



TÉCNICO
LISBOA



Cloud Computing and Virtualization

Lab. 3 - Introduction to AWS

Goals of the lab

**Goal: Set up an auto-scaling
web server**



Root user sign in ⓘ

Email: your-account@your-domain

Password

[Forgot password?](#)

.....

[Sign in](#)

[Sign in to a different account](#)

[Create a new AWS account](#)



Amazon Lightsail

Easily launch and manage a virtual private server with AWS

[Learn more](#)



About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our Terms of Use and Privacy Policy linked below. Your use of Amazon Web Services products and services is governed by the AWS Customer Agreement linked below unless you have entered into a separate agreement with Amazon Web Services or an AWS Value Added Reseller to purchase these products and services. The AWS Customer Agreement was updated on March 31, 2017. For more information about these updates, see [Recent Changes](#).

Select US East North Virginia

AWS Management Console



The screenshot shows the AWS Management Console homepage. At the top, there's a dark header bar with the AWS logo, a "Services" dropdown, a "Resource Groups" dropdown, and a user dropdown for "loureiro.leonor". To the right of the user dropdown, there's a "N. Virginia" dropdown and a "Support" link. Below the header, the main content area has a title "AWS services" and a "Find Services" search bar. Under "Recently visited services", there are links for EC2, IAM, DynamoDB, and Billing. Under "All services", there are sections for Compute (EC2, Lightsail, ECR, ECS, EKS, Lambda, Batch), Satellite (Ground Station), Quantum Technologies (Amazon Braket), Management & Governance, Security, Identity, & Compliance (IAM, Resource Access Manager, Cognito, Secrets Manager, GuardDuty, Inspector), and Explore AWS (Free Digital Training, Event-Driven Architecture, Amazon GuardDuty). A large green arrow points from the top right towards the "N. Virginia" dropdown.

Select EC2

The screenshot shows the AWS Management Console with the following interface elements:

- Header:** AWS logo, Services dropdown, Resource Groups dropdown, a small icon, a bell icon, user name "loureiro.leonor", region "N. Virginia", and Support link.
- # AWS Management Console
- AWS services section:**
 - Find Services:** A search bar with placeholder text "Example: Relational Database Service, database, RDS".
 - Recently visited services:** EC2, IAM, DynamoDB, Billing.
 - All services:** A grid of service icons and names. The "Compute" column includes EC2, Lightsail, ECR, ECS, EKS, Lambda, and Batch. The "Ground Station" row includes Satellite and Ground Station. The "Quantum Technologies" row includes Quantum Technologies and Amazon Braket. The "Management & Governance" row includes Management & Governance and Inspector. A large green arrow points to the Lightsail service in the Compute column.
- Access resources on the go:** A section with an icon of a smartphone and text: "Access the Management Console using the AWS Console Mobile App. Learn more" with a blue link.
- Explore AWS:**
 - Free Digital Training:** Text: "Get access to 350+ self-paced online courses covering AWS products and services. Learn more" with a blue link.
 - Event-Driven Architecture:** Text: "Decoupled apps with automatic scaling and simplified auditing. Write less code, save money, and move faster than ever. Learn more" with a blue link.
 - Amazon GuardDuty:** Partially visible at the bottom.

Launch an Instance

AWS Services Resource Groups ★

New EC2 Experience Tell us what you think C G

EC2 Dashboard New

- Events
- Tags
- Reports
- Limits

▼ **INSTANCES**

- Instances
- Instance Types
- Launch Templates New
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Scheduled Instances
- Capacity Reservations

▼ **IMAGES**

- AMIs
- Bundle Tasks

▼ **ELASTIC BLOCK STORE**

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Running instances	0	Elastic IPs	0
Dedicated Hosts	0	Snapshots	0
Volumes	2	Load balancers	0
Key pairs	2	Security groups	3
Placement groups	0		

(i) Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#) X

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▾

Note: Your instances will launch in US East (N. Virginia)

Service health

C Service Health Dashboard

Region	Status
US East (N. Virginia)	✔ This service is operating normally



Pick AWS Linux from QuickStart

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

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
 Free tier only (i)

Amazon Linux Free tier eligible

Amazon Linux AMI 2015.09.2 (HVM), SSD Volume Type - ami-8fceee4e5

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm

Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type - ami-2051294a

Red Hat Enterprise Linux version 7.2 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm

SUSE Linux Enterprise Server 12 SP1 (HVM), SSD Volume Type - ami-b7b4fedd

SUSE Linux Enterprise Server 12 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs Virtualization type: hvm

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-fce3c696

Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs Virtualization type: hvm

Microsoft Windows Server 2012 R2 Base - ami-3586ac5f

Select 64-bit **Select** 64-bit **Select** 64-bit **Select** 64-bit



Pick a free tier machine

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

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)

T2 instances are VPC-only. Your T2 instance will launch into your VPC. Learn more about T2 and VPC.

	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 <small>Free tier eligible</small>	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		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

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy



Select a subnet and enable monitoring

Screenshot of the AWS Lambda Step 3: Configure Instance Details page. A green arrow points to the Subnet dropdown menu, which shows 'subnet-89f953fe(172.30.0.0/24) | us-east-1a' and '251 IP Addresses available'. Another green arrow points to the 'Monitoring' section, where the checkbox 'Enable CloudWatch detailed monitoring' is checked. A third green arrow points to the 'Review and Launch' button at the bottom right.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances Launch into Auto Scaling Group [i](#)

Purchasing option Request Spot instances

Network [vpc-ed028988 \(172.30.0.0/16\)](#) [Create new VPC](#)

Subnet [subnet-89f953fe\(172.30.0.0/24\) | us-east-1a](#) [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP [Use subnet setting \(Enable\)](#)

IAM role [None](#) [Create new IAM role](#)

Shutdown behavior [Stop](#)

Enable termination protection Protect against accidental termination

Monitoring Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy [Shared - Run a shared hardware instance](#)
Additional charges will apply for dedicated tenancy

Network interfaces [i](#)

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses
eth0	New network interface ▼	subnet-89f953fe ▼	Auto-assign	Add IP

Add Device [Add Device](#)

Advanced Details [Advanced Details](#)

Cancel Previous Review and Launch Next: Add Storage [Use](#)

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy

Use default storage

Screenshot of the AWS EC2 Instance Creation Wizard - Step 4: Add Storage.

The screenshot shows the configuration for adding storage to an instance. A green arrow points to the 'Snapshot' field, which contains the identifier 'snap-6618acf0'. Another green arrow points to the 'Next: Tag Instance' button at the bottom right of the page.

Step 4: Add Storage

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. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-6618acf0	8	General Purpose SSD (GP2)	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

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

Cancel Previous Review and Launch Next: Tag Instance

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy

Don't use tags for now...

Screenshot of the AWS EC2 Instance Creation Wizard - Step 5: Tag Instance.

The screenshot shows the "Tag Instance" step of the wizard. The navigation bar at the top includes "AWS", "Services", "Edit", and user information "João Coelho Garcia", "N. Virginia", and "Support". Below the navigation is a progress bar with steps 1 through 7: "1. Choose AMI", "2. Choose Instance Type", "3. Configure Instance", "4. Add Storage", "5. Tag Instance" (which is highlighted in yellow), "6. Configure Security Group", and "7. Review".

The main content area is titled "Step 5: Tag Instance" with the sub-instruction: "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." It contains two input fields: "Key" (with placeholder "(127 characters maximum)" and "Name" entered) and "Value" (with placeholder "(255 characters maximum)" and an empty field). A "Create Tag" button is present below the fields, with the note "(Up to 10 tags maximum)".

At the bottom of the page are buttons for "Cancel", "Previous", "Review and Launch" (which is highlighted in blue), and "Next: Configure Security Group". A green arrow points upwards from the bottom right towards the "Review and Launch" button.

Page footer: "Feedback" (with a speech bubble icon), "English" (with a globe icon), "© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.", "Privacy Policy", and "Terms of Use".

Add inbound network rules to security group (1)

Screenshot of the AWS EC2 instance creation wizard, Step 6: Configure Security Group.

The page shows the configuration of a new security group named "launch-wizard-2".

Assign a security group: Create a new security group (selected).

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2020-02-12T17:22:03.500+00:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule button is visible.

Warning: Rules with source of 0.0.0.0 allow all IP addresses to access your instance. We recommend creating security group rules to allow access from known IP addresses only.



Add inbound network rules to security group (2)

Screenshot of the AWS EC2 instance creation wizard, Step 6: Configure Security Group.

The screenshot shows the configuration of a new security group named "launch-wizard-2". It includes three inbound rules:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8000	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

A warning message at the bottom states: "Warning: Rules with source 0.0.0.0/0 allow all IP addresses to access your instance. We recommend creating security group rules to allow access from known addresses only." Four large green arrows point upwards towards the warning message from the bottom left.

Add name to security group

Screenshot of the AWS EC2 instance creation wizard, Step 6: Configure Security Group.

The page shows the configuration of a new security group named "CNV-ssh+http". A green arrow points to the "Protocol" column of the rule table, highlighting the "TCP" entry for the first rule.

Step 6: Configure Security Group

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: CNV-ssh+http

Description: launch-wiz [2 created 2020-02-12T17:22:03.500+00:00]

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8000	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

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.

Review and Launch

AWS Services Edit

João Coelho Garcia N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, CNV-ssh+http, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
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 [Edit AMI](#)

Amazon Linux AMI 2015.09.2 (HVM), SSD Volume Type - ami-8fceee4e5
Free tier eligible The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

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

Security Groups [Edit security groups](#)

Security group name: CNV-ssh+http
Description: launch-wizard-1 created 2016-03-01T14:27:36.929+00:00

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
Custom TCP rule	TCP	8000	0.0.0.0/0

Instance Details [Edit instance details](#)

Storage [Edit storage](#)

