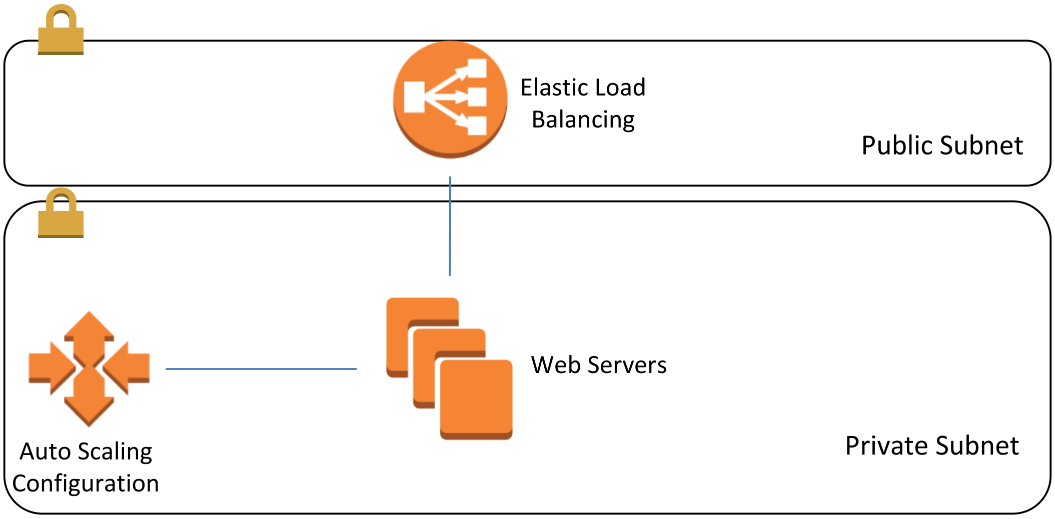
Introduction

In this lab you will apply some concepts as discussed before. We will create a scalable web server setup, as pictured below:

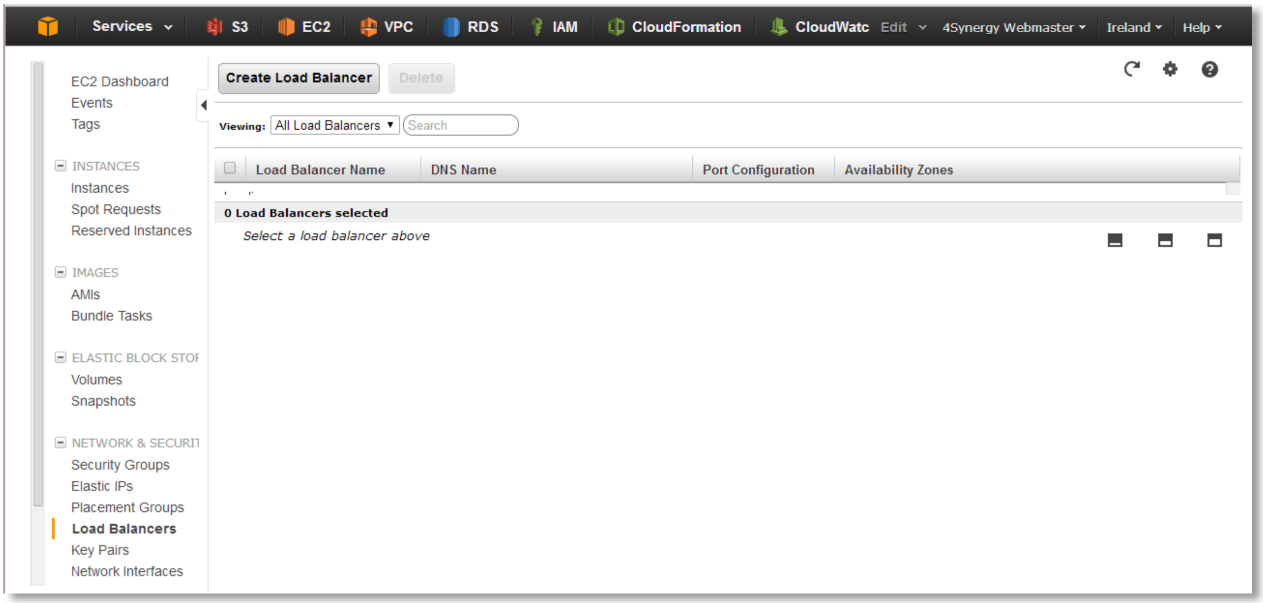


First we will create an elastic load balancer, after that we will configure the auto scaling group that will launch our web servers.

Create an ELB

In case you have multiple (web) servers, you need a load balancer in front of these servers to give your clients a single location for accessing these servers and to balance user requests across your server farm.

In the **EC2 console**, click on the **Load Balancers** link, and click on **Create Load Balancer** button.



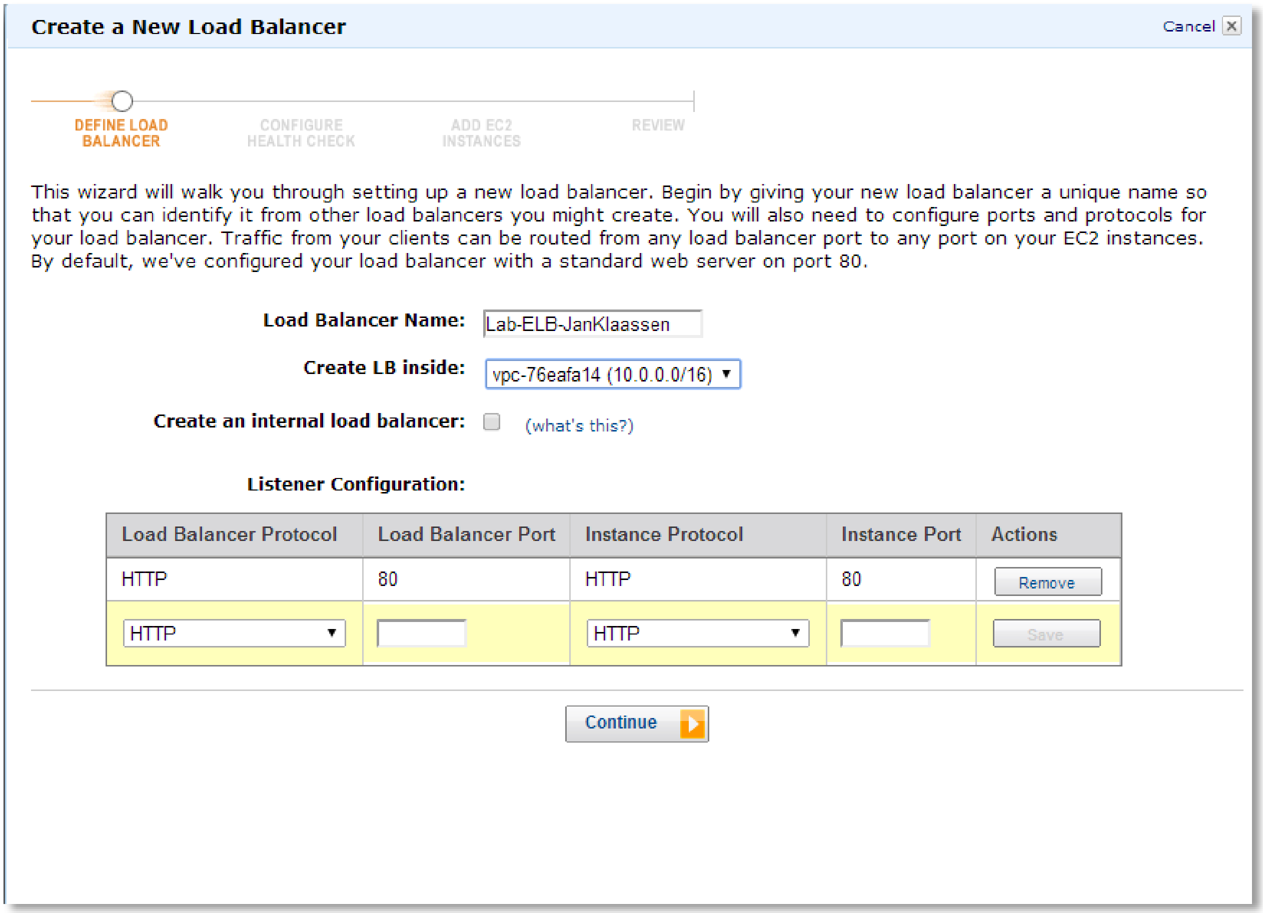
□	Load Balancer Name	DNS Name	Port Configuration	Availability Zones

0 Load Balancers selected

Select a load balancer above

We will just be creating a simple HTTP load balancer, so give your ELB a new name like **LabELB** and accept the default listener configuration.

It is important to understand the **Create LB inside** option. The (classic) EC2 option is basically one large network shared by all AWS customers, having a public network (i.e. the internet) and a private network. In this option, systems are primarily protected by security groups (think firewalls). The Virtual Private Cloud (VPC) is a virtual private network that you can configure to your own liking. This VPC configuration is getting the default and when you create a new AWS account by default you get a VPC only configuration.



Create a New Load Balancer

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: Lab-ELB-JanKlaassen

Create LB inside: vpc-76eafa14 (10.0.0.0/16) ▾

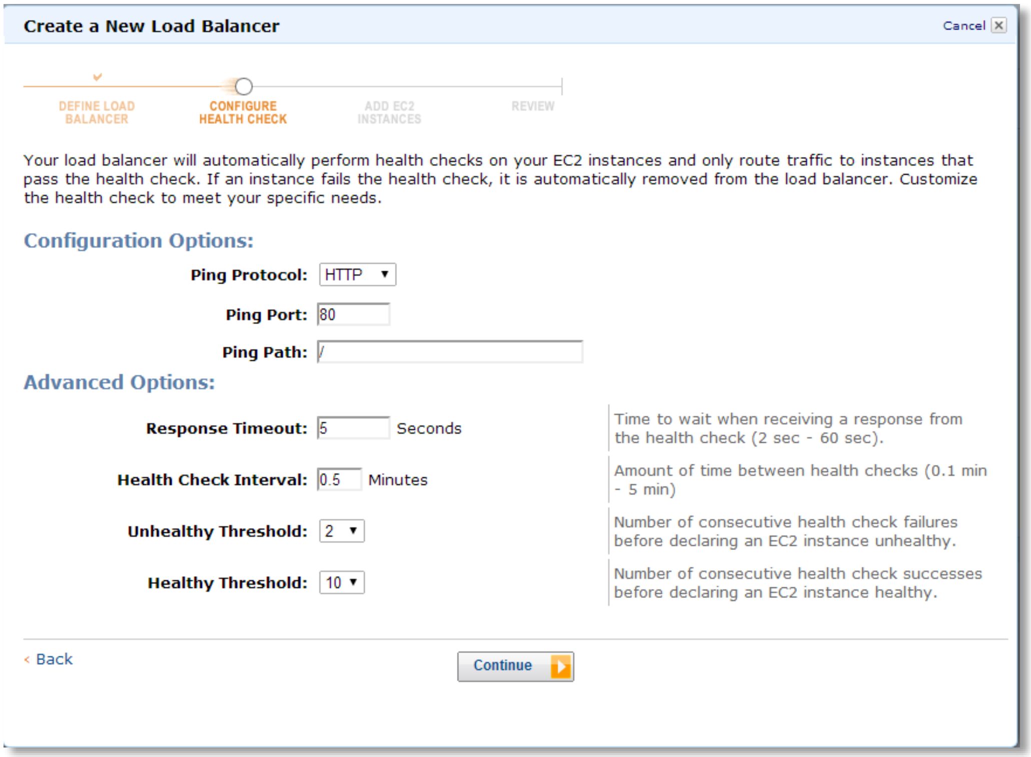
Create an internal load balancer: □ (what's this?)

