ASG and LB Session

1.Differences between ELB, ALB, and NLB. Where will you use which one?

“For http protocol require work i choose ALB and for TCP/UDP i choose NLB, ELB is now obselete.”

The Elastic Load Balancer (ELB) was released by AWS in the spring of 2009. An ELB is a software-based load balancer which can be set up and configured in front of a collection of AWS Elastic Compute (EC2) instances. The load balancer serves as a single entry point for consumers of the EC2 instances and distributes incoming traffic across all machines available to receive requests.

In addition to providing a single point of entry, the ELB also performs a vital role in improving the fault tolerance of the services which it fronts. The ELB regularly conducts a health check of all instances which have been registered with it, and only routes traffic to those machines which respond as active and healthy to the health check.

Diff between ALB and NLB:

The first difference is that the Application Load Balancer (as the name implies) works at the Application Layer (Layer 7 of the OSI model). The network load balancer works at layers 3 & 4 (network and transport layers). The network load balancer just forward requests whereas the application load balancer examines the contents of the HTTP request header to determine where to route the request. So, the application load balancer is performing content based routing.

The other difference between the two is important because network load balancing cannot assure availability of the *application.* This is because it bases its decisions solely on network and TCP-layer variables and has no awareness of the application at all. Generally a network load balancer will determine “availability” based on the ability of a server to respond to ICMP ping, or to correctly complete the three-way TCP handshake. An application load balancer goes much deeper, and is capable of determining availability based on not only a successful HTTP GET of a particular page but also the verification that the *content* is as was expected based on the input parameters.

This is also important to note when considering the deployment of multiple applications on the same host sharing IP addresses (virtual hosts in old school speak). A network load balancer will not differentiate between Application A and Application B when checking availability (indeed it cannot unless ports are different) but an application load balancer will differentiate between the two applications by examining the application layer data available to it. This difference means that a network load balancer may end up sending requests to an application that has crashed or is offline, but an application load balancer will never make that same mistake.

2.Differences between step scaling and target scaling.

(**SIMPLE SCALING**)

You pick ANY Cloud Watch metric

For this and other examples in THIS POST I am choosing CPU Utilization

You specify, a SINGLE THRESHOLD beyond which you want to scale and specify your response

EXAMPLE: how many EC2 instances do you want to add or take away when the CPU UTILIZATION breaches the threshold.

The scaling policy then acts.

THRESHOLD - add 1 instance when CPU Utilization is between 40% and 50%

NOTE: **This is the ONLY Threshold**

(**STEP SCALING**)

You specify MULTIPLE thresholds Along with different responses.

Threshold A - add 1 instance when CPU Utilization is between 40% and 50%

Threshold B - add 2 instances when CPU Utilization is between 50% and 70%

Threshold C - add 3 instances when CPU Utilization is between 70% and 90%

And so on and so on

Note: **There are multiple thresholds**

3.Differences between Launch configuration and launch template.

A launch template is similar to a launch configuration, in that it specifies instance configuration information. Included are the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that you use to launch EC2 instances. However, defining a launch template instead of a launch configuration allows you to have multiple versions of a template. With versioning, you can create a subset of the full set of parameters and then reuse it to create other templates or template versions. For example, you can create a default template that defines common configuration parameters such as tags or network configurations, and allow the other parameters to be specified as part of another version of the same template.

4.Differences between EC2 healthcheck and load balancer health check

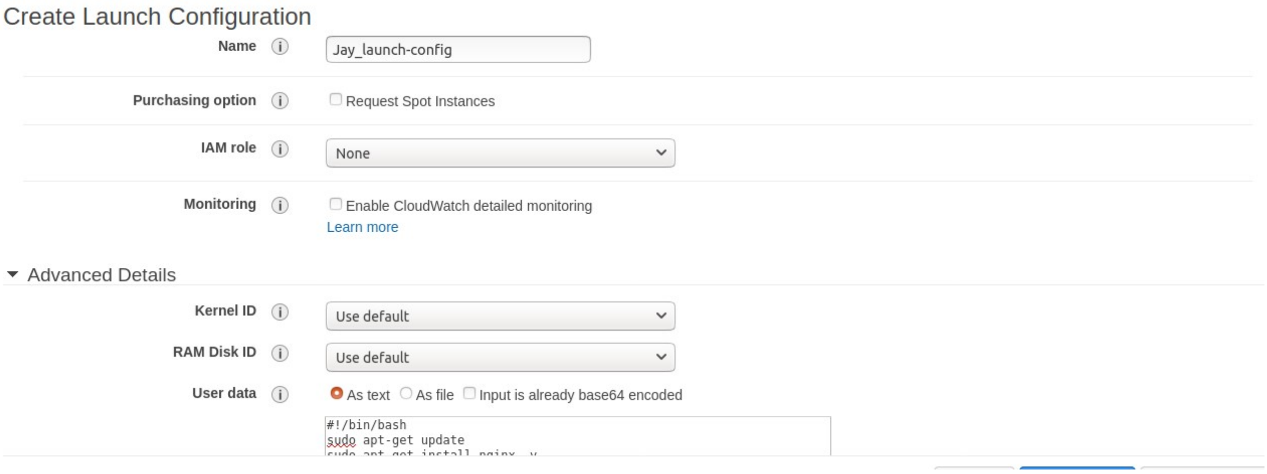
EC2 health check watches for instance availability from hypervisor and networking point of view. For example, in case of a hardware problem, the check will fail. Also, if an instance was misconfigured and doesn't respond to network requests, it will be marked as faulty.

ELB health check verifies that a specified TCP port on an instance is accepting connections OR a specified web page returns 2xx code. Thus ELB health checks are a little bit smarter and verify that actual app works instead of verifying that just an instance works.

That being said there is a third check type: custom health check. If your application can't be checked by simple HTTP request and requires advanced test logic, you can implement a custom check in your code and set instance health though API