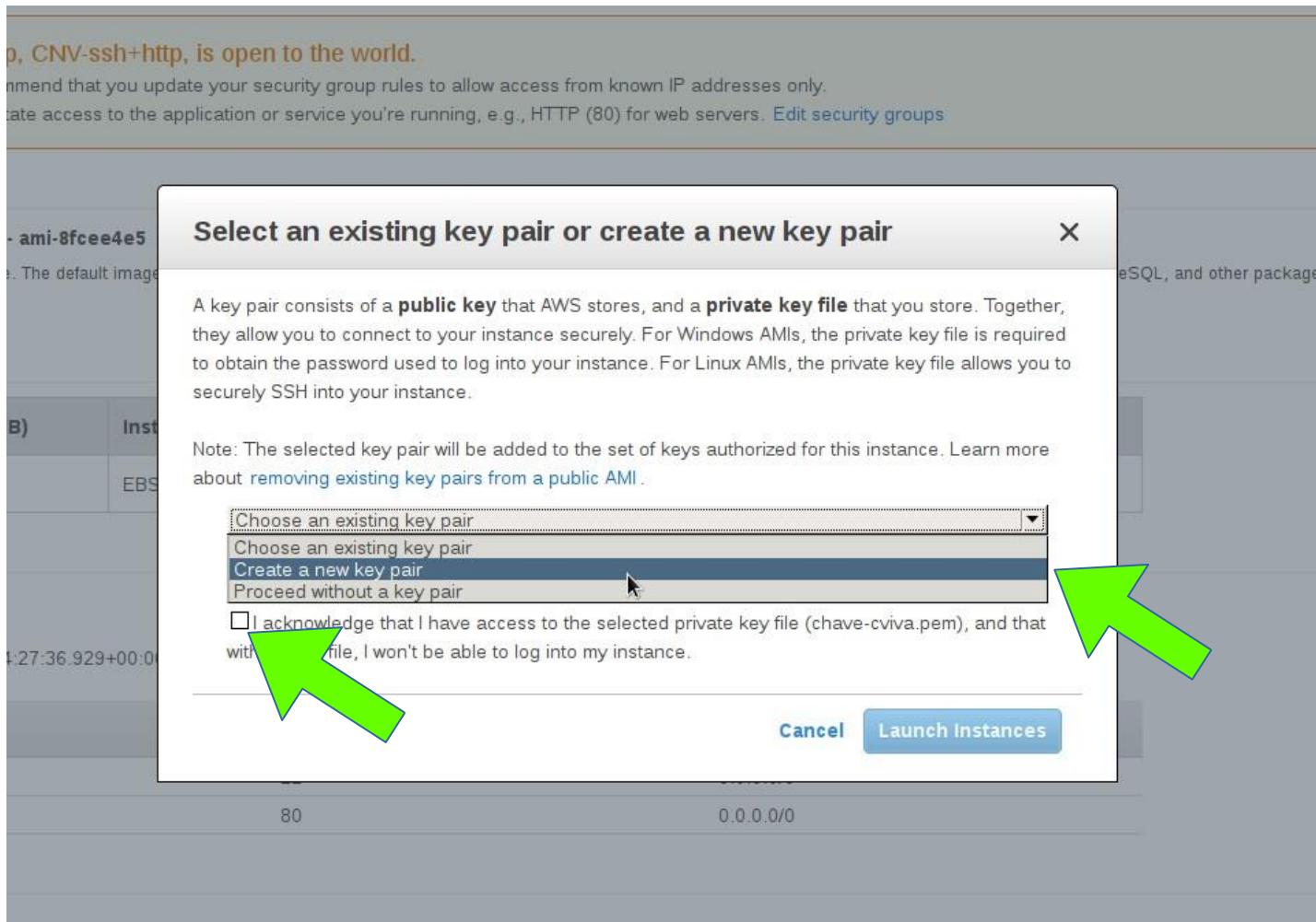
Tags [Edit tags](#)

Launch Define key pair at launch

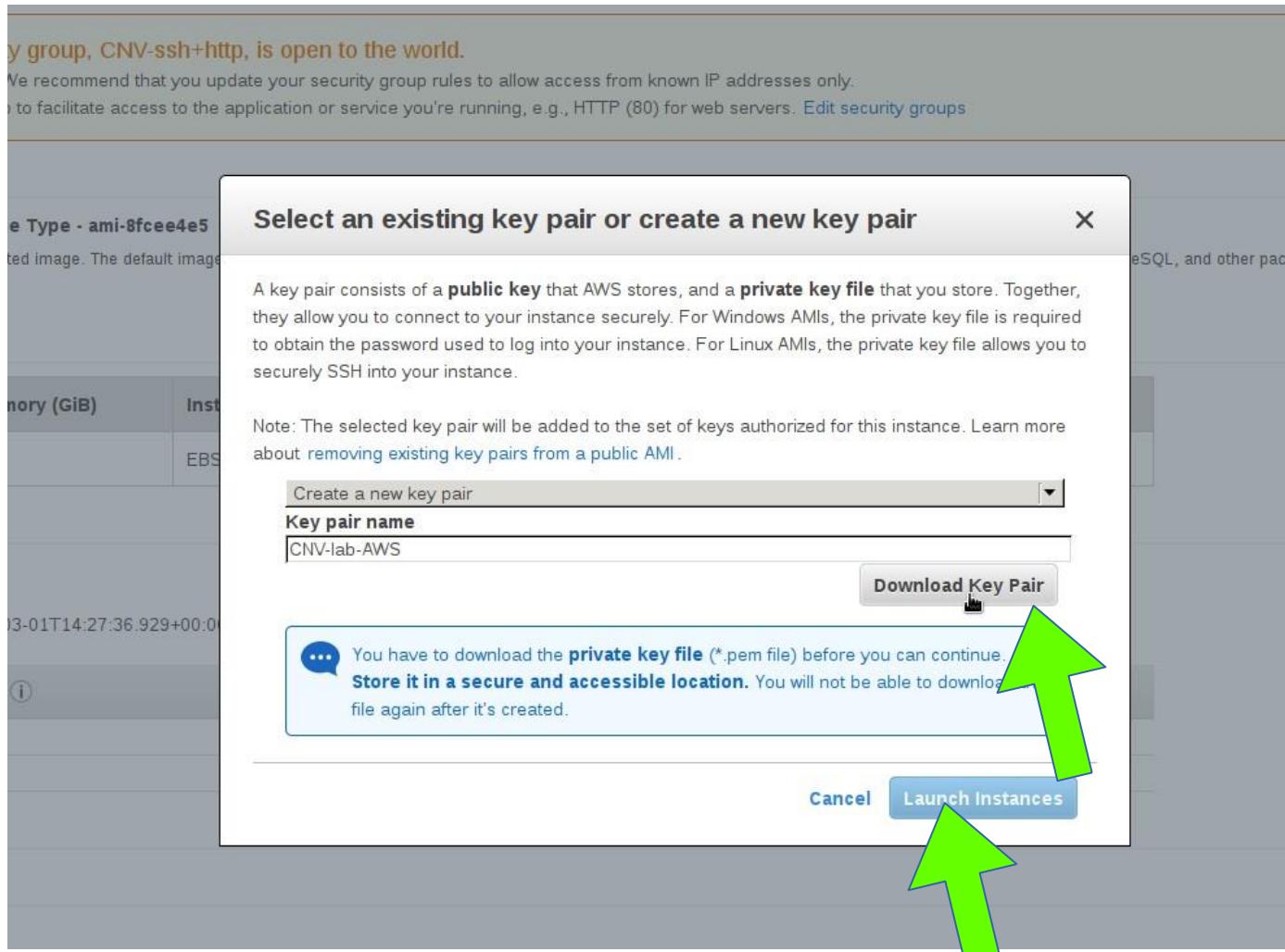
Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



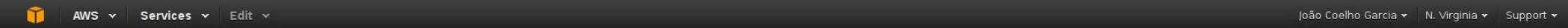
Create a key pair to connect to instance



Download key pair and launch



Instances starting...



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-34d477b0 [View launch log](#)

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

[Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)

[Create and attach additional EBS volumes](#) (Additional charges may apply)

[Manage security groups](#)

[View Instances](#)



Check starting instances

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (which is selected), Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts, Images, AMIs, and Bundle Tasks. Below that are sections for Elastic Block Store (Volumes, Snapshots) and Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces). Further down are Load Balancing (Load Balancers) and Auto Scaling (Launch Configurations, Auto Scaling Groups). At the bottom, there are links for Feedback, English, and other AWS services.

The main content area displays a table of instances. The columns are: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, Public IP, Key Name, Monitoring, and Launch Time. One instance is listed:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
	i-34d477b0	t2.micro	us-east-1a	Pending	Initializing	None		54.210.176.57	CNV-lab-AWS	Enabled	March 1, 2016 at 2:33:44 P...

Below the table, a detailed view for the instance i-34d477b0 is shown. It includes tabs for Description, Status Checks, Monitoring, and Tags. The Description tab is active. The instance details are as follows:

Instance ID	i-34d477b0	Public IP	54.210.176.57
Description	Instance state	pending	
Instance type	t2.micro		
Private DNS	ip-172-30-0-221.ec2.internal		
Private IPs	172.30.0.221		
Secondary private IPs	VPC ID: vpc-ed028988 Subnet ID: subnet-89f953fe		
Public DNS	-		
Public IP	54.210.176.57		
Elastic IP	-		
Availability zone	us-east-1a		
Security groups	CNV-ssh+http . view rules		
Scheduled events	-		
AMI ID	amzn-ami-hvm-2015.09.2.x86_64-gp2 (ami-8fce4e5)		
Platform	-		

At the bottom of the page, there are links for Feedback, English, and other AWS services. The footer contains the text "© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved." and links for Privacy Policy and Terms of Use.

Now it's running!!

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts, 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, and Auto Scaling Groups. The main content area shows a table of instances. One instance is selected, and its details are shown in a modal below. The instance ID is i-34d477b0, Public IP is 54.210.176.57, and its state is running. The modal also displays other details such as Instance Type (t2.micro), Availability Zone (us-east-1a), Status Checks (2/2 checks passed), Alarm Status (None), Public DNS (ip-172-30-0-221.ec2.internal), Private IPs (172.30.0.221), Secondary private IPs (VPC ID: vpc-ed028988, Subnet ID: subnet-89f953fe), Network interfaces (eth0), Source/dest. check (True), ClassicLink (-), EBS-optimized (False), Root device type (ebs), Root device (/dev/xvda), and Block devices (/dev/xvda). The Public DNS field is listed as '-'.

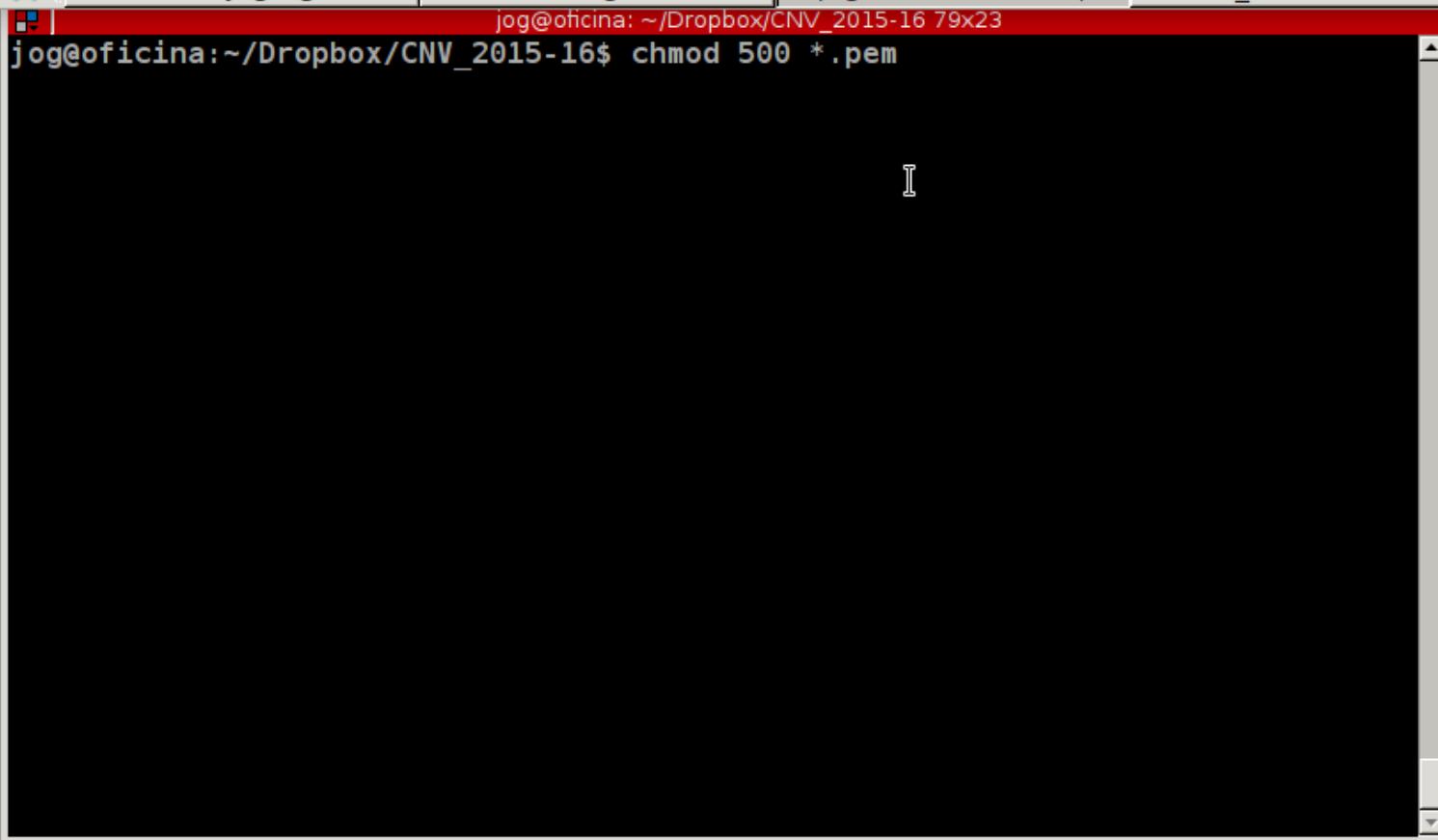
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
	i-34d477b0	t2.micro	us-east-1a	running	2/2 checks passed	None	-	54.210.176.57	CNV-lab-AWS	enabled	March 1, 2016 at 2:33:44 PM UTC (less than one hour)

