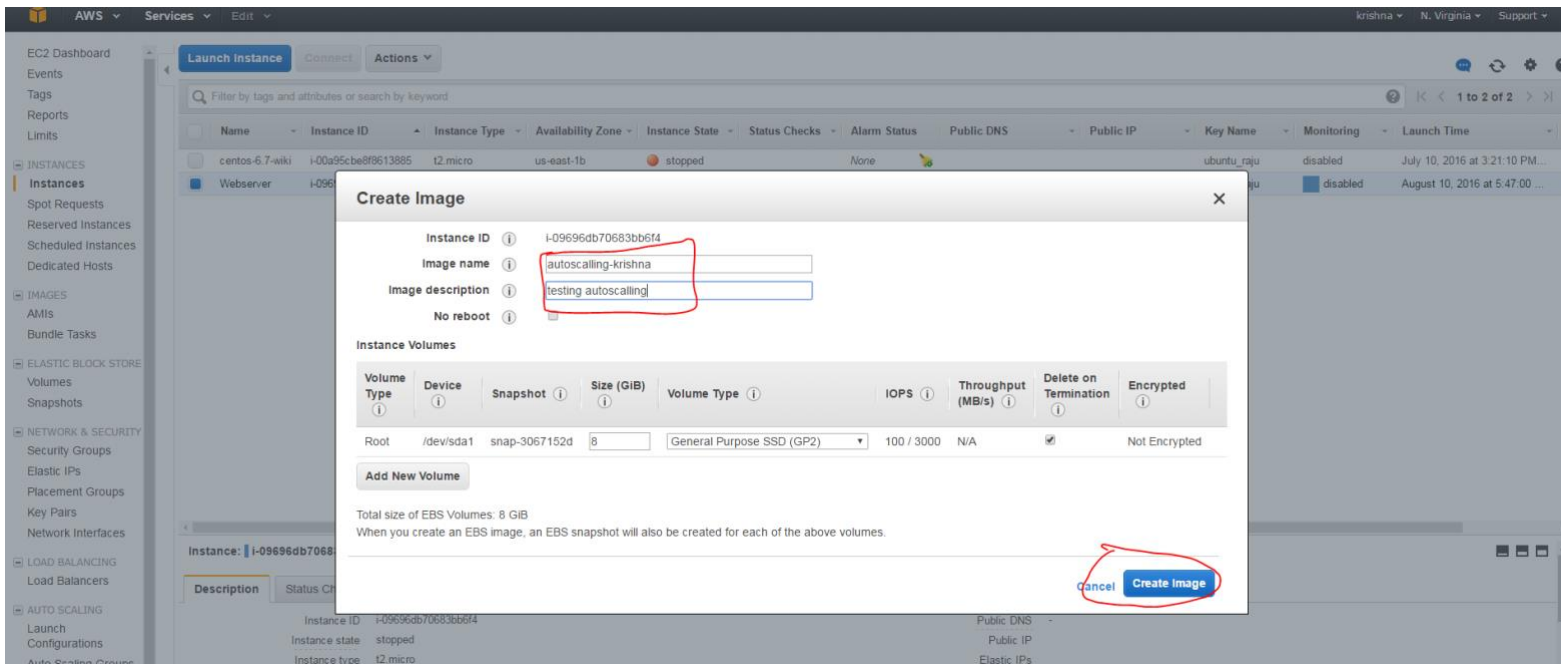
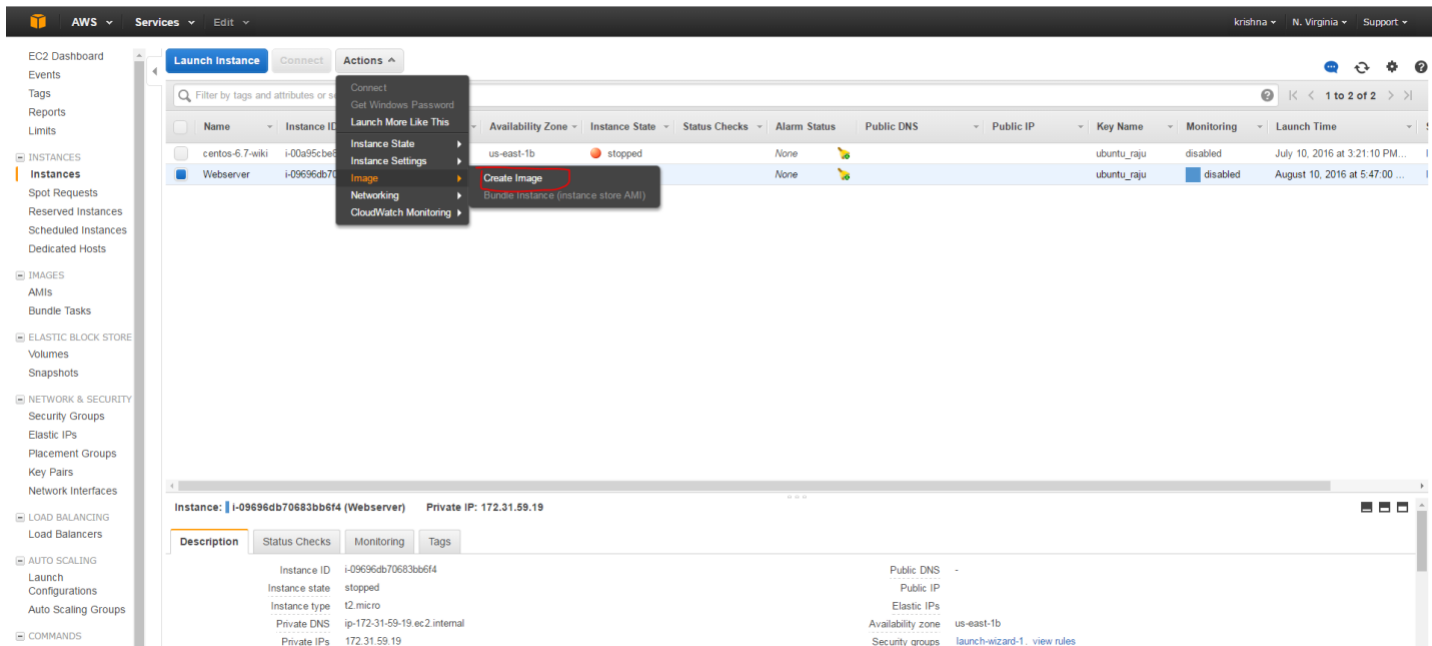
