



## Ch-9

# Auto scaling and Cloudwatch



# This chapter will cover following topics...

- Introduction to Auto scaling
- Why auto scaling
- Advantages of auto scaling
- Auto scaling components
  - 1. Launch configuration
  - 2. Auto scaling group
- Scale Up vs. Scale Out
- Hands on - Auto scaling
- Introduction to Cloudwatch
- Working of Cloudwatch
- Elements of Cloudwatch
- Hands on – Creating Cloudwatch Alarm
- Cloudwatch dashboard
- Monitoring types – Basic and Detailed

# Introduction to Auto scaling

Auto Scaling is a service that allows you to scale your Amazon EC2 capacity automatically by scaling out and scaling in according to criteria that you define.

With Auto Scaling, you can ensure that the number of running Amazon EC2 instances increases during demand spikes or peak demand periods to maintain application performance and decreases automatically during demand lulls or troughs to minimize costs.



# Why auto scaling

Load on application varies

A good design must take care of varying load

Since load spike cannot be anticipated always, manual scaling is not the solution

Other solution is to over provision resources

This is inefficient and costly

Auto scaling is the best solution for dealing with varying load

# Advantages of auto scaling

Better Fault Tolerance

Better Availability

Better Cost Management

Integrate with Load Balancing for maximum effect

# Auto scaling components

## 1. Launch configuration:

It is the configuration that Auto Scaling uses to create new instances, and it is composed of the configuration name, Amazon Machine Image (AMI), Amazon EC2 instance type, security group, and instance key pair.

Each Auto Scaling group can have only one launch configuration at a time.

The default limit for launch configurations is 100 per region.

If you exceed this limit, the call to create-launch-configuration will fail.

Auto Scaling may cause you to reach limits of other services, such as the default number of Amazon EC2 instances you can currently launch within a region, which is 20.

It is important to keep in mind the service limits for all AWS Cloud services you are using.

# Auto scaling components(Conti...)

## 2. Auto scaling group:

It is a collection of Amazon EC2 instances managed by the Auto Scaling service.

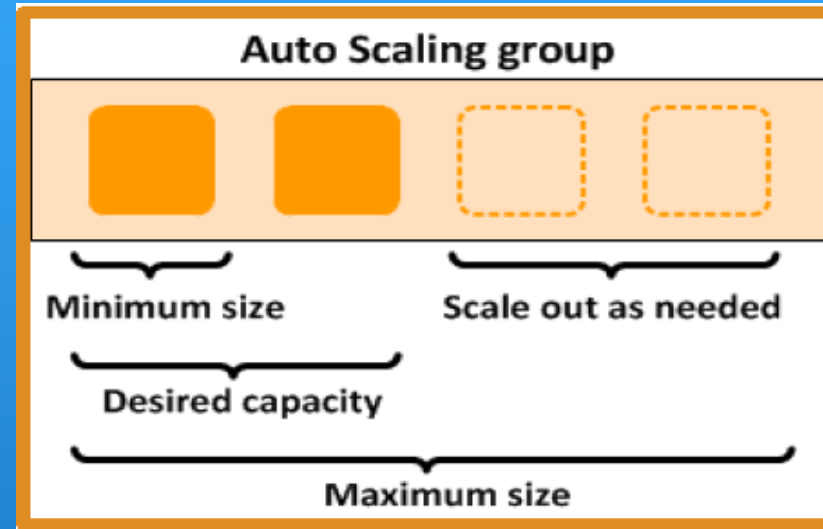
Each Auto Scaling group contains configuration options that control when Auto Scaling should launch new instances and terminate existing instances.

An Auto Scaling group must contain a name and a minimum and maximum number of instances that can be in the group.

You can optionally specify desired capacity, which is the number of instances that the group must have at all times.

# Auto scaling components(Conti...)

## 2. Auto scaling group(Conti...):



**Minimum Size** : Group should contain No. of minimum instance

**Desired Capacity** : When starting group how many instance should be start

**Maximum size** : No instance maximum that group can launch



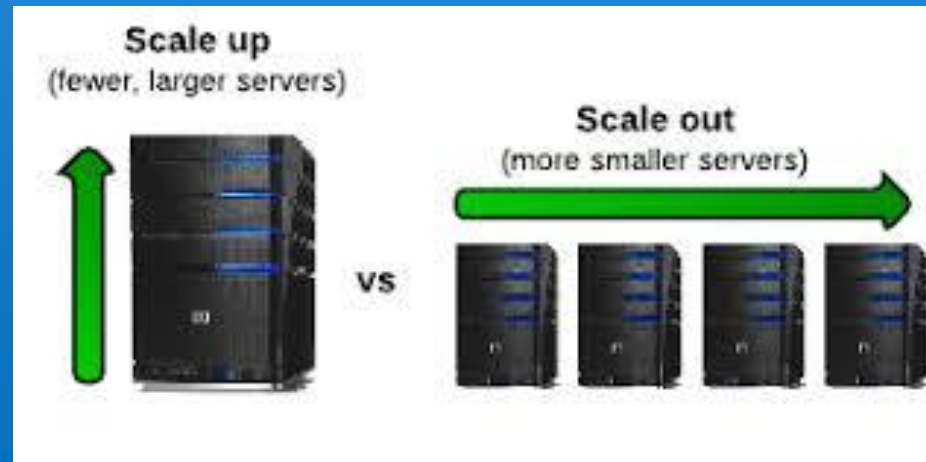
# Scale Up vs. Scale Out

Scale out Amazon EC2 instances seamlessly and automatically when demand increases.

