



TÉCNICO LISBOA



Cloud Computing and Virtualization

Lab 2 - Introduction to AWS

(Amazon Web Services)

Outline

- **Virtual Machines**
 - using the **AWS web console** and programmatically

How to launch an AWS EC2 Instance

The screenshot shows the AWS Console Home page. A yellow arrow points to the 'Recently visited' section on the left, which lists services like EC2, IAM, CloudWatch, Lambda, CloudFormation, S3, and DynamoDB. To the right, there's a section for 'Applications' with a message saying 'No applications'. A callout box with a blue border and a blue arrow points to the 'Region: US East (N. Virginia)' dropdown in the top right of the application section, with the text 'Note the region'.

Console Home Info

Recently visited Info

- EC2
- IAM
- CloudWatch
- Lambda
- CloudFormation
- S3
- DynamoDB

Elastic Container Service

Reset to default layout [Alt+S] + Add widgets

Applications (0) Info

Region: US East (N. Virginia)

us-east-1 (Current Region) ▼ Find applications

Name Description Region Originating account

No applications
Get started by creating an application.

Create application

Go to myApplications

Welcome to AWS

Getting started with AWS Info
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification Info
Learn from AWS experts and advance your skills and knowledge.

AWS Health Info

Open issues 0 Past 7 days

Scheduled changes 0 Upcoming and past 7 days

Other notifications 0 Past 7 days

Cost and usage Info

Data unavailable
You must have Cost Explorer enabled to view your cost and usage data.

Turn on Cost Explorer

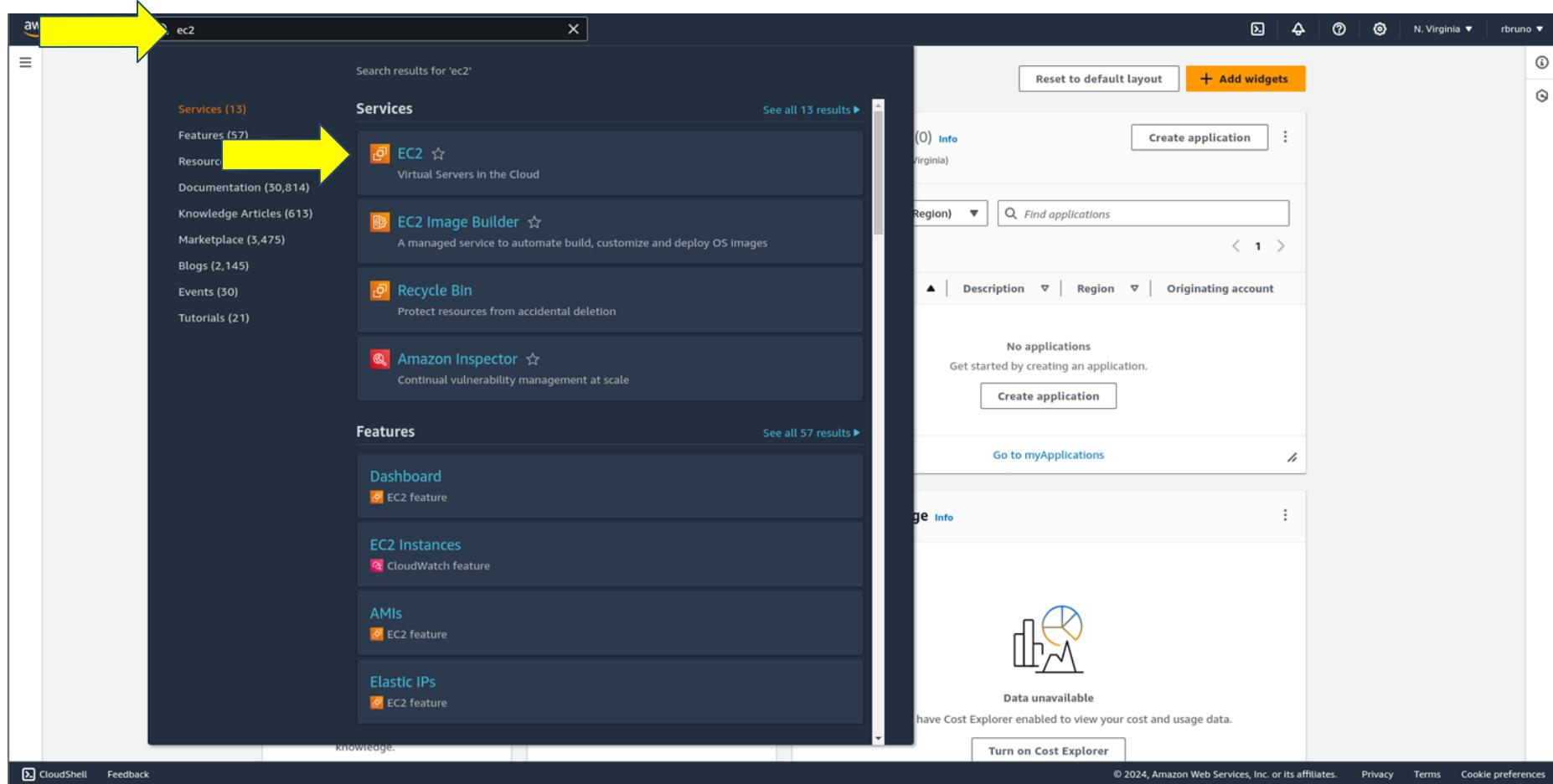
CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

