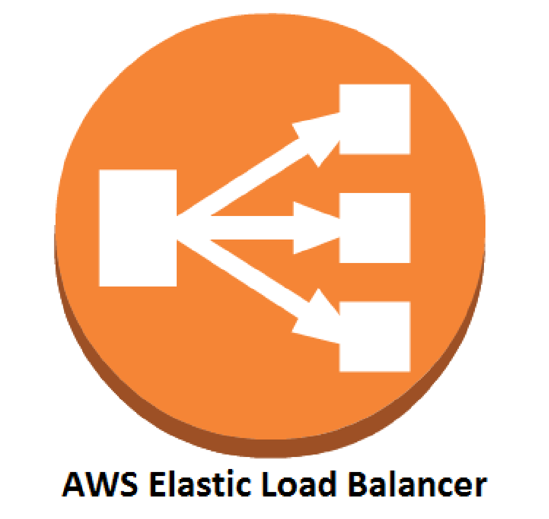
Elastic Load Balancing (ELB) & AutoScaling

Agenda

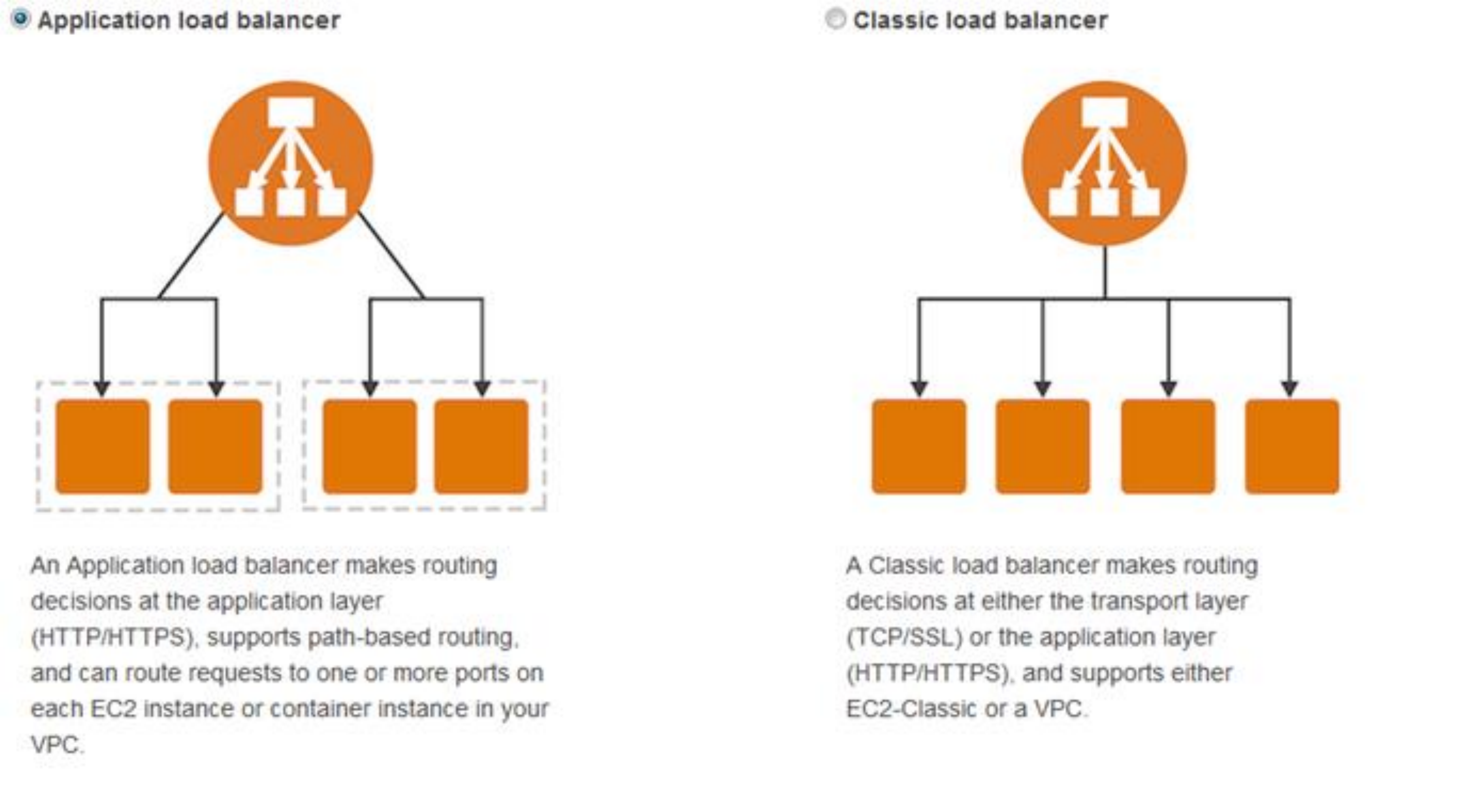
- What is AWS ELB
- Classic Load Balancer
  - Features
  - Health Check Configuration
  - Cross-Zone
  - Connection Draining
  - Sticky Sessions
  - Access Logs
  - Limitation
- Application Load Balancer
  - What is Application ELB
  - Features
  - Application Flow
  - Limitation
- Network Load Balancer
  - What is Network ELB
- AWS Autoscaling
- Hands-On Lab