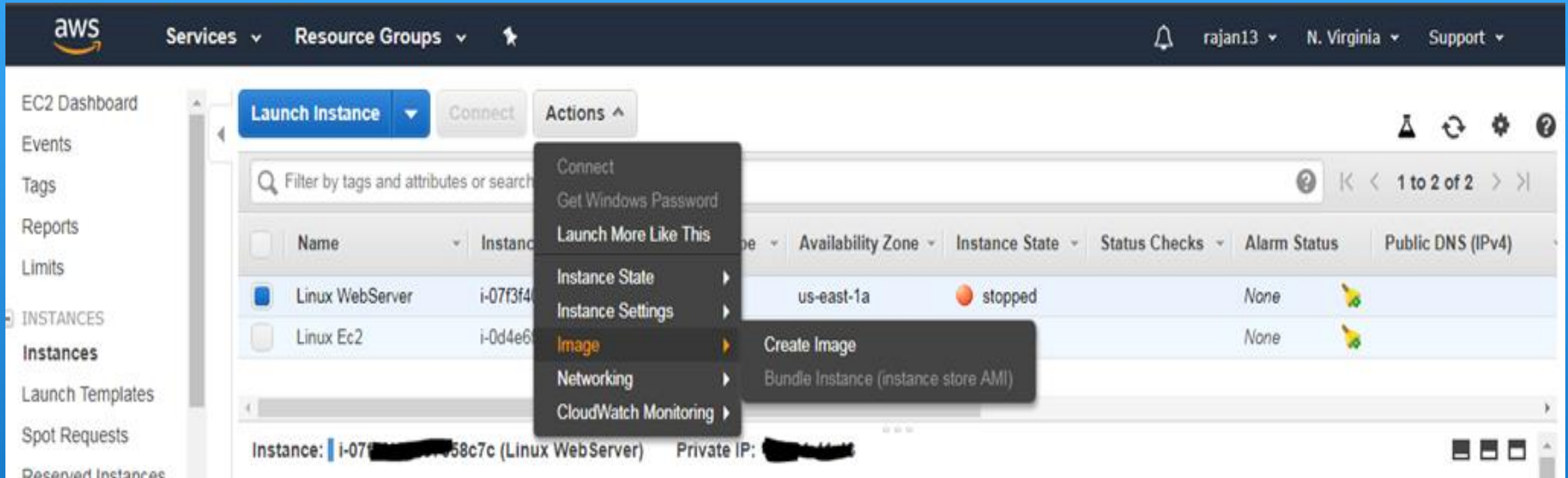
Scale in Amazon EC2 instances automatically and save money when demand decrease.

Increasing resources vertically is called Scale Up. Example: CPU Speed, Memory size etc.

Increasing resources horizontally is called Scale Out. Example: Deploying more instances.



# Hands on – Auto scaling



The screenshot displays the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (rajan13, N. Virginia, Support). The left sidebar shows the 'EC2 Dashboard' and 'INSTANCES' section, with 'Instances' selected. The main content area shows a table of EC2 instances. The 'Actions' menu is open, showing options like 'Connect', 'Get Windows Password', 'Launch More Like This', 'Instance State', 'Instance Settings', 'Image', 'Networking', and 'CloudWatch Monitoring'. The 'Image' option is highlighted, and a sub-menu is open, showing 'Create Image' and 'Bundle Instance (instance store AMI)'. The table lists two instances: 'Linux WebServer' (i-07f3f4...) and 'Linux Ec2' (i-0d4e6...). The 'Linux WebServer' instance is in the 'stopped' state in the 'us-east-1a' availability zone. The bottom of the console shows the details for the selected instance: 'Instance: i-07f3f4...58c7c (Linux WebServer) Private IP: ...'.

Name	Instance ID	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Linux WebServer	i-07f3f4...	us-east-1a	stopped	None	None	
Linux Ec2	i-0d4e6...			None	None	

# Hands on – Auto scaling(Conti...)

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (rajan13, N. Virginia, Support). The left sidebar lists navigation options: Instances, Launch Templates, Spot Requests (highlighted), Reserved Instances, Dedicated Hosts, and Scheduled Instances. The main content area shows the 'Launch' and 'Actions' buttons, a search bar, and a table of AMIs.

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
Linux httpd A...	Linux httpd AMI	ami-011d5fe54fc208384	227790519329/...	227790519329	Private	available	September 5, 2018 at 5:59:1...

# Hands on – Auto scaling(Conti...)

**aws** Services ▾ Resource Groups ▾ ★

🔔 rajan13 ▾ N. Virginia ▾ Support ▾

EC2 Dashboard  
Events  
Tags  
Reports  
Limits

INSTANCES  
Instances  
Launch Templates  
Spot Requests  
Reserved Instances  
Dedicated Hosts  
Scheduled Instances

**Launch Templates have arrived!**  
The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or [Learn more](#).

**Create launch configuration** Create Auto Scaling group Copy to launch template Actions ▾

Filter: 🔍 Filter launch configurations... ✕

⏪ < No Launch Configurations >

<input type="checkbox"/>	Name	AMI ID	Instance Type	Spot Price	Creation Time
No launch configurations found					

# Hands on – Auto scaling(Conti...)

The screenshot shows the AWS Management Console interface for creating a launch configuration. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (rajan13, N. Virginia, Support). The main header shows a progress bar with six steps: 1. Choose AMI (active), 2. Choose Instance Type, 3. Configure details, 4. Add Storage, 5. Configure Security Group, and 6. Review. A 'Cancel and Exit' link is in the top right.

## Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.


Quick Start

My AMIs

AWS Marketplace



Community AMIs


Search my AMIs

**Linux httpd AMI - ami-011d5fe54fc208384**  
Linux httpd AMI  
Root device type: ebs    Virtualization type: hvm    Owner: 227790519329

Select  
64-bit

# Hands on – Auto scaling(Conti...)

 Services ▾ Resource Groups ▾ 

 rajan13 ▾ N. Virginia ▾ Support ▾

[1. Choose AMI](#) [2. Choose Instance Type](#) [3. Configure details](#) [4. Add Storage](#) [5. Configure Security Group](#) [6. Review](#)

## Create Launch Configuration

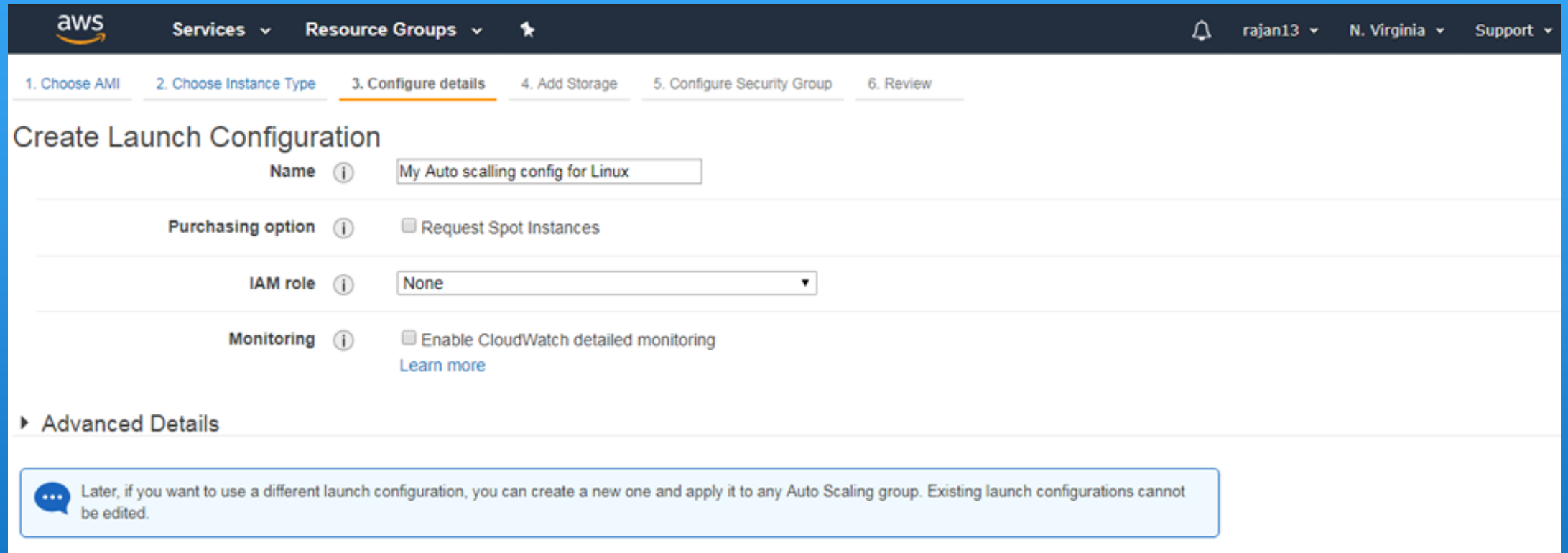
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types ▾ Current generation ▾ [Show/Hide Columns](#)

**Currently selected:** t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family ▾	Type ▾	vCPUs ⓘ ▾	Memory (GiB) ▾	Instance Storage (GB) ⓘ ▾	EBS-Optimized Available ⓘ ▾	Network Performance ⓘ ▾
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate

# Hands on – Auto scaling(Conti...)



The screenshot shows the AWS Management Console interface for creating a new Launch Configuration. The top navigation bar includes the AWS logo, 'Services' and 'Resource Groups' dropdown menus, a star icon, and user account information (rajan13, N. Virginia, Support). Below the navigation bar is a progress bar with six steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure details (highlighted), 4. Add Storage, 5. Configure Security Group, and 6. Review.



The main heading is 'Create Launch Configuration'. Below it are four configuration sections, each with an information icon (i):


- Name:** A text input field containing 'My Auto scaling config for Linux'.
- Purchasing option:** A checkbox labeled 'Request Spot Instances' which is currently unchecked.
- IAM role:** A dropdown menu currently set to 'None'.
- Monitoring:** A checkbox labeled 'Enable CloudWatch detailed monitoring' which is unchecked, with a 'Learn more' link below it.

Below these sections is a section titled 'Advanced Details' with a right-pointing triangle icon.

A light blue informational box at the bottom contains a speech bubble icon and the text: 'Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.'

# Hands on – Auto scaling(Conti...)

 Services ▾ Resource Groups ▾ 

 rajan13 ▾ N. Virginia ▾ Support ▾

1. Choose AMI

2. Choose Instance Type

3. Configure details










4. Add Storage

5. Configure Security Group

6. Review

## Create Launch Configuration

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes.  
<https://docs.aws.amazon.com/console/ec2/launchinstance/storage> about storage options in Amazon EC2.

