amazon
web services

AUTO SCALING

Topics to be covered--Autoscaling

1) Autoscaling Introduction

2) Autoscaling Configuration

3) Scale in

4) Scale out

5) Test the output

6) Integrating Autoscaling with ALB

Amazon EC2 Auto Scaling

- ✓ Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application.

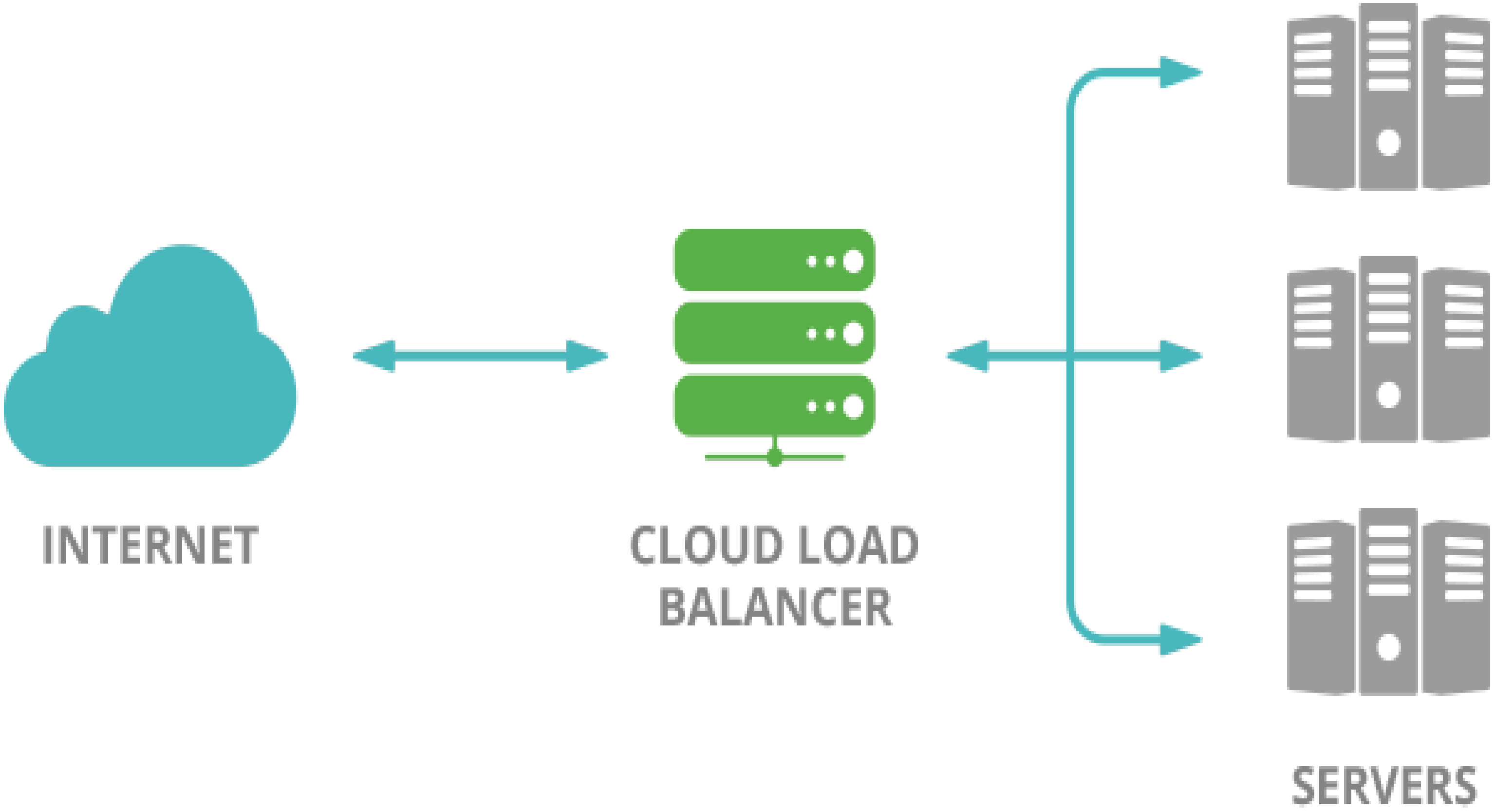
- ✓ You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size.

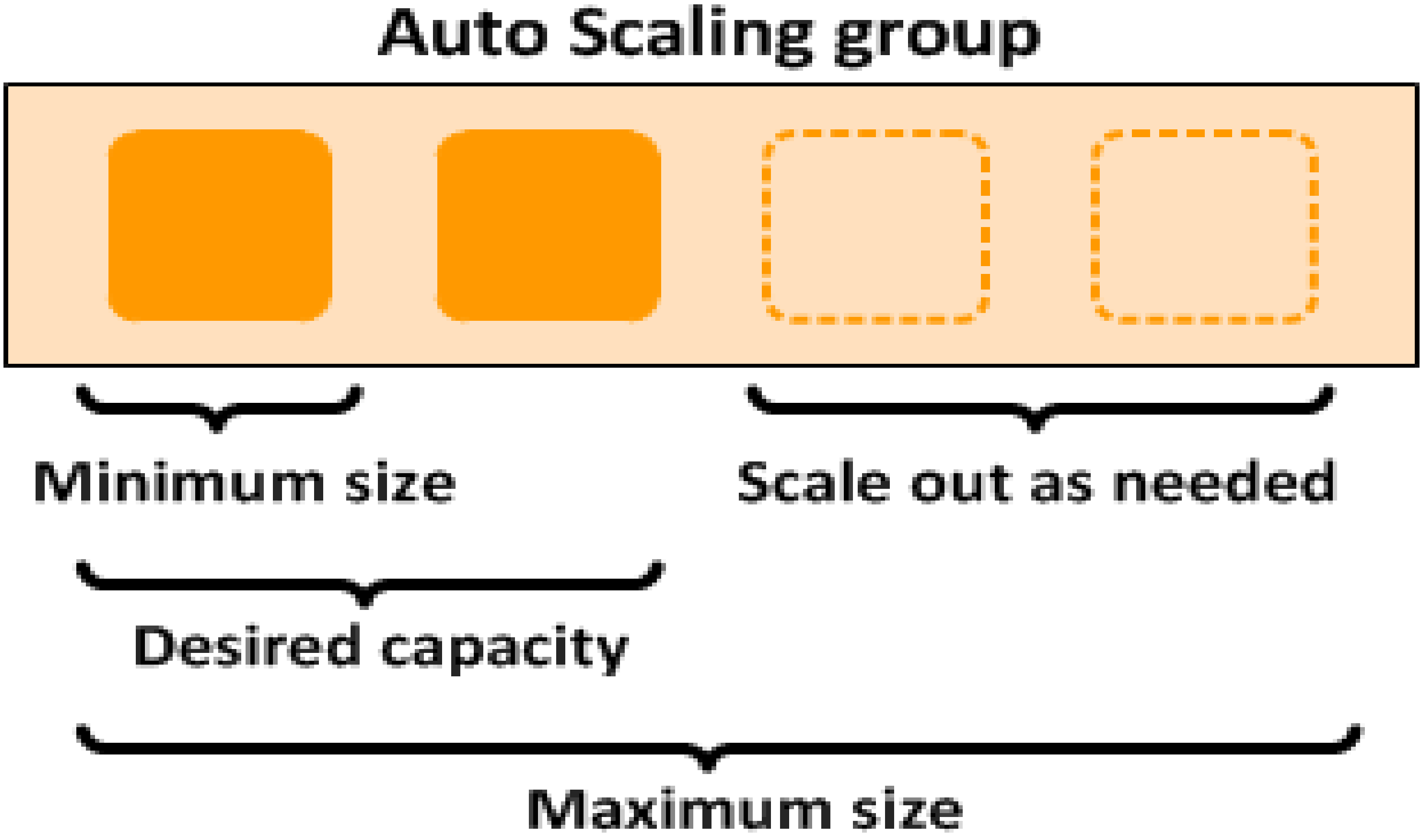
- ✓ You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.

- ✓ If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances.

Autoscaling

Do not pay for machines you’ are not using. Does most of your development and testing happen weekdays from 9 to 5? Use autoscaling to scale down all of your virtual machines at night or on the weekend when nobody is around and then have them ready to go Monday morning when you come in to work. The cloud is built to be elastic so you can be as cost-effective as possible.