Instance: i-34d477b0 Public IP: 54.210.176.57

Description		Status Checks	Monitoring	Tags																																																																		
Instance ID	i-34d477b0	Instance state	running	Instance type	t2.micro	Private DNS	ip-172-30-0-221.ec2.internal	Private IPs	172.30.0.221	Secondary private IPs	VPC ID	vpc-ed028988	Subnet ID	subnet-89f953fe	Network interfaces	eth0	Source/dest. check	True	ClassicLink	-	EBS-optimized	False	Root device type	ebs	Root device	/dev/xvda	Block devices	/dev/xvda	Public DNS	-	Public IP	54.210.176.57	Elastic IP	-	Availability zone	us-east-1a	Security groups	CNV-ssh+http . view rules	Scheduled events	No scheduled events	AMI ID	amzn-ami-hvm-2015.09.2.x86_64-gp2 (ami-8fce4e5)	Platform	-	IAM role	-	Key pair name	CNV-lab-AWS	Owner	479132948110	Launch time	March 1, 2016 at 2:33:44 PM UTC (less than one hour)	Termination protection	False	Lifecycle	normal	Monitoring	detailed	Alarm status	None	Kernel ID	-	RAM disk ID	-	Placement group	-	Virtualization	hvm	Reservation	r-2fcfd7bfd

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Decrease permissions on key

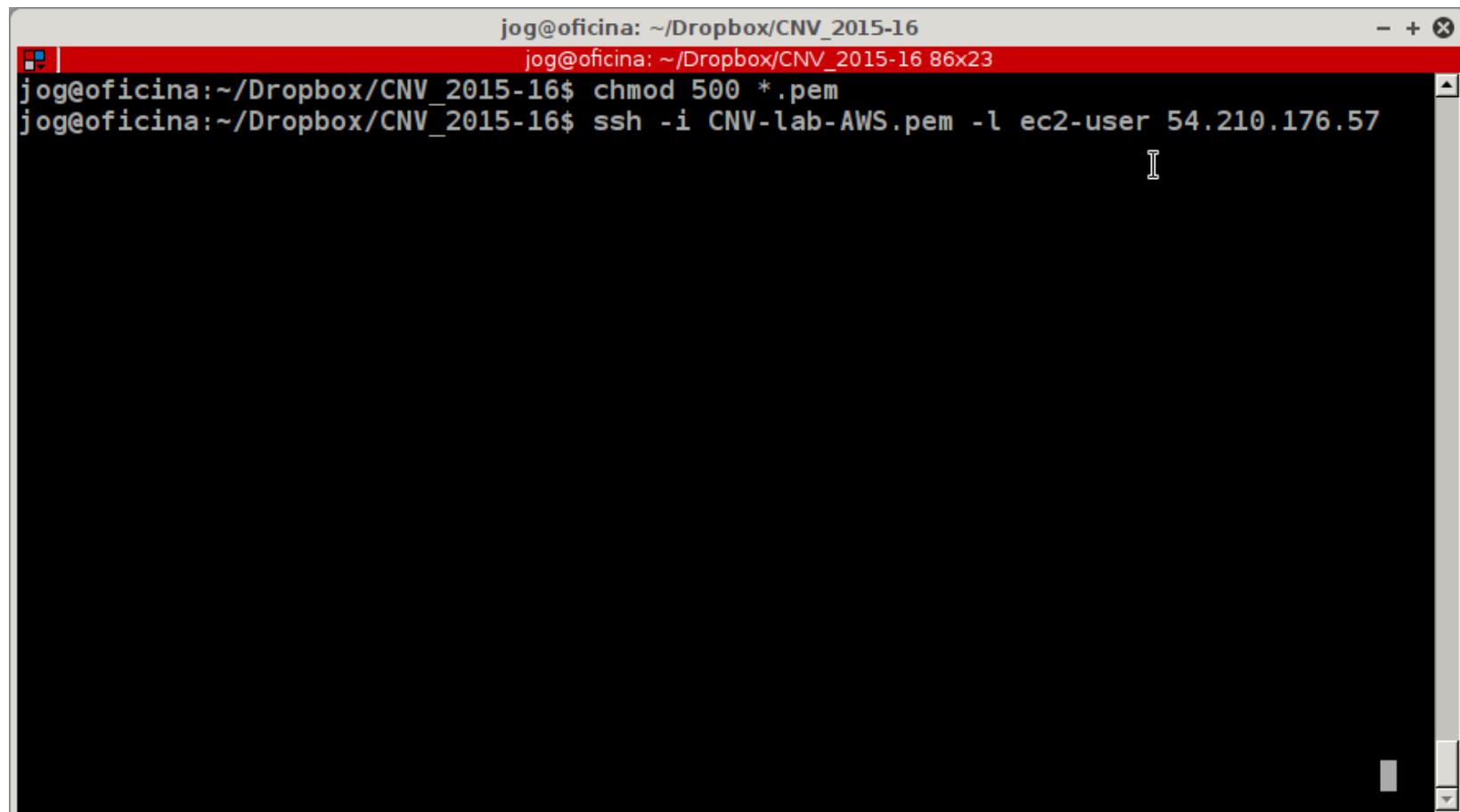


A screenshot of a terminal window titled 'Terminal' with a red header bar. The window shows a command-line interface with the following text:

```
jog@oficina: ~/Dropbox/CNV_2015-16 79x23
jog@oficina:~/Dropbox/CNV_2015-16$ chmod 500 *.pem
```

The terminal window has a dark background and light-colored text. It includes standard terminal controls like scroll bars on the right side.

Ssh into instance



A screenshot of a terminal window titled "jog@oficina: ~/Dropbox/CNV_2015-16". The window has a red header bar. The terminal content shows the user running two commands:

```
jog@oficina: ~/Dropbox/CNV_2015-16$ chmod 500 *.pem
jog@oficina: ~/Dropbox/CNV_2015-16$ ssh -i CNV-lab-AWS.pem -l ec2-user 54.210.176.57
```

Logged in!

```
ec2-user@ip-172-30-0-221:~  
ec2-user@ip-172-30-0-221:~ 86x23  
  
jog@oficina:~/Dropbox/CNV_2015-16$ ssh -i CNV-lab-AWS.pem -l ec2-user 54.210.176.57  
Last login: Tue Mar  1 14:47:53 2016 from oficina.gsd.inesc-id.pt  
  
      _| |_) ) Amazon Linux AMI  
     _| | / | |  
    _\ \ | | |  
  
https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/  
No packages needed for security; 3 packages available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-30-0-221 ~]$
```

Update packages

```
ec2-user@ip-172-30-0-221:~  
ec2-user@ip-172-30-0-221:~ 86x23  
  
jog@oficina:~/Dropbox/CNV_2015-16$ ssh -i CNV-lab-AWS.pem -l ec2-user 54.210.176.57  
Last login: Tue Mar  1 14:47:53 2016 from oficina.gsd.inesc-id.pt  
  
      _| |_) ) Amazon Linux AMI  
     _| | / | |  
  
https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/  
No packages needed for security; 3 packages available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-30-0-221 ~]$ sudo yum update
```

Updated!

```
ec2-user@ip-172-30-0-221:~  
ec2-user@ip-172-30-0-221:~ 86x23  
  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Atualizando: python27-boto-2.39.0-1.0.amzn1.noarch          1/6  
  Atualizando: tzdata-java-2016a-1.36.amzn1.noarch           2/6  
  Atualizando: tzdata-2016a-1.36.amzn1.noarch               3/6  
  Limpeza       : python27-boto-2.38.0-1.7.amzn1.noarch      4/6  
  Limpeza       : tzdata-java-2015g-1.35.amzn1.noarch        5/6  
  Limpeza       : tzdata-2015g-1.35.amzn1.noarch             6/6  
  Verificando   : tzdata-2016a-1.36.amzn1.noarch           1/6  
  Verificando   : tzdata-java-2016a-1.36.amzn1.noarch        2/6  
  Verificando   : python27-boto-2.39.0-1.0.amzn1.noarch      3/6  
  Verificando   : python27-boto-2.38.0-1.7.amzn1.noarch        4/6  
  Verificando   : tzdata-2015g-1.35.amzn1.noarch             5/6  
  Verificando   : tzdata-java-2015g-1.35.amzn1.noarch         6/6  
  
Atualizado:  
  python27-boto.noarch 0:2.39.0-1.0.amzn1      tzdata.noarch 0:2016a-1.36.amzn1  
  tzdata-java.noarch 0:2016a-1.36.amzn1  
  
Complete!  
[ec2-user@ip-172-30-0-221 ~]$
```

Install Java SDK

```
ec2-user@ip-172-30-0-221:~  
ec2-user@ip-172-30-0-221:~ 86x26  
Linux ip-172-30-0-221 4.1.17-22.30.amzn1.x86_64 #1 SMP Fri Feb 5 23:44:22 UTC 2016 x86_64 x86_64 GNU/Linux  
[ec2-user@ip-172-30-0-221 ~]$ sudo yum search jdk  
Loaded plugins: priorities, update-motd, upgrade-helper  
===== N/S matched: jdk =====  
java-1.6.0-openjdk.x86_64 : OpenJDK Runtime Environment  
java-1.6.0-openjdk-demo.x86_64 : OpenJDK Demos  
java-1.6.0-openjdk-devel.x86_64 : OpenJDK Development Environment  
java-1.6.0-openjdk-javadoc.x86_64 : OpenJDK API Documentation  
java-1.6.0-openjdk-src.x86_64 : OpenJDK Source Bundle  
java-1.7.0-openjdk.x86_64 : OpenJDK Runtime Environment  
java-1.7.0-openjdk-demo.x86_64 : OpenJDK Demos  
java-1.7.0-openjdk-devel.x86_64 : OpenJDK Development Environment  
java-1.7.0-openjdk-javadoc.noarch : OpenJDK API Documentation  
java-1.7.0-openjdk-src.x86_64 : OpenJDK Source Bundle  
java-1.8.0-openjdk.x86_64 : OpenJDK Runtime Environment  
java-1.8.0-openjdk-demo.x86_64 : OpenJDK Demos  
java-1.8.0-openjdk-devel.x86_64 : OpenJDK Development Environment  
java-1.8.0-openjdk-headless.x86_64 : OpenJDK Runtime Environment  
java-1.8.0-openjdk-javadoc.noarch : OpenJDK API Documentation  
java-1.8.0-openjdk-src.x86_64 : OpenJDK Source Bundle  
ldapjdk-javadoc.noarch : Javadoc for ldapjdk  
ldapjdk.noarch : The Mozilla LDAP Java SDK  
  
Name and summary matches only, use "search all" for everything.  
[ec2-user@ip-172-30-0-221 ~]$ sudo yum install java-1.7.0-openjdk-devel.x86_64
```

Setup your Application Software

- At this point, setup up the Java application that will be used (transfer it using scp or develop it at the AWS instance directly)
- Don't forget to make sure that the application server is running at startup, e.g. by launching it inside /etc/rc.local or an equivalent mechanism
- e.g., `java -cp ~ec2-user WebServer`
- *(you will need sudo rights)*

Once the web server is running at startup (e.g. /etc/rc.local), create image.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various service links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area shows a table of instances. A specific instance, i-34d477b0, is selected. A context menu is open over this instance, with the 'Image' option highlighted. A large green arrow points from the bottom left towards the 'Create Image' option in the menu. The 'Create Image' option is also highlighted with a yellow box. The table below shows detailed information for the selected instance, including its configuration, network interfaces, and monitoring status.