What is AWS ELB

- ❑ ELB distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones.
- ❑ ELB increases the fault tolerance of your applications.
- ❑ The load balancer serves as a single point of contact for clients.
- ❑ Enable health checks.
- ❑ Types of load balancers:
  - o Application Load Balancers
  - o Network Load Balancers
  - o Classic Load Balancers

AWS Load Balancer Types



● Application load balancer

An Application load balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports on each EC2 instance or container instance in your VPC.

○ Classic load balancer

A Classic load balancer makes routing decisions at either the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS), and supports either EC2-Classic or a VPC.

AWS ELB: Features

Availability Zone

Cross-Zone

Request Routing

Connection Draining

Internet-facing Load Balancer

Internal Load Balancer

Pay-Only What You Use

AWS ELB: Health Check Configuration

- Ping Protocol
- Ping Port
- Ping Path
- Response Timeout
- HealthCheck Interval
- Unhealthy Threshold
- Healthy Threshold

AWS ELB: Cross-Zone

- Cross-zone load balancing distribute incoming requests evenly across the Availability Zones enabled for your load balancer.

    - Example, if you have 10 instances in Availability Zone us-west-2a and 2 instances in us-west-2b, the requests are distributed evenly across all 12 instances if cross-zone load balancing is enabled.

    - Otherwise, the 2 instances in us-west-2b serve the same number of requests as the 10 instances in us-west-2a.

AWS ELB: Connection Draining

- ❑ Connection draining is use to stops sending requests to instances that are de-registering or unhealthy.
- ❑ Complete in-flight requests made to instances that are de-registering or unhealthy.
- ❑ Specify a maximum time for the load balancer to keep connections alive
- ❑ State:
  - InService: Instance deregistration currently in progress
  - OutOfService: Instance is not currently registered with the LoadBalancer

AWS ELB: Sticky Sessions

- Classic Load Balancer routes each request independently to the registered instance with the smallest load.

- You can use the *sticky session* feature (also known as *session affinity*), which enables the load balancer to bind a user's session.

- Duration-Based Session Stickiness
  - The stickiness policy configuration defines a cookie expiration, which establishes the duration of validity for each cookie.
  - After a cookie expires, the session is no longer sticky.
  - If an instance fails or becomes unhealthy, the load balancer stops routing requests to that instance.
  - The request is routed to the new instance as if there is no cookie and the session is no longer sticky.

AWS ELB: Access Logs

- ELB provides access logs that capture detailed information about requests sent to ELB.

- Each log contains information such as time the request, the client's IP address, etc.

- ELB captures the logs and stores them in the Amazon S3 bucket.

- You can use these access logs to troubleshoot issues.

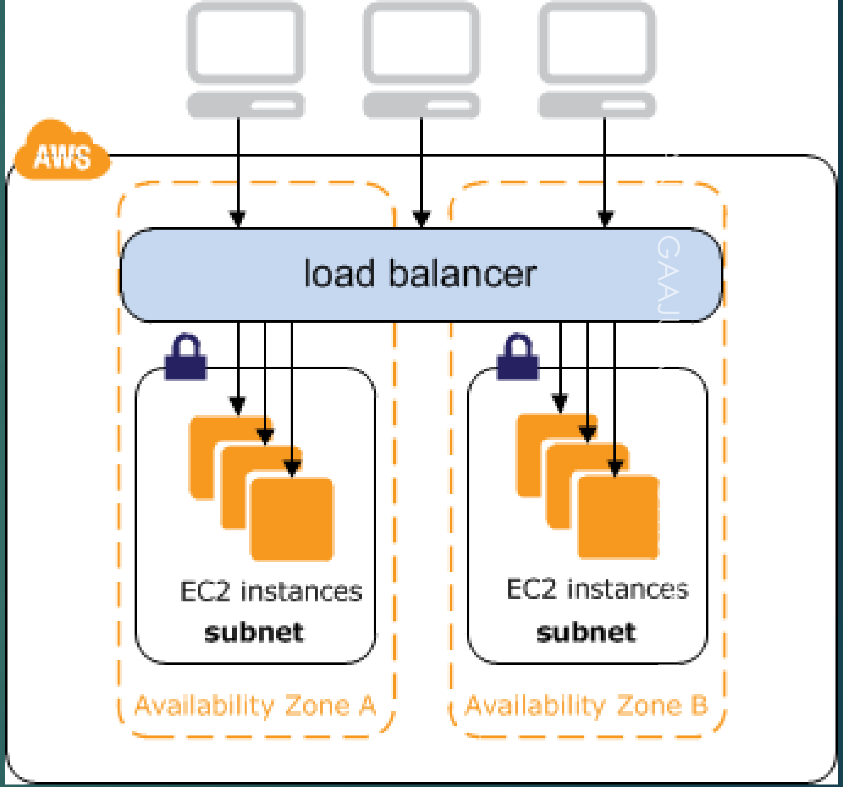
- Access logging is an optional feature of Elastic Load Balancing that is disabled by default.