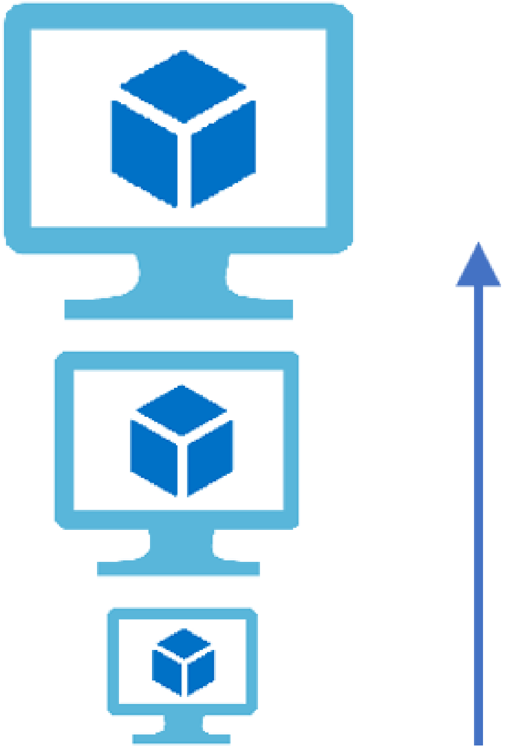

INTERNET
CLOUD LOAD BALANCER
SERVERS


Auto Scaling group
Minimum size
Scale out as needed
Desired capacity
Maximum size

Scaling Type

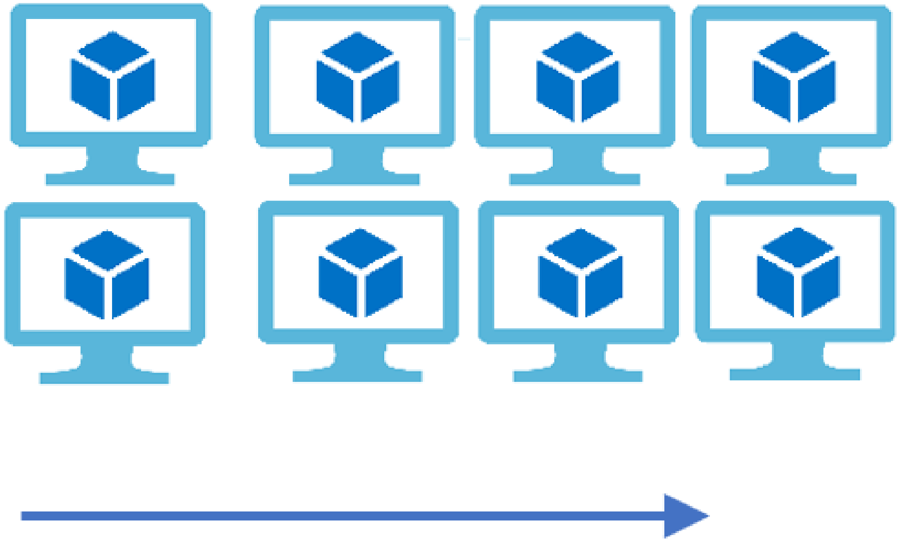
Vertical Scaling

(Increase size of instance (RAM , CPU etc.))



Horizontal Scaling

(Add more instances)



What is AWS Auto Scaling Group?

✓ Maintain application availability by scaling up/down your Amazon EC2 automatically

✓ Adds more instance during high load.

✓ Removes instances during less load.

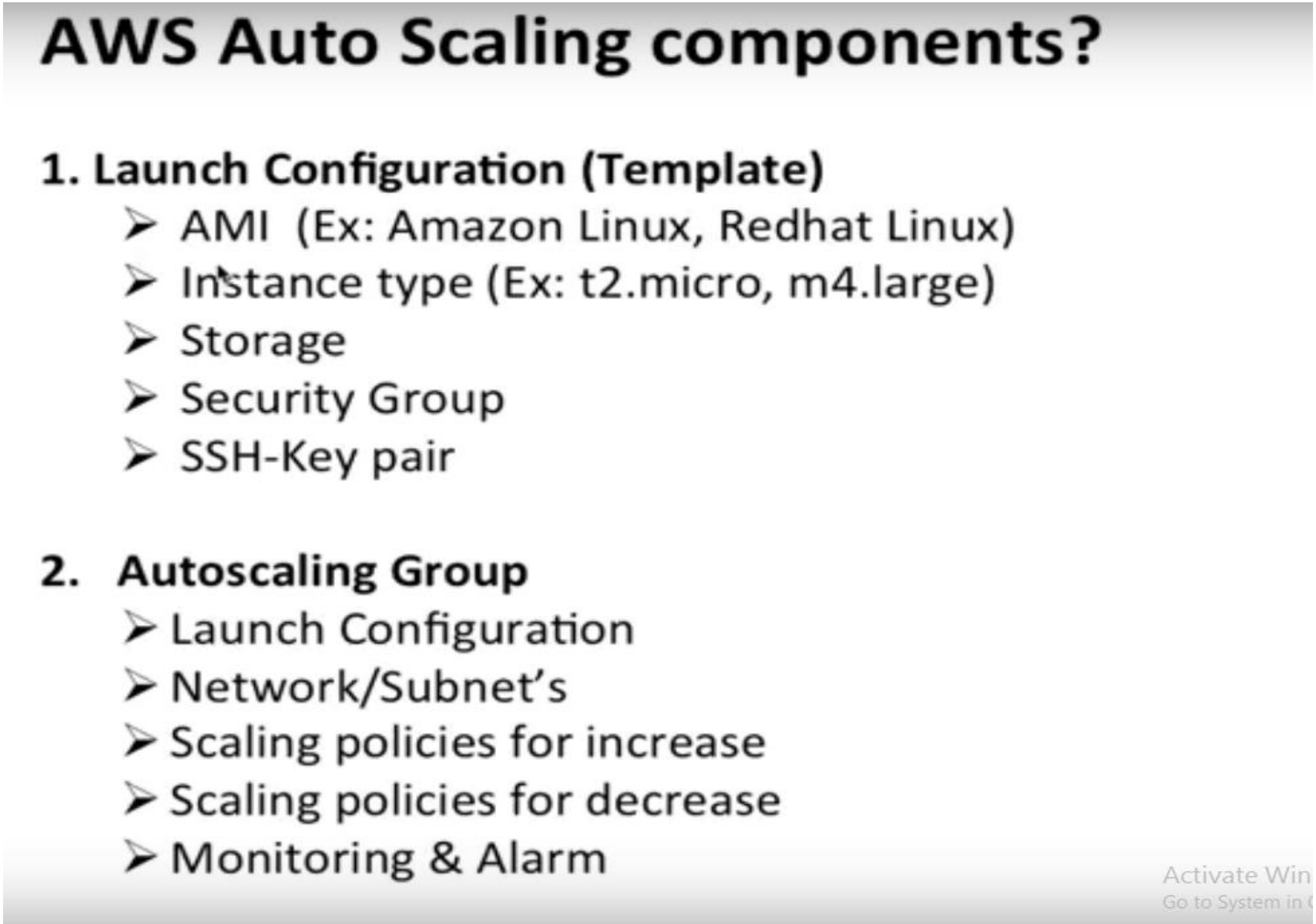
✓ Run desired number of Amazon EC2 instances. (Ex: Always 5 instances)

Benefits of Auto Scaling

- ✓ **Better fault tolerance.** Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.

- ✓ **Better availability**. Amazon EC2 Auto Scaling can help you ensure that your application always has the right amount of capacity to handle the current traffic demand.

- ✓ **Better cost management**. Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are actually needed and terminating them when they aren't needed.



AWS Auto Scaling components?

