AWS Solutions Architect Associate

Session 601

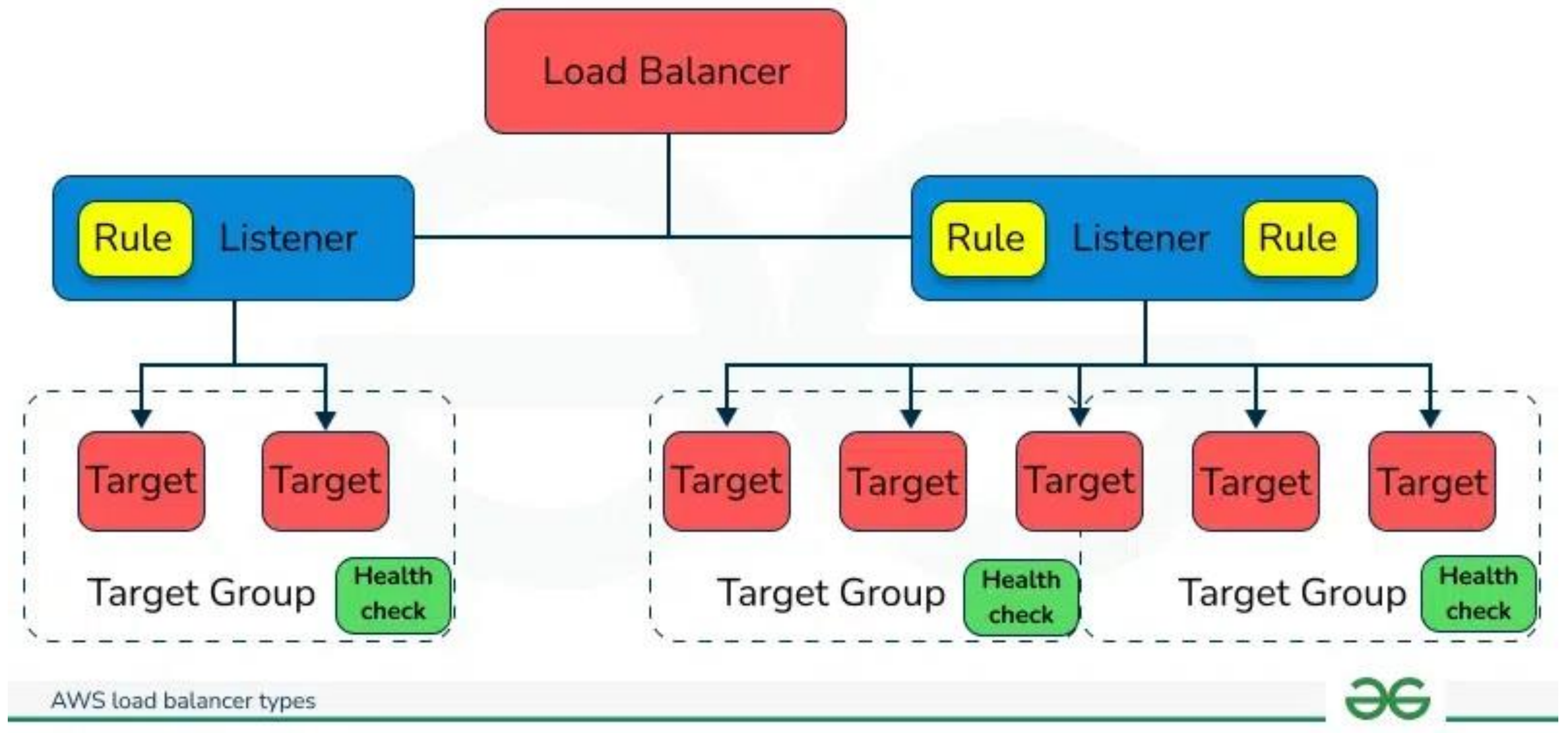
Compute: ELB and Mgmt &Gov: Cloudwatch

July/2024

Elastic Load Balancer - ELB

Elastic Load Balancing

A managed load balancing service that distributes incoming application traffic across multiple Amazon EC2 instances, containers, IP addresses and Lambda Functions.



Taken from https://www.geeksforgeeks.org/what-are-aws-load-balancer-types/ (18/07/2024)

ELB

Elastic Load Balancing

- Uses HTTP, HTTPS, TCP and SSL (secure TCP) protocols.
- Can be external or internal facing
- Each load balancer is given a DNS name
- Recognizes and responds to healthy instances

ELB Benefits

High availability

Health checks

Protocol, port, path

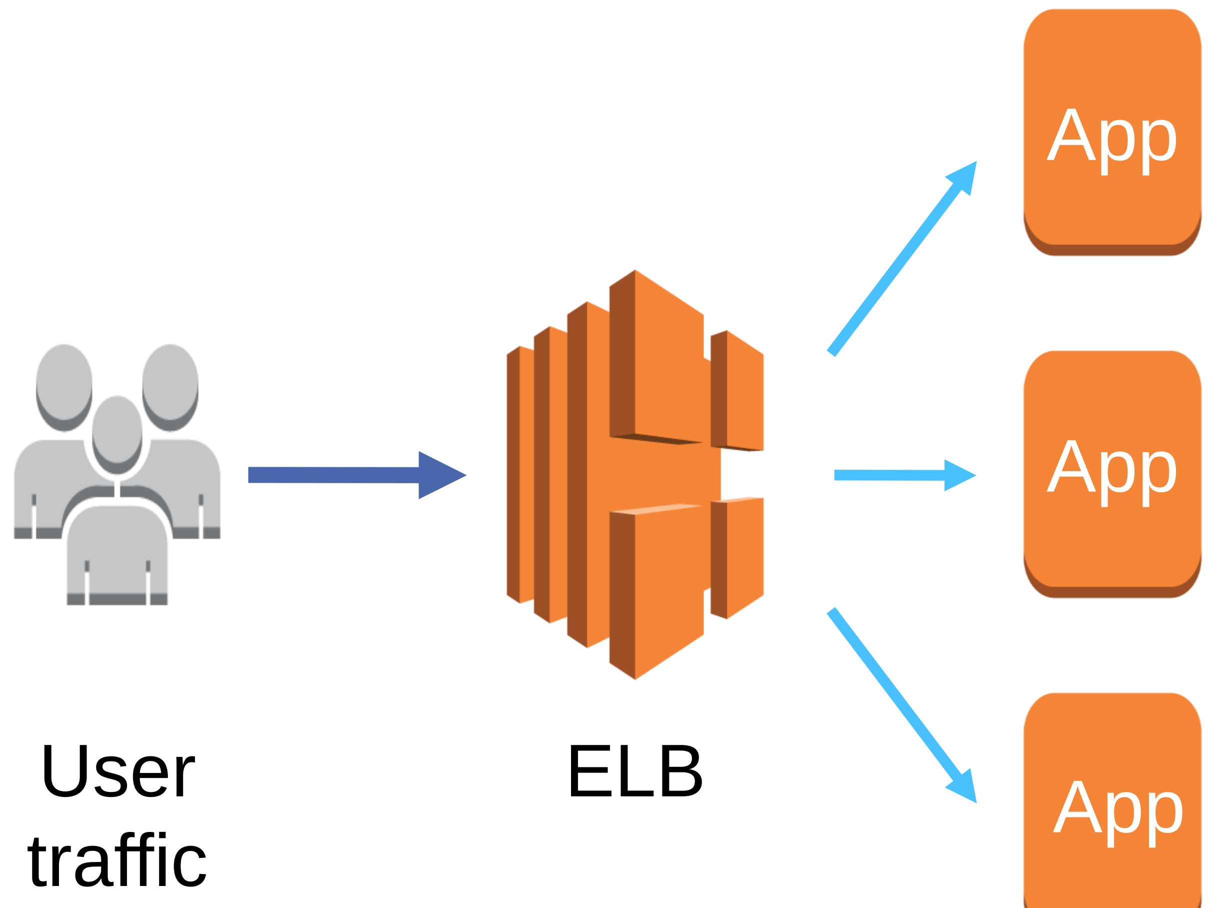
Security features

TLS termination

ELB

If you need to remove an instance from your production fleet, but don't want to affect your users:

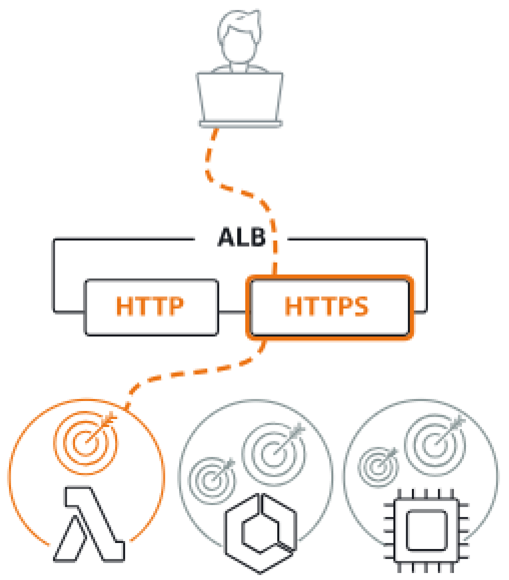
Affected backend instances will complete requests in progress before deregistration



Enable connection draining

ELB Types

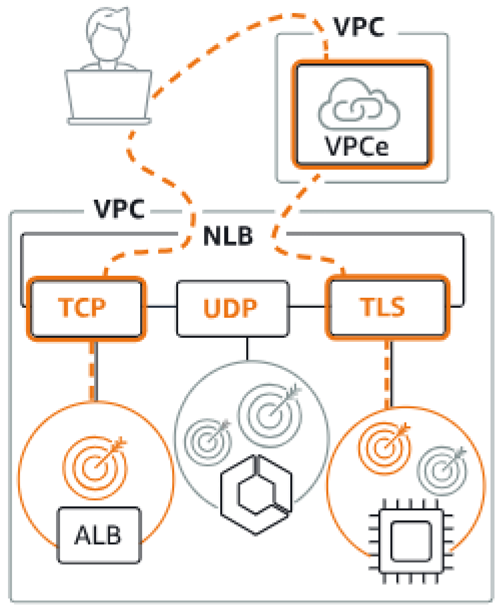
Application Load Balancer Info



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.

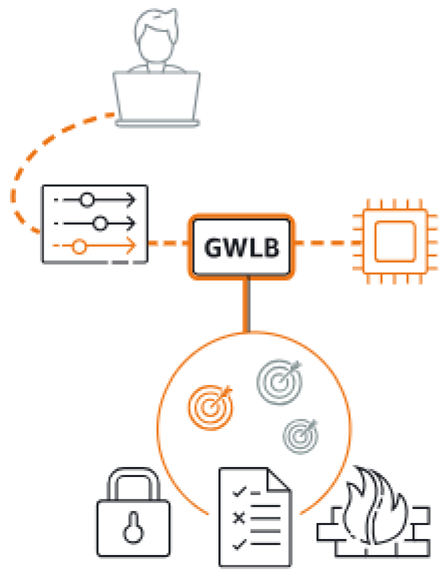
Create

Network Load Balancer Info



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.