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Actions
HTTP	80	HTTP	80	Remove
HTTP ▾		HTTP ▾		Save

For this lab, we assume a VPC setup, so select to create the ELB in the VPC, in its public subnet and click **Continue**.

On the next screen change **Ping Path** to **/** (delete index.html) and accept the advanced options. Note that these options can be changed in the future, and configure how the ELB Health Check will be performed including the health check protocol, port, and path as well as the health check interval, timeout, and heath thresholds.



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.

Configuration Options:

Ping Protocol: HTTP ▾

Ping Port: 80

Ping Path: /

Advanced Options:

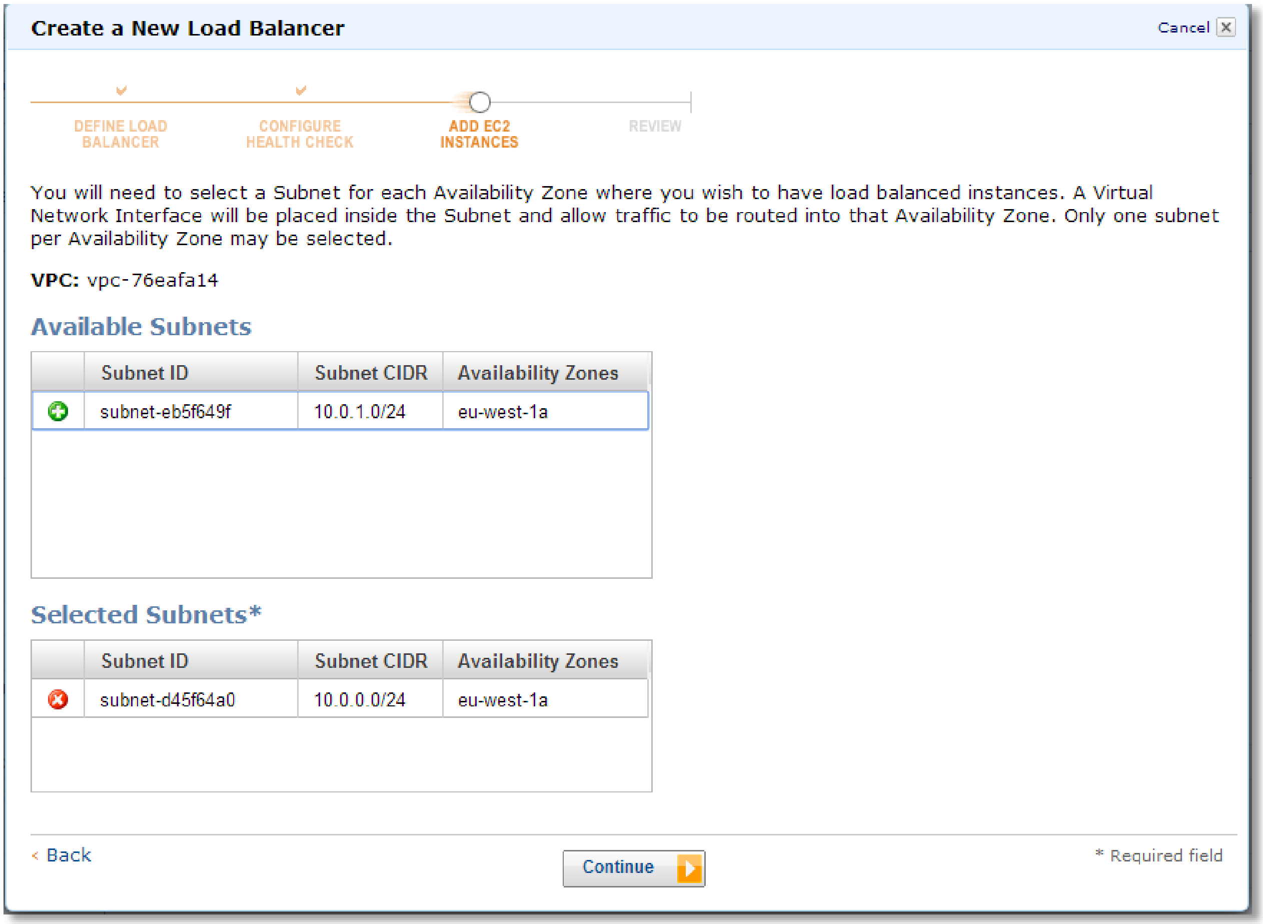
Response Timeout: 5 Seconds — Time to wait when receiving a response from the health check (2 sec - 60 sec).

Health Check Interval: 0.5 Minutes — Amount of time between health checks (0.1 min - 5 min)

Unhealthy Threshold: 2 ▾ — Number of consecutive health check failures before declaring an EC2 instance unhealthy.

Healthy Threshold: 10 ▾ — Number of consecutive health check successes before declaring an EC2 instance healthy.

In the following screen, you need to select the subnet in which your instances will live. Typically, a VPC consists of at least a public zone (consisting of one or more subnets) and a private zone, and your instances are then placed in the private zone.



Create a New Load Balancer

You will need to select a Subnet for each Availability Zone where you wish to have load balanced instances. A Virtual Network Interface will be placed inside the Subnet and allow traffic to be routed into that Availability Zone. Only one subnet per Availability Zone may be selected.

VPC: vpc-76eafa14

Available Subnets

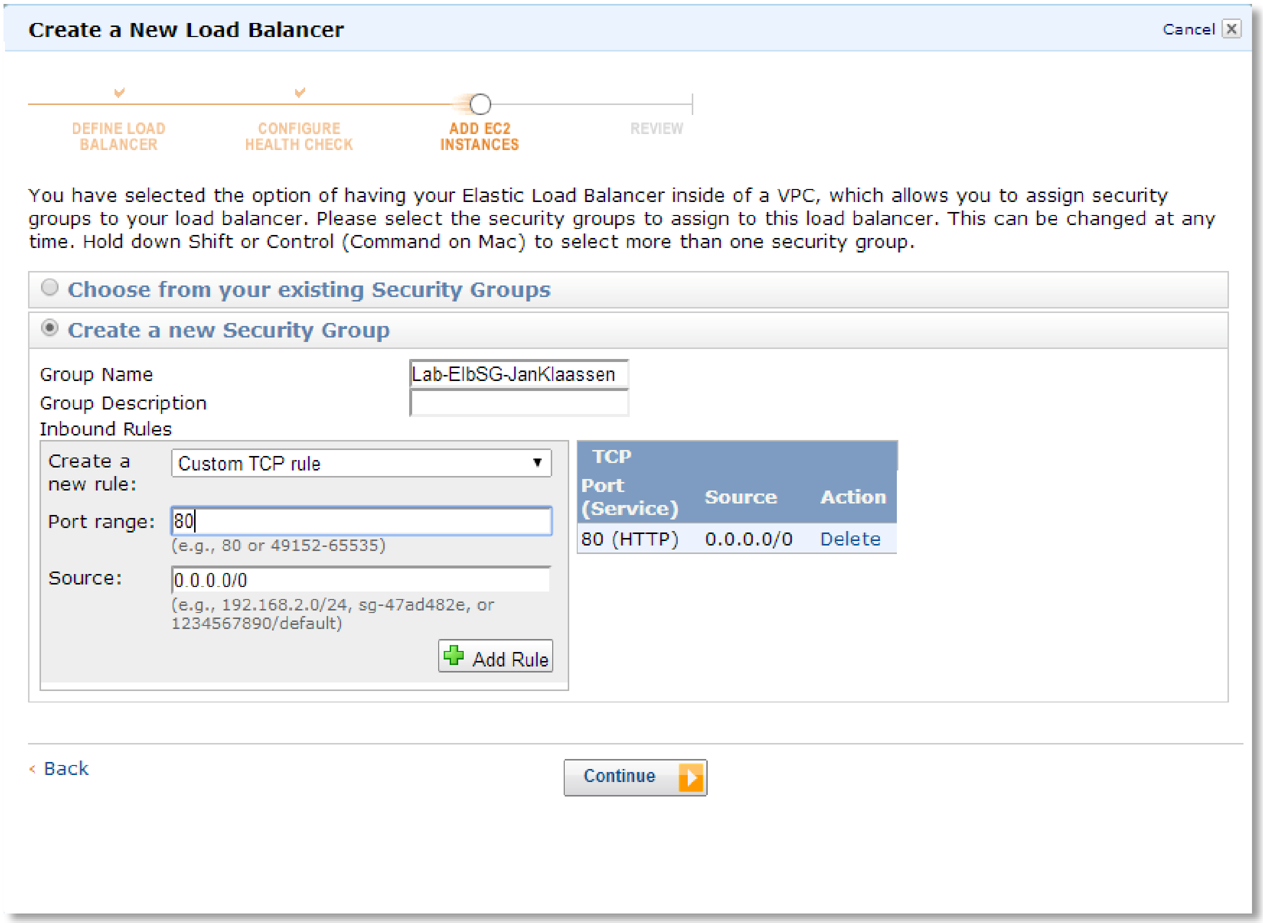
	Subnet ID	Subnet CIDR	Availability Zones
⊕	subnet-eb5f649f	10.0.1.0/24	eu-west-1a

Selected Subnets*

	Subnet ID	Subnet CIDR	Availability Zones
⊗	subnet-d45f64a0	10.0.0.0/24	eu-west-1a

So you need to understand your VPC network architecture to select the right subnet. In this case, the 10.0.0.0/24 subnet is the private one and is selected to host your load balancers. Click **Continue**.

In a VPC, a Load Balancer can have its own security group. Ensure that the world has access to it on port 80, and click **Continue**.



Create a New Load Balancer

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time. Hold down Shift or Control (Command on Mac) to select more than one security group.