- Syntax
  - Each log entry contains the details of a single request made to the load balancer.
  - timestamp elb client:port backend:port request_processing_time backend_processing_time response_processing_time elb_status_code backend_status_code received_bytes sent_bytes "request" "user_agent" ssl_cipher ssl_protocol

AWS ELB: Limitation

Resource	Default Limit
Load balancers per region	20
Listeners per load balancer	100
Security groups per load balancer	5
Subnets per Availability Zone per load balancer	1

Application Load Balancer



What is AWS Application ELB

- An Application Load Balancer functions at the application of the Open Systems Interconnection (OSI) model.

- It evaluates the listener rules to determine which rule to apply, and then selects a target from the target group for the rule action.

- Configure listener rules to route requests to different target groups based on the content of the application traffic.

- Configure health checks, which are used to monitor the health of the registered targets.

- Listeners support the HTTP/ HTTPS protocols.

AWS ELB: Features

Path-based Routing

Host-based Routing

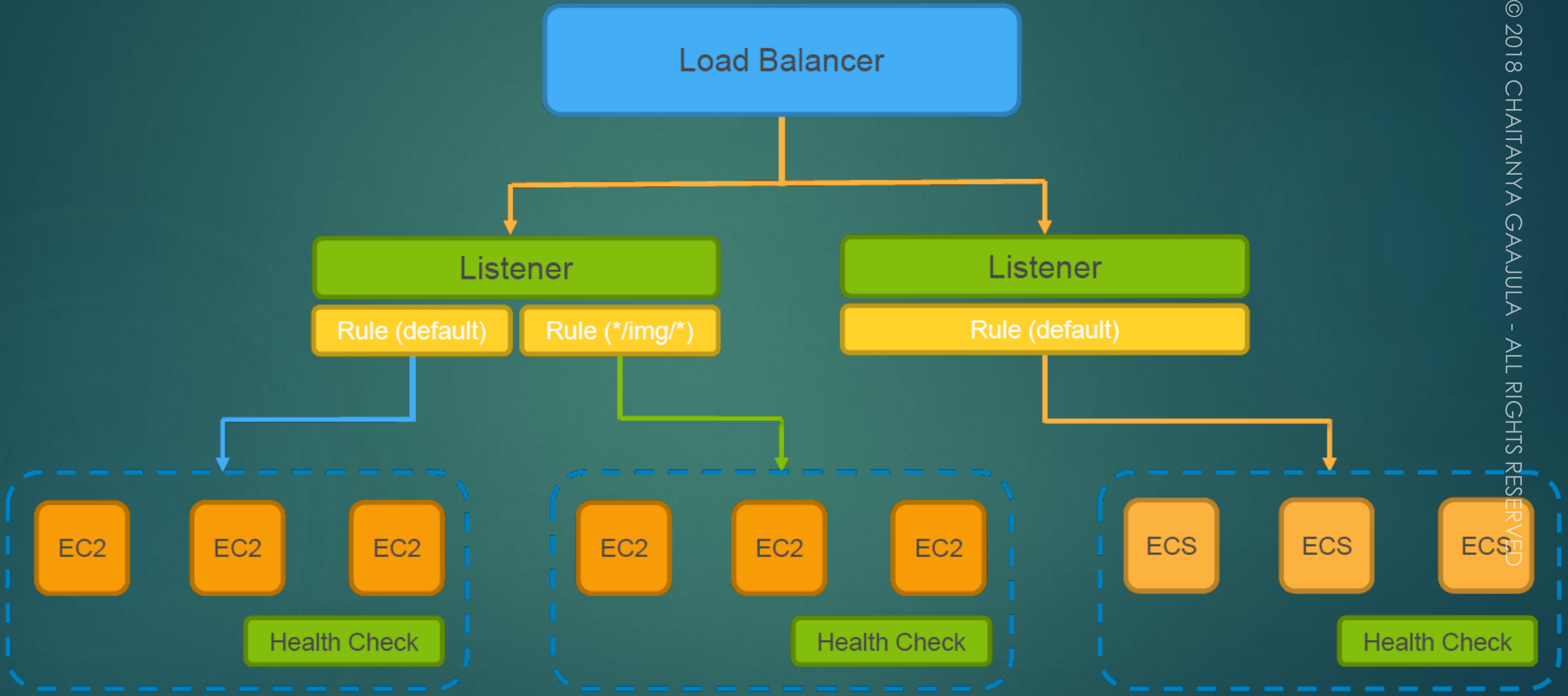
Path-based Routing

Multiple applications on a single EC2

Registering targets by IP address

Pay-Only What You Use

AWS ELB: Application Flow



AWS ELB: Limitation

Resource	Default Limit
Load balancers per region	20
Target groups per region	3000
Load balancers per target group	1
Targets per load balancer	1000
Targets per target group	1000
Listeners per load balancer	50
Rules per load balancer	100
Number of times a target can be registered per load balancer	100
Security groups per load balancer	5
Subnets per Availability Zone per load balancer	1
Certificates per listener	1
Conditions per rule (one host condition, one path condition)	2
Actions per rule	1
Target groups per action	1

Network Load Balancer

What is AWS Network ELB

- ❑ An Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model.
- ❑ Ability to handle volatile workloads and scale to millions of requests per second.
- ❑ Capable of handling millions of requests per second.
- ❑ Listeners support the TCP protocols.

Hands-On Lab

Hands-on Lab

- ❑ Configure Your Classic Load Balancer with 2 instances .

Auto Scaling

Agenda

- What is AWS Auto Scaling
- Auto Scaling Components
- Auto Scaling Group
- Auto Scaling Launch Configuration
- Auto Scaling Benefits
- Auto Scaling Lifecycle
- Auto Scaling Plans
- Manual Scaling
- Schedule Scaling
- Dynamic Scaling
- Auto Scaling Step Adjustment
- Auto Scaling Termination Policy
- Default Termination Policy
- Health Check
- Quiz
- Hands-On Lab

What is AWS AutoScaling

- **Scalability** is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged to accommodate that growth.

- Types of scaling:
  - **Horizontal Scaling** [scaling out and scaling in]
  - **Vertical Scaling** [scaling up and scaling down]