Gateway Load Balancer Info

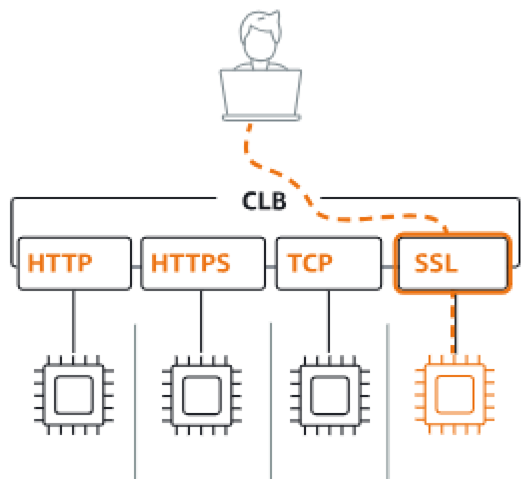


Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

▼ Classic Load Balancer - *previous generation*

Classic Load Balancer Info

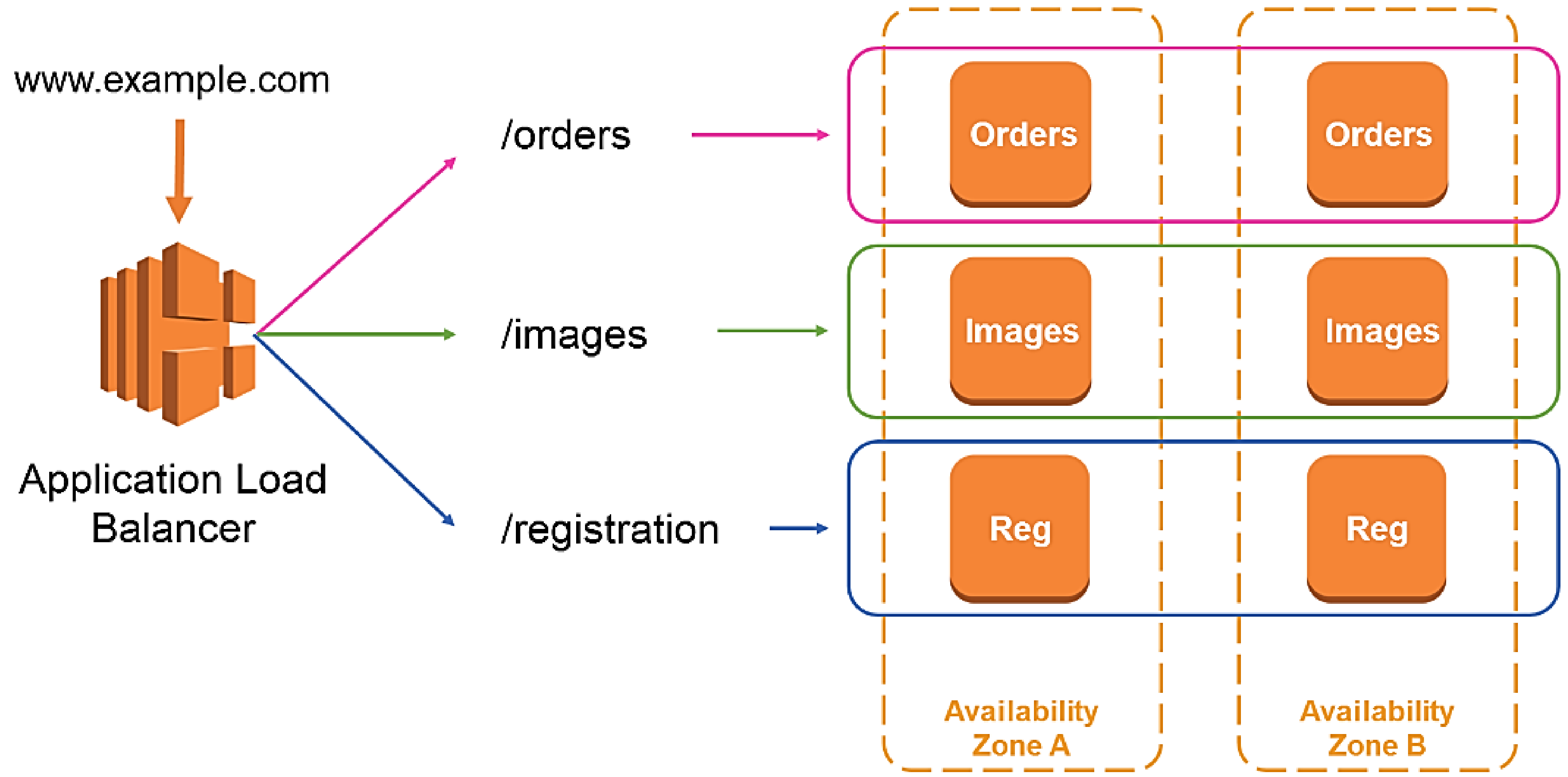


Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Create

Taken from AWS Console (18/07/2024)

Example of ALB Feature: Path Routing

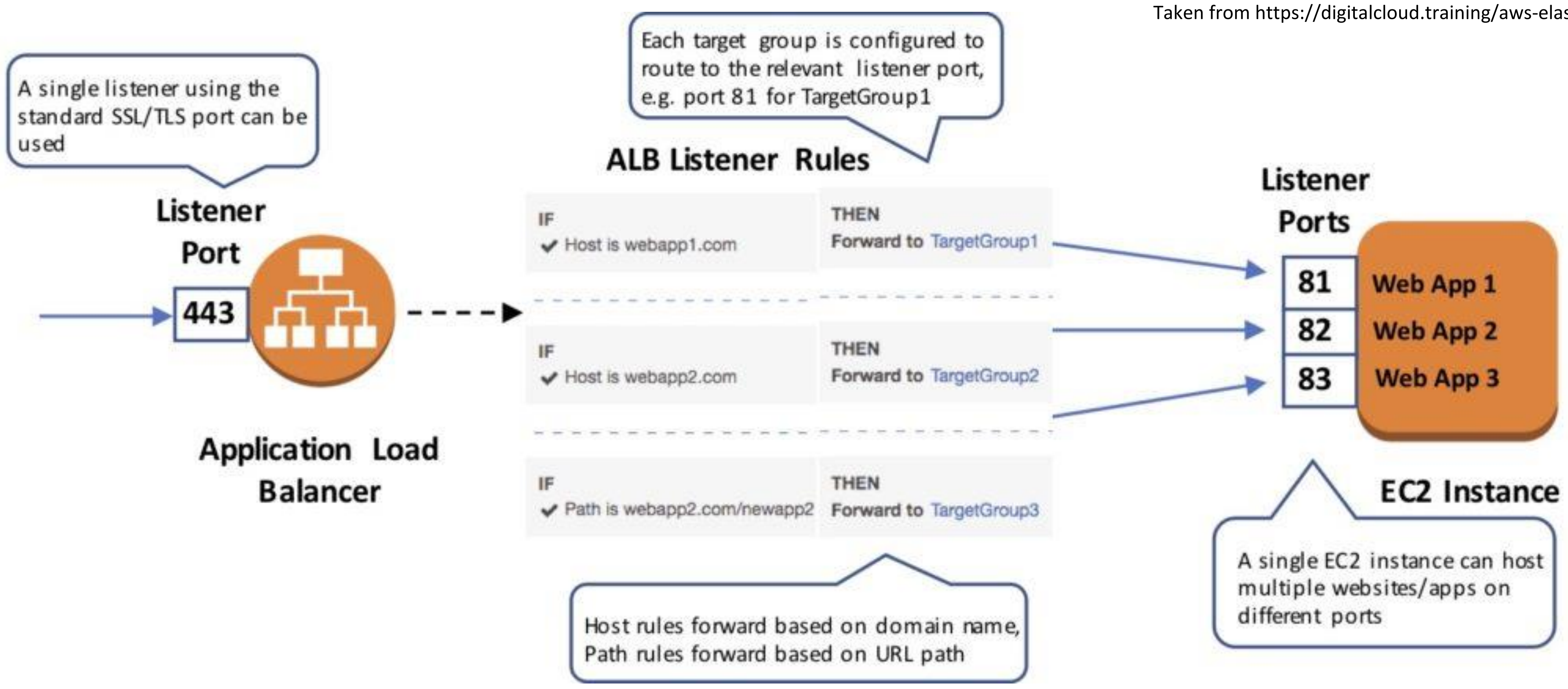


I: Target Groups: Type (Instance/IP), Port, Target Registration

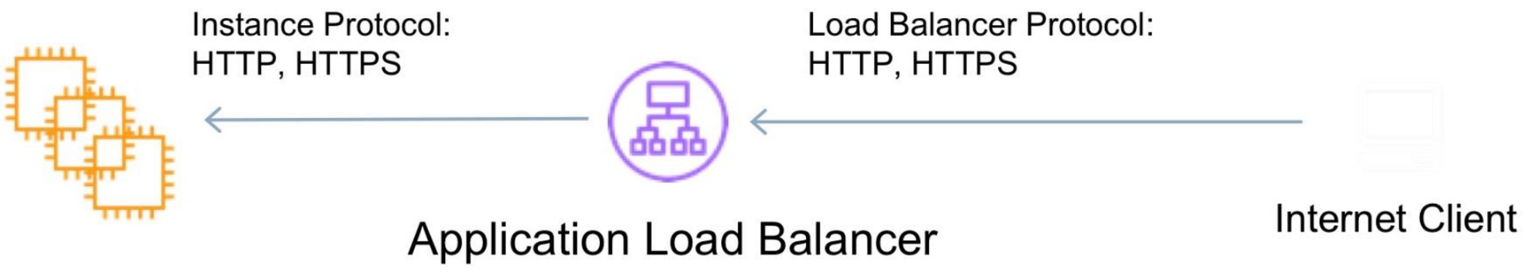
II: Load Balancer: Type and Listener.

Note: Can be done using Infrastructure or Kubernetes Commands.

Taken from https://digitalcloud.training/aws-elastic-load-balancing-aws-elb/ (18/07/2024)

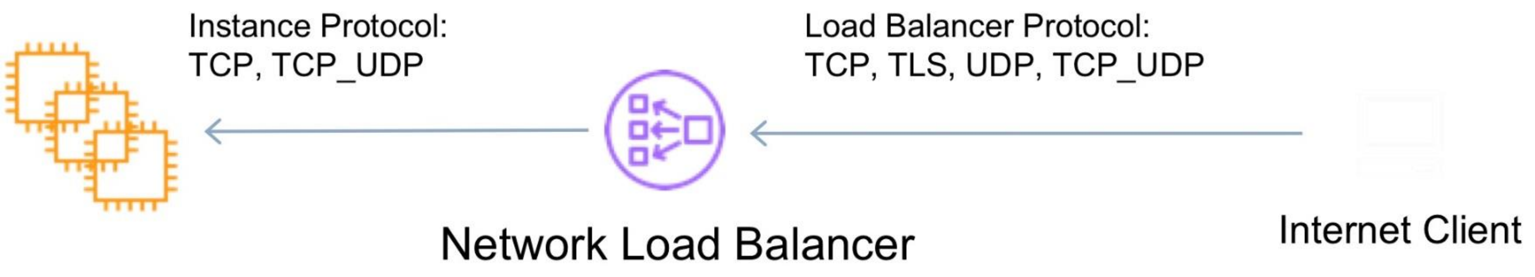

A single listener using the standard SSL/TLS port can be used
Listener Port
443
Application Load Balancer
Each target group is configured to route to the relevant listener port, e.g. port 81 for TargetGroup1
ALB Listener Rules
IF ✔ Host is webapp1.com
THEN Forward to TargetGroup1
IF ✔ Host is webapp2.com
THEN Forward to TargetGroup2
IF ✔ Path is webapp2.com/newapp2
THEN Forward to TargetGroup3
Host rules forward based on domain name, Path rules forward based on URL path
Listener Ports
81 Web App 1
82 Web App 2
83 Web App 3
EC2 Instance
A single EC2 instance can host multiple websites/apps on different ports

ELB Comparison



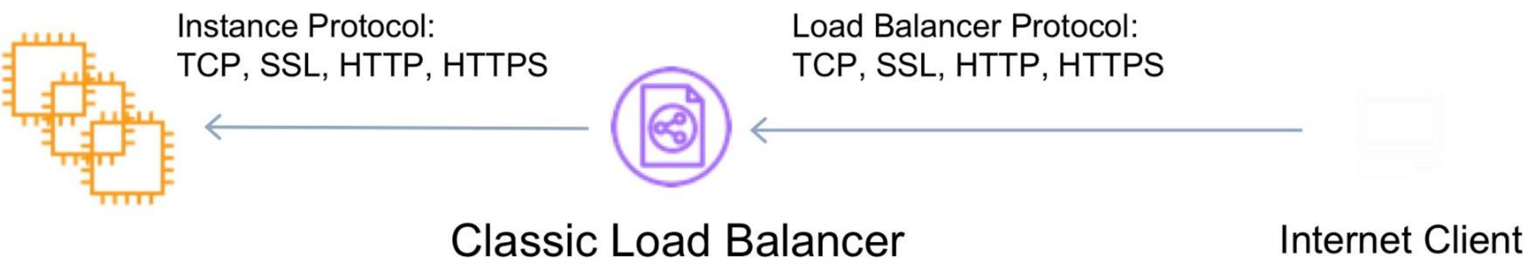
Application Load Balancer

- Operates at the request level
- Routes based on the content of the request (layer 7)
- Supports path-based routing, host-based routing, query string parameter-based routing, and source IP address-based routing
- Supports IP addresses, Lambda Functions and containers as targets