5.Create 2 auto-scaling groups with

- launch configuration and



Create Launch Configuration

Name: Jay_launch-config

Purchasing option: □ Request Spot Instances

IAM role: None

Monitoring: □ Enable CloudWatch detailed monitoring

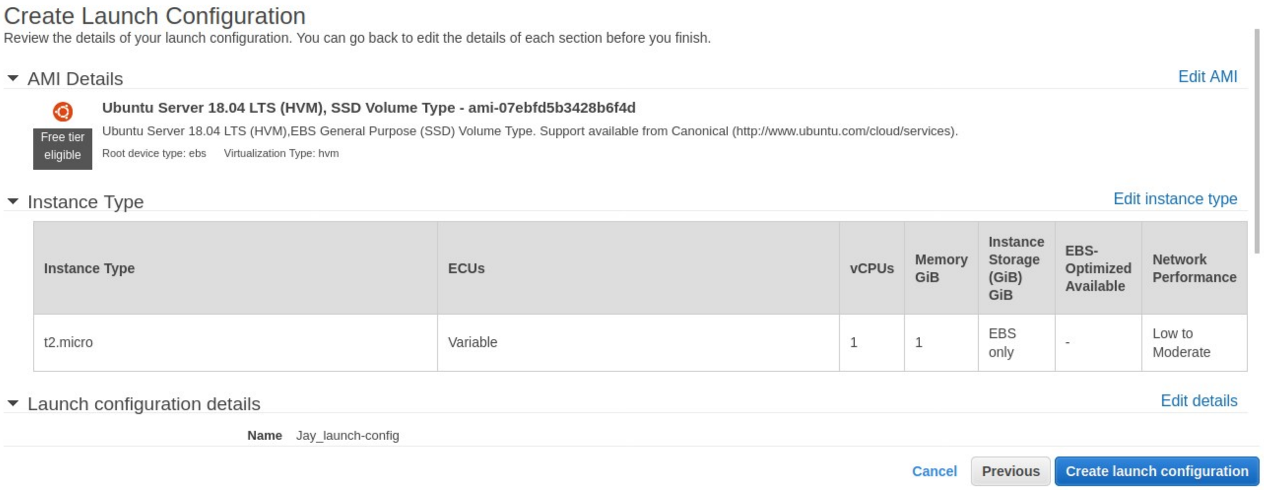
▾ Advanced Details

Kernel ID: Use default

RAM Disk ID: Use default

User data: ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/bash
sudo apt-get update
sudo apt-get install nginx -y
```



Create Launch Configuration

Review the details of your launch configuration. You can go back to edit the details of each section before you finish.

▾ AMI Details — Edit AMI

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-07ebfd5b3428b6f4d

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

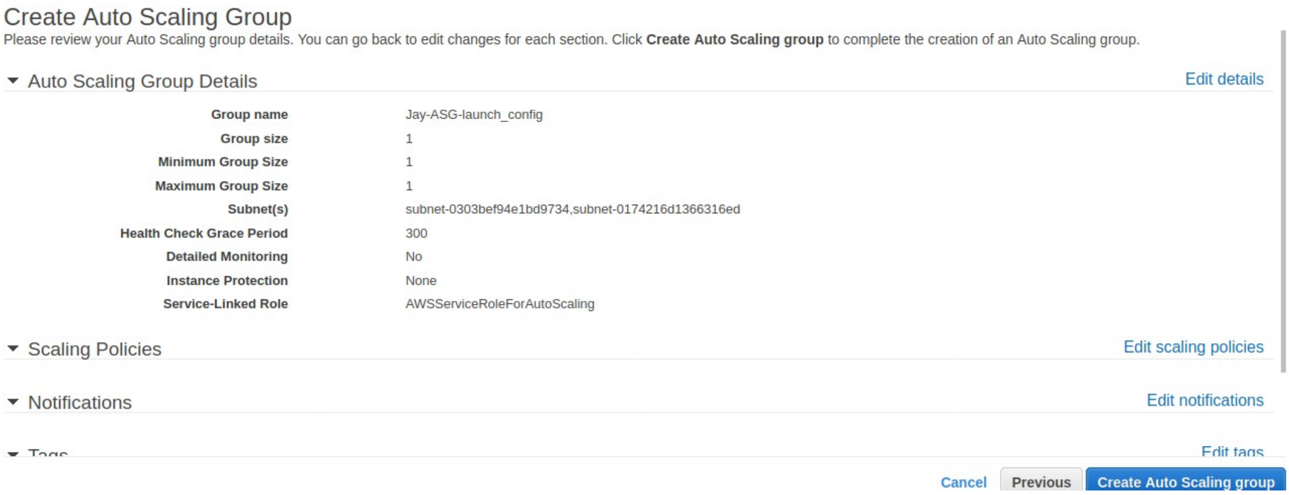
Root device type: ebs Virtualization Type: hvm

▾ Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory GiB	Instance Storage (GiB) GiB	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▾ Launch configuration details — Edit details

Name Jay_launch-config



Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

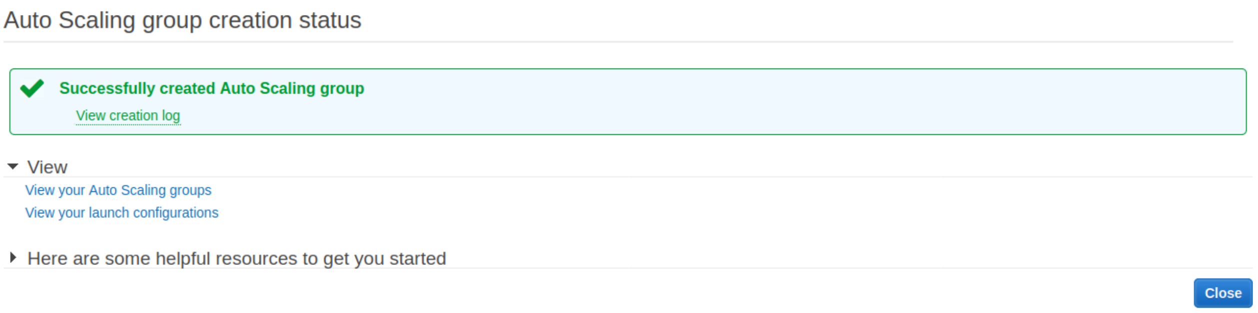
▾ Auto Scaling Group Details — Edit details

Group name	Jay-ASG-launch_config
Group size	1
Minimum Group Size	1
Maximum Group Size	1
Subnet(s)	subnet-0303bef94e1bd9734,subnet-0174216d1366316ed
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	None
Service-Linked Role	AWSServiceRoleForAutoScaling

▾ Scaling Policies — Edit scaling policies

▾ Notifications — Edit notifications

▾ Tags — Edit tags



Auto Scaling group creation status

✓ **Successfully created Auto Scaling group**

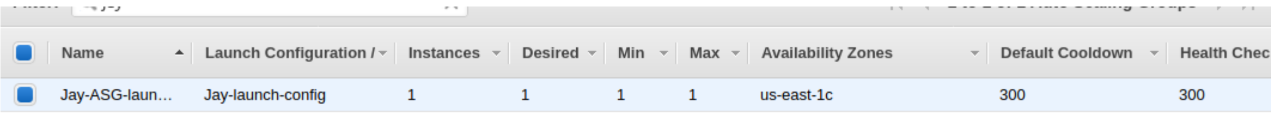
View creation log

▾ View

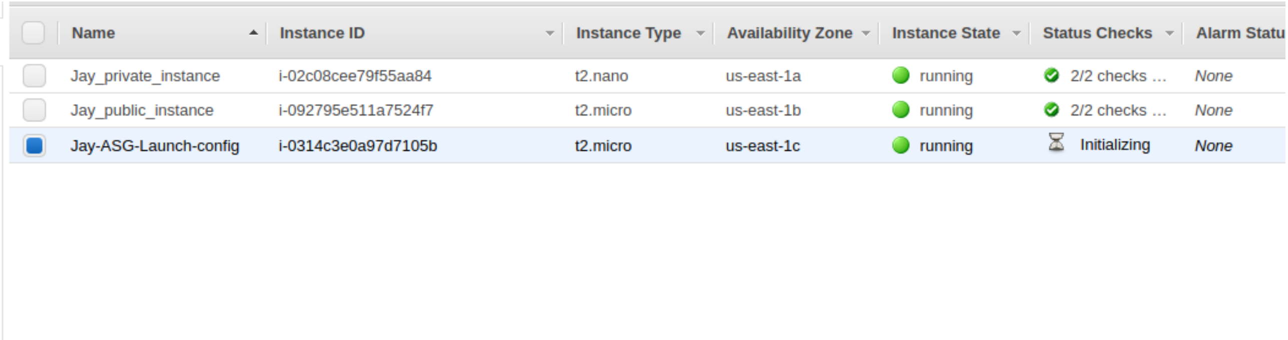
View your Auto Scaling groups

View your launch configurations

▸ Here are some helpful resources to get you started

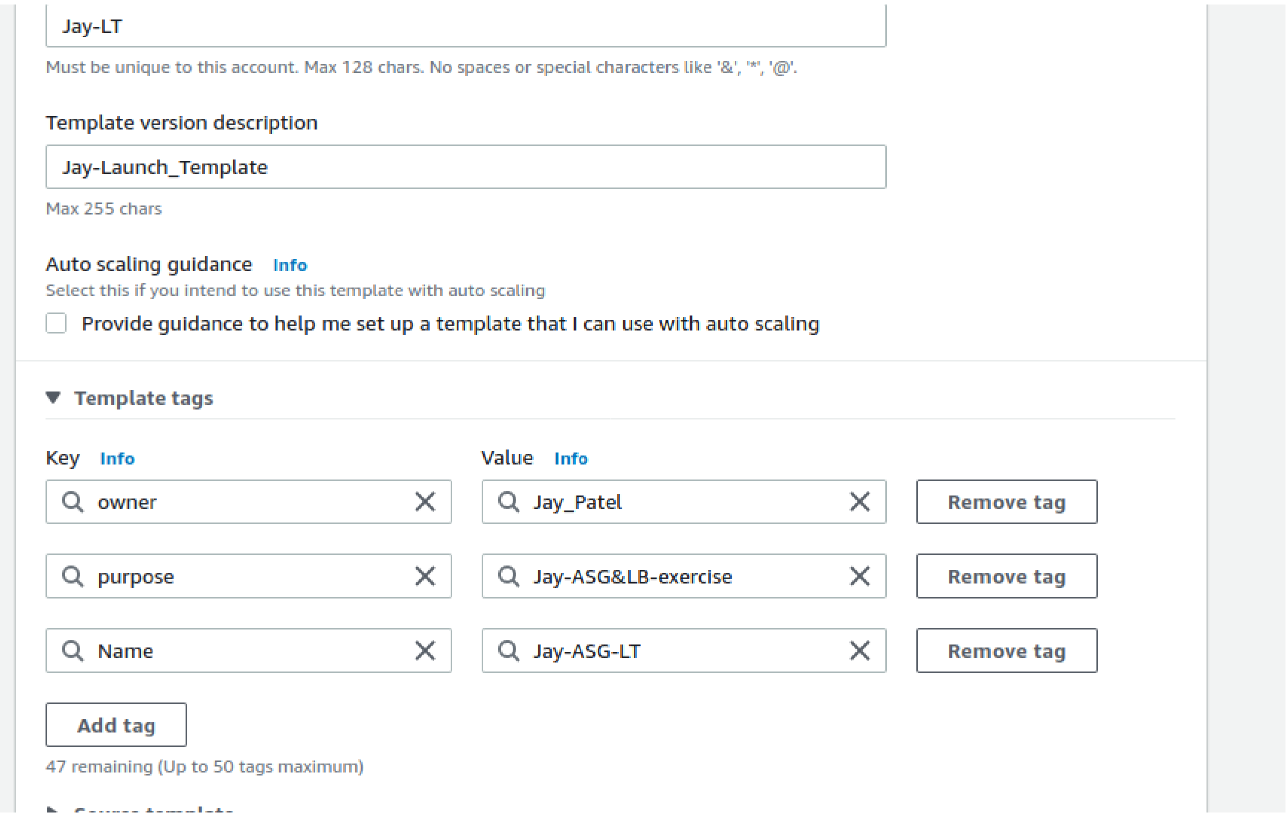


■	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Chec
■	Jay-ASG-laun…	Jay-launch-config	1	1	1	1	us-east-1c	300	300



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Statu
□	Jay_private_instance	i-02c08cee79f55aa84	t2.nano	us-east-1a	● running	✓ 2/2 checks ...	None