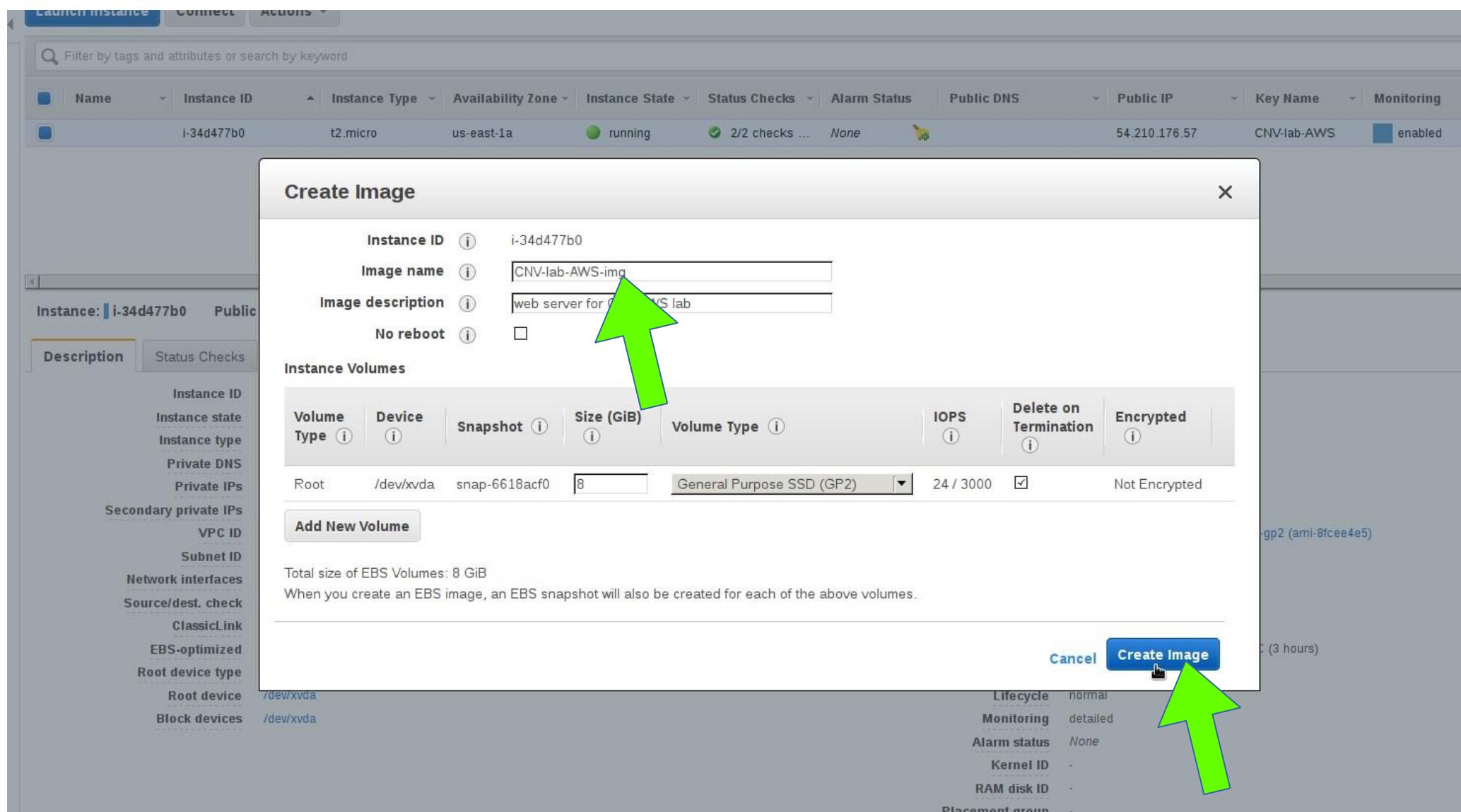
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
i-34d477b0	t2.micro	us-east-1a	running	/2 checks ...	None			54.210.176.57	CNV-lab-AWS	enabled

Instance: i-34d477b0 Public IP: 54.210.176.57

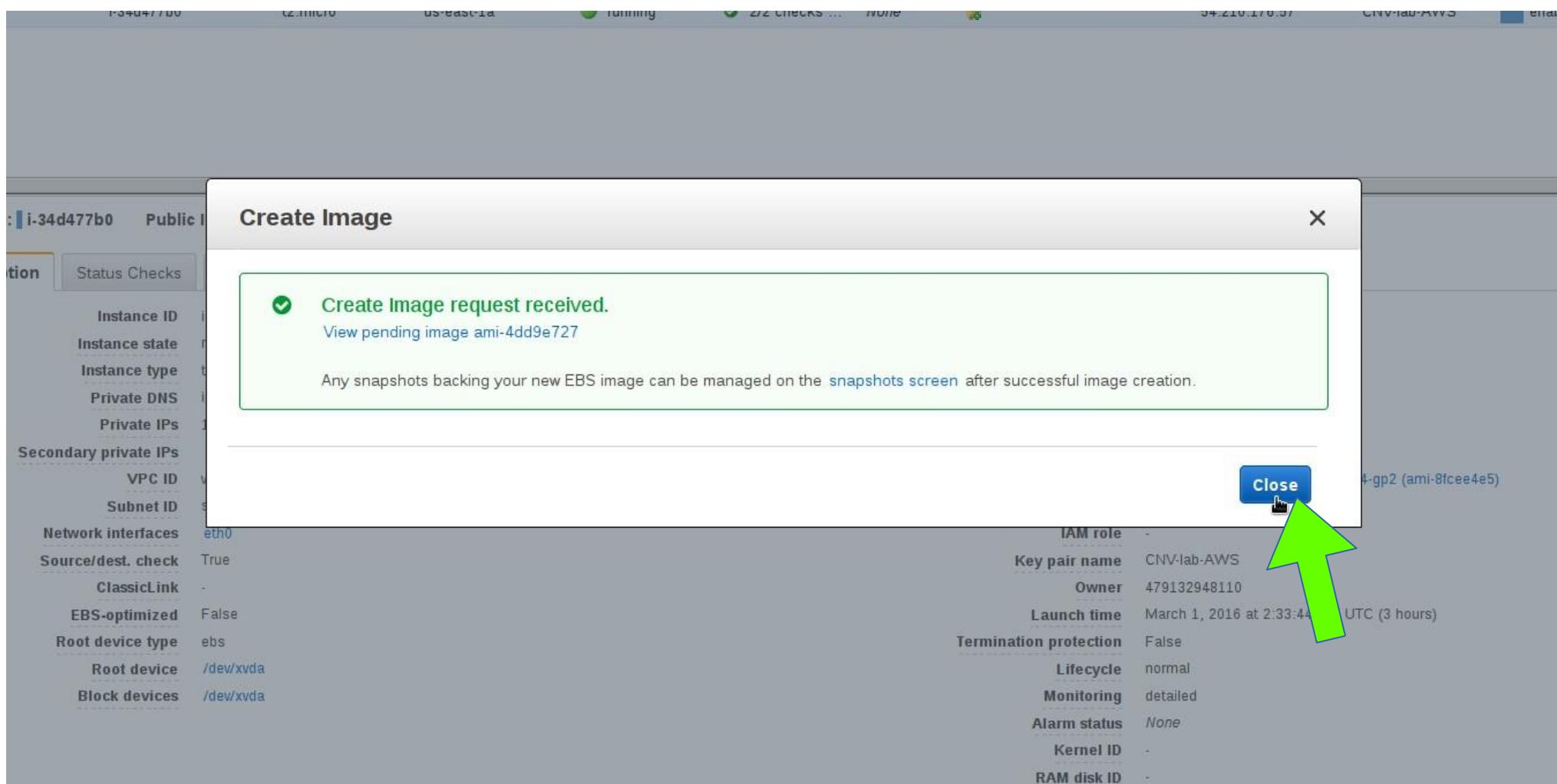
Description	Status Checks	Monitoring	Tags
Instance ID: i-34d477b0 Instance state: running Instance type: t2.micro Private DNS: ip-172-30-0-221.ec2.internal Private IPs: 172.30.0.221 Secondary private IPs: VPC ID: vpc-ed028988 Subnet ID: subnet-89f953fe Network interfaces: eth0 Source/dest. check: True ClassicLink: - EBS-optimized: False Root device type: ebs Root device: /dev/xvda Block devices: /dev/xvda			

Public DNS: -
Public IP: 54.210.176.57
Elastic IP: -
Availability zone: us-east-1a
Security groups: CNV-ssh+http . view rules
Scheduled events: No scheduled events
AMI ID: amzn-ami-hvm-2015.09.2.x86_64-gp2 (ami-8fce4e5)
Platform: -
IAM role: -
Key pair name: CNV-lab-AWS
Owner: 479132948110
Launch time: March 1, 2016 at 2:33:44 PM UTC (3 hours)
Termination protection: False
Lifecycle: normal
Monitoring: detailed
Alarm status: None
Kernel ID: -

Give it a name and create it



It may take a little time for the request to be fulfilled



Once the image is created, right click the instance and terminate it.

The screenshot shows the AWS EC2 Dashboard. On the left, the navigation menu includes 'Instances' (selected), 'Spot Requests', 'Reserved Instances', 'Scheduled Instances', 'Commands', 'Dedicated Hosts', 'AMIs', 'Bundle Tasks', 'Elastic Block Store', 'Volumes', 'Snapshots', 'Network & Security', 'Security Groups', 'Elastic IPs', 'Placement Groups', 'Key Pairs', 'Network Interfaces', 'Load Balancing', 'Load Balancers', and 'Auto Scaling'. The main content area displays a table of instances, with one row selected for instance i-34d477b0. A modal dialog box titled 'Stop Instances' is open, asking 'Are you sure you want to stop these instances?'. It lists the instance ID i-34d477b0 and contains a note: 'Note that when your instances are stopped: Any data on the ephemeral storage of your instances will be lost.' A green arrow points to the 'Yes, Stop' button at the bottom right of the dialog. The background table shows details for the selected instance, including its Public IP (54.210.176.57), Instance Type (t2.micro), and State (running). The 'Actions' tab is selected in the top navigation bar.

It's terminated...

Let's create a load balancer...

