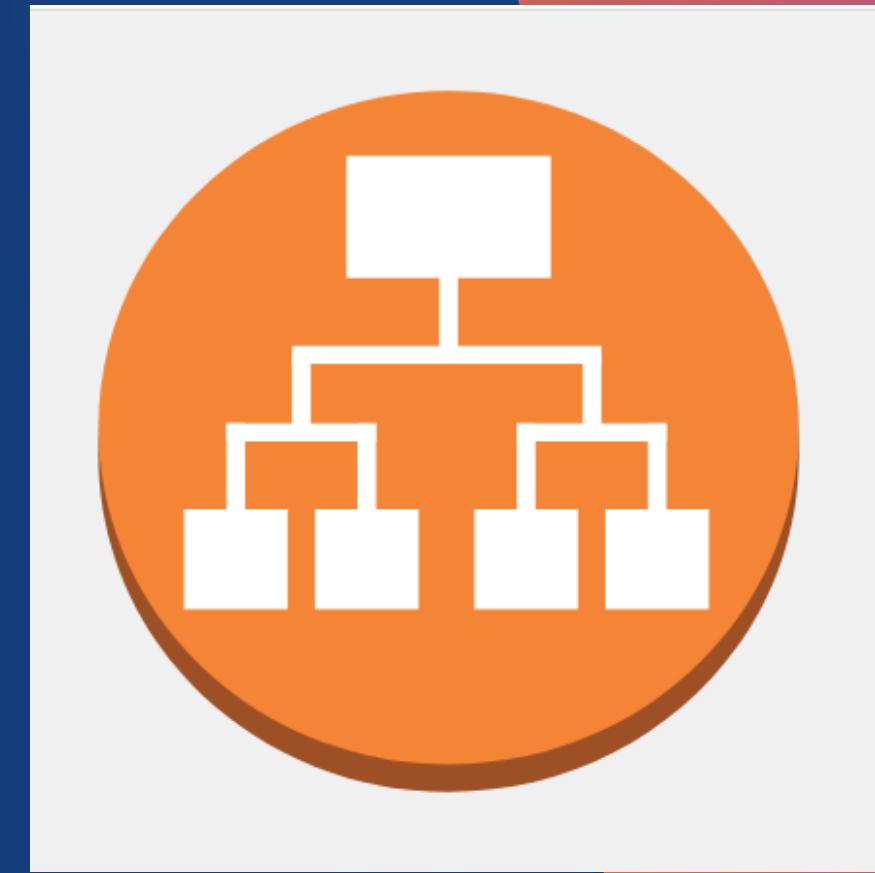


AWS ELB & cloudwatch



AMOL SHETE

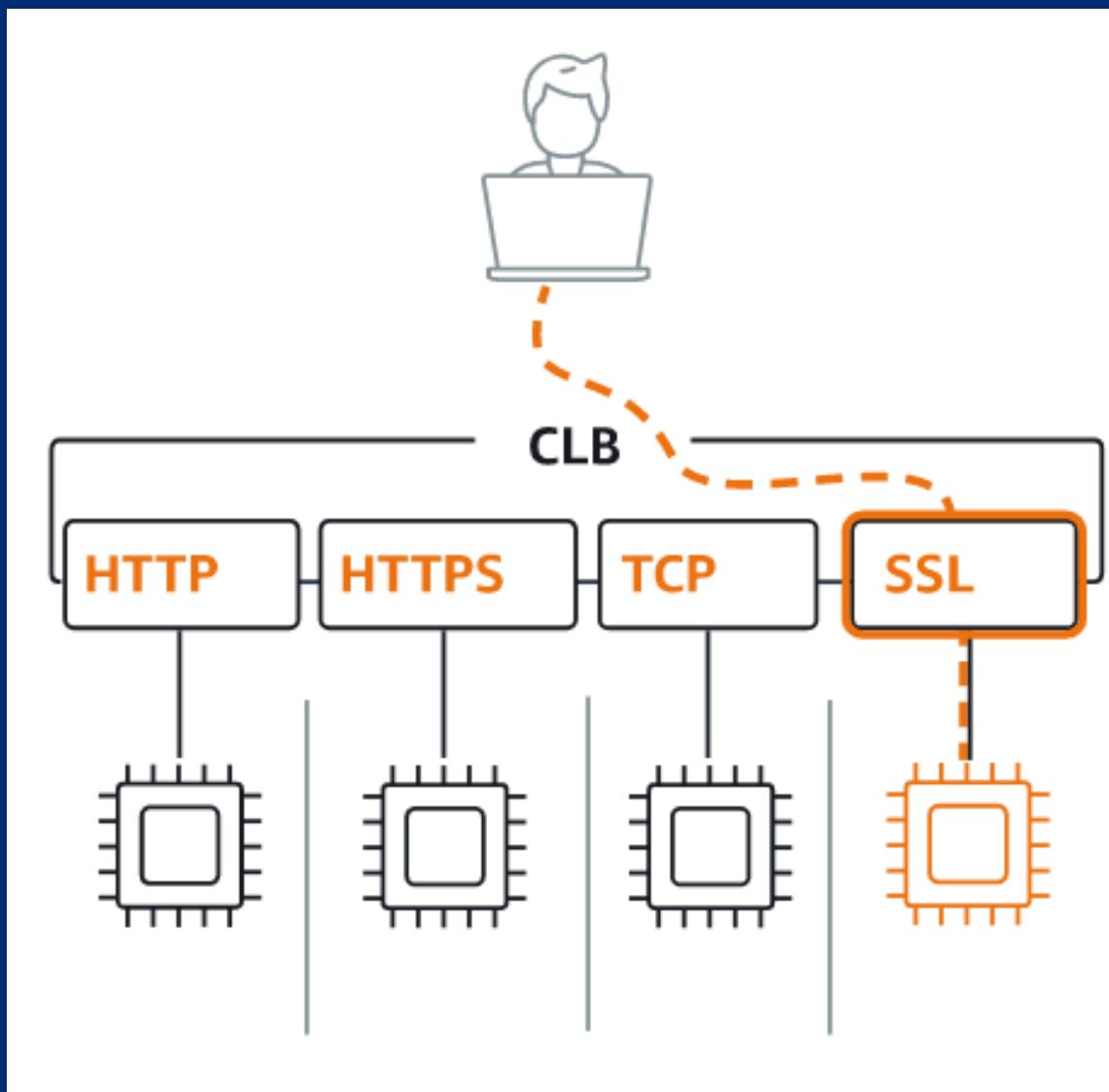
Elastic Load Balancer

Elastic Load Balancing distributes incoming application or network application or network traffic across multiple targets, such as amazon instances, containers and IP addresses in multiple az.