□	Jay_public_instance	i-092795e511a7524f7	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	None
■	Jay-ASG-Launch-config	i-0314c3e0a97d7105b	t2.micro	us-east-1c	● running	⌛ Initializing	None

- launch template



Jay-LT

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Jay-Launch_Template

Max 255 chars

Auto scaling guidance Info

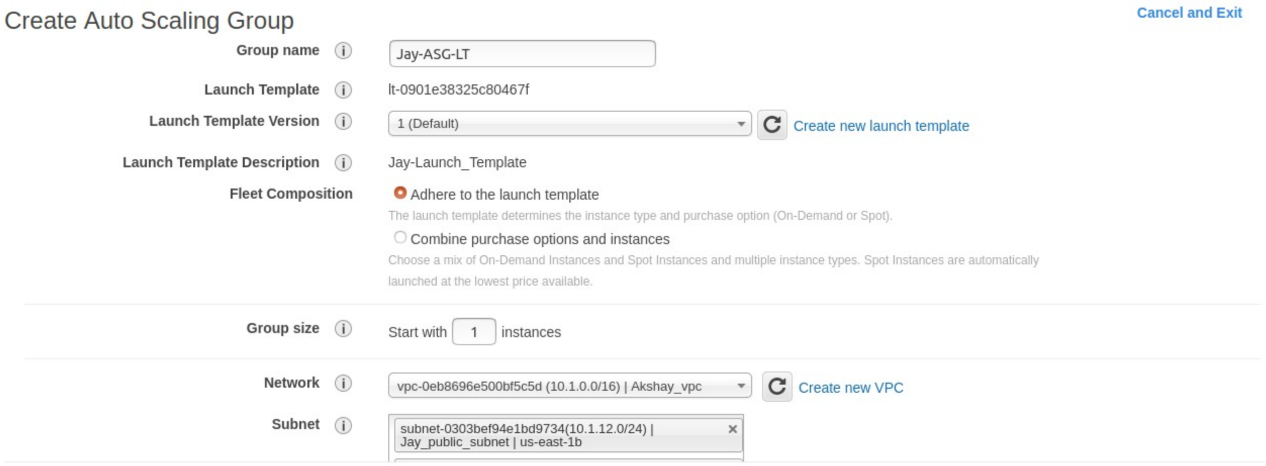
Select this if you intend to use this template with auto scaling

□ Provide guidance to help me set up a template that I can use with auto scaling

▼ Template tags

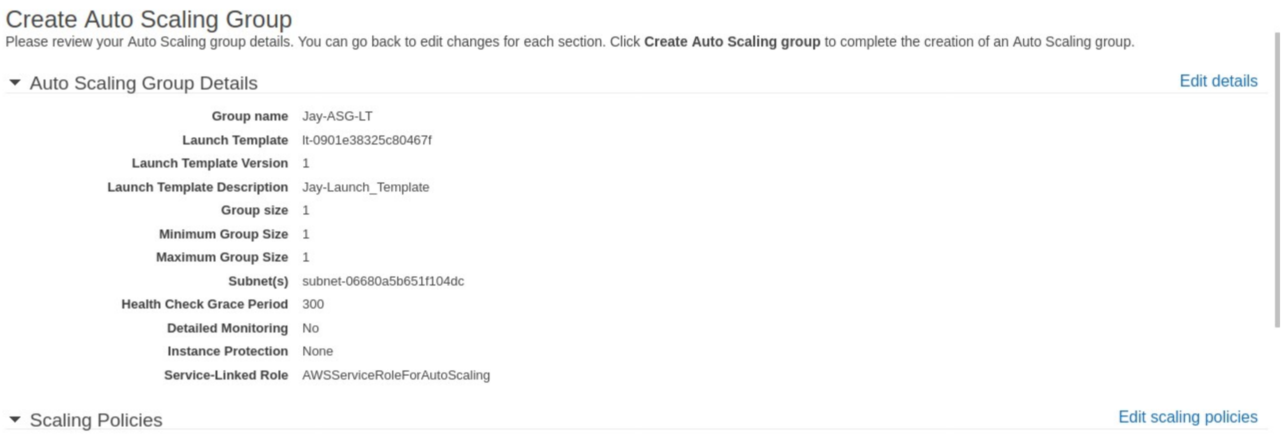
Key	Value	
owner	Jay_Patel	Remove tag
purpose	Jay-ASG&LB-exercise	Remove tag
Name	Jay-ASG-LT	Remove tag

47 remaining (Up to 50 tags maximum)



Create Auto Scaling Group

Group name	Jay-ASG-LT
Launch Template	lt-0901e38325c80467f
Launch Template Version	1 (Default)
Launch Template Description	Jay-Launch_Template
Fleet Composition	● Adhere to the launch template The launch template determines the instance type and purchase option (On-Demand or Spot). ○ Combine purchase options and instances Choose a mix of On-Demand Instances and Spot Instances and multiple instance types. Spot Instances are automatically launched at the lowest price available.
Group size	Start with 1 instances
Network	vpc-0eb8696e500bf5c5d (10.1.0.0/16) | Akshay_vpc
Subnet	subnet-0303bef94e1bd9734(10.1.12.0/24) | Jay_public_subnet | us-east-1b



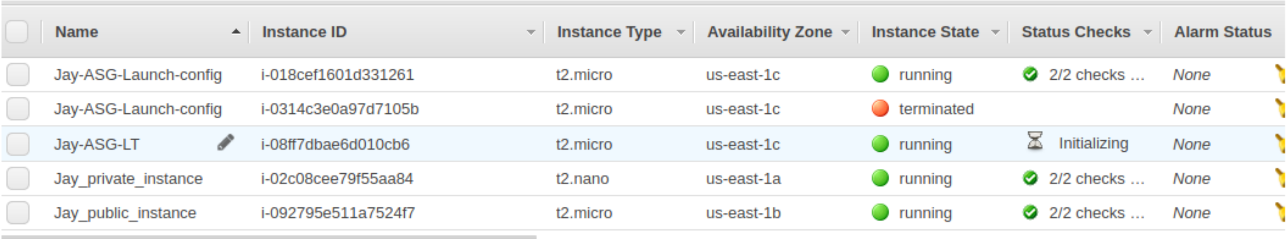
Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

Auto Scaling Group Details

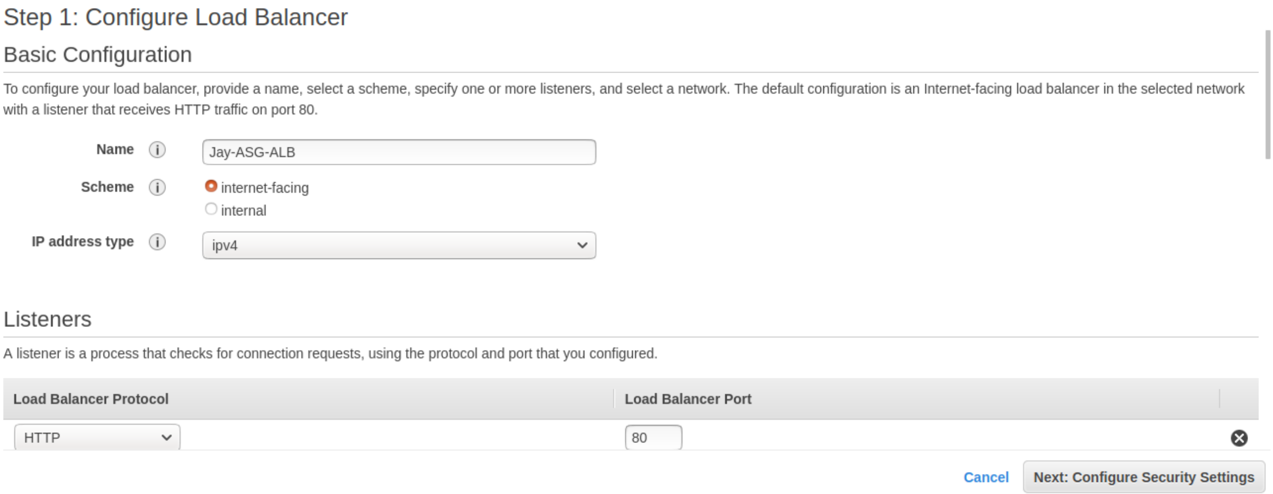
Group name	Jay-ASG-LT
Launch Template	lt-0901e38325c80467f
Launch Template Version	1
Launch Template Description	Jay-Launch_Template
Group size	1
Minimum Group Size	1
Maximum Group Size	1
Subnet(s)	subnet-06680a5b651f104dc
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	None
Service-Linked Role	AWSServiceRoleForAutoScaling

Scaling Policies



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Jay-ASG-Launch-config	i-018cef1601d331261	t2.micro	us-east-1c	● running	✓ 2/2 checks ...	None
Jay-ASG-Launch-config	i-0314c3e0a97d7105b	t2.micro	us-east-1c	● terminated		None