The screenshot shows the AWS EC2 Dashboard. On the left, there is a sidebar with the following navigation:

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits
- INSTANCES**
 - Instances** (selected)
 - Spot Requests
 - Reserved Instances
 - Scheduled Instances
 - Commands
 - 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** (selected)
 - Auto Scaling
- AUTO SCALING**
 - Launch Configurations
 - Auto Scaling Groups

The main content area displays a table of instances. One instance, **i-34d477b0**, is shown with the status **terminated**. A large green arrow points from the sidebar's **Load Balancers** link towards this instance. Below the table, a detailed view for the selected instance is provided.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-34d477b0	t2.micro	us-east-1a	terminated	None		

Instance: i-34d477b0 Private IP: 172.30.0.221

Description **Status Checks** **Monitoring** **Tags**

Instance ID	i-34d477b0	Public DNS	-
Instance state	stopped	Public IP	-
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-172-30-0-221.ec2.internal	Availability zone	us-east-1a
Private IPs	172.30.0.221	Security groups	CNV-ssh+http . view rules
Secondary private IPs		Scheduled events	-
VPC ID	vpc-ed028988	AMI ID	amzn-ami-hvm-2015.09.2.x86_64-gp2 (ami-8fce4e5)
Subnet ID	subnet-89f953fe	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	CNV-lab-AWS
ClassicLink	-	Owner	479132948110
EBS-optimized	False	Launch time	March 1, 2016 at 2:33:44 PM UTC (3 hours)
Root device type	ebs	Termination protection	False
Root device	/dev/xvda	Lifecycle	normal
Block devices	/dev/xvda	Monitoring	detailed
		Alarm status	None
		Kernel ID	-
		RAM disk ID	-
		Placement group	-
		Virtualization	hvm

<https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers;> © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Click on create...

The screenshot shows the AWS EC2 Management Console interface. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Help. The 'Load Balancing' section is currently selected, indicated by an orange vertical bar next to the 'Load Balancers' link. The main content area displays a message stating, "You do not have any load balancers in this region." Below this message is a call-to-action: "To learn about Elastic Load Balancing, see our [FAQ](#) and [Getting Started Guide](#). Click 'Create Load Balancer' to create a load balancer that distributes traffic across your instances." At the top of this section, there is a blue button labeled "Create Load Balancer". A large green arrow points upwards towards this button. The browser's address bar shows the URL: <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers>.

Pick a classic Load Balancer

Screenshot of the AWS Elastic Load Balancing "Select load balancer type" page.

The page shows three options:

- Application Load Balancer**: Handles HTTP and HTTPS traffic. A green arrow points to the "Create" button.
- Network Load Balancer**: Handles TCP traffic. A green arrow points to the "Create" button.
- Classic Load Balancer**: Handles HTTP, HTTPS, and TCP traffic. A large green arrow points to the "Create" button.

Classic Load Balancer is labeled as "PREVIOUS GENERATION" and is described as being suitable for existing applications running in the EC2 network.

Create buttons are located at the bottom of each section.

Cancel

Give it a name..

AWS Services Edit João Coelho Garcia 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 1: Define 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: CNV-lab-AWS-LB
Create LB Inside: vpc-ed028988 (172.30.0.0/16)
Create an internal load balancer: (what's this?)

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	8000

Add

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-ed028988 (172.30.0.0/16)

! Please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

Available Subnets				
Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
+	us-east-1b	subnet-eb5e8bb2	172.30.2.0/24	
+	us-east-1d	subnet-b5ea1e9e	172.30.3.0/24	
+	us-east-1e	subnet-209d0f1a	172.30.1.0/24	

Selected Subnets				
Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
-	us-east-1a	subnet-89f953fe	172.30.0.0/24	

Cancel Next: Assign Security Groups

Create it inside your default VPC

Screenshot of the AWS Load Balancer creation wizard, Step 1: Define 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: CNV-lab-AWS-LB

Create LB Inside: vpc-ed028988 (172.30.0.0/16) 

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

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	8000

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-ed028988 (172.30.0.0/16)

Available Subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-east-1b	subnet-eb5e8bb2	172.30.2.0/24	
	us-east-1d	subnet-b5ea1e9e	172.30.3.0/24	
	us-east-1e	subnet-209d0f1a	172.30.1.0/24	

Selected Subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-east-1a	subnet-89f953fe	172.30.0.0/24	

Next: Assign Security Groups

Connect external and internal traffic

Screenshot of the AWS Load Balancer creation wizard, Step 1: Define Load Balancer.

The screenshot shows the configuration for a new load balancer named "CNV-lab-AWS-LB". The "Create LB Inside:" dropdown is set to "vpc-ed028988 (172.30.0.0/16)". The "Listener Configuration" table has one entry: Load Balancer Protocol: HTTP, Load Balancer Port: 80, Instance Protocol: HTTP, Instance Port: 8000. A large green arrow points upwards from the "Selected Subnets" section towards the "Listener Configuration" table.

Step 1: Define 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: CNV-lab-AWS-LB

Create LB Inside: vpc-ed028988 (172.30.0.0/16)

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

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	8000

Add

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-ed028988 (172.30.0.0/16)

! Please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

Available Subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-east-1b	subnet-eb5e8bb2	172.30.2.0/24	
	us-east-1d	subnet-b5ea1e9e	172.30.3.0/24	
	us-east-1e	subnet-209d0f1a	172.30.1.0/24	

Selected Subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-east-1a	subnet-89f953fe	172.30.0.0/24	

Cancel Next: Assign Security Groups

Add a subnet

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: CNV-lab-AWS-LB
Create LB Inside: vpc-ed028988 (172.30.0.0/16)
Create an internal load balancer: (what's this?)

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	8000

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-ed028988 (172.30.0.0/16)

Available Subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-east-1b	subnet-eb5e8bb2	172.30.2.0/24	
	us-east-1d	subnet-b5ea1e9e	172.30.3.0/24	
	us-east-1e	subnet-209d0f1a	172.30.1.0/24	

Selected Subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
	us-east-1a	subnet-89f953fe	172.30.0.0/24	

Next: Assign Security Groups

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy

Create a New Security Group

AWS Services Resource Groups CNV2018 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 2: Assign Security Groups

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.

Assign a security group: Create a new security group Select an existing security group

Security group name: CNV-lab-Loadbalancer

Description: quick-create-1 created on Tuesday, March 13, 2018 at 1:07:41 PM UTC

Type	Protocol	Port Range	Source
Custom TCP F	TCP	80	Custom 0.0.0.0/0

Add Rule

Cancel Previous Next: Configure Security Settings

Ignore this warning..

The screenshot shows a step-by-step wizard for creating a load balancer. The current step is "Step 3: Configure Security Settings". A warning message is displayed: "Improve your load balancer's security. Your load balancer is not using any secure listener." It suggests using HTTPS or SSL for front-end connections and provides links to "Basic Configuration" and "Configure Health Check". At the bottom, there are "Cancel", "Previous", and "Next: Configure Health Check" buttons. A green arrow points to the "Next" button.

AWS Services Edit João Coelho Garcia N. Virginia

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

Step 3: Configure Security Settings

⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.

If your traffic to the load balancer needs to be secure, use either the HTTPS or the SSL protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under [Basic Configuration](#) section. You can also continue with current settings.

Cancel Previous Next: Configure Health Check

Feedback English Privacy Policy Terms of Use

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Configure the health check

The screenshot shows the AWS EC2 Management Console interface for creating a new load balancer. The current step is "4. Configure Health Check".

Ping Protocol: HTTP

Ping Port: 8000

Ping Path: /test

Advanced Details:

- Response Timeout:** 5 seconds
- Health Check Interval:** 30 seconds
- Unhealthy Threshold:** 2
- Healthy Threshold:** 10

Buttons at the bottom:

- Cancel
- Previous
- Next: Add EC2 Instances

Page footer:

- Feedback
- English
- © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
- Privacy Policy
- Terms of Use

Don't add instances

The screenshot shows the AWS EC2 Management Console interface for creating a Load Balancer. The current step is "Step 5: Add EC2 Instances".

VPC vpc-ed028988 (172.30.0.0/16)

Instance	Name	State	Security Group	Zone	Subnet ID	Subnet CIDR
i-34d477b0		stopped	CNV-ssh+http	us-east-1a	subnet-89f953fe	172.30.0.0/24
i-933bb217		running	CNV-ssh+http	us-east-1a	subnet-89f953fe	172.30.0.0/24

Availability Zone Distribution

Enable Cross-Zone Load Balancing (i)

Enable Connection Draining (i) 300 seconds

Buttons: Cancel, Previous, **Next: Add Tags** (highlighted with a green arrow), and Done.

Page Footer: Feedback, English, © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved., Privacy Policy, Terms & Conditions, Help.

You don't need tags

The screenshot shows a step-by-step wizard for creating a new AWS Lambda function. The current step is "Step 6: Add Tags".

At the top, there's a navigation bar with the AWS logo, "AWS Services Edit", and user information "João Coelho Garcia N. Virginia Sup". Below the navigation bar, a horizontal menu bar shows steps 1 through 5.

The main content area is titled "Step 6: Add Tags" and contains the following text:

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.

Below this text is a table with two columns: "Key" and "Value". There is one row in the table with empty input fields. To the right of the "Value" field is a small "X" icon.

At the bottom left of the table is a "Create Tag" button.

At the bottom right of the page are three buttons: "Cancel", "Previous", and "Review and Create". A large green arrow points upwards from the bottom right towards the "Review and Create" button.

At the very bottom of the page is a footer with links: "Feedback", "English", "© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.", "Privacy Policy", and "Terms of Use".

Review and Create

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

Step 7: Review

Please review the load balancer details before continuing

Define Load Balancer [Edit load balancer definition](#)

Load Balancer name: CNV-lab-AWS-LB
Scheme: internet-facing
Port Configuration: 80 (HTTP) forwarding to 8000 (HTTP)

Configure Health Check [Edit health check](#)

Ping Target: HTTP:8000/test
Timeout: 5 seconds
Interval: 30 seconds
Unhealthy Threshold: 2
Healthy Threshold: 10

Add EC2 Instances [Edit instances](#)

Cross-Zone Load Balancing: Enabled

[Cancel](#) [Previous](#) [Create](#)

[Feedback](#) [English](#) © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



Load Balancer Creation Status

Successfully created load balancer

Load balancer CPI*-lab-A .S-LB as successfully created

Note It may take a few minutes for your instances to become active in the new load balancer

[Close](#)



Create a Launch Configuration

The screenshot shows the AWS Management Console interface for creating a Launch Configuration. On the left, the navigation sidebar lists various services under the EC2 category, including Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, IMAGES (AMIs), ELASTIC BLOCK STORE (Volumes, Snapshots, Lifecycle Manager), NETWORK & SECURITY (Security Groups, Elastic IPs, Placement Groups, Key Pairs), LOAD BALANCING (Load Balancers, Target Groups), and AUTO SCALING (Launch Configurations, Auto Scaling Groups). The 'Launch Configurations' link under AUTO SCALING is highlighted with an orange color. The main content area is titled 'Launch configurations (0)' and contains a search bar, a table header with columns for Name, AMI ID, Instance type, Spot price, and Creation time, and a message stating 'No launch configurations found in this region.' A prominent orange button labeled 'Create launch configuration' is located at the bottom of this section. A large green arrow points from the top right towards this button.

Fill name, AMI, Instance type...

The screenshot shows the 'Create launch configuration' wizard in the AWS EC2 console. The left sidebar lists various services under 'AWS Services'. The main form has the following sections:

- Launch configuration name**: A text input field containing "cnv-launchconfig". A large green arrow points to this field.
- Amazon machine image (AMI)**: A dropdown menu showing "cnv-webserver". A large green arrow points to this field.
- Instance type**: A text input field showing "t2.micro (1 vCPUs, 1 GiB, EBS Only)" and a "Choose instance type" button. A large green arrow points to this field.
- Additional configuration - optional**: This section includes:
 - Purchasing option: A checkbox for "Request Spot Instances" which is unchecked.
 - IAM instance profile: A dropdown menu showing "Select IAM role".
 - Monitoring: A checkbox for "Enable EC2 instance detailed monitoring within CloudWatch" which is checked.