AWS AutoScaling: **Components**

Groups

Launch Configuration

Scaling Plans

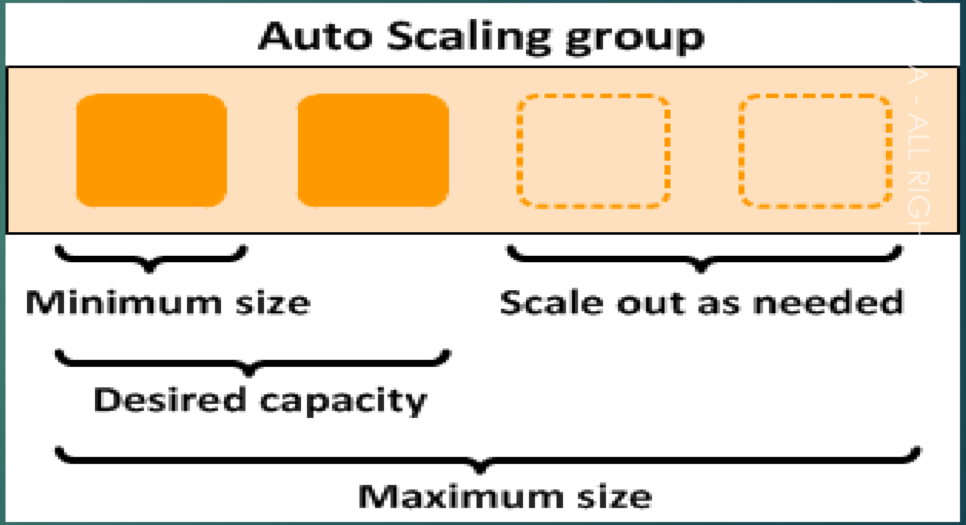
AWS AutoScaling: **Group**

Auto Scaling Group

Maximum Size

Desired Capacity

Minimum Size



AWS AutoScaling: **Launch Configuration**

Creating Launch Configuration from Scratch

