

Considerations

1. The client intends to migrate more applications in the future. How will the solution remain intuitive to new developers and scale to their needs?

The current deployment model is holding the basic infrastructure for the current microservice application. So the new developers have got an idea to scale to their needs.

They can reuse the current module wherever they want for additional resources.

So the implementation time will be reduced.

2. Encrypting resources at rest and in transit will help the client improve their security position.

AWS recommends encryption as an additional access control to complement the identity, resource, and network-oriented access controls already described. AWS provides a number of features that enable customers to easily encrypt data and manage the keys.

All AWS services offer the ability to encrypt data at rest and in transit

AWS KMS is a fully managed service. As your use of encryption grows, the service automatically scales to meet your needs. It enables you to manage thousands of KMS keys in your account and to use them whenever you want.

To protect data in transit, AWS encourages customers to leverage a multi-level approach. All network traffic between AWS data centres is transparently encrypted at the physical layer. All traffic within a VPC and between peered VPCs across regions is transparently encrypted at the network layer when using supported Amazon EC2 instance types. At the application layer, customers have a choice about whether and how to use encryption using a protocol like Transport Layer Security

(TLS). All AWS service endpoints support TLS to create a secure HTTPS connection to make API requests.

Ensure strong security isolation between containers. AWS provides the latest security updates and lets you set granular access permissions for every container. AWS offers over 210 security, compliance, and governance services, plus key features to best suit your needs.

3. How can the application be deployed in such a way to scale with demand?

AWS Auto Scaling:

AWS Auto Scaling makes scaling simple with recommendations that allow you to optimise performance, costs, or balance between them. If you're already using Amazon EC2 Auto Scaling, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, you can set up scaling for multiple resources across multiple services in minutes. AWS Auto Scaling provides a simple, powerful user interface that lets you build scaling plans for Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables, and Amazon Aurora Replicas.

There is no additional charge for AWS Auto Scaling. You pay only for the AWS resources needed to run your applications and Amazon CloudWatch monitoring fees. To get started you can use the AWS Management Console, Command Line Interface (CLI), or SDK.

Amazon ECS cluster Auto Scaling:

Amazon ECS cluster auto scaling provides control over how you scale the Amazon EC2 instances within a cluster. When you use managed scaling, Amazon ECS creates the Auto Scaling group capacity provider infrastructure, and manages the scale-in and scale-out actions of the Auto Scaling group based on your clusters' tasks load.

AWS App Runner:

Production of web applications at scale made it easy for developers.

Automatic Deployments

When you connect App Runner to your code repository or container image registry, App Runner can automatically build and deploy your application when you update your source code or container image.

Load Balancing

App Runner automatically balances traffic to provide high levels of reliability and availability for your applications.

Auto Scaling

Enabled by default, App Runner automatically scales the number of containers up or down to meet the needs of your application.