0 (SAML 2.0). You can use an identity provider that supports SAML with Amazon Cognito to provide a simple onboarding flow for your users. Your SAML-supporting identity provider specifies the IAM roles that can be assumed by your users so that different users can be granted different sets of permissions.

59. Immutable updates are an alternative to rolling updates where a temporary Auto Scaling group is launched outside of your environment with a separate set of instances running on the new configuration, which are placed behind your environment's load balancer. Old and new instances both serve traffic until the new instances pass health checks. The new instances are then moved into your environment's Auto Scaling group and the temporary group and old instances are terminated.

60. When an ***ECS*** container is stopped, the Container instance status remains Active, but the container Agent status changes to FALSE after a few minutes.

61. Global Secondary Index does not support Consistent read. It only supports Eventual Read. For other tables, Query with Consistent Read will provide the latest results without scanning the whole table.

62. We recommend that you use the following pattern to encrypt data locally in your application. Use this operation (GenerateDataKey) to get a data encryption key. Use the plaintext data encryption key (returned in the Plaintext field of the response) to encrypt data locally, then erase the plaintext data key from memory. Store the encrypted data key (returned in the Cipher-text-Blob field of the response) alongside the locally encrypted data.

63. For strongly consistent read request from an application, DAX Cluster pass all request to DynamoDB & does not cache for these requests. Because it only works with the eventually consistent.

64. S3 is the only service supported by AWS for receiving ALB access logs.

65. To send logs to Amazon Kinesis Data Firehose, you send logs from your web ACL to an Amazon Kinesis Data Firehose with a configured storage destination. After you enable logging, AWS WAF delivers logs to your storage destination through the HTTPS endpoint of Kinesis Data Firehose.

66. *VPC Flow Logs* is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs or Amazon S3. After you create a flow log, you can retrieve and view its data in the chosen destination.

67. Application Load Balancers offer several features that make them attractive for use with Amazon ECS services. Application Load Balancers allow containers to use dynamic host port mapping (so that multiple tasks from the same service are allowed per container instance). Application Load Balancers support path-based routing and priority rules (so that multiple services can use the same listener port on a single Application Load Balancer). • Network Load Balancers do support dynamic host port mapping.

68. ***X-Ray*** uses trace data from the AWS resources that power your cloud applications to generate a detailed service graph. The service graph shows the client, your front-end service, and backend services that your front-end service calls to process requests and persist data. Use the service graph to identify bottlenecks, latency spikes, and other issues to solve to improve the performance of your applications.

69. ***IAM permission simulator*** is the name of the AWS tool that will enable you to identify which specific statement in a policy result in allowing or denying access to a particular resource or action.

70. The Handler property specifies the Lambda function's entry point. For example, if the Lambda function was wri tten in Python, and Handler was set to lambda_function.lambda_handler, execution would begin with the lambda_handler function, contained within the lambda_function.py file.

71. Lambda to access resources in a private VPC.

    - Provide VPC config information to the Lambda function. Private subnet ID, and security group ID.
    - Lambda uses the VPC Information. Lambda configures an ENI, using an IP from the private subnet CIDR range.
    - The security group then allows your function to access resources in VPC.

72. DelaySeconds in SQS allows the delivery of a message to be delayed for between 0 (default) and 900 seconds before the message becomes visible to consumers of the queue for the first time.

73. A canary deploy allows us to gain confidence in an application update by initially just releasing it to a subsection of users. Once we are satisfied that the update is working as expected, the update is then rolled out to the remaining users. The concept of a canary deployment is covered in the AWS Well-Architected Framework, and is a feature of API Gateway. It can also be performed manually using Route53 Weighted Records, or using an Application Load Balancer with a Forward Action and Weighted Target Groups.

74. Boolean is not a supported data type for a ***DynamoDB*** sort key attribute.

75. ***AWS Fargate*** is a compute engine for *ECS* that allows you to run containers without having to manage servers or clusters.

76. You need the *X-Ray SDK* and the X-Ray daemon on your EC2 instances and on-premises systems. You then need to instrument your application to send the required data to X-Ray.

77. AWS X-Ray is a service that collects data about requests that your application serves, and provides tools you can use to view, filter, and gain insights into that data to identify issues and opportunities for optimisation. For any traced request to your application, you can see detailed information not only about the request and response, but also about calls that your application makes to downstream AWS resources, micro-services, databases, and HTTP web APIs. When you instrument your application, the X-Ray SDK records information about incoming and outgoing requests, the AWS resources used, and the application itself. You can add other information to the segment document as annotations and metadata. Annotations are simple key-value pairs that are indexed for use with filter expressions. Use annotations to record data that you want to use to group traces in the console.