○ Choose from your existing Security Groups

● Create a new Security Group

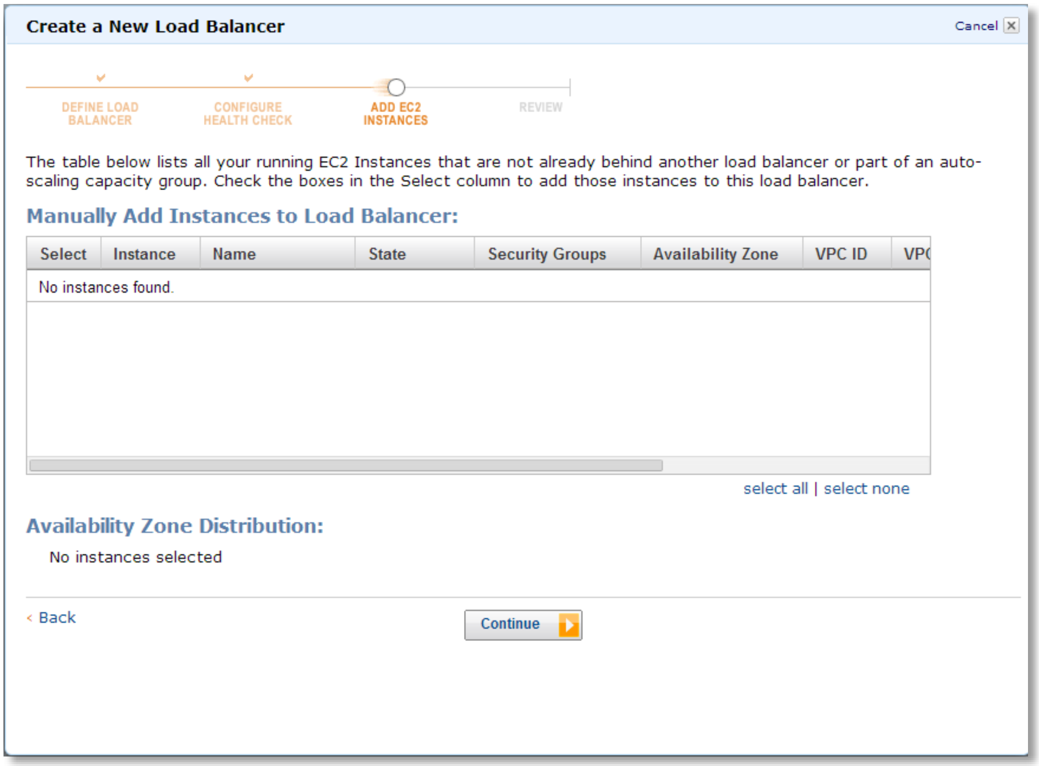
Group Name: Lab-ElbSG-JanKlaassen
Group Description:
Inbound Rules

Create a new rule: Custom TCP rule
Port range: 80
(e.g., 80 or 49152-65535)
Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP

Port (Service)	Source	Action
80 (HTTP)	0.0.0.0/0	Delete

In the following screen, you (might) see a list of running instances. Select **none** of them and click **Continue**.



Create a New Load Balancer

The table below lists all your running EC2 Instances that are not already behind another load balancer or part of an auto-scaling capacity group. Check the boxes in the Select column to add those instances to this load balancer.

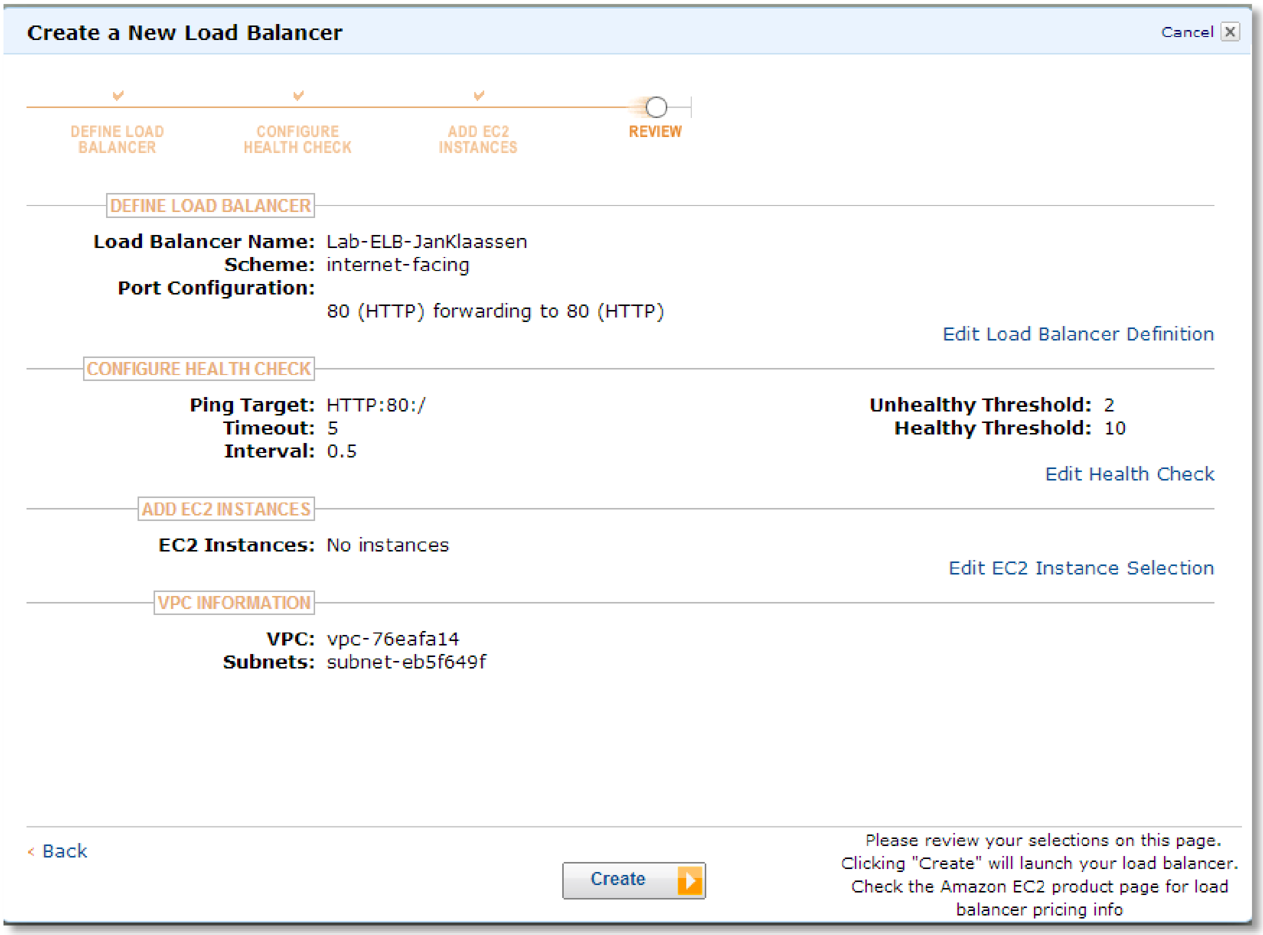
Manually Add Instances to Load Balancer:

Select	Instance	Name	State	Security Groups	Availability Zone	VPC ID	VP(
No instances found.							

Availability Zone Distribution:

No instances selected

Review your ELB settings and click **Create** (followed by **Close**).



Create a New Load Balancer

DEFINE LOAD BALANCER

Load Balancer Name: Lab-ELB-JanKlaassen
Scheme: internet-facing
Port Configuration:
80 (HTTP) forwarding to 80 (HTTP)

CONFIGURE HEALTH CHECK

Ping Target: HTTP:80:/
Timeout: 5
Interval: 0.5
Unhealthy Threshold: 2
Healthy Threshold: 10

ADD EC2 INSTANCES

EC2 Instances: No instances

VPC INFORMATION

VPC: vpc-76eafa14
Subnets: subnet-eb5f649f

Please review your selections on this page. Clicking "Create" will launch your load balancer. Check the Amazon EC2 product page for load balancer pricing info

AWS is now creating your ELB.

Autoscaling

Important! Make certain that you delete Auto Scaling at the end of the day. As you will see in this lab, the service does exactly what you tell it to, and if you simply terminate servers, it will notice and start them right back up. 30 days later you will wonder why you received a large bill from AWS. Instructions for tearing down the service are at the very end of this workbook.

Autoscaling Principles

First, Autoscaling is a way to set the Cloud temperature. You use rules to “set the thermostat”, and under the hood Autoscaling controls the heat by adding and subtracting EC2 resources on an as-needed basis, in order to maintain the “temperature” (capacity).

Second, autoscaling assumes a set of homogeneous servers. That is, autoscaling is not smart enough to know that Server A is a 64-bit XL instance that is more capable than a 32-bit Small instance. In fact, this is a core tenet of Cloud Computing: scale horizontally using a fleet of fungible resources. An individual resource is not important, and accordingly the philosophy is “easy come, easy go”.

The 4 Key Components of Autoscaling

When you launch a server manually, you provide parameters such as which AMI to launch in, what instance size to launch on, which security group to launch in, etc. Autoscale calls this a **Launch Configuration**. It’s simply a set of parameters.

Auto Scaling Groups tell the system what to do with the AMI once it launches. This is where you specify which AZ it should it launch in, which load balancer to use, and – most importantly this is where you specify the minimum and maximum number of servers to run at any given time.

You need rules that tell the system when to add or subtract servers. These are known as **Scaling Triggers**, and have rules such as “if average CPU across servers in the autoscaling group exceeds 65% for 10 minutes, scale the fleet up by 10%” and “if average CPU across servers in the autoscaling group drops below 40% for 14 minutes, scale down by 1 server”.

When a trigger fires, it starts a **Scaling Event**. That’s nothing more than the act of scaling up or down.

Timing Matters

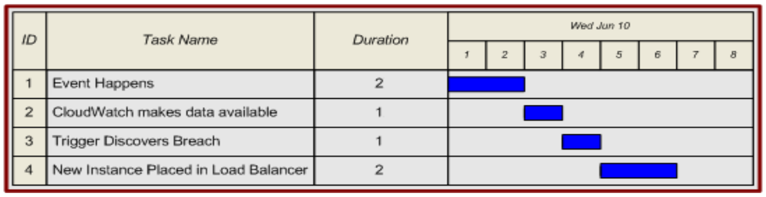
You literally earn your keep – or burn a lot of dollar bills – using Autoscaling. There are two important concepts that directly affect the cost of AWS, and also the manner in which your application will scale:

The Minimum Unit of Cost for EC2 is One Hour

It does not matter whether an EC2 instance runs for 60 seconds or 60 minutes: AWS bills for the full hour. Accordingly it is very important to avoid a short-cycle situation where a server is added to the fleet for 10 minutes, decommissioned, and then another server is added a few minutes later.

Scaling Takes Time

Consider the graph below. In most situations a considerable amount of time passes between when there is the need for a scaling event, and when the event happens.