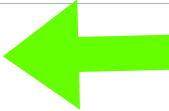
Enable Monitoring...

The screenshot shows the AWS CloudFormation console interface for creating a new launch configuration. The left sidebar lists various AWS services like Spot Requests, Savings Plans, and Auto Scaling. The main panel is titled "Additional configuration - optional". It includes sections for "Purchasing option" (Request Spot Instances), "IAM Instance profile" (Select IAM role dropdown), "Monitoring" (checkbox checked for "Enable EC2 Instance detailed monitoring within CloudWatch"), and "EBS-optimized instance" (Launch as EBS-optimized instance checkbox). A large green arrow points to the "Monitoring" section. Below these are "Advanced details" and a note about creating new launch configurations. The bottom section is titled "Storage (volumes)" and shows an "EBS volumes" table with one row: Root volume type General purpose SSD (gp2), device /dev/xvda, snapshot snap-02eec88a295f60263, and size 8 GiB. There's also a "+ Add new volume" button and a note about free tier storage.

Additional configuration - *optional*

Purchasing option [Info](#)
 Request Spot Instances

IAM Instance profile [Info](#)
Select IAM role

Monitoring [Info](#)
 Enable EC2 Instance detailed monitoring within CloudWatch 

EBS-optimized instance
 Launch as EBS-optimized instance

► 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.

Storage (volumes) [Info](#)

	Volume type	Devices	Snapshot	Size (GiB)	Volume type
<input type="checkbox"/>	Root	/dev/xvda	snap-02eec88a295f60263	8	General purpose SSD (gp2)

+ 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.

Select the existing security group and key pair. Finish creation.

The screenshot shows the AWS Management Console interface for creating a new instance. The left sidebar navigation bar includes options like Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, IMAGES (AMIs), ELASTIC BLOCK STORE (Volumes, Snapshots, Lifecycle Manager), NETWORK & SECURITY (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), LOAD BALANCING (Load Balancers, Target Groups), and AUTO SCALING (Launch Configurations, Auto Scaling Groups). The main content area displays the 'Security groups' section. It has a heading 'Assign a security group' with two radio button options: 'Create a new security group' (unchecked) and 'Select an existing security group' (checked). Below this is a table titled 'Security groups' with columns: Security group ID, Name, VPC ID, and Description. Three rows are listed: 1) sg-01ecb35e7fcc88572 (selected, highlighted with a blue border), CNV-ssh+http, vpc-37fe424a, launch-wizard-1 created 2021-03-21T15:34:35.678+00:00; 2) sg-0d357070b5d431505, cnv-loadbalander, vpc-37fe424a, quick-create-1 created on Sunday, March 21, 2021 at 6:41:35 PM UTC; 3) sg-7ed41470, default, vpc-37fe424a, default VPC security group. A warning message at the bottom of this table states: '⚠ 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 specific IP addresses only.' A large green arrow points from the top right towards this warning message. The bottom section of the page shows the 'Key pair (login)' configuration, with a dropdown for 'Key pair options' set to 'Choose an existing key pair' and a dropdown for 'Existing key pair' set to 'cnv-keypair'. A checkbox at the bottom of this section is checked and reads: 'I acknowledge that I have access to the selected private key file (cnv-keypair.pem), and that without this file, I won't be able to log into my instance.'

You should see the new-ly created launch configuration

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar lists various services under the EC2 category, including Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, AMIs, Volumes, Snapshots, Lifecycle Manager, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancers, Target Groups, and Auto Scaling groups. The main content area displays a success message: "Successfully created launch configuration: cnv-launchconfig". Below this, the "Launch configurations" list shows one entry:

Name	AMI ID	Instance type	Spot price	Creation time
cnv-launchconfig	ami-0b506a680e...	t2.micro	-	Tue Mar 23 2021 12:22:42 GMT+0000 (Western European Standard Time)

Create an Auto Scaling group

Screenshot of the AWS EC2 Auto Scaling homepage. A large green arrow points to the "Create Auto Scaling group" button.

Amazon EC2 Auto Scaling
helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

How it works

An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical unit. You configure settings for a group and its instances as well as define the group's minimum, maximum, and desired capacity. Setting different minimum and maximum capacity values forms the bounds of the group, which allows the group to scale as the load on your application spikes higher or lower, based on demand. To scale the Auto Scaling group, you can either make manual adjustments to the desired capacity or let Amazon EC2 Auto Scaling automatically add and remove capacity to meet

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

Create Auto Scaling group

Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started

- What Is Amazon EC2 Auto Scaling?
- Getting started with Amazon EC2 Auto Scaling
- Set up a scaled and load-balanced application
- FAQ

Fill the name and select the previously created launch config...

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 1: Choose launch template or configuration. The left sidebar lists various EC2 services like Instances, AMIs, and Network & Security. The main pane shows a step-by-step process:

- Step 1: Choose launch template or configuration**: An input field for the Auto Scaling group name contains "cnv-scaling".
- Step 2: Configure settings**: This step is collapsed.
- Step 3 (optional): Configure advanced options**: This step is collapsed.
- Step 4 (optional): Configure group size and scaling policies**: This step is collapsed.
- Step 5 (optional): Add notifications**: This step is collapsed.
- Step 6 (optional): Add tags**: This step is collapsed.
- Step 7: Review**: This step is collapsed.

Launch configuration: A dropdown menu shows "cnv-launchconfig". Other options include "Create a launch configuration" and a "Switch to launch template" link. The "cnv-launchconfig" entry is selected, showing details:

- Launch configuration: cnv-launchconfig
- AMI ID: ami-0b506a680ee01029d
- Date created: Tue Mar 23 2021 12:22:42 GMT+0000 (Western European Standard Time)
- Security groups: sg-01ecb35e7fcc88572
- Instance type: t2.micro
- Key pair name: -

At the bottom right are "Cancel" and "Next" buttons.

Select the exiting VPC and subnet...

The screenshot shows the AWS EC2 Auto Scaling group creation process at Step 2: Configure settings. The 'Network' section is highlighted with two large green arrows pointing to the VPC and Subnets dropdown menus.

Configure settings

Configure the settings below. Depending on whether you chose a launch template, these settings may include options to help you make optimal use of EC2 resources.

Network

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your Instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

vpc-37fe424a
172.31.0.0/16 Default

Create a VPC

Subnets

Select subnets

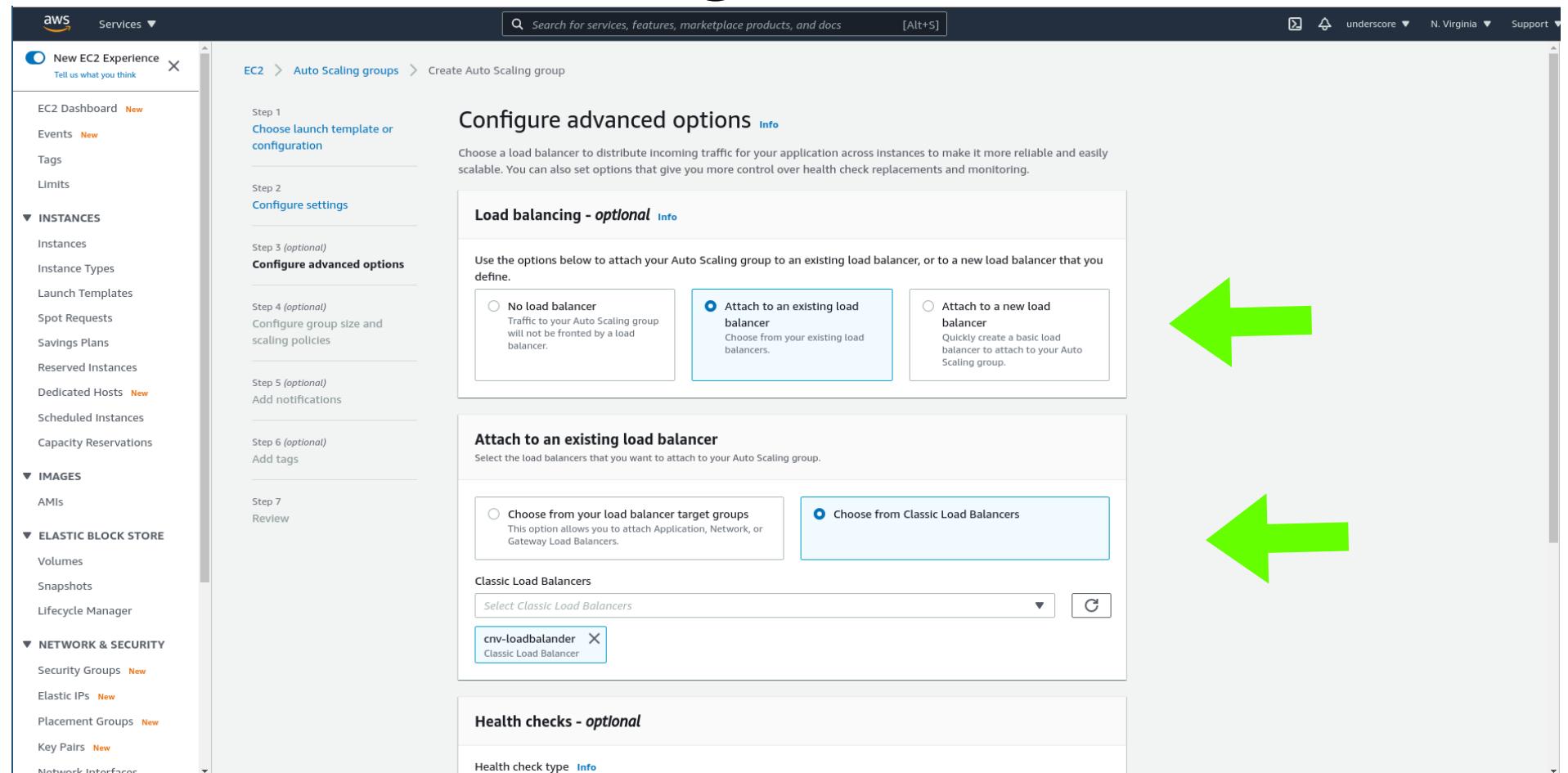
us-east-1a | subnet-050a3a48
172.31.16.0/20 Default

Create a subnet

Cancel Previous Skip to review Next

The left sidebar shows the navigation menu for the EC2 service, including Instances, Images, and Elastic Block Store sections.

Select the exiting load balander...



The screenshot shows the AWS Management Console interface for creating an Auto Scaling group. The left sidebar lists various AWS services like EC2, Lambda, and S3. The main area is titled "Create Auto Scaling group" and is on "Step 1: Choose launch template or configuration".

In the "Configure advanced options" section, there's a note about choosing a load balancer to distribute traffic. Three options are shown:

- No load balancer: Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer: Choose from your existing load balancers. This option is highlighted with a blue border.
- Attach to a new load balancer: Quickly create a basic load balancer to attach to your Auto Scaling group.

Below this, the "Attach to an existing load balancer" section shows a list of target groups and classic load balancers. The "Choose from Classic Load Balancers" option is also highlighted with a blue border. A green arrow points to this section.

At the bottom, there's a "Health checks - optional" section with a note about health check type.

