AWS Foundation

Introduction to Load Balancing, Autoscaling, and Route 53

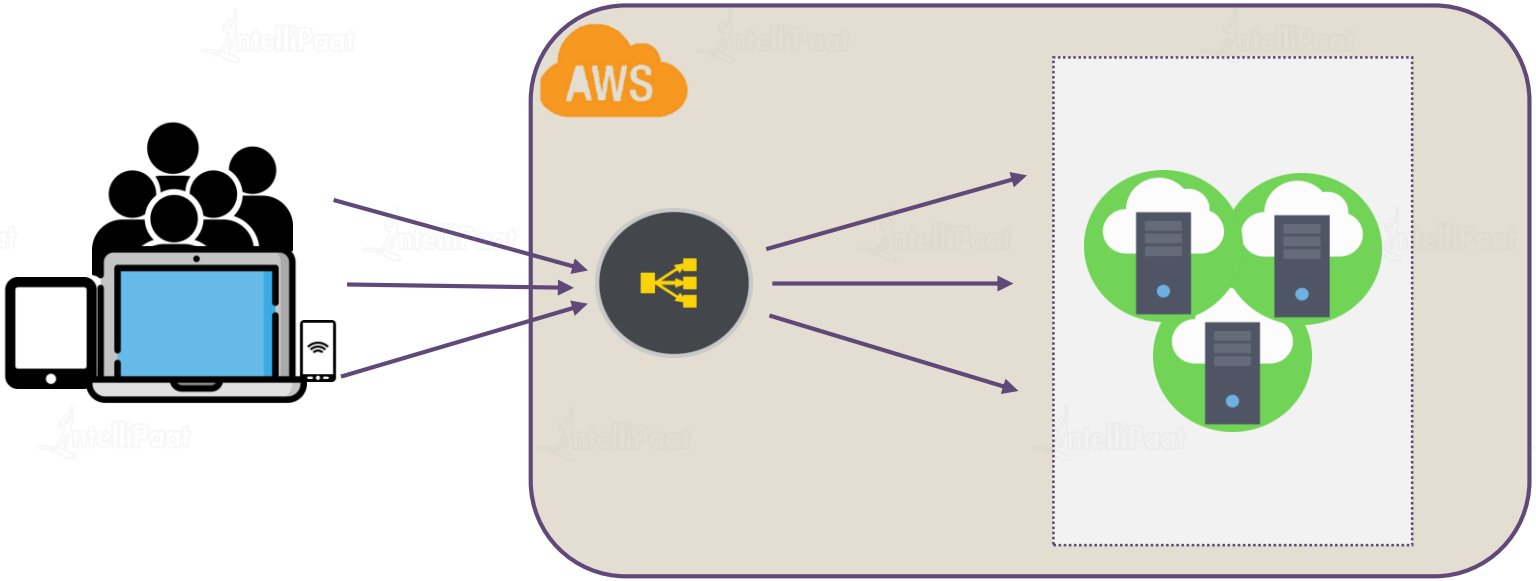
Agenda

1. Introduction to Elastic Load Balancer
2. Types of ELB: Classic, Network, & Application
3. Load Balancer Architecture
4. Demo 1 - Classic
5. Demo 2 - ALB
6. Demo 3 - NLB
7. Cross-zone Load Balancing
8. Introduction to Autoscaling
9. Vertical & Horizontal Scaling
10. Lifecycle of Autoscaling
11. Components of Autoscaling
12. Scaling Policy
13. Instance Termination
14. Using Load Balancer with Autoscaling with Demo
15. Pre-Route 53: How DNS Works, Name Record, CNAME, Alias, & Latency
16. Routing Policy
17. Route 53 Terminology

Introduction to ELB

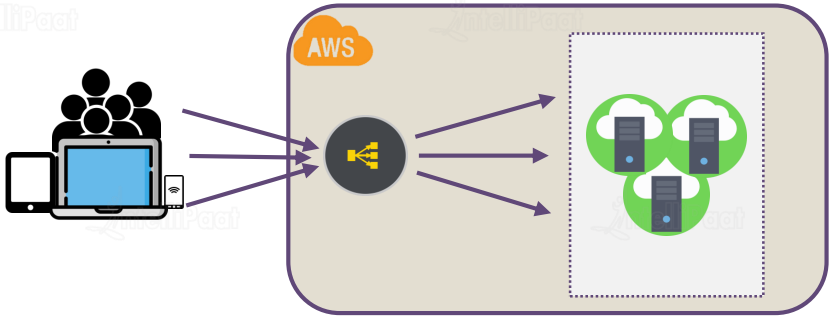
Load Balancer

Load balancer is a service that uniformly distributes the network traffic and workloads across multiple servers or a cluster of servers. Load balancer increases the availability and fault tolerance of an application



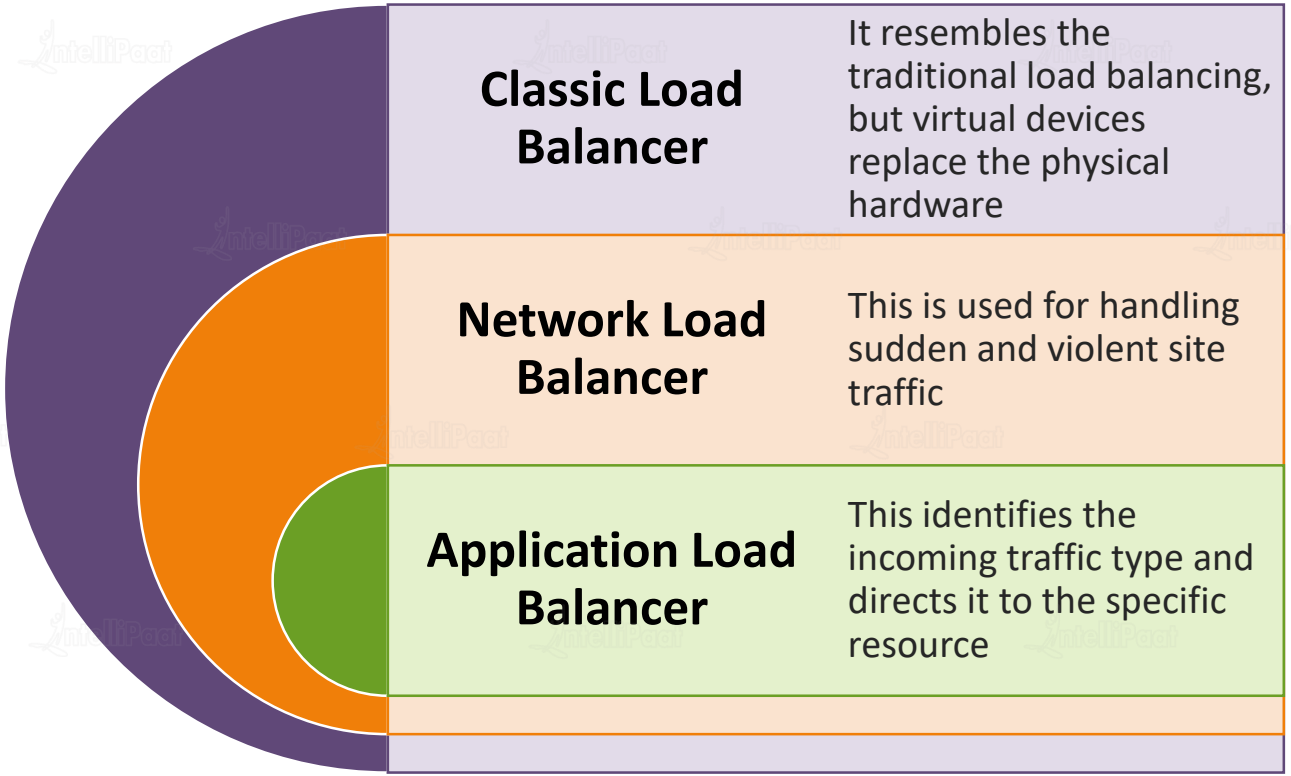
Elastic Load Balancer

- Elastic Load Balancer (ELB) is a load balancing service for the AWS deployments
- ELB scales itself as necessary to handle the load
- Incoming traffic is distributed across EC2 instances in multiple availability zones
- Load balancer is the single point of contact for clients



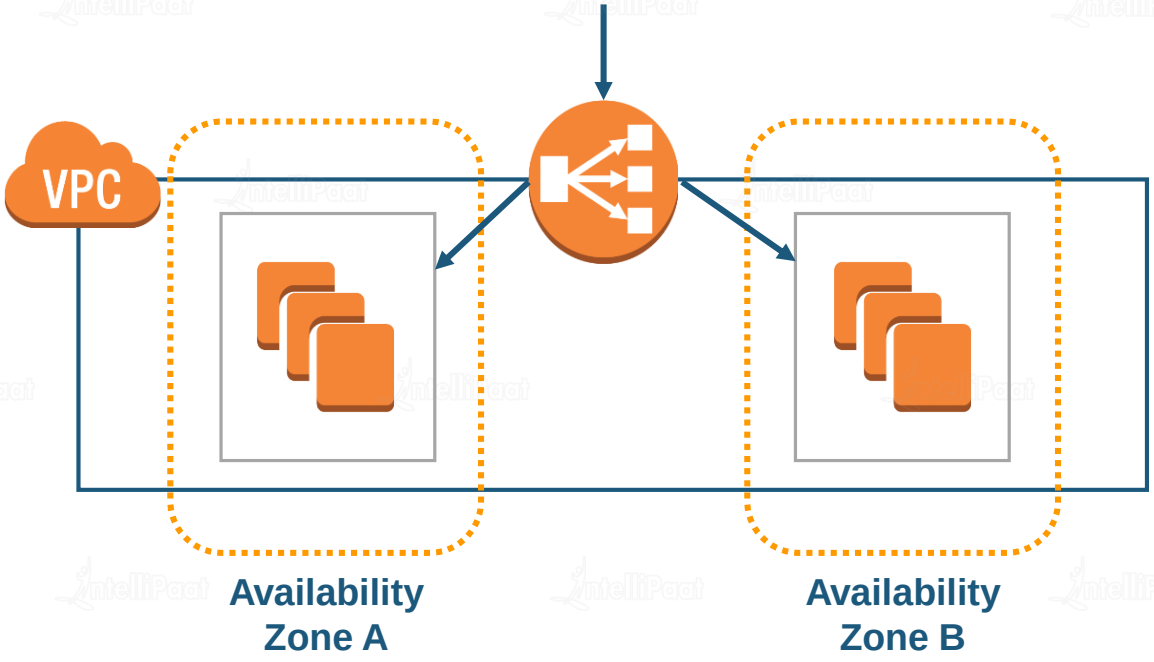
Types of Elastic Load Balancer (ELB)

Types of Elastic Load Balancer (ELB)