Elastic Load Balancer supports three types of load balancers

- Application Load Balancer
- Network Load Balancer
- Classic Load Balancer

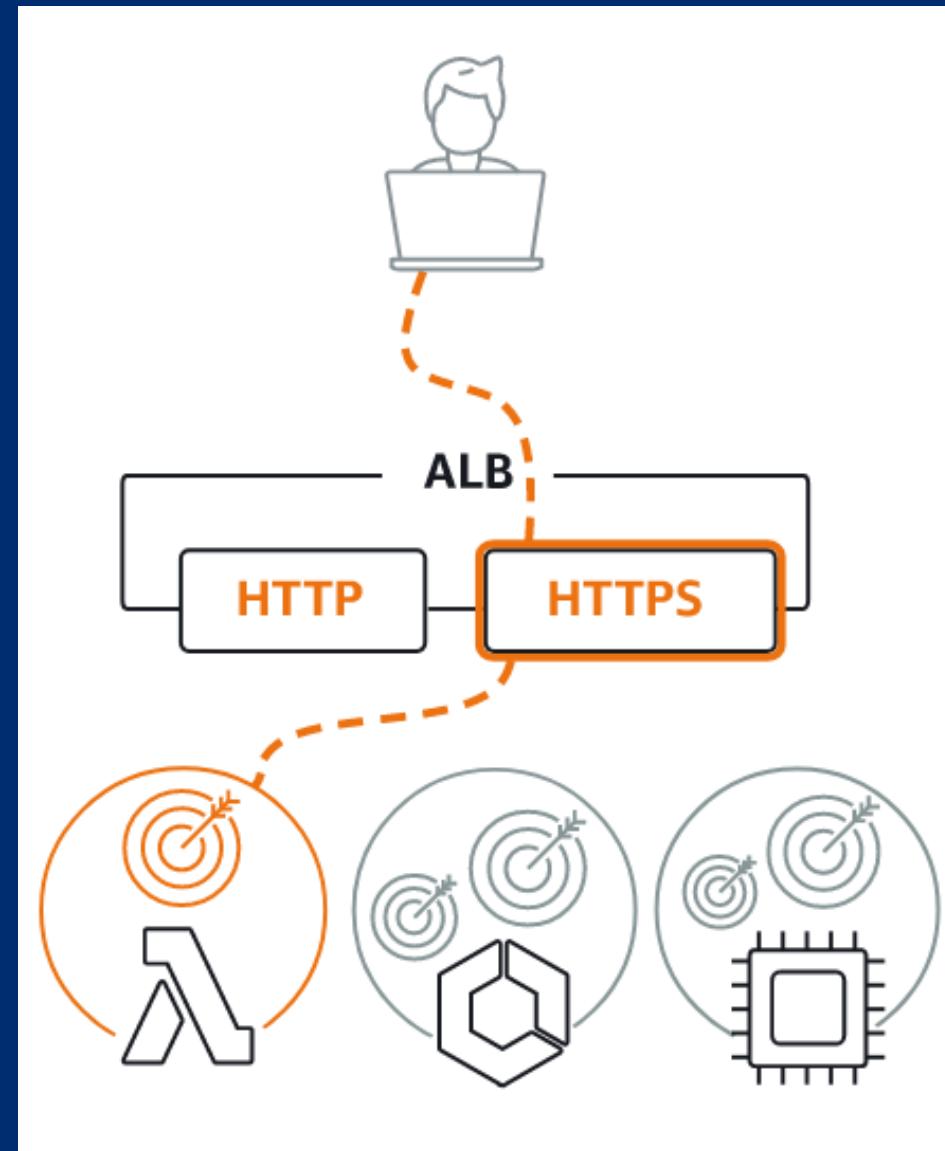
Classic Load Balancer



The classic load Balancer that routes traffic based on either application or network level information

The classic load balancer is ideal for simple load balancing of traffic across multiple EC2 instances