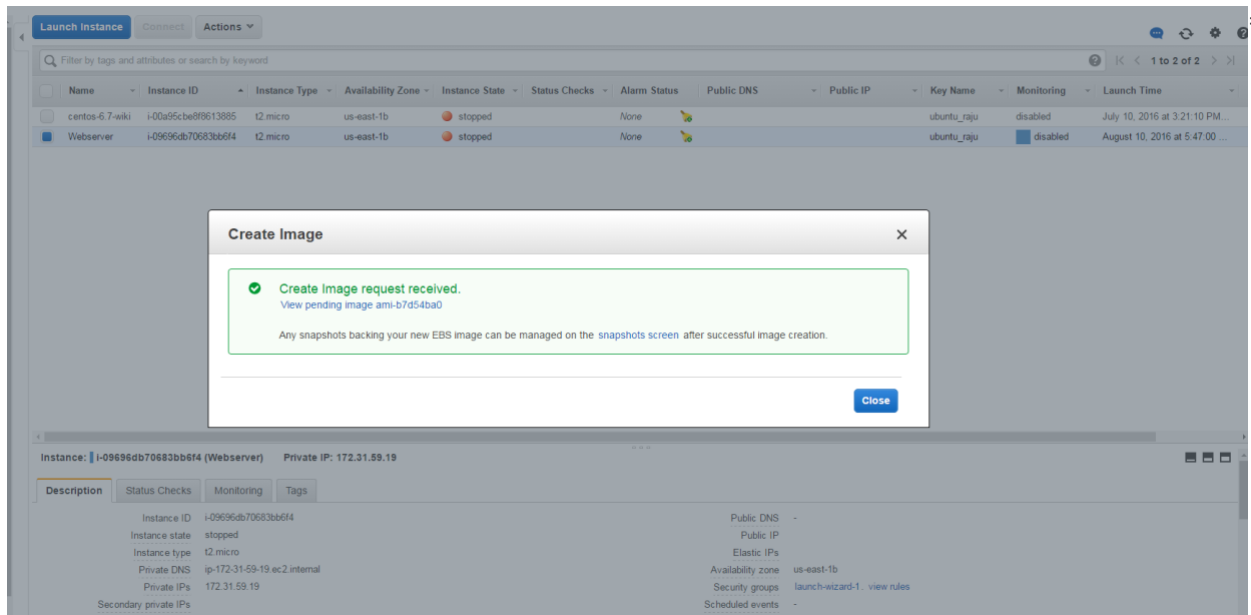