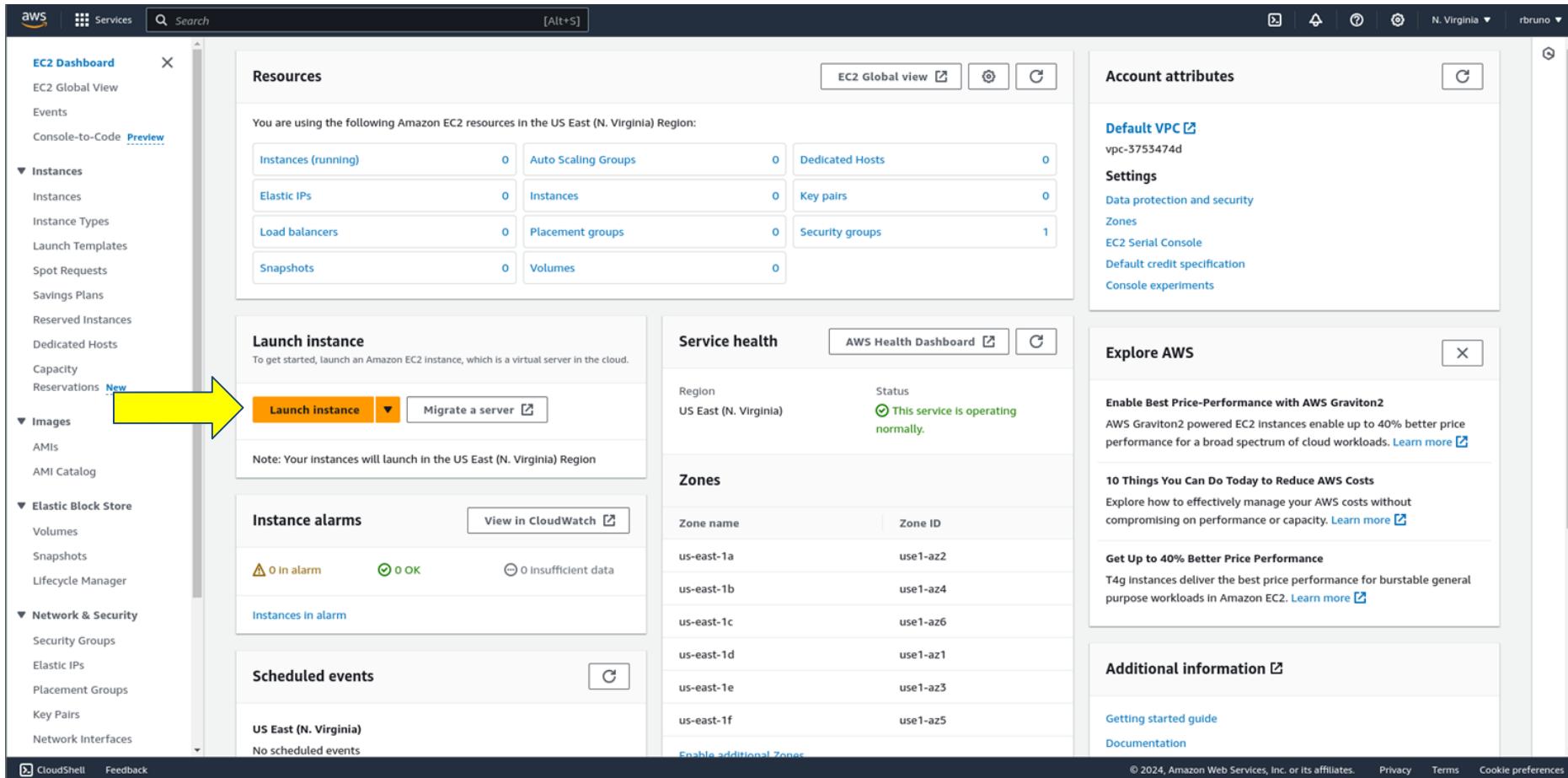
Note the region

After login...

How to launch an AWS EC2 Instance



How to launch an AWS EC2 Instance



The screenshot shows the AWS EC2 Dashboard in the US East (N. Virginia) Region. The left sidebar contains navigation links for EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and other services. A yellow arrow points to the 'Launch instance' button in the 'Launch instance' section of the main content area.

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

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	1
Snapshots	0	Volumes	0		

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance **Migrate a server**
Note: Your Instances will launch in the US East (N. Virginia) Region

Service health **AWS Health Dashboard**
Region: US East (N. Virginia) Status: This service is operating normally.

Zones

Zone name	Zone ID
us-east-1a	use1-az2
us-east-1b	use1-az4
us-east-1c	use1-az6
us-east-1d	use1-az1
us-east-1e	use1-az3
us-east-1f	use1-az5

Explore AWS

- Enable Best Price-Performance with AWS Graviton2
- 10 Things You Can Do Today to Reduce AWS Costs
- Get Up to 40% Better Price Performance

Additional information

- Getting started guide
- Documentation

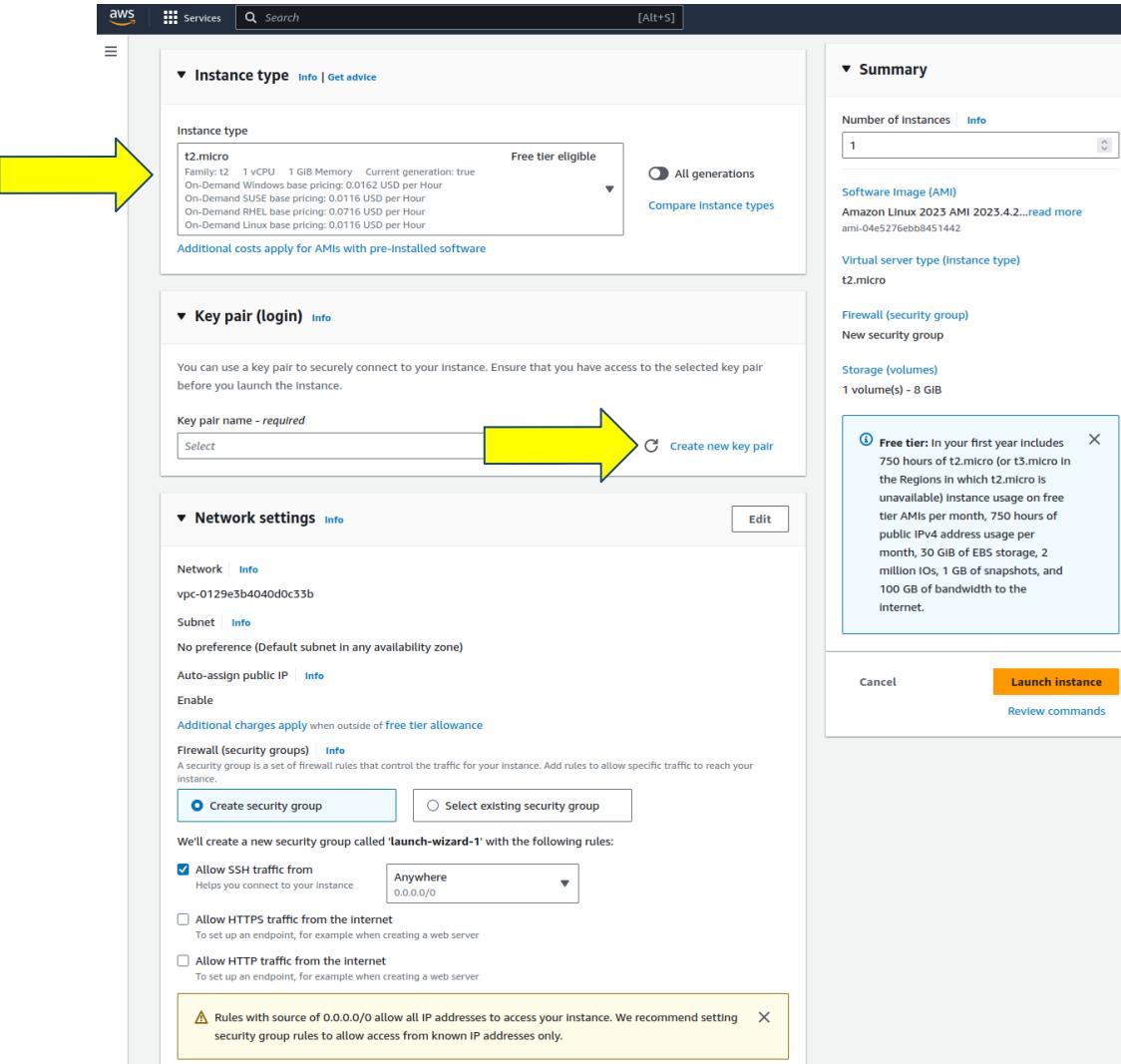
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Pick instance type, image, architecture