Network Load Balancer

- Operates at the connection level
- Routes connections based on IP protocol data (layer 4)
- Offers ultra high performance, low latency and TLS offloading at scale
- Can have static IP / Elastic IP
- Supports UDP and static IP addresses as targets



Classic Load Balancer

- Old generation; not recommended for new applications
- Performs routing at Layer 4 and Layer 7
- Use for existing applications running in EC2-Classic

Taken from https://digitalcloud.training/certification-training/aws-solutions-architect-associate/compute/elastic-load-balancing/ (06/07/2020)

High Availability - HA

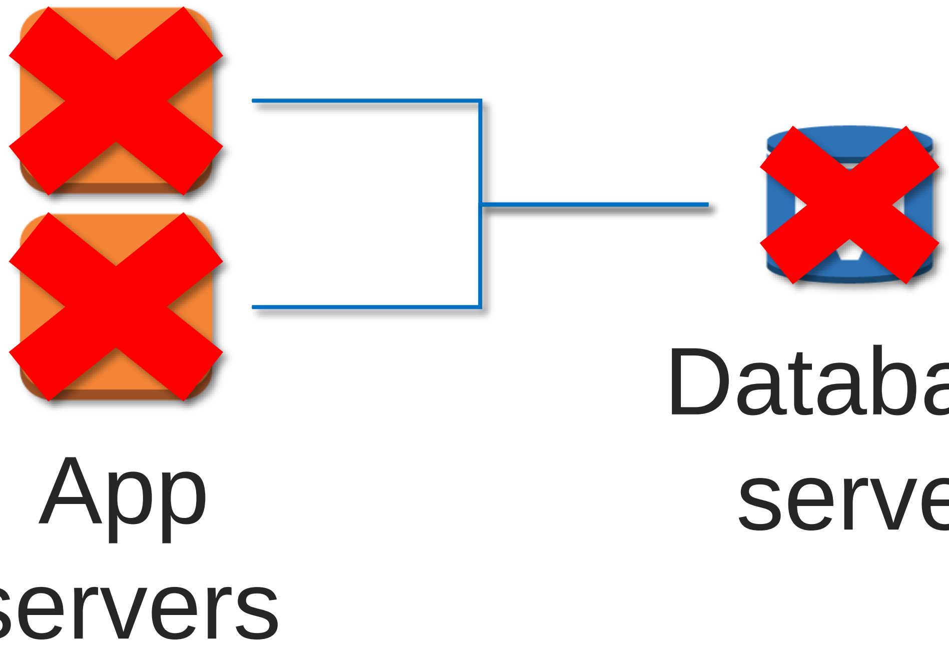
Your application can recover from a failure or roll over to a secondary source within an acceptable amount of degraded performance time.

Percent of Uptime	Max Downtime per Year	Equivalent Downtime per Day
90%	36.5 days	2.4 hrs
99%	3.65 days	14 min
99.9%	8.76 hrs	86 sec
99.99%	52.6 min	8.6 sec
99.999%	5.25 min	.86 sec

ELB HA

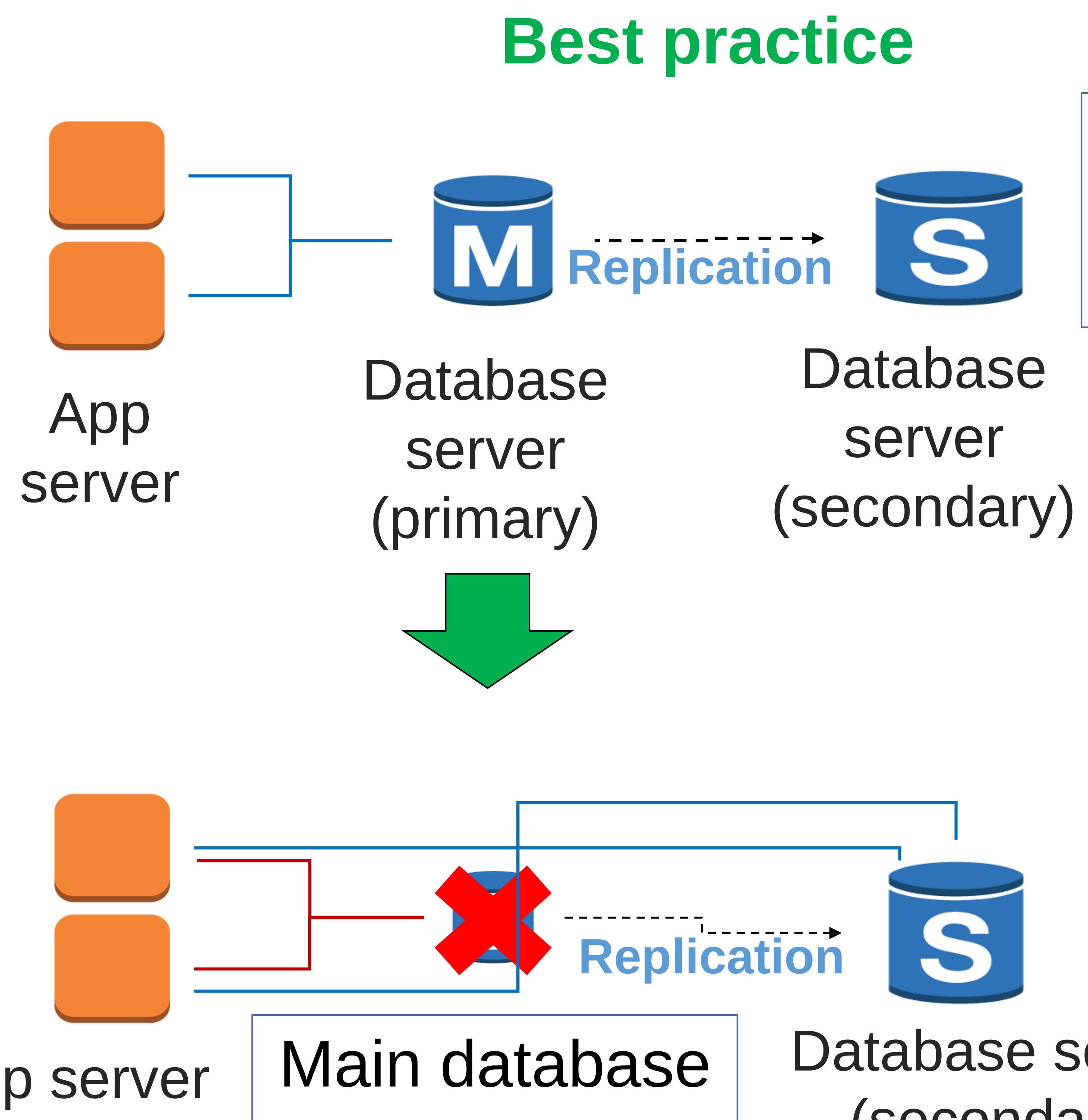
Assume everything fails, and design backward

Implement redundancy where possible in order to prevent single failures from bringing down an entire system.



Anti-pattern

ELB HA - Best Practice


Best practice
M
Replication
S
App server
Database server (primary)
Database server (secondary)
Replication
S
App server
Database server (secondary)