78. ***X-Ray*** Annotations needs to be enabled to get Filtered output for User A from all other traces

79. Run the start build AWS CLI command with buildspecOverride property set to the new buildspec.yml file. If developer wants to override a built.

80. ***Envelope key***: First, the data is encrypted using a plaintext Data Key. The Data Key is then further encrypted using a plaintext Master Key.

81. ***CodeStar*** service from AWS could help you handle all aspects of development and deployment like codeBuild, codePipeline, codeDeploy.

82. **Weighted routing** lets you associate multiple resources with a single domain name (example.com) or subdomain name (acme.example.com) and choose how much traffic is routed to each resource. This can be useful for various purposes, including load balancing and testing new versions of software.

83. Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's `IP` address, `latencies`, request `paths`, and server responses. You can use these access logs to `analyze traffic patterns` and `troubleshoot issues`.

84. To retag docker images in ***Amazon ECR***, it is not required to pull or push these images again to the ECR depository. The option of the put-image command can be used to retag the existing image in the repository. This command is useful for retagging large images as this will save network bandwidth by avoiding retrieving images.

85. ***DynamoDB*** is an alternative solution that can be used for the storage of session management. The latency of access to data is less. Hence, this can be used as a data store for session management.

86. Fine-Grained Access Control with IAM. IAM condition parameter, "dynamodb :_LeadingKeys" allows users toaccess only the items where the partition key value matches their User_ID.

87. ***Amazon kinesis*** to ingest, analyze, and persist their streaming data. One of the easiest ways to gain real-time insights into your streaming data is to use kinesis analytics. It enables you to query the data in your stream or build entire streaming applications using SQL. Customers use Kinesis Analytics for things like filtering, aggregation, and anomaly detection.

88. Three Important service of ***Amazon kinesis:***

STREAMS → Capture and store streaming video and data for real-time processing and analysis. Consumer applications process and analyse the data in real time.

FIREHOSE →Capture, transform and load data continuously into AWS data stores (or other service providers such as Datadog or Splunk). Existing BI applications and tools can be used for near real-time analysis of the stored data

ANALYTICS → Real-time analytics using standard SQL on data received by Kinesis Data Streams and Kinesis Data Firehose. Stores the processed data in AWS data stores, e.g. S3, Redshift, Elasticsearch.

89. When multiple ***Amazon SQS*** queues are subscribed to a single topic within an Amazon SNS, each queue will receive an identical message. This is useful for parallel independent processing of messaging.

90. Using Rolling with additional batch deployment, a new batch of the Amazon EC2 instance is launched before taking a batch of instances out of service for deploying a new version. Once all Amazon EC2 instances are upgraded to a new version of the application, this additional batch of Amazon EC2 instances is terminated. This will ensure full capacity during deployment. Blue-green doesn't work on on-premies but in-place does.

91. You can enable ***DynamoDB streams*** to record both old and new image and send it to a Lambda function which can store the changes on S3.

92. While enabling ***DynamoDB TTL***, you can provide any name of your choice to use as a key attribute name to be used for storing timestamp value.

93. DynamoDBCrudPolicy will give, create, read, update and delete permissions to a DynamoDB table which is tighter and more secure inline with the best practice of least privilege. It is also managed by AWS which would make it AWS's responsibility to maintain the policy.

94. When you send HTTP requests to AWS, you sign the requests so that AWS can identify who sent them. You sign requests with your AWS access key*,* which consists of an access key ID and secret access key. Some requests do not need to be signed, such as anonymous requests to Amazon Simple Storage Service (Amazon S3) and some API operations in AWS Security Token Service (AWS STS) such as

95. Parameters within an AWS *CloudFormation* Template allow to input custom values while creating or updating a stack. Supported data types for Parameters are as follows. String, Number, List, CommaDelimitedList, AWS-Specific Parameter Types, SSM parameter Types.

96. You need to publish a custom metric to handle application-specific events. If you want to collect metrics at 10-second intervals, you need to use high-resolution metrics.

97. BatchGetItem API allows you to pass multiple Partition Key values in a single request.

98. The optional `Mappings` section matches a key to a corresponding set of named values. For example, if you want to set values based on a region, you can create a mapping that uses the region name as a key and contains the values you want to specify for each specific region. It cannot be used to reference values exported by another template.

99. You can use the intrinsic function **`Fn::ImportValue`** to import CloudFormation stack values that have been exported within the same region. also, can use to import values from `another` CloudFormation `template.` The condition function Fn::GetAtt to return the value of the attribute. **Nested stacks** are stacks created as part of other stacks. You create a nested stack within another stack by using the AWS::CloudFormation::Stack resource.