Volume Type 	Device 	Snapshot 	Size (GiB) 	Volume Type 	IOPS 	Throughput 	Delete on Termination 	Encrypted 
Root	/dev/xvda	snap-0e2c11b57884e2fdd	<input type="text" value="8"/>	General Purpose (SSD) ▾	100 / 3000	N/A	<input checked="" type="checkbox"/>	No

Add New Volume



# Hands on – Auto scaling(Conti...)

The screenshot shows the AWS Management Console interface for creating a launch configuration. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (rajan13, N. Virginia, Support). The breadcrumb trail shows the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure details, 4. Add Storage, 5. Configure Security Group (highlighted), and 6. Review.

## Create Launch Configuration

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

**Assign a security group:**

- ☒ Create a **new** security group
- ☐ Select an **existing** security group

**Security group name:**

**Description:**

Type <small>i</small>	Protocol <small>i</small>	Port Range <small>i</small>	Source <small>i</small>
SSH ▾	TCP	22	Anywhere ▾ 0.0.0.0/0

# Hands on – Auto scaling(Conti...)

## Launch configuration creation status



**Successfully created launch configuration: My Auto scaling config for Linux**

[View creation log](#)

### ▼ View

[View your launch configurations](#)

[View your Auto Scaling groups](#)

► Here are some helpful resources to get you started

Create an Auto Scaling group using this launch configuration

Close

# Hands on – Auto scaling(Conti...)

**aws** Services ▾ Resource Groups ▾ ★

rajan13 ▾ N. Virginia ▾ Support ▾

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

## Create Auto Scaling Group [Cancel and Exit](#)

**Launch Configuration** ⓘ My Auto scaling config for Linux

**Group name** ⓘ

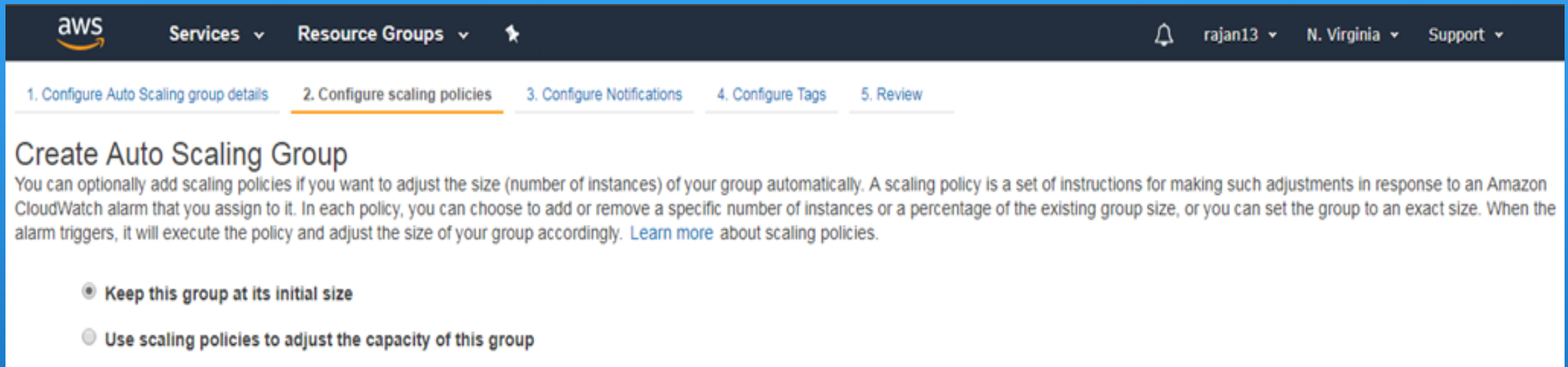
**Group size** ⓘ Start with  instances

**Network** ⓘ  [Create new VPC](#)

**Subnet** ⓘ  [Create new subnet](#)

Each instance in this Auto Scaling group will be assigned a public IP address. ⓘ

# Hands on – Auto scaling(Conti...)



The screenshot shows the AWS Management Console interface for creating an Auto Scaling Group. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'rajan13' in the 'N. Virginia' region. The breadcrumb trail indicates the current step is '2. Configure scaling policies'. The main heading is 'Create Auto Scaling Group'. Below the heading, a paragraph explains that scaling policies can be added to adjust the group size automatically based on CloudWatch alarms. Two radio button options are presented: 'Keep this group at its initial size' (which is selected) and 'Use scaling policies to adjust the capacity of this group'.

aws Services ▾ Resource Groups ▾

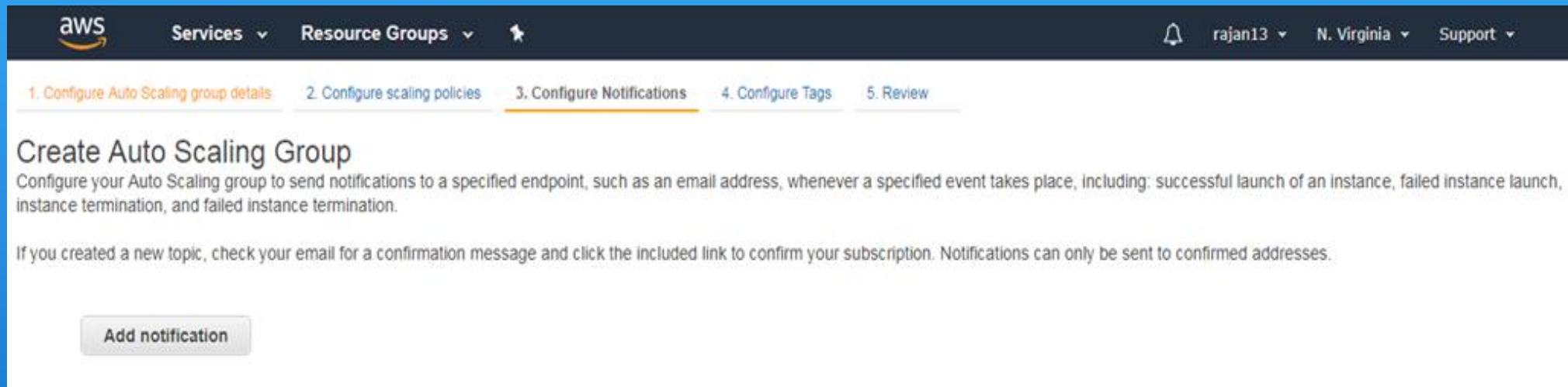
1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

## Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. [Learn more](#) about scaling policies.

- ☒ Keep this group at its initial size
- ☐ Use scaling policies to adjust the capacity of this group

# Hands on – Auto scaling(Conti...)



The screenshot shows the AWS Management Console interface for creating an Auto Scaling Group. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (rajan13, N. Virginia, Support). Below the navigation bar is a progress bar with five steps: 1. Configure Auto Scaling group details, 2. Configure scaling policies, 3. Configure Notifications (active), 4. Configure Tags, and 5. Review.

## Create Auto Scaling Group

Configure your Auto Scaling group to send notifications to a specified endpoint, such as an email address, whenever a specified event takes place, including: successful launch of an instance, failed instance launch, instance termination, and failed instance termination.

If you created a new topic, check your email for a confirmation message and click the included link to confirm your subscription. Notifications can only be sent to confirmed addresses.

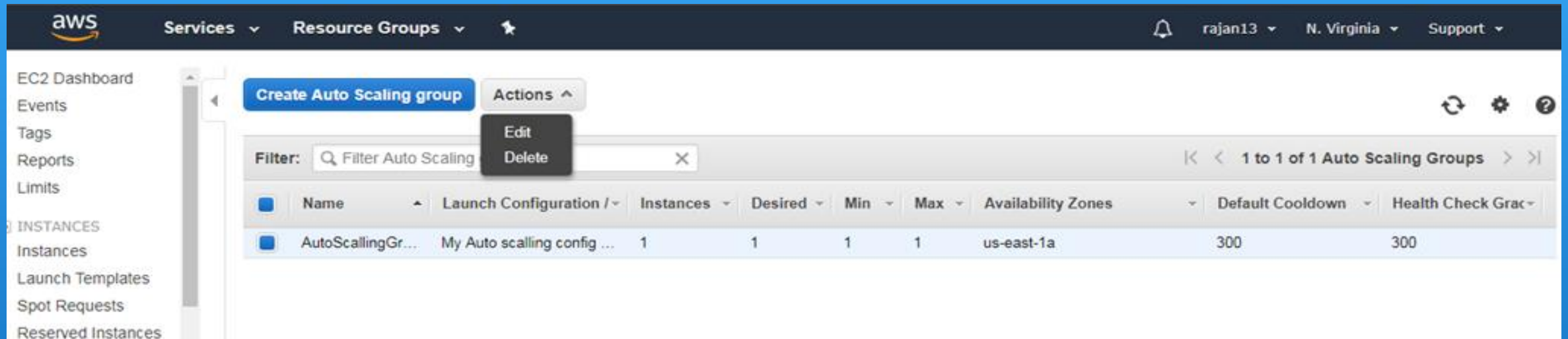
[Add notification](#)

# Hands on – Auto scaling(Conti...)

The screenshot displays the AWS Management Console interface for the EC2 instances page. The top navigation bar shows the AWS logo, 'Services', 'Resource Groups', and user information (rajan13, N. Virginia, Support). The left sidebar lists various AWS services under categories like 'NETWORK & SECURITY' and 'LOAD BALANCING'. The main content area features a table of EC2 instances with the following columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IPv4). The table contains four instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Linux WebServer	i-07f3f405587058c7c	t2.micro	us-east-1a	stopped		None	
Linux using AS	i-0932c1e21113066c9	t2.micro	us-east-1a	terminated		None	
Linux using AS new	i-0d1494187e0503b...	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-54-175-226-221.co...
Linux Ec2	i-0d4e6fe310ebb7b6f	t2.micro	us-east-1a	stopped		None	

