SECTION 2: -AWS Fundamentals ELB+ASG

**Vertical Scalability** : means increasing the size of the instance

Example : application running on t2.large

**Horizontal Scalability :** increasing the number of instance for the application

**High availability :** running application in atleast two AZ

Machine generated alternative text:
High Availability & Scalability For EC2 
• Vertical Scaling: Increase instance size (z scale up / down) 
• From: t2.nano - O.5G of RAM, I vCPU 
• To: u- - 12.3 TB of RAM, 448 vCPUs 
• Horizontal Scaling: Increase number of instances (z scale out / in) 
• Auto Scaling Group 
• Load Balancer 
• High Availability: Run instances for the same application across multi AZ 
• Auto Scaling Group multi AZ 
• Load Balancer multi AZ 

**Load Balancer :** are server that forward internet traffic to multiple server(ec2 instance)

Machine generated alternative text:
User 1 
User 2 
User 3 
Loa Balancer 
EC2 
Instance 
EC2 
Instance 
Instance 

* Forwards user traffic to multiple instances
* Spreads load across multiple downstream instances
* Handle failures of application instances
* Regular health checks of instances
* SSL termination (No need to have SSL certificates on instances)
* Stickiness with cookies (Ensures all request from a user session are forwarded to same instance)
* High availability across zones (if an AZ fails)
* Separate public traffic from private traffic

**Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.**

Machine generated alternative text:
Health Checks 
• Health Checks are crucial for Load Balancers 
• They enable the load balancer to know if instances it forwards traffc to 
are available to reply to requests 
• The health check is done on a port and a route (/health is common) 
• If the response is not 200 (OK), then the instance is unhealthy 
Classic Load 
Balancer (VI) 
Health Checks 
Port 4567 
Route /health 
EC2 
Instances 

**health check** : if it is ok means ec2 instance is up then load balancer will route traffic

Machine generated alternative text:
Types of load balancer on AWS 
• AWS has 3 kinds of managed Load Balancers 
• Classic Load Balancer (VI - old generation) — 2009 
. HTTPS, TCP 
• Application Load Balancer (v2 - ne —2016 
• Network Load Balancer (v2 - new en ,4dOn) 2017 
• TCR TLS (secure TCP) & UDP 
• Overall, it is recommended to use the newer / v2 generation load balancers as they 
provide more features 
• You can setup internal (private) or external (public) ELBs 

Important:

Machine generated alternative text:
Load Balancer Security Groups 
LOAD BALANCER 
HTTPS/HTTP 
From anywhere 
Users 
Load Balancer Security Group: 
Port Range 
80 
HTTP Restricted 
to Load balancer 
Source 
0.0.0.0,'0 
0.0.0.0/0 
Source (j) 
EC2 
Description O) 
Allow HTTP from an... 
Allow HTTPS from 
Type O) 
Protocol 
TCP 
TCP 
Application Security Group: Allow traffic only from Load Balancer 
Type (0) 
Protocol (j) 
TCP 
Port Range O) 
80 
Description O) 
Allow Traffic only... 
sg-054b5ff5ea02f2b6e (load-b 

Ec2 security group(application security) references to load balancer security group which refer to users

Note : ec2 have own security group and load balancer have own security group

Machine generated alternative text:
Load Balancer 
Good to Know 
• LBS can scale but not instantaneously — contact AWS for a "warm-up" 
• Troubleshooting 
• 4xx errors are client induced errors 
• 5xx errors are application induced errors 
• Load Balancer Errors 503 means at capacity or no registered target 
• If the LB can't connect to your application, check your security groups! 
• Monitoring 
• ELB access logs will log all access requests (so you can debug per request) 
• CloudWatch Metrics will give you aggregate statistics (ex: connections count) 

2)Classic Load Balancer

Machine generated alternative text:
Classic Load Balancers (VI) 
internal 
CLB 
EC2 
• Supports TCP (Layer 4), HIT P & 
HTTPS (Layer 7) 
• Health checks are TCP or H 
based 
• Fixed hostname 
XXX.region.elb.amazonaws.com 
listener 
ient 