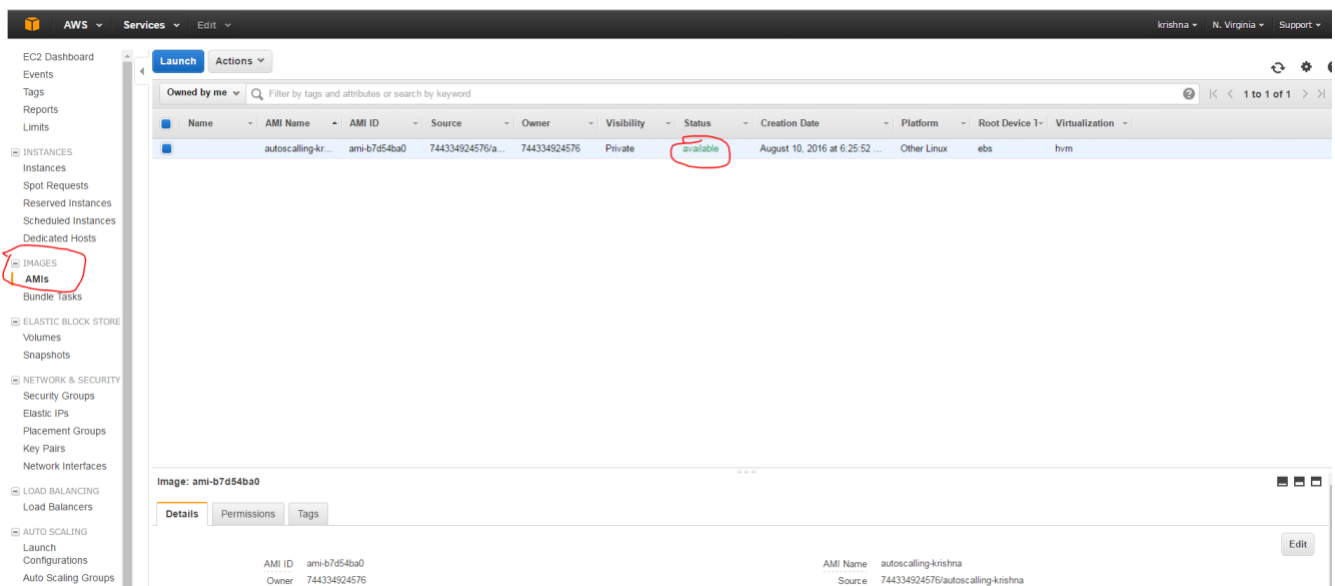
AWS Auto Scaling for ELB

Step 1 : Create AMI





Check AMI status



* Create Loadbalancer

AWS

Services

Edit

krishnaN. VirginiaSupport

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Scheduled Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

LOAD BALANCING

Load Balancers

Target Groups

AUTO SCALING

Create Load Balancer

Actions

Filter: Search

You do not have any of these resources in this region.

Select a load balancer

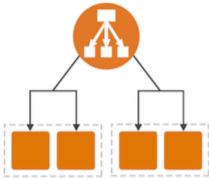
Welcome to Elastic Load Balancing

Select load balancer type

Elastic Load Balancing supports two types of load balancers: Application Load Balancers (new) and Classic Load Balancers. Choose the load balancer type that meets your needs.