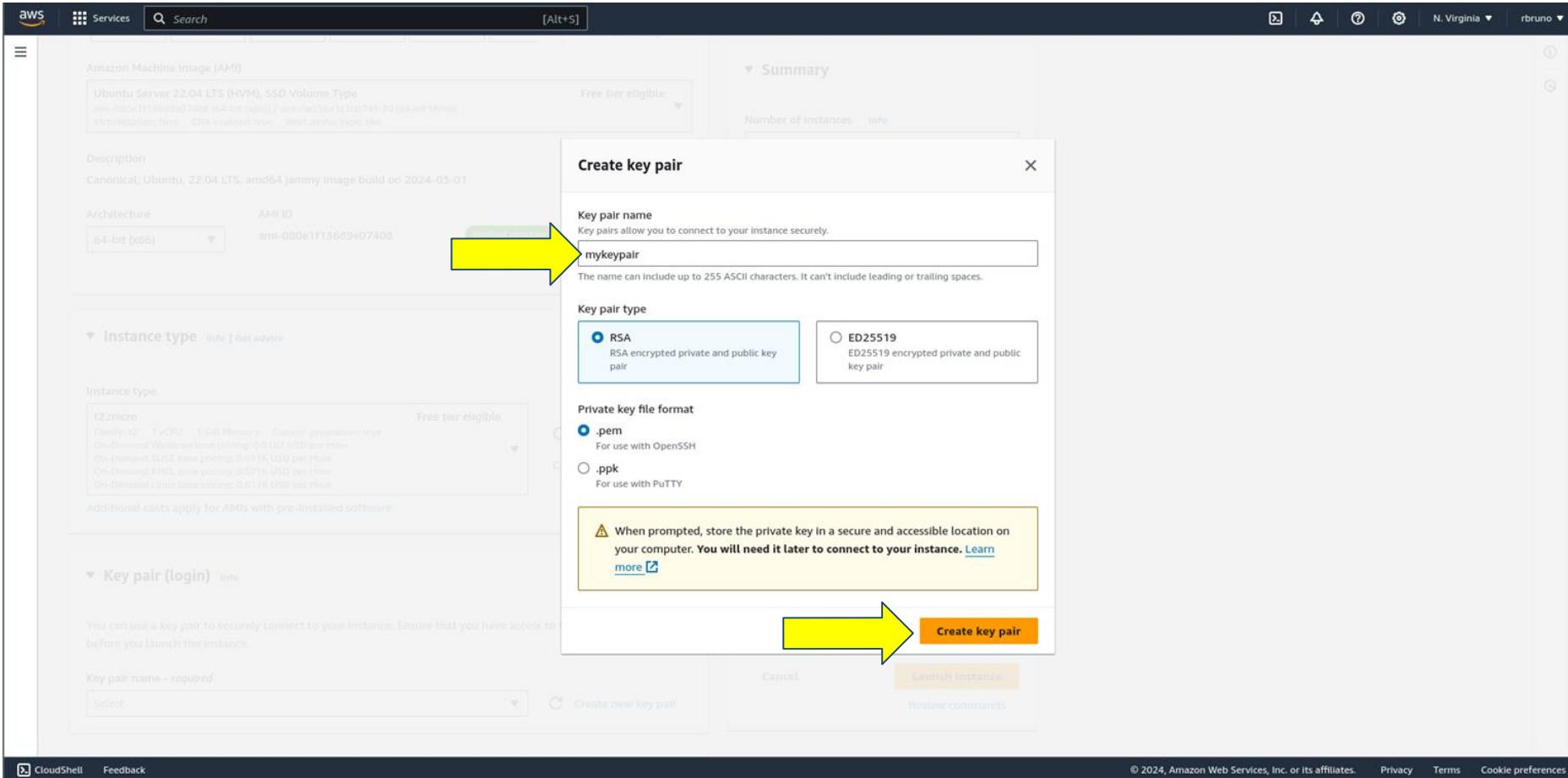
The screenshot shows the AWS EC2 'Launch an instance' wizard. The process is divided into several steps:

- Name and tags:** A yellow arrow points to the 'Name' field where 'test-vm' is entered.
- Application and OS Images (Amazon Machine Image):** A yellow arrow points to the 'Quick Start' section, which includes icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE, and a search bar.
- Amazon Machine Image (AMI):** A yellow arrow points to the 'Amazon Linux 2023 AMI' card, which lists the AMI ID as 'ami-04e5276ebb8451442'. A red box highlights the 'Free tier eligible' status.
- Summary:** On the right, a yellow box highlights the 'Free tier' information: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) Instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 10 GB of bandwidth to the internet.'
- Buttons:** At the bottom are 'Cancel', 'Launch instance' (in orange), and 'Review commands'.