1. **Launch Configuration (Template)**
   - AMI (Ex: Amazon Linux, Redhat Linux)
   - Instance type (Ex: t2.micro, m4.large)
   - Storage
   - Security Group
   - SSH-Key pair

2. **Autoscaling Group**
   - Launch Configuration
   - Network/Subnet's
   - Scaling policies for increase
   - Scaling policies for decrease
   - Monitoring & Alarm

Auto Scaling Limits

Regional Limits

- ✓ Launch configurations per region: 200
- ✓ Auto Scaling groups per region: 200

Auto Scaling Group Limits

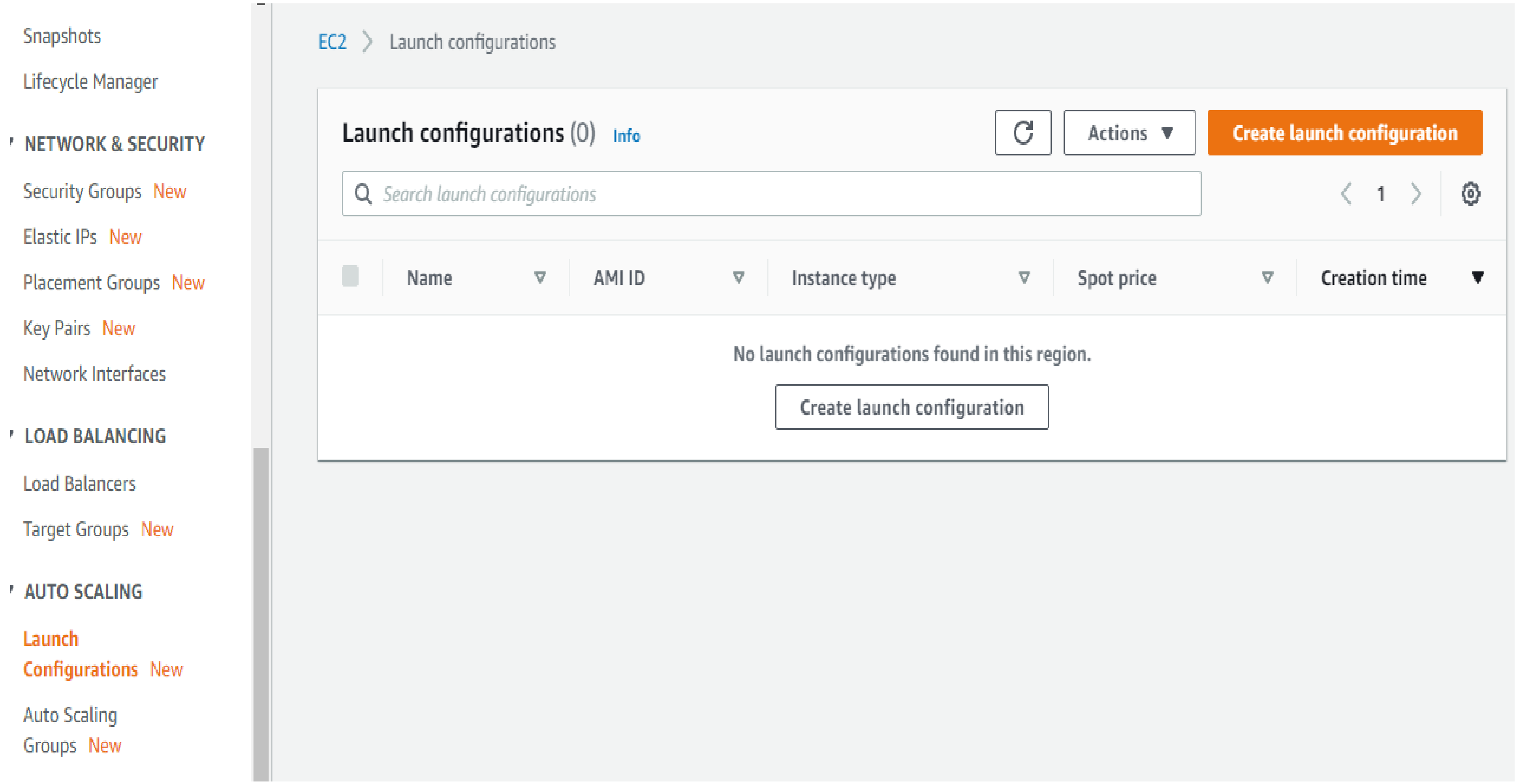
- ✓ Scaling policies per Auto Scaling group: 50
- ✓ Scheduled actions per Auto Scaling group: 125
- ✓ Lifecycle hooks per Auto Scaling group: 50
- ✓ SNS topics per Auto Scaling group: 10
- ✓ Classic Load Balancers per Auto Scaling group: 50
- ✓ Target groups per Auto Scaling group: 50

Scaling Policy Limits

- ✓ Step adjustments per scaling policy: 20

Creating a Launch Configuration Using an EC2 Instance

Open EC2 – scroll down –left -- click on Auto scaling ---Create launch configuration

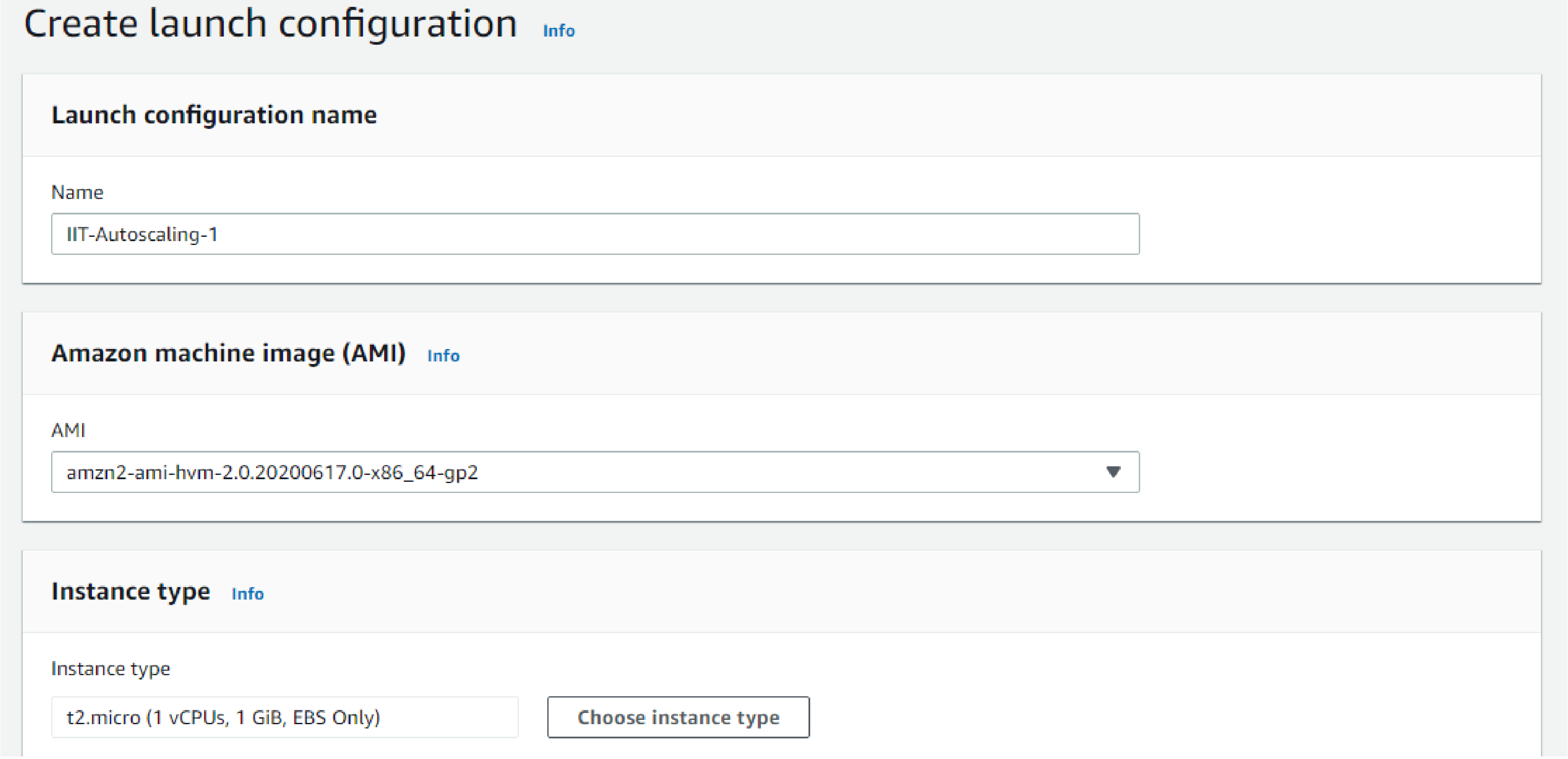


□	Name	AMI ID	Instance type	Spot price	Creation time

No launch configurations found in this region.

