AWS

INTRODUCTION TO CLOUD

- Cloud provides IT Resources and Services to all kind of businesses and/or individuals.
- Instead of "Self Owned and Managed" Servers and Environments, one can use resources available in large pool from Vendors like Amazon, google or Microsoft.
- Cloud Vendors provide resources and services for per-use basis
- Eliminates the capital expenditure

CLOUD BENEFITS

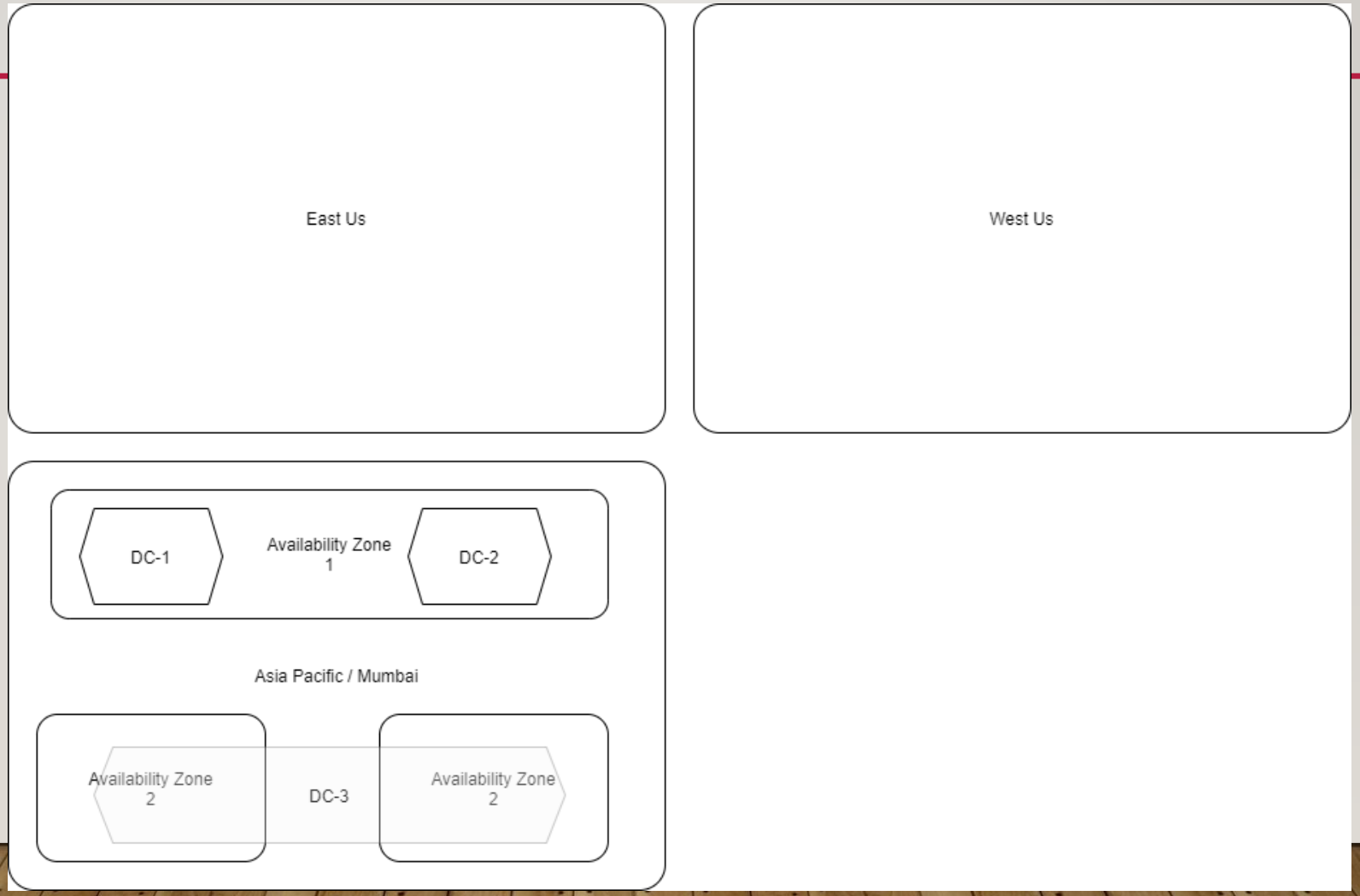
- Large resource pool from Vendors.
- Data centres managed and operated by Vendors.
- Security & Availability guarantee from vendor.
- No Capital investment (Per per use)
- Elasticity (Easy to operate and scale)

CLOUD : REGION, DATACENTERS AND AVZONES

- Cloud vendors have setup "Data-Centers" across globe.
- Vendors have identified a geo-graphical region with term "Region"
- A Region may have one to three Availability zone
- Each Availability Zone may have one or more data-centers

- NOTE:
  - In few regions, a single data-center might actually host one or two Availability Zones.

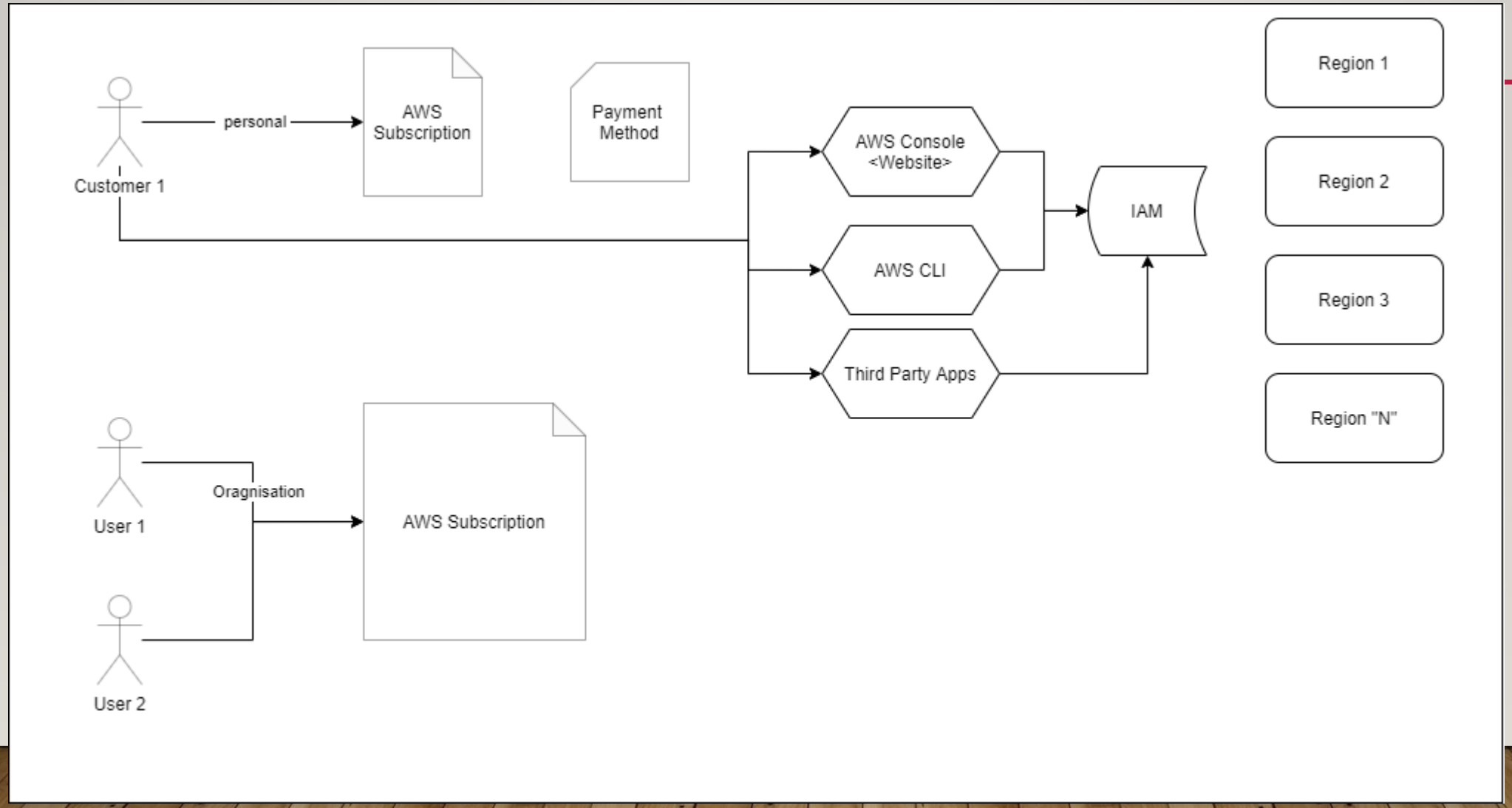
REGIONS, DC AND AZ



INTRODUCTION TO AWS

- Amazon Web Services (AWS) Is public cloud offering from Amazon.
- Available across more than 26 regions.
- Easy access / operations through choice of
  - Web Console
  - AWS CLI
  - Third party automation tools
- Option to terminate account
- Built in Security and Monitoring
- Budgets and Alarms