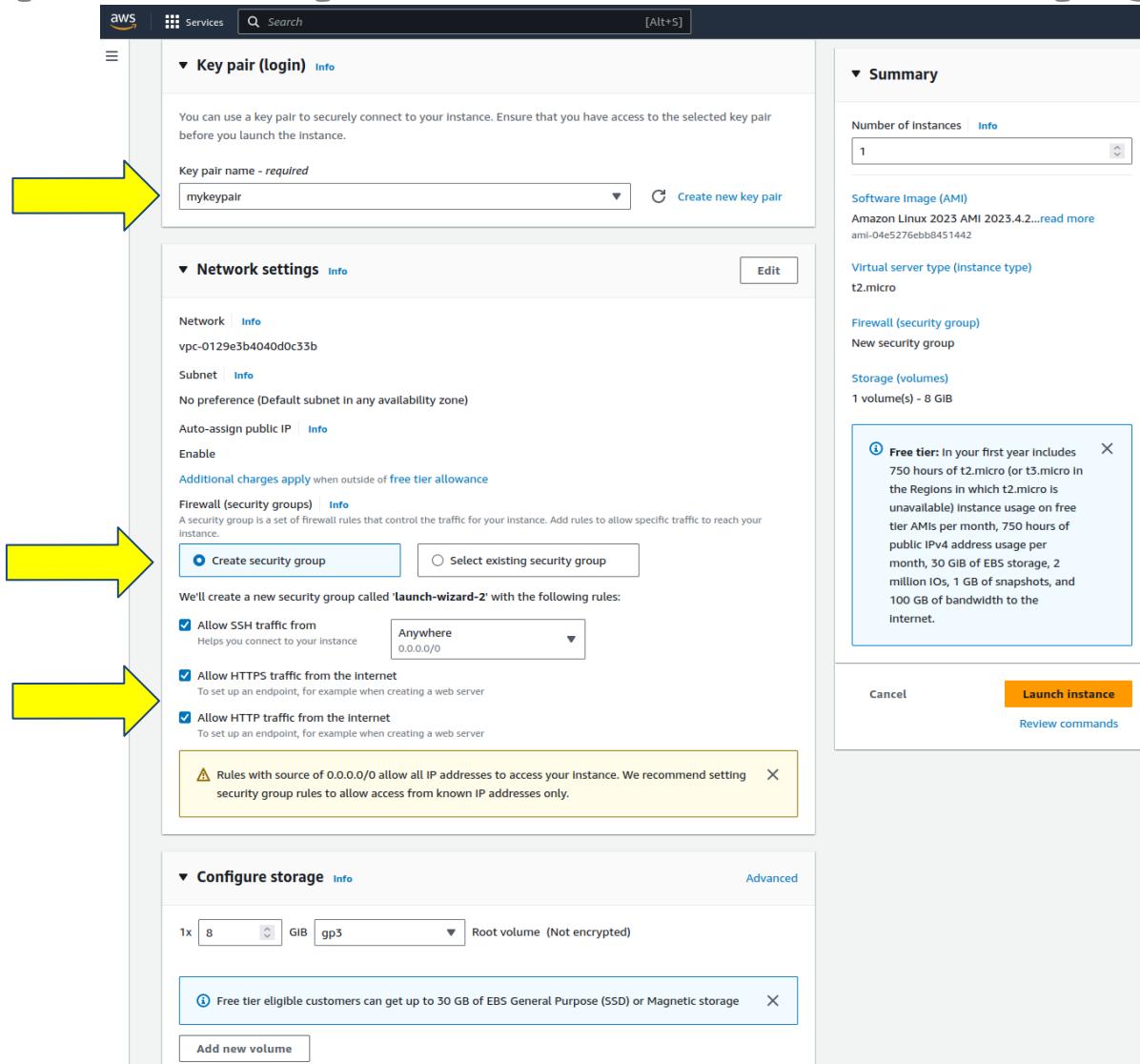
Pick instance type, create key pair



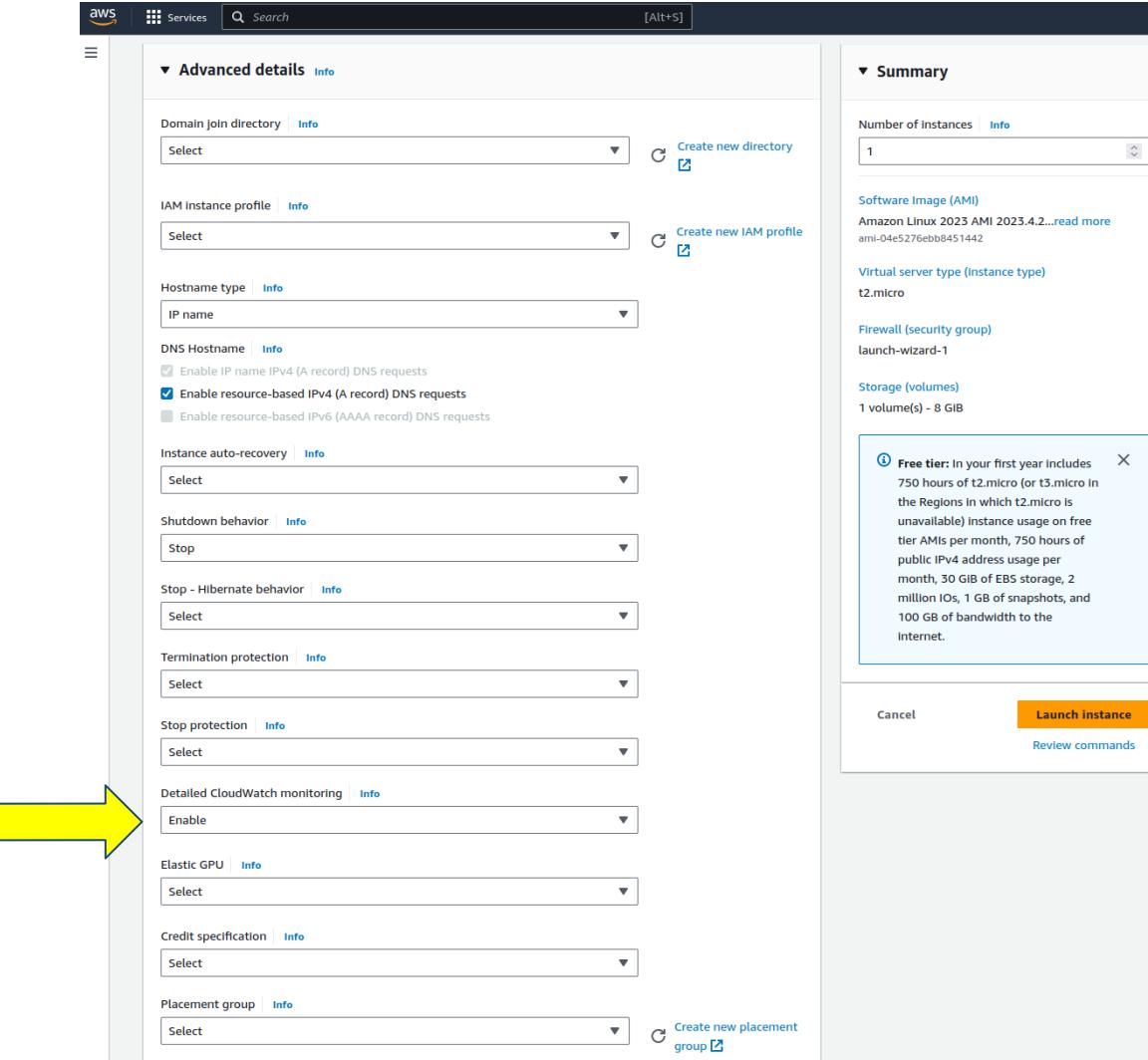
Create keypair



Ensure proper key, create security group



Enable monitoring



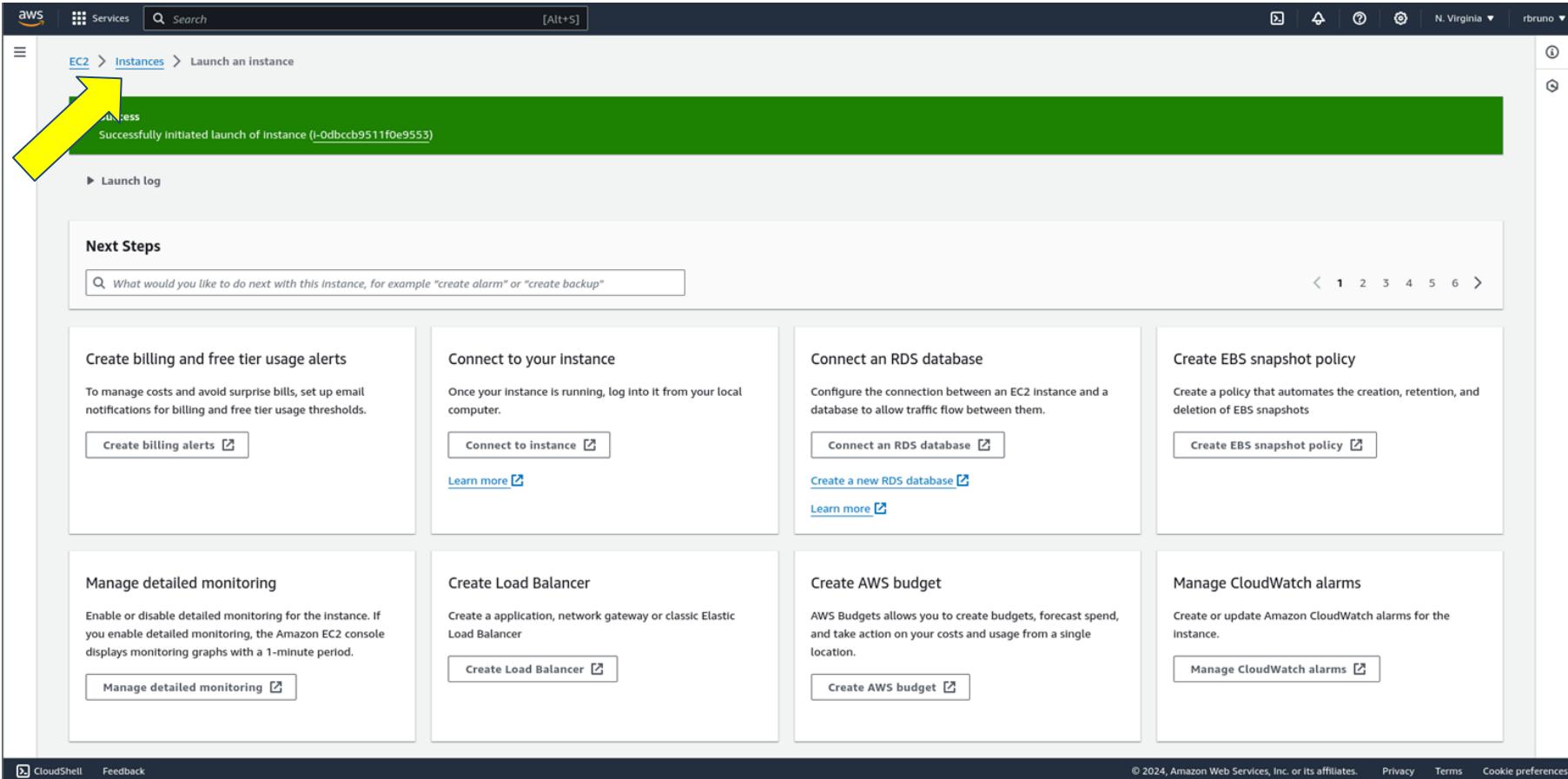
Launch your instance

The screenshot shows the AWS Launch Wizard interface for launching a new Amazon EC2 instance. The process is divided into several steps:

- Security Group Configuration:** A security group named 'launch-wizard-1' is being created with the following rules:
 - Allow SSH traffic from Anywhere (0.0.0.0/0)
 - Allow HTTPS traffic from the internet
 - Allow HTTP traffic from the internetA warning message 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 access from known IP addresses only."
- Configure Storage:** The root volume is set to 8 GiB using gp2 storage. A note indicates: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage". An "Add new volume" button is available.
- Summary:** The summary step provides an overview of the launch configuration:
 - Number of instances: 1
 - Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ... (with a "read more" link)
 - Virtual server type (Instance type): t2.micro
 - Firewall (security group): New security group
 - Storage (volumes): 1 volume(s) - 8 GiBA callout box highlights the "Free tier" information: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) Instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet."
- Launch Instance:** The final step is the "Launch instance" button, preceded by a large yellow arrow pointing right.