Create a secondary (standby) database server and replicate the data

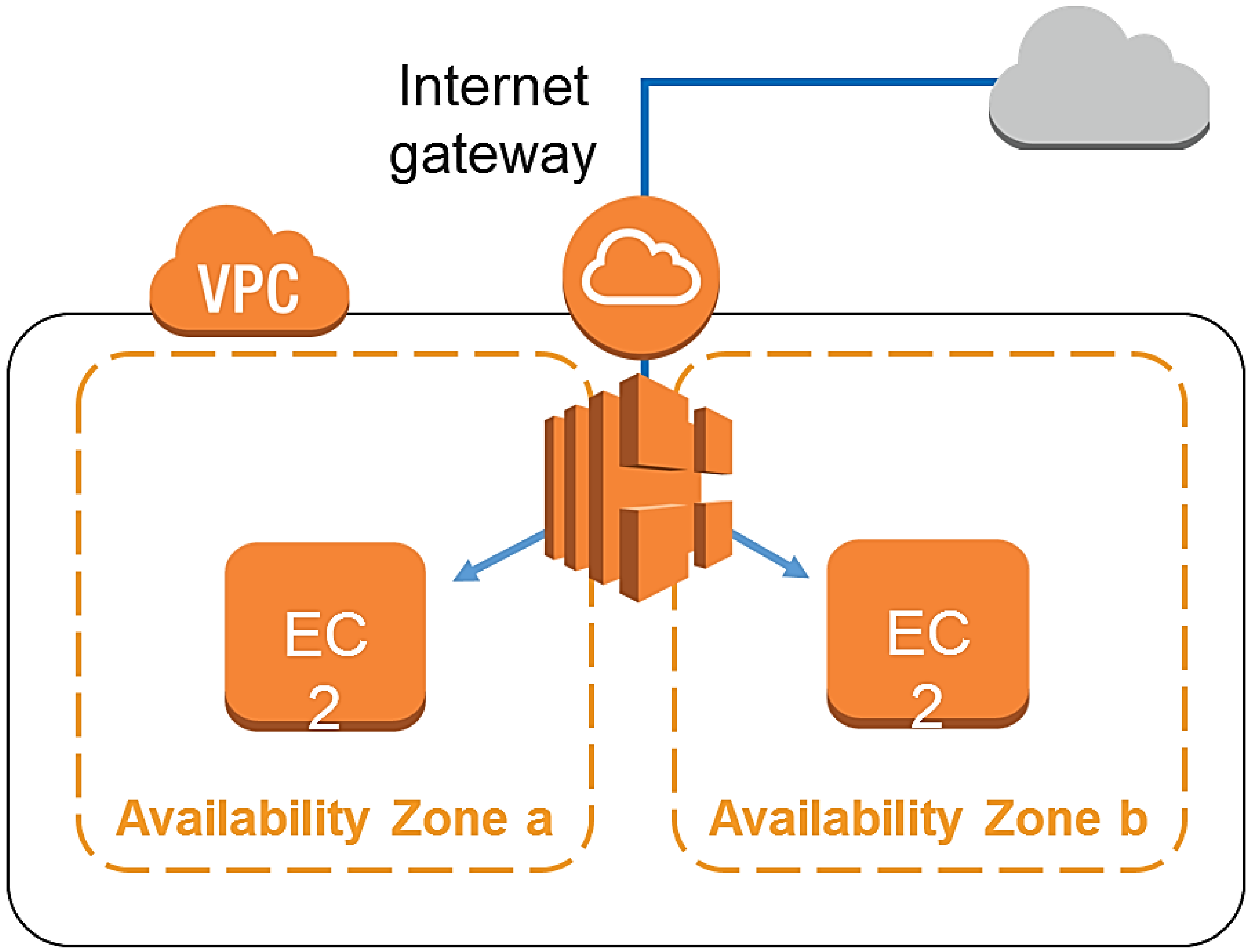
Secondary server picks up the load

Main database server goes offline

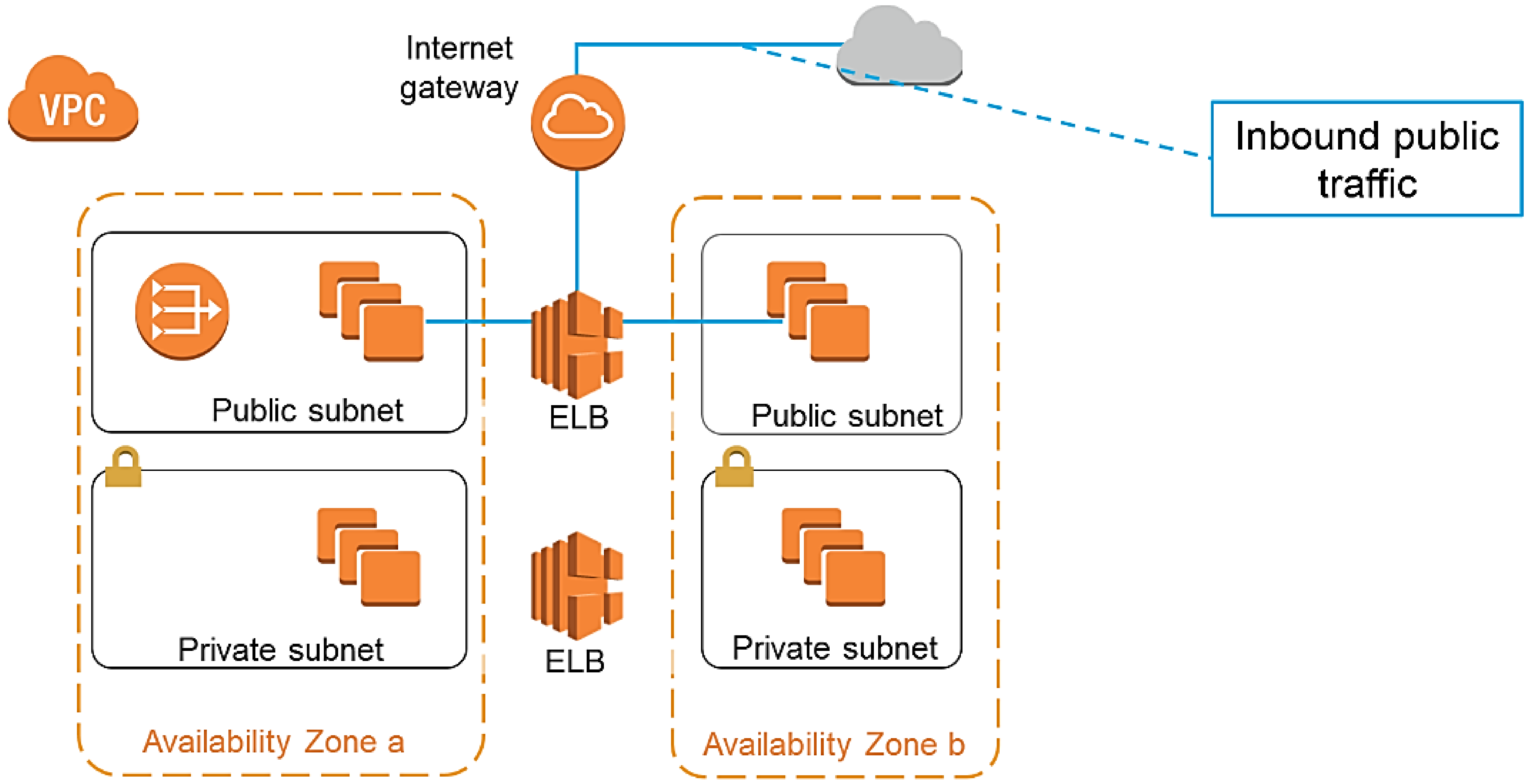
ELB HA

Start with two Availability Zones per AWS Region.

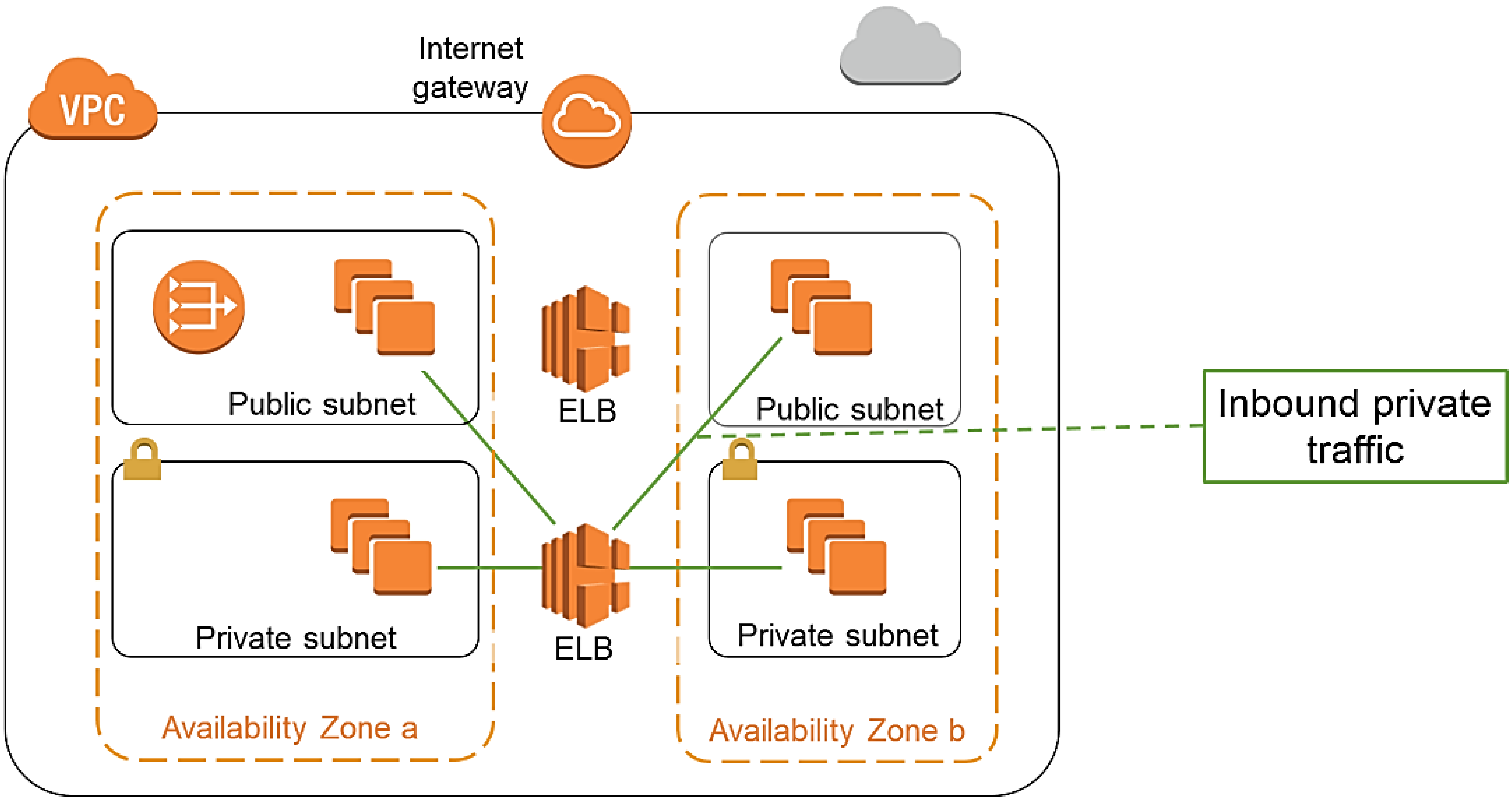
If resources in one Availability Zone are unreachable, your application shouldn't fail.



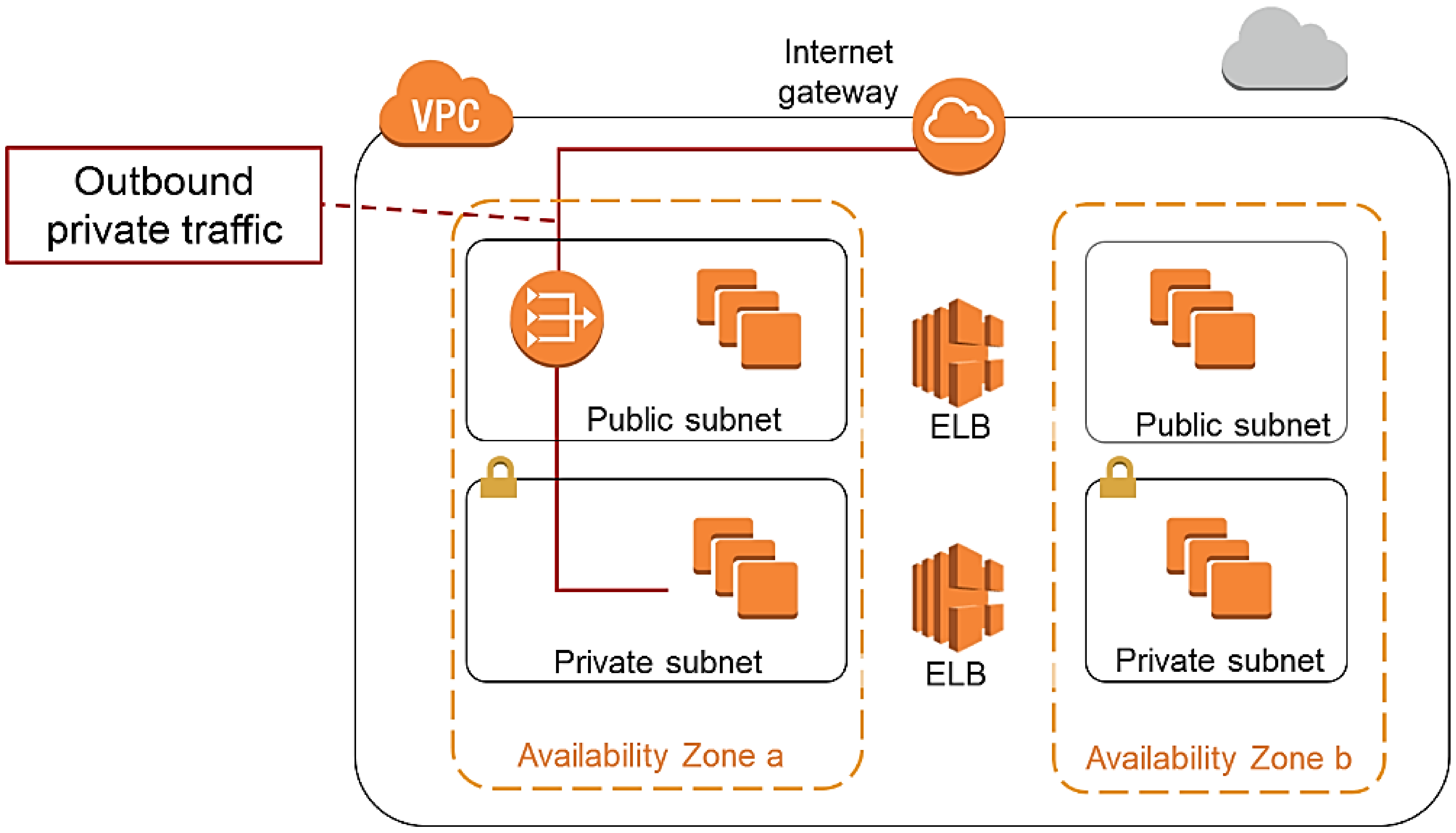
ELB HA


VPC
Internet gateway
Inbound public traffic
Public subnet
ELB
Public subnet
Private subnet
ELB
Private subnet
Availability Zone a
Availability Zone b

ELB HA


VPC
Internet gateway
ELB
Public subnet
Private subnet
Public subnet
Private subnet
ELB
Availability Zone a
Availability Zone b
Inbound private traffic