**Step 1:) load balancer -classic load balancer**

**Demo**

Machine generated alternative text:
Step 1 : Define Balancer 
Basic ConfWatbn 
LB 
('72.3' 

**Fill all tag and create clb :**

**IMPORTANT** once it created then check status if it is inService then it passing health check

Then load balancer is ready

Machine generated alternative text:
-oad balancer: MyFirstCLB 
Listeners 
Description 
Instances 
Health check 
Monitoring 
Availability Zone 
eu-west-3b 
Tags 
Migration 
Status 
lnService 
Connection Draining: Enabled, 300 seconds (Edit) 
Edit Instances 
Instance ID 
i-09aec878c3d4dfa3b 
Name 
First Instance 
Actions 
Remove from Load Balancer 

**Copy dns name from description and enter in url**

MyFirstCLB-1346685099.eu-west-3.elb.amazonaws.com

**Output Hello world**

<http://35.180.35.138/> : ec2 instance ip is also working:

So this means clb is working

NOTE: as of now we can access both but we need to exposed load balancer not ec2 directly

**So for this just go to security group and clb allow http inbound and for ec2 linked**

**Clb load balancer by searching sg**

Machine generated alternative text:
Edit inbound rules 
TCP 
NOTE: éts 
tr•t to a vey CM 
HTrp tramc 
trut 

**3)Application Load Balancer Hands on**

Machine generated alternative text:
Application Load Balancer (v2) 
• Application load balancers is Layer 7 (HIT P) 
• Load balancing to multiple HIT P applications across machines 
(target groups) 
• Load balancing to multiple applications on the same machine 
(ex: containers) 
• Support for HITP/2 and WebSocket 
• Support redirects (from HIT P to HIT PS for example) 

Machine generated alternative text:
• ALB are a great fit for micro services & container-based application 
(example: Docker & Amazon ECS) 
• Has a port mapping feature to redirect to a dynamic port in ECS 

Machine generated alternative text:
- Ion oa alancer 
(ALB) with Hands On 
HTTP Based Traffic 
O 
0.0 
o 
o 
c 
o 
www 
www 
Route /user 
External 
Application 
Load Balancer 
Route /search 

Machine generated alternative text:
r (ALB) with Hands On 
Target Groups 
• EC2 instances (can be managed by an Auto Scaling Group) — H IT P 
• ECS tasks (managed by ECS itself) - HTTP 
• Lambda functions — HIT P request is translated into a JSON event 
• IP Addresses — must be private IPS 
• ALB can route to multiple target groups 
• Health checks are at the target group level 

Machine generated alternative text:
Application Load Balancer (v2) 
Good to Know 
• Fixed hostname (XXX.region.elb.amazonaws.com) 
• The application servers don't see the IP of the client directly 
• The true IP of the client is inserted in the header X-Fonwarded-For 
• We can also get Port (X-Forwarded-Port) and proto (X-Fotwarded-Proto) 
Client IP 
12.34.56.78 
Load Balancer IP 
(Private IP) 
Connection termination 
EC2 
Instance 

NOTE:

**Once created ALB IT SHOW PROVISIONING**

Machine generated alternative text:
Name 
MyFirstALB 
DNS name 
MyFirstALB-948282906.eu-... 
State 
rowsormn 
VPC ID 
vpc-c14dafa9 
Availability Zones 
eu-west-3a, eu-west-3b... 

Machine generated alternative text:
Name 
MyFirstALB 
DNS name 
MyFirstALB-948282906.eu-... 
State 
active 
VPC ID 
vpc-c14dafa9 
Availability Zones 
eu-west-3a, eu-west-3b. 