Creating a Launch Configuration Using an EC2 Instance

Type configuration name – Select AMI (Amazon Linux2 AMI)–Select Instance type –t2.micro



Create launch configuration Info

Launch configuration name

Name

IIT-Autoscaling-1

Amazon machine image (AMI) Info

AMI

amzn2-ami-hvm-2.0.20200617.0-x86_64-gp2

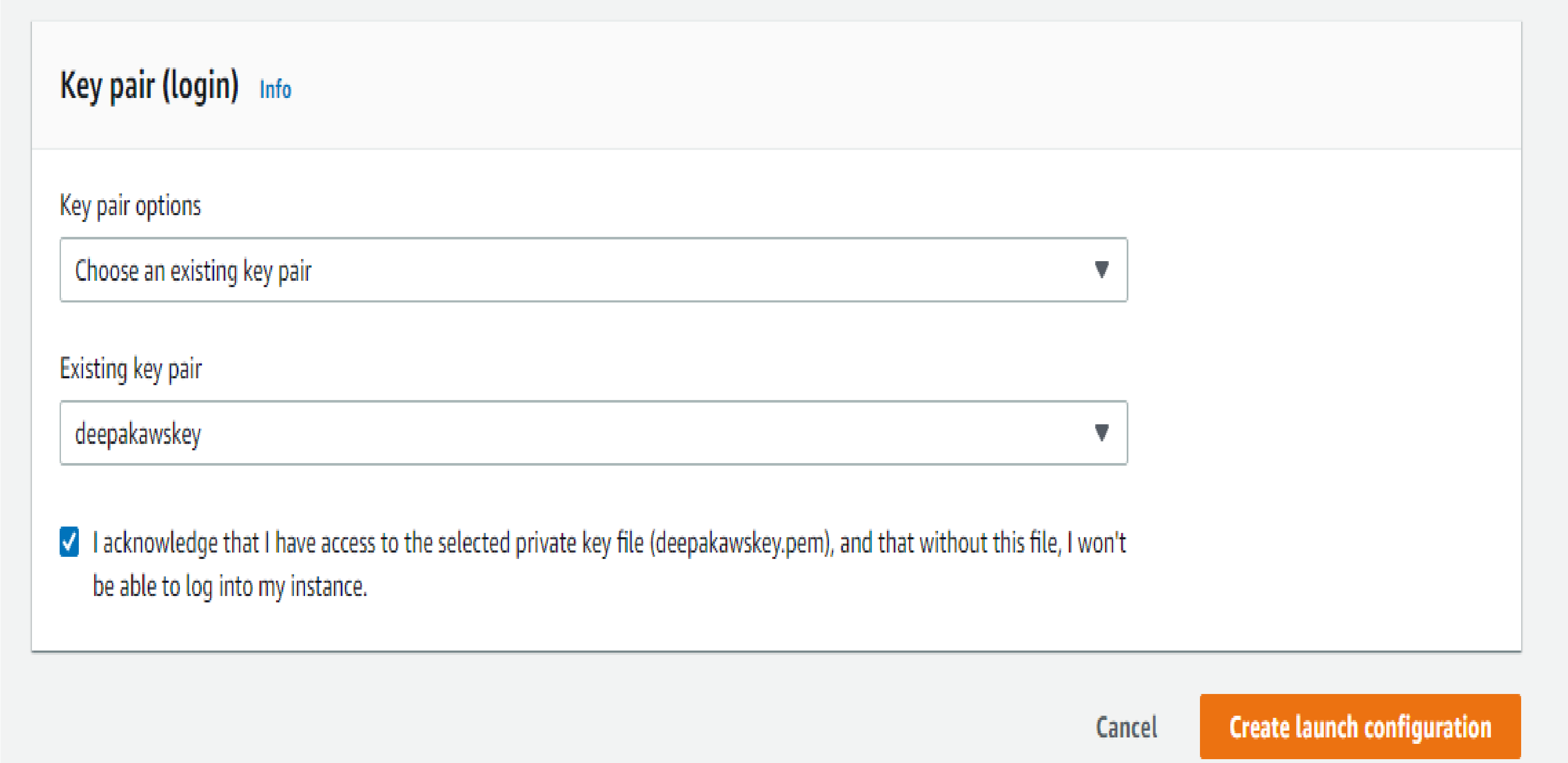
Instance type Info

Instance type

t2.micro (1 vCPUs, 1 GiB, EBS Only)

Creating a Launch Configuration Using an EC2 Instance

Select existing Security group-- Select existing key pair



Key pair (login)

Key pair options

Choose an existing key pair ▼

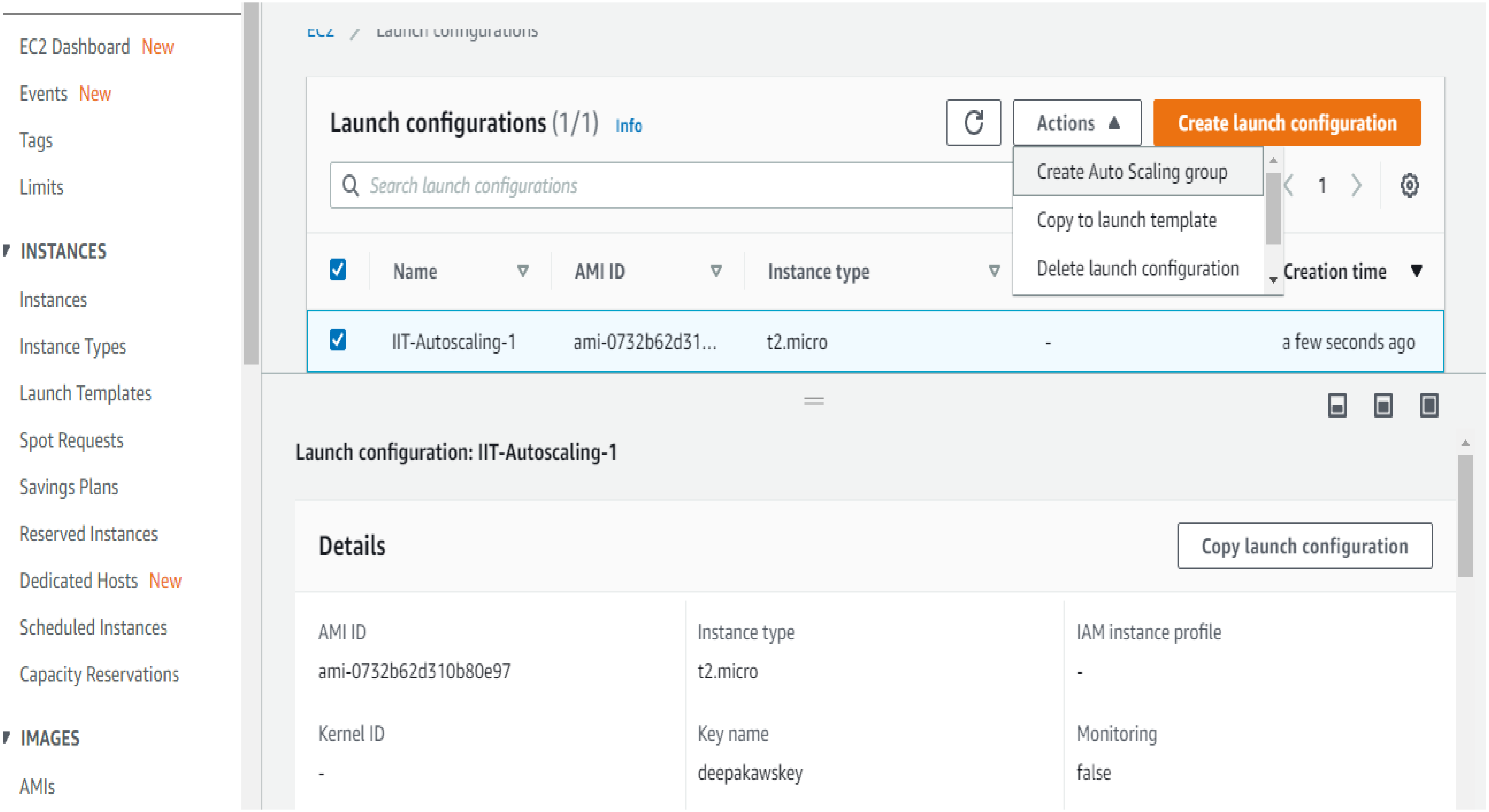
Existing key pair

deepakawskey ▼

☑ I acknowledge that I have access to the selected private key file (deepakawskey.pem), and that without this file, I won't be able to log into my instance.