ELB HA

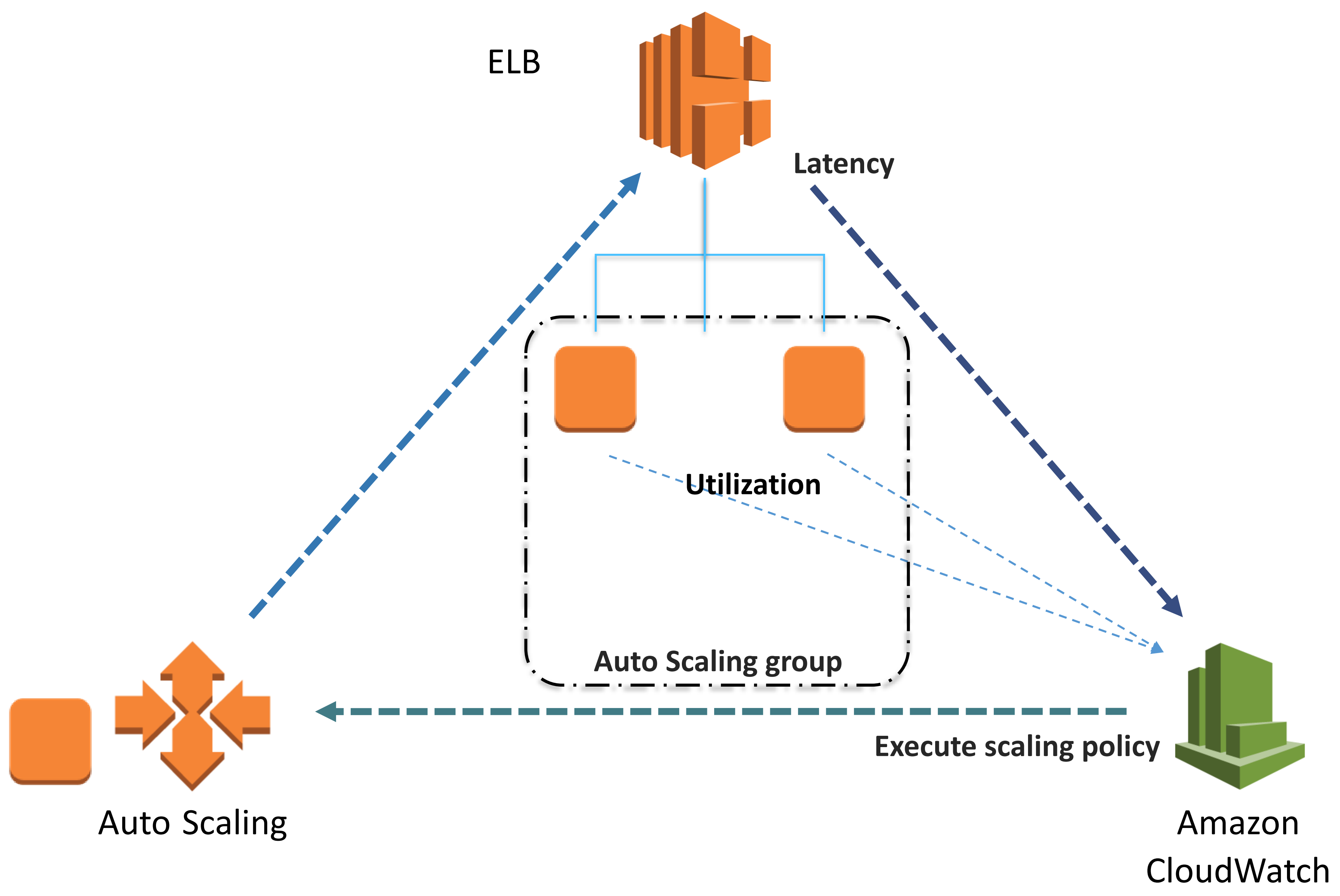

Outbound private traffic
Internet gateway
VPC
Public subnet
Private subnet
ELB
ELB
Public subnet
Private subnet
Availability Zone a
Availability Zone b

EC2 AutoScaling

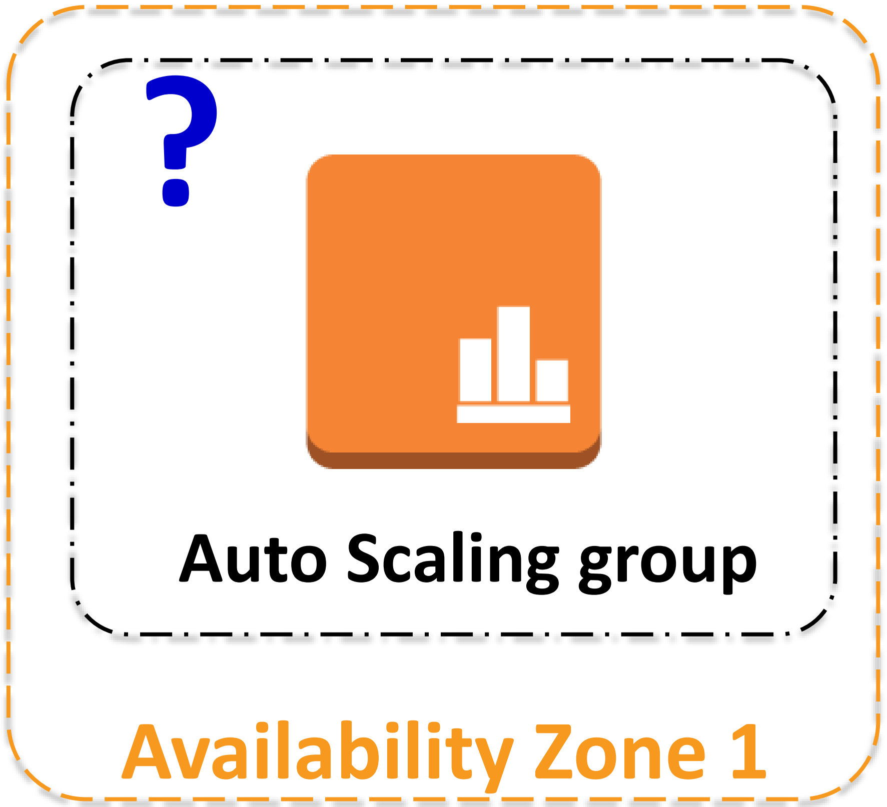
Auto Scaling

- Launches or terminates instances based on specified conditions
- Automatically registers new instances with load balancers when specified
- Can launch across Availability Zones
- Additional AutoScaling instead of EC2: Application AS (ECS, EMR, DynamoDB), AWS AutoScaling (Managed Both).

EC2 AutoScaling



EC2 AutoScaling

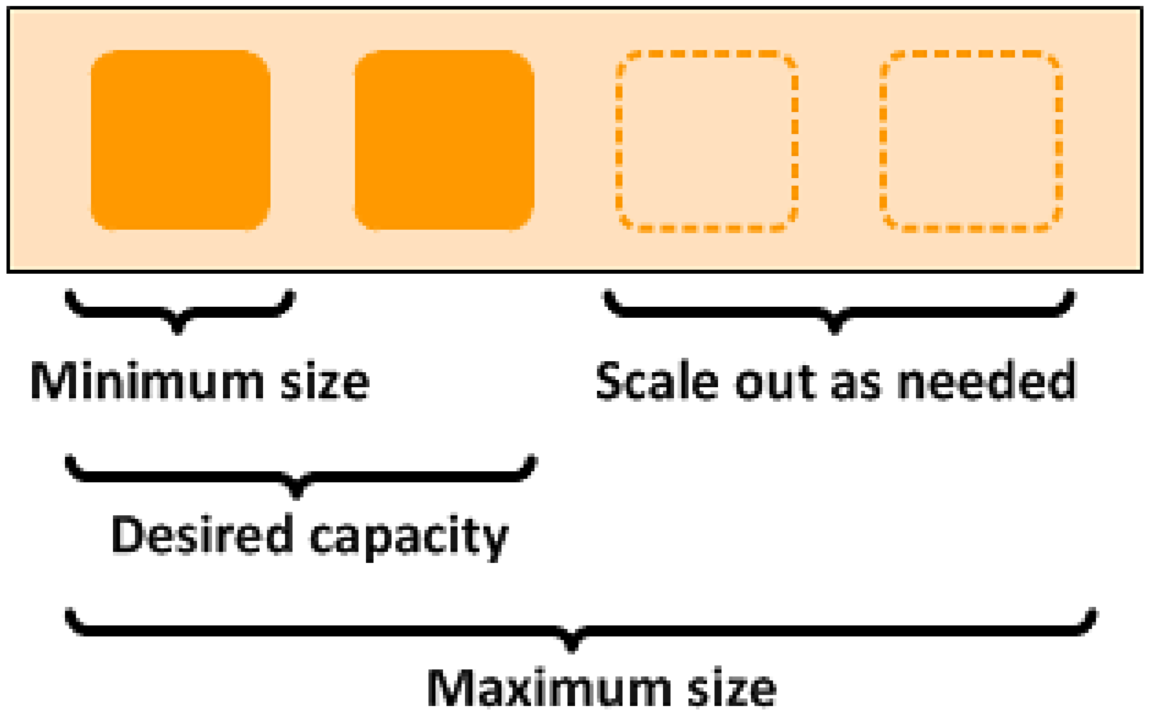


- Desired capacity
- Minimum capacity
- Maximum capacity

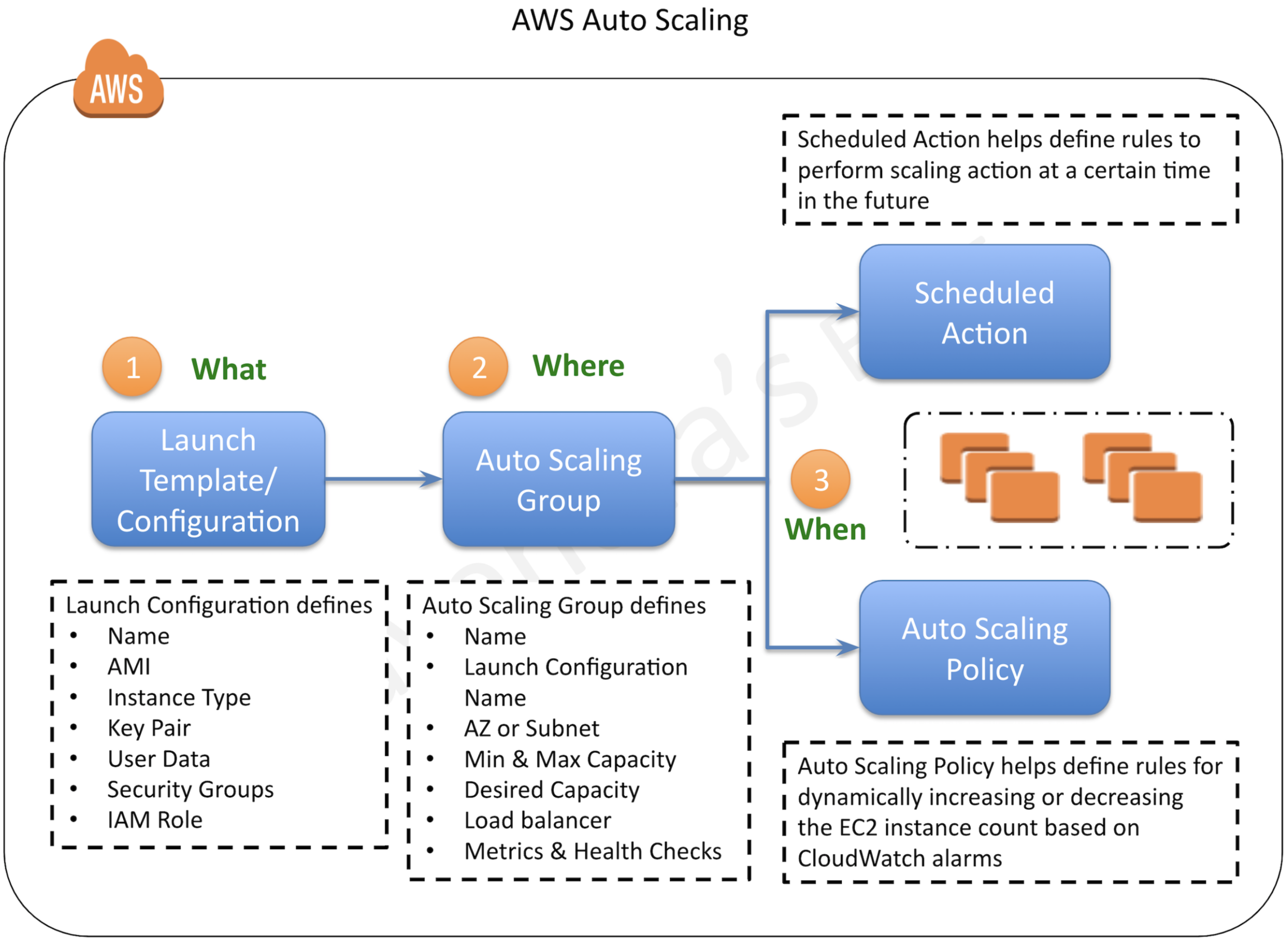
What would be a good **minimum** capacity to set it to?

What would be a good **maximum** number?

By default, **Desired** (# instances all the time) is equal to **minimum**.