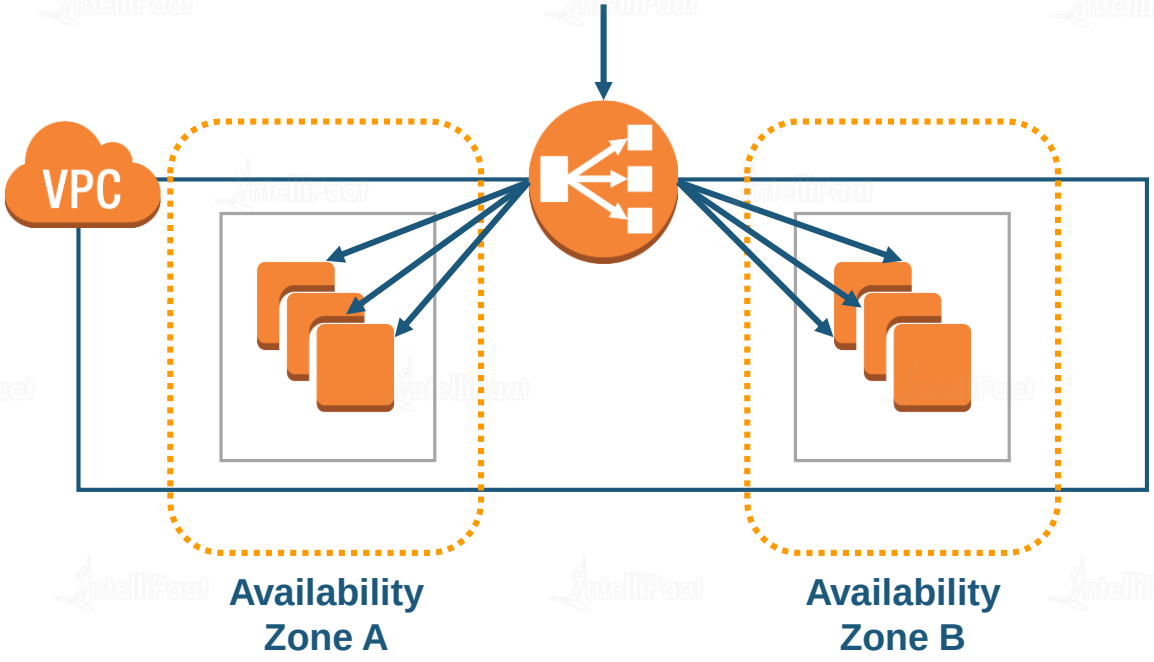
Classic Load Balancer

- Distributes the incoming application traffic across EC2 instances in multiple AZs and functions at Layer 7 of the OSI model
- Routes the traffic to healthy instances only, and it is evenly distributed
- Internet and Internal-facing load balancer



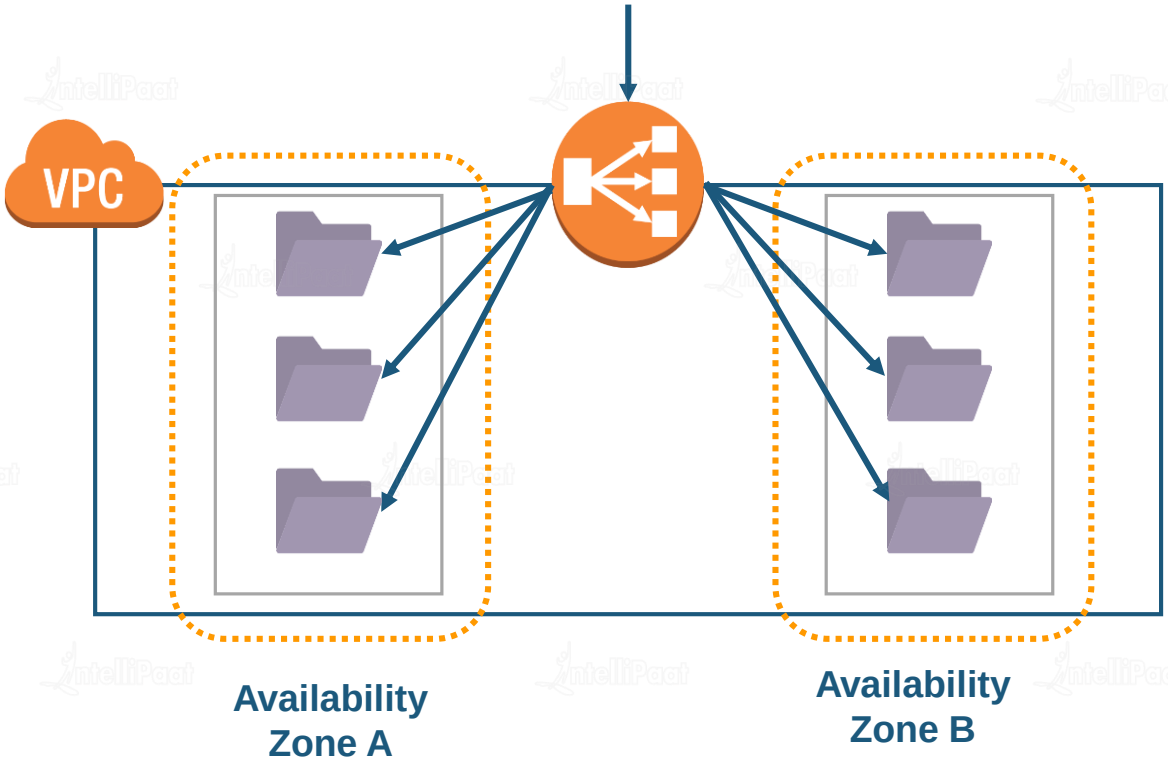
Network Load Balancer

- Functions at the 4th layer of the OSI model
- Handles millions of requests per second and maintains low latency
- Ideal for load balancing the TCP traffic and supports elastic or static IP



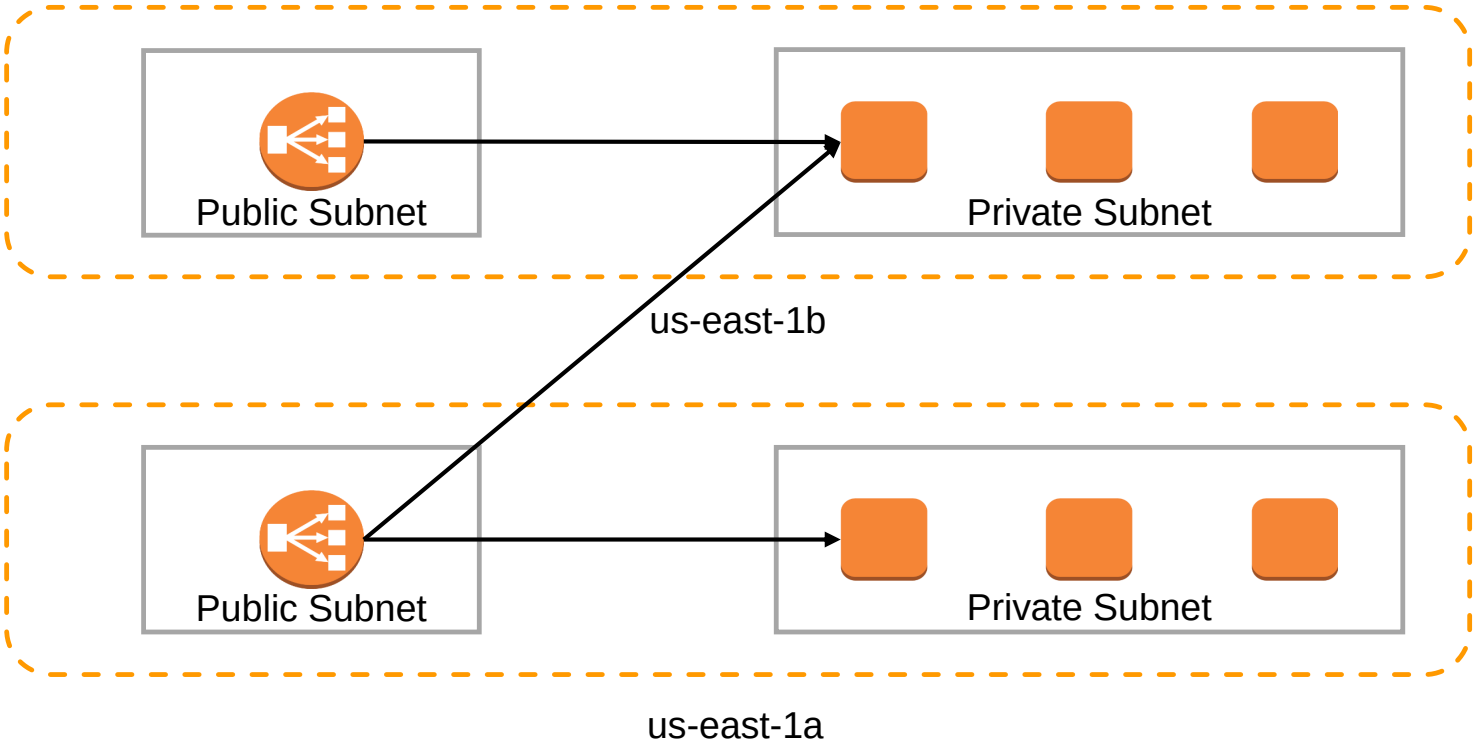
Application Load Balancer

- Functions at the 7th layer of the OSI model
- Identifies the incoming resource type and directs it to the appropriate server



Load Balancer Architecture

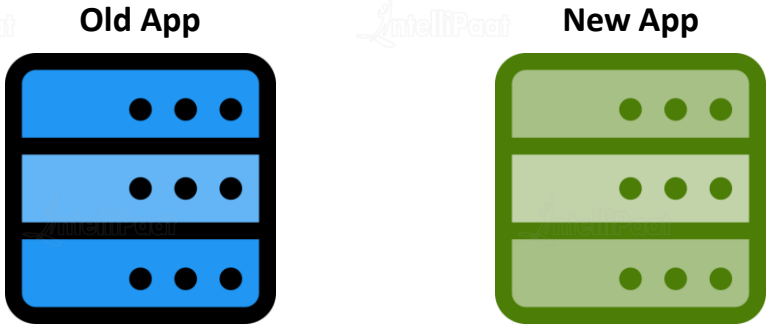
Load Balancer Architecture



Blue/Green Deployments Using Weighted Routes

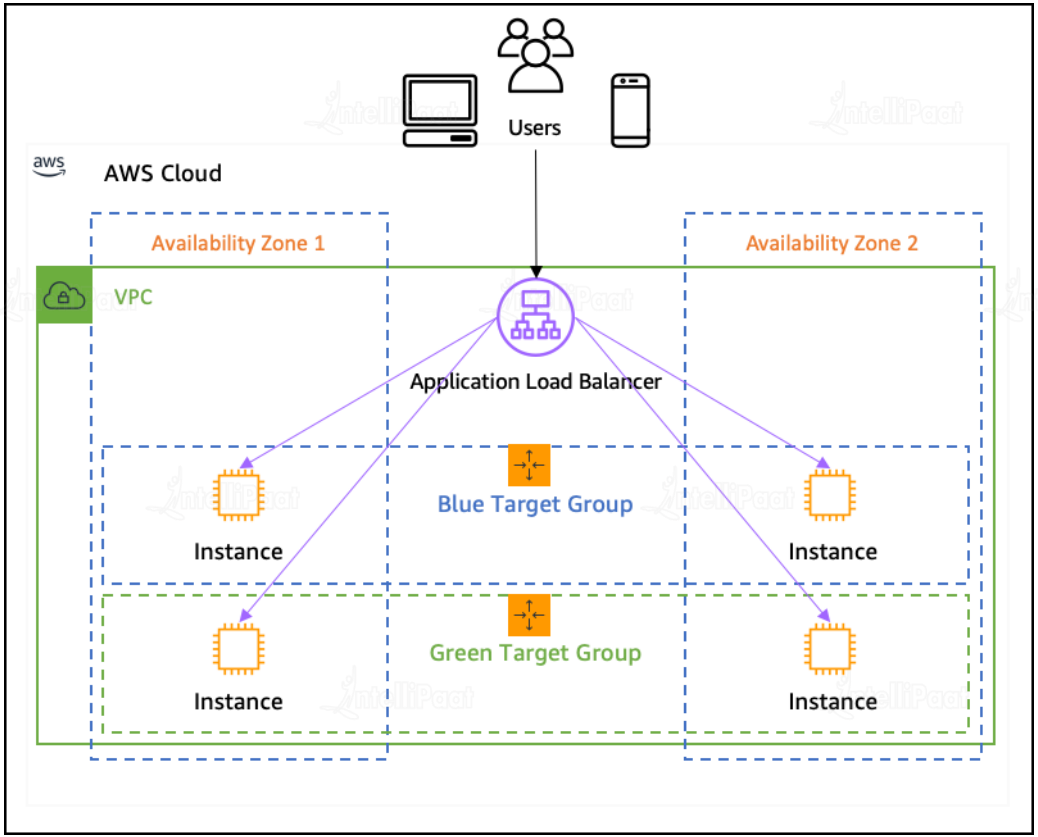
Blue/Green Deployments with Weighted Routes

With weighted routing, we can switch the traffic between the versions of our application. This configuration allows us to control the distribution of the traffic to our application



