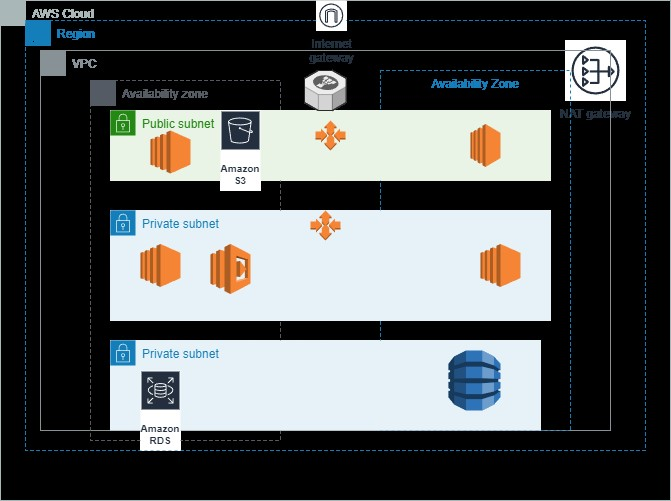
Common 3 -Tier architecture



1. **Server-Oriented:**

* Elastic Load Balancer (ELB): Use AWS Application Load Balancer (ALB) or Network Load Balancer (NLB) to distribute incoming traffic to the web servers and application servers.
* Web Servers: These are typically EC2 instances running web server software (e.g., Apache, Nginx). Can use Auto Scaling Groups for high availability and scalability.
* Application Servers: EC2 instances or AWS Elastic Beanstalk (if you need a Platform-as-a-Service) running application code (e.g., Node.js, Java, Python).
* Database Servers: Amazon RDS (Relational Database Service) for relational databases (e.g., MySQL, PostgreSQL, SQL Server), or Amazon DynamoDB for NoSQL databases.

1. **Serverless:**

* API Gateway: AWS API Gateway can act as the front-end for serverless architectures, managing API endpoints and routing requests.
* Lambda Functions: AWS Lambda for serverless compute. You can run your application logic as Lambda functions, which are event-driven and automatically scalable.
* DynamoDB: Use Amazon DynamoDB for serverless, NoSQL database needs. It can handle high-traffic workloads and scales automatically.
* Static Website Hosting: For serving static content, you can use Amazon S3 for hosting and Amazon CloudFront for content delivery.
* Authentication and Authorization: AWS Cognito for user authentication and authorization.
* Storage: AWS S3 for object storage, and Amazon Aurora Serverless for relational database needs.
* Notification: Amazon SNS for notifications, if needed.

**Considerations:**

* Monitoring and Logging: Use AWS CloudWatch for monitoring and AWS CloudTrail for auditing and logging in both server-oriented and serverless architectures.
* Security: Implement security groups, network ACLs, and AWS Identity and Access Management (IAM) roles to secure your resources.
* Scaling: In server-oriented architecture, use Auto Scaling for instances. In serverless, services like Lambda and DynamoDB scale automatically.
* High Availability: Design your architecture across Availability Zones (AZs) for high availability in both server-oriented and serverless setups.

Choice between server-oriented and serverless architectures depends on your specific application requirements, scalability needs, and cost considerations. Serverless architectures can provide cost savings and simplified management, while server-oriented setups offer more control and flexibility.