ASG Components


AWS Auto Scaling
AWS
Scheduled Action helps define rules to perform scaling action at a certain time in the future
Scheduled Action
1 What
2 Where
Launch Template/ Configuration
Auto Scaling Group
3 When
Auto Scaling Policy

Launch Configuration defines
- Name
- AMI
- Instance Type
- Key Pair
- User Data
- Security Groups
- IAM Role

Auto Scaling Group defines
- Name
- Launch Configuration Name
- AZ or Subnet
- Min & Max Capacity
- Desired Capacity
- Load balancer
- Metrics & Health Checks

Auto Scaling Policy helps define rules for dynamically increasing or decreasing the EC2 instance count based on CloudWatch alarms

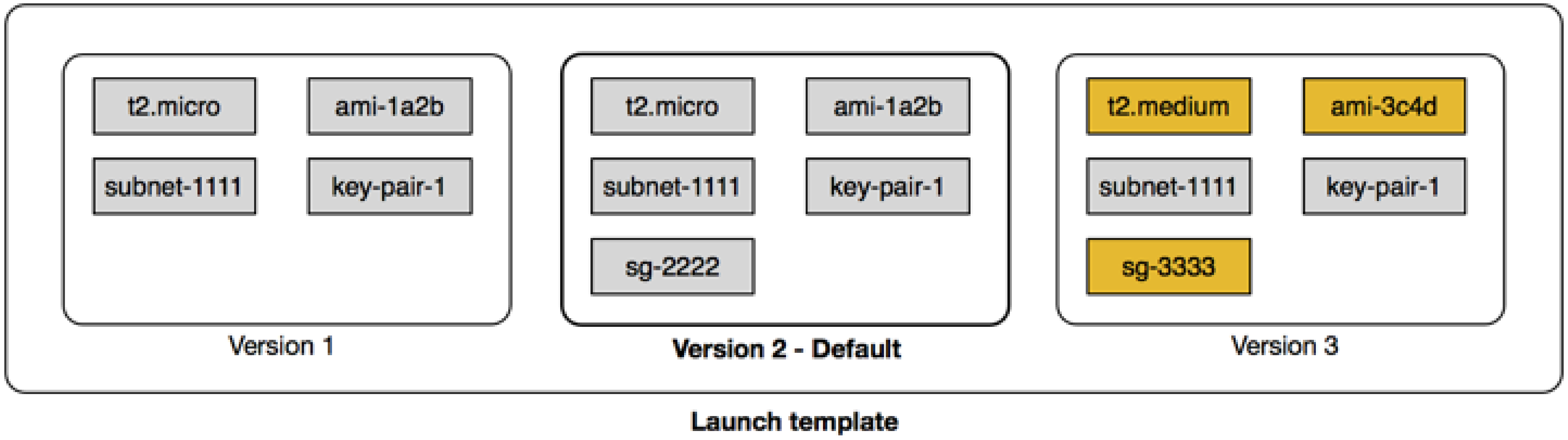
Taken from https://jayendrapatil.com/aws-auto-scaling/ (18/07/2024)

Launch Templates vs Configuration

Launch template is a better option tan Launch Configuration because it allows **reused** and add configuration to new templates.

In addition, launch templates allows all configuration from AutoScaling groups, for instance mix OnDemand and Spot Instance on the same group or launch on Dedicated Hosts.

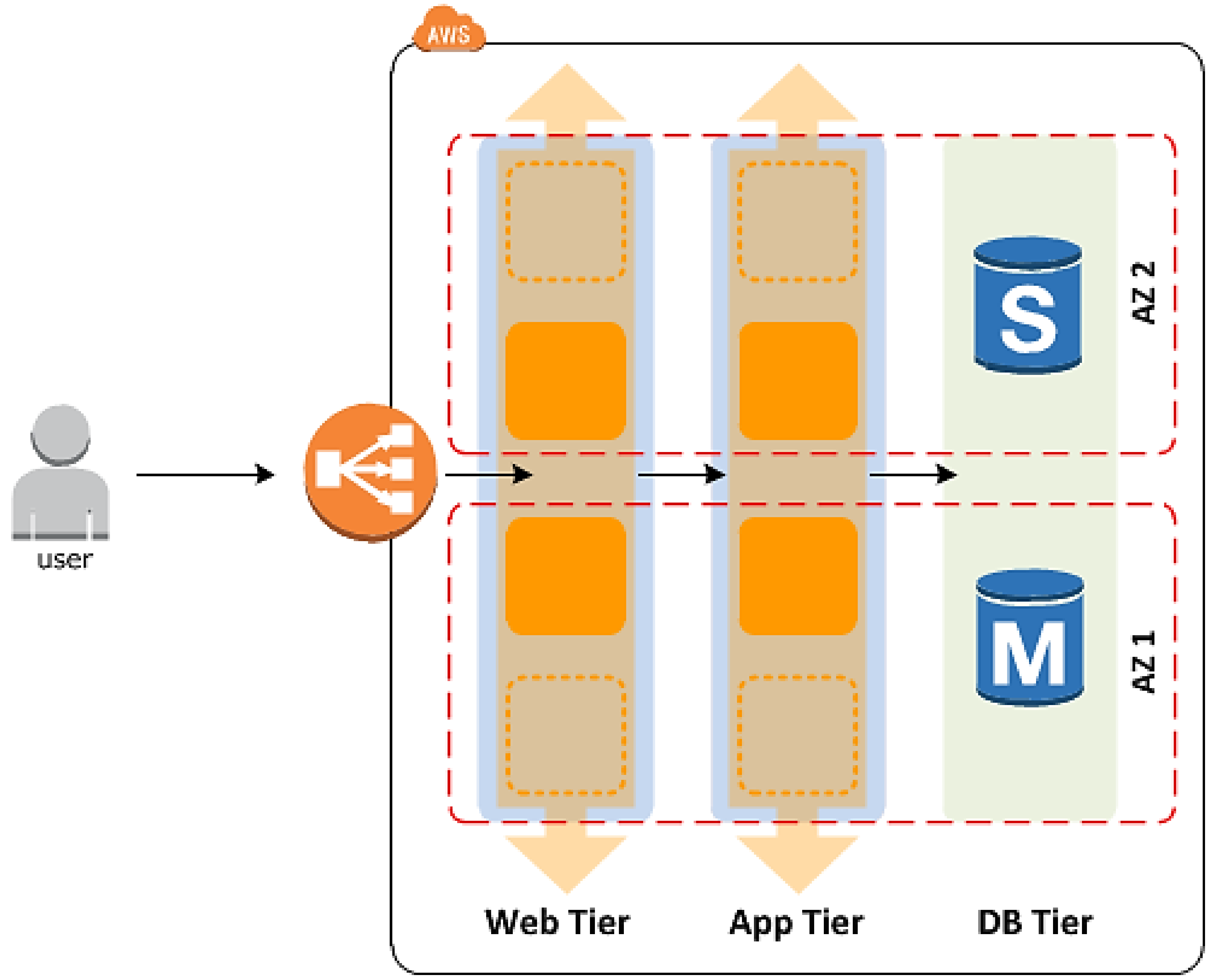
Both options contains the same information about instances: type, AMI, Keypair, Security Group, storage, etc.



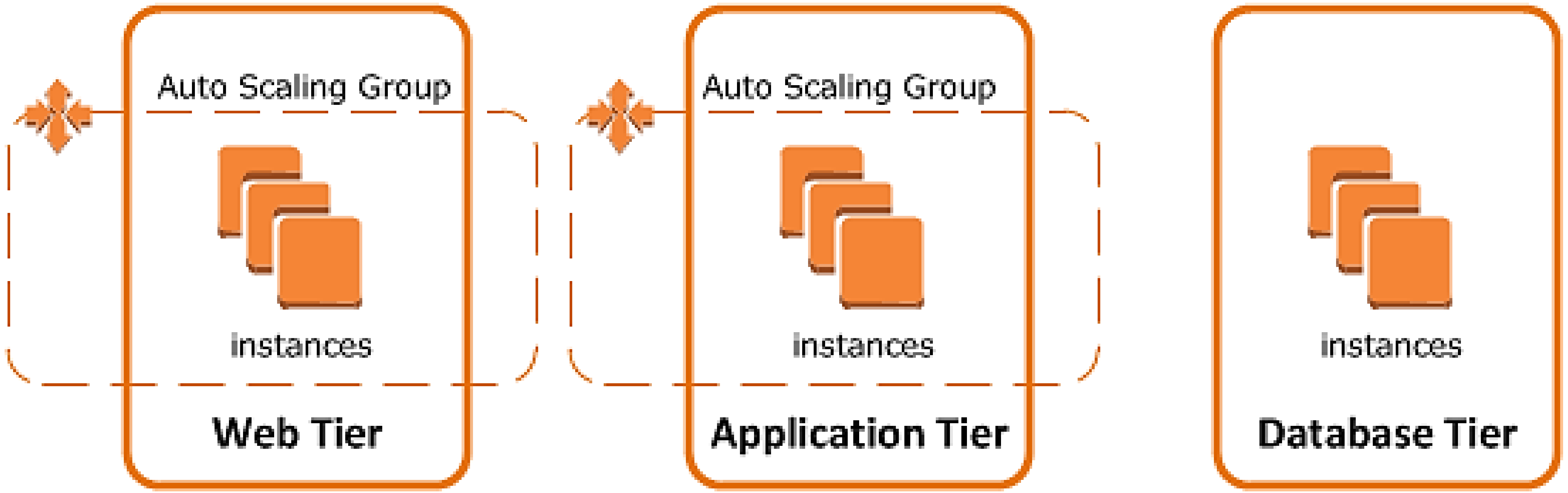
Taken from https://docs.aws.amazon.com/autoscaling/ec2/userguide/LaunchTemplates.html and https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-templates.html (18/07/2024)

ASG Components

fmorenod.co
©2024

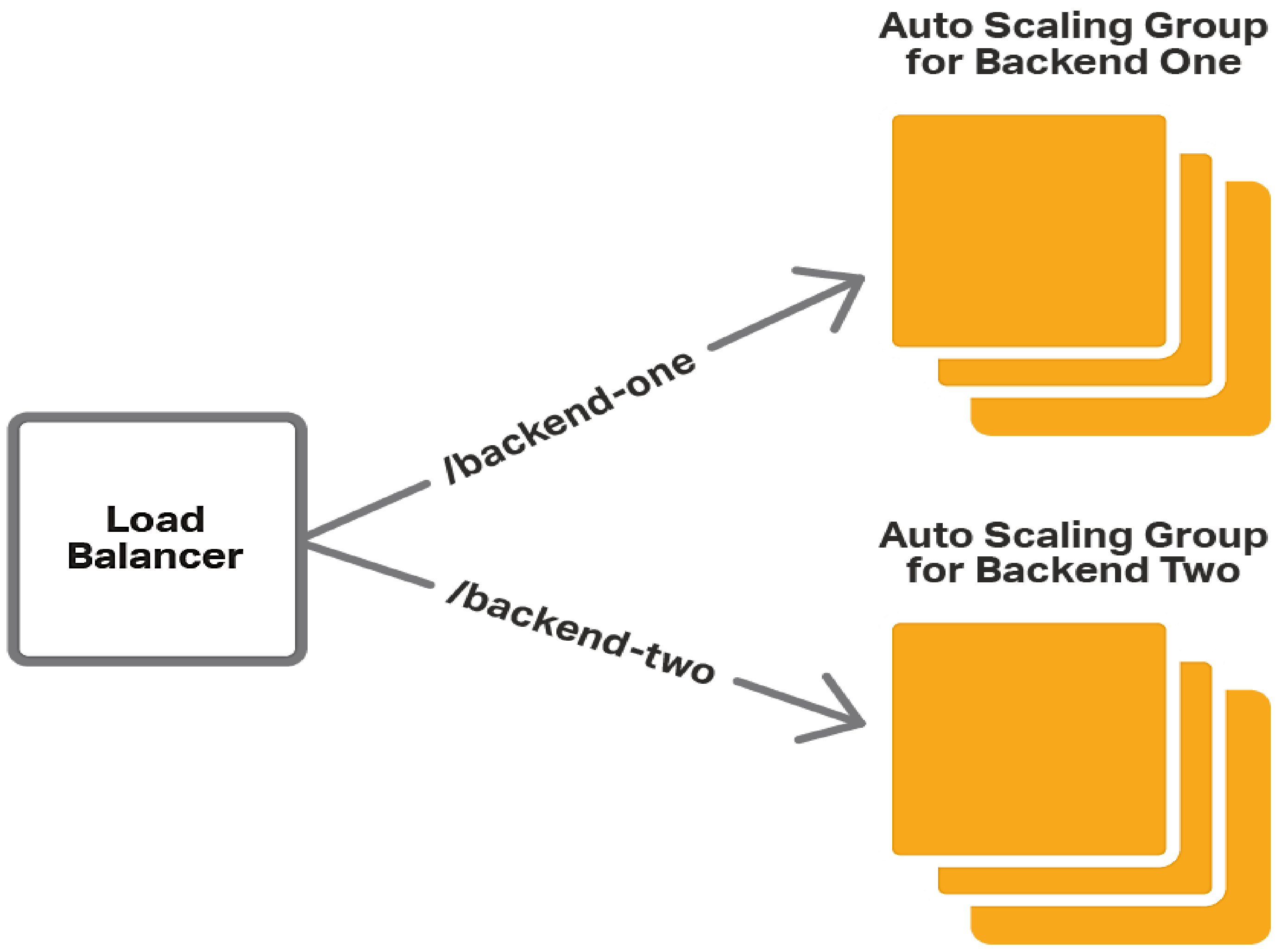


ASG Components and Policies



Policies:
- Manual
- Scheduled
- Dynamic
- Dynamic: Target Tracking Policy
- Dynamic: Simple Tracking Policy → Wait until health check and cooldown before evaluate
- Dynamic: Step Tracking Policy

- Metrics on CloudWatch apply on Dynamic.