Jay-ASG-LT	i-08ff7dbae6d010cb6	t2.micro	us-east-1c	● running	⌛ Initializing	None
Jay_private_instance	i-02c08cee79f55aa84	t2.nano	us-east-1a	● running	✓ 2/2 checks ...	None
Jay_public_instance	i-092795e511a7524f7	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	None

6.Setup autoscaling Wordpress application with the Application load balancer. Auto-scaling should be triggered based on CPU usage of EC2 instances.



Step 1: Configure Load Balancer

Basic Configuration

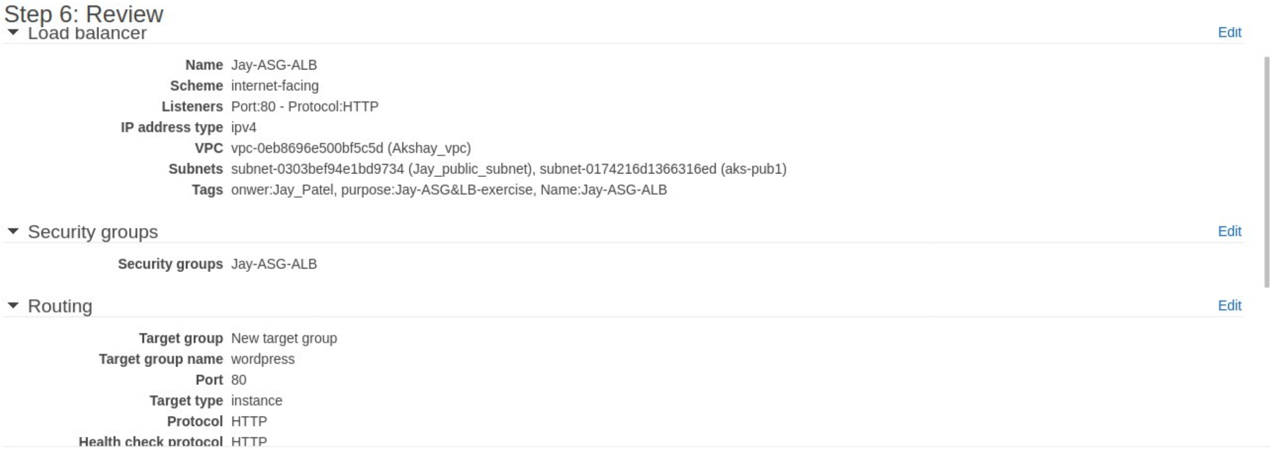
To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name	Jay-ASG-ALB
Scheme	● internet-facing ○ internal
IP address type	ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80



Step 6: Review

Load balancer

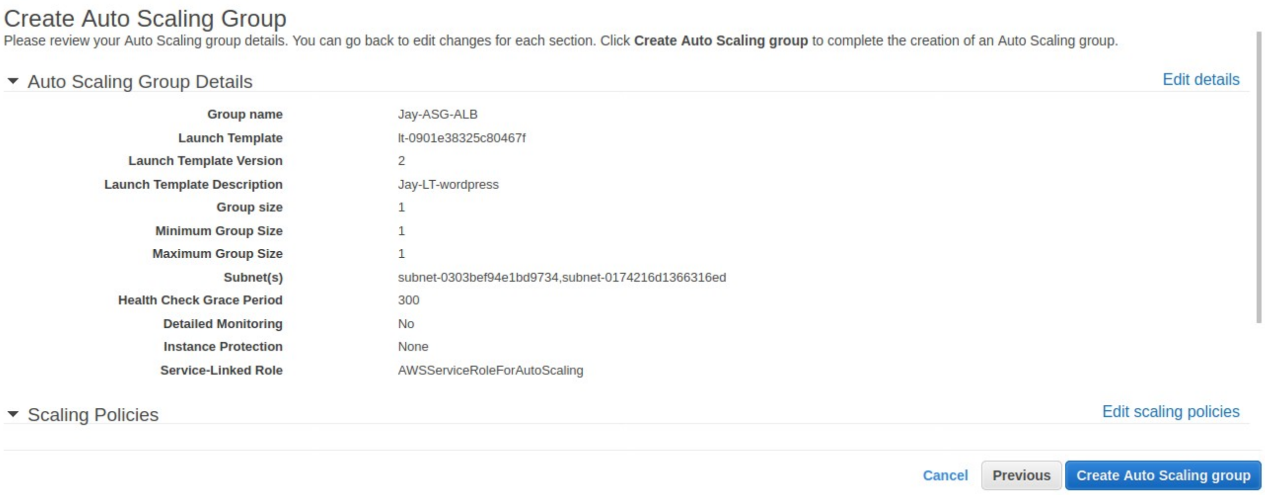
Name	Jay-ASG-ALB
Scheme	internet-facing
Listeners	Port:80 - Protocol:HTTP
IP address type	ipv4
VPC	vpc-0eb8696e500bf5c5d (Akshay_vpc)
Subnets	subnet-0303bef94e1bd9734 (Jay_public_subnet), subnet-0174216d1366316ed (aks-pub1)
Tags	onwer:Jay_Patel, purpose:Jay-ASG&LB-exercise, Name:Jay-ASG-ALB

Security groups

Security groups	Jay-ASG-ALB

Routing

Target group	New target group
Target group name	wordpress
Port	80
Target type	instance
Protocol	HTTP
Health check protocol	HTTP



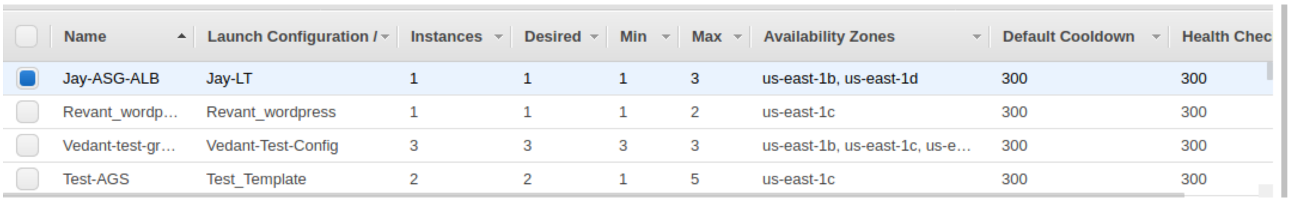
Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

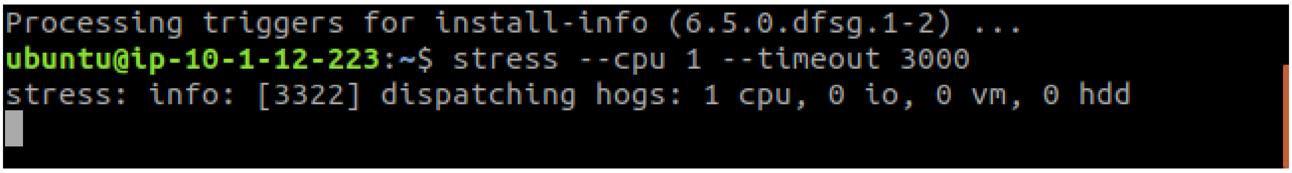
Auto Scaling Group Details

Group name	Jay-ASG-ALB
Launch Template	lt-0901e38325c80467f
Launch Template Version	2
Launch Template Description	Jay-LT-wordpress
Group size	1
Minimum Group Size	1
Maximum Group Size	1
Subnet(s)	subnet-0303bef94e1bd9734,subnet-0174216d1366316ed
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	None
Service-Linked Role	AWSServiceRoleForAutoScaling

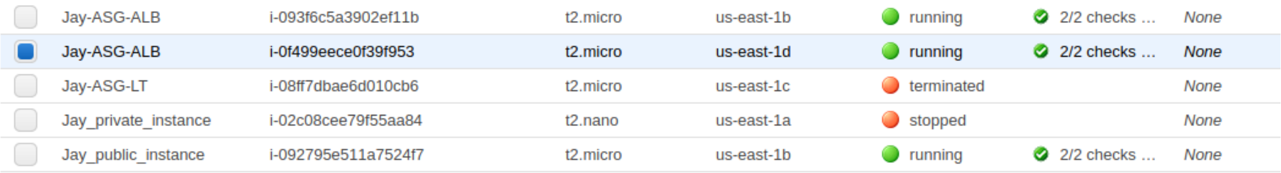
Scaling Policies



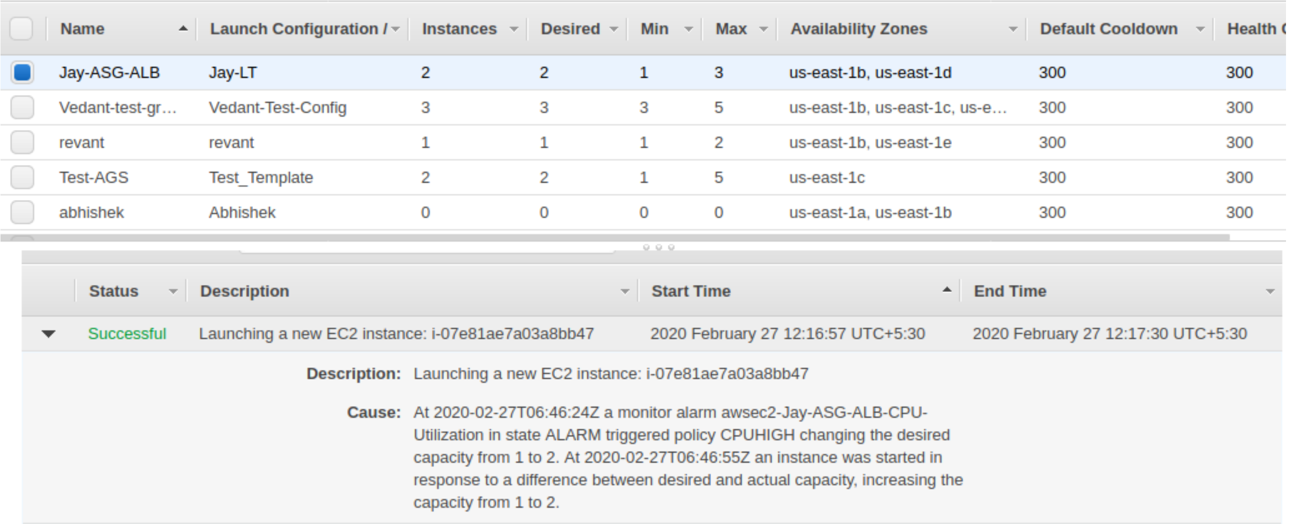
□	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Chec
☑	Jay-ASG-ALB	Jay-LT	1	1	1	3	us-east-1b, us-east-1d	300	300
□	Revant_wordp…	Revant_wordpress	1	1	1	2	us-east-1c	300	300
□	Vedant-test-gr…	Vedant-Test-Config	3	3	3	3	us-east-1b, us-east-1c, us-e…	300	300
□	Test-AGS	Test_Template	2	2	1	5	us-east-1c	300	300