Amazon Machine Image

Key Pair

Security Group

Instance Type

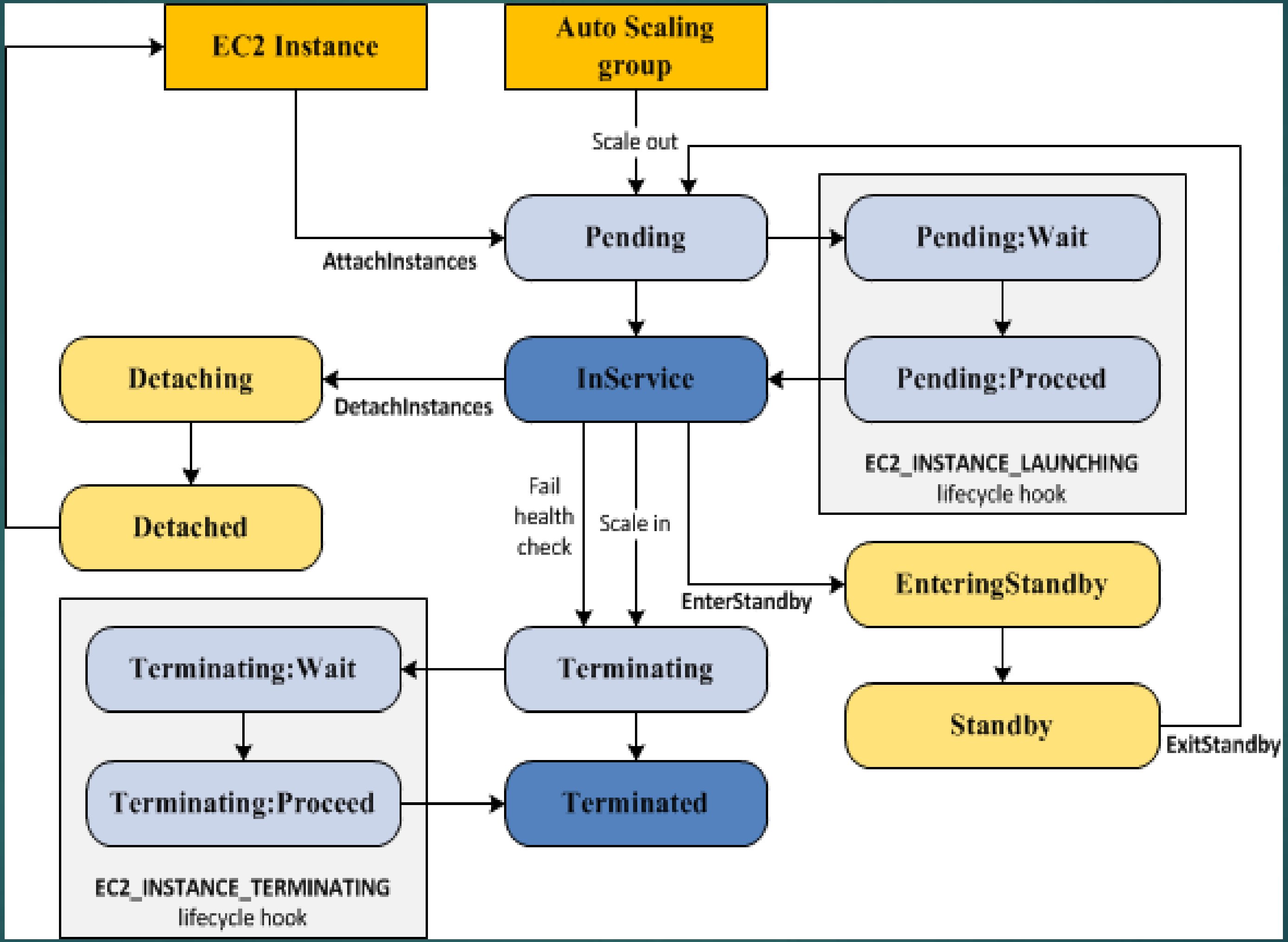
Block Device Mapping

Creating Launch Configuration from a running EC2 Instance

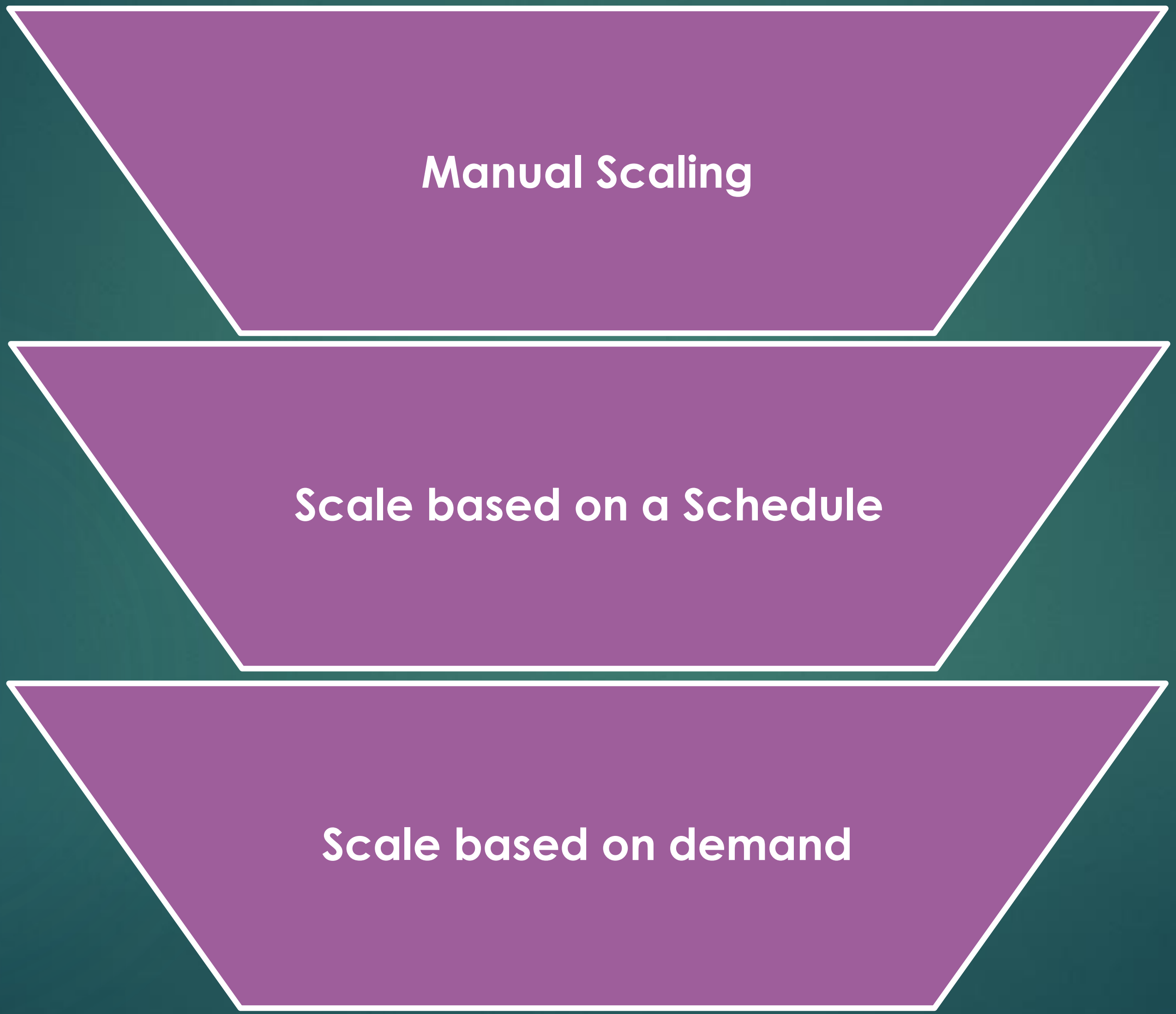
AWS AutoScaling: **Benefits**

- Fault Tolerance
- Multiple AZ(s)
- Availability