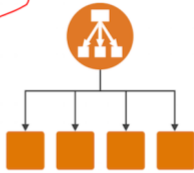
Learn more

Application Load Balancer



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

Classic Load Balancer



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

Cancel

Continue

Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name:

krishna-auto-load-balancer

Create LB inside:

My Default VPC (172-31-0-0/16)

Create an internal load balancer:

☐

What's this?

Enable advanced VPC configuration:

☐

Listener Configuration

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	
HTTP	80	HTTP	80	✕
Add				

Cancel

Next: Assign Security Groups

* If you have any specification change it accordingly , otherwise make it same as default

Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

Ping Protocol

Ping Port

Ping Path

Advanced Details

Response Timeout seconds

Interval seconds

Unhealthy threshold

Healthy threshold

[Cancel](#) [Previous](#) [Next: Add EC2 Instances](#)

* No need to add any Instances as of now , it will add automatically once configuration done.

Step 5: Add EC2 Instances

The table below lists all your running EC2 Instances. Check the boxes in the Select column to add those instances to this load balancer.

VPC vpc-f9bfa49d (172.31.0.0/16)

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input type="checkbox"/>	i-089ca583...	stopped	launch-wizard-3	us-east-1b	subnet-5836a372	172.31.48.0/20
<input type="checkbox"/>	i-0076a3c8...	stopped	launch-wizard-9	us-east-1b	subnet-5836a372	172.31.48.0/20

Availability Zone Distribution

☒ Enable Cross-Zone Load Balancing

☒ Enable Connection Draining seconds

[Cancel](#) [Previous](#) [Next: Add Tags](#)

* Mentioned tag name and create a load balancer

AWS Services Edit

krishna N. Virginia Support

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

Step 6: Add Tags

Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more about tagging your Amazon EC2 resources.](#)

Key	Value
Name	krishna-load-auto

Create Tag

Cancel Previous **Review and Create**

* After create a load balancer , your ELB dashboard looks like below.

AWS Services Edit

krishna N. Virginia Support

EC2 Dashboard

- Events
- Tags
- Reports
- Limits
- INSTANCES
 - Instances
 - Spot Requests
 - Reserved Instances
 - Scheduled Instances
 - Dedicated Hosts
- IMAGES
 - AMIs
 - Bundle Tasks
- ELASTIC BLOCK STORE
 - Volumes
 - Snapshots
- NETWORK & SECURITY
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs
 - Network Interfaces
- LOAD BALANCING**
 - Load Balancers**
 - Target Groups
- AUTO SCALING
 - Launch Configurations
 - Auto Scaling Groups
- COMMANDS
 - Command History
 - Resource Groups

Create Load Balancer Actions

Filter: Search

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
krishna-auto-load-balancer	krishna-auto-load-balancer-1...		vpc-45bfa49d	us-east-1b, us-east-1c...	classic	August 24, 2016 at 12:45:01...	

Load balancer: krishna-auto-load-balancer

Description Instances Health Check Listeners Monitoring Tags

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
There are no instances registered to this load balancer.				

* Create Autoscaling group and Launch Configuration

Welcome to Auto Scaling

You can use Auto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, operate a healthy group of instances, and scale it according to your needs.

[Learn more](#)

[Create Auto Scaling group](#)

Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.

Benefits of Auto Scaling

- Reusable Instance Templates**
Provision instances based on a reusable template you define, called a launch configuration.
[Learn more](#)
- Automated Provisioning**
Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000.
[Learn more](#)
- Adjustable Capacity**
Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics.
[Learn more](#)

Additional Information

- [Getting Started Guide](#)
- [Documentation](#)
- [All EC2 Resources](#)
- [Forums](#)
- [Pricing](#)
- [Contact Us](#)

Left Sidebar:

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits
- INSTANCES
 - Instances
 - Spot Requests
 - Reserved Instances
 - Scheduled Instances
 - Dedicated Hosts
- IMAGES
 - AMIs
 - Bundle Tasks
- ELASTIC BLOCK STORE
 - Volumes
 - Snapshots
- NETWORK & SECURITY
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs
 - Network Interfaces
- LOAD BALANCING
 - Load Balancers
- AUTO SCALING
 - Launch Configurations
 - Auto Scaling Groups**
- CONFIGURATIONS

Create Auto Scaling Group

[Cancel and Exit](#)

To create an Auto Scaling group, you will first need to choose a template that your Auto Scaling group will use when it launches instances for you, called a launch configuration. Choose a launch configuration or create a new one, and then apply it to your group.