Application Load Balancer

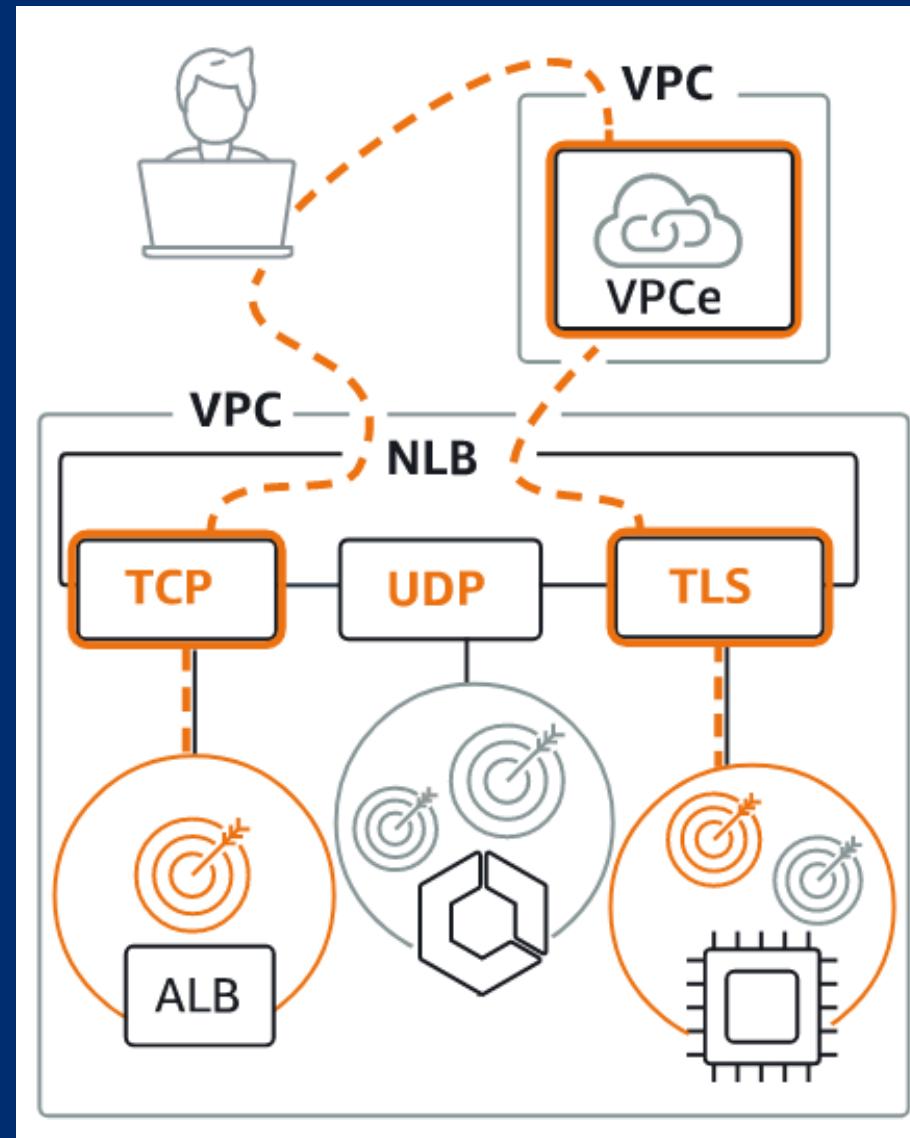


Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications.

Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.



AWS Cloudwatch

Monitor performance of AWS environment standard infrastructure metrics

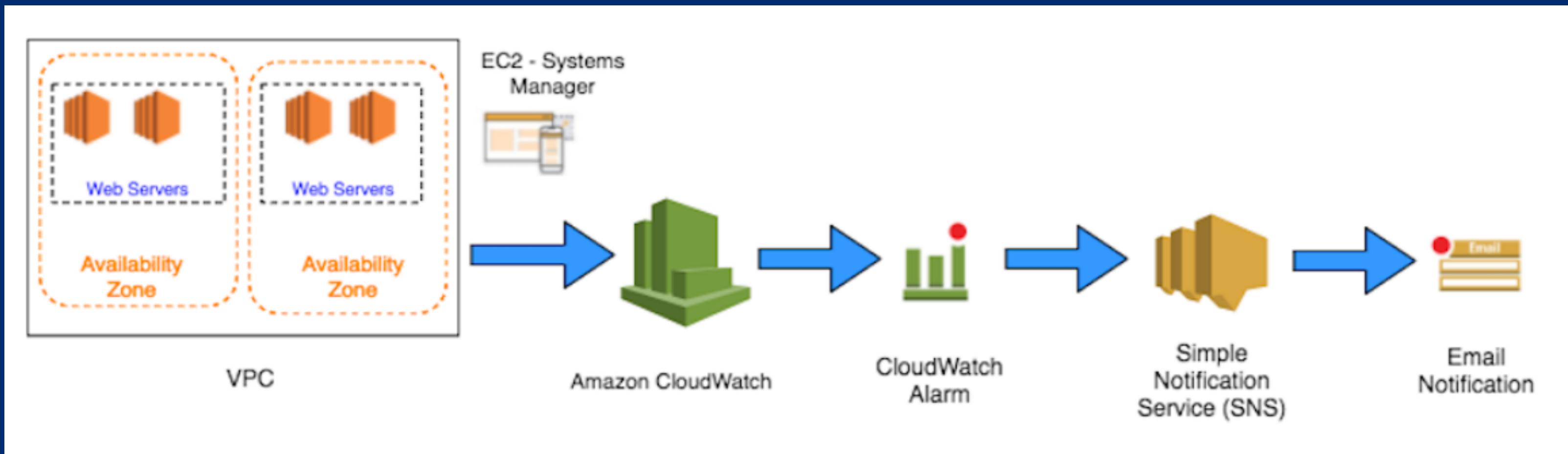
Metrics: AWS cloud watch allows you to record metrics for services such as EBS, EC2 ,ELB, Route 53 health checks, RDS, Amazon S3, Cloudfront etc

Events: AWS Events delivers a near real time stream of system events that describe changes in Amazon Web Services (AWS) resources

AWS Cloudwatch

Logs: You can use Amazon cloudwatch Logs to monitor, store, and access your logs files from Amazon Elastic Compute Cloud (Amazon EC2) instances , AWS cloudtrails, Route 53 and other services

Demo



AWS Services

AWS EFS



AWS EFS SERVICE:-

AWS EFS is a scalable, fully managed file storage service that is designed to provide shared access to files across multiple Amazon Elastic Compute Cloud (EC2) instances. It is built on the Network File System (NFS) protocol, which is a distributed file system protocol that allows multiple clients to access files on a remote server as if they were on a local disk.