For example, if we define a rule having two target groups with weights 8 and 2, the load balancer will route 80% of the traffic to the first target group and 20% to the other

Blue/Green Deployments with Weighted Routes



Demo 1

Demo 1: Creating Load Balancers

Classic Load Balancer Creation

1. Open AWS Management Console; click on the Services drop-down box, and choose EC2
2. Scroll down, and choose ‘Load Balancers’
3. Select ‘Create load balancer,’ and choose Classic
4. Configure all the settings one by one: Give a name; create a new VPC; add a tag; choose review, and then choose launch
5. Review (optional), and choose launch
6. The Classic load balancer is successfully created

Demo 2

Demo 2: Creating Load Balancers

Application Load Balancer Creation

1. Open AWS Management Console; click on the Services drop-down box and choose EC2
2. Scroll down, and choose ‘Load Balancers’
3. Select ‘Create load balancer,’ and choose Application
4. Configure all the settings one by one: Give a name; create a new VPC; add a tag; choose review, and then choose launch
5. Review (optional), and choose launch
6. The Application load balancer is successfully created
7. Also, use weighted routing in the listener rules page

Demo 3

Demo 3: Creating Load Balancers

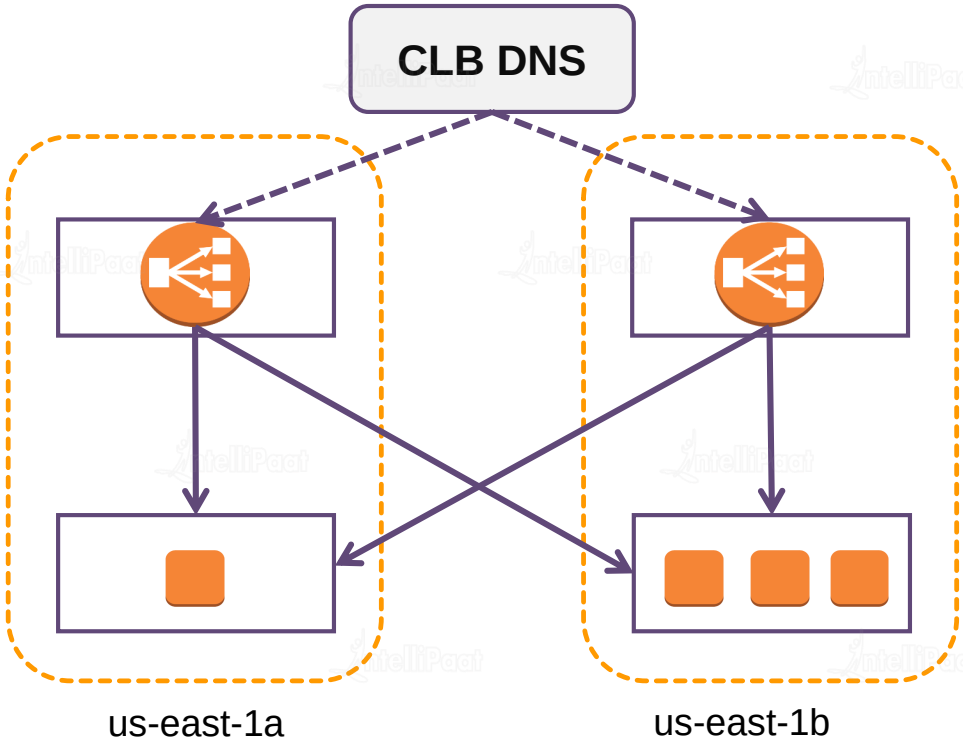
Network Load Balancer Creation

1. Open AWS Management Console; click on the Services drop-down box, and choose EC2
2. Scroll down, and choose ‘Load Balancers’
3. Select ‘Create load balancer,’ and then choose Network
4. Configure all the settings one by one: Give a name; create a new VPC; add a tag; choose review, and then choose launch
5. Review (optional), and choose launch
6. The Network load balancer is successfully created

Cross-zone Load Balancing

Cross-zone Load Balancing (CLB)

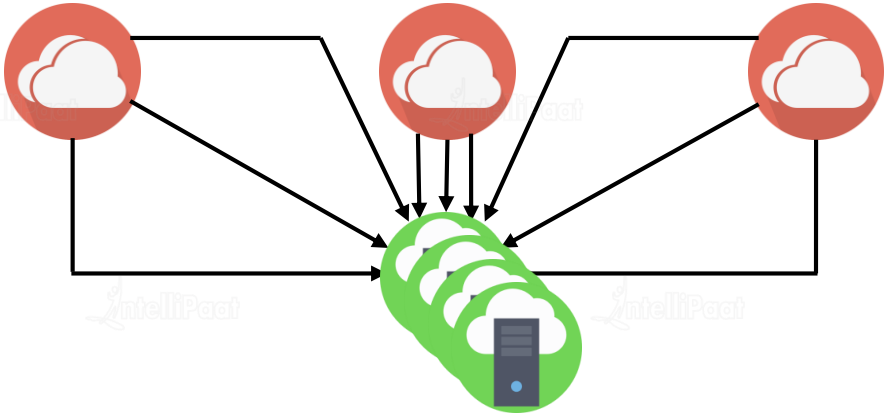
- By default, CLB nodes distribute the traffic to instances in their availability zone only
- We enable cross-zone load balancing to route evenly across EC2 instances
- CLB routes each request to the instance with the smallest load



Introduction to Autoscaling

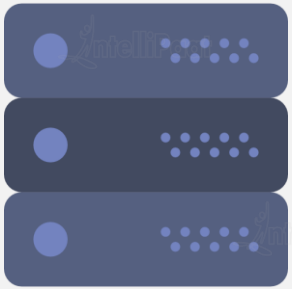
Introduction to Autoscaling

- Scaling is the process of adding/removing capacity/resources as needed
- Scale out is adding the capacity/resources
- Scale in is removing the capacity/resources
- Types: Vertical and Horizontal



Introduction to Autoscaling

- Scaling Types: Vertical and Horizontal
- Vertical:
  - CPU: 2.0 GHz to 3.2 GHz
  - RAM: 1024 GB to 2048 GB
  - N/W Bandwidth: 4 Gbps to 10 Gbps



- Horizontal:
  - CPU: 1 server with 1.0 GHz to 3 servers with 1.0 GHz
  - RAM: 1 server with 500 GB to 3 servers with 500 GB

Introduction to Autoscaling

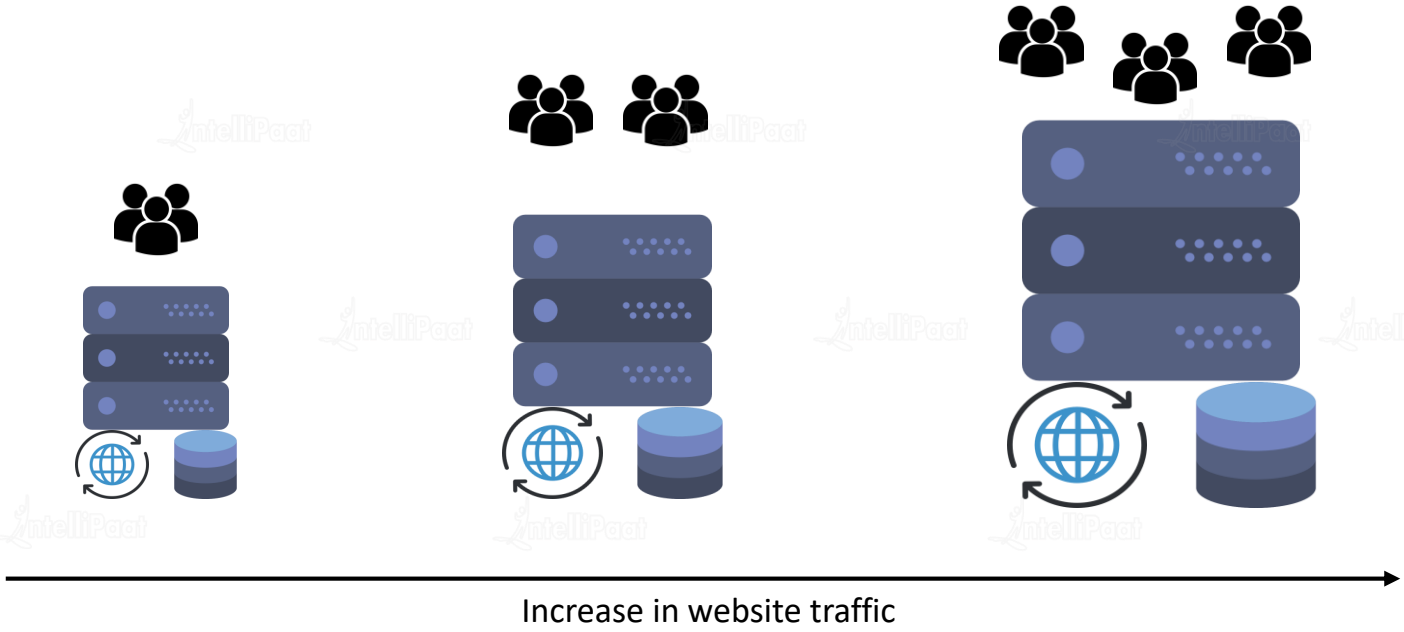
- Autoscaling is scaling out/in automatically without any manual intervention
- It helps ensure that the correct number of EC2 instances are available to handle the load
- Multi-AZ EC2 instances provide high availability solutions