AWS SUBSCRIPTION AND ACCESS



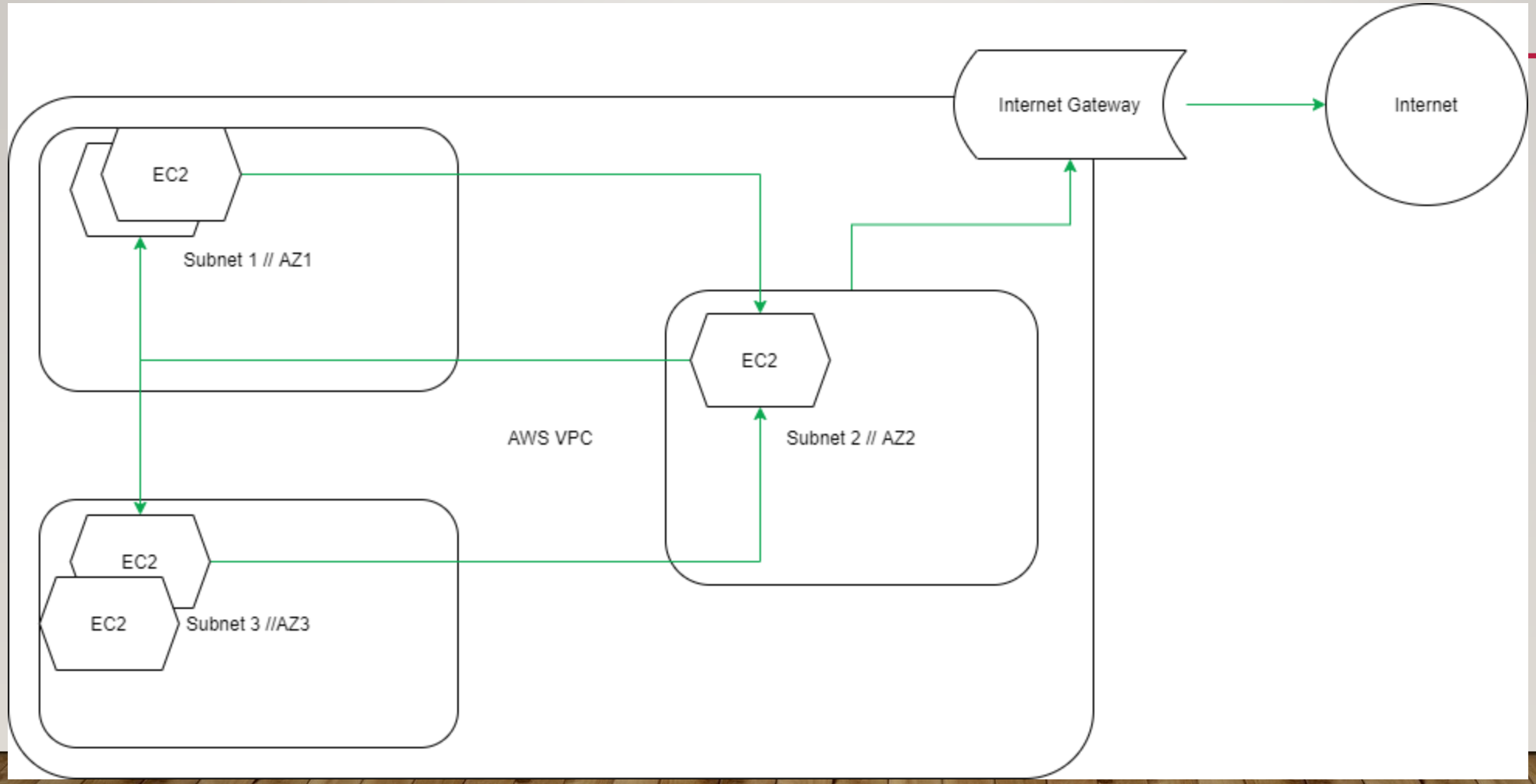
CLOUD SERVICE MODELS

Model	Description	Services
IaaS	Infrastructure As A Service, Offers better control for IT Admins	EC2, VPC, Elastic Load Balancer, S3
PaaS	Platform As A Service, easy to use platform for developers.	Elastic BeanStalk, Lambda, RDS

AWS SERVICES:VPC

- Virtual Private Cloud (VPC) Represents a "private network" on cloud where EC2 instances and other services can be deployes.
- Allows services like ec2 instances communicate with each other.
- Access internet via "Internet Gateway" which is optional.
- Organised into Subnets
- A Subnet could be "private" (No Internet gateway) or "public" (with Internet Gateway)

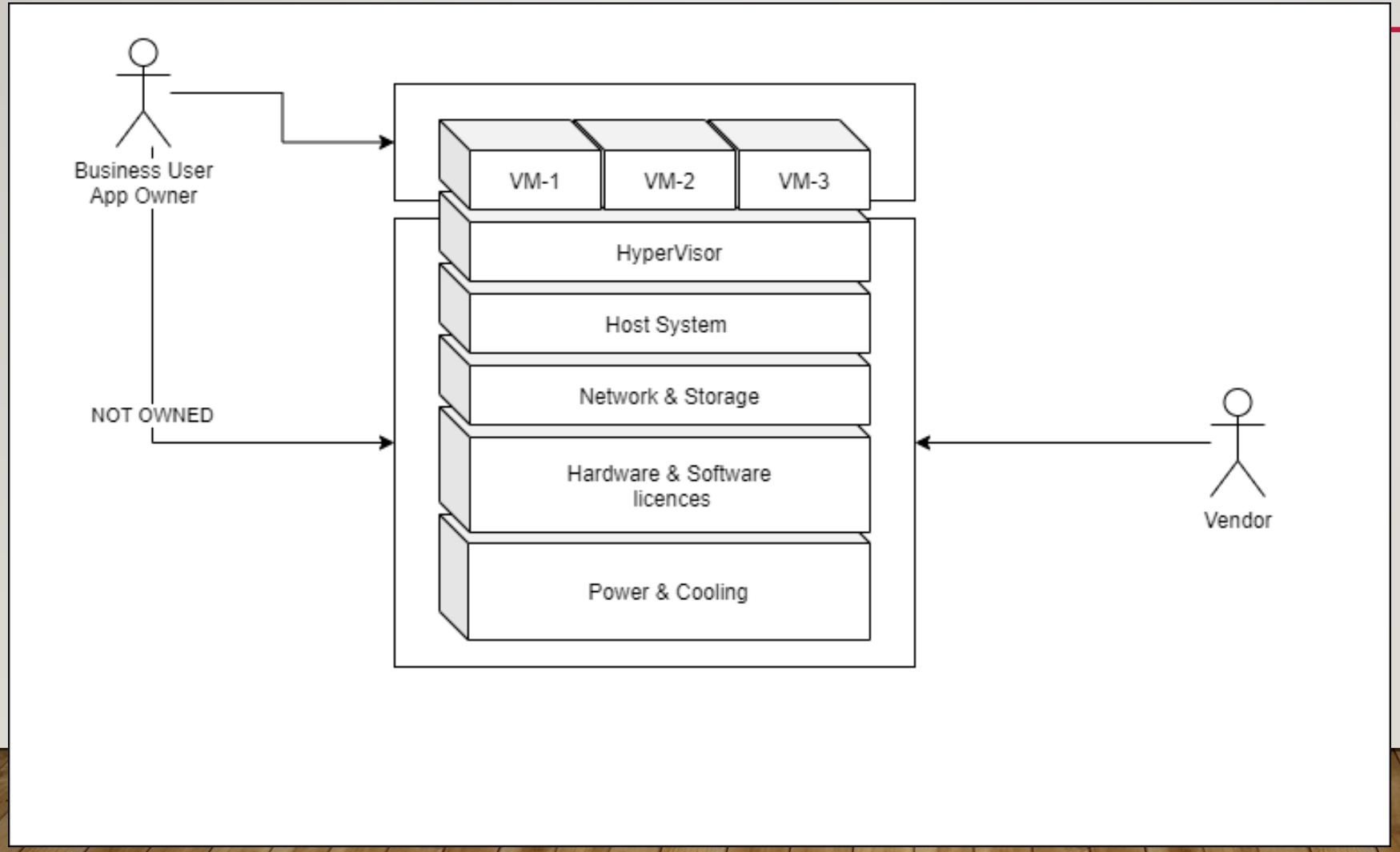
A TYPICAL VPC



EC2 INSTANCE

- Windows / Linux Server on AWS Cloud
- Flexible pricing plans
  - Free Tier : t2.micro
- Use "EBS" (Elastic Block storage) for Disk
  - Free Tier : Use only 10 GB (Max 30 GB free)
- With Linux / Windows Administration skills, can convert this server into application environment or database !!
- Administrator access via:
  - Linux = > SSH
  - Windows => RDP

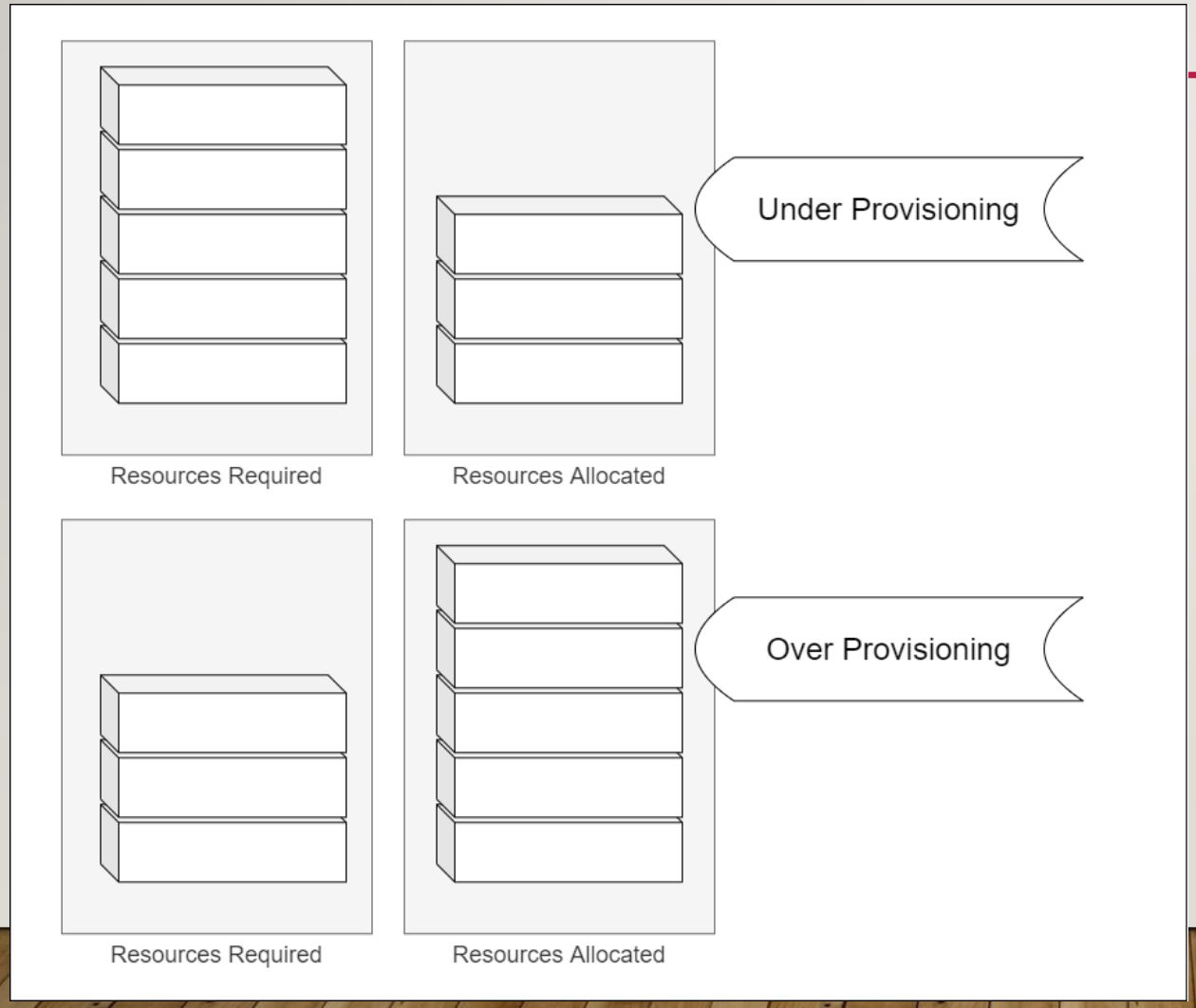
EC2 : WHO MANAGES WHAT ?