**Add instance in target group**

Application (V2 - New - 2016) (Recommended)

* Layer 7 (HTTP level)
* Balance multiple HTTP apps across machines
* Balance multiple apps on same machine
* Balance based on route in URL
* Balance based on Hostname in URL
* Great for microservics and container-based apps (Docker, Amazon ECS)
* Port mapping feature to redirect to dynamic port
* Can have multiple Target Groups
* Each target group can have multiple EC2 instances and a health checkup logic
* Load Balancer redirects traffic from HTTP to target group

1. Network Load Balancer( Hands On)

Note : NLB have static ip while ALB and CLB does not have static IP but static hostname

Machine generated alternative text:
Network Load Balancer (v2) 
• Network load balancers (Layer 4) allow to: 
• Forward TCP & UDP traffc to your instances 
• Handle millions of request per seconds 
• Less latency 1 00 ms (vs 400 
or AL 
• NLB has one static IP per AZ, d up rts assigning Elastic IP 
(helpful for whitelisting specific IP 
• NLB are used for extreme performance, TCP or UDP traffic 
• Not included in the AWS free tier 

Machine generated alternative text:
Network Load Balancer (v2) 
TCP (Layer 4) Based Traffic 
O 
O 
www 
www 
TCP + Rules 
External 
Network Load 
Balancer (v2) 
TCP + Rules 
a-yo 
TCP 
HTTP 

Machine generated alternative text:
Edit inbound rules 
Type O) 
TCP 
TCP 
TCP 
(0 
TCP tramc from. „ 
SH from . 
NOTE: Any Orl re"t a 
on that to a •zy of 

1. **Load Balancer Stickiness**

Machine generated alternative text:
Load Balancer Stickiness 
• It is possible to implement stickiness so 
that the same client is always redirected 
to the same instance behind a load 
balancer 
• This works for Classic Load Balancers & 
Application Load Balancers 
• The "cookie" used for stickiness has an 
expiration date you control 
• Use case: make sure the user doesn't lose 
his session data 
• Enabling stickiness may bring imbalance to 
the load over the backend EC2 instances 
Client A 
EC2 
Instance 
Client B 
Client C 
EC2 
Instance 

**Stickiness**

: if same request come from same client it will go to same target group from load balancer

Stickiness can be enabled at the target group level

6)Elastic Load Balancer -cross zone load balancing

Machine generated alternative text:
Cross-Zone Load Balancing 
e 
AZ 2 
e 
e 
• With Cross Zone Load 
Balancing: each load 
balancer instance 
distributes evenly 
across all registered 
instances in all AZ 
• Otherwise, each load 
balancer node 
distributes requests 
evenly across the 
registered instances in 
its Availability Zone 
only. 
e 
WITH 
WITHOUT 

Machine generated alternative text:
Cross-Zone Load Balancing 
• Classic Load Balancer 
• Disabled by default 
• No charges for inter AZ data if enabled 
• Application Load Balancer 
• Always on (can't be disabled) 
• No charges for inter AZ data 
• Network Load Balancer 
• Disabled by default 
• You pay charges ($) for inter AZ data if enabled 

1. Elastic Load balancer SSL Certificate

Machine generated alternative text:
SSUTLS - Basics 
• An SSL Certificate allows traffic between your clients and your load balar 
to be encrypted in transit (in-flight encryption) 
• SSL refers to Secure Sockets Laye 
encrypt connections 
• TLS refers to Transport Layer S u whi 
IS a newer version 
• Nowadays, TLS certificates are ai 
d, ut people still refer as SSL 
• Public SSL certificates are issued by rtificate Authorities (CA) 
• Comodo, Symantec, GoDaddy, GlobalSign, Digicert, Letsencrypt, etc... 
• SSL certificates have an expiration date (you set) and must be renewed 

Machine generated alternative text:
Load Balancer - SSL Certificates 
LOAD BALANCER 
HTTPS (encrypted) 
Over wv•Æ' 
Users 
Over private VPC 
Instance 
• The load balancer uses an X.509 certificate (SSIJTLS server certificate) 
• You can manage certificates using ACM (AWS Certificate Manager) 
• You can create upload your own certificates alternatively 
• HIT PS listener: 
• You must specify a default certificate 
• You can add an optional list of certs to support multiple domains 
• Clients can use SNI (Server Name Indication) to specifr the hostname they reach 
• Ability to specify a security policy to support older versions of SSL / TLS (legacy clients) 