Select ELB health checks with 60s grace period. Enable monitoring...

The screenshot shows the AWS EC2 Auto Scaling configuration interface, specifically Step 6 (optional). The left sidebar lists various EC2 services like Instances, AMIs, and Network & Security. The main panel has sections for 'Attach to an existing load balancer', 'Health checks - optional', and 'Additional settings - optional'. In the 'Health checks - optional' section, there's a note about replacing failed instances with EC2 or ELB health checks. A green arrow points to the 'ELB' checkbox. Another green arrow points to the '60 seconds' input field for the 'Health check grace period'. A third green arrow points to the 'Enable group metrics collection within CloudWatch' checkbox in the 'Monitoring' section. At the bottom, there are 'Cancel', 'Previous', 'Skip to review', and 'Next' buttons.

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Search for services, features, marketplace products, and docs [Alt+S]

Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Classic Load Balancers
Select Classic Load Balancers

cnv-loadbalancer

cnv-loadbalancer
Classic Load Balancer

Health checks - optional

Health check type [Info](#)
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

EC2 ELB

Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

60 seconds

Additional settings - optional

Monitoring [Info](#)
 Enable group metrics collection within CloudWatch

Cancel Previous Skip to review Next

Pick a max capacity (eg, 2). Select no scaling policies (for now)...

The screenshot shows the 'Create Auto Scaling group' wizard in the AWS EC2 console. The left sidebar lists various EC2 services like Instances, AMIs, and Network & Security. The main page has a breadcrumb trail: EC2 > Auto Scaling groups > Create Auto Scaling group. It's Step 4 (optional): Configure group size and scaling policies.

Configure group size and scaling policies

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - optional

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity: 1

Minimum capacity: 1

Maximum capacity: 2

Scaling policies - optional

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

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

None

Instance scale-in protection - optional

Two large green arrows point to the 'Maximum capacity' field and the 'None' radio button, highlighting them as the key configuration points for the task.

No notifications needed...

The screenshot shows the AWS EC2 service interface. On the left, there's a sidebar with various navigation links under categories like Instances, Images, and Network & Security. The main content area is titled "Create Auto Scaling group" and is currently on "Step 1: Choose launch template or configuration". A sub-section titled "Add notifications" is highlighted with a blue border. Below it, a note says: "Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group." There is a large "Add notification" button. To the right of the main content, there are buttons for "Cancel", "Previous", "Skip to review" (which is in red), and "Next".

No tags needed...

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 3 (optional). The left sidebar lists various EC2 services like Instances, AMIs, and Network & Security. The main pane shows the 'Add tags' step, which is optional for creating an Auto Scaling group. A callout box explains that tags can be added to instances and their attached EBS volumes by specifying tags in the launch template. It also notes that tag values from the launch template will be overridden if there are any duplicates. Below this, a 'Tags (0)' section shows an 'Add tag' button and indicates 50 remaining tags. Navigation buttons for 'Cancel', 'Previous', and 'Next' are at the bottom.

New EC2 Experience X

Tell us what you think

EC2 Dashboard New

Events New

Tags

Limits

INSTANCES

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts New
- Scheduled Instances
- Capacity Reservations

IMAGES

- AMIs

ELASTIC BLOCK STORE

- Volumes
- Snapshots
- Lifecycle Manager

NETWORK & SECURITY

- Security Groups New
- Elastic IPs New
- Placement Groups New
- Key Pairs New
- Network Interfaces

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Add tags Info

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

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

Tags (0)

Add tag

50 remaining

Cancel Previous Next

After review and create, you should see the new scaling group.

The screenshot shows the AWS EC2 Auto Scaling groups page. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces). The main content area displays the 'Auto Scaling groups (1)' section. A search bar at the top of this section allows searching for 'Auto Scaling groups'. Below it is a table with the following data:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
cnv-scaling	cnv-launchconfig	1	-	1	1	2	us-east-1a

At the top right of the main content area, there are buttons for 'Edit', 'Delete', and 'Create an Auto Scaling group'. The top navigation bar includes the AWS logo, services dropdown, search bar ('Search for services, features, marketplace products, and docs'), and account information ('underscore', 'N. Virginia', 'Support').

Create two Alarms in CloudWatch

The screenshot shows the AWS Management Console Services page. On the left, there's a sidebar with sections for Favorites, Recently visited, and a long list of services starting with EC2, Lightsail, Lambda, and Batch. A large green arrow points from the text "Create two Alarms in CloudWatch" down to the "CloudWatch" link in the Recently visited section. The main content area lists all AWS services under categories like Compute, Storage, Database, Migration & Transfer, Customer Enablement, Blockchain, Satellite, Quantum Technologies, Management & Governance, AWS Auto Scaling, CloudFormation, Config, OpsWorks, Service Catalog, Systems Manager, AWS AppConfig, Trusted Advisor, Control Tower, AWS License Manager, AWS Well-Architected Tool, Personal Health Dashboard, AWS Chatbot, Launch Wizard, Machine Learning, Amazon SageMaker, Amazon Augmented AI, Amazon CodeGuru, Amazon DevOps Guru, Amazon Comprehend, Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, Amazon Translate, AWS DeepComposer, AWS DeepLens, AWS DeepRacer, AWS Panorama, Amazon Monitron, Amazon HealthLake, Amazon Lookout for Vision, Amazon Lookout for Equipment, Amazon Lookout for Metrics, Analytics, Athena, Amazon Redshift, EMR, CloudSearch, Elasticsearch Service, Kinesis, Front-end Web & Mobile, AWS Amplify, Mobile Hub, AWS AppSync, Device Farm, Amazon Location Service, AR & VR, Amazon Sumerian, Application Integration, Step Functions, Amazon AppFlow, Amazon EventBridge, Amazon MQ, Simple Notification Service, Simple Queue Service, SWF, Managed Apache Airflow, AWS Cost Management, AWS Cost Explorer, AWS Budgets, AWS Marketplace Subscriptions, and Business Applications, Amazon Connect, Amazon Pinpoint, Amazon Honeycode, Amazon Chime, Amazon Simple Email Service, Amazon WorkDocs, and Amazon WorkMail.

Screenshot of the AWS CloudWatch Alarms page.

The left sidebar shows the CloudWatch navigation menu:

- Dashboards
- Alarms** (selected)
- In alarm (0)
- Insufficient data (0)
- OK (0)
- Billing
- Logs
- Log groups
- Insights
- Metrics
- Explorer (New)
- Events
- Rules
- Event Buses
- ServiceLens
- Service Map
- Traces
- Container Insights (New)
- Resources
- Performance monitoring
- Lambda Insights (New)
- Performance monitoring
- Synthetics
- Canaries
- Contributor Insights
- Settings
- Favorites
- + Add a dashboard

The main content area displays the Alarms (0) page. It includes a search bar, filters for State and Type, and a prominent orange "Create alarm" button. A large green arrow points to the "Create alarm" button.

AWS Services ▾

CloudWatch > Alarms > Create alarm

Step 1 Specify metric and conditions

Step 2 Configure actions

Step 3 Add name and description

Step 4 Preview and create

Search for services, features, marketplace products, and docs [Alt+S]

Specify metric and conditions

Metric

Graph

Preview of the metric or metric expression to define the alarm threshold.

Select metric

Cancel Next



Select ‘CPUUtilization’ metric for the previously created scaling group...

The screenshot shows the AWS CloudWatch Metrics 'Select metric' dialog. At the top, there's a graph titled 'Untitled graph' showing CPUUtilization over time. Below the graph, the 'Metrics' section lists two metrics: 'CPUUtilization' and 'cnv-scaling'. A green arrow points to the 'cnv-scaling' entry, which has a checkbox next to it that is checked. The 'Graph search' and 'Graphed metrics (1)' buttons are visible at the bottom right of the metrics list.

CloudWatch

Services ▾

CloudWatch Alarms Create alarm

Search for services, features, marketplace products, and docs [Alt+S]

1h 3h 12h 1d 3d 1w Custom Line 10s

Percent

1.86

1.1

0.332

09:40 09:45 09:50 09:55 10:00 10:05 10:10 10:15 10:20 10:25 10:30 10:35 10:40 10:45 10:50 10:55 11:00 11:05 11:10 11:15 11:20 11:25 11:30 11:35 11:40 11:45 11:50 11:55 12:00 12:05 12:10 12:15 12:20 12:25 12:30 12:35 12:40

CPUUtilization

Metrics (2) Info

All > EC2 > By Auto Scaling Group

Graph search Graphed metrics (1)

CPUUtilization

AutoScalingGroupName (2)

cnv-autoscaling

cnv-scaling

Metric Name

CPUUtilization

CPUUtilization

Select metric

Select 'Average' for statistic and 'Period' (e.g., 1 minute)

Screenshot of the AWS CloudWatch 'Create alarm' wizard - Step 1: Specify metric and conditions.

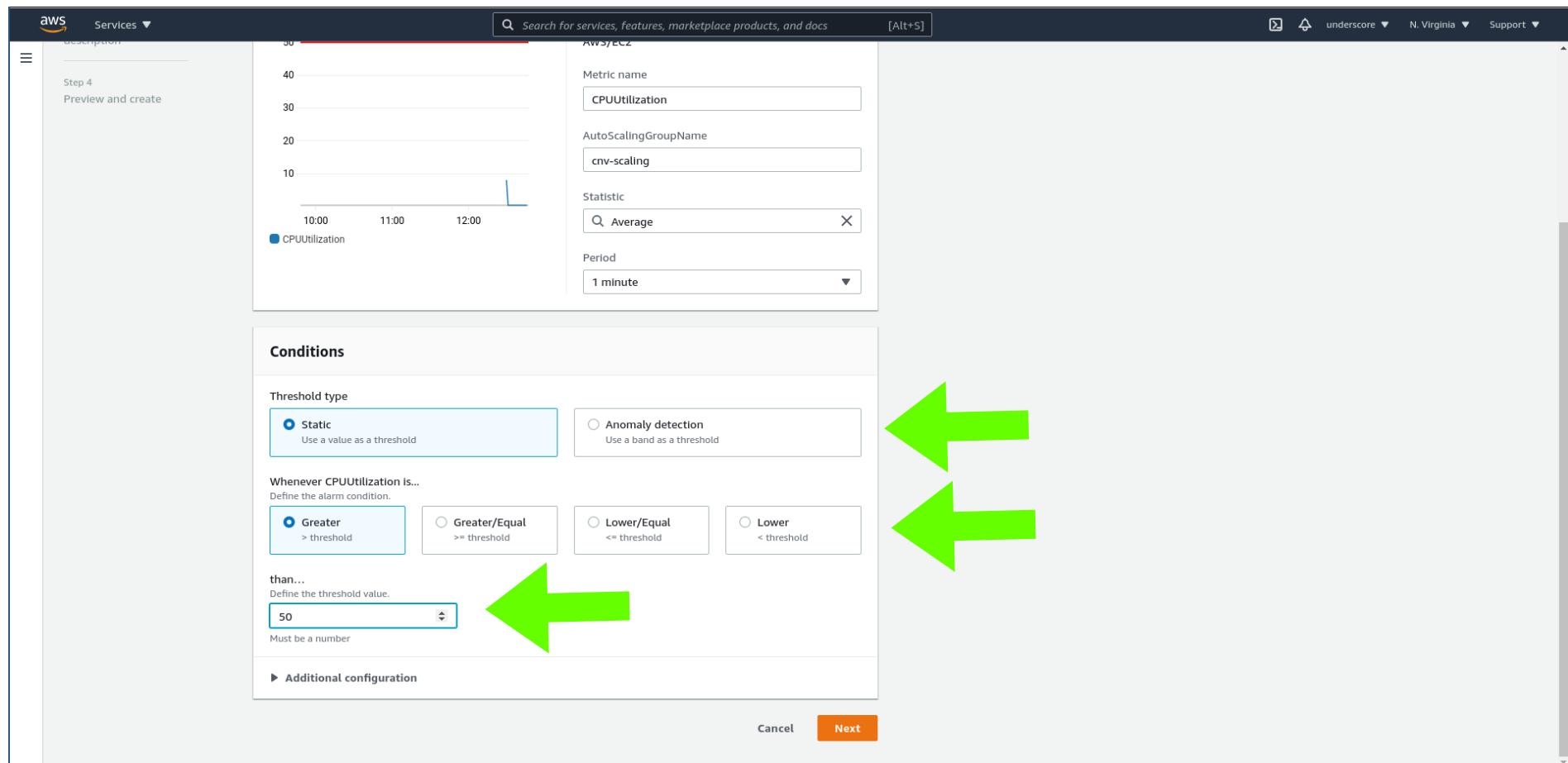
The 'Metric' section shows a graph of CPUUtilization over time, with a blue line above a red threshold line. The graph spans from 10:00 to 12:00. The 'Metric' details are as follows:

- Namespace: AWS/EC2
- Metric name: **cpuutilization** (highlighted with a red box)
- AutoScalingGroupName: cnv-scaling
- Statistic: **Average** (highlighted with a red box)
- Period: **1 minute** (highlighted with a red box)

Two large green arrows point to the 'Statistic' and 'Period' fields, indicating they are the key configuration points being highlighted.