AWS EFS SERVICE:-

With AWS EFS, you can easily create and configure file systems that can be accessed by multiple EC2 instances simultaneously. EFS supports multiple file systems and can scale automatically to petabyte-scale, so you can store and access virtually any amount of data.

One of the key benefits of AWS EFS is its ability to provide consistent and low-latency performance, even with multiple EC2 instances accessing the same file system simultaneously. This is achieved through EFS's use of SSD-based storage and its ability to automatically scale storage capacity and throughput based on usage patterns.

AWS EFS SERVICE:-

AWS EFS also provides a number of advanced features, such as support for file locking, access control, and lifecycle management. EFS integrates seamlessly with other AWS services, including AWS Lambda, Amazon Elastic Container Service (ECS), and Amazon Elastic Kubernetes Service (EKS).

In summary, AWS EFS is a fully managed, scalable file storage service that enables you to store and share files across multiple EC2 instances. It provides consistent performance, advanced features, and seamless integration with other AWS services.

EFS Mount Process:

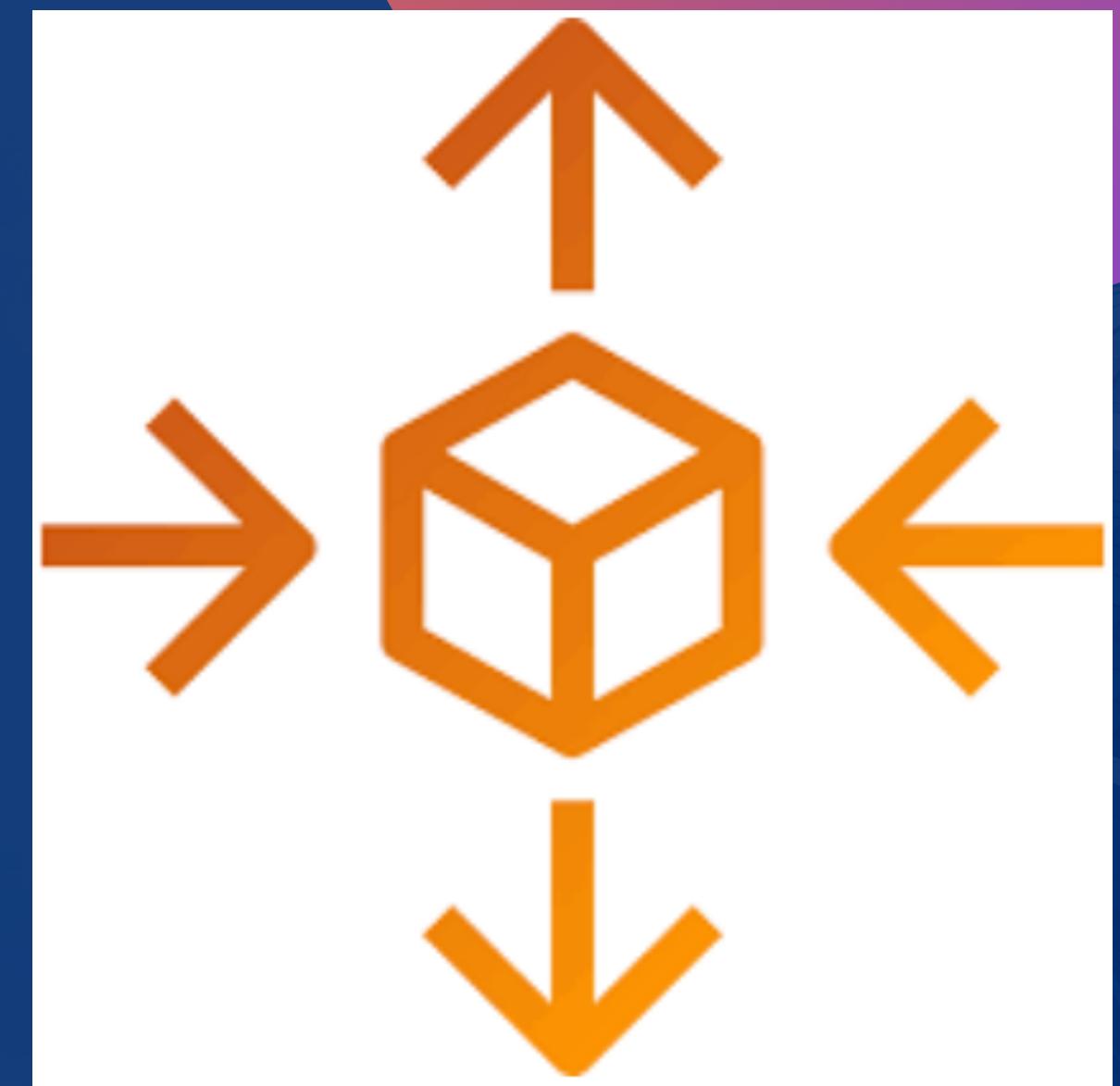
Please check the other EFS mount process.