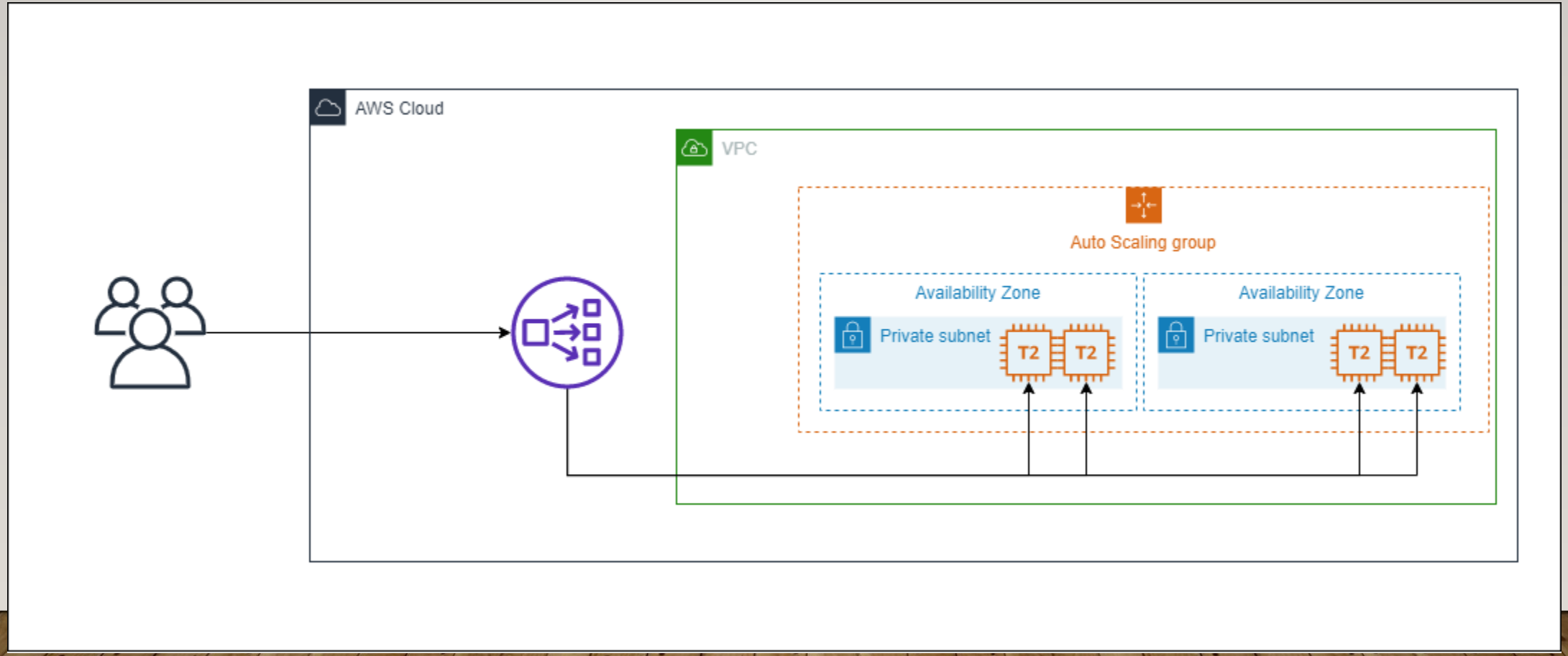
AUTOSCALING GROUP

- Deploy multiple EC2 Instances spread into Availability Zones
- Offers high availability
- Offers auto-scaling based on cpu usages.
  - Scale In (Remove unwanted instances)
  - Scale Out (Add more instances)

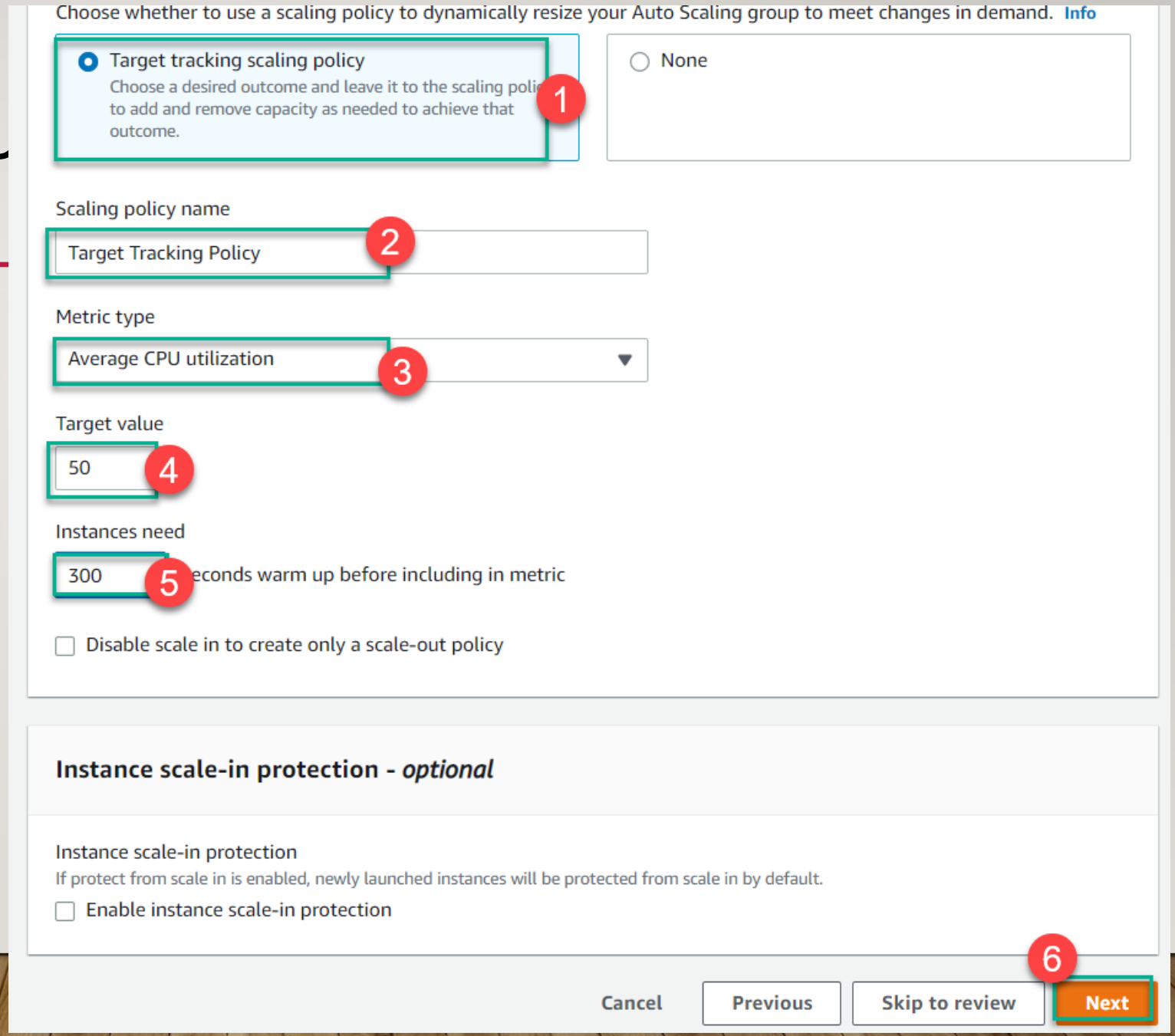
OVER / UNDER PROVISIONING



AUTOSCALING GROUP AND VPC

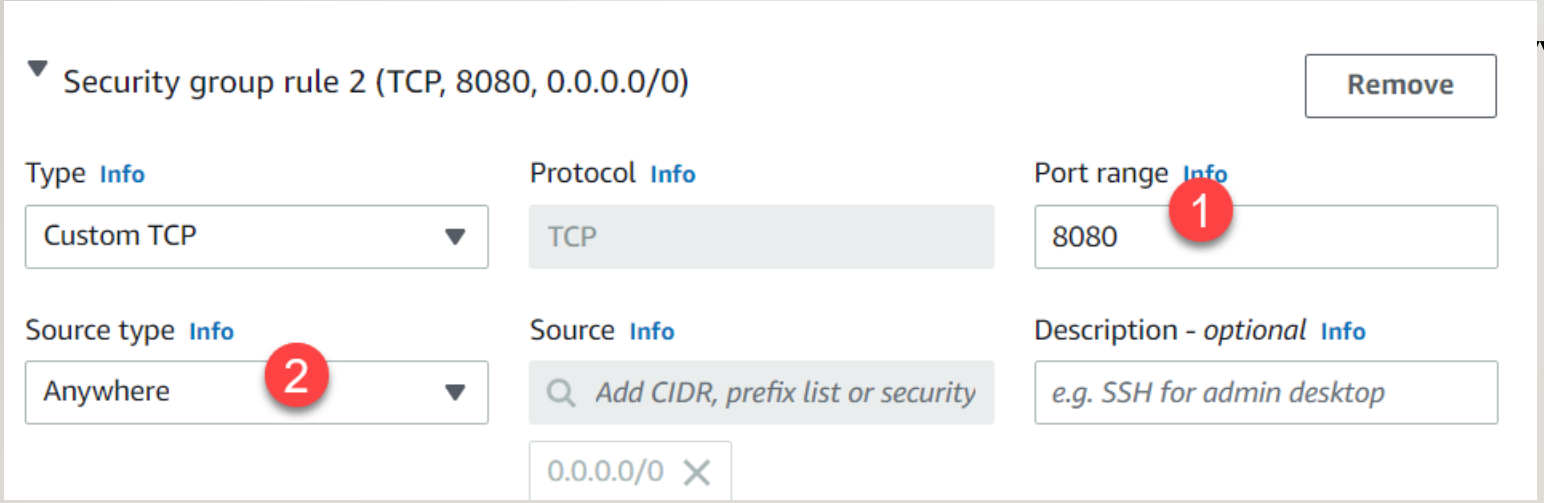


AUTOSCALING RU



SECURITY GROUPS

- Security Group Allows/Deny access to IP/PORT Range
  - Inbound rule: EC2 Access from External Client
  - Outbound rule : EC2 Instances accessing external server / service