- Autoscaling can dynamically increase or decrease capacity as needed

Vertical & Horizontal Scaling

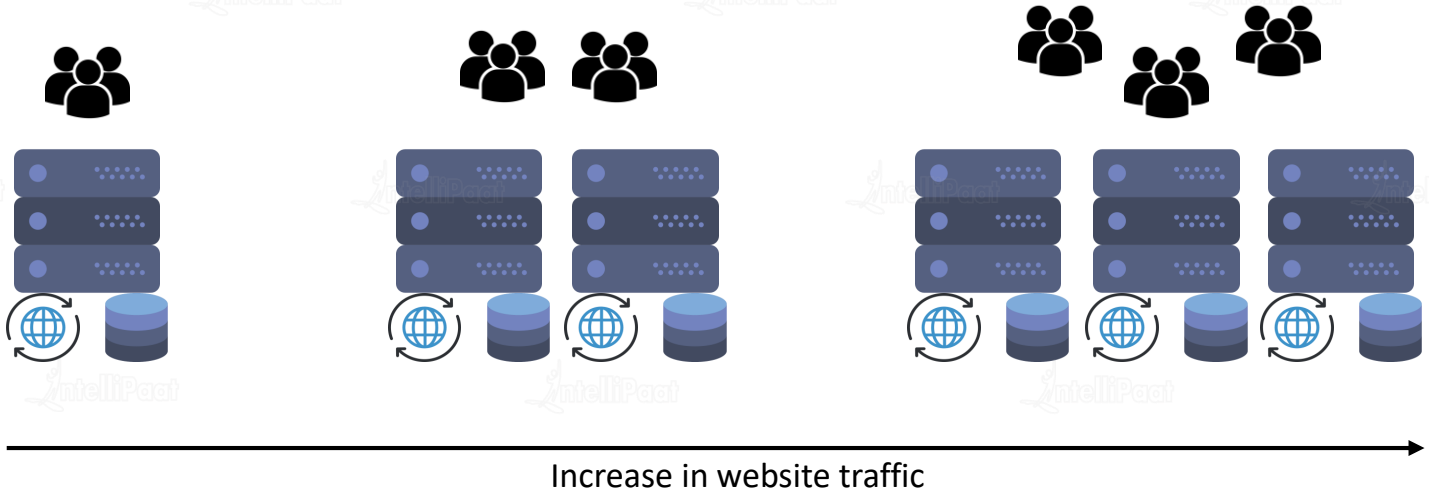
Vertical and Horizontal Scaling

Vertical Scaling



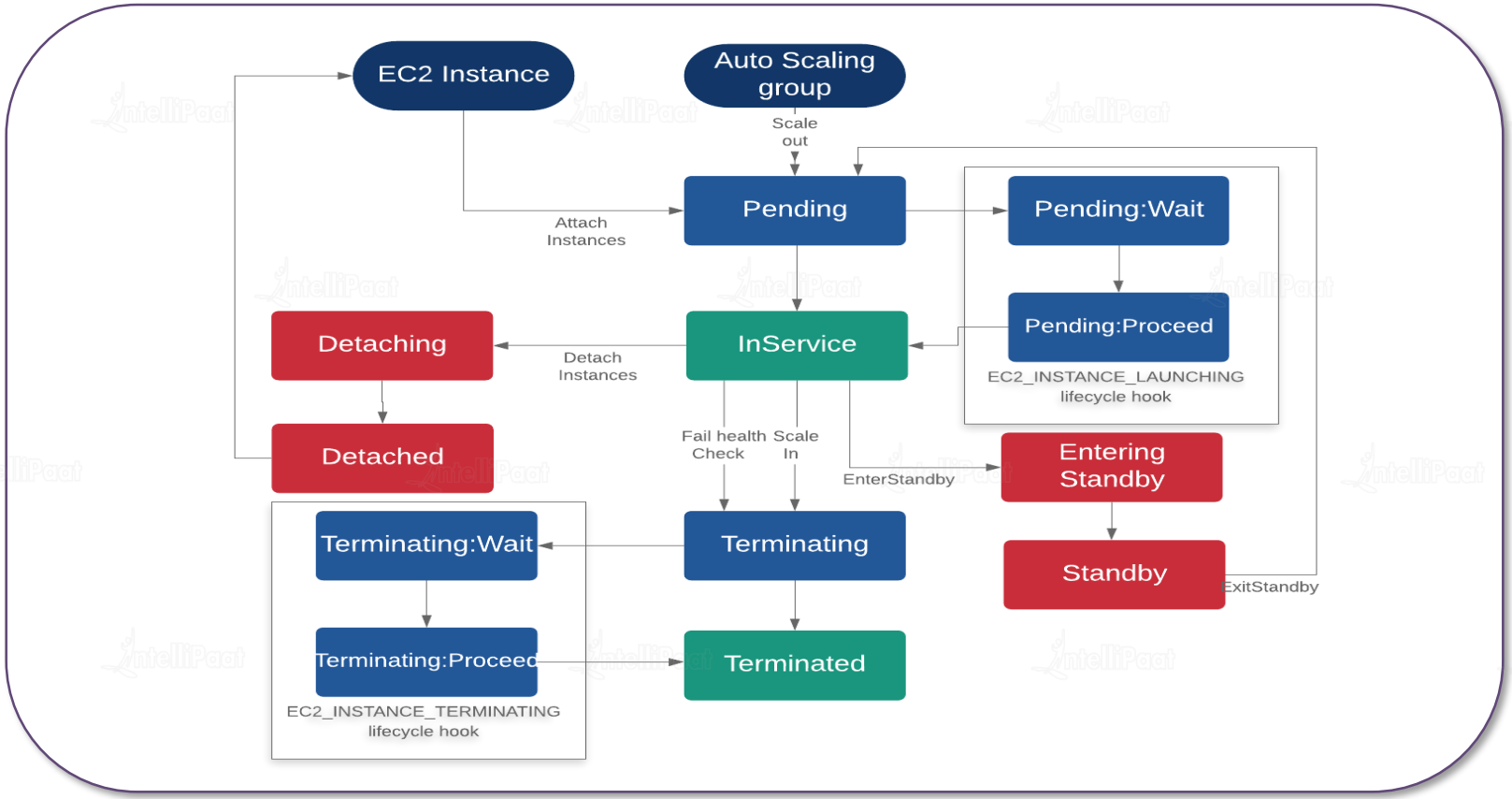
Vertical and Horizontal Scaling

Horizontal Scaling



Lifecyle of Autoscaling

Lifecycle of Autoscaling



Components of Autoscaling

Autoscaling Components

Groups

- EC2 instances are in groups so that they can be considered as a logical unit (for scaling and management)
- When we create a group, we can mention the following attributes: The max, min, and desired number of instances

Configuration Templates

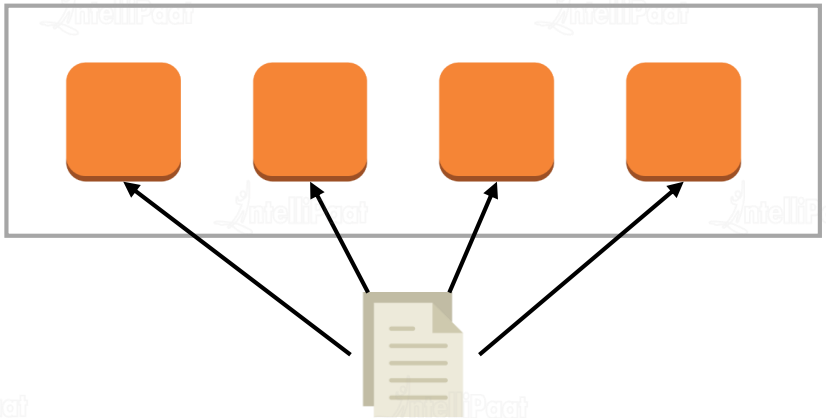
- These are used as configuration templates for the EC2 Instances
- Launch template or Launch configuration is also used

Scaling Options

- Autoscaling provides several ways to scale the group
- Manual scaling
- Dynamic scaling
- Scaling based on demand or schedule

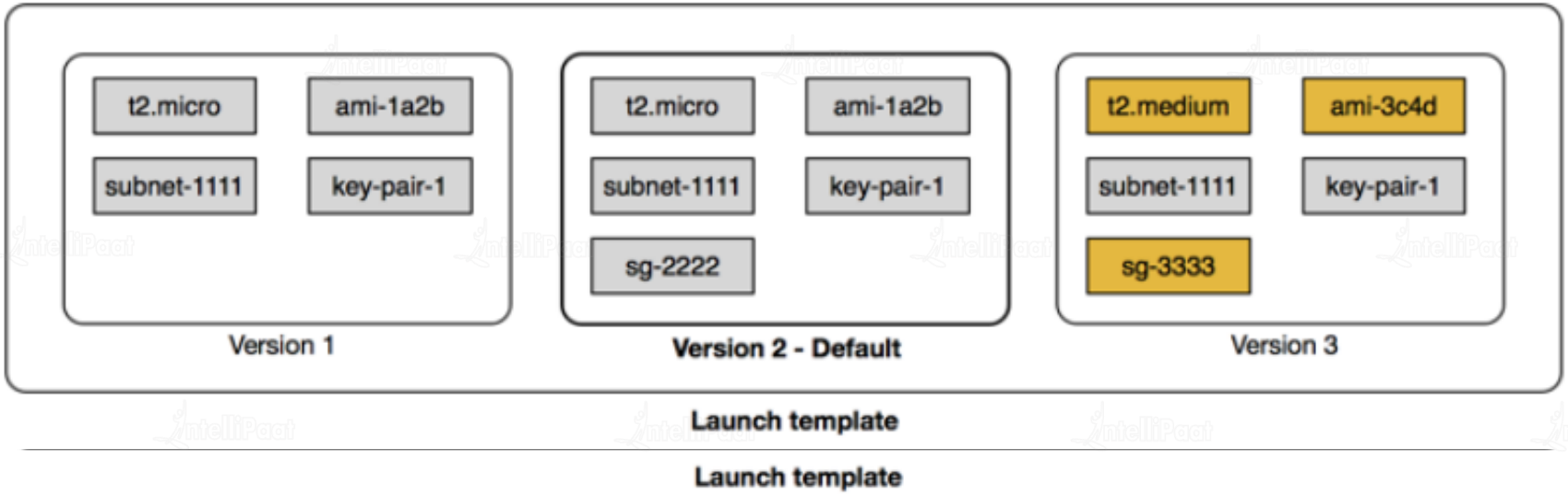
Autoscaling Groups

- An autoscaling group contains a collection of EC2 instances that are exactly the same
- While creating an autoscaling group, the launch configuration must be specified
- After specifying, the launch configuration cannot be changed
- New instances are launched using a new configuration
- EC2 instances are launched and terminated using scaling policies



Configuration Templates

Launch template can also be used with autoscaling groups



Configuration Templates

- **Launch configuration** is a template that is used to launch EC2 instances for the autoscaling purpose
- Autoscaling groups (the next topic) use launch configuration to launch instances
- Launch configuration cannot be modified after creation
- It can be created in two ways:
  - From scratch: Image ID, instance type, storage devices, etc.
  - From an EC2 instance: Attributes from the instance are copied. Block device mapping of the AMI is included. Any additional device that was attached after launching the instances is not considered in the launch configuration

Scaling Options: Dynamic Scaling

- Scaling policies and alarms:
  - Scaling policies mention how to scale, and alarms decide when to scale
  - CloudWatch alarms are set to monitor individual metrics, e.g., CPU utilization, etc.
  - When the threshold is breached, scaling policies are executed
- Minimum, maximum, and the desired capacity

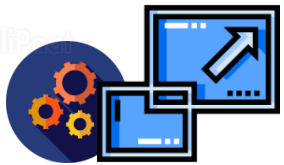
Scaling Policy:

- Increase 2 instances at a time
- Decrease 1 instance at a time

Alarm:

If CPU utilization > 80% for more than 10 mins, ring the bell

- Minimum capacity: 2
- Desired capacity: 4
- Maximum capacity: 10



Other Scaling Options

- Scaling based on a schedule: This type of a scaling method is used to scale at a given time and date
- Scaling based on demand: Here, scaling occurs when the CPU utilization of the current running instances grows beyond a fixed usage limit

Scaling based on a schedule:

- Increase the instances by 2 at 2:30 pm today
- Decrease the instances by 1 at 12:00 am tomorrow

Scaling based on demand:

- If CPU utilization > 80% for more than 10 mins, increase the instance by 1
- If CPU utilization < 50% for more than 5 mins, decrease the instances by 2

Scaling Policy

Scaling Policy

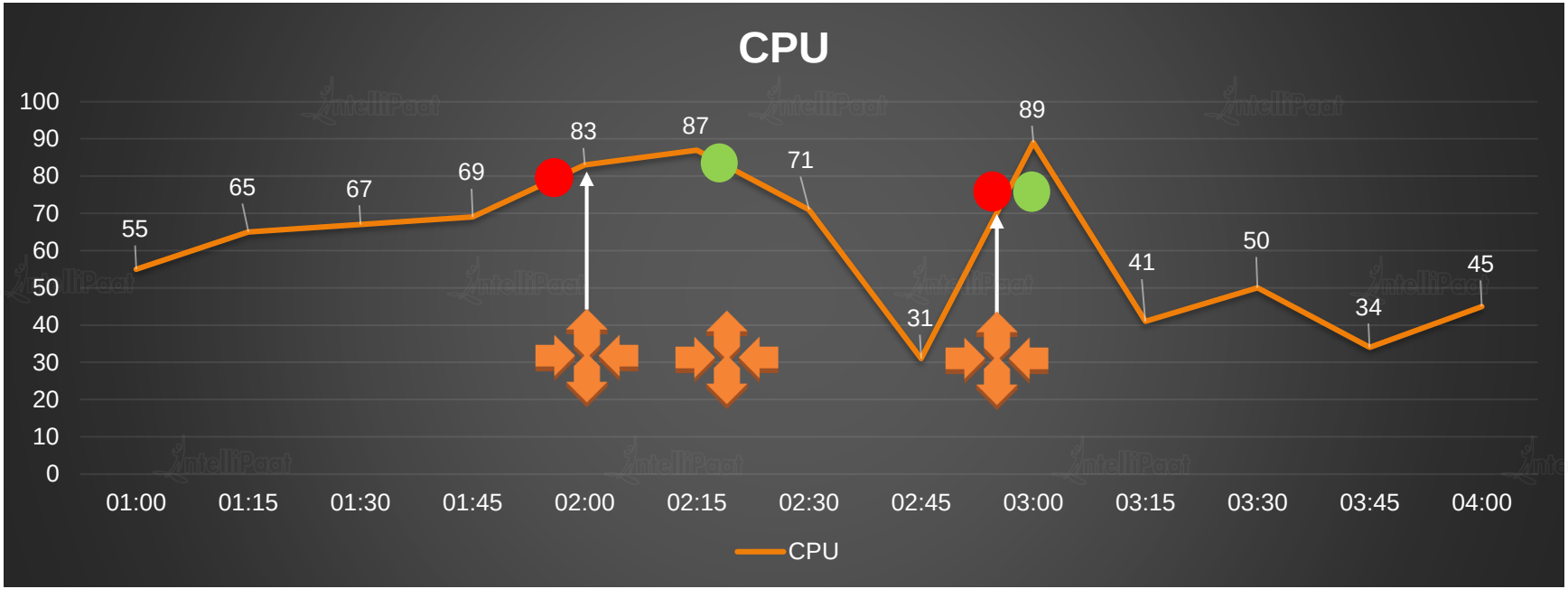
Scaling Policy:

- Increase 2 instances at a time
- Decrease 1 instance at a time

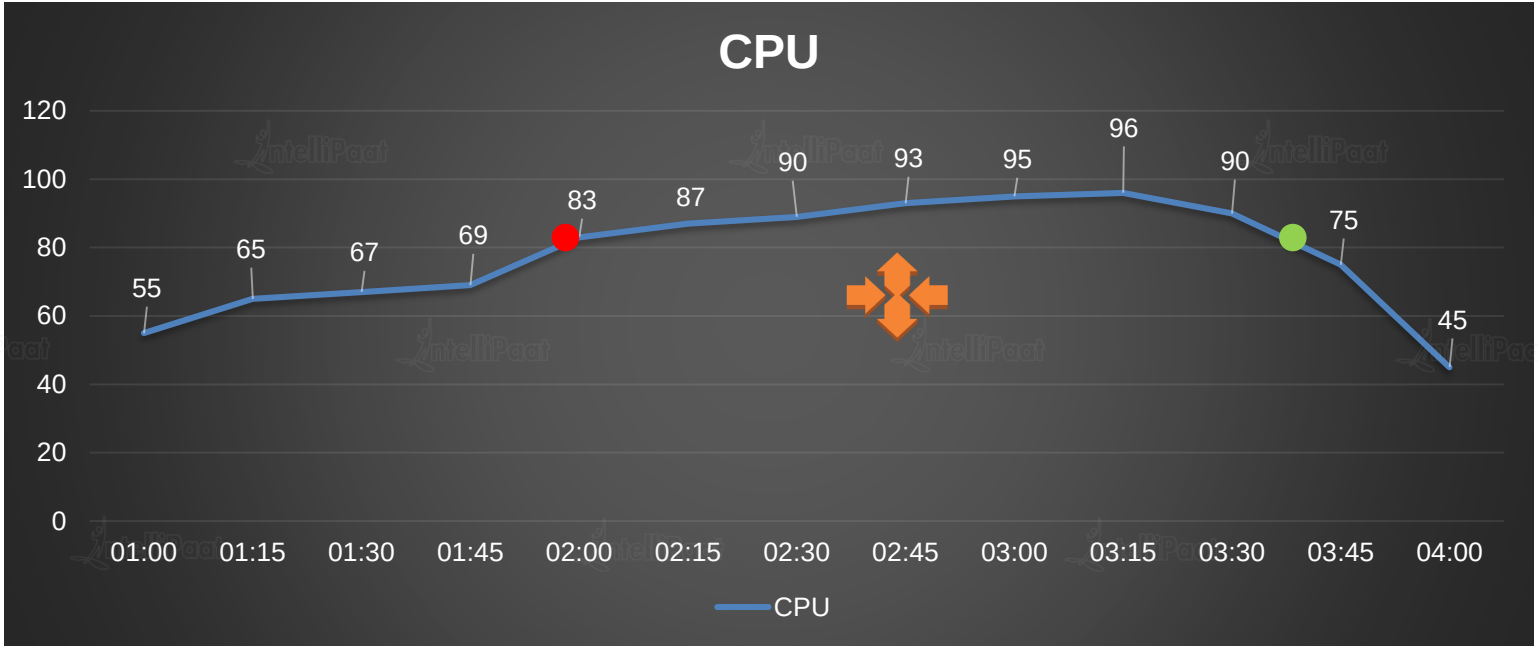
Alarm:

If CPU utilization > 80% for more than 10 mins, ring the bell

- Minimum capacity: 2
- Desired capacity: 4
- Maximum capacity: 10



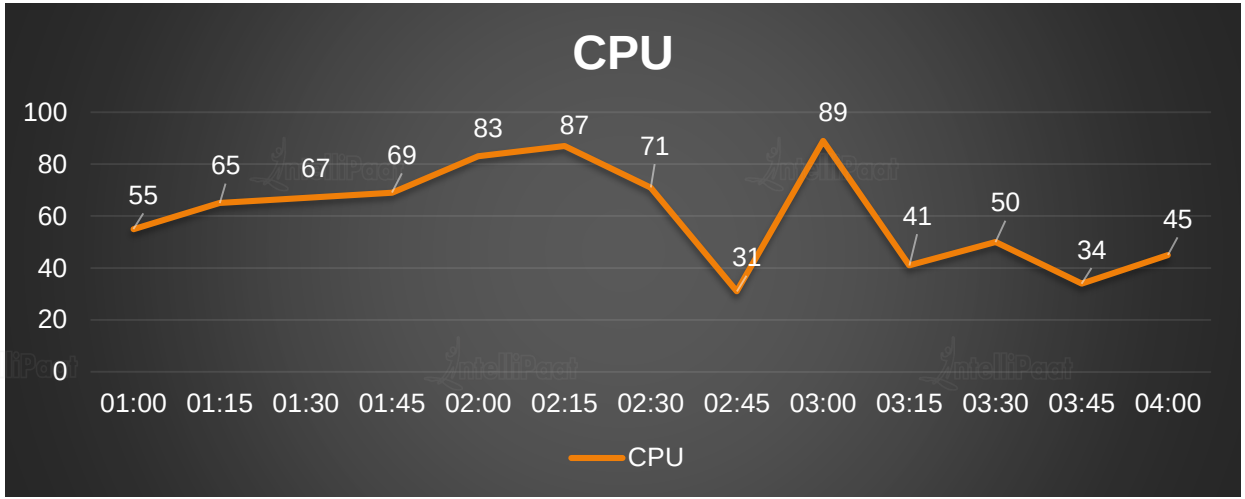
Scaling Policy



Scaling Policy

Cool-down Period: Simple Scaling Policy

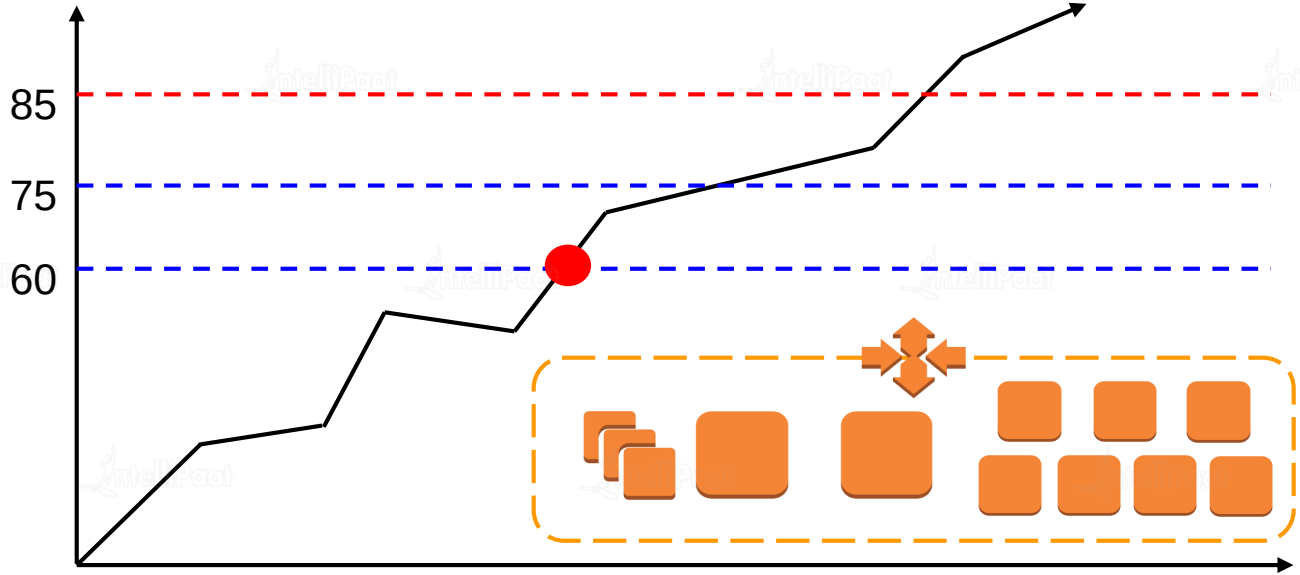
The cool-down period ensures that autoscaling does not launch or terminate any more instances until a specified period is completed. Scaling activity is suspended until the cool-down period is in effect



Scaling Policy

Step Scaling

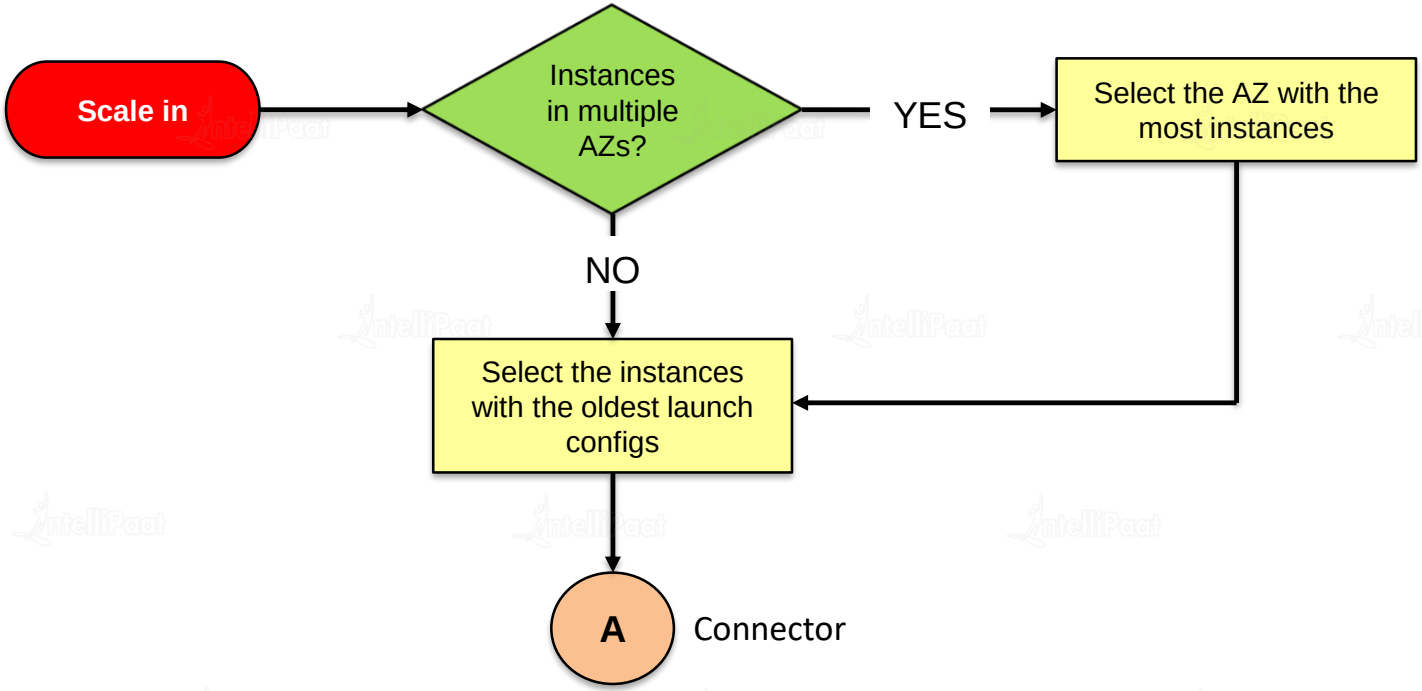
- **Alarm: CPU > 60%**
- **Action: Add 2 Instances**