Also you can refer below link for mount process

AWS Services

AWS

Autoscaling



AWS Autoscaling:

AWS Autoscaling is a service that allows you to automatically scale your EC2 instances up or down based on demand. Autoscaling helps you optimize your application's performance and minimize costs by adjusting the number of instances in response to changes in traffic.

How it works:

1. You start by creating an Autoscaling group, which is a collection of EC2 instances that share the same configuration and scaling policies.
2. You can then define scaling policies that determine when Autoscaling should add or remove instances based on factors such as CPU usage, network traffic, or other custom metrics.
3. Autoscaling monitors the health of your EC2 instances and can automatically replace any unhealthy instances with new ones.

How it works:

1. You can configure Autoscaling to automatically adjust the size of your EC2 instances based on demand, or to maintain a fixed number of instances at all times.
2. Autoscaling can work with other AWS services, such as Elastic Load Balancing (ELB), to distribute traffic across multiple instances.
3. Autoscaling integrates with Amazon CloudWatch, allowing you to monitor and analyze performance metrics for your instances and adjust your scaling policies accordingly.

AWS Autoscaling:

With Autoscaling, you can create highly available, fault-tolerant applications that can handle unpredictable traffic spikes.

Autoscaling is available for a variety of EC2 instance types, including Amazon Machine Images (AMIs), Docker containers, and more.

In summary, AWS Autoscaling is a powerful tool that allows you to scale your application automatically, without requiring manual intervention. By optimizing the number of instances based on demand, Autoscaling helps you minimize costs and improve your application's performance.

AWS Services

AWS S3



What is AWS S3:

An S3 bucket is a storage service provided by Amazon Web Services (AWS) that allows users to store and retrieve data from anywhere on the web.

AWS S3 stands for Simple Storage Service, which is a scalable object storage service provided by Amazon Web Services (AWS).

S3 allows users to store and retrieve data from anywhere on the web, making it a popular choice for hosting static websites, storing data backups, and more. S3 provides a highly durable and available storage infrastructure, ensuring that data is always accessible.

AWS S3:

S3 stores data as objects, each of which is identified by a unique key and can range in size from a few bytes to terabytes of data.

S3 buckets are the primary means of storing and organizing data in S3. Each bucket must have a unique name and is identified by a globally unique DNS address.

S3 allows users to define lifecycle policies, which automatically transition objects to different storage classes based on their age and access patterns.

AWS S3:

S3 integrates with other AWS services such as AWS Lambda, AWS CloudTrail, and Amazon CloudFront, providing a highly scalable and flexible storage solution for modern cloud applications.

S3 provides a pay-as-you-go pricing model, allowing users to only pay for the storage and data transfer they use.

S3 is highly scalable and can handle an unlimited number of objects, making it an ideal choice for storing large datasets or hosting high-traffic websites.

AWS Services

AWS RDS



DB Administration Challenges:

- Install
- Patching
- Monitoring
- Performance Tuning
- Backups
- Scaling
- Security
- Hardware upgrades
- Storage Management

AWS RDS:

AWS RDS is a managed database service provided by Amazon Web Services (AWS) that simplifies the process of setting up, operating, and scaling a relational database in the cloud.

RDS supports a range of popular relational database engines, including Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server.

RDS automates many of the time-consuming tasks associated with database administration, such as hardware provisioning, software patching, and database backups.

AWS RDS:

RDS provides automatic software patching for database engines, ensuring that your database is always up-to-date with the latest security patches and bug fixes.

RDS allows you to easily scale your database up or down based on your performance needs, using features such as read replicas and multi-AZ deployments.

RDS supports a range of security features, including encryption at rest and in transit, network isolation using Virtual Private Cloud (VPC), and access controls using IAM.

AWS RDS:

RDS provides a range of monitoring and logging capabilities, including Amazon CloudWatch metrics, automated backups, and database logs, to help you troubleshoot issues and optimize database performance.