- Cost Management

AWS AutoScaling: **Lifecycle**



AWS AutoScaling: **Plans**



Manual Scaling

Scale based on a Schedule

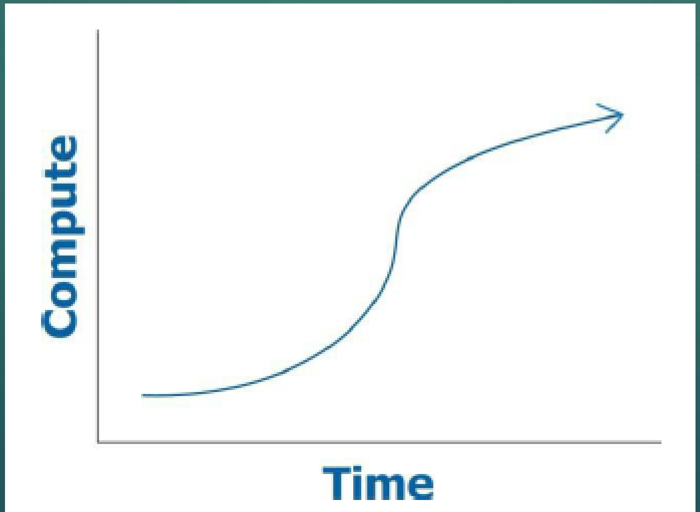
Scale based on demand

AWS AutoScaling: Manual Scaling

- At any time, you can change the size of an existing Auto Scaling group.