Later, if you want to use a different template, you can create another launch configuration and apply it to this group, even if you already have instances running in it. Using this method, you can update the software that your group uses when it launches new instances.

Step 1: Create launch configuration

First, define a template that your Auto Scaling group will use to launch instances. You can change your group's launch configuration at any time.

Step 2: Create Auto Scaling group

Next, give your group a name and specify how many instances you want to run in it. Your group will maintain this number of instances, and replace any that become unhealthy or impaired. You can optionally configure your group to adjust in capacity according to demand, in response to Amazon CloudWatch metrics.

[Cancel](#) [Create launch configuration](#)

* Click on my ami **SELECT** it.

AWS Services Edit

krishna 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

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

Search my AMIs

1 to 1 of 1 AMIs

autoscalling-krishna - ami-b7d54ba0

testing autoscalling

Root device type: ebs Virtualization type: hvm Owner: 744334924576

Select

64-bit

Ownership

- ☒ Owned by me
- ☐ Shared with me

Architecture

- ☐ 32-bit
- ☐ 64-bit

Root device type

- ☐ EBS
- ☐ Instance store

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 (GiB)	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	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate
<input type="checkbox"/>	General purpose	m4.xlarge	4	16	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
<input type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High

Cancel Previous **Next: Configure details**

AWS Services Edit

krishna N. Virginia Support

1. Choose AMI2. Choose Instance Type3. Configure details4. Add Storage5. Configure Security Group6. Review

Create Launch Configuration

Name ⓘKrishna-Autoscaling

Purchasing option ⓘ☐ Request Spot Instances

IAM role ⓘNone

Monitoring ⓘ☐ Enable CloudWatch detailed monitoring
[Learn more](#)

Advanced Details

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.

[Cancel](#) [Previous](#) [Skip to review](#) [Next: Add Storage](#)

AWS Services Edit

krishna N. Virginia Support

1. Choose AMI2. Choose Instance Type3. Configure details4. Add Storage5. Configure Security Group6. 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/sda1	snap-779f7eea	8	General Purpose (SSD)	24 / 3000	N/A	<input checked="" type="checkbox"/>	No

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Skip to review](#) [Next: Configure Security Group](#)

AWS

Services

Edit

krishnaN. VirginiaSupport

1. Choose AMI2. Choose Instance Type3. Configure details4. Add Storage5. Configure Security Group6. 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:AutoScaling-Security-Group-1

Description:AutoScaling-Security-Group-1 (2016-08-10 18:32:57.040+05:30)

Type	Protocol	Port Range	Source	
SSH	TCP	22	Anywhere0.0.0.0/0	
HTTP	TCP	80	Anywhere0.0.0.0/0	

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

CancelPreviousReview

AWS

Services

Edit

krishnaN. VirginiaSupport

1. Choose AMI2. Choose Instance Type3. Configure details4. Add Storage5. Configure Security Group6. Review

Create Launch Configuration

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

autoscalling-krishna - ami-b7d54ba0

testing autoscalling

Root device type: ebsVirtualization Type: hvm

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

NameKrishna-Autoscaling

Purchasing optionOn demand

EBS OptimizedNo

MonitoringNo

IAM roleNone

TenancyShared tenancy (multi-tenant hardware)