At the bottom, there are links for CloudShell, Feedback, and various AWS footer links: © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Instance starting...



A screenshot of the AWS EC2 Instances Launch an instance page. At the top, there's a green success message: "Successfully initiated launch of instance (i-0dbccb9511f0e9553)". A yellow arrow points to the "Launch log" link below it. The page has a "Next Steps" section with several cards:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. [Create billing alerts](#)
- Connect to your instance**: Once your instance is running, log into it from your local computer. [Connect to instance](#)
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. [Connect an RDS database](#)
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. [Create EBS snapshot policy](#)
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. [Manage detailed monitoring](#)
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. [Create Load Balancer](#)
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. [Create AWS budget](#)
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. [Manage CloudWatch alarms](#)

At the bottom, there are links for CloudShell, Feedback, and various legal and preference links.

Edit your Security Group

The screenshot shows the AWS EC2 Security Groups page. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, Network & Security (with 'Security Groups' highlighted by a yellow arrow), and Load Balancing. The main content area displays a table of security groups with the following data:

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0eb535ab19e2f02f6	launch-wizard-1	vpc-0129e3b4040d0c33b	launch-wizard-1 created 2024-04-25T...	752085098994
-	cd2f8f4262a50	default	vpc-0129e3b4040d0c33b	default VPC security group	752085098994

A yellow arrow points to the 'Security Groups' link in the sidebar, and another yellow arrow points to the first security group ID in the table.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Edit your Security Group

The screenshot shows the AWS EC2 Security Groups page for the security group `sg-0eb335ab19e2f02f6 - launch-wizard-1`. The left sidebar contains navigation links for EC2 Dashboard, Services, Search, and various EC2 management options like Instances, Images, and Network & Security.

Details:

- Security group name: `launch-wizard-1`
- Security group ID: `sg-0eb335ab19e2f02f6`
- Description: `launch-wizard-1 created 2024-04-25T21:31:40.823Z`
- VPC ID: `vpc-0129e3b4040d0c33b`
- Owner: `752085098994`
- Inbound rules count: `3 Permission entries`
- Outbound rules count: `1 Permission entry`

Inbound rules (3):

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-041d72b17ba939...	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-0ce84b622780063...	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sgr-0cc98d0cf4f4e3733	IPv4	HTTPS	TCP	443	0.0.0.0/0	-

Actions: Includes buttons for `Actions ▾`, `Manage tags`, and `Edit inbound rules`.

Page Footer: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences.

Edit your Security Group

Add a rule like this to expose instance's port 8000 to public

Inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	Delete
sgr-041d72b17ba9392c6	SSH	TCP	22	Custom	0.0.0.0/0 X	<input type="button" value="Delete"/>
sgr-0ce84b62278006335	HTTP	TCP	80	Custom	0.0.0.0/0 X	<input type="button" value="Delete"/>
sgr-0cc98d0cf4f4e3733	HTTPS	TCP	443	Custom	0.0.0.0/0 X	<input type="button" value="Delete"/>

-

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

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Copy public IP address

Instances (1/2) [Info](#)

Find Instance by attribute or tag (case-sensitive)

All states ▾

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
test-vm	i-09e5786f52ddbc5c6	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-34-228-160-154.co...	34.228.160.154	-	-	disabled	launch-wizard-1	mykeypair
test-vm	i-059036e3f8950a6ee	Terminated	t2.micro	-	View alarms +	us-east-1b	-	-	-	-	disabled	-	mykeypair

Go back to the list of EC2 instances

i-09e5786f52ddbc5c6 (test-vm)

Details Status and alarms New Monitoring Security Networking Storage Tags

Instance summary [Info](#)

Instance ID
i-09e5786f52ddbc5c6 (test-vm)

IPv6 address
-

Hostname type
IP name: ip-172-31-29-20.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
34.228.160.154 [Public IP]

IAM Role
-

Public IPv4 address copied
34.228.160.154 | [open address](#)

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-29-20.ec2.internal

Instance type
t2.micro

VPC ID
vpc-0129e5b4040d0c35b

Subnet ID
subnet-077681e8f2cf58277

Private IPv4 addresses
172.31.29.20

Public IPv4 DNS
ec2-34-228-160-154.compute-1.amazonaws.com | [open address](#)

Elastic IP addresses
-

AWS Compute Optimizer finding
[Opt-in to AWS Compute Optimizer for recommendations.](#) | Learn more

Decrease permissions on key

```
rbruno@elara~> ssh -i ~/Downloads/mykeypair.pem ec2-user@<public IP>  
@ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @  
@     WARNING: UNPROTECTED PRIVATE KEY FILE!     @  
@ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @  
Permissions 0644 for '/home/rbruno/Downloads/esckeypair.pem' are too open.  
It is required that your private key files are NOT accessible by others.  
This private key will be ignored.  
Load key "/home/rbruno/Downloads/mykeypair.pem": bad permissions  
ec2-user@mykeypair: Permission denied (publickey).  
  
~> chmod 400 ~/Downloads/mykeypair.pem
```

SSH, update, install JDK, setup WebServer

```
// Copy webserver code to the instance  
~> scp -i mykeypair.pem WebServer.java ec2-user@<public IP>:  
  
// SSH to the instance  
~> ssh -i mykeypair.pem ec2-user@<public IP>  
(...)  
:  
~$ sudo yum update  
(...)  
:  
~$ sudo yum install java-11-amazon-corretto-devel.x86_64 -y  
(...)  
:  
~$ javac WebServer.java  
~$ sudo nano /etc/rc.local  
(...)  
:  
~$ sudo chmod +x /etc/rc.local  
(...)           # Proceed with the auto-start instructions from README.md  
:  
~$ cat /etc/rc.local # The final result should show have the following lines:  
#!/bin/sh -e  
  
java -cp /home/ec2-user WebServer &> /tmp/webserver.log
```

Script that runs at VM boot time.
Look for README.md of this lab to find detailed instructions
about how to set up auto-start

The WebServer should automatically start after a reboot

Reboot and ensure that WebServer is up

The screenshot shows the AWS EC2 Instances page with two instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
test-vm	i-059036e3f8958a6ee	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-
<input checked="" type="checkbox"/> test-vm	i-09e5786f52ddbc5c6	Running	t2.micro	2/2 checks passed	View alarms	us-east-1c	ec2-34-228-160-154.co...	34.228.160.154	-