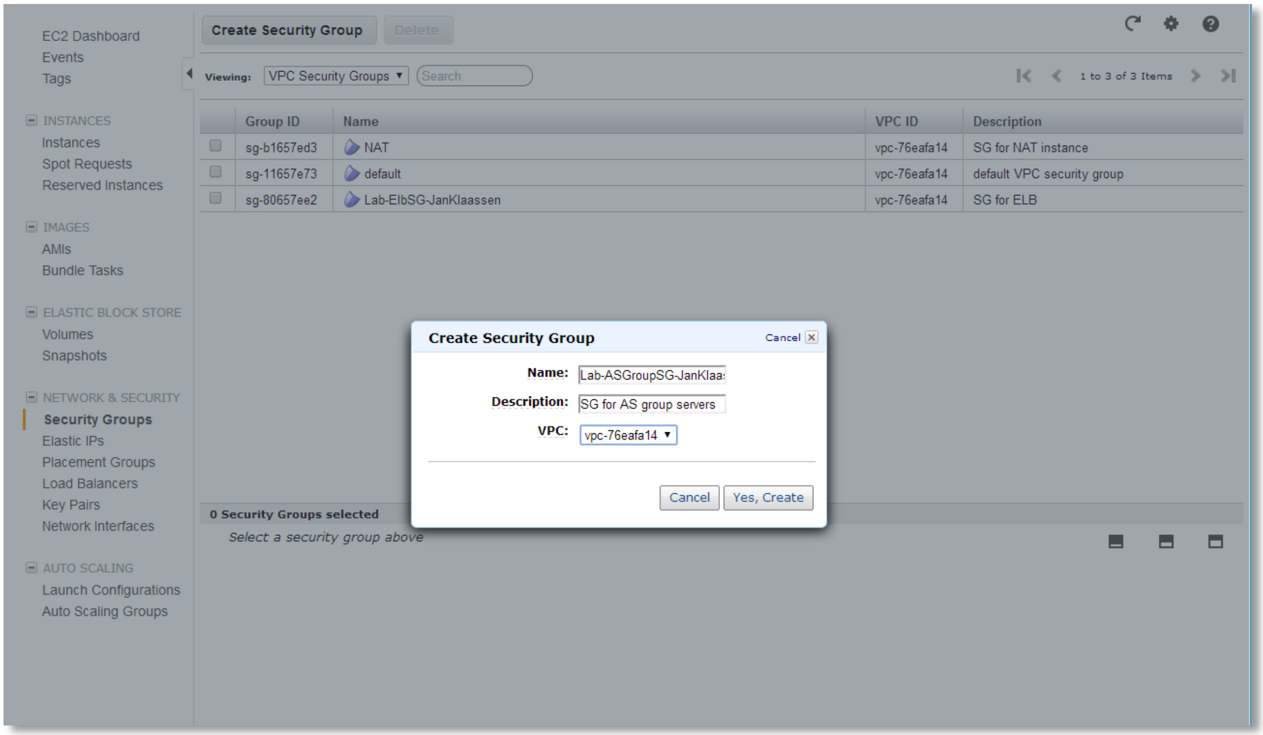
ID	Task Name	Duration	Wed Jun 10							
			1	2	3	4	5	6	7	8
1	Event Happens	2	■	■						
2	CloudWatch makes data available	1			■					
3	Trigger Discovers Breach	1				■				
4	New Instance Placed in Load Balancer	2					■	■		

- In this example the rule says that we need to be in a particular condition for at least 2 minutes.
- CloudWatch is the underlying data collection system that monitors statistics such as CPU utilization. It is a polling protocol, and in general takes 60 seconds to aggregate the data.
- Autoscaling is also a polling system, and it takes another 60 seconds.
- Then there is boot time for your server. A large, complex, server may take many minutes to launch.

- Finally, the load balancer needs to poll the server for a few cycles before it is comfortable that the server is healthy and accepting requests.

Create a Security Group

In the **EC2 Console**, click on **Security Groups** and **Create Security Group**.



Viewing: VPC Security Groups

1 to 3 of 3 Items

	Group ID	Name	VPC ID	Description
□	sg-b1657ed3	NAT	vpc-76eafa14	SG for NAT instance
□	sg-11657e73	default	vpc-76eafa14	default VPC security group
□	sg-80657ee2	Lab-ElbSG-JanKlaassen	vpc-76eafa14	SG for ELB

Create Security Group

Name: Lab-ASGroupSG-JanKlaa:

Description: SG for AS group servers

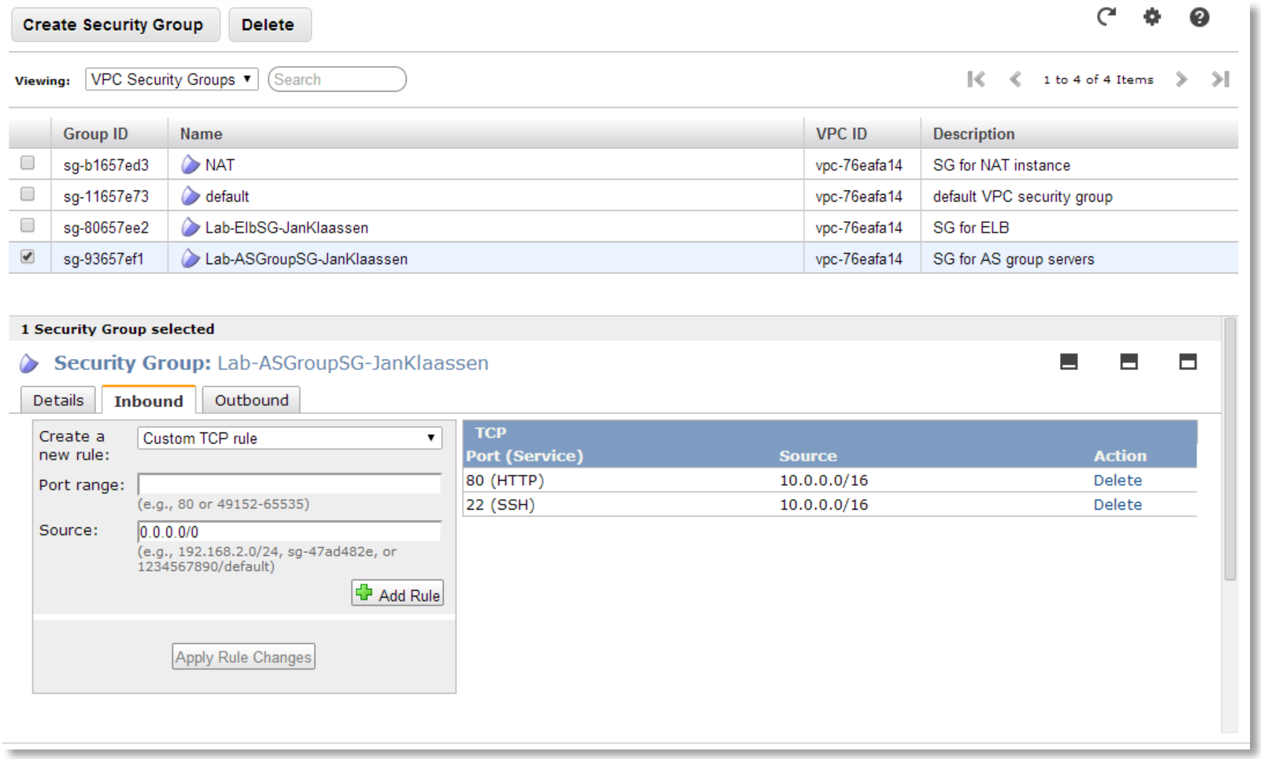
VPC: vpc-76eafa14

0 Security Groups selected

Select a security group above

Give the security group a logical (and unique) name, and make sure it is created in the right VPC.

Then select the newly created security group (make sure you are viewing the **VPC Security Groups**) and create inbound rules for ssh and http. Open these up for all IP addresses in the VPC (10.0.0.0/16).



	Group ID	Name	VPC ID	Description
□	sg-b1657ed3	NAT	vpc-76eafa14	SG for NAT instance
□	sg-11657e73	default	vpc-76eafa14	default VPC security group
□	sg-80657ee2	Lab-ElbSG-JanKlaassen	vpc-76eafa14	SG for ELB
☑	sg-93657ef1	Lab-ASGroupSG-JanKlaassen	vpc-76eafa14	SG for AS group servers

1 Security Group selected

Security Group: Lab-ASGroupSG-JanKlaassen

Create a new rule: Custom TCP rule

Port range:
(e.g., 80 or 49152-65535)

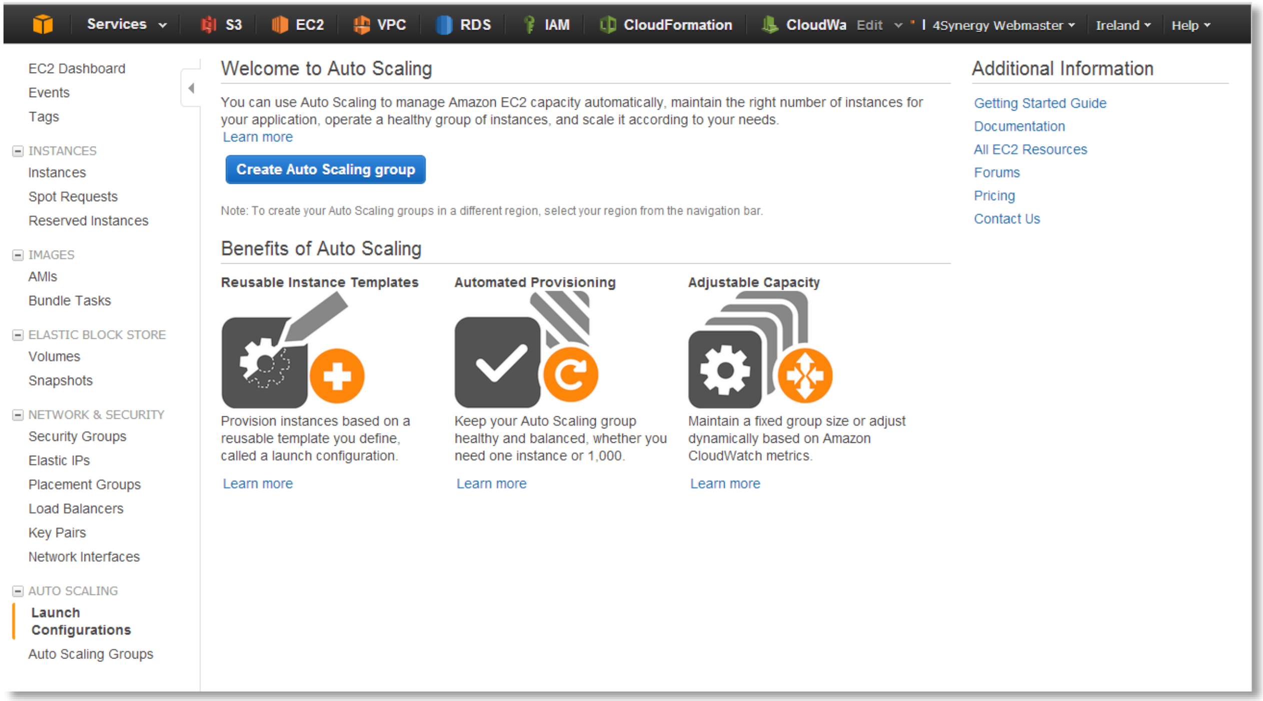
Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP Port (Service)	Source	Action
80 (HTTP)	10.0.0.0/16	Delete
22 (SSH)	10.0.0.0/16	Delete

Do not forget to **Apply Rule Changes**!

Create a Launch Configuration

In the **EC2 Console**, click on **Launch Configurations** and **Create Auto Scaling Group**.