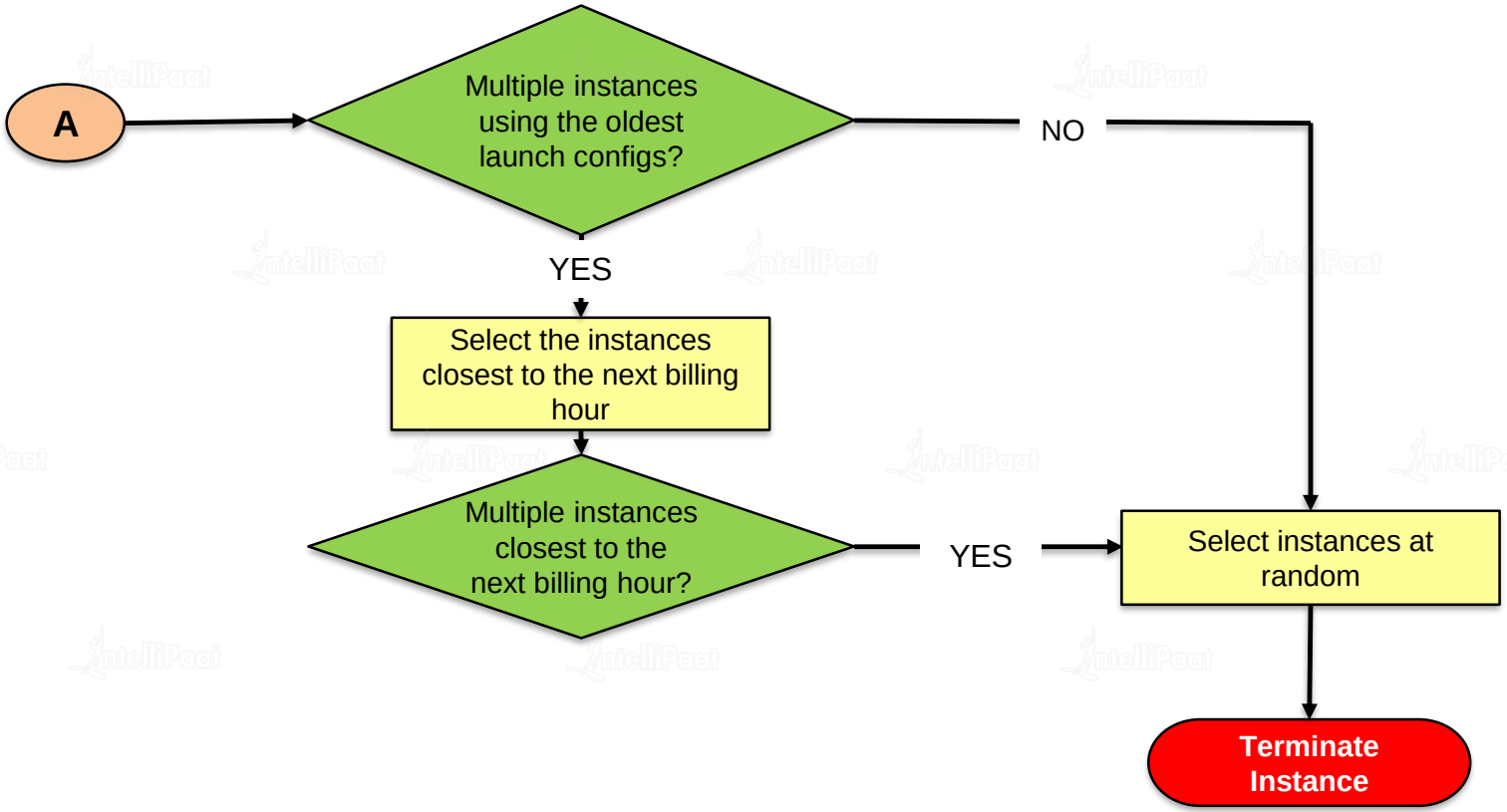
CPU	Add
> 60%	2
> 75%	3
> 85%	4

Instance Termination

Instance Termination



Instance Termination



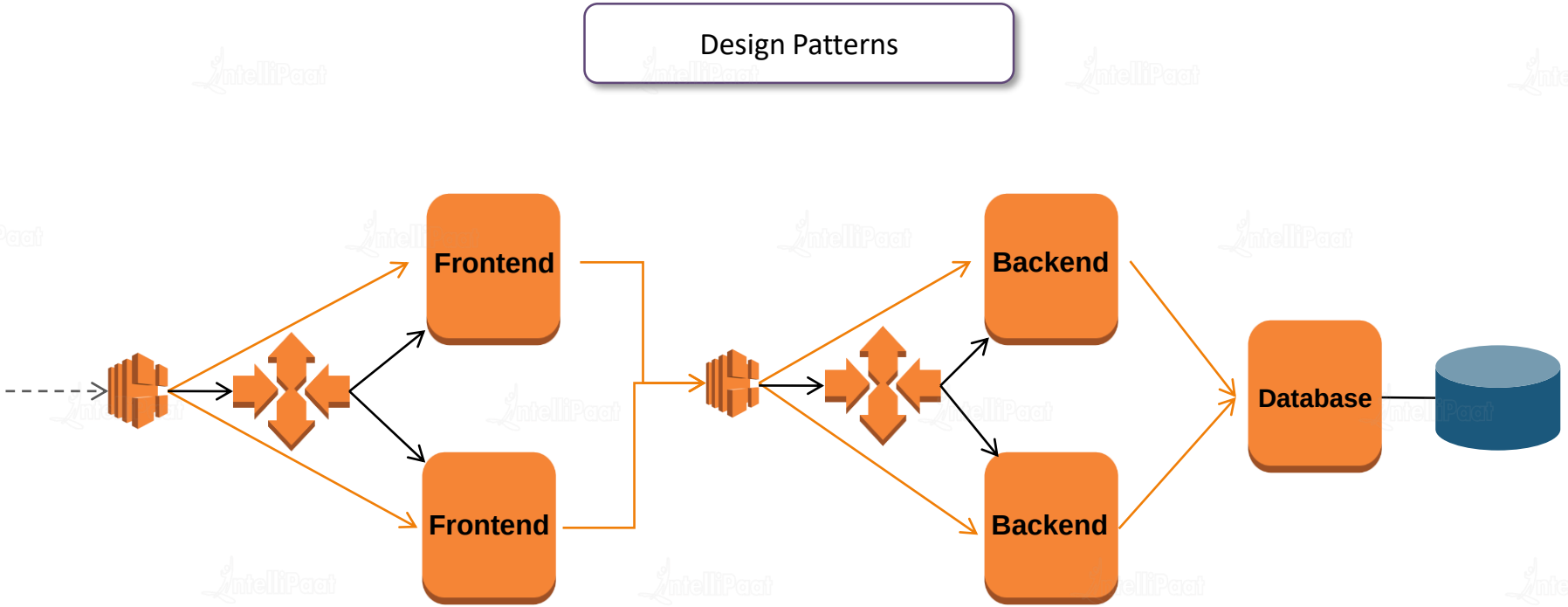
Instance Termination

- Termination Policies (Other than the default)
  - Oldest instance
  - Newest instance
  - Oldest launch configuration
  - Closest to the next instance hour
- Instance protection does not terminate an instance during a scale in event. It can be enabled at the autoscaling group or individual instance level

Autoscaling Pricing

- No additional fees
- Underlying instances are charged hourly
- For details, visit: https://aws.amazon.com/autoscaling/pricing/

Autoscaling Design Patterns

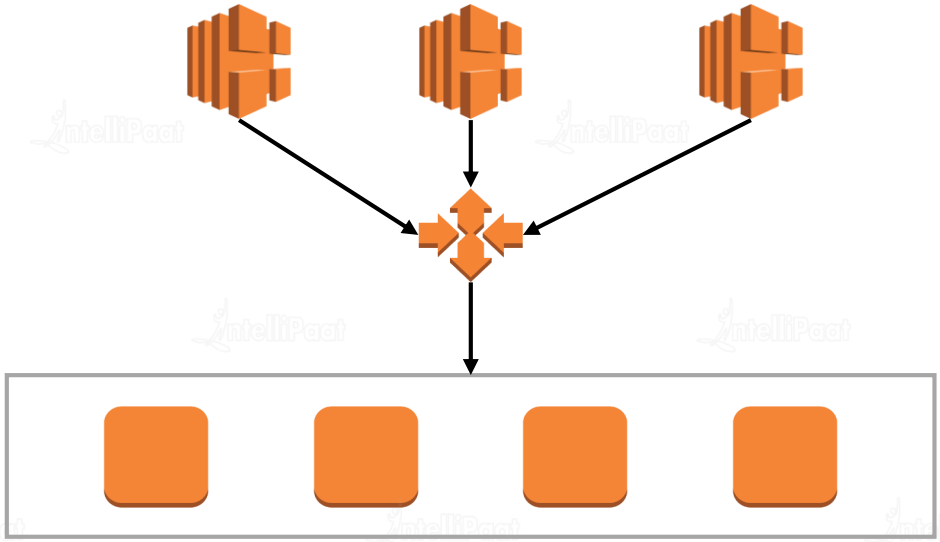


Demo: Creating Auto Scaling Groups

ELB & AS Integration

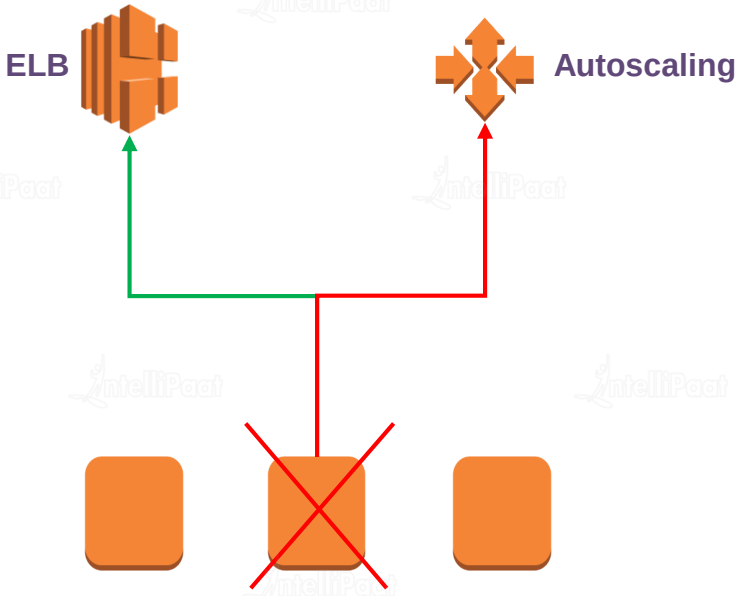
ELB and AS Integration

- **Autoscaling:** Adds and removes capacity as per requirement
- **Load Balancer:** Distributes the incoming traffic evenly across all EC2 instances
- Placing ELB in front of AS makes sure that all the incoming traffic are distributed, dynamically changing the number of EC2 instances
- ELB is the point of contact between the clients and the backend EC2 instances



ELB and AS Integration

- Load balancer automatically registers instances in the group
- Health checks:
  - EC2 instance only: EC2 status checks are considered
  - EC2 and ELB health checks: An instance is considered unhealthy if either of the health checks fail