Auto Scaling Group for Backend One
Load Balancer
/backend-one
/backend-two
Auto Scaling Group for Backend Two

ASG Best Practices

1. Choose LB (i.e. SSL Termination, Globl Balancing with R53)
2. Choose the appropriate instance families and sizing based on the workload in that ASG
3. Consider placement groups in ASG –if applicable-
4. Use Launch Templates
5. Group instances by purpose (i.e. Domains)
6. Set up health checks
7. Utilize target tracking scaling policies
8. Implement cooldown periods
9. Configure notifications
10. Implement proper security (NACLs? Sec Groups)
11. Implement instance termination policies
12. Distribute instances across Availability Zones
13. Use Auto Scaling lifecycle hooks (Launch/Terminate)

Taken from https://www.nops.io/blog/aws-ec2-autoscaling/ (18/07/2024)

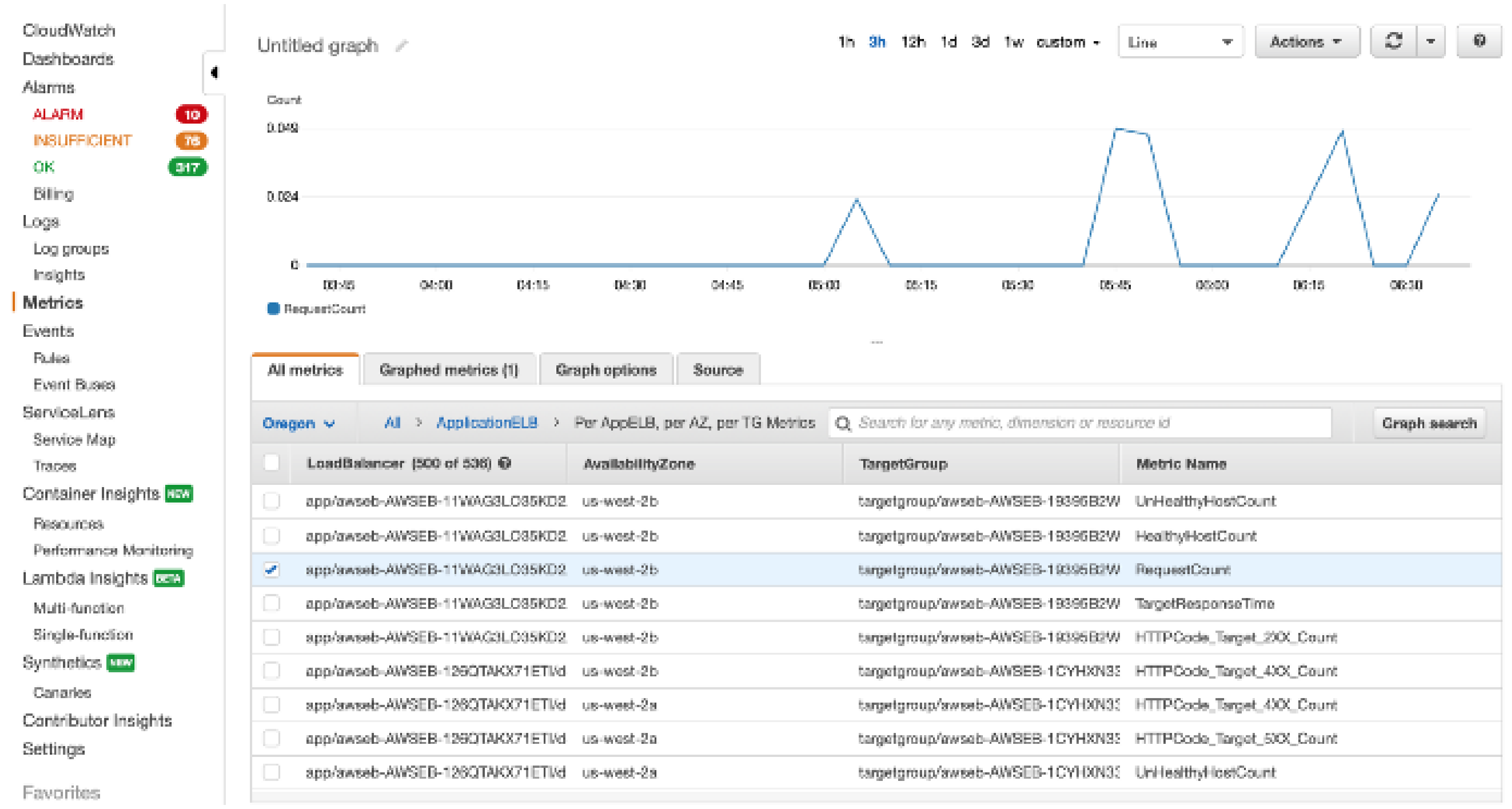
CloudWatch

Amazon CloudWatch Alarm Rule Event (event-based) Event (time-based)

- Monitoring and Observability Service to cover AWS Resources, especially computing services such Applications and Containers, in addition Logs, Databases. It collects and track metrics from those resources.
- You can create: Dashboards, Insights, Alarms and Events. You collect: Metrics and Logs.
- Services associated: Cloud Trail, IAM
- Services to trigger actions: SNS – Simple Notification Service, AutoScaling Groups, Lambda; however for Events you can trigger more services.

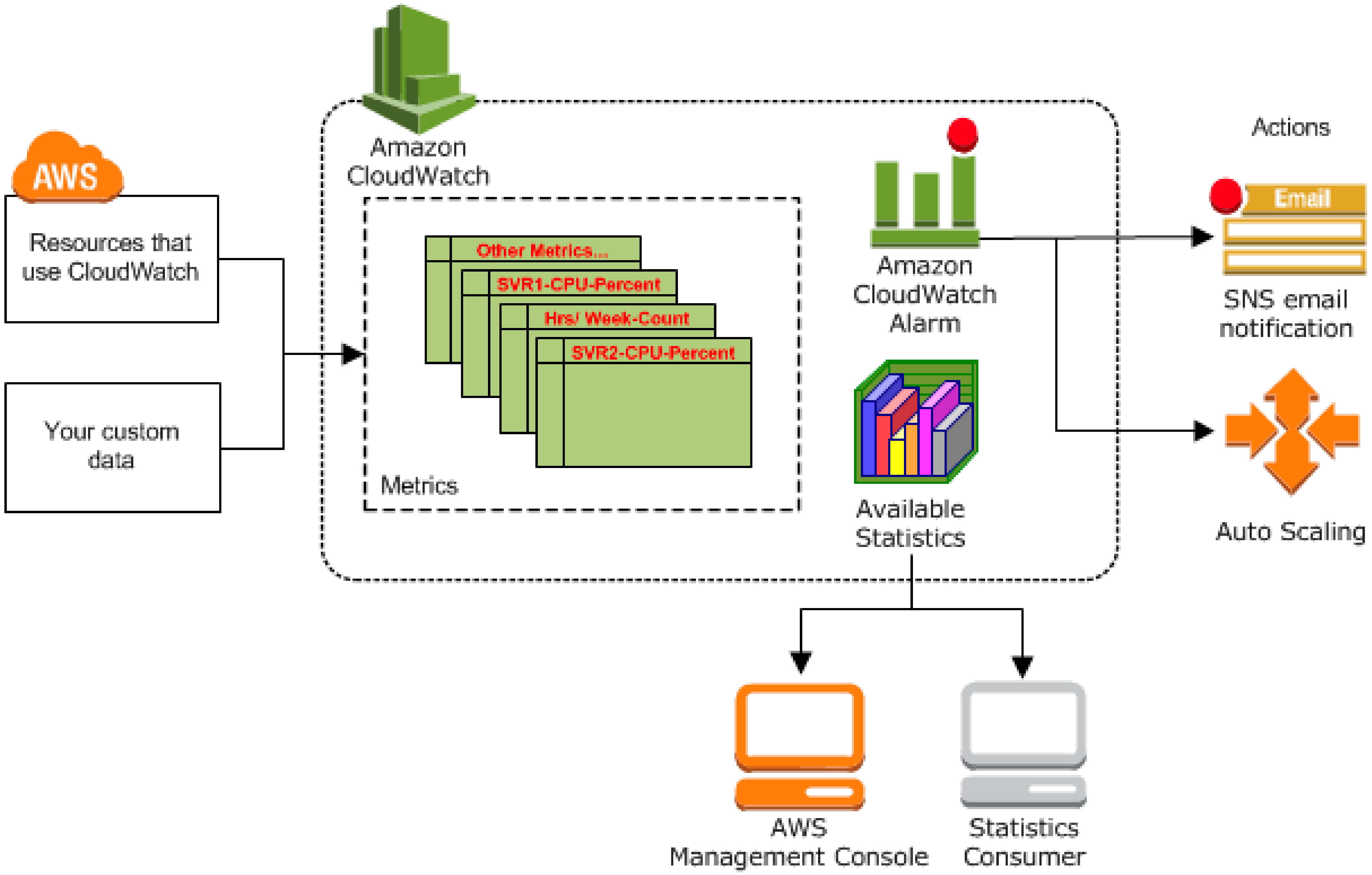
CloudWatch Key Terms

Metrics:
<u>Metrics: Timestamp (UTC), Retention</u>
High-Resolution Monitoring: - Retention: 3 h.
Detailed monitoring: 1 mins (Paid) – R: 15 d.
Basic monitoring: 5 mins – R: 63 d.
<u>Statistics:</u> Max, Min, Average, Sum, Count, Percentiles
<u>Units, Period</u>
<u>Aggregation</u>
Concepts: Namespaces (i.e. Service) / Dimensions (i.e. Group Metrics)
Alarms: Trigger SNS and/or AutoScaling Groups
Events: Trigger something by Event or Schedule



	LoadBalancer (500 of 536)	AvailabilityZone	TargetGroup	Metric Name
☐	app/awseb-AWSEB-11WAG3LC35KD2	us-west-2b	targetgroup/awseb-AWSEB-19396B2W	UnHealthyHostCount
☐	app/awseb-AWSEB-11WAG3LC35KD2	us-west-2b	targetgroup/awseb-AWSEB-19396B2W	HealthyHostCount
✓	app/awseb-AWSEB-11WAG3LC35KD2	us-west-2b	targetgroup/awseb-AWSEB-19396B2W	RequestCount
☐	app/awseb-AWSEB-11WAG3LC35KD2	us-west-2b	targetgroup/awseb-AWSEB-19396B2W	TargetResponseTime
☐	app/awseb-AWSEB-11WAG3LC35KD2	us-west-2b	targetgroup/awseb-AWSEB-19396B2W	HTTPCode_Target_2XX_Count
☐	app/awseb-AWSEB-126QTAKX71ETVd	us-west-2b	targetgroup/awseb-AWSEB-1CYHXN3	HTTPCode_Target_4XX_Count
☐	app/awseb-AWSEB-126QTAKX71ETVd	us-west-2a	targetgroup/awseb-AWSEB-1CYHXN3	HTTPCode_Target_4XX_Count
☐	app/awseb-AWSEB-126QTAKX71ETVd	us-west-2a	targetgroup/awseb-AWSEB-1CYHXN3	HTTPCode_Target_5XX_Count
☐	app/awseb-AWSEB-126QTAKX71ETVd	us-west-2a	targetgroup/awseb-AWSEB-1CYHXN3	UnHealthyHostCount

