

Problem - 3 Propose solutions for the following scenarios

Scenario 1 - Hosting a Website

Host a static website with 10 pages and 500MB in assets with a SSL certificate. The domain is already configured in Route 53. Propose a solution that would have the least management overhead while being cost effective.

Services should be used -:

```
S3 bucket -> to save all the files related to the website
Cloudfront -> Works as a content delivery network (CDN) which can deliver the
content in s3 as a static website.
SSL/TLS -> We can configure SSL certs in the cloudfront.
Route53 -> And we need to add a CNAME record in Route53 pointing to the CDN
DNS name.
BitBucket -> We can use a bitbucket pipeline to build the project and copy
the artifacts to the s3 and create invalidations to clear cache in
edgelocations which are used by cloudfront to cache and deliver content.
```

Scenario 2 - Application Deployment

A business analyst has requested your input on how to deploy a python based data processing application.

Application Functionality

- The application will query data from multiple sources including RDS/Redshift/S3.
- Temporary store the data in the application and do the necessary processing.
- Output the data to a RDS instance.

Considerations

- The data processing activity will be automatically carried out every 6 hours.
- Data processing will take an average of 45-60 minutes
- Application must be automatically scalable to handle any data load.
- Application must be easily deployable in multiple AWS environments with minimal downtime between updates.y
- Application will not require a persistent data store as any processed data will be discarded immediately after its been copied over to the RDS instance.

Services should be used -:

AWS Lambda:

Use AWS Lambda to run the Python-based data processing application. Lambda functions can be triggered by an AWS Step Functions state machine every 6 hours. Lambda automatically scales based on the number of incoming requests, ensuring it can handle any data load.

AWS Step Functions:

Create a Step Functions state machine to orchestrate the workflow. This includes querying data from RDS, Redshift, and S3, processing the data, and then writing the output to the RDS instance. Step Functions allows you to break down the processing into multiple steps, handle retries, and maintain state between each step.

Amazon S3:

Use S3 for temporary data storage during the processing. Data can be stored in S3 buckets and accessed by the Lambda functions during processing. S3 provides high availability and durability for storing intermediate data.

Amazon RDS:

Use Amazon RDS (e.g., PostgreSQL or MySQL) to store the final processed data. Lambda functions can write the output data to the RDS instance. Ensure the RDS instance is configured for high availability using Multi-AZ deployments.

Amazon Redshift:

If your data processing involves querying large datasets, use Amazon Redshift for fast querying. Lambda functions can connect to Redshift to fetch the necessary data.

AWS Systems Manager Parameter Store:

Store configuration parameters, such as database connection strings and API keys, in Parameter Store. Lambda functions can securely retrieve these parameters at runtime.

AWS CloudFormation:

Use CloudFormation to define and deploy the infrastructure as code. This ensures the application can be easily deployed across multiple AWS environments with minimal downtime.

Define all necessary resources, including Lambda functions, Step Functions, S3 buckets, RDS instances, and IAM roles, in a CloudFormation template.

Scenario 3 - Data Transfer

What would be the most cost effective solution to migrate two S3 buckets containing 500GB of data in the Sydney region to a new set of buckets in Oregon. Provide a cost breakdown.

AWS DataSync:

Use AWS DataSync to transfer the data from the source S3 buckets in Sydney to the destination S3 buckets in Oregon. DataSync offers efficient and cost-effective data transfer with built-in features like incremental transfers and automatic error recovery.

S3 Cross-Region Replication:

Enable S3 Cross-Region Replication on the destination buckets in Oregon to keep them synchronized with the source buckets in Sydney. This ensures that any new data added to the source buckets is automatically replicated to the destination buckets.

Cost Breakdown

AWS DataSync:

Data Transfer Cost: DataSync charges are based on the amount of data transferred.

Assuming 500GB of data transfer from Sydney to Oregon.

Data Transfer Out from Sydney to Oregon: \$0.02 per GB

Total Data Transfer Cost = 500GB * \$0.02 = \$10

DataSync Agent Cost: DataSync agent instances are charged per hour of usage.

Assuming the migration completes within a few hours.

DataSync Agent Cost: Varies based on instance type and duration of usage.

Let's assume \$0.10 per hour for a small instance.

Total DataSync Agent Cost = Number of Hours * Cost per Hour

S3 Cross-Region Replication:

There are no additional charges for enabling S3 Cross-Region Replication.

However, standard S3 storage charges apply for storing the replicated data in the destination buckets in Oregon.

Total Cost Estimate

Data Transfer Cost (DataSync): \$10

DataSync Agent Cost: Variable, depending on the duration of usage.

S3 Storage Cost (Destination Buckets in Oregon): Standard S3 storage charges apply.