100. Explicitly denying requests that are identified as "aws:SecureTransport": "false" would deny requests that are using HTTP and are unencrypted. Explicitly denying requests that are identified as "s3:x-amz-server-side-encryption": "false” would deny requests that do not use server-side encryption.

101. Amazon *Cognito* Sync is an AWS service and client library that enables cross-device syncing of application-related user data. You can use it to synchronize user profile data across mobile devices and web applications. The client libraries cache data locally so your app can read and write data regardless of device connectivity status. When the device is online, you can synchronize data, and if you set up push sync, notify other devices immediately that an update is available.

Other AWS services and concepts you need to know:

data stream consumer: A consumer is an application that processes all data from a Kinesis data stream, allowing multiple consumers to read data from the same stream in parallel, without contending for read throughput with other consumers

annotations: it is a key value pair used in X-Ray for filtering purposes.

NAT endpoints: NAT Gateway is a highly available AWS managed service that makes it easy to connect to the Internet from instances within a private subnet in an VPC. and also, other AWS services

Amazon Aurora is a relational database management system (RDBMS) built for the cloud with full MySQL and PostgreSQL compatibility. Aurora gives you the performance and availability of commercial-grade databases at one-tenth the cost.

DynamoDB transactions feature provides ability to group multiple items into a single atomic transaction and perform all-or-nothing coordinated operations. This can be done programmatically using the TransactWriteItems operation.

S3:

- Object based, key pair value and name should be unique,
- versionID and metadata (content type or last modified date) for files
- S3 secure by default, policies applied at a bucket level, ACL applied on object level.
- Encryption In transit - SSL/TLS and HTTPS
- Encryption at rest:
  - Server-side encryption - 3 types
    - SSE-S3 (AES 256-bit) - advances uses and keys managed within S3
    - SSE-KMS - keys managed by KMS services
    - SSE-C - manage keys by customer
- Client-side encryption: first encrypt the file and then upload to S3
- We can enforce encryption within a bucket policy like on all PUT request, ***x-amz-server-side-encryption*** param should be in the request header. Deny request that don't have a ***aws:secureTranport*** to enforce the use of HTTPS or SSL
- Max file size 5TB and 0kb is the minimum, and unlimited storage, all bucket should be unique as its cross region
- Using Put method call (return 200 status code for a successful upload) the max allowed size is 5GB (in the case of PUT operation). File larger than 100 MB, recommended to use multi part upload capability

Cloud Front

- Edge location -
  - This is where the content will be cached
  - for both read and write
- Origin: origin for all the files, s3 or any custom server or ec2

- Distribution: - name given to the origin and the settings
- S3 transfer accelerator - utilise to reduce the latency for S3 upload
- TTL - default is for one day. we can clear cached sooner for a charge
- Get, Head and Option (read only)

Private S3 files or Secure URL access using following

- CF Signed cookies
- CF Origin access identity
- CF Signed URLs

Dynamo DB

- Low latency noSQL.
- Three models
    - Eventually consistent
    - Strongly consistent
    - DynamoDB Transactions (ACID) all or nothing, no partial transactions
- Data store in tables, items (row) and attributes (column)
- Primary Key
    - Partition key → unique identifier.
    - Composite key → partition key + sort key)
        - e.g. customerID + sort key (or timestamp)
- Fine grained access with IAM, use dynamoDB:LeadingKey allow where matching with user id
- Secondary index - fast response on specific columns
- Local secondary index -
    - Same partition key and different sort key
    - Create at the time of table creation,
- Global secondary key;
    - Different partition key and different sort key
    - create or delete anytime
- 20 global secondary indexes per table.
- Param to refine
- Set a page size - and segregate from mission critical data
- Scan
    - Done on every item on the table, and return all the data from the table
    - Use ProjectionExpression param to refine the results (i.e filter)
    - Parallel scan to improve response. Try parallel scans used when
        - Table size is 20+ GB
        - Table read throughput is not fully used
        - Sequential scan ops are too slow
- Query (its preferable for most cases)
    - Find using the primary key name and distinct value to search
    - Must provide the name of the partition key, (optional) provides sort key attributes to filter the results.
    - Use limit param to allow maximum number of items to return
    - Use ScannedCount (before filter) and Count (after filter)