Creating a Launch Configuration Using an EC2 Instance

After creating Select it –Action –Create Auto scaling group



Launch configurations (1/1) Info

☑	Name	AMI ID	Instance type		Creation time
☑	IIT-Autoscaling-1	ami-0732b62d31...	t2.micro	-	a few seconds ago

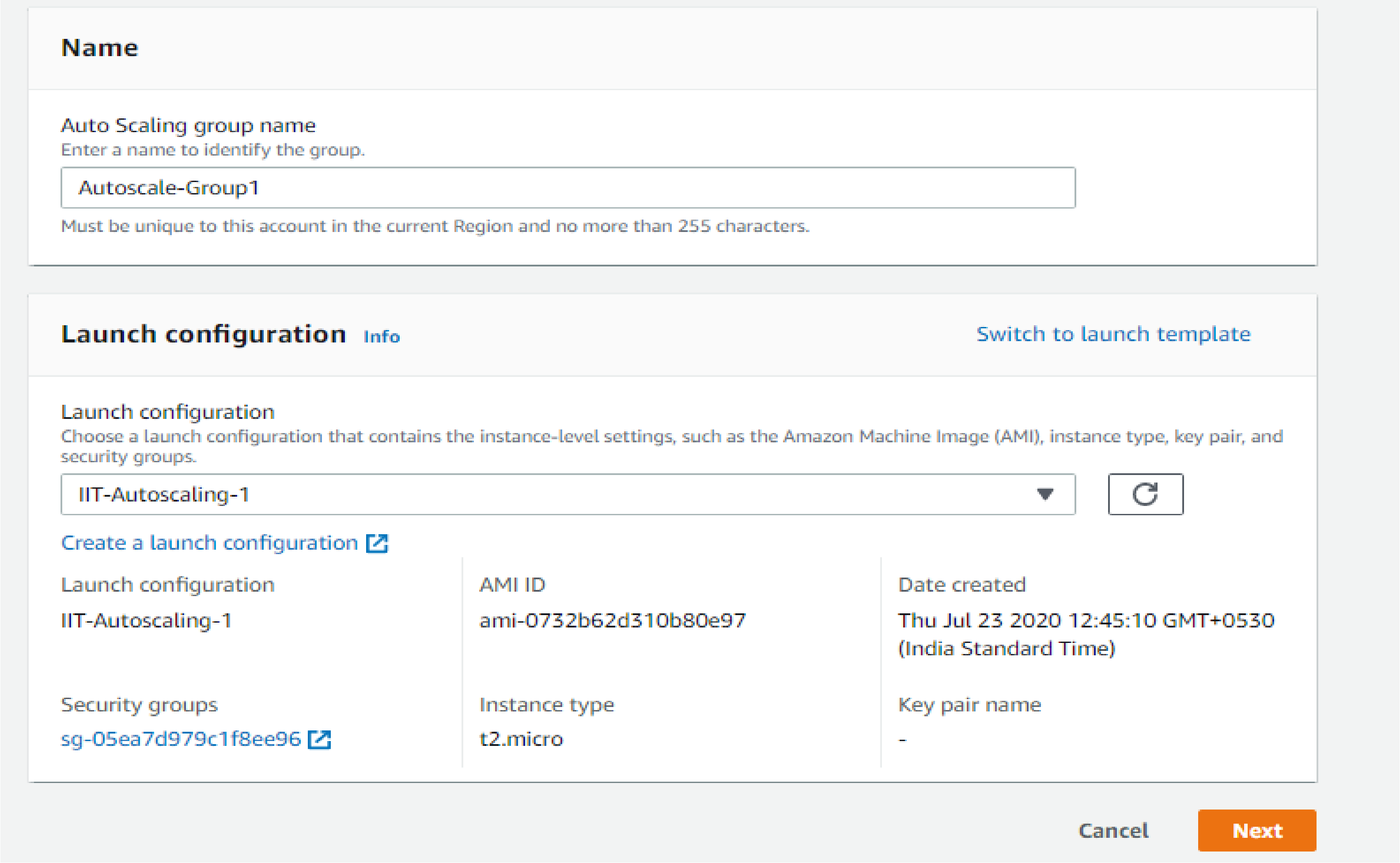
Launch configuration: IIT-Autoscaling-1

Details

AMI ID	Instance type	IAM instance profile
ami-0732b62d310b80e97	t2.micro	-
Kernel ID	Key name	Monitoring
-	deepakawskey	false

Creating a Launch Configuration Using an EC2 Instance

Give name –Autoscale group1--Next



Name

Auto Scaling group name

Enter a name to identify the group.

Autoscale-Group1

Must be unique to this account in the current Region and no more than 255 characters.

Launch configuration Info

Launch configuration

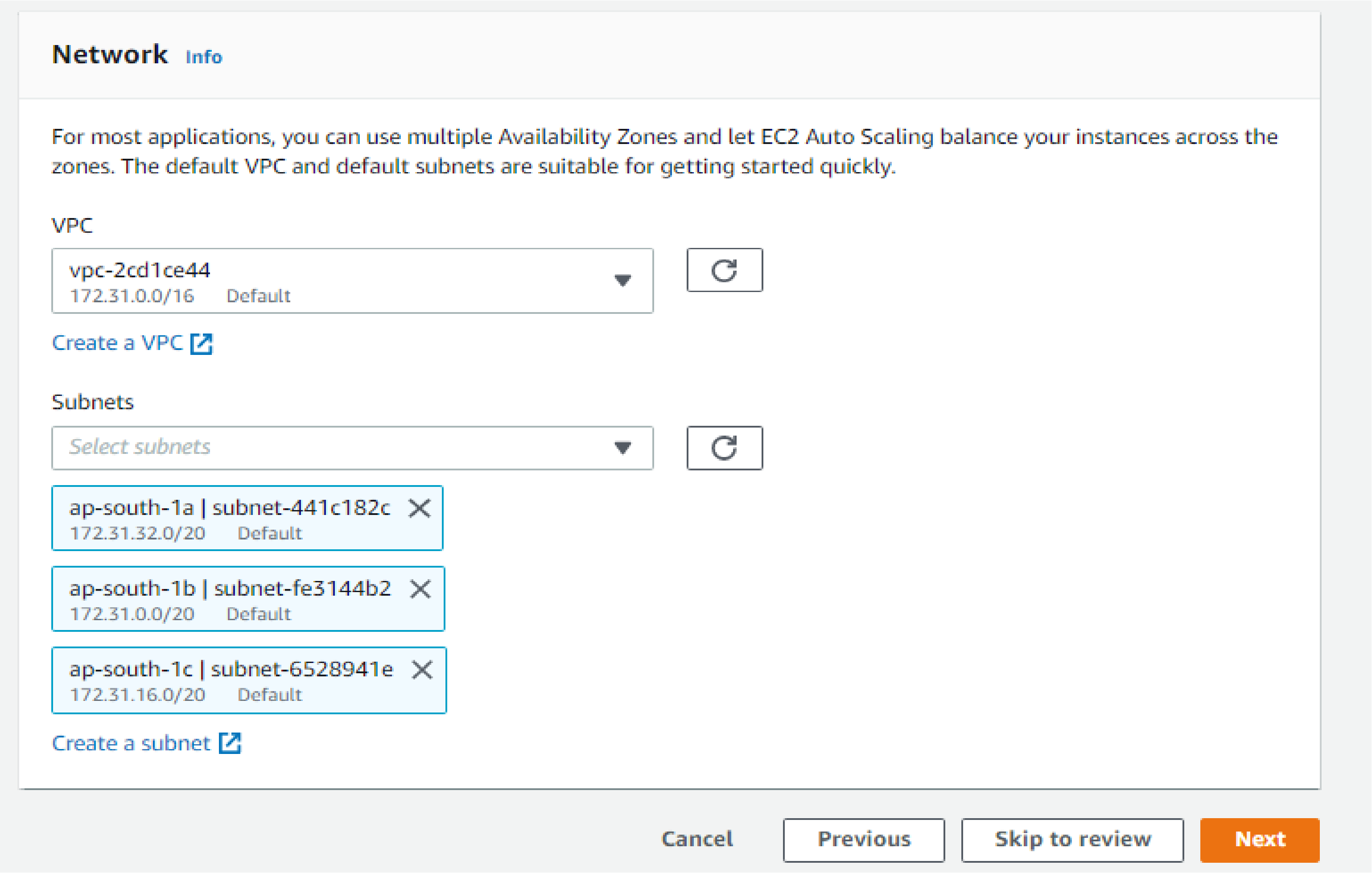
Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

IIT-Autoscaling-1

Launch configuration	AMI ID	Date created
IIT-Autoscaling-1	ami-0732b62d310b80e97	Thu Jul 23 2020 12:45:10 GMT+0530 (India Standard Time)
Security groups	Instance type	Key pair name
sg-05ea7d979c1f8ee96	t2.micro	-