Machine generated alternative text:
SSL — Server Name Indication 
ALB 
Target group for 
wv.w.mycorp.com 
Target group for 
Domainl.example.com 
cert: 
DomainI.example.com 
ssL cert: 
www.mycorp.com 
• SNI solves the problem of loading multiple SSL 
certificates onto one web server (to serve 
multiple websites) 
• It's a "newer" protocol, and requires the client 
to indicate the hostname of the target server 
in the initial SSL handshake 
• The server will then find the correct 
certificate, or return the default one 
Note: 
• Only works for ALB & NLB (newer 
generation), CloudFront 
• Does not work for CLB (older gen) 
I like 
Client 
Use the correct 

Machine generated alternative text:
Elastic Load Balancers 
• Classic Load Balancer (VI) 
• Support only one SSL certificate 
SSL Certificates 
• Must use multiple CLB for multiple hostname with multiple SSL certificates 
• Application Load Balancer (v 
• Supports multiple listeners w" 
—SS certificates 
m 
• Uses Server Name Indication ( I tom it work 
• Network Load Balancer (v2) 
• Supports multiple listeners with multiple SSL certificates 
• Uses Server Name Indication (SNI) to make it work 

Machine generated alternative text:
ELB — Connection Draining 
Waiting for existing 
connections to complete 
ELB 
New connections 
Established to all other instances 
• Feature naming: 
• CLB: Connection Draining 
• Target Group: Deregistration Delay 
(for ALB & NLB) 
• Time to complete "in-flight requests" while 
the instance is de-registering or unhealthy 
• Stops sending new requests to the instance 
which is de-registering 
• Between I to 3600 seconds, default is 300 
seconds 
• Can be disabled (set value to O) 
• Set to a low value if your requests are short 
Users 
DRAINING 
EC2 
EC2 

1. Autoscaling groups(ASG) overview

Machine generated alternative text:
What's an Auto Scaling Group? 
• In real-life, the load on your websites and application can change 
• In the cloud, you can create and get rid of servers very quickly 
• The goal of an Auto Scaling 01.1 NAS is to: 
• Scale out (add EC2 instances) N$ån creased load 
• Scale in (remove EC2 instances) mat a decreased load 
• Ensure we have a minimum and a maximum number of machines running 
• Automatically Register new instances to a load balancer 

Machine generated alternative text:
Auto Scaling Group in AWS 
Maximum size 
Actual Size / Desired Capacity 
Scale Out as Needed 
Minimum size 
EC2 
Instance 
Instance 
Instance 
EC2 
Instance 
EC2 
Instance 
EC2 
Instance 
AUTO SCALING GROUP 

NOTE:

Load balancer and ASG really works hands to hands:

Means if asg add new instance then load balancer will automatically register to target group

**Important step to create autoscaling**

Machine generated alternative text:
ASGs have the following attributes 
• A launch configuration 
• AMI + Instance Type 
• EC2 User Data 
• EBS Volumes 
• Security Groups 
• SSH Key Pair 
• Min Size / Max Size / Initial Capacity 
• Network + Subnets Information 
• Load Balancer Information 
• Scaling Policies 

Machine generated alternative text:
Auto Scaling Alarms 
• It is possible to scale an ASG based on CloudWatch alarms 
• An Alarm monitors a metric (such as Average CPU) 
• Metrics are computed for the overall ASG instances 
• Based on the alarm: 
• We can create scale-out policies (increase the number of instances) 
• We can create scale-in policies (decrease the number of instances) 
Alarm 
EC2 
Instance 
Certified Developer O Stephane Maarek 
EC2 
Instance 
EC2 
Instance 
EC2 
Instance 
Trigger scaling 

Machine generated alternative text:
ASG Brain Dump 
• Scaling policies can be on CPU, Network... and can even be on custom metrics or 
based on a schedule (if you know your visitors patterns) 
• ASGs use Launch configurations or Launch Templates (newer) 
• To update an ASG, you must provide a new launch configuration / launch template 
• IAM roles attached to an ASG will get assigned to EC2 instances 
• ASG are free. You pay for the underlying resources being launched 
• Having instances under an ASG means that if they get terminated for whatever reason, 
the ASG will automatically create new ones as a replacement Extra safety! 
• ASG can terminate instances marked as unhealthy by an LB (and hence replace them) 