    - ScanIndexForward param - set to false to make it reverse (cant use for scan) only for query
- Try to design the table so that you can use Query, Get or BatchGetItem APIs
- We can define a maximum of 5 local secondary indexes.
- Max 20 global secondary index per table.
- TTL (time to live):
    - Define an expiry time and then it will expire
    - Removing old data like logs and session data
    - Help save money to reduce the size
- Streams
    - Audit trail and storing all the transactions
    - Encrypted data - only for 24 hours
    - Sequence of modifications of tables - time ordered seq of items level
    - lambda event
    - source - can be use as a event
- Two type of pricing
    - On demand pricing model → when unpredictable traffic, pay per model
    - Use provisioned capacity → when read and write capacity requirement can be forecasted. application traffic is consistent or inc gradually.
- Provisioned capacity Throughput measured:
    - Write capacity unit - 1 x 1kb per sec
    - Strongly consistent reads - 1 x 4kb per sec
    - Eventually consistent reads - 2 x 4kb per sec (default)
    - Divide total kb by 4 and the round it up to floor.

Projection Expression param:

When using Query, or Scan, DynamoDB returns all of the item attributes by default. To get just some, rather than all of the attributes, use a Projection Expression.

Elastic Container Service

- Fully managed container orchestration service for both docker or Windows Containers (only for windows)
- No need to install configure and manage, good integration with other AWS services
- Use VM or use Fargate for server-less (no need to know for underlying EC2))
- Use EC2 for more control and personalisation
- Highly scalable and fault tolerant (everything is a stateless) Easy to main because
- Amazon Sagemaker (for ML model) using ECS
- Amazon Lex for chat bot
- ECR for storing the container images.

AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. This includes how the resources are related to one another and how they were configured in the past so that you can see how the configurations and relationships change over time.

AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals.

The BatchGetItem operation returns the attributes of one or more items from one or more tables. You identify requested items by primary key.

Amazon *EMR* Elastic MapReduce is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyse vast amounts of data.

Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for software vulnerabilities and unintended network exposure.

AWS *Trusted Advisor* provides recommendations that help you follow AWS best practices. Trusted Advisor evaluates your account by using checks. These checks identify ways to optimize your AWS infrastructure, improve security and performance, reduce costs, and monitor service quotas.

Sticky sessions *use cookies to help the client maintain a connection to the same instance over a cookie's lifetime*. Using sticky sessions configures a load balancer to bind user sessions to a specific instance. This means that all requests from a user during a session are sent to the same instance

A prefix (in S3) is a string of characters at the beginning of the object key name, like Europe/France/Nouvelle-Aquitaine/Bordeaux.

NACL: A network access control list (NACL) is *an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets*. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

invocation mean invoking/trigerings.

EventBridge (formerly CloudWatch Events) allows targets to be triggered using a schedule expression. A schedule expression can define a rate, for example, every 24 hours. Or can accept a standard cron job expression. EventBridge and CloudWatch Events use the same underlying API.

CF change set CloudFormation change sets enable the preview of proposed changes to a stack in order to assess the impact on running resources. This functionality allows the developer to check if any existing resources will be deleted or replaced upon application of the CloudFormation template

AWS SDKs :The AWS Software Development Kit (SDK) is a set of libraries that allow you to integrate your application with AWS Services.

Distributed application: A distributed application consists of one or more local or remote clients that communicate with one or more servers on several machines linked through a network. With this type of application, business operations can be conducted from any geographical location.

Athena provides a simplified, flexible way to analyze petabytes of data where it lives. Analyze data or build applications from an Amazon Simple Storage Service (S3) data lake and 25-plus data sources, including on-premises data sources or other cloud systems using SQL or Python.

AWS AppSync allows your applications to access exactly the data they need. Create a flexible API to securely access, manipulate, and combine data from multiple sources.

AppSpec file —> AppSpec file is used by CodeDeploy to specify and manage deployments. CodeBuild uses the BuildSpec file as a specification of build commands and settings. “secrets-manager” syntax can be used to retrieve API keys stored in AWS Secrets Manager. vs BuildSpec file

BuildSpec file —> CodeBuild uses the BuildSpec file as a specification of build commands and settings. “secrets-manager” syntax can be used to retrieve API keys stored in AWS Secrets Manager.

AppSpec file is used by CodeDeploy to specify and manage deployments. CodeBuild uses the BuildSpec file as a specification of build commands and settings. “secrets-manager” syntax can be used to retrieve API keys stored in AWS Secrets Manager.

The STS:AssumeRole API call returns a set of temporary security credentials which can be used to access AWS resources, including those in a different account. This API can be called by an IAM user or IAM role with existing temporary security credentials.

1. A user authenticates with Facebook first. They are then given an ID token by Facebook. An API call, AssumeRoleWithWebIdentity, is then used in conjunction with the ID token. A user is then granted temporary security credentials to access AWS services.