Creating a Launch Configuration Using an EC2 Instance

Select VPC and all Subnets—next—set health check -60 sec –next--



Network Info

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-2cd1ce44
172.31.0.0/16 Default

Create a VPC

Subnets

Select subnets

ap-south-1a | subnet-441c182c ×
172.31.32.0/20 Default

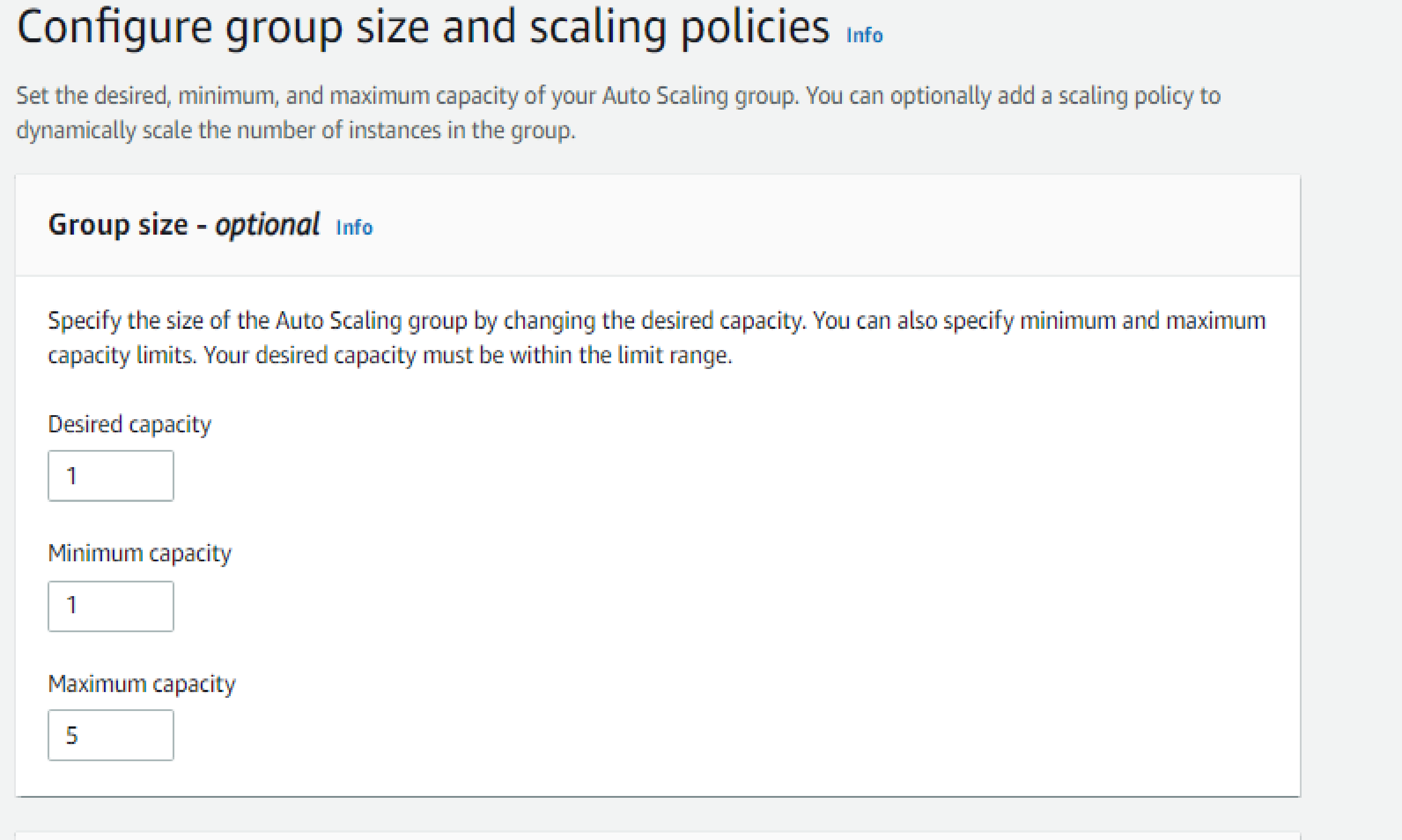
ap-south-1b | subnet-fe3144b2 ×
172.31.0.0/20 Default