RDS integrates with other AWS services, such as AWS Lambda, AWS CloudFormation, and Amazon S3, allowing you to build scalable and flexible applications that rely on a reliable database backend.

RDS provides a pay-as-you-go pricing model, allowing you to only pay for the resources you use.

AWS Services

AWS Elastic Beanstalk



AWS Elastic Beanstalk:

Elastic Beanstalk is a fully managed service provided by AWS (Amazon Web Services) that enables developers to quickly deploy and run applications in various programming languages and frameworks on AWS infrastructure. It automates the process of setting up and configuring the resources required for an application, such as EC2 instances, load balancers, databases, and storage.

AWS Elastic Beanstalk:

With Elastic Beanstalk, developers can focus on writing code and not worry about the underlying infrastructure. They can easily deploy and manage their applications, monitor their performance, and scale them up or down based on demand. Elastic Beanstalk also supports various deployment options, including rolling updates, blue/green deployments, and canary deployments.

AWS Elastic Beanstalk:

Elastic Beanstalk supports a wide range of programming languages and frameworks, such as Java, .NET, Python, Ruby, Node.js, PHP, and Go. It also integrates with other AWS services, such as RDS (Relational Database Service), S3 (Simple Storage Service), and CloudFront.

Overall, Elastic Beanstalk helps developers to easily deploy and manage their applications on AWS without worrying about the underlying infrastructure, allowing them to focus on delivering new features and functionality to their customers.

AWS Services

AWS

CodeCommit



AWS Code Commit:

AWS CodeCommit is a fully managed source control service that makes it easy for teams to host secure and highly scalable private Git repositories.

It offers features such as pull requests, branch management, and code reviews, which enable teams to collaborate effectively and maintain high code quality.

CodeCommit integrates with other AWS services such as AWS CodeBuild, AWS CodePipeline, and AWS CodeDeploy to provide a complete continuous delivery and deployment solution.

AWS Code Commit:

CodeCommit also provides integrations with popular Git clients and IDEs, including Git command line, Eclipse, Visual Studio, and IntelliJ IDEA.

The service offers high availability and durability, with automatic backup and replication of data across multiple availability zones.

Pricing is based on the number of active users and the amount of data stored in repositories, with no upfront fees or long-term commitments required.

AWS Services

AWS

CodeBuild



AWS Code Build:

AWS CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy.

It supports multiple programming languages and build tools, such as Java, Python, Node.js, Ruby, Maven, Gradle, and Docker.

CodeBuild can integrate with various AWS services, such as AWS CodeCommit, AWS CodePipeline, and AWS Elastic Beanstalk, to provide a complete continuous integration and delivery (CI/CD) pipeline.

AWS Code Build:

It is highly scalable and can parallelize builds across multiple compute instances to reduce build times.

CodeBuild provides a range of built-in build environments, including Amazon Linux, Windows, and Docker, or you can create your own custom build environment using a Docker image.

It supports both Linux and Windows operating systems.

CodeBuild can also integrate with third-party build tools and services, such as GitHub, Jenkins, and Atlassian Bamboo.

AWS Code Build:

You only pay for the build time and resources you use, with no upfront fees or minimum commitments.

CodeBuild provides a simple web interface and a command-line interface (CLI) to manage builds and view build logs.

Overall, AWS CodeBuild simplifies and automates the process of building and testing code, enabling developers to focus on writing code and delivering new features rather than managing build infrastructure.

AWS CodeBuild buildspec:

1. Create a text file named **buildspec.yml** in the root directory of your project.
2. Define the version of the build specification format you want to use at the top of the file. For example, to use version 0.2, you can add the following line: `version: 0.2`
3. Specify the phases of the build process in the **phases** section. Each phase represents a step in the build process, such as installing dependencies, building the application, and testing it. You can specify any number of phases, and each phase can contain one or more commands to be executed.

AWS CodeBuild buildspec:

4. You can check the syntax from official document with following link: <https://docs.aws.amazon.com/codebuild/latest/userguide/build-spec-ref.html>
5. Specify any optional artifacts to be produced by the build process in the artifacts section. Artifacts are files or directories that are produced by the build and can be used for deployment or further processing. p in the build process, such as installing dependencies, building the application, and testing it. You can specify any number of phases, and each phase can contain one or more commands to be executed.