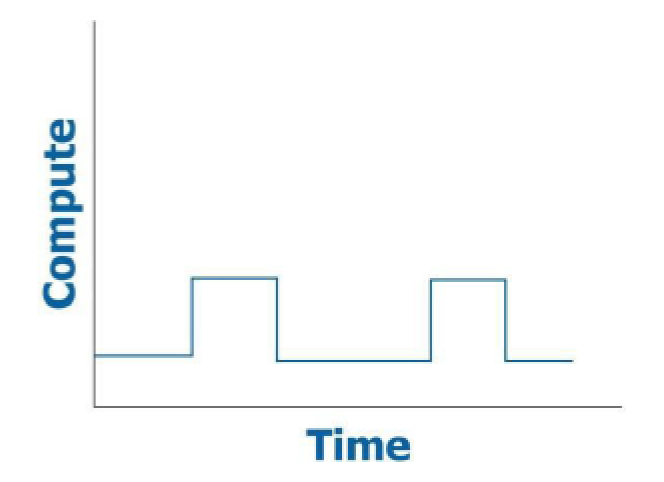
- Update the desired capacity of the Auto Scaling group, or update the instances that are attached to the Auto Scaling group.

- After changes, verify that your Auto Scaling group has launched/ terminated additional instance.



AWS AutoScaling: Schedule Scaling

- Scaling based on a schedule allows you to scale your application in response to predictable load changes.
- For example, every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday.
- You can plan your scaling activities based on the predictable traffic patterns.

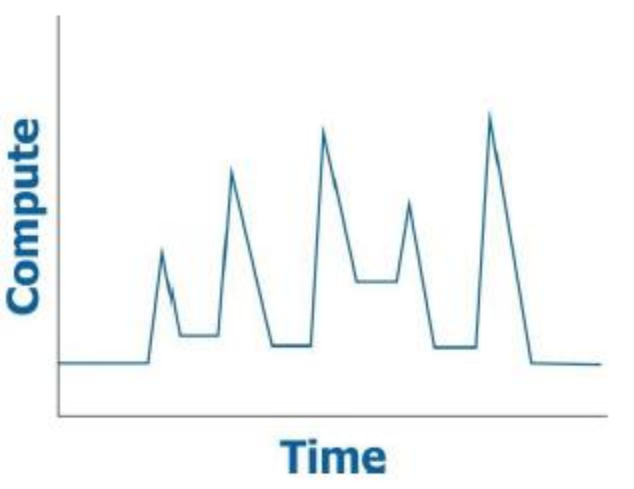


AWS AutoScaling: Dynamic Scaling

- When you use Auto Scaling to scale dynamically, you must define how you want to scale in response to changing demand.

- For example, say you have a web application that currently runs on two instances and you do not want the CPU utilization of the Auto Scaling group to exceed 70 percent.

- Scaling Policy Types:
  - Simple scaling
  - Step scaling
  - Target tracking scaling



AWS AutoScaling: Step Adjustment

Scale Out Policy				
Lower bound	Upper bound	Adjustment	Metric value	Changes
0	10	0	50 <= value < 60	maintains the desired capacity while the aggregated metric value is less than 60
10	20	10	60 <= value < 70	increases the desired capacity of the group by 1 instance, to 11 instances (add 10 percent of 10 instances)
20	null	30	70 <= value < +infinity	increase the desired capacity by another 3 instances, to 14 instances (add 30 percent of 11 instances, 3.3 instances, rounded down to 3 instances).

AWS AutoScaling: Step Adjustment

Scale In Policy				
Lower bound	Upper bound	Adjustment	Metric value	Changes
-10	0	0	40 < value <= 50	maintains the desired capacity while the aggregated metric value is greater than 40
-20	-10	-10	30 < value <= 40	if the metric value gets to 40, decreases the desired capacity of the group by 1 instance, to 13 instances, (remove 10 percent of 14 instances, 1.4 instances, rounded down to 1 instance).
null	-20	-30	-infinity < value <= 30	if the metric value falls to 30, decreases the desired capacity of the group by another 3 instances, to 10 instances, (remove 30 percent of 13 instances, 3.9 instances, rounded down to 3 instances).