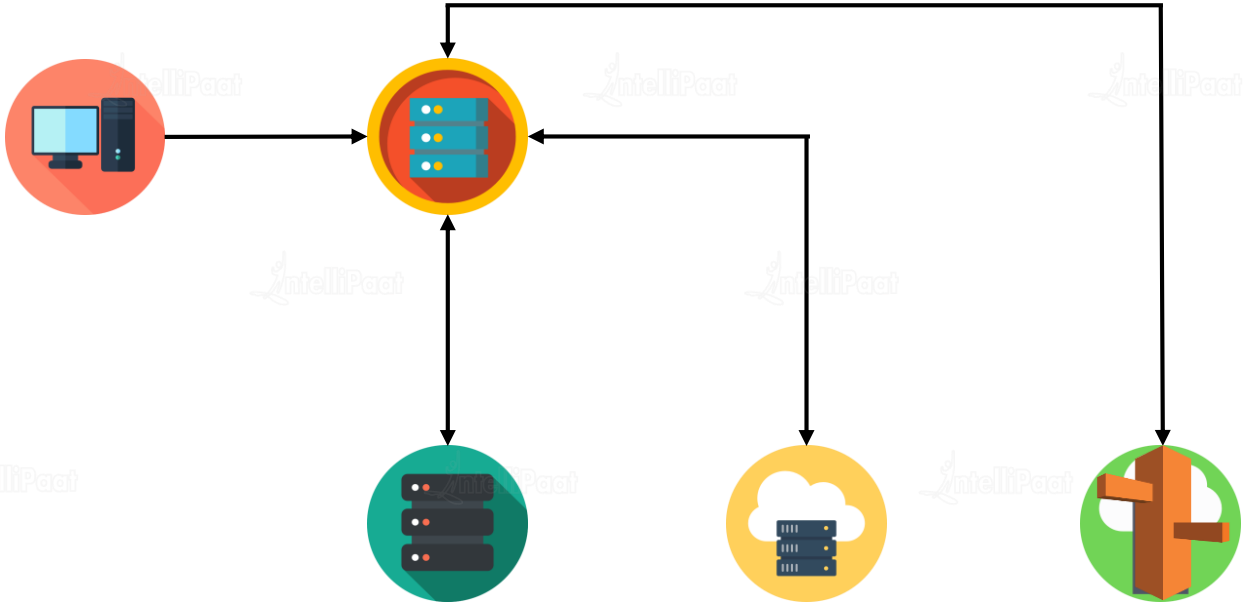


Demo: Integrating Auto Scaling group to the ALB

Pre-Route 53

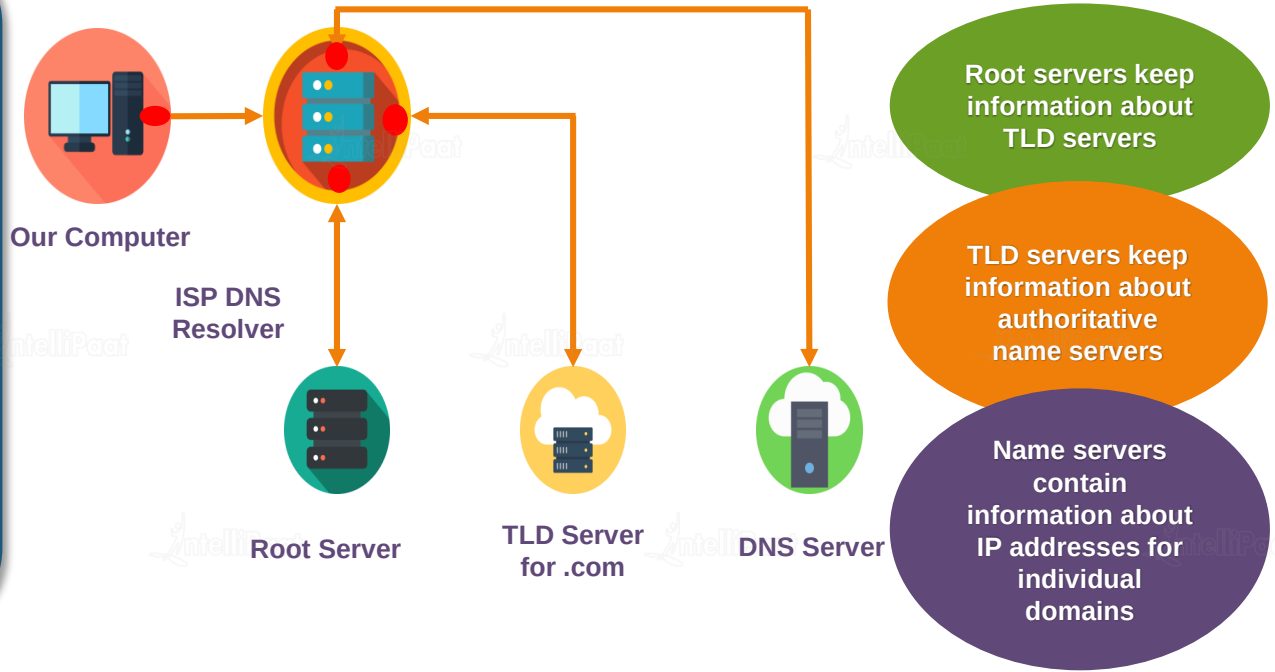
What is Route 53?

Route 53 is the highly available and scalable Domain Name System provided by AWS

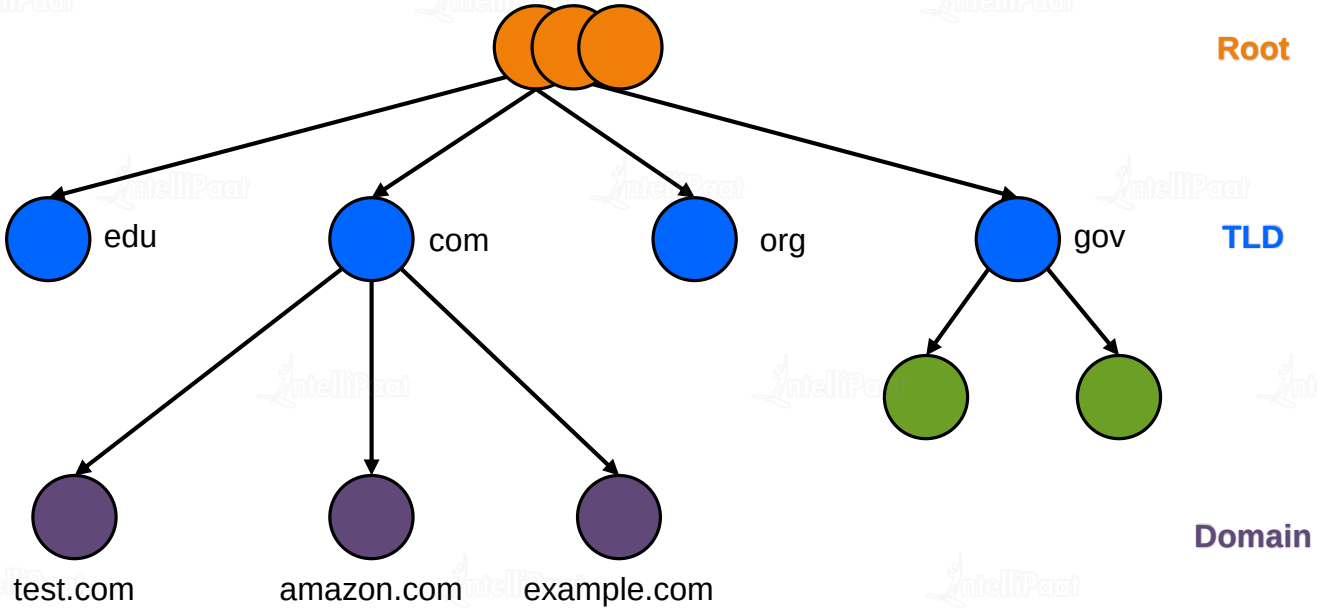


Domain Name System

- In www.amazon.com
- com: Top-level domain name
- Amazon: Domain name
- **Domain Name System** is an Internet service that translates the domain names into IP addresses



DNS Hierarchy

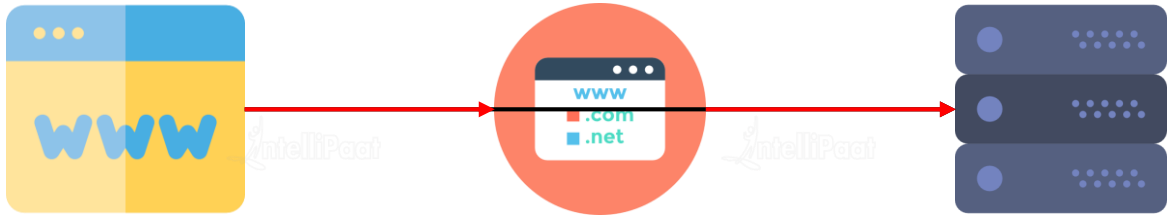


Hosting Our Website

Step 1: Start up a server/host where the web service will run (Say, the IP address of the server is 10.20.30.40)

Step 2: Get a domain name from the domain name providers such as GoDaddy, Freenom, etc.

Step 3: Link the domain name with the IP address from Step 1 by using the Domain Name Service/System



www.example.com	10.20.30.40

DNS Literature

- Authoritative Name Server: The server component in Domain Name System (DNS) that holds actual DNS records such as A Name, CNAME, Alias, etc.
- 'A' NAME Record: It maps the domain name to the IP address of the backend host. 'A' is for address. The A NAME record format is mentioned below:

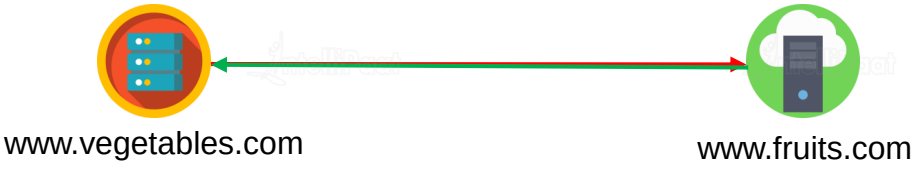
Type	Domain/Host Name	Address	TTL
A	www.abc.com	101.202.30.40	60
A	www.apple-orange.com	54.28.14.6	300
AAAA	www.example.com	fe80::1cb2:373a:3dd1:8f46	600

DNS Literature

- CNAME (Canonical Name) Record: It maps one name to another name instead of an IP address

Type	Domain/Host Name	Address	TTL
CNAME	www.fruits.com	www.apple-orange.com	300
CNAME	www.vegetables.com	www.fruits.com	600
A	www.apple-orange.com	54.28.14.6	900

- Alias Name is similar to the CNAME record with a ‘little’ difference



DNS Literature

Type	Domain/Host Name	Address	TTL
CNAME	www.fruits.com	www.apple-orange.com	300
CNAME	www.vegetables.com	www.fruits.com	600
A	www.apple-orange.com	54.28.14.6	900

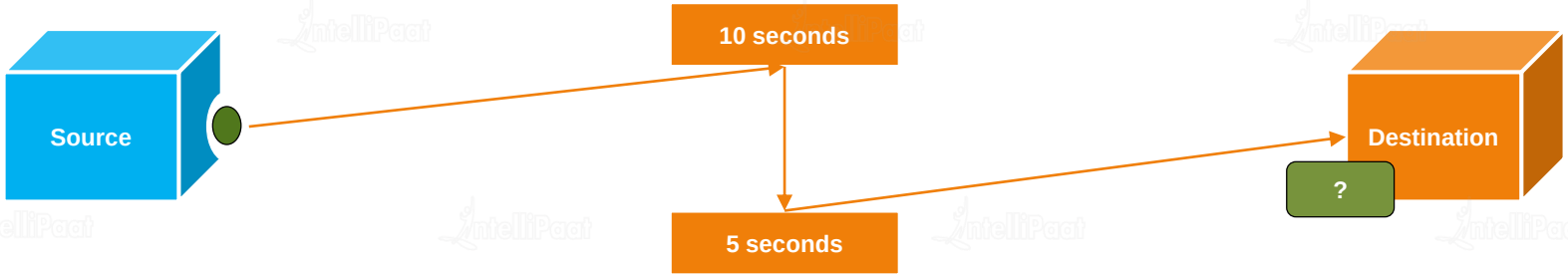
www.fruits.com

www.apple-orange.com

54.28.14.6

Network Latency and Bandwidth

Network latency is the amount of time taken to deliver some amount of data over n/w