```
Processing triggers for install-info (6.5.0.dfsg.1-2) ...
ubuntu@ip-10-1-12-223:~$ stress --cpu 1 --timeout 3000
stress: info: [3322] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```



☐	Jay-ASG-ALB	i-093f6c5a3902ef11b	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	*None*
☑	Jay-ASG-ALB	i-0f499eece0f39f953	t2.micro	us-east-1d	● running	✓ 2/2 checks ...	*None*
☐	Jay-ASG-LT	i-08ff7dbae6d010cb6	t2.micro	us-east-1c	● terminated		*None*
☐	Jay_private_instance	i-02c08cee79f55aa84	t2.nano	us-east-1a	● stopped		*None*
☐	Jay_public_instance	i-092795e511a7524f7	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	*None*

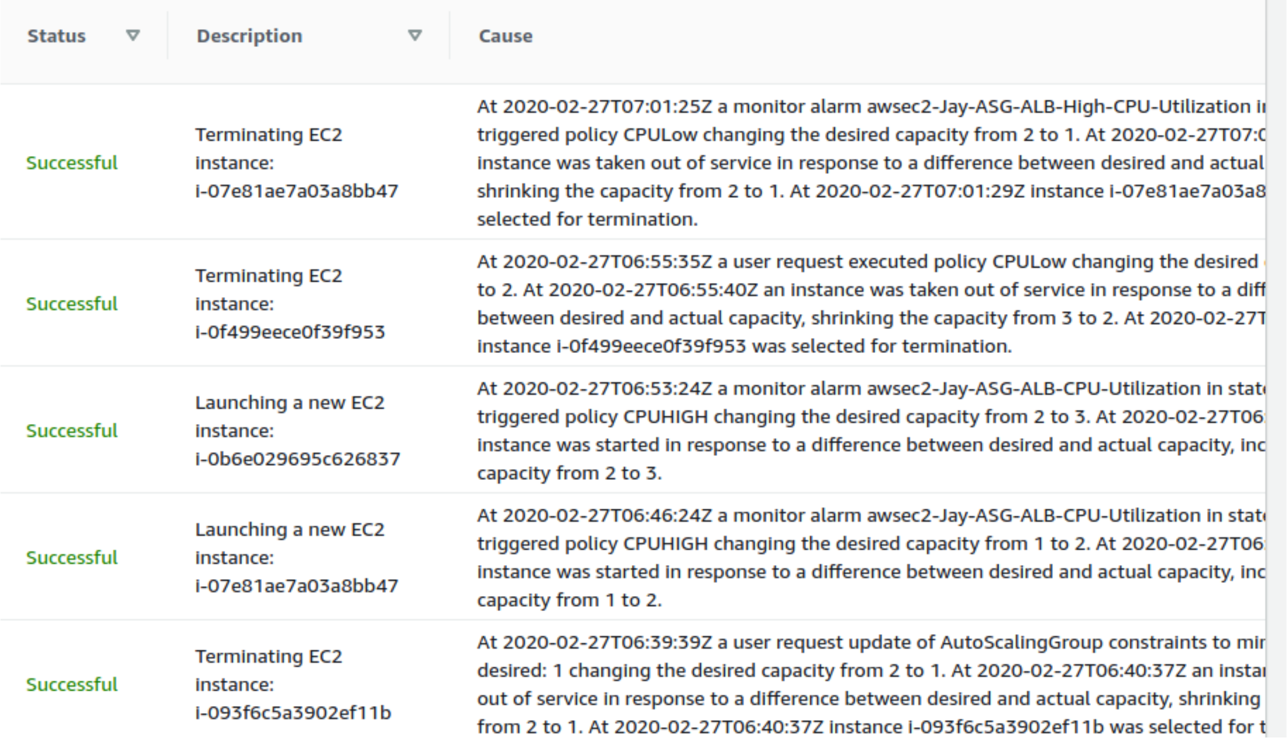


☐	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health (
☑	Jay-ASG-ALB	Jay-LT	2	2	1	3	us-east-1b, us-east-1d	300	300
☐	Vedant-test-gr...	Vedant-Test-Config	3	3	3	5	us-east-1b, us-east-1c, us-e...	300	300
☐	revant	revant	1	1	1	2	us-east-1b, us-east-1e	300	300
☐	Test-AGS	Test_Template	2	2	1	5	us-east-1c	300	300
☐	abhishek	Abhishek	0	0	0	0	us-east-1a, us-east-1b	300	300

	Status	Description	Start Time	End Time
▼	Successful	Launching a new EC2 instance: i-07e81ae7a03a8bb47	2020 February 27 12:16:57 UTC+5:30	2020 February 27 12:17:30 UTC+5:30

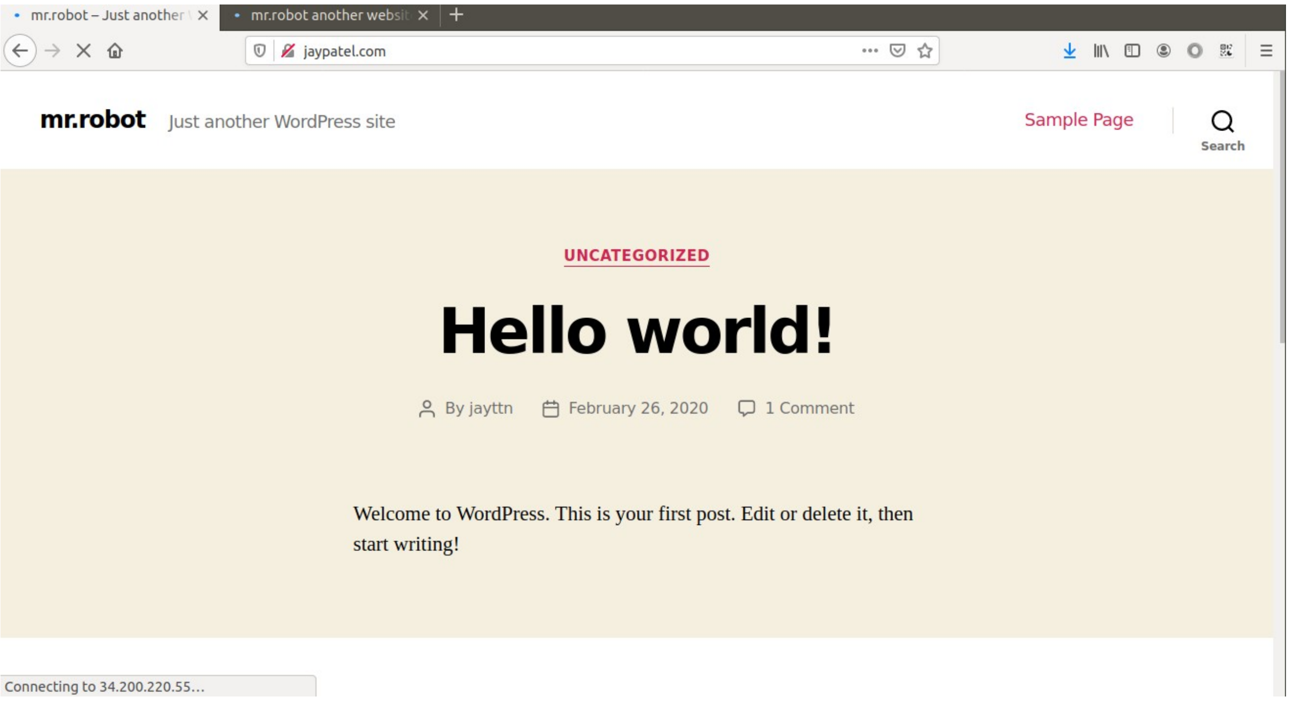
Description: Launching a new EC2 instance: i-07e81ae7a03a8bb47

Cause: At 2020-02-27T06:46:24Z a monitor alarm awsec2-Jay-ASG-ALB-CPU-Utilization in state ALARM triggered policy CPUHIGH changing the desired capacity from 1 to 2. At 2020-02-27T06:46:55Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.



Status	Description	Cause
Successful	Terminating EC2 instance: i-07e81ae7a03a8bb47	At 2020-02-27T07:01:25Z a monitor alarm awsec2-Jay-ASG-ALB-High-CPU-Utilization i triggered policy CPULow changing the desired capacity from 2 to 1. At 2020-02-27T07:0 instance was taken out of service in response to a difference between desired and actual shrinking the capacity from 2 to 1. At 2020-02-27T07:01:29Z instance i-07e81ae7a03a8 selected for termination.
Successful	Terminating EC2 instance: i-0f499eece0f39f953	At 2020-02-27T06:55:35Z a user request executed policy CPULow changing the desired to 2. At 2020-02-27T06:55:40Z an instance was taken out of service in response to a diff between desired and actual capacity, shrinking the capacity from 3 to 2. At 2020-02-27T instance i-0f499eece0f39f953 was selected for termination.
Successful	Launching a new EC2 instance: i-0b6e029695c626837	At 2020-02-27T06:53:24Z a monitor alarm awsec2-Jay-ASG-ALB-CPU-Utilization in stat triggered policy CPUHIGH changing the desired capacity from 2 to 3. At 2020-02-27T06 instance was started in response to a difference between desired and actual capacity, inc capacity from 2 to 3.
Successful	Launching a new EC2 instance: i-07e81ae7a03a8bb47	At 2020-02-27T06:46:24Z a monitor alarm awsec2-Jay-ASG-ALB-CPU-Utilization in stat triggered policy CPUHIGH changing the desired capacity from 1 to 2. At 2020-02-27T06 instance was started in response to a difference between desired and actual capacity, inc capacity from 1 to 2.
Successful	Terminating EC2 instance: i-093f6c5a3902ef11b	At 2020-02-27T06:39:39Z a user request update of AutoScalingGroup constraints to mi desired: 1 changing the desired capacity from 2 to 1. At 2020-02-27T06:40:37Z an insta out of service in response to a difference between desired and actual capacity, shrinking from 2 to 1. At 2020-02-27T06:40:37Z instance i-093f6c5a3902ef11b was selected for t

7.Create another Wordpress website and use the ALB created above to send traffic to this website based on the hostname



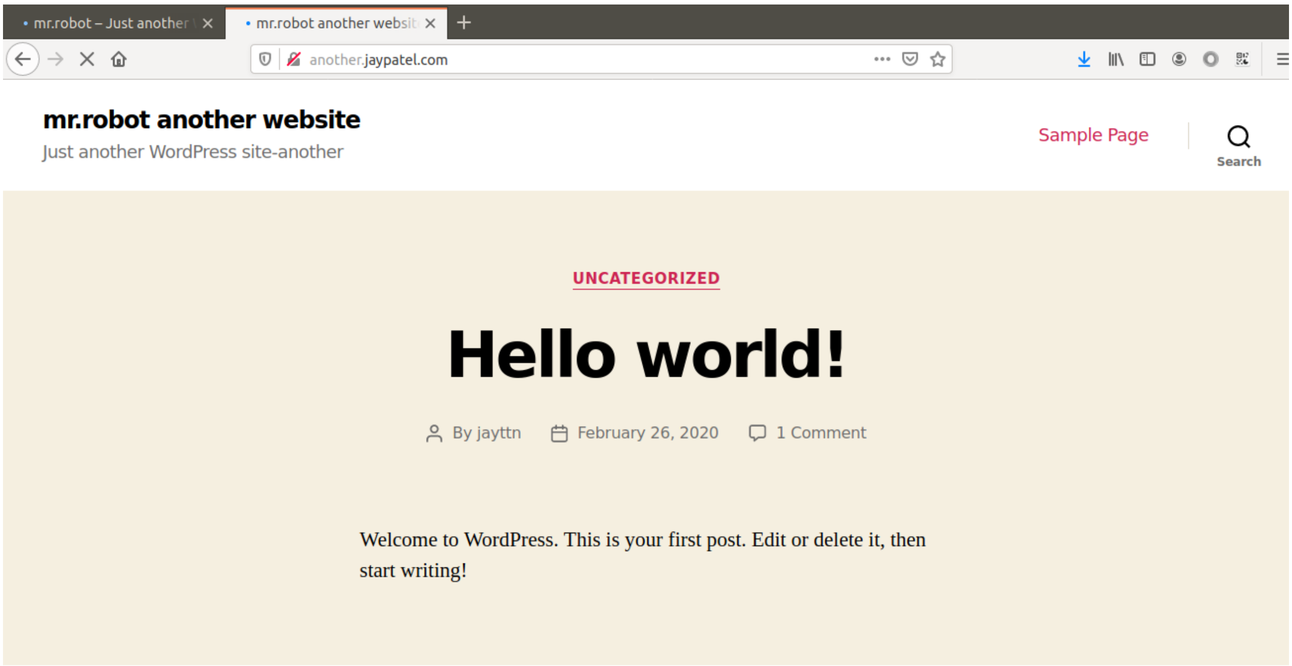
mr.robot Just another WordPress site

UNCATEGORIZED

Hello world!