# Hands on – Auto scaling(Conti...)

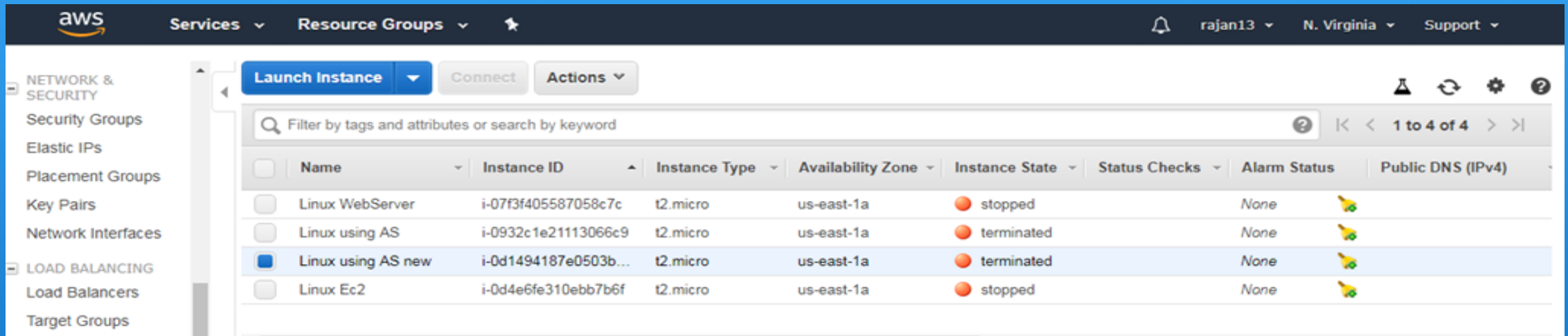


The screenshot shows the AWS Management Console interface for Auto Scaling Groups. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (rajan13, N. Virginia, Support). The left sidebar lists navigation options like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, Instances, Launch Templates, Spot Requests, and Reserved Instances.

The main content area displays the 'Auto Scaling Groups' page. A blue button 'Create Auto Scaling group' is visible. Below it, a filter bar shows 'Filter: Filter Auto Scaling'. A table lists the Auto Scaling Groups, with one group named 'AutoScalingGr...' having 1 instance. The 'Actions' menu is open, showing 'Edit' and 'Delete' options.

<input type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
<input checked="" type="checkbox"/>	AutoScalingGr...	My Auto scaling config ...	1	1	1	1	us-east-1a	300	300

# Hands on – Auto scaling(Conti...)



The screenshot displays the AWS Management Console interface. On the left, the navigation menu shows 'NETWORK & SECURITY' and 'LOAD BALANCING'. The main content area shows the 'Launch Instance' button and a table of EC2 instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IPv4). The instance 'Linux using AS new' is highlighted in blue.

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
<input type="checkbox"/>	Linux WebServer	i-07f3f405587058c7c	t2.micro	us-east-1a	stopped		None	
<input type="checkbox"/>	Linux using AS	i-0932c1e21113066c9	t2.micro	us-east-1a	terminated		None	
<input checked="" type="checkbox"/>	Linux using AS new	i-0d1494187e0503b...	t2.micro	us-east-1a	terminated		None	
<input type="checkbox"/>	Linux Ec2	i-0d4e6fe310ebb7b6f	t2.micro	us-east-1a	stopped		None	



# Introduction to CloudWatch

CloudWatch is an AWS service that can be used on the AWS cloud for monitoring various infrastructure and application resources running on your AWS cloud.

CloudWatch can be used to collect a number of metrics from the AWS resources.

It allows you to track these metrics and also initiate actions based on the threshold you set.

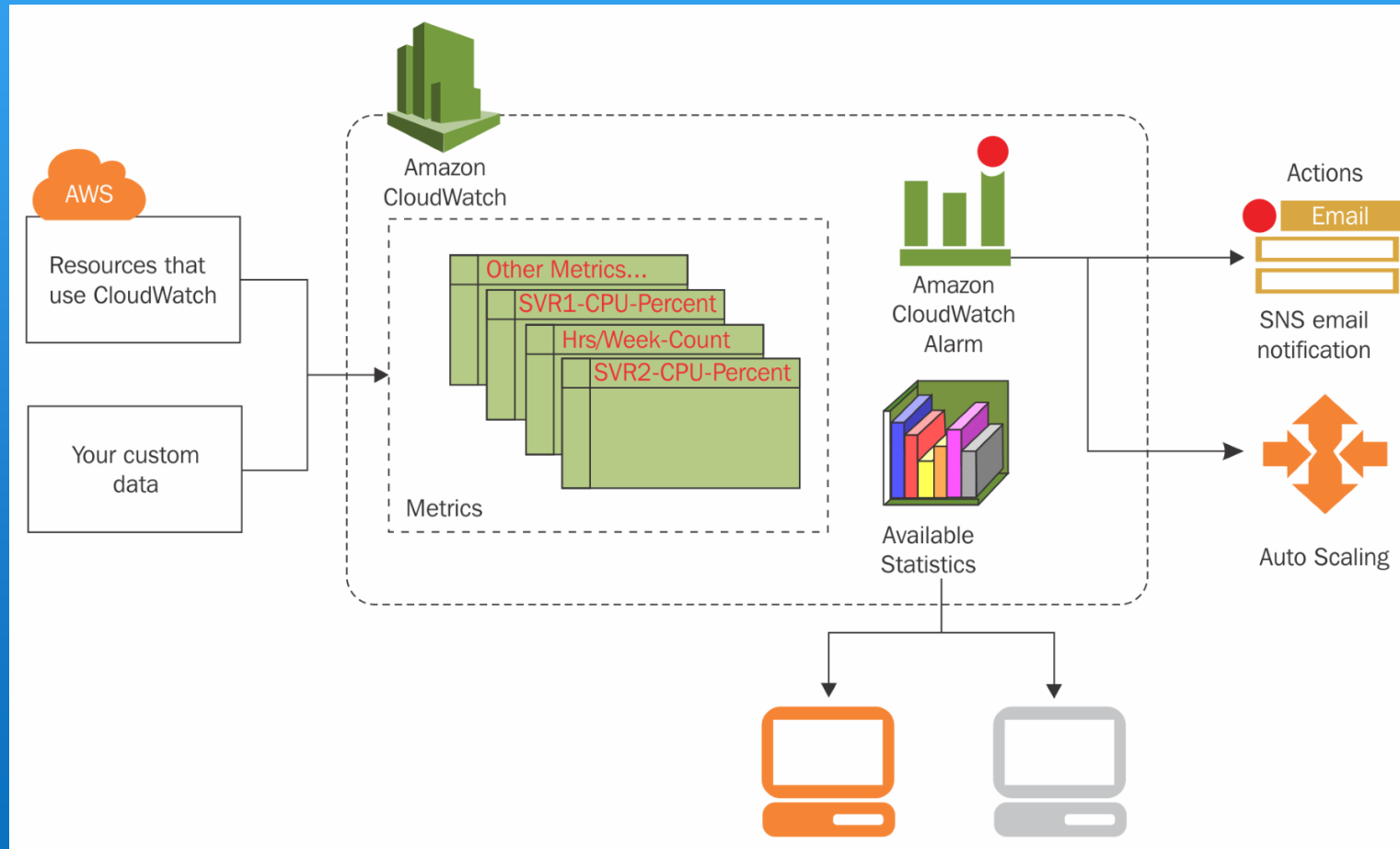
CloudWatch can also collect log files, generate metrics out of them, and help to monitor log files.