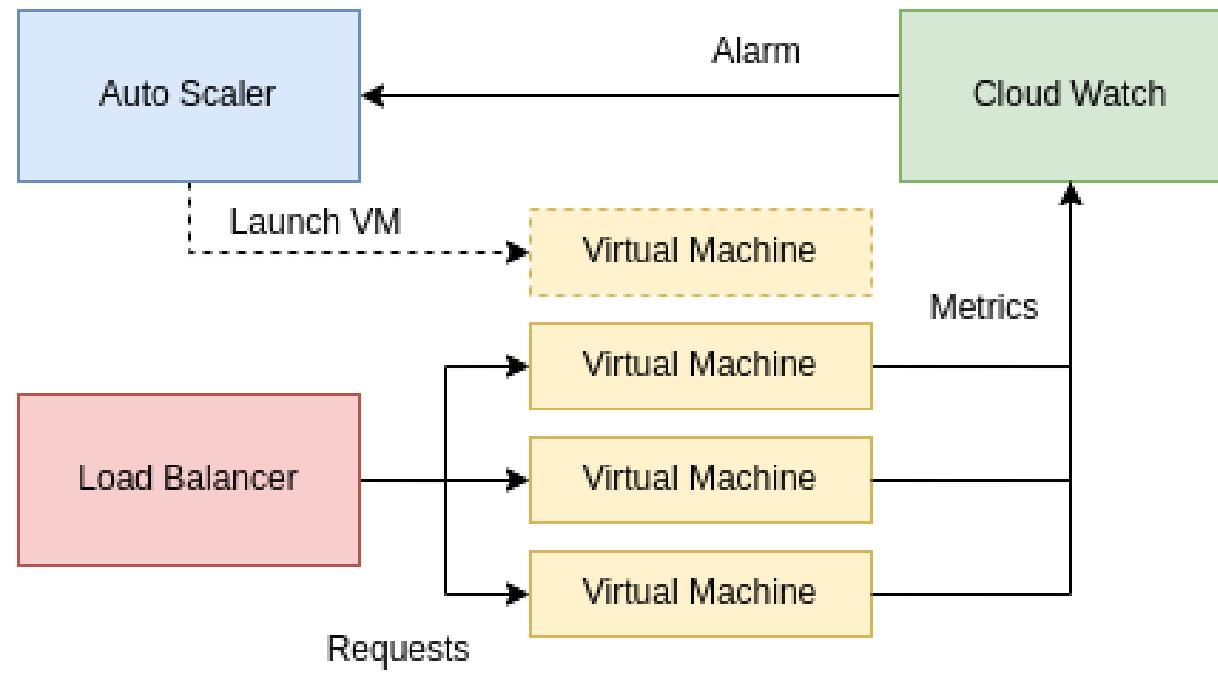
The Actions menu is open for the selected instance (the one with the checked checkbox). The menu includes:

- Stop Instance
- Start Instance
- Reboot Instance** (highlighted with a yellow arrow)
- Hibernate Instance
- Terminate Instance

Outline

- **Load Balancing and VM Scaling**
 - using the **AWS web console** and the Command Line Interface

Load Balancing and VM Scaling



Step 1: Create VM image

Instances (1/2) Info

All states ▾

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitoring
test-vm	i-059036e3f8958a6ee	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-	-	disabled
<input checked="" type="checkbox"/> test-vm	i-09e5786f52ddbc5c6	Running	t2.micro	2/2 checks passed	View alarms	us-east-1c	ec2-34-228-160-154.co...	34.228.160.154	-	-	disabled

Actions ▾ **Launch instances** ▾

- Connect
- Instance state ▾
- Actions ▾ **Launch instances** ▾
- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Create Image**
- Create template from instance
- Launch more like this

VM launched using the steps in previous slides via the Web Console

Step 1: Create VM image

EC2 > Instances > i-09e5786f52ddbc5c6 > Create image

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 Instance. You can create an Image from the configuration of an existing instance.

Instance ID
i-09e5786f52ddbc5c6 (test-vm)

Image name
 Maximum 127 characters. Can't be modified after creation.

Image description - optional
 Maximum 255 characters

No reboot
 Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	8	EBS General Purpose S...	3000	Enable	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel **Create image**



Step 1: Create VM image

A screenshot of the AWS EC2 console showing the creation of an Amazon Machine Image (AMI). The left sidebar shows various navigation options like EC2 Dashboard, Instances, and Images. A yellow arrow points to the 'AMIs' link under the 'Images' section. The main content area shows a table of existing AMIs, with one row selected. A large callout box covers the middle section of the table, containing the text 'Creating the image may take a few minutes'. The bottom section shows detailed information for the selected AMI, including its ID, name, owner, and various configuration details.

Creating the image may take a few minutes

Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date	Platform	Root device type	Block device mapping
CNV-lab-AWS-img	ami-0cdfa7707cae2a336	752085098994/CNV-lab-AWS-img	752085098994	Private	Available	2024/04/25 21:18 GMT+1	Linux/UNIX	ebs	/dev/xvda	

AMI ID: ami-0cdfa7707cae2a336

Details		Permissions		Storage		Tags	
AMI ID	ami-0cdfa7707cae2a336	image type	machine	Platform details	Linux/UNIX	Root device type	EBS
AMI name	CNV-lab-AWS-img	Owner account ID	752085098994	Architecture	x86_64	Usage operation	RunInstances
Root device name	/dev/xvda	Status	Available	Source	752085098994/CNV-lab-AWS-img	Virtualization type	hvm
Boot mode	uefi-preferred	State reason	-	Creation date	Tue Apr 25 2024 21:18:24 GMT+0100 (Western European Summer Time)	Kernel ID	-
Description	Web server for the CNV AWS lab	Product codes	-	RAM disk ID	-	Deprecation time	-
Last launched time	-	Block devices	/dev/xvda=snap-Dedd543aa325cb8cc8:true:gp5				

Step 1: Create VM image

Instances (1/1) [Info](#)

Find Instance by attribute or tag (case-sensitive)