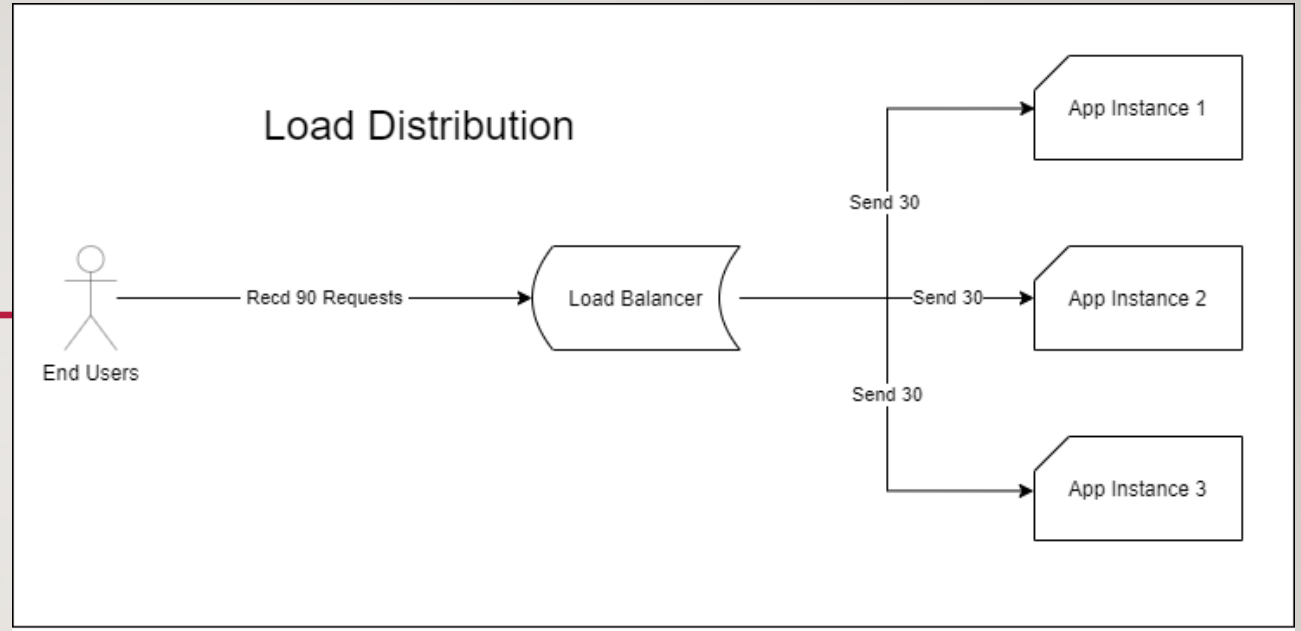


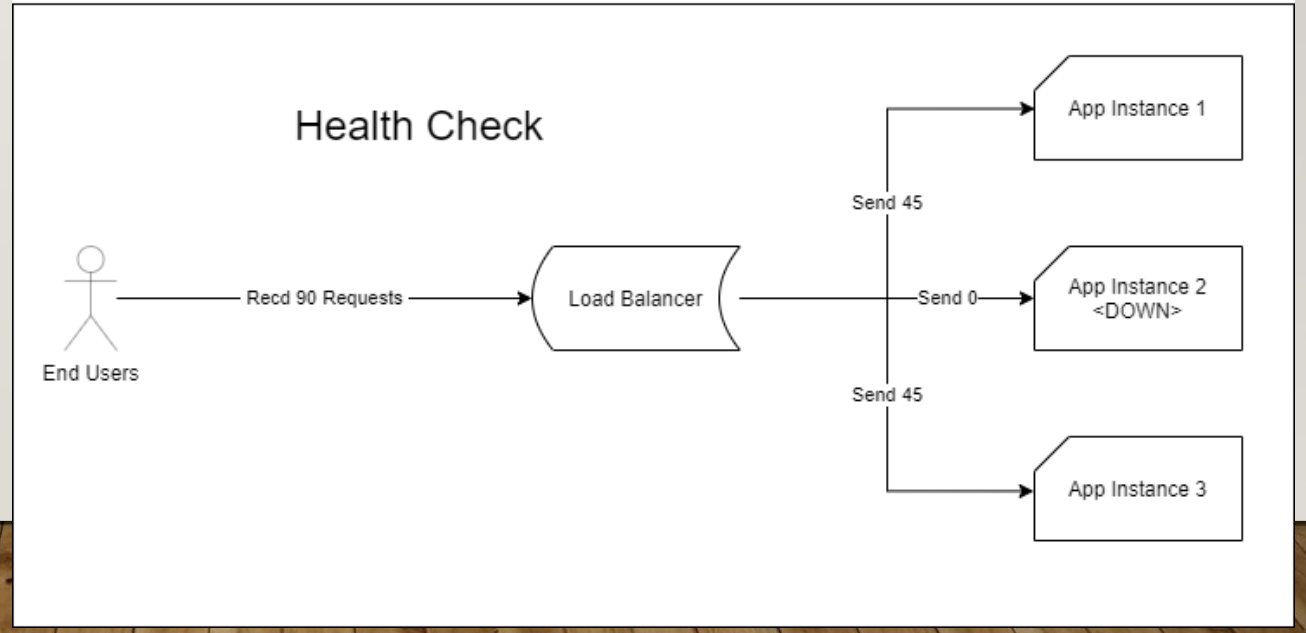
where

LOAD BALANCER

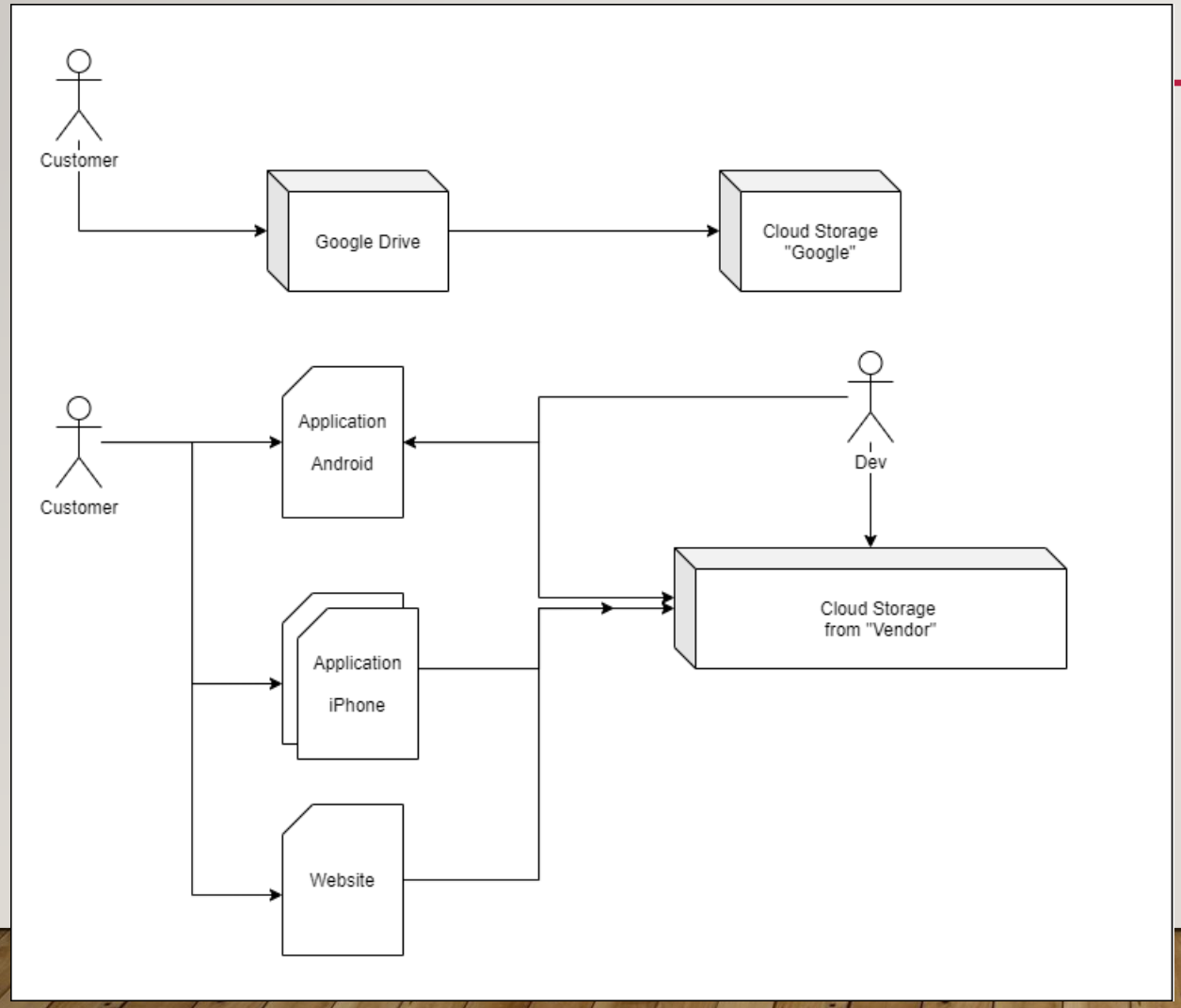
- Component of IaaS
- Distributes the load across instances.
- Ensures High Availability when one of the instance is down.
  - Redirect all traffic to health instances.

LOAD BALANCER





STORAGE SERVICES ON CLOUD



STORAGE SERVICE : S3

- Simple Storage Service (S3) provides cheaper, highly available, durable and secure storage on cloud.
- Free Trial benefits:
  - 5GB Storage Space
  - 20,000 Get (Read Access)
  - 2,000 Write / Put / Upload reuests

S3 BUCKETS

- Buckets represent the Single “resource” unit .
- Buckets can be accessible to public (Anonymous access) or made private.
- Programmatic Access using SDKs
  - Java
  - Python
  - DOTNET

PLATFORM SERVICES

- RDS (Relational Databases)
- Dynamo DB (Non-relational Databases)
- Elastic BeanStalk (Web applications)
- Lambda Functions