CloudWatch supports the monitoring of many AWS services such as EC2 instances, DynamoDB, RDS, and so on.

Amazon services like Auto Scaling uses CloudWatch alarms to automatically scale an environment up or down, based on the traffic on an environment.

# Working of CloudWatch

CloudWatch acts as a repository of metrics, by collating raw data from various AWS services or applications, converting it into metrics, statistics, graphs, and facilitates certain actions based on specific data points in metrics.



# Elements of CloudWatch

## Namespace

CloudWatch namespaces are containers in which metrics for different applications are stored. It is a mechanism to isolate metrics of different applications from each other.

## Metrics

Metrics are sets of data collected over a period of time with a specific time interval for quantitative assessment, measurement, and comparison of performance data generated by a specific application or a service.

For example, CPU utilization data for an EC2 instance is stored in a relevant CloudWatch metric at a time interval of one minute. Each AWS service stores several metrics in CloudWatch.

## Dimensions

A dimension in a CloudWatch metric is a mechanism to uniquely identify metrics. It is a name/value pair that is associated with metrics.

For example, CloudWatch stores metrics for EC2 instances in a namespace called AWS/EC2.

# Elements of CloudWatch(Conti...)

## Statistics

Statistics are a collection of aggregated metrics data for a specific period of time. Metrics data is aggregated using namespace, metric name, dimensions, and several data points in a given time period.

CloudWatch provides the statistics on the metrics data like: sum, average, sample count, minimum, maximum etc.

## Percentile

A percentile helps in finding the comparative standing of a value in a set of data.

# Elements of CloudWatch(Conti...)

## Alarms

CloudWatch alarms help in defining a threshold value that is constantly monitored, and an action is triggered when the threshold condition is breached.

For example, you can define a threshold of 80% CPU utilization on an EC2 instance and trigger an action whenever the CPU utilization is  $\geq 80$  for three consecutive periods.

An alarm can have **three possible states**:

Alarm status displays **OK** when the metric is within the defined threshold

Alarm status displays **ALARM** when the metric is outside of the defined threshold

Alarm status displays **INSUFFICIENT\_DATA** when the alarm is just configured, the metric is not available, or not enough data is available for the metric to determine the alarm state

# Hands on – Creating CloudWatch Alarm

The screenshot shows the AWS CloudWatch Alarms console. The top navigation bar includes the AWS logo, a 'Services' dropdown, 'Resource Groups', and icons for EC2, S3, IAM, and VPC. On the right of the top bar, there is a notification bell, the username 'rajan13', the region 'N. Virginia', and a 'Support' link. The left-hand navigation menu lists 'CloudWatch' (selected), 'Dashboards', 'Alarms', 'ALARM' (with a count of 0), 'INSUFFICIENT' (with a count of 0), 'OK' (with a count of 1), 'Billing', 'Events', 'Rules', 'Event Buses', 'Logs', 'Metrics', and 'Favorites'. The main content area is divided into two sections: 'Metric Summary' and 'Alarm Summary'. The 'Metric Summary' section contains a description of Amazon CloudWatch metrics, a 'Browse Metrics' button, and a 'Search Metrics' input field. The 'Alarm Summary' section features a yellow warning box with an exclamation mark icon, stating that one or more alarms have empty notification topics or unconfirmed subscriptions. Below the warning box, it states that all alarms are in an 'OK' state in the US East (N. Virginia) region. On the right side of the console, under 'Additional Info', there are links to the 'Getting Started Guide', 'Monitoring Scripts Guide', 'Overview and Features', 'Documentation', 'Forums', and 'Report an Issue'. A 'Create Alarm' button is located in the bottom right corner of the main content area.

**aws** Services ▾ Resource Groups ▾ EC2 S3 IAM VPC 🔔 rajan13 ▾ N. Virginia ▾ Support ▾

## CloudWatch

- Dashboards
- Alarms
  - ALARM 0
  - INSUFFICIENT 0
  - OK 1
- Billing
- Events
- Rules
- Event Buses
- Logs
- Metrics
- Favorites


### Metric Summary

Amazon CloudWatch monitors operational and performance metrics for your AWS cloud resources and applications. You currently have [290 CloudWatch metrics available](#) in the US East (N. Virginia) region.

Browse or search your metrics to get started graphing data and creating alarms.

[Browse Metrics](#)

### Alarm Summary

 You have one or more alarms with notification topics that are empty or have unconfirmed subscriptions. This means you may not receive a notification if these alarms trigger. Please go to the Alarms page and audit all affected alarms (those with an entry in the "Config Status" column).