**NOTE: ASG create new instance automatically**

**EC2- user\_data**

#!/bin/bash

sudo su

yum update -y

yum install httpd -y

service httpd start

service httpd status

echo "hello world from $(hostname)" > /var/www/html/index.html

**Autoscaling group hands on**

**IMP using ALB we create instance and attach to load balancer**

**Minimum and maximum we can configure to create instance**

**Myfirstsecurity group**

Machine generated alternative text:
Inbound rules 
Inbound rules 
Type 
HTTp 
Outbound rules 
Protocol 
TCP 
TCP 
Tags 
Port range 
80 
22 
Source 
sg-0796d105f16a42bfd (my-first-load- 
balancer) 
0.0.0.0/0 
Edit inbound rules 
Description - optional 
allowed http traffic from load 
balancer 
ssh allowed 

**Myfirstloadbancer security group**

Machine generated alternative text:
Details 
Security group name 
my-first-load-balancer 
Owner 
398678778168 
Security group ID 
sg-0796d105f16a42bfd 
Inbound rules count 
1 Permission entry 
Description 
my first load balancer 
security group 
Outbound rules count 
1 Permission entry 
Source 
0.0.0.0/0 
VPC ID 
vpc-c14dafa9 
Edit inbound rules 
Description - optional 
Inbound rules 
Inbound rules 
Type 
HTTp 
Tags 
Port range 
80 
Outbound rules 
Protocol 
TCP 

**Linked load balancer security group to ec2 security group**

**Steps**

**Step 1)** First create application load balancer

**NOTE**: target group :--0 instance

<http://myfirstalb-776683954.eu-west-3.elb.amazonaws.com/>

503 (since we have don’t any instance attached)

**Creating asg**

**Step 2)** EC2: Autoscaling group ---default launch template

Machine generated alternative text:
Step 1 
Choose launch template or 
configuration 
Step 2 
Configure settings 
Step 3 (optional) 
Configure advanced options 
Step 4 (optional) 
Configure group size and 
scaling policies 
Step 5 (optional) 
Add notifications 
Step 6 (optional) 
Add tags 
Step 7 
Choose launch template or configuration 
Info 
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If 
you currently use launch configurations, you might consider migrating to launch templates. 
Name 
Auto Scaling group name 
Enter a name to identify the group. 
Must be unique to this account in the current Region and no more than 255 characters. 
Launch template Info 
Launch template 
Switch to launch configuration 
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and 
security groups. 
Select a launch template 
Create a launch template 

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

**IMPORTANT**

To configure Amazon EC2 instances that are launched by your Auto Scaling group, you can specify a **launch template**, a **launch** configuration, or an EC2 instance. ... Each **launch template** includes the information that Amazon EC2 needs to **launch** instances, such as an AMI and instance type.

**create Launch Template** : describe how to create EC2 instance

Step 3 :configure setting

Machine generated alternative text:
Step 1 
Choose launch template or 
configuration 
Step 2 
Configure settings 
Step 3 (optional) 
Configure advanced options 
Step 4 (optional) 
Configure group size and 
scaling policies 
Step 5 (optional) 
Add notifications 
Step 6 (optional) 
Add tags 
Step 7 
Review 
Configure settings 
I nfo 
Configure the settings below. Depending on whether you chose a launch template, these settings may include options to helF 
you make optimal use of EC2 resources. 
Purchase options and instance types Info 
O Adhere to launch template 
The launch template determines the purchase option (On- 
Demand or Spot) and instance type. 
Network 
Info 
O 
Combine purchase options and instance types 
Specify how much On-Demand and Spot capacity to 
launch and multiple instance types (optional). This choice 
is most helpful for optimizing the scale and cost for a fleet 
of instances. 
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the 
zones. The default VPC and default subnets are suitable for getting started quickly. 
VPC 
vpc-cl 4dafa9 

**Step 4) imp step : select load balancing**

* Enable load balancing and select alb and target group
* Enable health check :---- means if health check fails it will create instance automatically

Note : once instance create it will show to target group of ALB automatically

Health check type[**Info**](javascript:void(0);)