The 'Conditions' section below shows the threshold type selected as 'Static' (radio button highlighted with a red box). The condition is defined as 'Whenever CPUUtilization is... Greater than threshold'.

Static threshold, greater, 50% (i.e. alarm goes off if avg CPU > 50%)



No notifications needed...

The screenshot shows the AWS CloudWatch Alarms 'Create alarm' configuration page. On the left, the CloudWatch navigation pane is visible, showing various services like Dashboards, Alarms, Logs, Metrics, Events, ServiceLens, Container Insights, Lambda Insights, Synthetics, Favorites, and a '+ Add a dashboard' option. The 'Alarms' section is currently selected.

The main area is titled 'Configure actions'. It starts with 'Step 1: Specify metric and conditions' (which is collapsed), followed by 'Step 2: Configure actions' (which is expanded). Under 'Step 2', there are three sections: 'Notification', 'Auto Scaling action', and 'EC2 action'. The 'Notification' section contains the following configuration:

- Alarm state trigger:** 'In alarm' is selected (radio button is checked).
 - Description: 'The metric or expression is outside of the defined threshold.'
- OK:** 'OK' is also listed as an option.
 - Description: 'The metric or expression is within the defined threshold.'
- Insufficient data:** 'Insufficient data' is listed as an option.
 - Description: 'The alarm has just started or not enough data is available.'

Next to the 'In alarm' trigger, there is a large green arrow pointing towards the 'Remove' button. Below the triggers, there are options for selecting an SNS topic: 'Select an existing SNS topic' (selected), 'Create new topic', and 'Use topic ARN'. There is also a field to 'Send a notification to...' with a placeholder 'Select an email list' and a note 'Only email lists for this account are available.' At the bottom of the 'Notification' section is a 'Add notification' button.

The 'Auto Scaling action' and 'EC2 action' sections are currently empty, each with a single 'Add [Action] action' button.

Screenshot of the AWS CloudWatch Alarms 'Create alarm' configuration page. The left sidebar shows the CloudWatch navigation menu with 'Alarms' selected. The main content area is titled 'Configure actions' and lists four steps: Step 1 (Specify metric and conditions), Step 2 (Configure actions), Step 3 (Add name and description), and Step 4 (Preview and create). Step 2 is currently active, showing sections for 'Notification', 'Auto Scaling action', 'EC2 action', and 'Systems Manager OpsCenter action'. A large green arrow points from the bottom right towards the 'Next' button at the bottom right of the page.



Give a name to the alarm...

Screenshot of the AWS CloudWatch Create alarm wizard, Step 3: Add name and description.

The left sidebar shows the CloudWatch navigation menu with the 'Alarms' section selected. The main area displays the 'Add name and description' step, which includes fields for 'Alarm name' and 'Alarm description - optional'. The 'Alarm name' field contains the value 'HighCPUUtilizationAlarm'. The 'Alarm description - optional' field is empty. At the bottom right of the step panel are 'Cancel', 'Previous', and 'Next' buttons. Two large green arrows are overlaid on the image: one pointing to the 'Alarm name' input field, and another pointing to the 'Next' button.

Screenshot of the AWS CloudWatch Alarms creation wizard.

The sidebar shows the CloudWatch navigation menu:

- CloudWatch
- Dashboards
- Alarms In alarm 0
- Insufficient data 0
- OK 0
- Billing
- Logs
- Log groups
- Insights
- Metrics
- Explorer New
- Events
- Rules
- Event Buses
- ServiceLens
- Service Map
- Traces
- Container Insights New
- Resources
- Performance monitoring
- Lambda Insights New
- Performance monitoring
- Synthetics
- Canaries
- Contributor Insights
- Settings
- Favorites
- + Add a dashboard

The main content area shows the creation steps:

- Step 1: Set conditions**

Conditions

Threshold type: Static

Whenever **HighCPUUtilization** is Greater (>) than... 50

Additional configuration
- Step 2: Configure actions**

Actions

No actions

You don't have any actions for this alarm.
- Step 3: Add name and description**

Name and description

Name: HighCPUUtilizationAlarm

Description: -

Buttons at the bottom:

- Cancel
- Previous
- Create alarm Create alarm



After repeating for a low CPU utilization alarm...

Successfully created alarm [LowCPUUtilizationAlarm](#).

CloudWatch > Alarms

Alarms (2)

Name	State	Last state update	Conditions	Actions
LowCPUUtilizationAlarm	Insufficient data	2021-03-23 12:51:54	CPUUtilization < 25 for 1 datapoints within 1 minute	No actions
HighCPUUtilizationAlarm	OK	2021-03-23 12:50:26	CPUUtilization > 50 for 1 datapoints within 1 minute	No actions

Repeat the previous steps to create a second alarm that tracks low CPU utilization.

Create two Scaling Policies in EC2

Go back to EC2 and select the previously created Auto Scaling group.

The screenshot shows the AWS EC2 Auto Scaling groups page. On the left, there's a navigation sidebar with sections like EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main area displays a table titled "Auto Scaling groups (1)". The table has columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. One row is visible for "cnv-scaling" with "cnv-launchconfig" as the launch template, 1 instance, and 1 desired capacity. At the top right of the table are buttons for Edit, Delete, and Create an Auto Scaling group. A large green arrow points from the text "Go back to EC2 and select the previously created Auto Scaling group." towards the "Edit" button.

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
cnv-scaling	cnv-launchconfig	1	-	1	1	2	us-east-1a

The screenshot shows the AWS Management Console interface for the Auto Scaling service. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area shows the 'Auto Scaling groups' section, with the 'Automatic scaling' tab selected. A green arrow points down to the 'Scaling policies (0)' section, which displays a message: 'No scaling policies are currently specified' and a 'Add policy' button. Another green arrow points up to the 'Scheduled actions (0)' section, which also displays a message: 'No scheduled actions are currently specified' and a 'Create scheduled action' button.

Select ‘Step scaling’, give a name to the policy and select the alarm...

The screenshot shows the 'Create scaling policy' wizard in the AWS Management Console. The left sidebar lists various EC2 services like Instances, AMIs, and Network & Security. The main form is titled 'Create scaling policy' and has the following fields:

- Policy type:** Step scaling
- Scaling policy name:** IncreaseGroupSize
- CloudWatch alarm:** HighCPUUtilizationAlarm (selected from a dropdown) with a note: "Choose an alarm that can scale capacity whenever: HighCPUUtilizationAlarm". A link "Create a CloudWatch alarm" is provided.
- Take the action:** An 'Add' button is selected, showing one step: "1 capacity units when 50 <= CPUUtilization < +infinity". An "Add step" button is also present.
- Instances need:** 500 seconds warm up before including in metric

A callout box with a green arrow points to the 'Create' button at the bottom right of the form. Another callout box contains the text: "Create the scaling policy using the previously created Alarm."

Repeat for the other alarm

The screenshot shows the AWS Management Console interface for creating a scaling policy. The left sidebar lists various services like EC2 Dashboard, Events, Tags, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area is titled "Create scaling policy" under "Auto Scaling groups". It shows a "Step scaling" policy type, a scaling policy name "DecreaseGroupSize", and a CloudWatch alarm named "LowCPUUtilizationAlarm". The alarm condition is set to trigger when CPUUtilization >= 25 for 1 consecutive periods of 60 seconds. Below this, there's a section for "Take the action" where a step is defined to remove capacity units when the alarm triggers. A large green arrow points from the bottom right towards the "Create" button at the bottom right of the form. A callout box on the right side of the screen contains the text: "Create the scaling policy using the previously created Alarm."

EC2 Dashboard [New](#)

Events [New](#)

Tags

Limits

INSTANCES

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts [New](#)
- Scheduled Instances
- Capacity Reservations

IMAGES

- AMIs

ELASTIC BLOCK STORE

- Volumes
- Snapshots
- Lifecycle Manager

NETWORK & SECURITY

- Security Groups [New](#)
- Elastic IPs [New](#)
- Placement Groups [New](#)
- Key Pairs [New](#)
- Network Interfaces

LOAD BALANCING

EC2 > Auto Scaling groups > cnv-scaling

Create scaling policy

Policy type: Step scaling

Scaling policy name: DecreaseGroupSize

CloudWatch alarm: LowCPUUtilizationAlarm

Choose an alarm that can scale capacity whenever:

breaches the alarm threshold: CPUUtilization < 25 for 1 consecutive periods of 60 seconds for the metric dimensions:

AutoScalingGroupName = cnv-scaling

Take the action:

Remove

capacity units when 25 >= CPUUtilization > -infinity

Add step

Cancel **Create**

Create the scaling policy using the previously created Alarm.

You should see now both policies...

The screenshot shows the AWS Auto Scaling Groups page for the 'cnv-scaling' group. The 'Automatic scaling' tab is selected. A success message box is displayed: 'Scaling policy created or edited successfully'. The 'Scaling policies' section shows two entries:

- DecreaseGroupSize**: Policy type: Step scaling. Enabled or disabled?: Enabled. Execute policy when: LowCPUUtilizationAlarm. breaches the alarm threshold: CPUUtilization < 25 for 1 consecutive periods of 60 seconds for the metric dimensions: AutoScalingGroupName = cnv-scaling. Take the action: Remove 1 capacity units when 25 >= CPUUtilization > -infinity.
- IncreaseGroupSize**: Policy type: Step scaling. Enabled or disabled?: Enabled. Execute policy when: HighCPUUtilizationAlarm. breaches the alarm threshold: CPUUtilization > 50 for 1 consecutive periods of 60 seconds for the metric dimensions: AutoScalingGroupName = cnv-scaling. Take the action: Add 1 capacity units when 50 <= CPUUtilization < +infinity. Instances need: 300 seconds to warm up after each step.

The 'Scheduled actions' section is currently empty, showing '(0)'.

Try it!

Wait for an instance of the Auto-Scale group to initialize

Generate (lots of) accesses to your web server cluster

(e.g. several browser tabs)

Attention: Before you leave

Shut your Auto-Scaler down!

Don't waste credits!

Good work in the Cloud!