Latency = 10 + 5 = 15 seconds

Routing Policy

Routing Policy

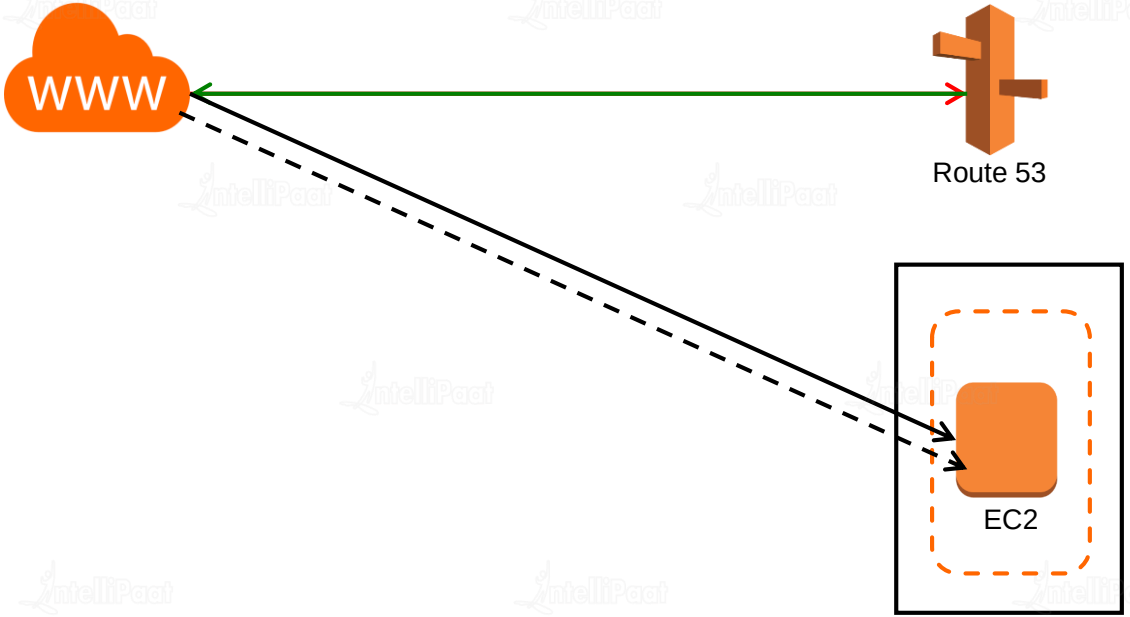
- **Public Hosted Zone** contains information about how the traffic on the Internet should be routed for a domain
- NS record set: The authoritative name servers for a domain name
- SOA (Start of Authority) record set: Contains the base DNS information about the domain

```
ns-2048.awsdns-64.net. hostmaster.example.com.  1  7200  900  1209600  86400
```

  - ns-2048.awsdns-64.net: Host that created the SOA record
  - hostmaster.example.com: The email address of the admin with '@' being replaced by '.'
  - 86400: Minimum TTL
- **Private Hosted Zone** contains information about how to route the traffic for a domain within one or more VPCs
  - Note: To use private hosted zones, following VPC settings have to be set to TRUE:
    - enableDnsHostnames
    - enableDnsSupport

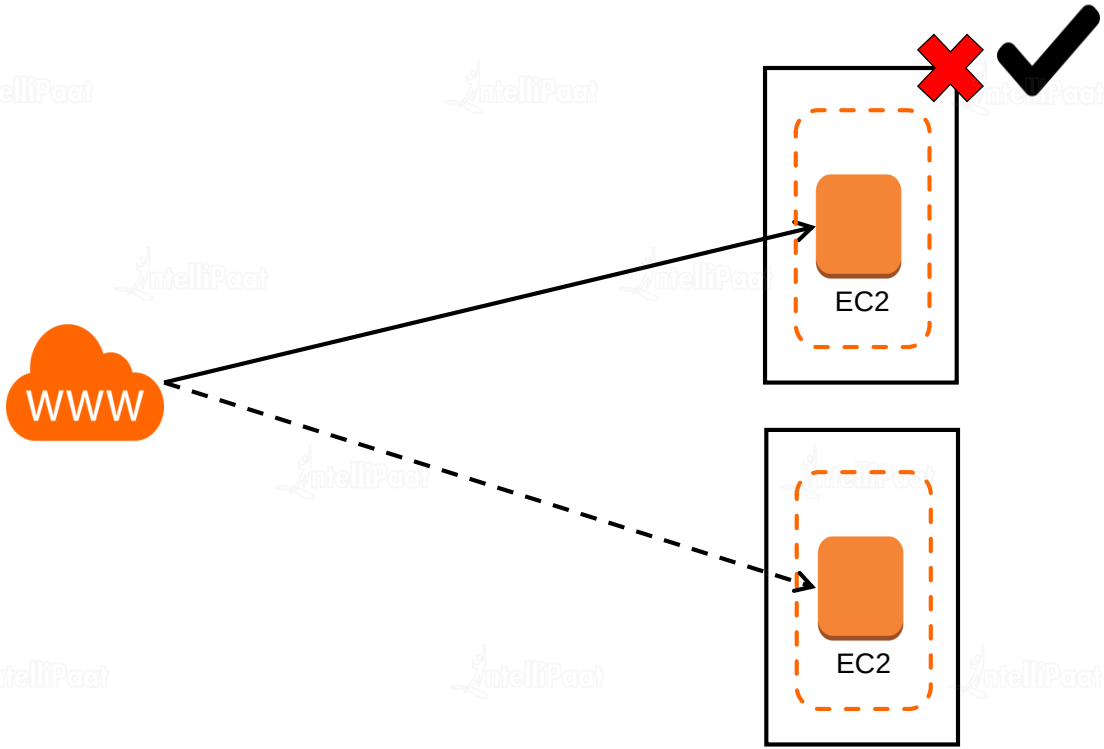
Routing Policy

Simple Routing Policy: A single server performing the desired operation



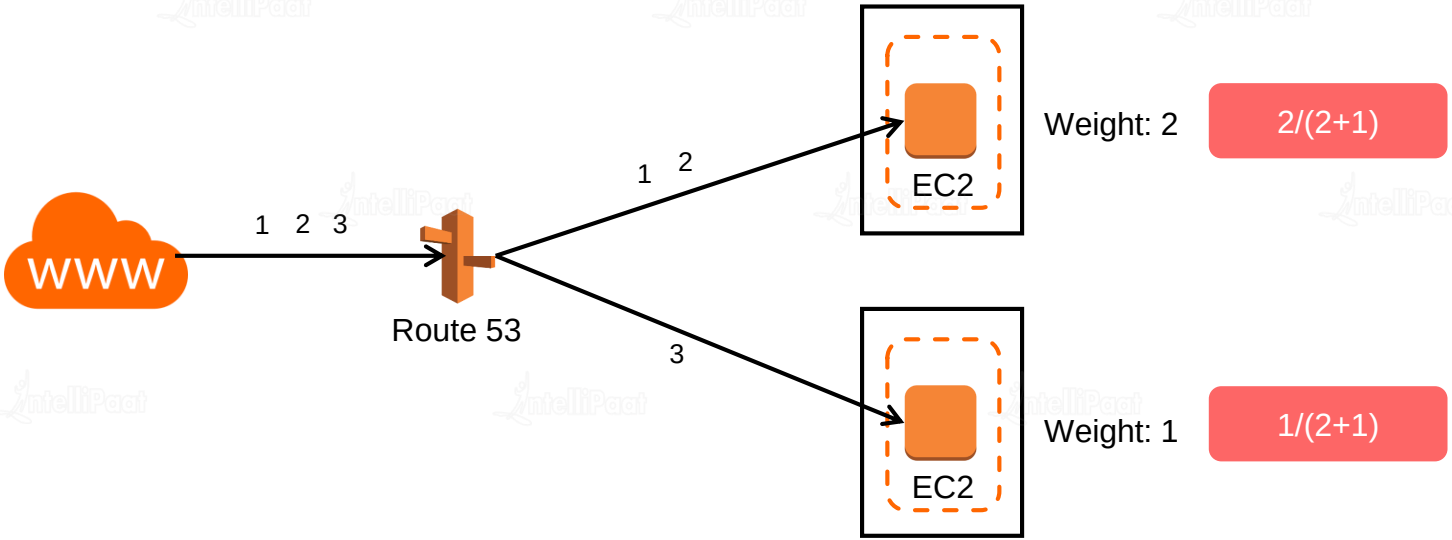
Routing Policy

Failover Routing Policy: Two servers performing the Active–Passive routing



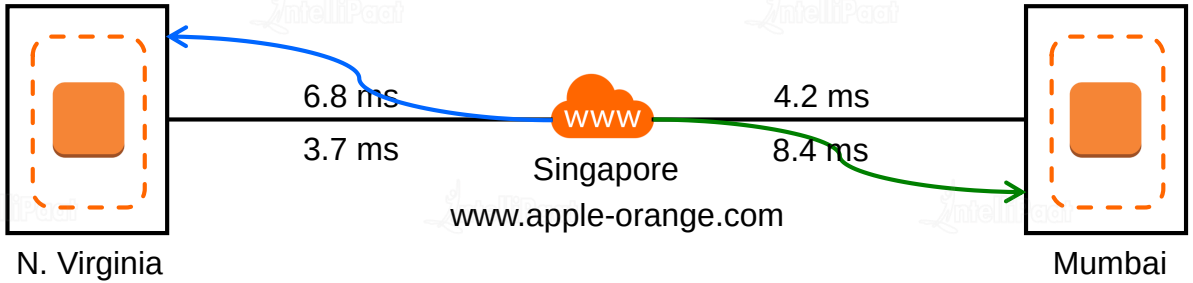
Routing Policy: Weighted

- It associates multiple resources with the same DNS name and type
- Each record set is given a weight and a set ID



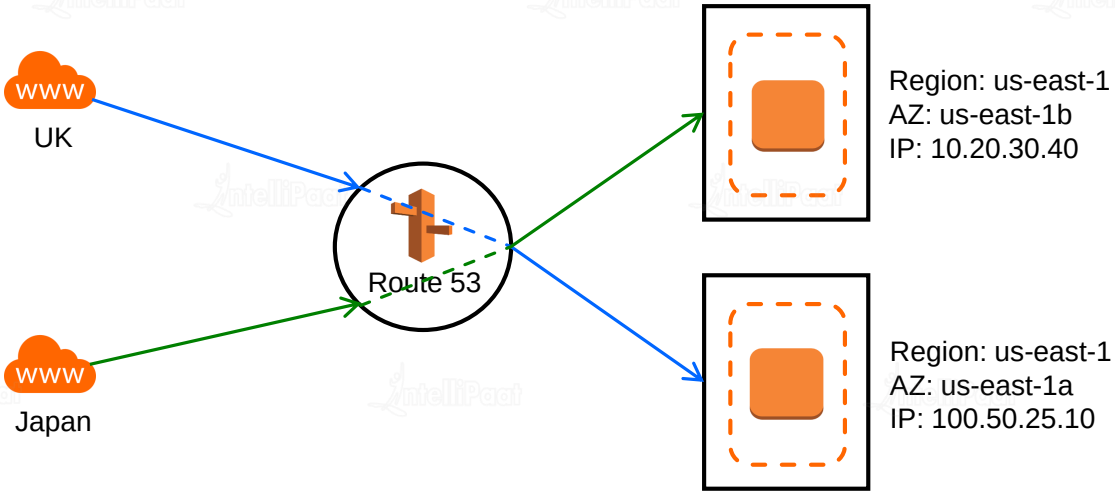
Routing Policy: Latency-based

- If an application is hosted on EC2 instances in multiple regions, user latency can be reduced by serving requests from the region where network latency is the lowest
- We have to create a latency resource record set for the Amazon EC2 resource in each region that hosts the application
- Latency record sets can be created for both ELB and EC2 instances
- Latency on the Internet can change over time due to changes in routing or something else



Routing Policy: Geolocation

- Geolocation routing can be used to send the traffic to resources based on the geographical location of users. For example, all queries from Europe can be routed to the IP address 10.20.30.40
- Geolocation works by mapping IP addresses, irrespective of regions, to locations



Demo: Use Route 53 to route traffic to ELB

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24/7 Chat with Our Course Advisor