All states ▾

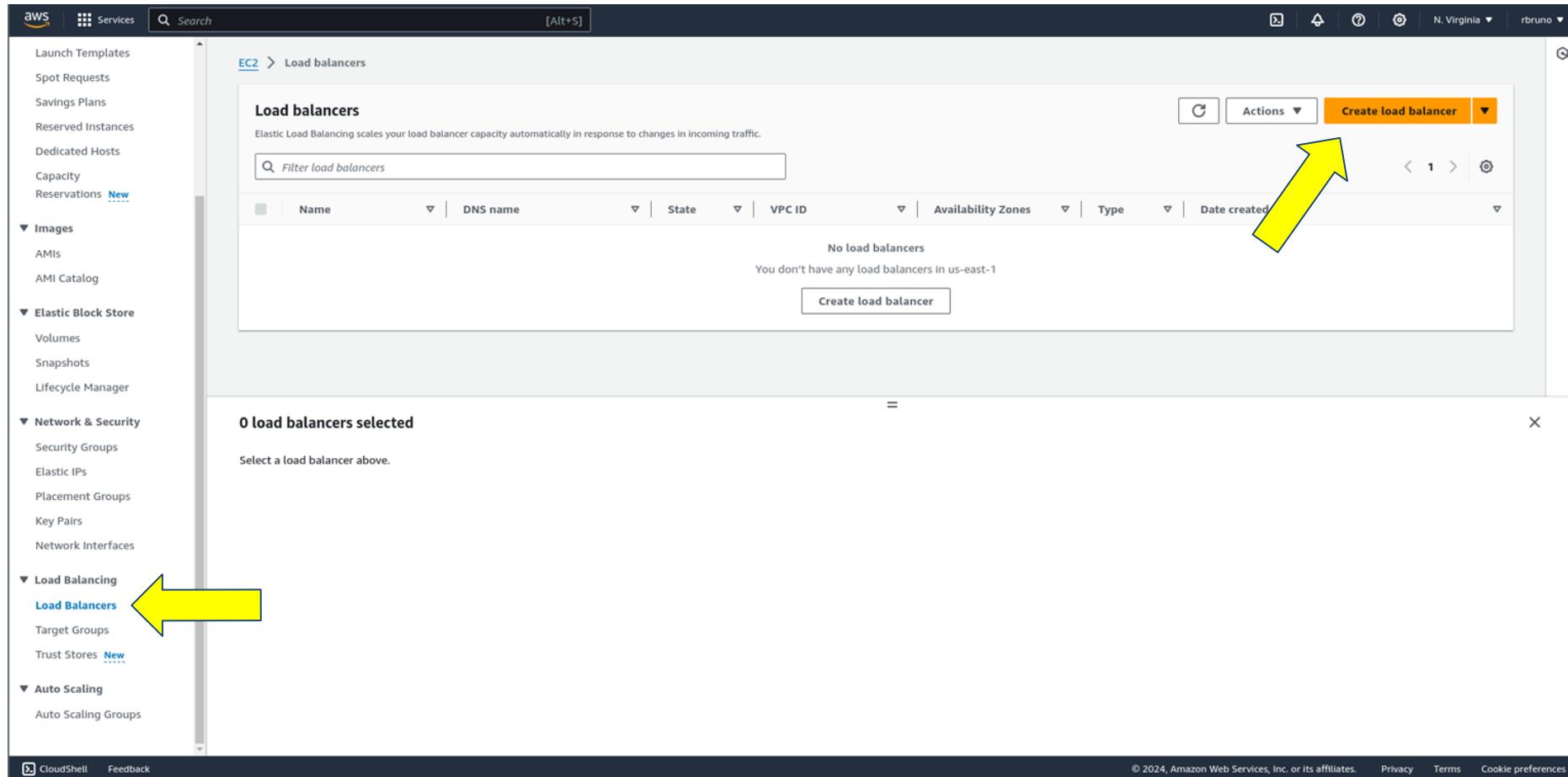
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs
<input checked="" type="checkbox"/> test-vm	i-09e5786f52ddbc5c6	Running View details Logs	t2.micro	2/2 checks passed View alarms +	us-east-1c	ec2-34-228-160-154.co...	34.228.160.154	-	-	

Actions ▾ [Launch instances](#) ▾

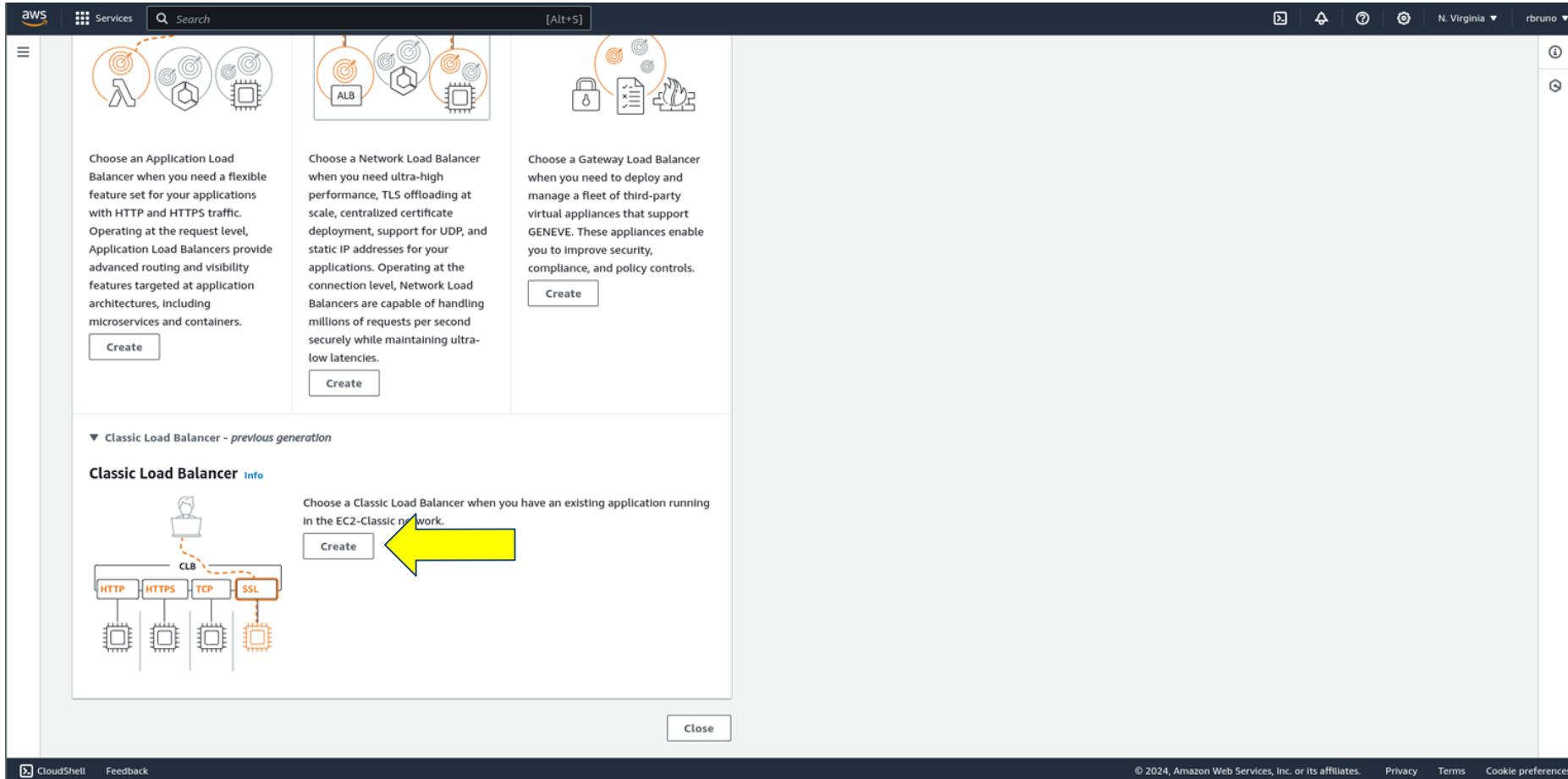
- Stop Instance
- Start Instance
- Reboot Instance
- Hibernate instance
- [Terminate instance](#)

Once finished, go ahead and terminate the instance

Step 2: Create a Load Balancer



Step 2: Create a Load Balancer



Step 2: Create a Load Balancer

EC2 > Load balancers > Create Classic Load Balancer

Create Classic Load Balancer Info

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

▶ How Classic Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.
 Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)
 Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your network settings.

VPC Info
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are available for selection. The selected VPC cannot be changed after the load balancer is created. When selecting a VPC for your load balancer, ensure each subnet has a CIDR block with at least a /27 bitmask and at least 8 free IP addresses. [Learn more](#)

vpc-0129e3b4040d0c33b
IPv4 VPC CIDR: 172.31.0.0/16

Mappings
Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1a (use1-az1)
Subnet

IPv4 address
Assigned by AWS

us-east-1b (use1-az2)
 us-east-1c (use1-az4)
 us-east-1d (use1-az6)
 us-east-1e (use1-az3)

Any of these should be okay

Step 2: Create a Load Balancer

We can reuse the previous security group or create a new one with TCP protocol and port 80

The screenshot shows the AWS Load Balancer configuration interface. It includes:

- Security groups:** A dropdown menu allows selecting up to 5 security groups. One item, "launch-wizard-1" (sg-01550a9daaed37af1 VPC: vpc-0129e3b4040d0c33b), is selected.
- Listeners and routing:** A section for defining listeners. It shows a single listener named "Listener HTTP:80" with the following settings:
 - Listener protocol: HTTP
 - Listener port: 80
 - Instance protocol: HTTP
 - Instance port: 8000
- Health checks:** A section for defining health check parameters. It shows a ping target with the following settings:
 - Ping protocol: HTTP
 - Ping port: 8000
 - Ping path: /test
- Advanced health check settings:** A section for configuring response timeout, interval, unhealthy threshold, and healthy threshold.