Kernel IDUse default

RAM Disk IDUse default

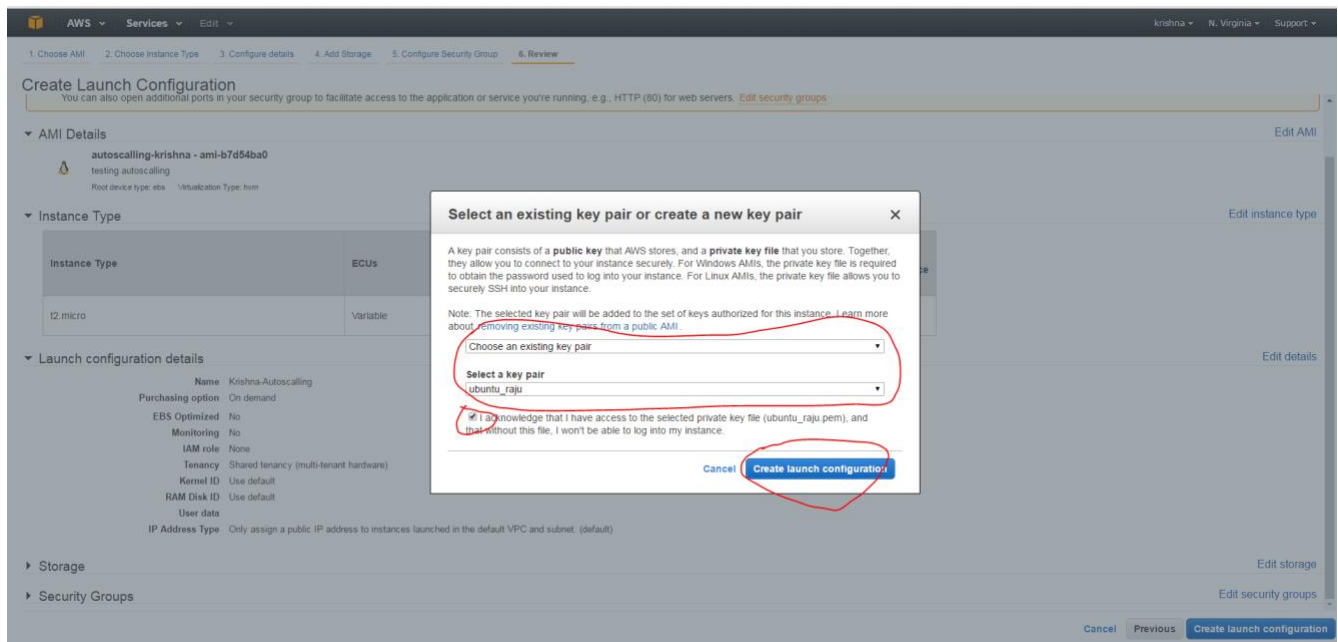
User data

IP Address TypeOnly assign a public IP address to instances launched in the default VPC and subnet. (default)

Storage

Security Groups

CancelPreviousCreate launch configuration



* Now create Auto Scaling Group

- Enter the auto scaling group name
- Enter Group size as 1 (It will launch one instance first, later it will add based on scalling)
- Select default Network
- Select all subnets for high availability
- Click on Advanced Details
- Click on Loadbalancer check box(Receive traffic from one or more load balancers)
- Click on **Classic Load Balancers** (automatically detect our ELB)
- Then click on '**Health Check Type** ' as ELB
- Then press Configure Secure policy.

Create Auto Scaling Group

Launch Configuration: krishna-test-auto-load

Group name: krishna-auto-load-balancer

Group size: Start with 1 instances

Network: vpc-95bfa49d (172.31.0.0/16) (default)

Subnet: subnet-2d1d5a5b (172.31.0.0/20) | Default in us-east-1c
 subnet-9657cac3 (172.31.16.0/20) | Default in us-east-1d
 subnet-10064d24 (172.31.32.0/20) | Default in us-east-1e
 subnet-5836a372 (172.31.48.0/20) | Default in us-east-1f

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

Advanced Details

Load Balancing: ☒ Receive traffic from one or more load balancers

Classic Load Balancers: krishna-auto-load-balancer

Target Groups:

Health Check Type: ☒ ELB ☐ EC2

Health Check Grace Period: 300 seconds

Monitoring: Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration krishna-test-auto-load. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency.