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

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

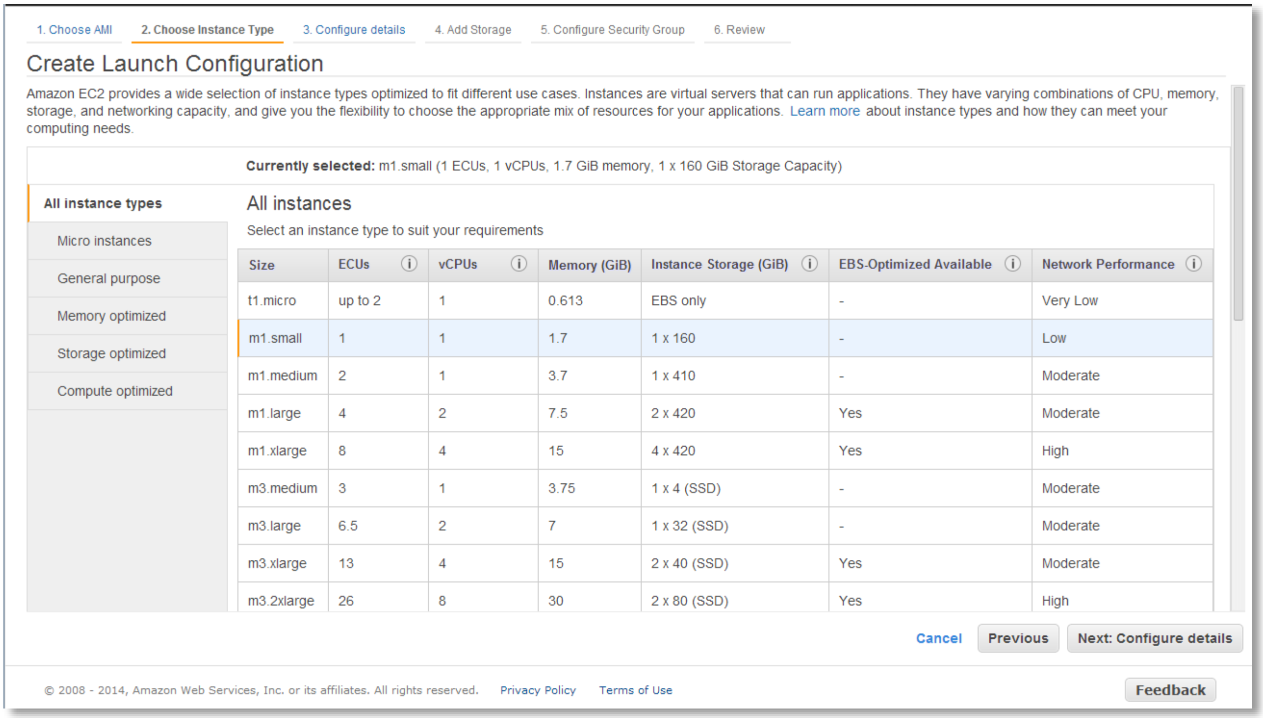
Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000. Learn more

Adjustable Capacity

Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics. Learn more

Read the following informational screen, and then click the **Create launch configuration** link.

Now you need to select the server image (called AMI) that will be used. We have prepared a simple AMI for this purpose, and to configure that, select **Community AMIs** and search for “ami-aebe54d9”. Click **Select**.



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.

Currently selected: m1.small (1 ECUs, 1 vCPUs, 1.7 GiB memory, 1 x 160 GiB Storage Capacity)

All instances

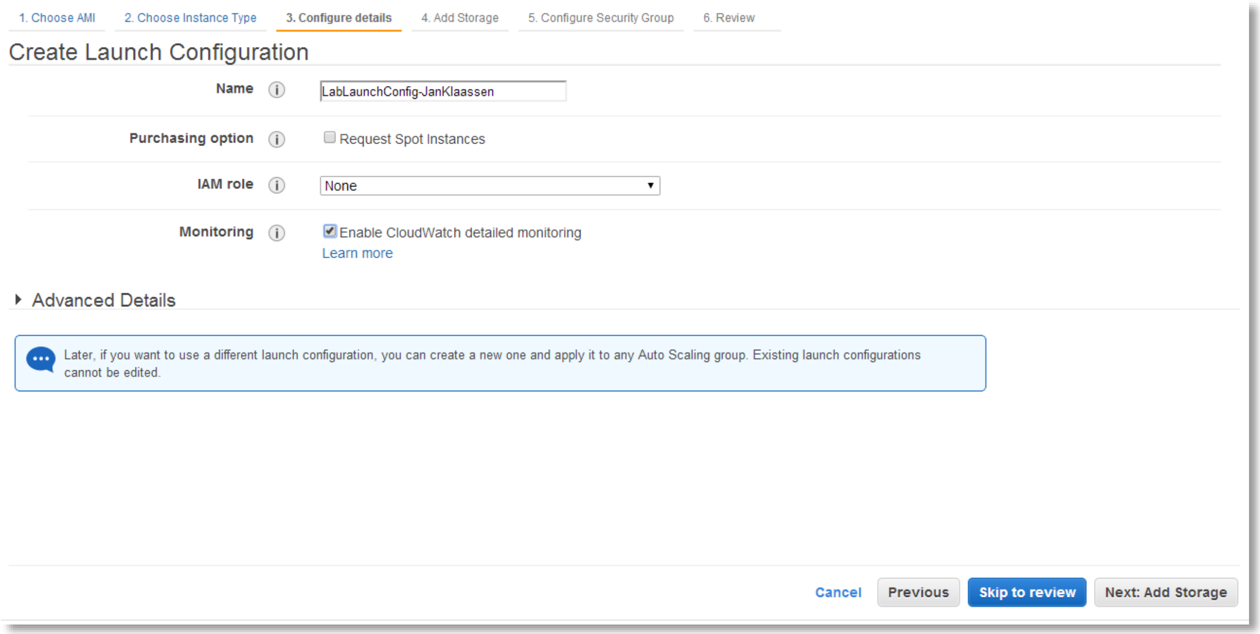
Select an instance type to suit your requirements

Size	ECUs ⓘ	vCPUs ⓘ	Memory (GiB)	Instance Storage (GiB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ
t1.micro	up to 2	1	0.613	EBS only	-	Very Low
m1.small	1	1	1.7	1 x 160	-	Low
m1.medium	2	1	3.7	1 x 410	-	Moderate
m1.large	4	2	7.5	2 x 420	Yes	Moderate
m1.xlarge	8	4	15	4 x 420	Yes	High
m3.medium	3	1	3.75	1 x 4 (SSD)	-	Moderate
m3.large	6.5	2	7	1 x 32 (SSD)	-	Moderate
m3.xlarge	13	4	15	2 x 40 (SSD)	Yes	Moderate
m3.2xlarge	26	8	30	2 x 80 (SSD)	Yes	High

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Next click the type of instance (the hardware specifications) you want to use. As this is a simple web server, you can do with a Micro or Small instance. Click **Next: Configure details**.

To state the obvious, the more powerful instance type you select, the higher the hourly costs will be.



Create Launch Configuration

Name: LabLaunchConfig-JanKlaassen

Purchasing option: □ Request Spot Instances

IAM role: None

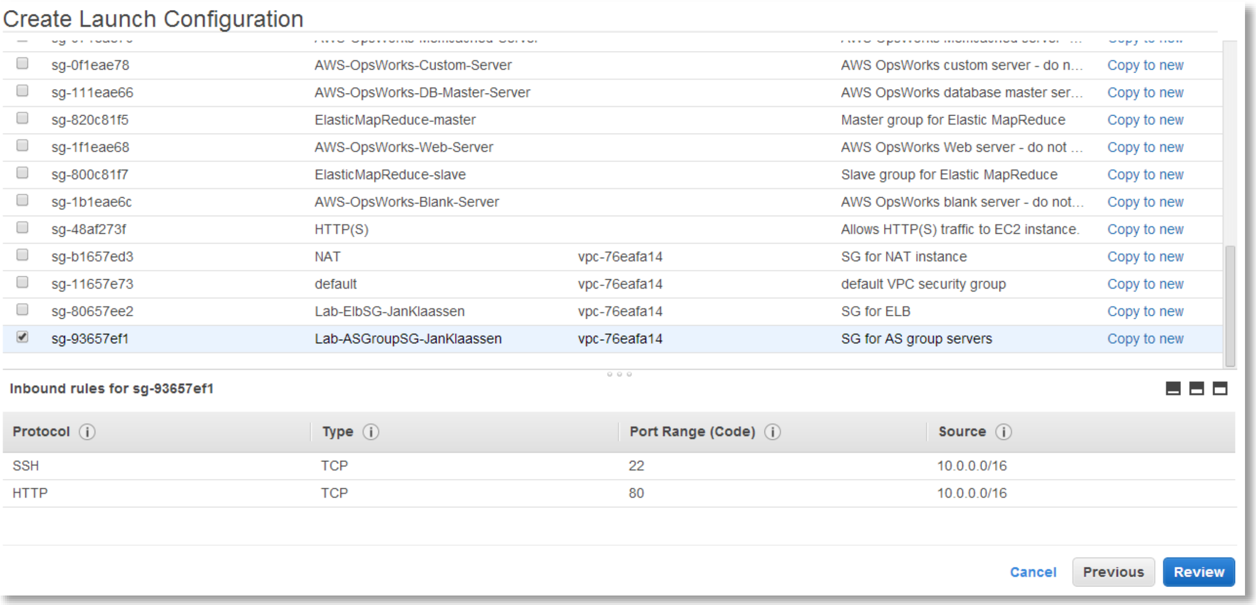
Monitoring: ☑ Enable CloudWatch detailed monitoring
Learn more

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.

Give the launch configuration a logical (and unique) name. Evaluate the other options, but for now only select the **Enable CloudWatch detailed monitoring**. This will increase the frequency of updates from the instance to Cloudwatch, which is important for having a smooth scaling behavior.

Accept the defaults at the **Add Storage** page and click **Next: Configure Security Group**. Select the security group you've just created and then click **Review**.



Create Launch Configuration

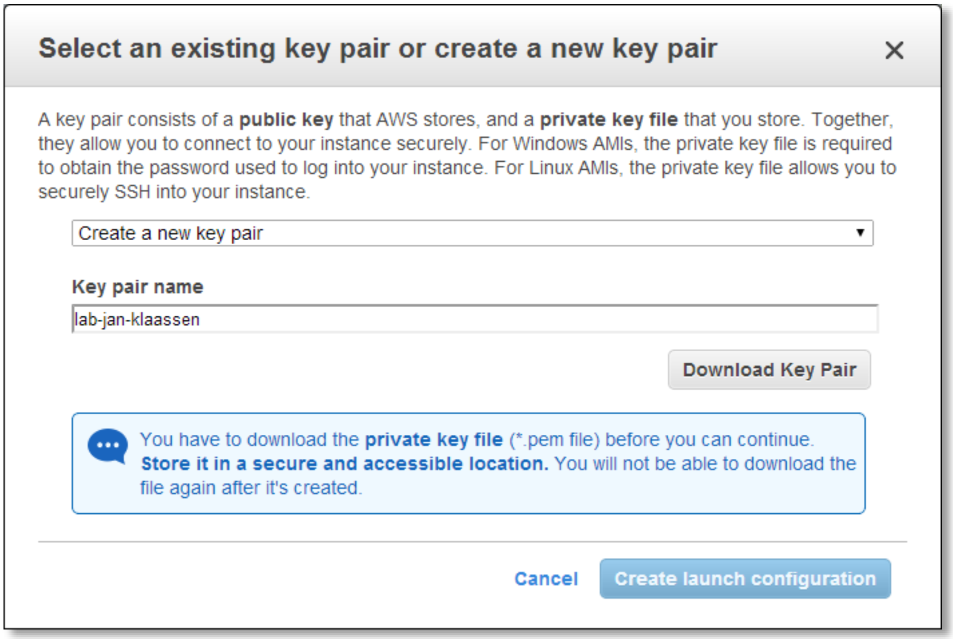
□	sg-0f1eae78	AWS-OpsWorks-Custom-Server		AWS OpsWorks custom server - do n...	Copy to new
□	sg-111eae66	AWS-OpsWorks-DB-Master-Server		AWS OpsWorks database master ser...	Copy to new
□	sg-820c81f5	ElasticMapReduce-master		Master group for Elastic MapReduce	Copy to new
□	sg-1f1eae68	AWS-OpsWorks-Web-Server		AWS OpsWorks Web server - do not ...	Copy to new
□	sg-800c81f7	ElasticMapReduce-slave		Slave group for Elastic MapReduce	Copy to new
□	sg-1b1eae6c	AWS-OpsWorks-Blank-Server		AWS OpsWorks blank server - do not...	Copy to new
□	sg-48af273f	HTTP(S)		Allows HTTP(S) traffic to EC2 instance.	Copy to new
□	sg-b1657ed3	NAT	vpc-76eafa14	SG for NAT instance	Copy to new
□	sg-11657e73	default	vpc-76eafa14	default VPC security group	Copy to new
□	sg-80657ee2	Lab-ElbSG-JanKlaassen	vpc-76eafa14	SG for ELB	Copy to new
☑	sg-93657ef1	Lab-ASGroupSG-JanKlaassen	vpc-76eafa14	SG for AS group servers	Copy to new

Inbound rules for sg-93657ef1

Protocol ⓘ	Type ⓘ	Port Range (Code) ⓘ	Source ⓘ
SSH	TCP	22	10.0.0.0/16
HTTP	TCP	80	10.0.0.0/16

Review your settings, and if you're happy with it click **Create launch configuration**.