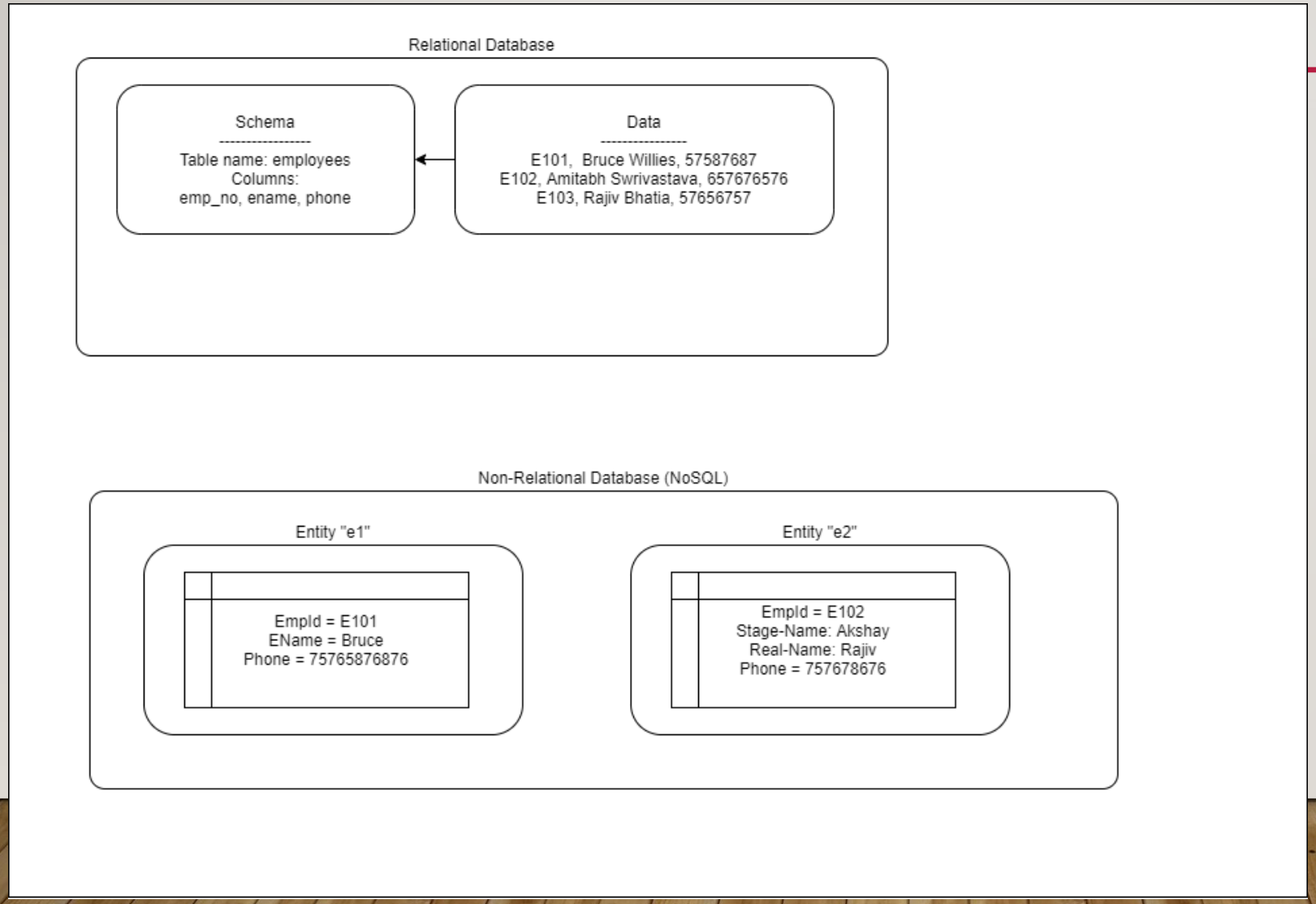
RELATIONAL DATABASES

- Store Data and Schema
- Schema is fixed (difficult to modify once data / records added)
- Popular Relational databases:
  - MySQL
  - MariaDB
  - Oracle DB
  - Microsoft SQL Server

NON-RELATIONAL DATABASES

- No Fixed schema.
- Every record would carry it own schema.
- High Available and Scalable.
- Popular Databases:
  - Mongo DB
  - Casandra
  - Redis
  - etcD

RELATIONAL VS NON-RELATIONAL DATABASES



AMAZON RDS DATABASE

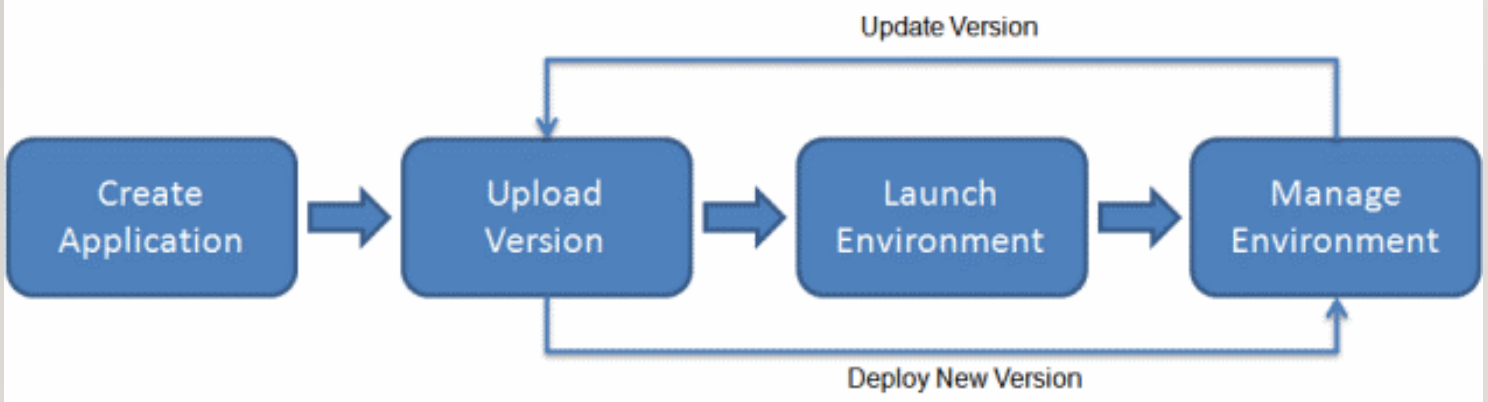
- PaaS service managed by AWS.
- Easy deploy and use for developers.
- Choice of Databases:
  - Aurora (from AWS)
  - MySQL
  - MariaDB
  - MS-SQL
  - Oracle

DYNAMO DB

- NoSQL Offering from AWS
- Highly Scalable
- Natively built by Amazon
- AWS SDK for Programmatic access

ELASTIC BEAN STALK

- Deploy and Managed web-based application on ready to deploy environments.
- No Server Management (Infra)
- Supports CI/CD (Automation)
- Rolling Updates