By jayttn February 26, 2020 1 Comment

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!



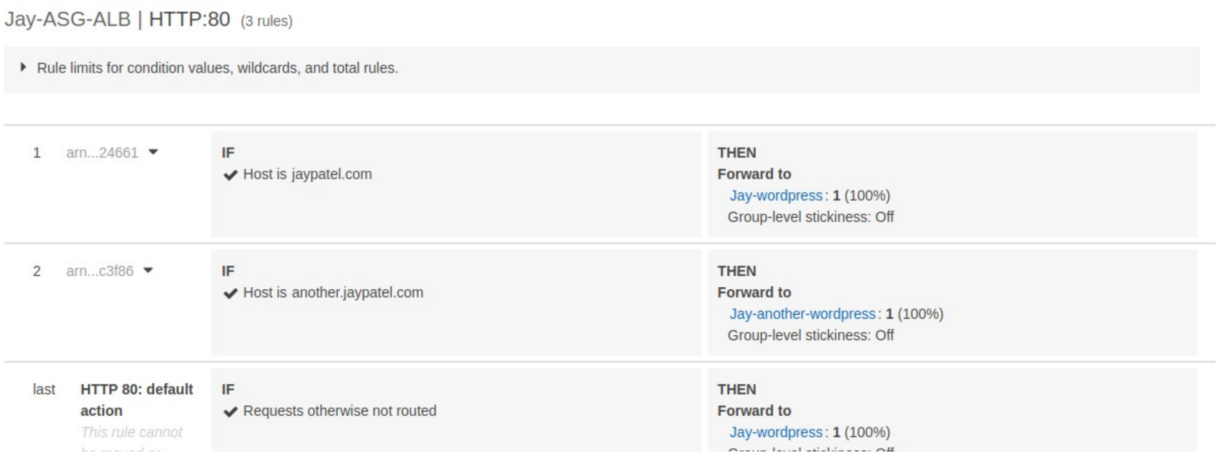
mr.robot another website
Just another WordPress site-another

UNCATEGORIZED

Hello world!

By jayttn February 26, 2020 1 Comment

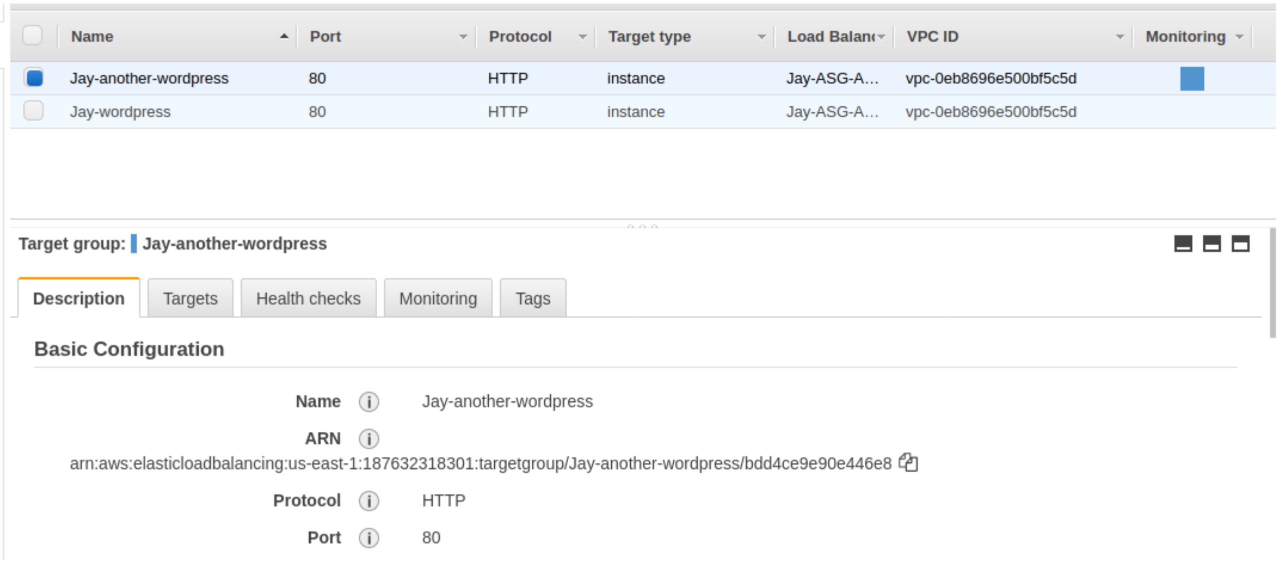
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!



Jay-ASG-ALB | HTTP:80 (3 rules)

▸ Rule limits for condition values, wildcards, and total rules.

		IF	THEN
1	arn...24661 ▾	✓ Host is jaypatel.com	Forward to Jay-wordpress: 1 (100%) Group-level stickiness: Off
2	arn...c3f86 ▾	✓ Host is another.jaypatel.com	Forward to Jay-another-wordpress: 1 (100%) Group-level stickiness: Off
last	HTTP 80: default action This rule cannot	✓ Requests otherwise not routed	Forward to Jay-wordpress: 1 (100%)



□	Name	Port	Protocol	Target type	Load Balan	VPC ID	Monitoring
■	Jay-another-wordpress	80	HTTP	instance	Jay-ASG-A...	vpc-0eb8696e500bf5c5d	■
□	Jay-wordpress	80	HTTP	instance	Jay-ASG-A...	vpc-0eb8696e500bf5c5d	

Target group: Jay-another-wordpress

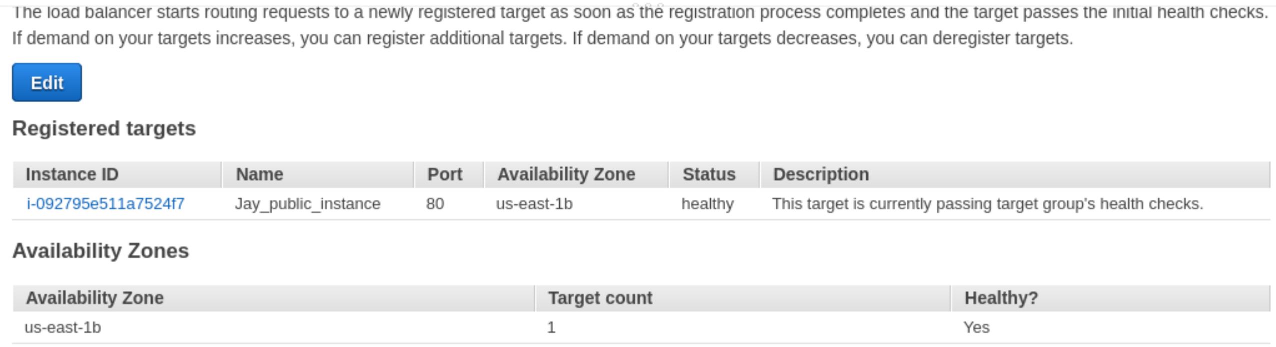
Basic Configuration

Name: Jay-another-wordpress

ARN: arn:aws:elasticloadbalancing:us-east-1:187632318301:targetgroup/Jay-another-wordpress/bdd4ce9e90e446e8

Protocol: HTTP

Port: 80



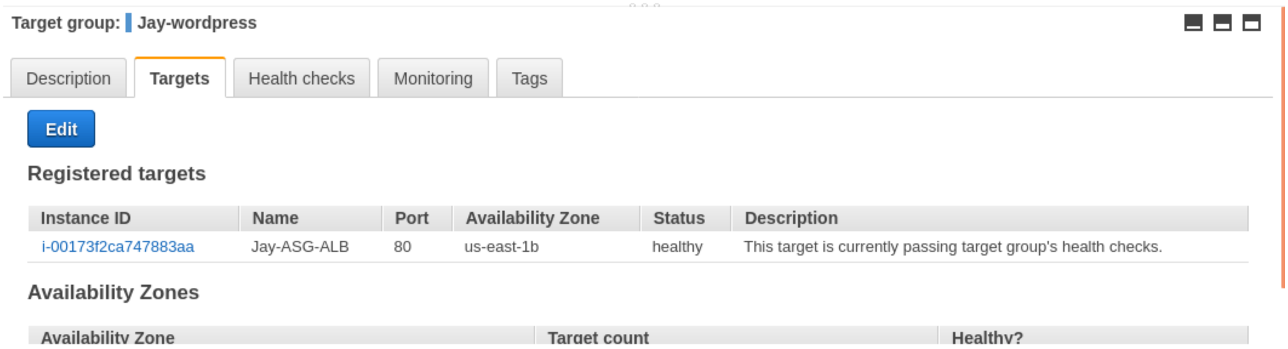
The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Registered targets

Instance ID	Name	Port	Availability Zone	Status	Description
i-092795e511a7524f7	Jay_public_instance	80	us-east-1b	healthy	This target is currently passing target group's health checks.