AWS AutoScaling: Termination Policy

Oldest Instance

Newest Instance

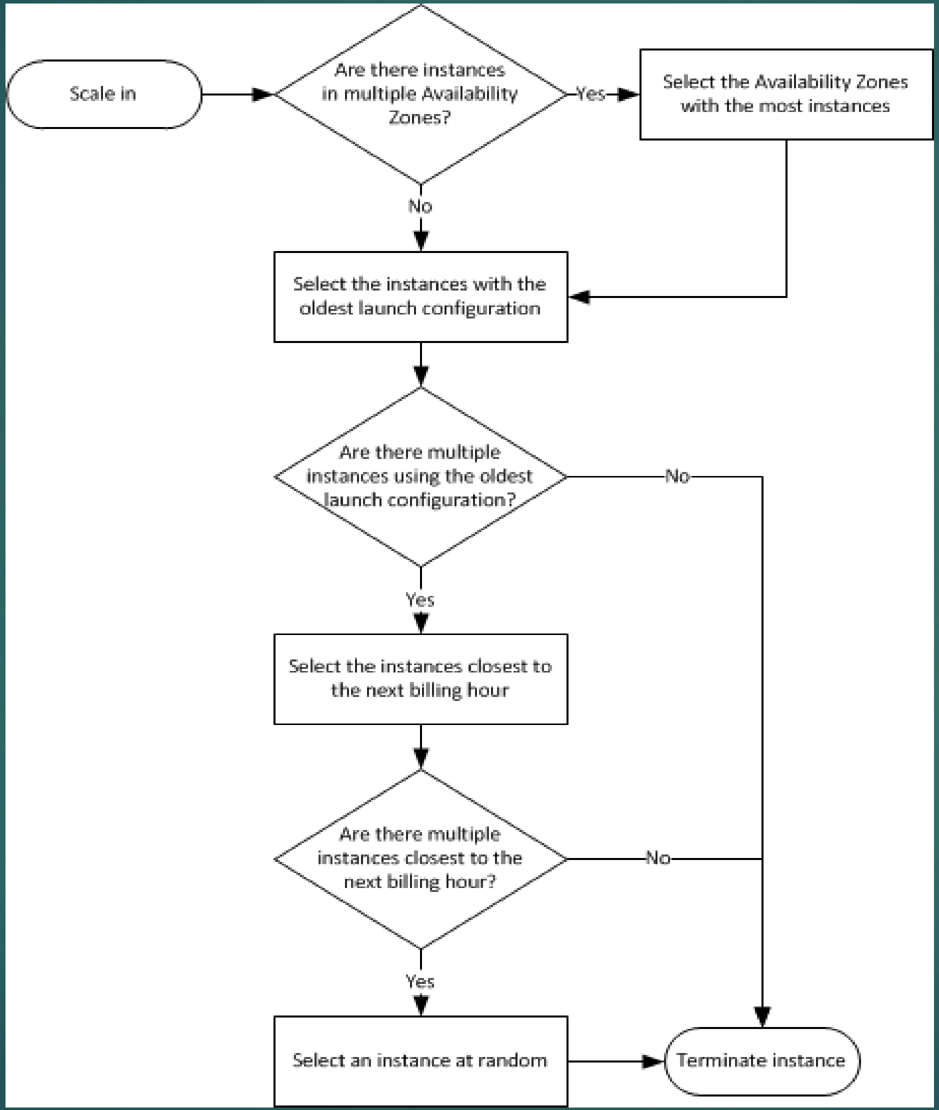
Closest To Next Instance Hour

Oldest Launch Configuration

Default

Instance Protection

AWS AutoScaling: Default Termination Policy


Scale in
Are there instances in multiple Availability Zones?
Yes
Select the Availability Zones with the most instances
No
Select the instances with the oldest launch configuration
Are there multiple instances using the oldest launch configuration?
No
Yes
Select the instances closest to the next billing hour
Are there multiple instances closest to the next billing hour?
No
Yes
Select an instance at random
Terminate instance

AWS AutoScaling: Health Check

- Auto Scaling determines the health status of an instance using one or more of the following:
  - Status checks provided by Amazon EC2.
  - Health checks provided by Elastic Load Balancing.
  - Custom health checks.

- By default, Auto Scaling health checks use the results of the EC2 status checks to determine the health status of an instance.

- If you attached a load balancer to your Auto Scaling group, you can configure Auto Scaling to mark an instance as unhealthy if Elastic Load Balancing reports the instance as **OutOfService**.

AWS AutoScaling: **Limitation**

Resource	Default Limit
Launch configurations per region	100
Auto Scaling groups per region	20
Scaling policies per Auto Scaling group	50
Scheduled actions per Auto Scaling group	125
Lifecycle hooks per Auto Scaling group	50

Hands – On –Lab

- Maintaining High Availability using Auto Scaling

Thank You