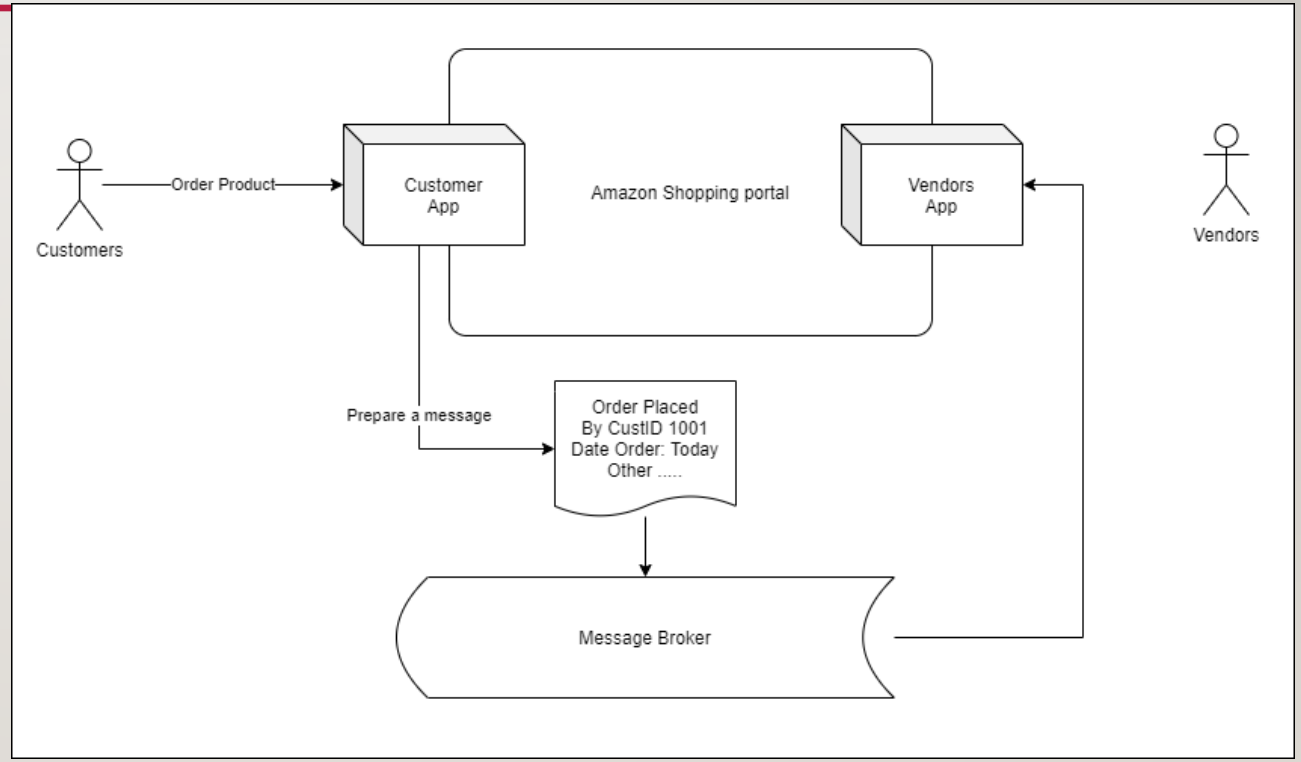


ELASTIC BEANSTALK PLATFORM SUPPORT

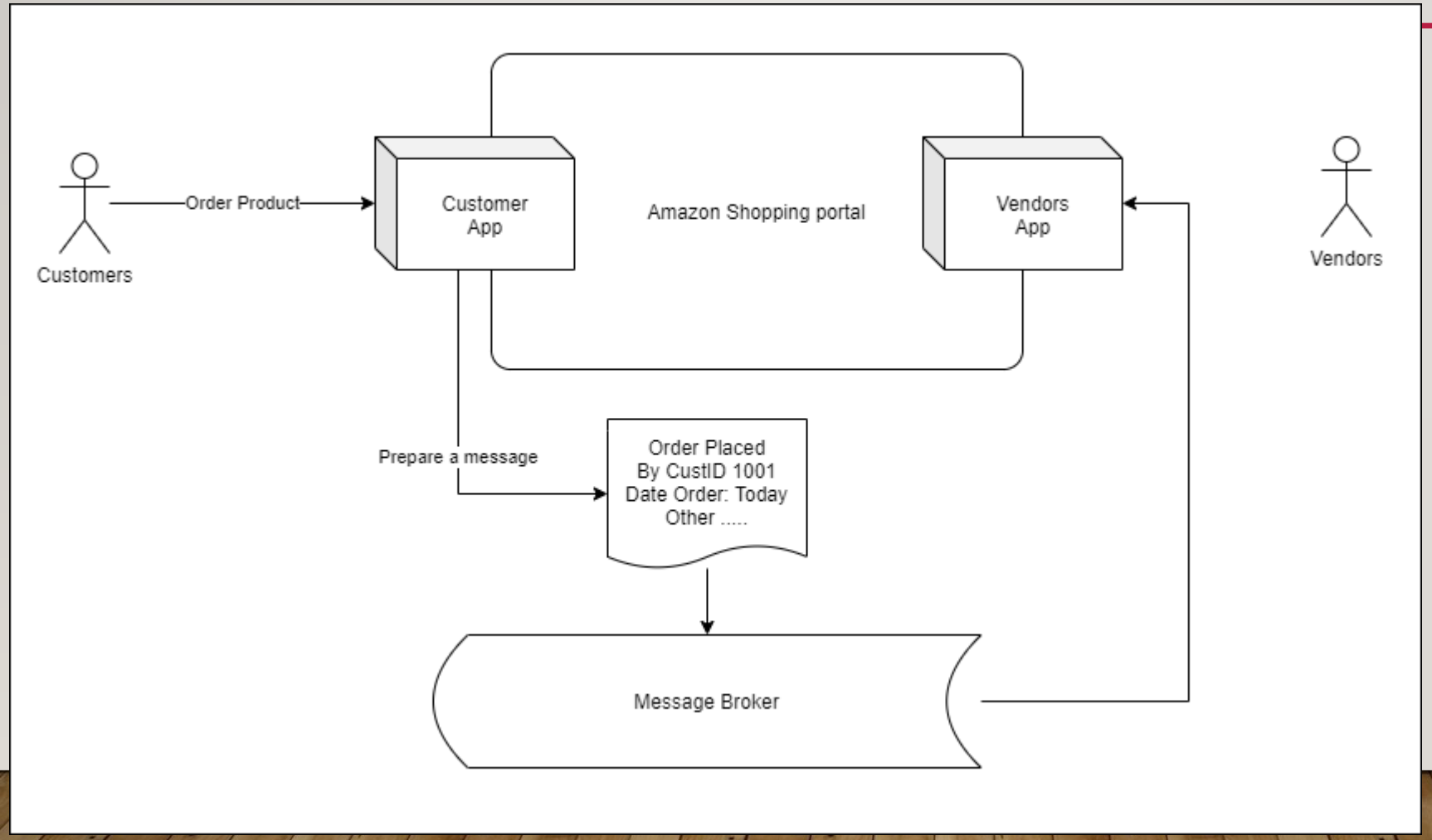
Platform	Description
Java	Java Standard Edition, Supports Jars (Also Spring boot)
Tomcat	Java based application on managed tomcat
.Net core on Linux	.net core applications on linux servers
.Net core on Windows	.net core applications on Windows servers
Go	Applications built with GoLang
Node.JS	NodeJS Based applications (e.g. Angular, React or Plain NodeJS)
Python	Applications built with Python
Docker	Containerized Applications. Supports Docker images

SIMPLE QUEUE SERVICE

- Facilitates "Async" communication between applications.
- Secure, Reliable, Durable and Available Queue Service



SQS USAGE


Customers
Order Product
Customer App
Amazon Shopping portal
Vendors App
Vendors
Prepare a message
Order Placed
By CustID 1001
Date Order: Today
Other .....
Message Broker

AWS SDK FOR JAVA : DEPENDENCIES

```
<!-- https://mvnrepository.com/artifact/software.amazon.awssdk/sqs -->
<dependency>
    <groupId>software.amazon.awssdk</groupId>
    <artifactId>sqs</artifactId>
    <version>2.18.12</version>
</dependency>
```

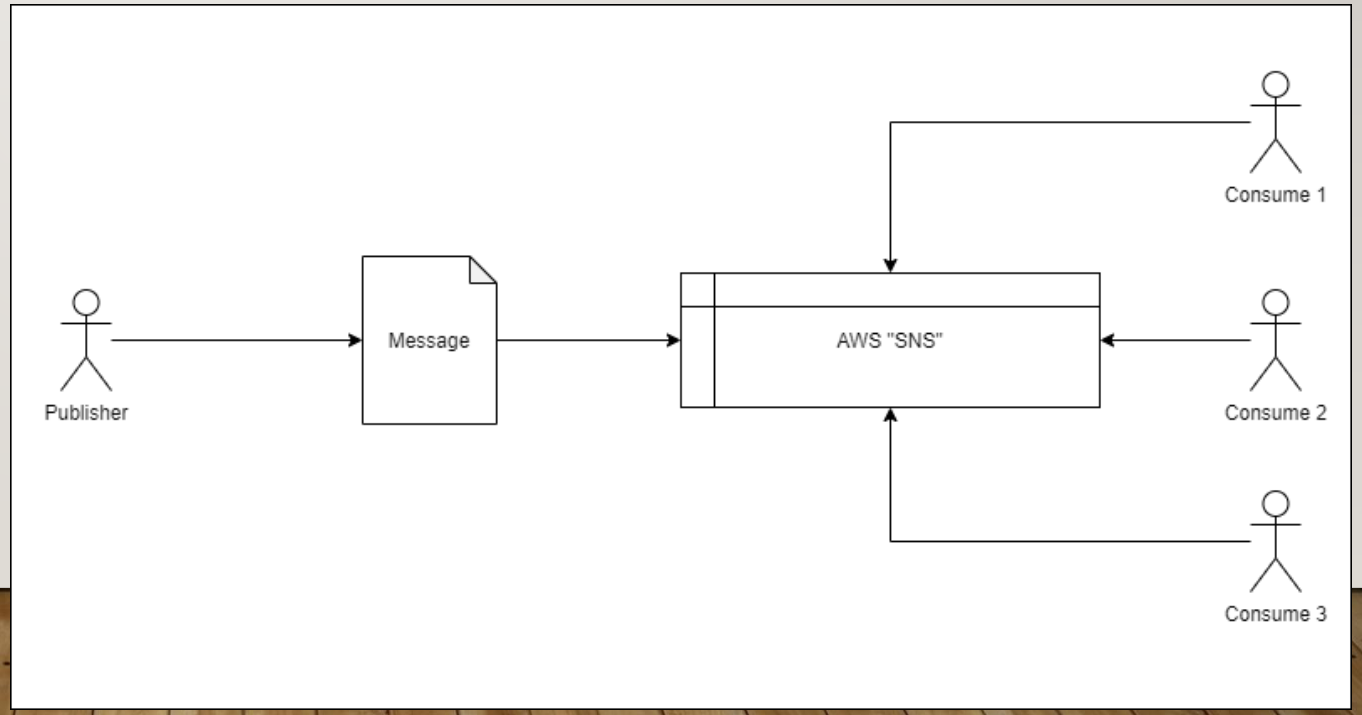
AWS SDK FOR JAVA

```
SqsClient client = SqsClient.builder().region(Region.AP_SOUTH_1)
                .credentialsProvider(ProfileCredentialsProvider.create()).build();
```

```
SendMessageRequest request = SendMessageRequest.builder()
                                                .queueUrl(QUEUE_URL)
                                                .messageBody(msg)
                                                .messageGroupId("group1")
                                                .messageDeduplicationId("d1")
                                                .build();
client.sendMessage(request);
System.out.println("Message sent !!");
```

SIMPLE NOTIFICATION SERVICE

- Application to Application messaging
- Application to person notifications
- Standard and FIFO topics
- Message attributes and filtering



SNS DEPENDENCIES

```
<!-- https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk-sns -->
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-java-sdk-sns</artifactId>
    <version>1.12.337</version>
</dependency>
```