Availability Zones

Availability Zone	Target count	Healthy?
us-east-1b	1	Yes



Target group: Jay-wordpress

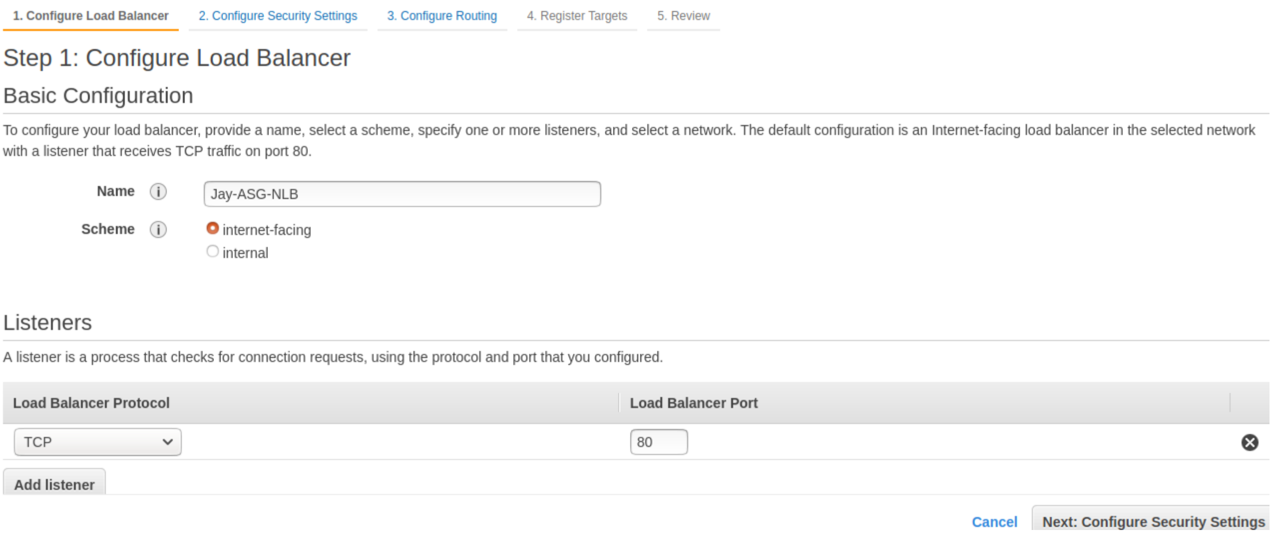
Registered targets

Instance ID	Name	Port	Availability Zone	Status	Description
i-00173f2ca747883aa	Jay-ASG-ALB	80	us-east-1b	healthy	This target is currently passing target group's health checks.

Availability Zones

Availability Zone	Target count	Healthy?

8.Use NLB that replaces the ALB in the above setup.



Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives TCP traffic on port 80.

Name: Jay-ASG-NLB

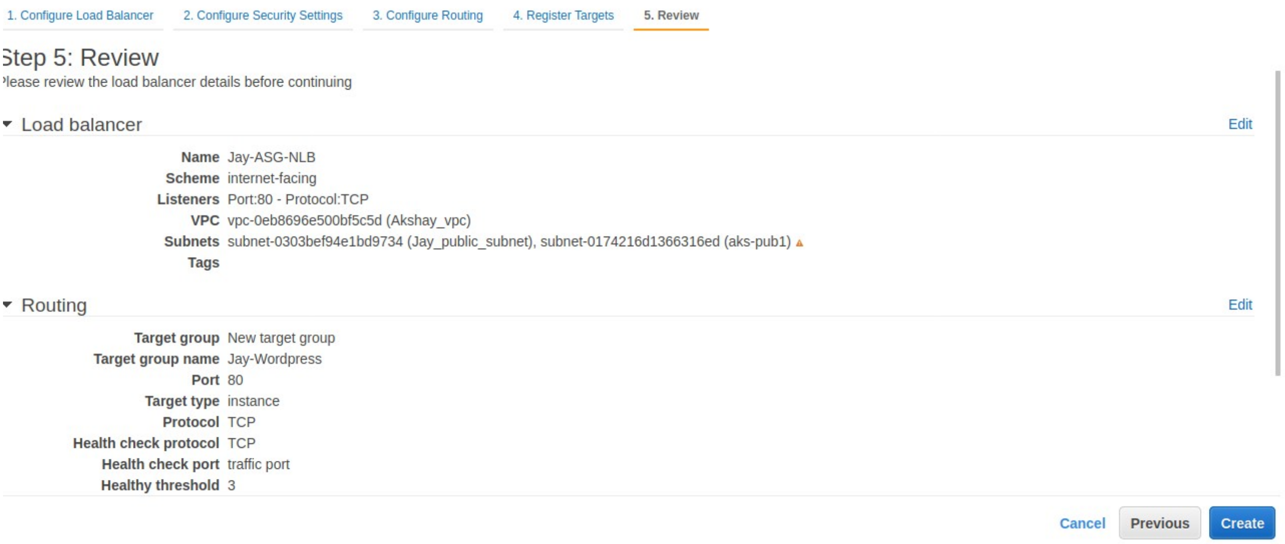
Scheme: ● internet-facing ○ internal

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
TCP	80

Add listener
Cancel
Next: Configure Security Settings


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Routing
4. Register Targets
5. Review

Step 5: Review

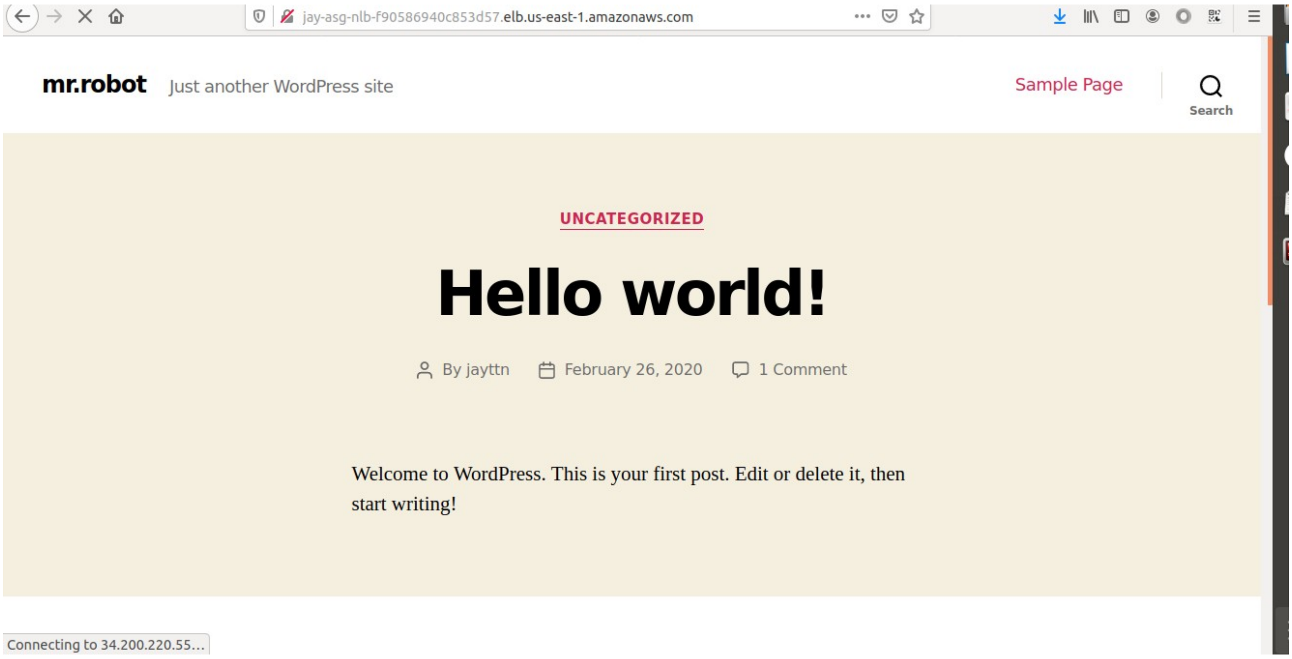
Please review the load balancer details before continuing

Load balancer

Name	Jay-ASG-NLB
Scheme	internet-facing
Listeners	Port:80 - Protocol:TCP
VPC	vpc-0eb8696e500bf5c5d (Akshay_vpc)
Subnets	subnet-0303bef94e1bd9734 (Jay_public_subnet), subnet-0174216d1366316ed (aks-pub1) ▲
Tags	

Routing

Target group	New target group
Target group name	Jay-Wordpress
Port	80
Target type	instance
Protocol	TCP
Health check protocol	TCP
Health check port	traffic port
Healthy threshold	3


jay-asg-nlb-f90586940c853d57.elb.us-east-1.amazonaws.com
mr.robot
Just another WordPress site
Sample Page
Search

UNCATEGORIZED

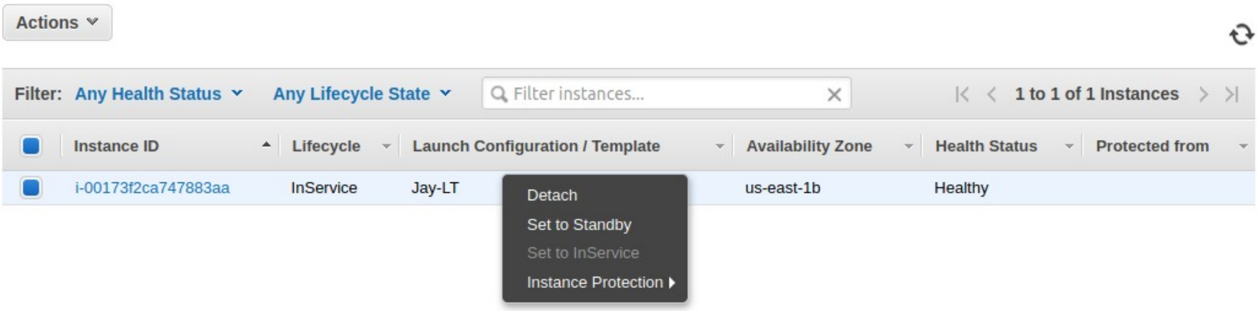
Hello world!