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

**STEP 5**

Machine generated alternative text:
Auto Scaling groups 
EC2 
Step 1 
Choose launch template or 
configuration 
Step 2 
Configure settings 
Step 3 (optional) 
Configure advanced options 
Step 4 (optional) 
Configure group size and 
scaling policies 
Step 5 (optional) 
Add notifications 
Step 6 (optional) 
Add tags 
Create Auto Scaling group 
Configure group size and scaling policies 
Info 
Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to 
dynamically scale the number of instances in the group. 
Group size - optional Info 
Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum 
capacity limits. Your desired capacity must be within the limit range. 
Desired capacity 
Minimum capacity 
Maximum capacity 

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

NOTE:

Instance will create using ASG and also instance will appear in **TARGET GROUP**

**For testing go to alb : and check dns name**

<http://myfirstalb-776683954.eu-west-3.elb.amazonaws.com/>

hello world from ip-172-31-23-205.eu-west-3.compute.internal

NOTE:

Go to asg and click edit under details

**Once we increase desired capacity it in will increase Instance**

**Auto scaling group --scaling policies**

Machine generated alternative text:
Auto Scaling Groups — Scaling Policies 
• Target Tracking Scaling 
• Most simple and easy to set-up 
• Example: I want the average ASG CPU to stay at around 40% 
• Simple / Step Scaling 
mple CPU 70%), then add 2 units 
• When a CloudWatch alarm is i 
• When a CloudWatch alarm is •ggerä( mple CPU 30%), then remove I 
• Scheduled Actions 
• Anticipate a scaling based on known usage patterns 
• Example: increase the min capacity to 10 at 5 pm on Fridays 

Machine generated alternative text:
Auto Scaling Groups - Scaling Cooldowns 
• The cooldown period helps to ensure that your Auto Scaling group doesnt 
launch or terminate additional instances before the previous scaling activity takes 
• In addition to default cooldown for Auto Scaling grou we can create cooldowns 
that apply to a specific simple scaling policy 
• A scaling-specific cooldown period overrides t 
• One common use for scaling-specific cool 
is 
coo 
period. 
policy—a policy 
that terminates instances based on a specific criteria —etric. Because this policy 
terminates instances, Amazon EC2 Auto Scalin 
ti to determine 
whether to terminate additional instances. 
• If the default cooldown period of 300 seconds is t 
can reduce costs 
by applying a scaling-specific coo Idown period of 180 seco to the scale-in 
policy. 
• If your application is scaling up and down multiple times each hour, modify the 
Auto Scafing Groups cool-down timers and the CloudWatch Alarm Period that 
triggers the scale in 

**Scaling policies hands on**

Step 1) based on cpu utilization

Machine generated alternative text:
New EC2 Experience 
x 
Tell us what you think 
EC2 Dashboard New 
Events New 
Tags 
Limits 
INSTANCES 
Instances 
Instance Types 
Launch Templates 
Spot Requests 
Savings Plans 
Reserved Instances 
Dedicated Hosts New 
Scheduled Instances 
Capacity Reservations 
EC2 
Auto Scaling groups ) MyFirstASG 
Activity 
Details 
Automatic scaling 
Info 
Instance refresh 
Add policy 
Create scheduled action 
Instance management 
Monitoring 
Actions 
(0) 
Scaling policies 
No scaling policies are currently specified 
Add policy 
Actions 
End time v 
Recurrence v 
(0) 
Scheduled actions 
Q 
Filter scheduled actions 
Min 
Name 
Start time 
Ma 
Desired capacity 
No scheduled actions are currently specified 

NOTE:

Initial desired capacity :-1

**suppose we gave desired capacity** :-2

Not set automatic scaling ----cpu utilization

We can check in cloudwatch

**NOTE : if cpu utilization more than 40% one instance will be terminate**

QUIZ

**1)Load Balancers provide a**

**:--**static DNS name we use in our application

Network Load Balancers expose a public static IP, whereas an Application or Classic Load Balancer exposes a static DNS (URL)

SNI (Server Name Indication) is a feature allowing you to expose multiple SSL certs if the client supports it.

A **target tracking scaling policy** assumes that it should scale out your Auto **Scaling** group when the specified metric is above the **target** value. You cannot use a **target tracking scaling policy** to scale out your Auto **Scaling** group when the specified metric is below the **target** value.