ap-south-1c | subnet-6528941e ×
172.31.16.0/20 Default

Create a subnet

Creating a Launch Configuration Using an EC2 Instance

Set Desired, Minimum and Maximum capacity



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

1

Minimum capacity

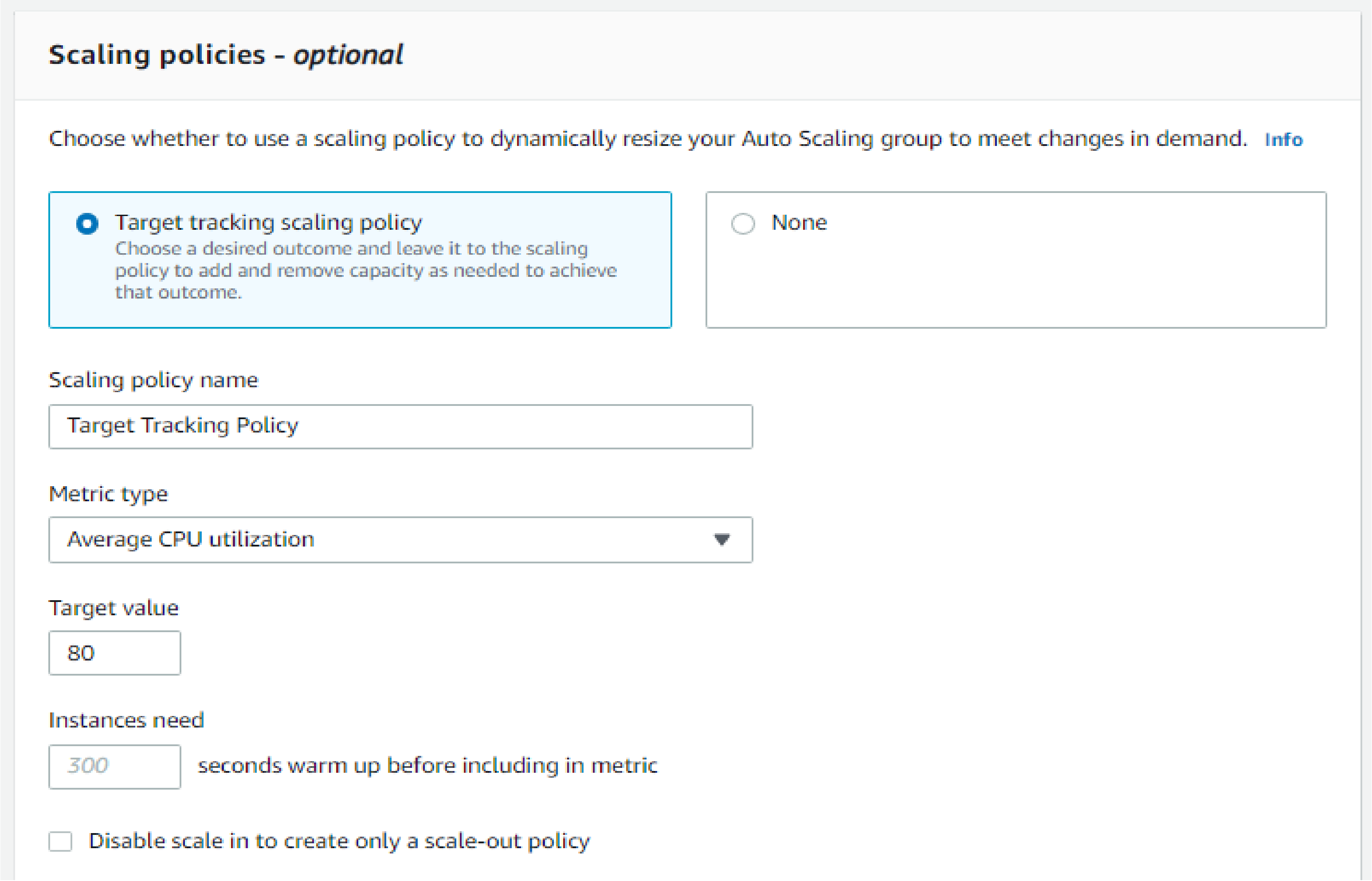
1

Maximum capacity

5

Creating a Launch Configuration Using an EC2 Instance

Configure Scaling Policies – Set CPU target value- next--



Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

● Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

○ None

Scaling policy name: Target Tracking Policy

Metric type: Average CPU utilization ▼

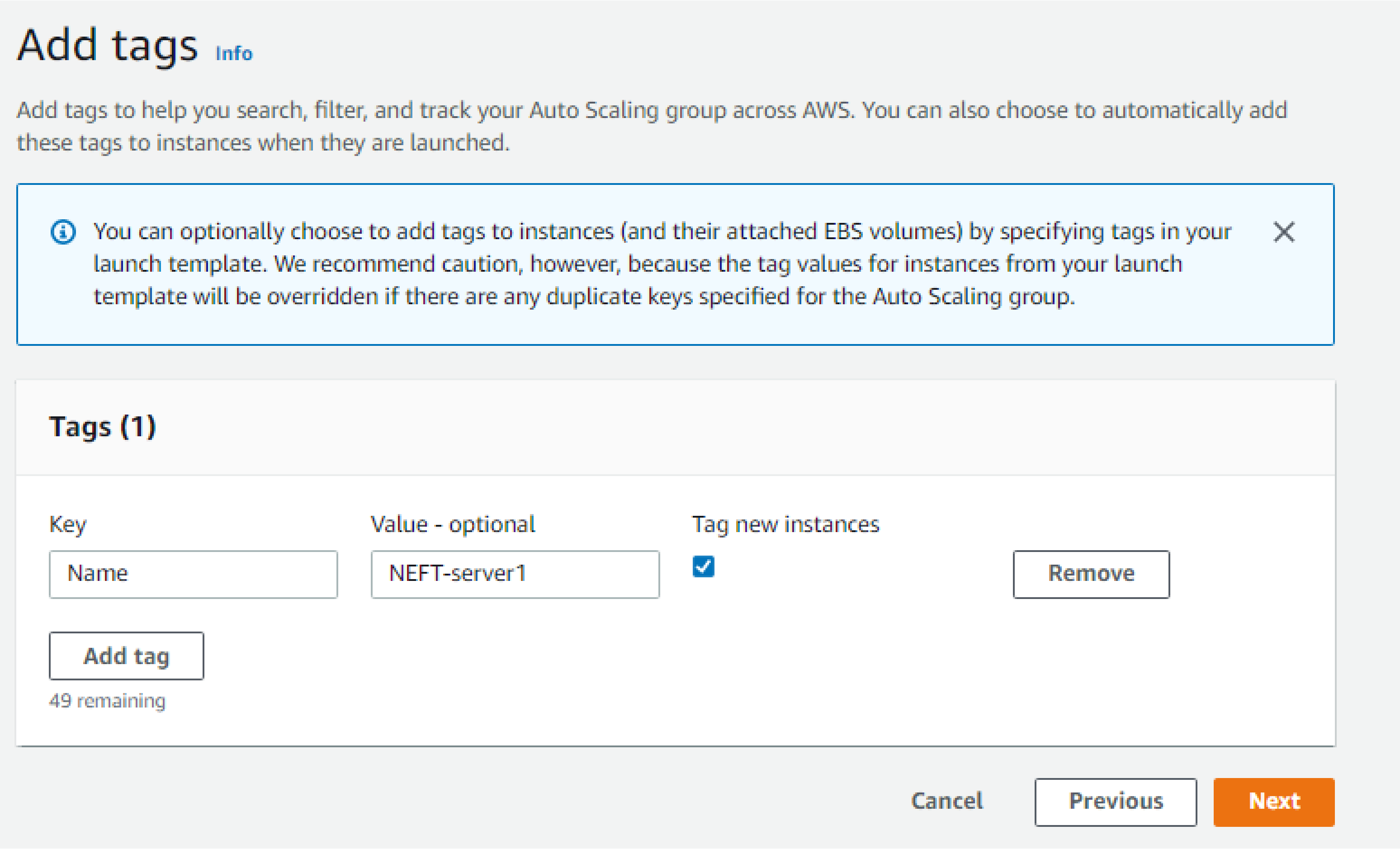
Target value: 80

Instances need 300 seconds warm up before including in metric

□ Disable scale in to create only a scale-out policy

Creating a Launch Configuration Using an EC2 Instance

Give the Instance name—Next—Review and Create--Create



Add tags

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

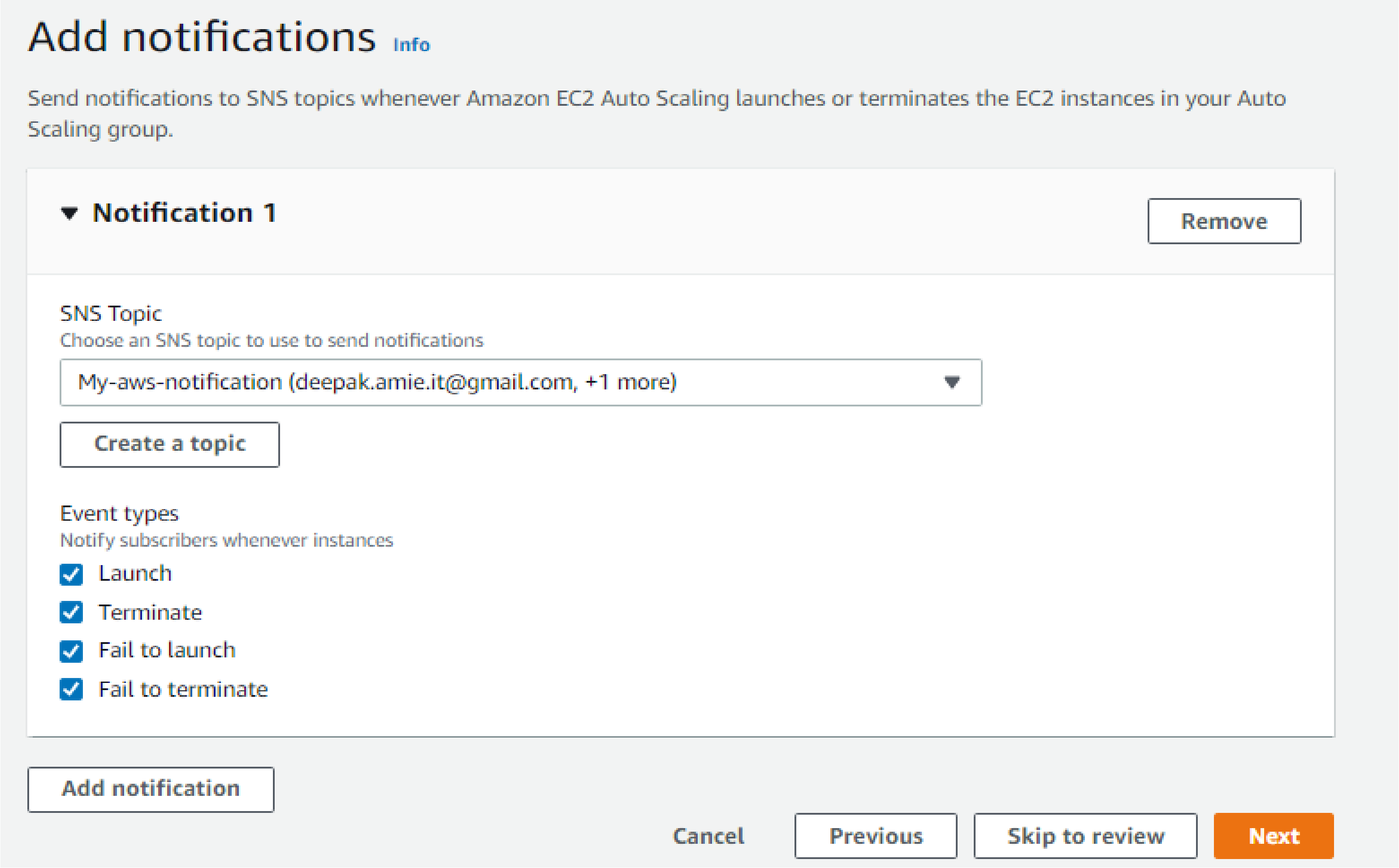
Tags (1)