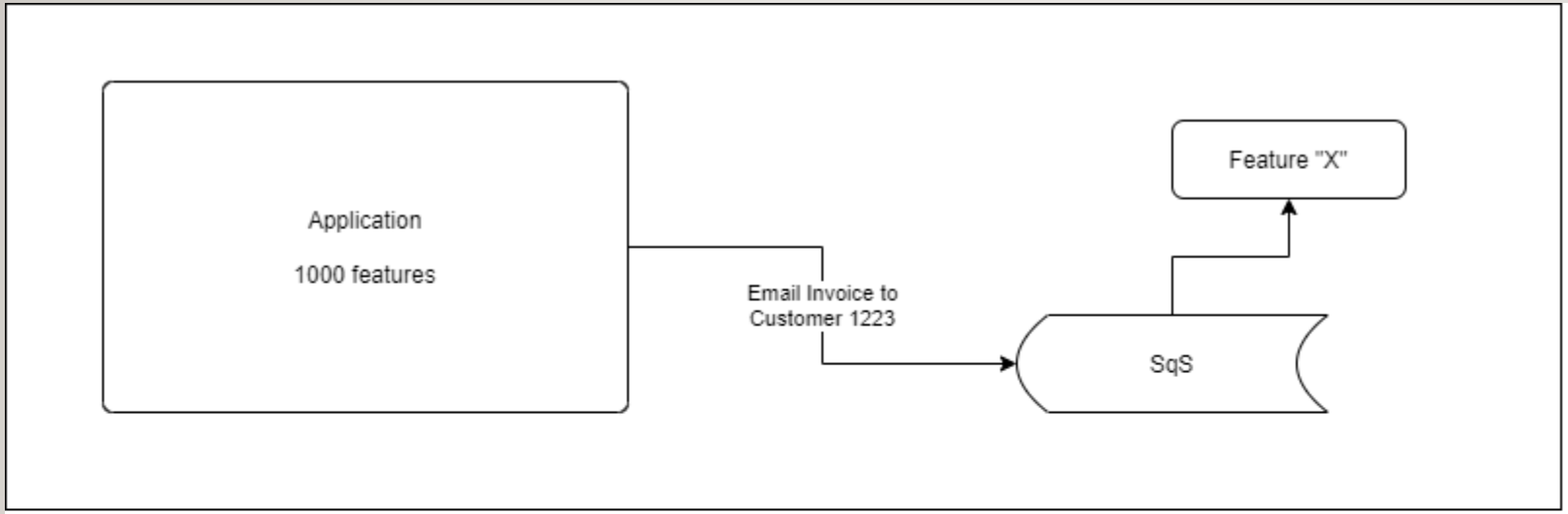
SNS DEMO

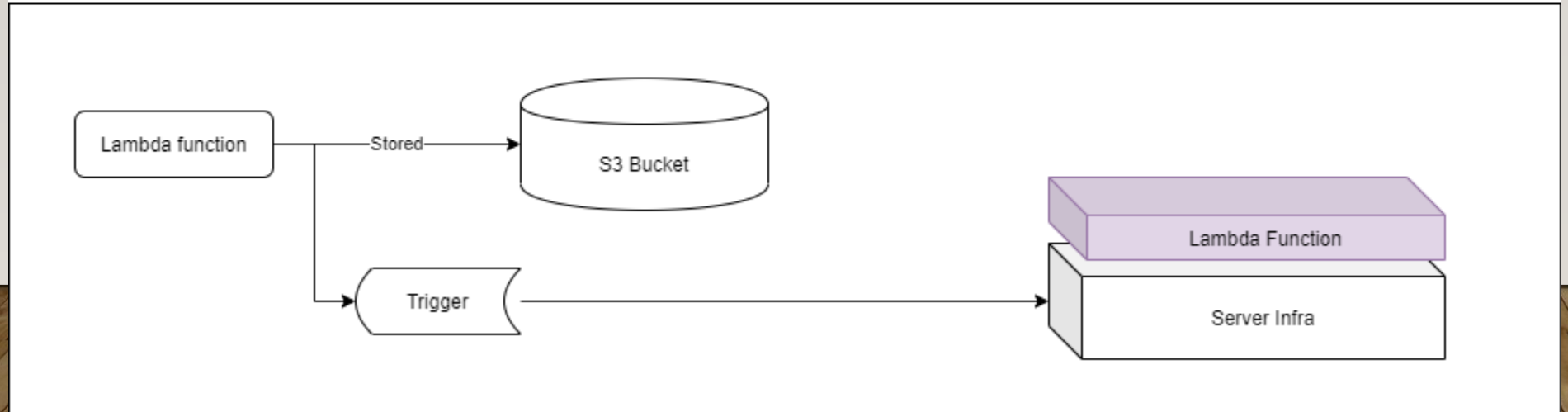
```
AmazonSNS client = AmazonSNSClient.builder().withRegion(Regions.AP_SOUTH_1)
        .withCredentials(new ProfileCredentialsProvider()).build();

try {
    PublishRequest request = new PublishRequest();
    request.setMessage("Hello from Amazon SnS");
    request.setTopicArn("arn:aws:sns:ap-south-1:851889547214:news");
    System.out.println(request.getMessage());
    PublishResult response = client.publish(request);
    System.out.println("Response : " + response.toString());
    client.shutdown();
} catch (AmazonSNSException ex) {
    ex.printStackTrace();
}
```

AWS LAMBDA

- Run code without server provisioning.
- Pay only for compute times.
- Zero Administration (of servers)
- SDK to build using Java, DotNet, python etc.
- Built in Triggers to invoke functions on-demand.

LAMBDA USE-CASE





BASIC (JAVA) LAMBDA FUNCTION

```
public String handleRequest(Map<String,String> event, Context context) {
    // Add log to AWS Console (Lambda Logs)
    LambdaLogger log = context.getLogger();
    String response =" Welcome to My Lambdas !";
    log.log("Processing the request....");
    log.log("Context: "+context.toString());
    log.log("Event :"+event.toString());
    return response;
}
```

LAMBDA USING SPRING BOOT

```
@Bean
public Function<String,String> uppercase(){
    return value -> {
        if(value.equals("exception")) {
            throw new RuntimeException("The error ");
        }
        else {
            return value.toUpperCase();
        }
    };
}
```

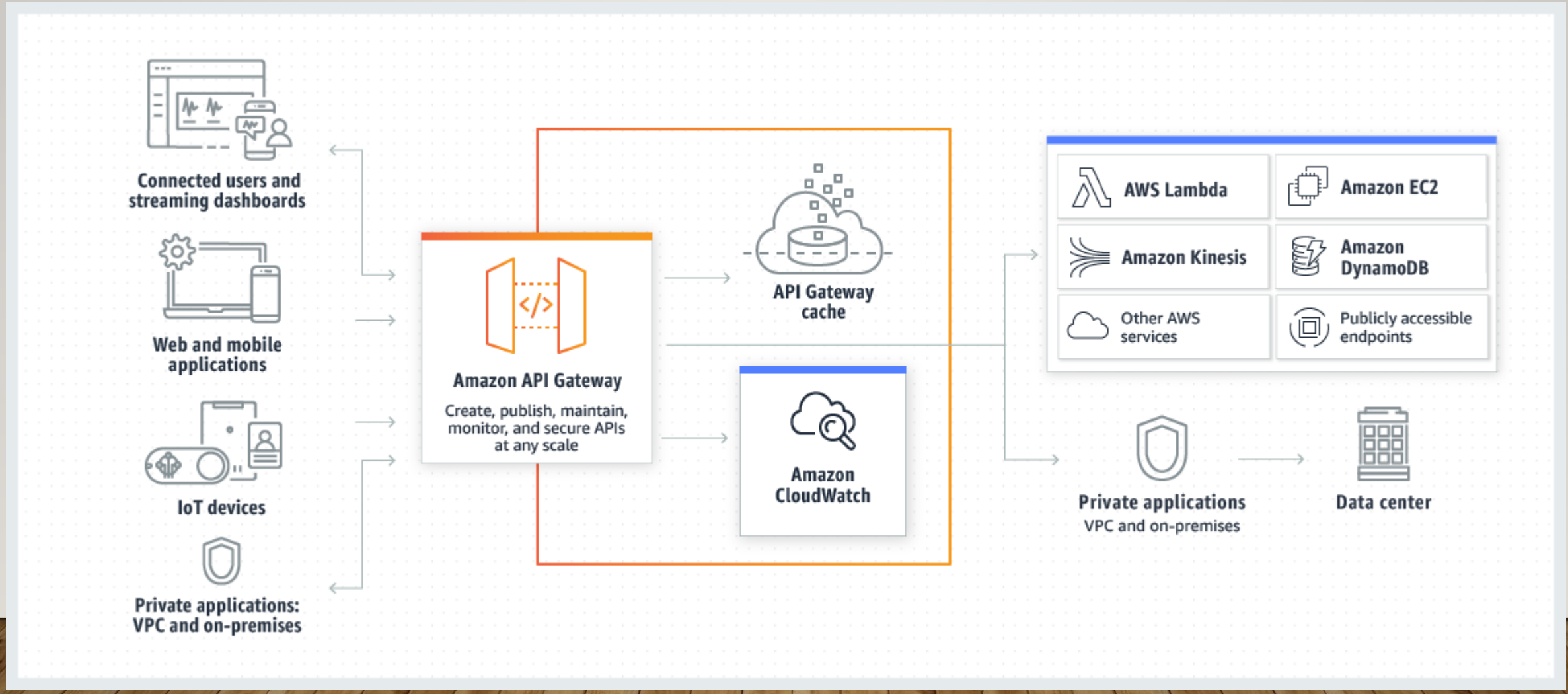
LAMBDA FOR JAVA DEPENDENCIES

```
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-lambda-java-core</artifactId>
    <version>1.2.1</version>
</dependency>
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-lambda-java-log4j2</artifactId>
    <version>1.5.1</version>
</dependency>
```

LAMBDA WITH SPRING BOOT

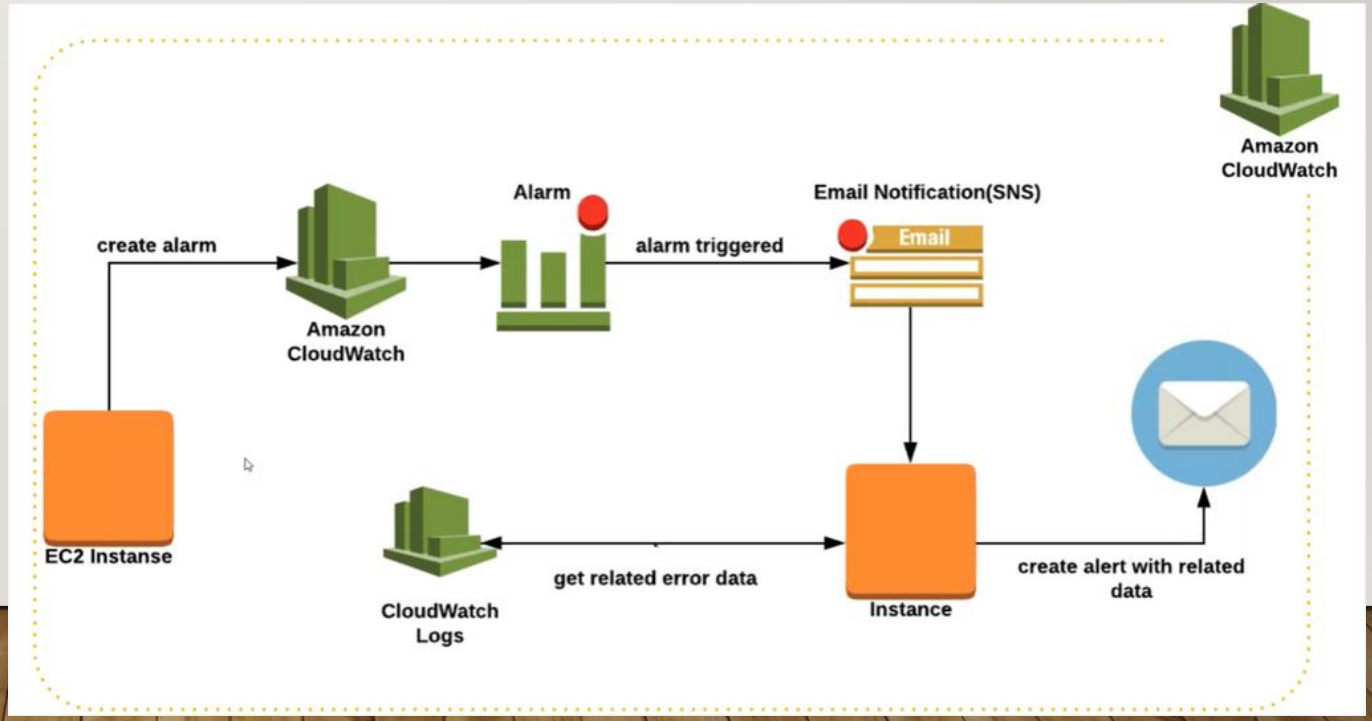
```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-function-context</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-function-adapter-aws</artifactId>
</dependency>
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-lambda-java-core</artifactId>
    <version>1.2.1</version>
</dependency>
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-lambda-java-log4j2</artifactId>
    <version>1.5.1</version>
</dependency>
```