In the following screen you are asked to select or create a key pair. This key pair is needed when you want to log in (with SSH) to your instance.



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Create a new key pair ▾

Key pair name: lab-jan-klaassen

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

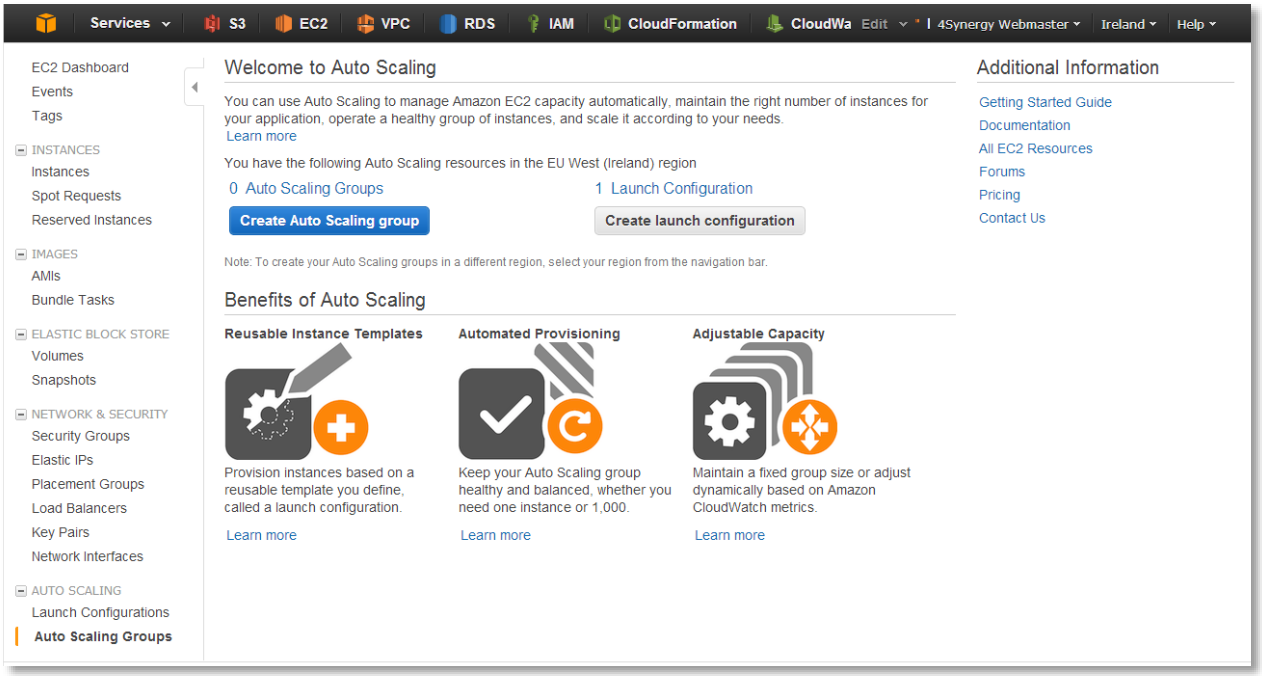
Select **Create a new key pair**, give it a logical (and unique) name and click **Download Key Pair**. Then (again) click **Create launch configuration**. Your launch configuration is now created, including related resources such as the security group you specified.

Important: the key (.pem) file you just download is the private key of a key pair and can only be downloaded once. So make sure you don't lose it, and make sure you don't share it as it gives access to your servers (and hence your applications and data).

Create an Auto Scaling Group

Please note: the next two pages are optional and are skipped when you create a launch configuration.

Now we are going to create an auto scaling group. In the **EC2 Console**, click **Auto Scaling Groups** and then **Create Auto Scaling Group**.



EC2 Dashboard
Events
Tags
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces
AUTO SCALING
Launch Configurations
Auto Scaling Groups

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

You have the following Auto Scaling resources in the EU West (Ireland) region

0 Auto Scaling Groups
1 Launch Configuration

Create Auto Scaling group
Create launch configuration

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

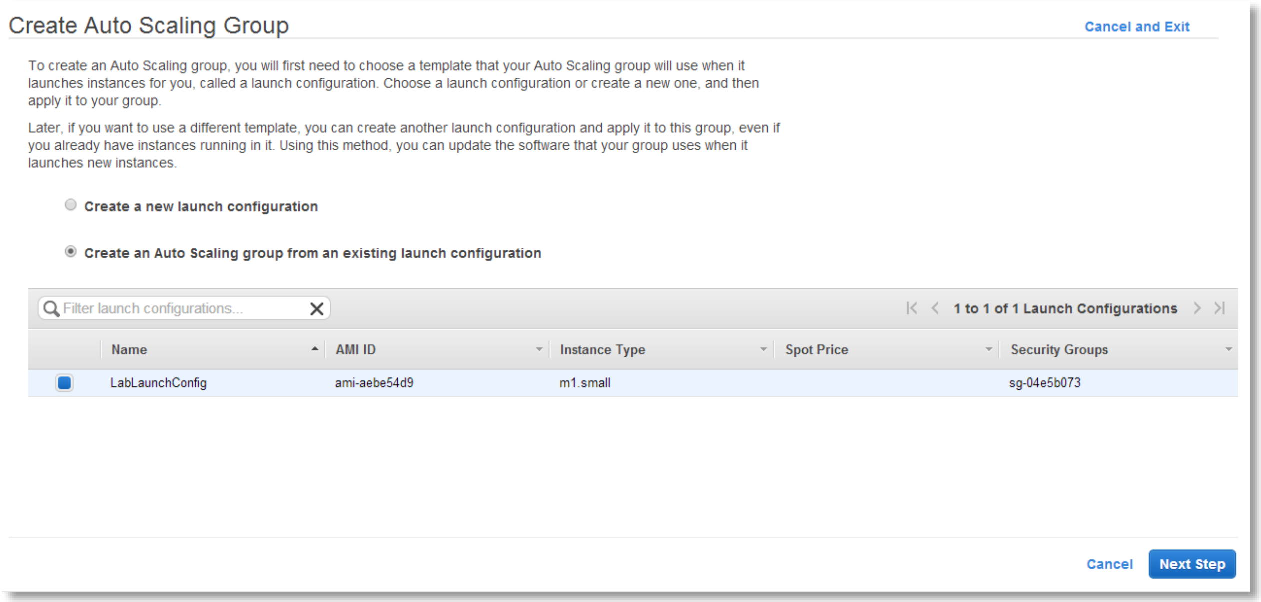
Benefits of Auto Scaling

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



Create Auto Scaling Group

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.

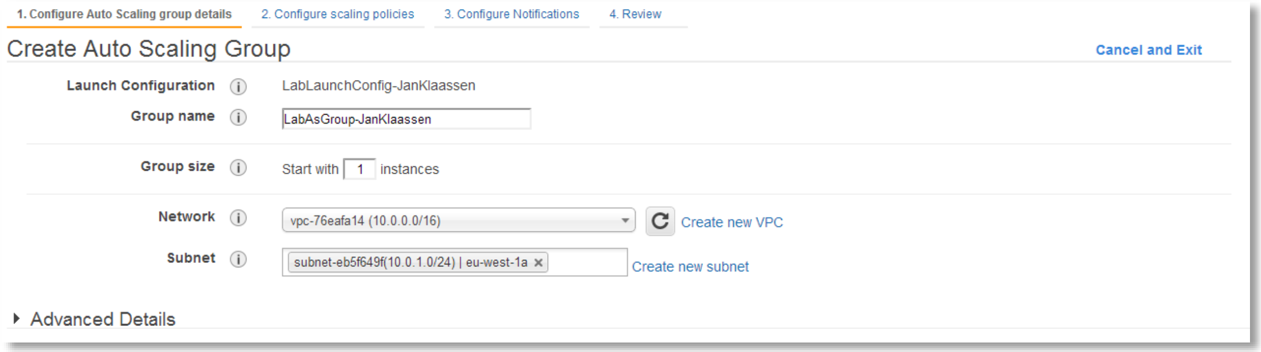
○ Create a new launch configuration

● Create an Auto Scaling group from an existing launch configuration

Filter launch configurations...
1 to 1 of 1 Launch Configurations

	Name	AMI ID	Instance Type	Spot Price	Security Groups
■	LabLaunchConfig	ami-aebe54d9	m1.small		sg-04e5b073

Select **Create an Auto Scaling group from an existing launch configuration** and select the launch config you just created.



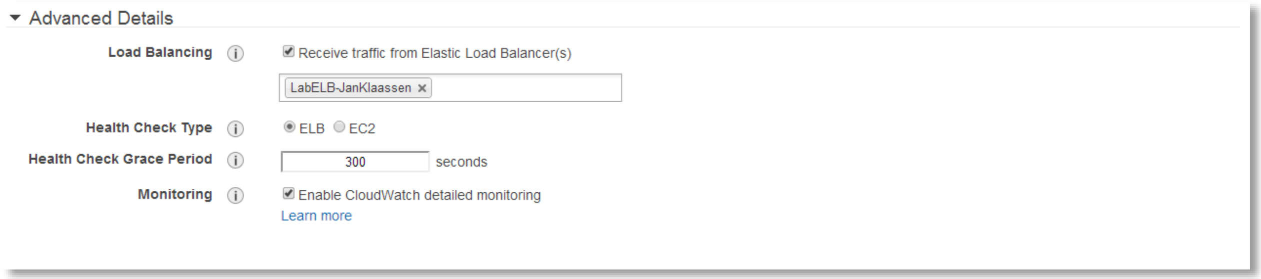
1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Review
Cancel and Exit

Create Auto Scaling Group

Launch Configuration	LabLaunchConfig-JanKlaassen
Group name	LabAsGroup-JanKlaassen
Group size	Start with 1 instances
Network	vpc-76eafa14 (10.0.0.0/16)
Subnet	subnet-eb5f649f(10.0.1.0/24) | eu-west-1a

Create new VPC
Create new subnet
Advanced Details

Give the Auto scaling group a logical (and unique) name, specify the number of instances to start with. Also select the network (must be the same VPC) and the (private) subnet.