By jayttn February 26, 2020 1 Comment

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Connecting to 34.200.220.55…

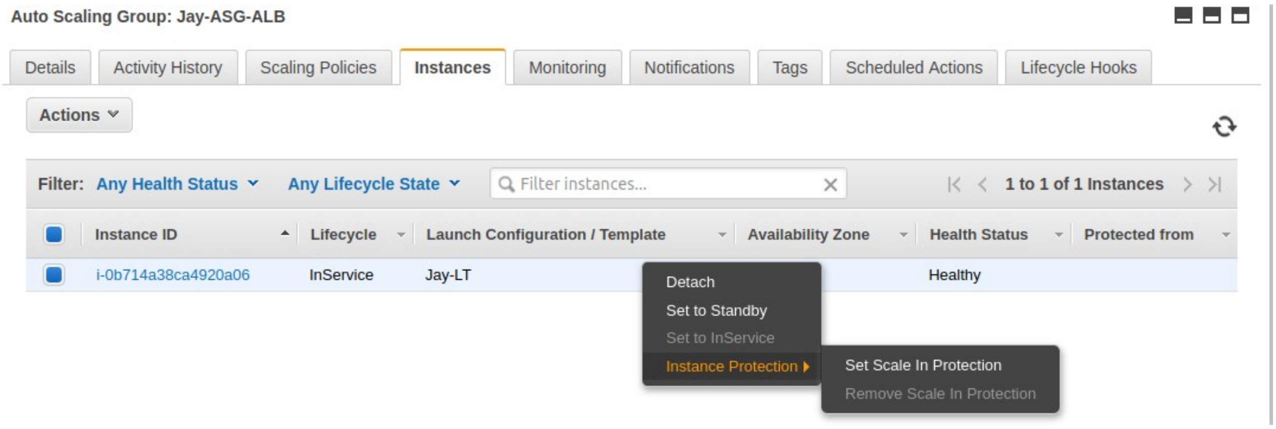
9.Take an instance out of the ASG.



Filter: Any Health Status Any Lifecycle State Filter instances... 1 to 1 of 1 Instances

	Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
■	i-00173f2ca747883aa	InService	Jay-LT	us-east-1b	Healthy	

10.Put scale-in protection on an instance in the ASG.

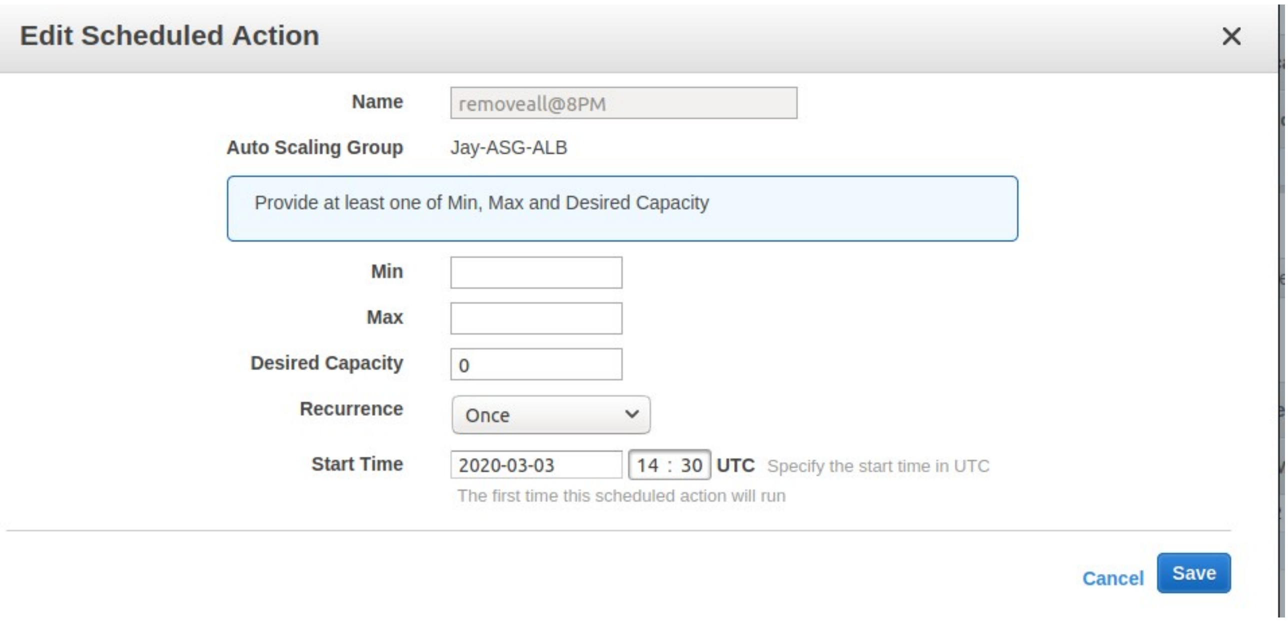


Filter: Any Health Status Any Lifecycle State Filter instances... 1 to 1 of 1 Instances

	Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
■	i-0b714a38ca4920a06	InService	Jay-LT		Healthy	

11.Put Schedules in ASG to:

- Remove all instances of the ASG at 8 PM



Edit Scheduled Action

Name: removeall@8PM

Auto Scaling Group: Jay-ASG-ALB

Provide at least one of Min, Max and Desired Capacity

Min:

Max:

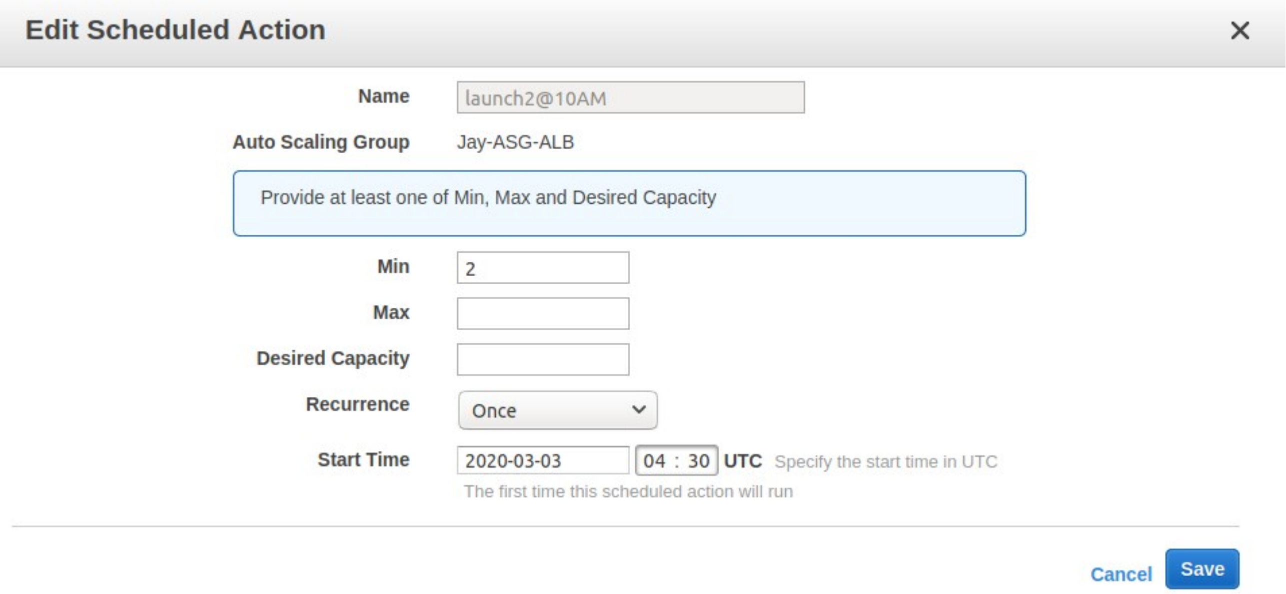
Desired Capacity: 0

Recurrence: Once

Start Time: 2020-03-03 14 : 30 UTC Specify the start time in UTC

The first time this scheduled action will run

- Launch a minimum of 2 instances at 10 AM



Edit Scheduled Action

Name: launch2@10AM

Auto Scaling Group: Jay-ASG-ALB

Provide at least one of Min, Max and Desired Capacity

Min: 2

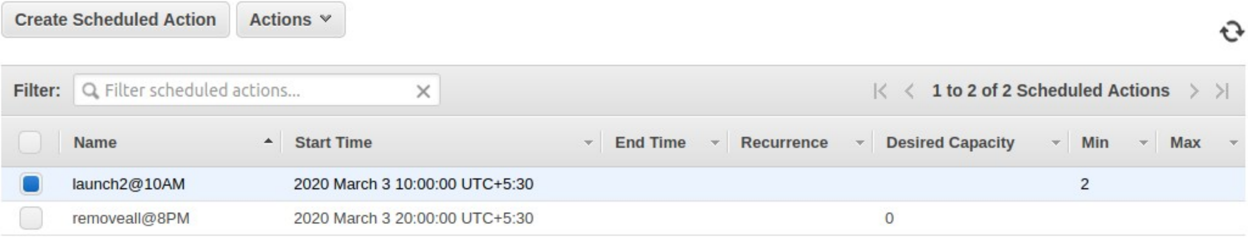
Max:

Desired Capacity:

Recurrence: Once

Start Time: 2020-03-03 04 : 30 UTC Specify the start time in UTC

The first time this scheduled action will run



1 to 2 of 2 Scheduled Actions

□	Name	Start Time	End Time	Recurrence	Desired Capacity	Min	Max
☑	launch2@10AM	2020 March 3 10:00:00 UTC+5:30				2	
□	removeall@8PM	2020 March 3 20:00:00 UTC+5:30			0		