Instance Protection:

Cancel Next: Configure scaling policies

- Select the “Use scaling policies to adjust the capacity of this group “
- Select between 1 and 4 (so minimum 1 instance maximum 4 instances will launch)
- Create a policy for “Execute policy when “ (Now I create policy when the cpu usage reached morethan or equal to 30 % in 1 minutes then add one instance).
- In Take the action field , mentioned 1 (when ever above policy or alarm reached it will add one instances , if you need to add two instances when the policy or alarm reached thresold mentioned 2.
- Like in the same way create policy for Decrese group size also.

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

Scale between 1 and 4 instances. These will be the minimum and maximum size of your group.

Increase Group Size

Name: Increase Group Size

Execute policy when: ~~aws:ec2:krishna-auto-load-balancer-CPU-Utilization~~ Edit Remove
 breaches the alarm threshold: CPUUtilization >= 30 for 60 seconds
 or the metric dimensions AutoScalingGroupName = krishna-auto-load-balancer

Take the action: Add 1 instances when 30 <= CPUUtilization < infinity

Instances need: 300 seconds to warm up after each step

Decrease Group Size

Name: Decrease Group Size

Execute policy when: ~~aws:ec2:krishna-auto-load-balancer-High-CPU-Utilization~~ Edit Remove
 breaches the alarm threshold: CPUUtilization < 15 for 60 seconds
 or the metric dimensions AutoScalingGroupName = krishna-auto-load-balancer

Take the action: Remove 1 instances when 15 >= CPUUtilization > infinity

Cancel Previous Review Next: Configure Notifications

* Click on “Add notification “ option and add it and create a tag for it then create

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.

Send a notification to: create topic

Whenever instances:

- ☒ launch
- ☒ terminate
- ☒ fail to launch
- ☒ fail to terminate

Add notification

[Cancel](#) [Previous](#) [Review](#) **Next: Configure Tags**

* Once you launched auto scaling group , Click on Instances , New instance will create

Instances

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
Doelker	i-0076e3c8086c09...	t2.micro	us-east-1b	stopped	Initializing	None	ec2-52-87-155-14.co...	52.87.155.14	ubuntu_raju	disabled	August 15, 20...
nagios-ansible-code	i-029e16d073a5ee...	t2.micro	us-east-1e	pending		None			ubuntu_raju	disabled	August 24, 20...
krishna-load-test1	i-089ca583e49a3...	t2.micro	us-east-1b	stopped		None			ubuntu_raju	disabled	August 20, 20...
	i-0c67f41b7bc910...	t2.micro	us-east-1c	terminated		None			ubuntu_raju	disabled	August 24, 20...
	i-0c68704aaca4a6...	t2.micro	us-east-1b	terminated		None			ubuntu_raju	disabled	August 24, 20...

* Once it is running then click on Load balancer , it will automatically add this instance to ELB.

Instances

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
Doelker	i-0076e3c8086c09...	t2.micro	us-east-1b	stopped		None			ubuntu_raju	disabled	August 15...
nagios-ansible-code	i-089ca583e49a3...	t2.micro	us-east-1b	stopped		None			ubuntu_raju	disabled	August 20...
krishna-load-test1	i-0c67f41b7bc910...	t2.micro	us-east-1c	terminated		None			ubuntu_raju	disabled	August 24...
krishna-load-test1	i-029e16d073a5ee...	t2.micro	us-east-1e	running	Initializing	None	ec2-52-87-155-14.co...	52.87.155.14	ubuntu_raju	disabled	August 24...

The screenshot shows the AWS Management Console interface for Load Balancers. The left sidebar contains a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, and LOAD BALANCING. The 'Load Balancers' option under 'LOAD BALANCING' is highlighted with a red circle. The main panel shows a list of load balancers with columns: Name, DNS name, State, VPC ID, Availability Zones, Type, Created At, and Monitoring. The 'krishna-auto-load-balancer' is selected and its name is circled in red. Below the list, the 'Load balancer: krishna-auto-load-balancer' section is visible, with tabs for Description, Instances, Health Check, Listeners, Monitoring, and Tags. The 'Instances' tab is active and its label is circled in red. It shows 'Connection Draining: Enabled, 300 seconds (Edit)' and an 'Edit Instances' button. Below this is a table of instances with columns: Instance ID, Name, Availability Zone, Status, and Actions. The first instance, 'i-02fe16d073a5eea20' with name 'krishna-auto-load-test1', is circled in red.