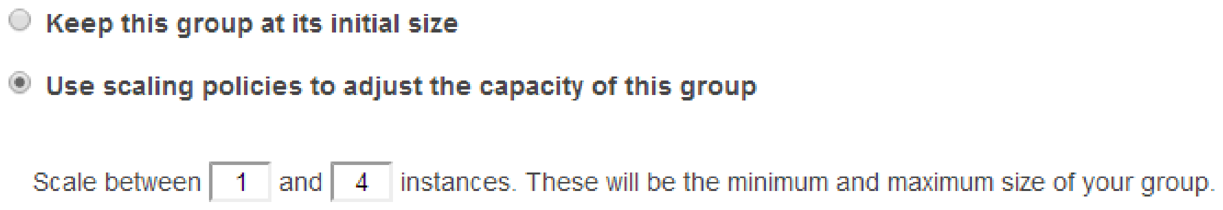


Advanced Details

Load Balancing	☑ Receive traffic from Elastic Load Balancer(s) LabELB-JanKlaassen
Health Check Type	● ELB ○ EC2
Health Check Grace Period	300 seconds
Monitoring	☑ Enable CloudWatch detailed monitoring

Now open the **Advanced Details** section and click **Receive traffic from Elastic Load Balancer(s)**. Select the load balancer you just created. Also evaluate the other options and select as shown above. Click **Next: Configure scaling policies**.

In this section you are going to define the automatic scaling behavior. First select that you want to use **Scaling policies** and provide the lower and upper boundaries of your group Auto scaling will never exceed these.

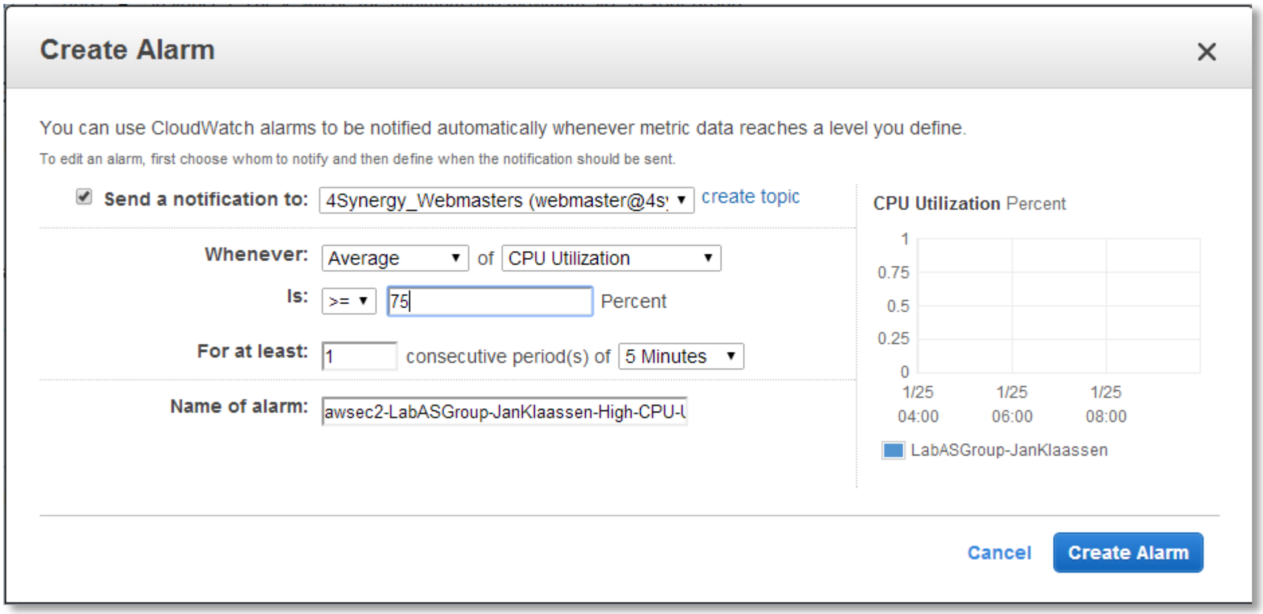


○ Keep this group at its initial size

● 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.

Then you need to define the conditions when the group will be scaled up and when it will be scaled down again. For that you need to Add a new alarm.



Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

☑ Send a notification to: 4Synergy_Webmasters (webmaster@4s) create topic

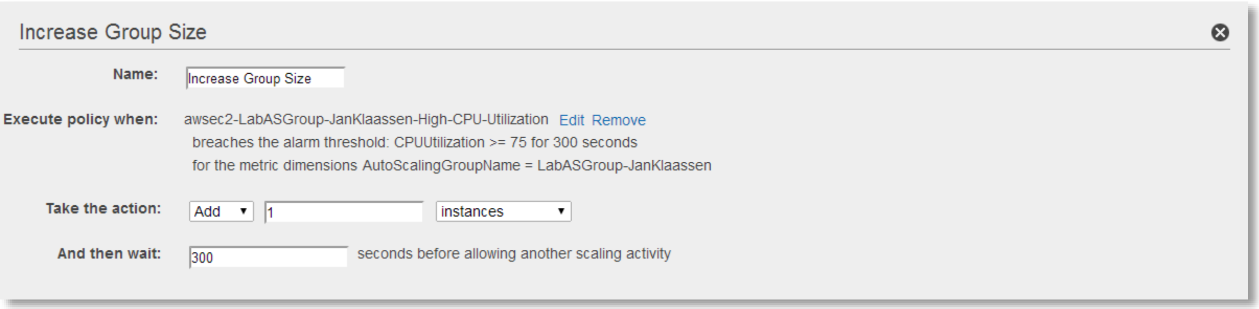
Whenever: Average of CPU Utilization

Is: >= 75 Percent

For at least: 1 consecutive period(s) of 5 Minutes

Name of alarm: awsec2-LabASGroup-JanKlaassen-High-CPU-U

For the Scale up policy, create an alarm as shown above, and then complete the **Increase Group Size** policy.



Increase Group Size

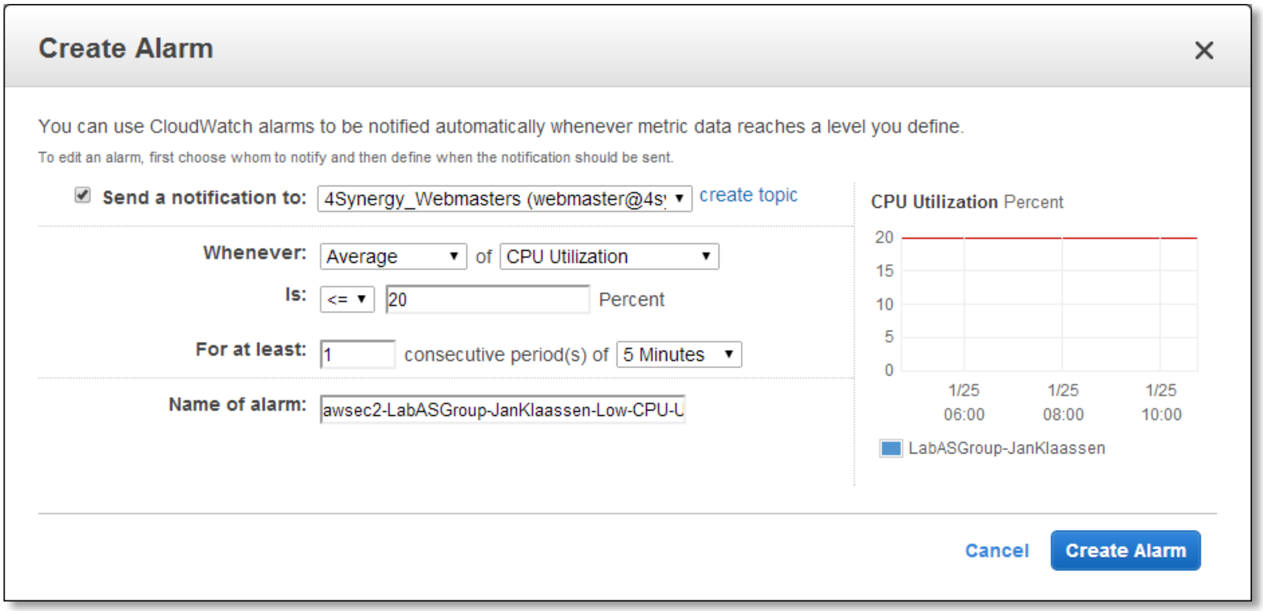
Name: Increase Group Size

Execute policy when: awsec2-LabASGroup-JanKlaassen-High-CPU-Utilization
breaches the alarm threshold: CPUUtilization >= 75 for 300 seconds
for the metric dimensions AutoScalingGroupName = LabASGroup-JanKlaassen

Take the action: Add 1 instances

And then wait: 300 seconds before allowing another scaling activity

Repeat the process for the **Decrease Group Size** policy, but (obviously) with other (non-conflicting!) alarm rules. Make sure you give the alarm a new name!



Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

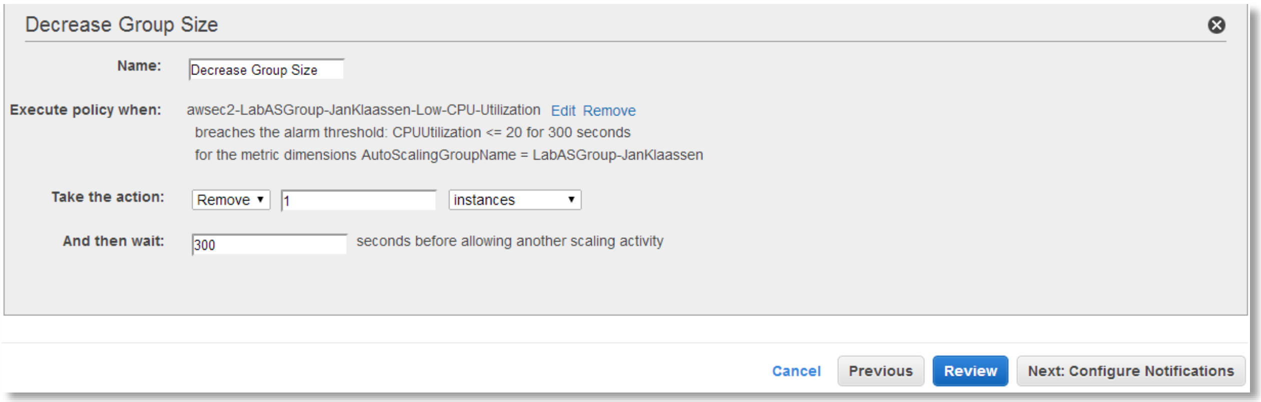
☑ Send a notification to: 4Synergy_Webmasters (webmaster@4s

Whenever: Average of CPU Utilization

Is: <= 20 Percent

For at least: 1 consecutive period(s) of 5 Minutes

Name of alarm: awsec2-LabASGroup-JanKlaassen-Low-CPU-L



Decrease Group Size

Name: Decrease Group Size

Execute policy when: awsec2-LabASGroup-JanKlaassen-Low-CPU-Utilization
breaches the alarm threshold: CPUUtilization <= 20 for 300 seconds
for the metric dimensions AutoScalingGroupName = LabASGroup-JanKlaassen

Take the action: Remove 1 instances

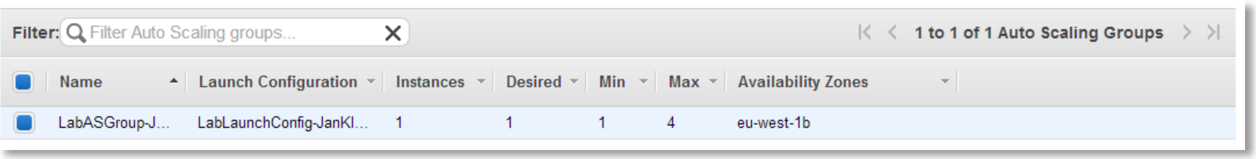
And then wait: 300 seconds before allowing another scaling activity

In the next screen you can select whether you want to receive notifications when your auto scaling group is changing in size. For this lab, we will ignore this option, click **Review** and then **Create auto scaling group**.