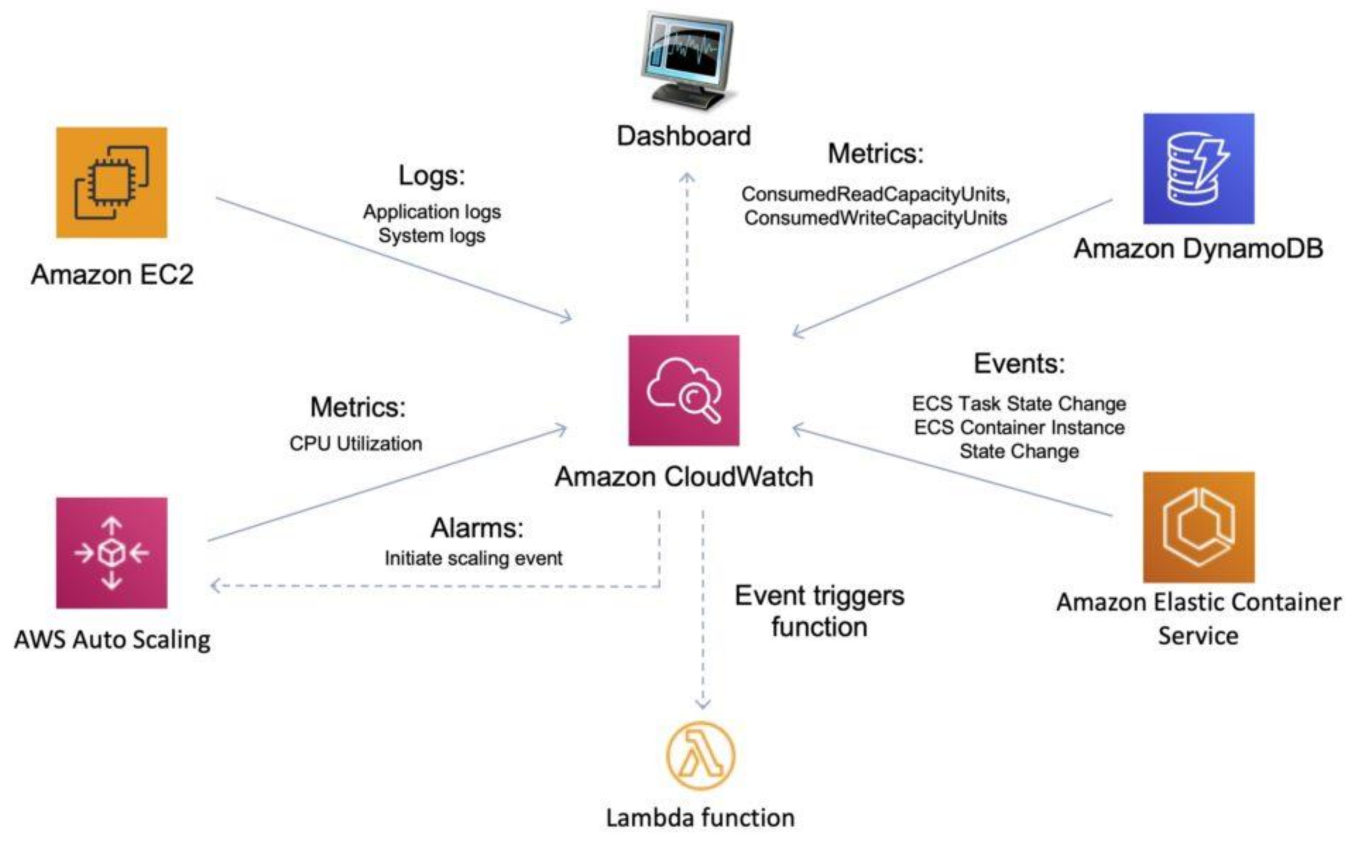
CloudWatch



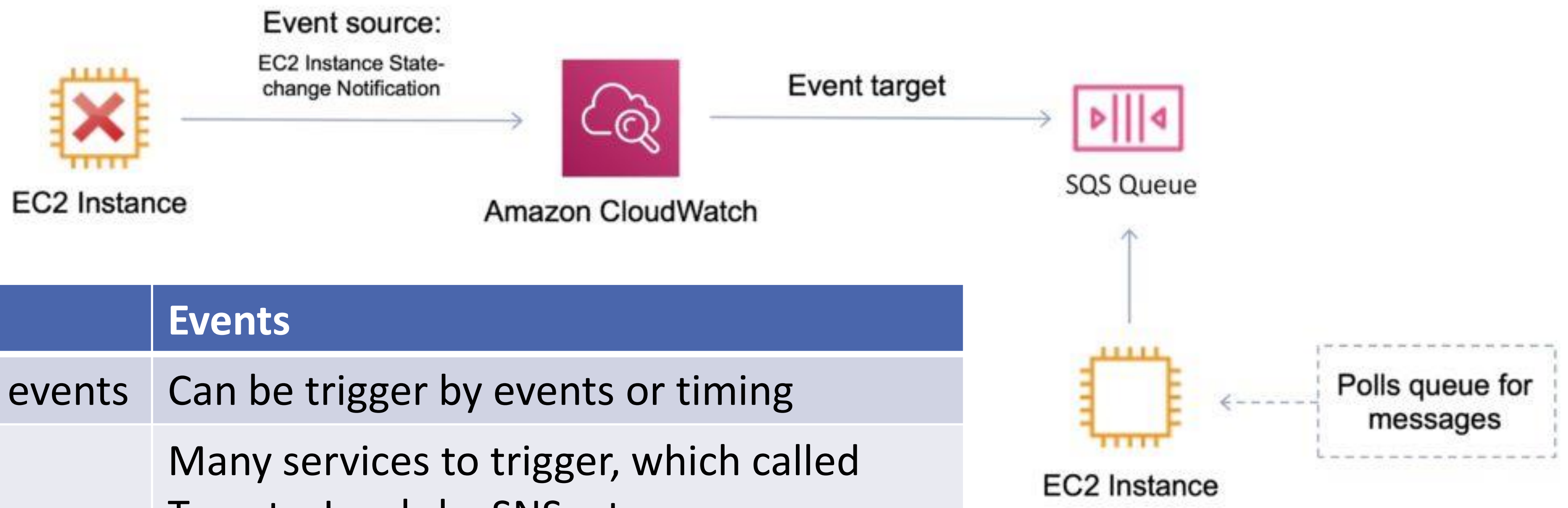
Taken from https://jayendrapatil.com/aws-cloudwatch-overview/ (18/07/2024)

CloudWatch Alarms vs Events

CloudWatch Alarms
use <u>conditions</u> to trigger <u>notifications/actions</u>



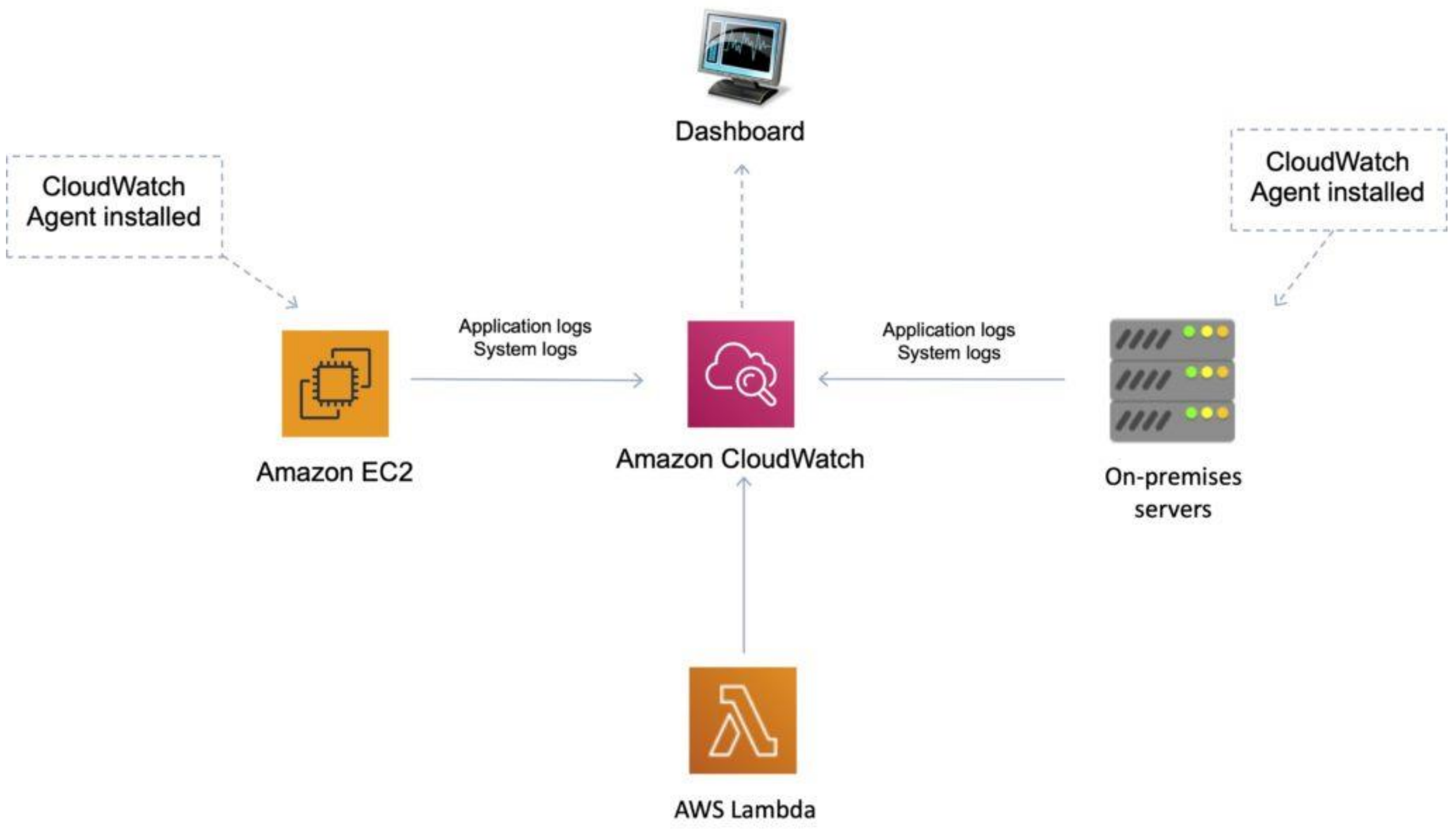
CloudWatch Events
use <u>rules</u> to trigger <u>targets</u>



Alarms	Events
Can be trigger by events/sustained events	Can be trigger by events or timing
Limited number of actions: SNS, AutoScaling Group, EC2 Actions.	Many services to trigger, which called Targets: Lambda, SNS, etc.
Watch Single Metric (Defined by a period)	Respond to Actions (Near real-time)
Can be added to Dashboard	No can be added to Dashboard

Taken from https://digitalcloud.training/amazon-cloudwatch/ and https://stackoverflow.com/questions/58205306/what-is-the-difference-between-a-cloudwatch-alarm-and-a-cloudwatch-event#:~:text=Some%20key%20differences%3A,change%20in%20your%20AWS%20environment) (18/07/2024)

Cloudwatch Logs



It receives Application and System Logs, and act as centralized repository and then, it can be feed by a Lambda or ElasticSearch Cluster.

Taken from https://digitalcloud.training/amazon-cloudwatch/ (18/07/2024)