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
krishna-auto-load-balancer	krishna-auto-load-balancer...		vpc-f9bfa49d	us-east-1b, us-east-1c...	classic	August 24, 2016 at 12:45:0...	

Instance ID	Name	Availability Zone	Status	Actions
i-02fe16d073a5eea20	krishna-auto-load-test1	us-east-1e	InService ⓘ	Remove from Load Balancer

Testing :

* For testing purpose , I will login to this server and increase the cpu load , so based on our scenario , if cpu load increase to more than 30 % ,it will add instance as well as it will attach new instance to ELB.

* Login to server and executed command

```

root@ip-172-31-35-166:~# stress -c 4
stress: info: [1679] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd

```

* Let's wait for 10 minutes

* After few minutes , Instances dashboard shows like this , it will automatically adding instance because we increase CPU load intensinally , based on our autoscaling policy , it will add new instance and attach to ELB. Like in the same way , it will add instance upto 4 , because we mentioned max 4 instances in autoscaling group configuration.

Name	Instance ID	Instance Type	Availability Zone	State	Status Checks	Public DNS	Public IP	Key Name	Monitoring
krishna-load-auto-instance	i-0c67f41b7bc910...	t2.micro	us-east-1c	terminated				ubuntu_raju	disabled
krishna-load-auto-instance	i-0c68704aac44a5...	t2.micro	us-east-1b	terminated				ubuntu_raju	disabled
krishna-auto-load-test1	i-02b16d073a5ee...	t2.micro	us-east-1e	running	2/2 checks passed	ec2-52-87-155-14.co...	52.87.155.14	ubuntu_raju	disabled
krishna-auto-load-test1	i-042d56db064a2248c	t2.micro	us-east-1d	initializing		ec2-54-181-158-163.c...	54.181.158.163	ubuntu_raju	disabled

Load balancer: krishna-auto-load-balancer

Connection Draining: Enabled, 300 seconds (Edit)

Instance ID	Name	Availability Zone	Status	Actions
i-02b16d073a5ee...	krishna-auto-load-test1	us-east-1e	InService	Remove from Load Balancer
i-042d56db064a2248c	krishna-auto-load-test1	us-east-1d	InService	Remove from Load Balancer

* Now stop the script so that CPU load comes back to normal , so based our scenario , Autoscaling group will automatically removes the instances from ELB and terminate it.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
Docker	i-007fe3c8096c0958a	t2.micro	us-east-1b	stopped		None			ubuntu_raju	disabled
nagios-anabile-code	i-089ca583e49a37cd	t2.micro	us-east-1b	stopped		None			ubuntu_raju	disabled
krishna-auto-load-test1	i-01a963b7d82aeb62	t2.micro	us-east-1c	running	2/2 checks passed	None	ec2-52-90-231-127.co...	52.90.231.127	ubuntu_raju	disabled
krishna-auto-load-test1	i-02b16d073a5ee...	t2.micro	us-east-1e	terminated		None			ubuntu_raju	disabled
krishna-auto-load-test1	i-042d56db064a2248c	t2.micro	us-east-1d	terminated		None			ubuntu_raju	disabled

* From ELB also it will remove automatically.

The screenshot shows the AWS Management Console interface for Load Balancers. The left sidebar contains navigation links for various AWS services, with 'Load Balancers' highlighted. The main content area displays a list of load balancers, with 'krishna-auto-load-balancer' selected. Below this, the 'Instances' tab is active, showing a table of instances associated with the selected load balancer.

Instance ID	Name	Availability Zone	Status	Actions
i-01a983b7f082aebb2	krishna-auto-load-test1	us-east-1c	InService	Remove from Load Balancer