API GATEWAYS

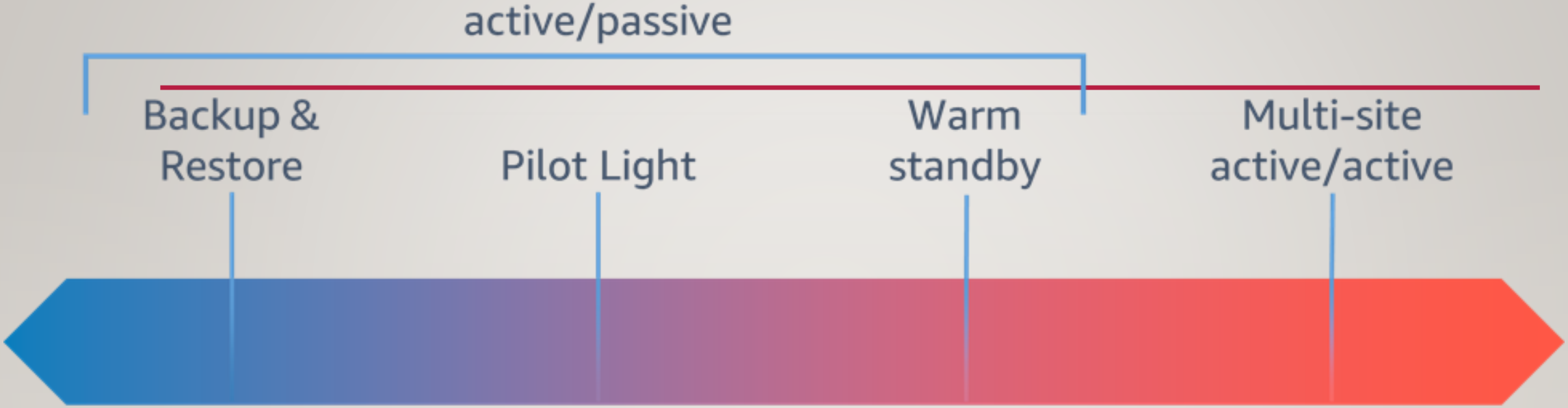


MONITORING WITH CLOUDWATCH

- Regional Service to Monitor all AWS Services / Resources.
- Collects, Tracks and Report Metrics
- Customizable Dashboards
- Alarms to capture events that need attention.



DISASTER RECOVERY OPTION



RPO / RTO: Hours

- Lower priority use cases
- Provision all AWS resources after event
- Restore backups after event
- Cost $

RPO / RTO: 10s of minutes

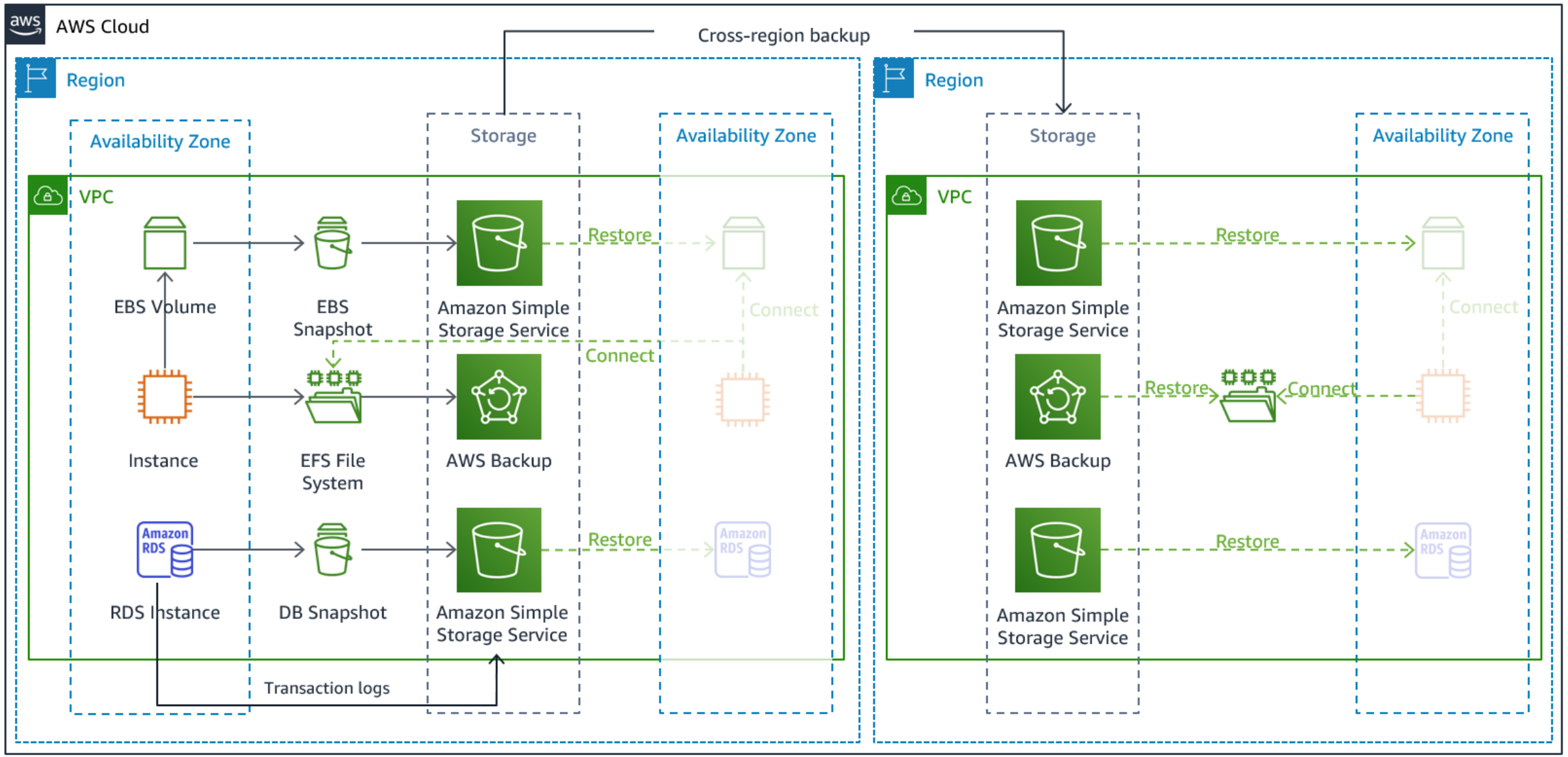
- Data live
- Services idle
- Provision some AWS resources and scale after event
- Cost: $$

RPO / RTO: Minutes

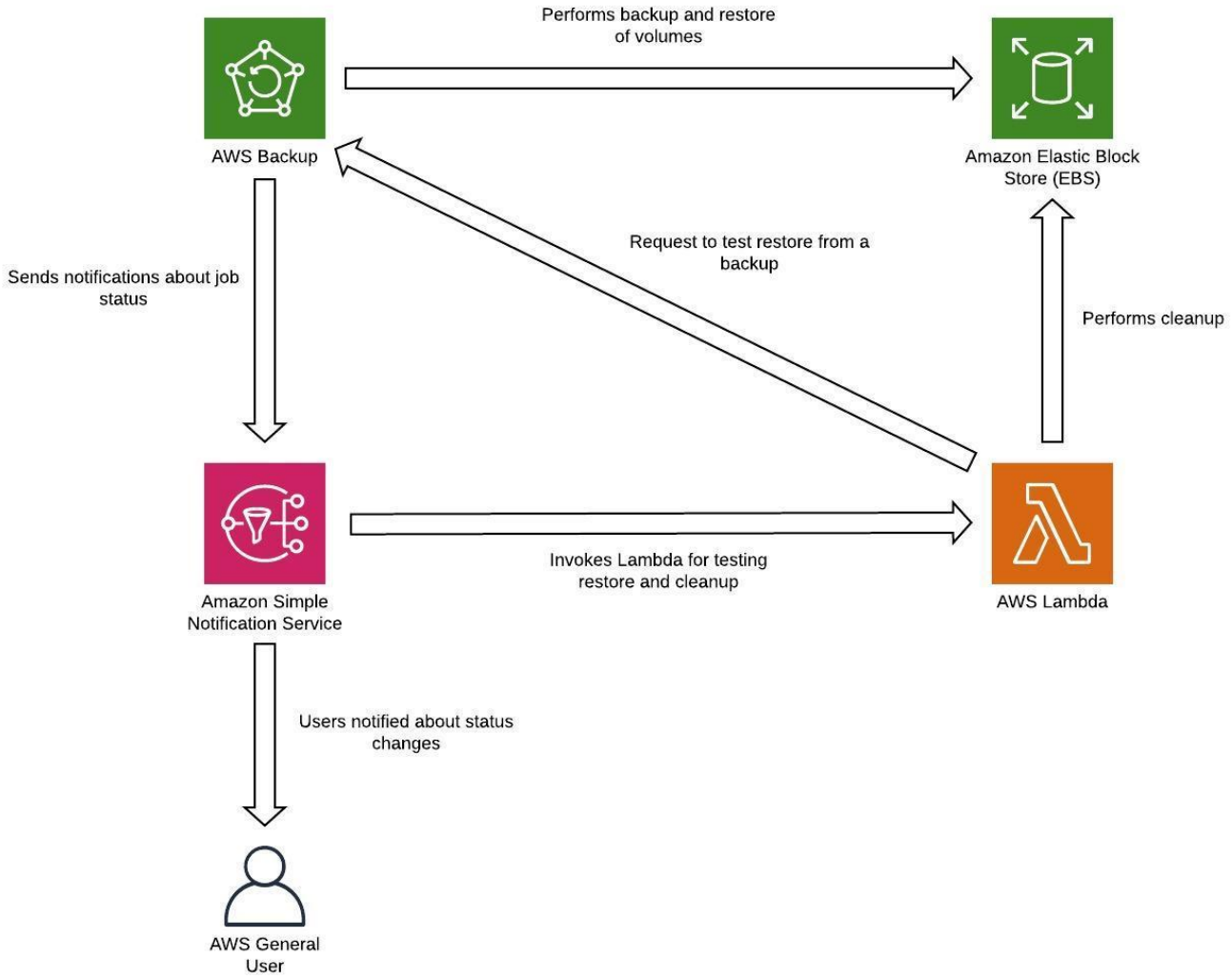
- Always running, but smaller
- Business critical
- Scale AWS resources after event
- Cost $$$

RPO / RTO: Real-time

- Zero downtime
- Near zero data loss
- Mission Critical Services
- Cost $$$$


AWS Cloud
Cross-region backup
Region
Availability Zone
Storage
Availability Zone
VPC
EBS Volume
EBS Snapshot
Amazon Simple Storage Service
Restore
Connect
Instance
EFS File System
AWS Backup
Amazon RDS
RDS Instance
DB Snapshot
Amazon Simple Storage Service
Transaction logs
Region
Storage
Availability Zone
VPC
Amazon Simple Storage Service
Restore
Connect
AWS Backup
Amazon Simple Storage Service

AWS BACKUP


Performs backup and restore of volumes
AWS Backup
Amazon Elastic Block Store (EBS)
Request to test restore from a backup
Sends notifications about job status
Performs cleanup
Amazon Simple Notification Service
Invokes Lambda for testing restore and cleanup
AWS Lambda
Users notified about status changes
AWS General User

AWS ELASTIC DISASTER RECOVERY