Key	Value - optional	Tag new instances
Name	NEFT-server1	☑

49 remaining

Creating a Launch Configuration Using an EC2 Instance

Click on Add notification—select SNS topic



Add notifications

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

▼ Notification 1

SNS Topic

Choose an SNS topic to use to send notifications

My-aws-notification (deepak.amie.it@gmail.com, +1 more) ▼

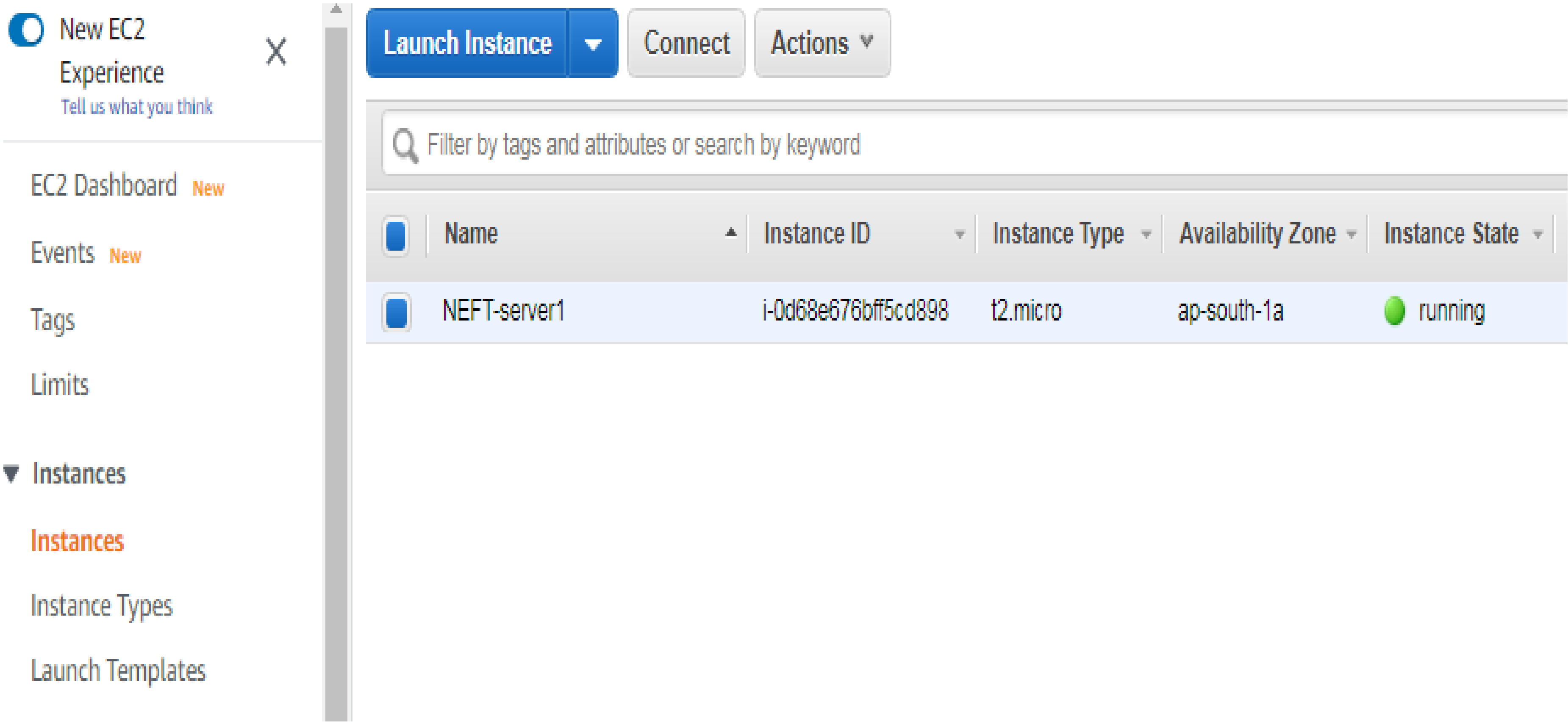
Event types

Notify subscribers whenever instances

- ☑ Launch
- ☑ Terminate
- ☑ Fail to launch
- ☑ Fail to terminate

Creating a Launch Configuration Using an EC2 Instance

Now click on Instance –Check a new Instance launched here



	Name	Instance ID	Instance Type	Availability Zone	Instance State
	NEFT-server1	i-0d68e676bff5cd898	t2.micro	ap-south-1a	● running

How to increase CPU load manually ?

- **Open Linux Instance**

```
# sudo su
# top
%cpu = 0
# yes >/dev/null  &
# top
%cpu = 99
```

How to decrease cpu load manually ?

Ans:

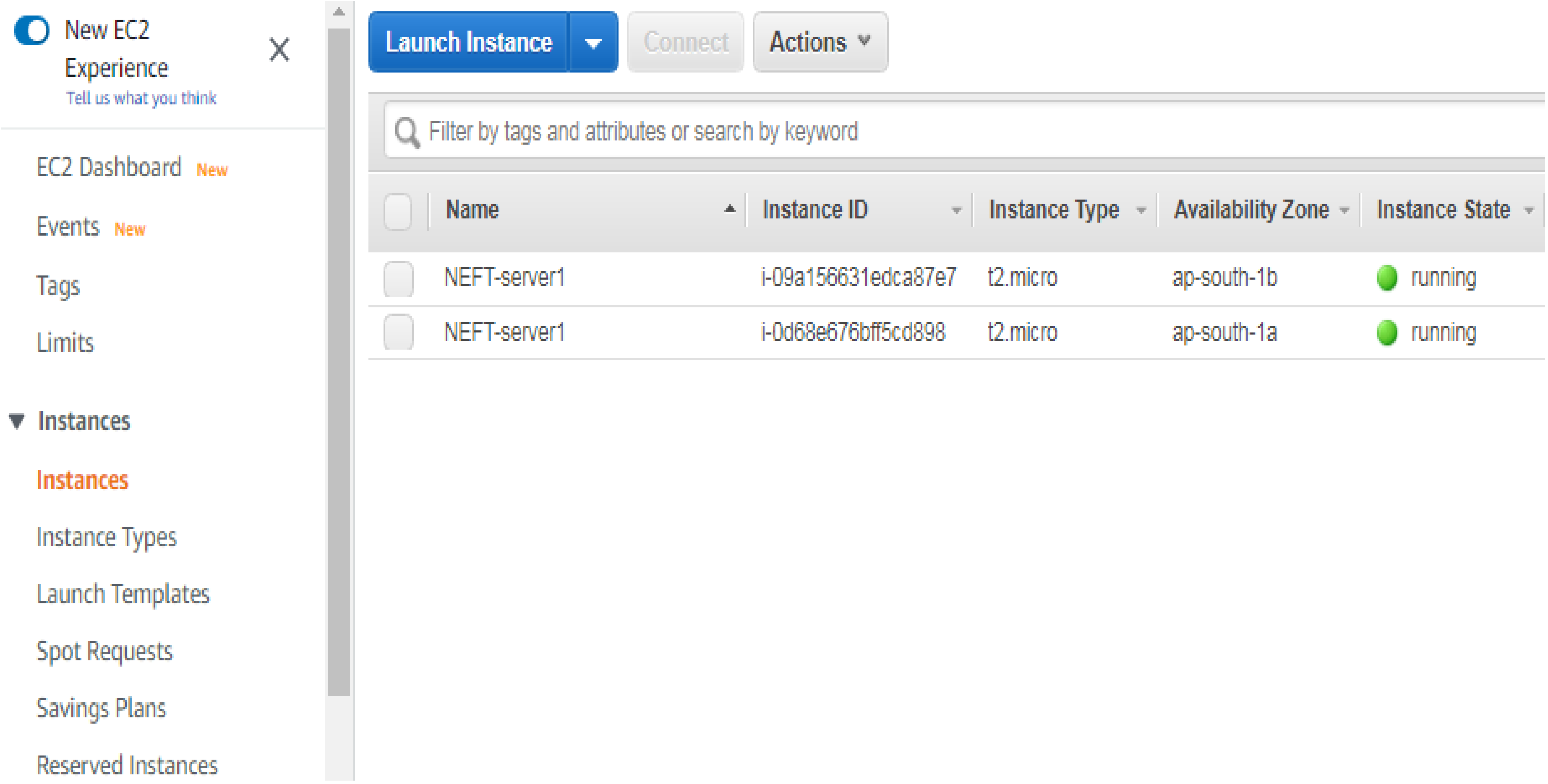
```
# top
%cpu = 99
```

check th pip consuming more cpu usage (eg : 3147)

```
# kill  -9  3147
```

Final Output

Now after 5 minute check in Instances –New instance will be launched automatically one by one



□	Name	Instance ID	Instance Type	Availability Zone	Instance State
□	NEFT-server1	i-09a156631edca87e7	t2.micro	ap-south-1b	● running
□	NEFT-server1	i-0d68e676bff5cd898	t2.micro	ap-south-1a	● running