All your alarms are [in OK](#) state in US East (N. Virginia) region.

[Create Alarm](#)

### Additional Info

- [Getting Started Guide](#)
- [Monitoring Scripts Guide](#)
- [Overview and Features](#)
- [Documentation](#)
- [Forums](#)
- [Report an Issue](#)

# Hands on – Creating CloudWatch Alarm

## Create Alarm

×

1. Select Metric

2. Define Alarm

Browse Metrics

Search Metrics

×

### CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: 290

<b>Billing Metrics : 11</b> Total Estimated Charge : 1 By Service : 10	<b>EBS Metrics : 63</b> Per-Volume Metrics : 63	<b>EC2 Metrics : 119</b> Per-Instance Metrics : 103 By Auto Scaling Group : 16
<b>ELB Metrics : 89</b> Per-LB Metrics : 20 Per LB, per AZ Metrics : 26 By Availability Zone : 19 Across All LBs : 8 By Namespace : 8 By Service : 8	<b>S3 Metrics : 2</b> Storage Metrics : 2	<b>SNS Metrics : 6</b> Metrics with no dimensions : 1 Country,SMSType : 1 Topic Metrics : 4

# Hands on – Creating CloudWatch Alarm

## Create Alarm

1. Select Metric

2. Define Alarm

EC2

Search Metrics

1 to 50 of 103 metrics

<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	DiskWriteBytes
<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	DiskWriteOps
<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	NetworkIn
<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	NetworkOut
<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	NetworkPacketsIn
<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	NetworkPacketsOut
<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	StatusCheckFailed
<input checked="" type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	StatusCheckFailed_Instance
<input type="checkbox"/>	i-07f3f405587058c7c	Linux WebServer	StatusCheckFailed_System
<input type="checkbox"/>	i-0932c1e21113066c9		CPUCreditBalance
<input type="checkbox"/>	i-0932c1e21113066c9		CPUCreditUsage

Title: StatusCheckFailed\_Instance

Average

5 Minutes

Update Graph

Time Range

Relative Absolute UTC (GMT)

From: 12 hours ago

1.00

0.5

0

16:30 17:30 18:30 19:30 20:30 21:30 22:30 23:30 00:30 01:30 02:30 03:30



# Hands on – Creating CloudWatch Alarm

## Create Alarm

1. Select Metric 2. Define Alarm

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**Name:**

**Description:**

**Whenever:** StatusCheckFailed\_Instance

**is:** >=

**for:** 1 out of  datapoints

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for 1 datapoints within 15 minutes

StatusCheckFailed\_Instance >= 1 for 3 datapoints wi...

Time	Value
9/06 01:00	1.0
9/06 02:00	1.0
9/06 03:00	1.0

### Additional settings

Provide additional configuration for your alarm.

**Treat missing data as:** missing

### Actions

Define what actions are taken when your alarm changes state.

**Namespace:** AWS/EC2

**InstanceId:**

**InstanceName:** Linux WebServer

**Metric Name:**

**Period:** 5 Minutes

**Statistic:** ☒ Standard ☐ Custom

# Hands on – Creating CloudWatch Alarm

## Create Alarm

1. Select Metric

2. Define Alarm

Provide additional configuration for your alarm.

Treat missing data as: missing

InstanceName: Linux WebServer

Metric Name: StatusCheckFailed\_Inst

Period: 5 Minutes

Statistic: ☒ Standard ☐ Custom

Average

### Actions

Define what actions are taken when your alarm changes state.

Notification

Delete

Whenever this alarm: State is ALARM

Send notification to: Select a notification list

New list Enter list

EC2 Action

Delete

Whenever this alarm: State is ALARM

Take this action:

☒ Recover this instance

☐ Stop this instance

☐ Terminate this instance

☐ Reboot this instance

This will auto recover your EC2 instance (i-07f3f405587058c7c).  
You can only recover certain EC2 instance types. [Please see documentation.](#)

# CloudWatch dashboard

Amazon CloudWatch provides a customizable dashboard inside a web console. It can display a set of critical metrics together.

You can create multiple dashboards where each dashboard can focus on providing a distinct view of your environment.

You can create a custom dashboard to view and monitor the selected AWS resources from the same or different regions.

# Monitoring types : Basic and Detailed

## Basic monitoring

Basic monitoring is free and it collects data at a five-minute time interval.

By default, when you provision AWS resources, all AWS resources except ELB and RDS start with a basic monitoring mode only.

ELB and RDS monitors the resources at a one-minute interval.

For other resources, optionally, you can switch the monitoring mode to detailed monitoring.

## Detailed monitoring

Detailed monitoring is chargeable and it makes data available at a one-minute time interval. Currently, AWS charges \$0.015 per hour, per instance.

Detailed monitoring does not change the monitoring on ELB and RDS which by default collates data at a one-minute interval.

Similarly, detailed monitoring does not change the EBS volumes which are monitored at five-minute intervals.

# Summary

In this chapter, we have gone through following topics:

- Introduction to Auto scaling
- Why auto scaling
- Advantages of auto scaling
- Auto scaling components
  - 1. Launch configuration
  - 2. Auto scaling group
- Scale Up vs. Scale Out
- Hands on - Auto scaling
- Introduction to Cloudwatch
- Working of Cloudwatch
- Elements of Cloudwatch
- Hands on – Creating Cloudwatch Alarm
- Cloudwatch dashboard
- Monitoring types – Basic and Detailed

See you soon...

*Thank You!*