Your group is now being created, usually it takes a few minutes before the first instance is actually created.

Verify that the Servers Launched

Check the launch configuration and see whether actual instances are added to the group.

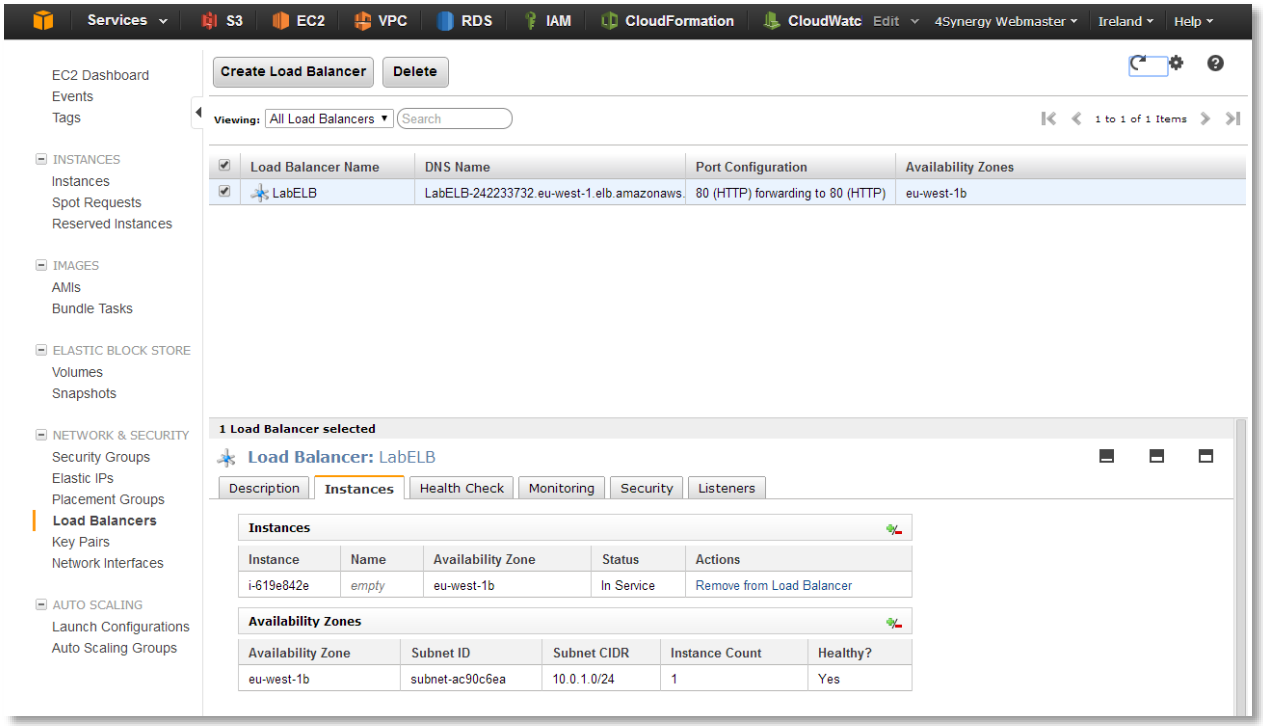


Filter: Filter Auto Scaling groups...   1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones
LabASGroup-J...	LabLaunchConfig-JanKl...	1	1	1	4	eu-west-1b

The instance(s) should also show up in the **EC2 Instances** section.

Next, inspect the ELB's instances section, and wait until the instance's status is turning into **In Service**. This might take about 5 to 10 minutes.



Load Balancer Name	DNS Name	Port Configuration	Availability Zones
LabELB	LabELB-242233732.eu-west-1.elb.amazonaws.	80 (HTTP) forwarding to 80 (HTTP)	eu-west-1b

1 Load Balancer selected

Load Balancer: LabELB

Instances

Instance	Name	Availability Zone	Status	Actions
i-619e842e	*empty*	eu-west-1b	In Service	Remove from Load Balancer

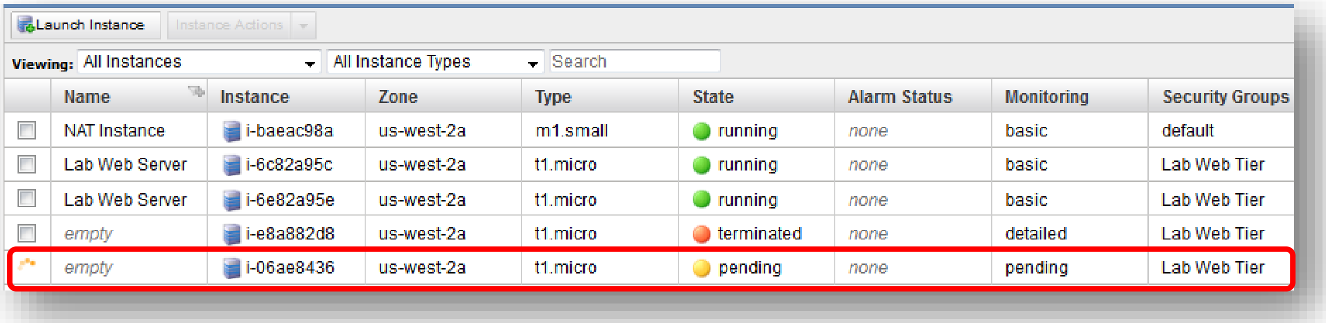
Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?
eu-west-1b	subnet-ac90c6ea	10.0.1.0/24	1	Yes

Now you should be able to access your instance through the DNS name of the ELB (which is shown in the **Description** tab).

Bonus: Verify How Auto Scaling Works

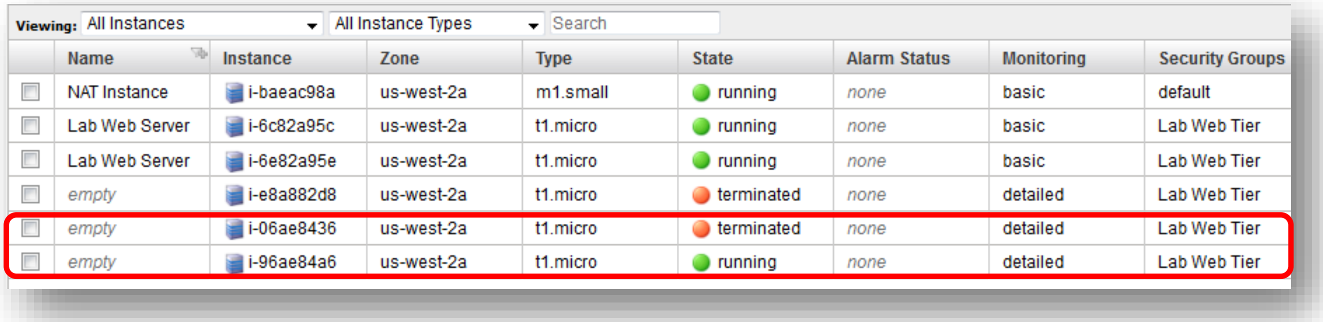
Try terminating the server(s). In a few minutes they will re-appear, because Auto Scaling will notice that the fleet size is below minimum size (This is why you need to delete the autoscaling group via as-delete-auto-scaling-group at the end of the day).



Viewing: All Instances | All Instance Types | Search

	Name	Instance	Zone	Type	State	Alarm Status	Monitoring	Security Groups
□	NAT Instance	i-baeac98a	us-west-2a	m1.small	running	none	basic	default
□	Lab Web Server	i-6c82a95c	us-west-2a	t1.micro	running	none	basic	Lab Web Tier
□	Lab Web Server	i-6e82a95e	us-west-2a	t1.micro	running	none	basic	Lab Web Tier
□	*empty*	i-e8a882d8	us-west-2a	t1.micro	terminated	none	detailed	Lab Web Tier
	empty	i-06ae8436	us-west-2a	t1.micro	pending	none	pending	Lab Web Tier

Try doing the same thing by shutting down the instance (rather than outright terminating it) either by logging into the new instance and issuing a shutdown command (sudo shutdown -h now) or by right-clicking on the instance and selecting **Stop** from the AWS Management Console. Notice that Auto Scaling will detect that the instance is non-responsive and will automatically terminate it and launch a replacement instance for you.



Viewing: All Instances | All Instance Types | Search

	Name	Instance	Zone	Type	State	Alarm Status	Monitoring	Security Groups
□	NAT Instance	i-baeac98a	us-west-2a	m1.small	running	none	basic	default
□	Lab Web Server	i-6c82a95c	us-west-2a	t1.micro	running	none	basic	Lab Web Tier
□	Lab Web Server	i-6e82a95e	us-west-2a	t1.micro	running	none	basic	Lab Web Tier
□	*empty*	i-e8a882d8	us-west-2a	t1.micro	terminated	none	detailed	Lab Web Tier
□	*empty*	i-06ae8436	us-west-2a	t1.micro	terminated	none	detailed	Lab Web Tier
□	*empty*	i-96ae84a6	us-west-2a	t1.micro	running	none	detailed	Lab Web Tier

"oldest" instances are terminated first. This allows you to roll in new changes to your application by updating your launch config to a newer AMI, then triggering Auto Scaling events (e.g. increase min size).

Wrapping Up

How Large an Auto-Scaling Farm Can I Have?

By default each account is assigned limits on a per-region basis. Initially accounts are set to a maximum of 20 EC2 instances, 100 spot instances, and 5000 EBS volumes or an aggregate size of 20 TB (whichever is smaller).

Does the max-instance limit only count towards running instances? In other words, do Stopped Instances (not terminated) also count towards this limit?

The max-instances limit only applies for instances that are in the pending, running, shutting-down and stopping states. There is another limit, which is set to 4x the max-instances limit that governs the total number of instances in any state. So, for a default account with no limit overrides, you can have 20 "running" instances and up to 60 stopped instances; bringing this to a total of 80 instances.

All of these defaults can be changed by making a request online at http://aws.amazon.com/contact-us/.

CLI References

Please visit the Auto Scaling documentation page for additional information about Auto Scaling.
http://aws.amazon.com/documentation/autoscaling/

Also, check out the command line quick reference card for additional command line commands and options.
http://awsdocs.s3.amazonaws.com/AutoScaling/latest/as-qrc.pdf

Terminate

Make sure you delete your auto scaling group and load balancer to ensure you have no billable services running!