Step 2: Create a Load Balancer

Attributes
Creating your load balancer using the console gives you the opportunity to specify additional features at launch. You can also find and adjust these settings in the load balancer's "Attributes" section after your load balancer is created.

Enable cross-zone load balancing
With cross-zone load balancing, each load balancer node for your Classic Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones. If cross-zone load balancing is disabled, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only. Classic Load Balancers created with the API or CLI have cross-zone load balancing disabled by default. After you create a Classic Load Balancer, you can enable or disable cross-zone load balancing at any time.

Enable connection draining
Applicable to instances that are deregistering, this feature allows existing connections to complete (during a specified draining interval) before reporting the instance as deregistered.
[Learn more](#)

Timeout (draining interval)
The maximum time for the load balancer to allow existing connections to complete. When the maximum time limit is reached, the load balancer forcibly closes any remaining connections and reports the instance as deregistered.
300 seconds
Valid values: 1-3600 (integers only)

► Load balancer tags - optional
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

Review
Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose [Create load balancer](#).

Summary
Review and confirm your configurations. [Estimate cost](#)

Basic configuration Edit	Network mapping Edit	Security groups Edit	Listeners and routing Edit
CNV-lab-AWS-LB • Internet-facing	VPC vpc-0129e3b4040d0c33b • us-east-1a subnet-0cb6b4781c26ac0e5	• launch-wizard-1 sg-01550a9daed57af1	• HTTP:80
Health checks Edit	Instances Edit	Attributes Edit	Tags Edit
HTTP:8000/test • Timeout: 5 seconds • Interval: 30 seconds • Unhealthy threshold: 2 • Unhealthy threshold: 10	No instances added yet	• Cross-zone load balancing: On • Connection draining: On • Connection draining timeout: 300 seconds	None

[Cancel](#) **Create load balancer**



Step 2: Create a Load Balancer

Successfully created load balancer: CNV-lab-AWS-LB

It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers > CNV-lab-AWS-LB

Load balancers (1)

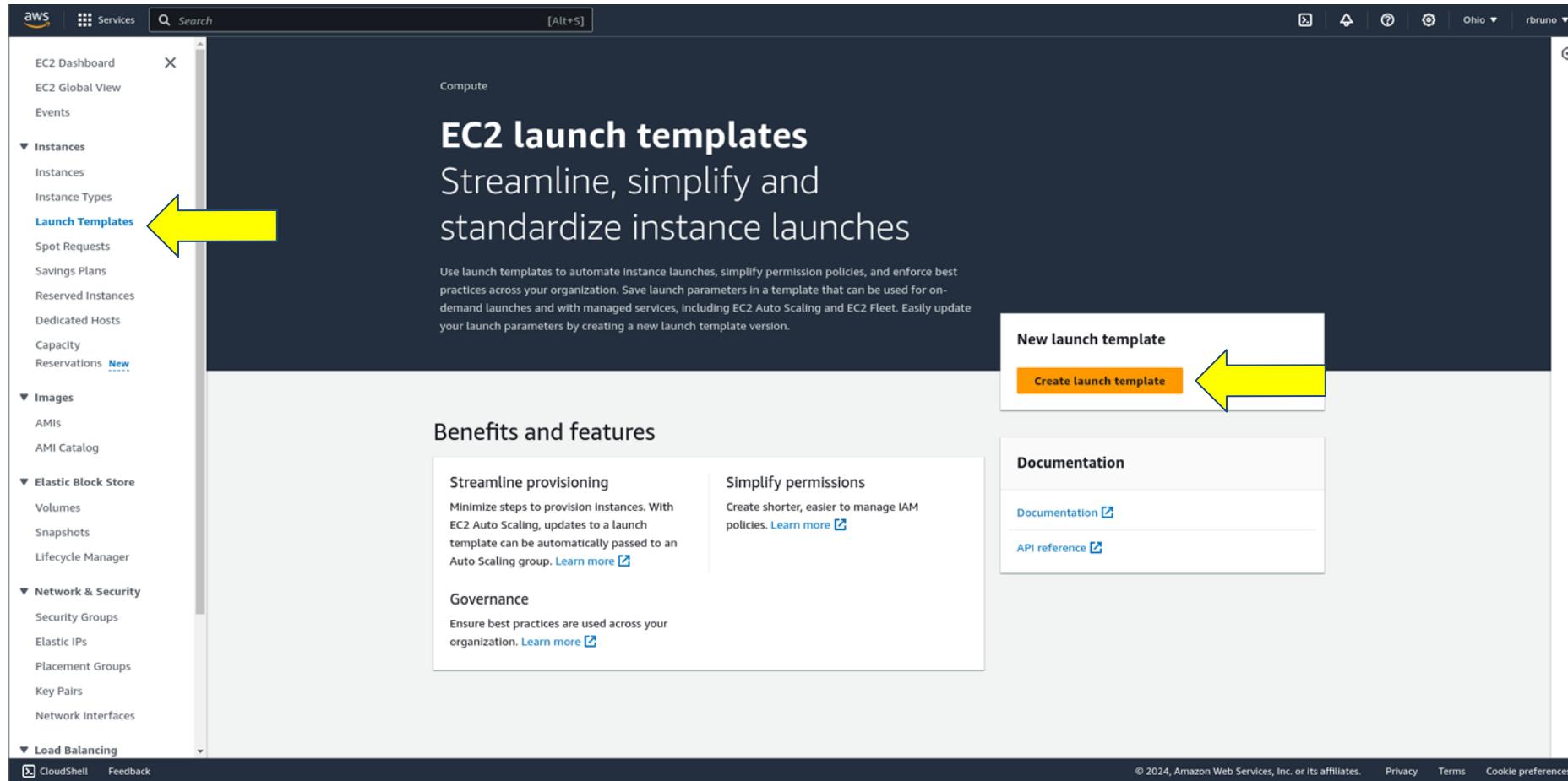
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers 1 match

CNV-lab-AWS-LB

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<input type="checkbox"/>	CNV-lab-AWS-LB	CNV-lab-AWS-LB-535842...	-	vpc-0129e3b4040d0c...	us-east-1a (use1-az1)	classic	April 23, 2024, 21:45 (UTC+01:00)

Step 3: Create a Launch Template



Step 3: Create a Launch Template

The screenshot shows the 'Create launch template' wizard. The first section, 'Launch template name and description', contains fields for the launch template name ('cnv-launchconfig') and a template version description ('A prod webserver for MyApp'). A yellow arrow points to the 'Launch template name' field. The second section, 'Launch template contents', shows the 'Application and OS Images (Amazon Machine Image)' section. It includes a search bar ('Search our full catalog including 1000s of application and OS images'), a filter for 'My AMIs' (selected), and a list of AMIs. An AMI named 'CNV-lab-AWS-img' is selected, and a yellow arrow points to its details. The right side of the screen shows a 'Summary' panel with instance type (t2.micro), storage (1 volume(s) - 8 GiB), and a note about free tier usage.

Launch template name and description

Launch template name - required
cnv-launchconfig
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description
A prod webserver for MyApp
Max 255 chars

Auto Scaling guidance [Info](#)
Select this if you intend to use this template with EC2 Auto Scaling
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► [Template tags](#)
► [Source template](#)

Launch template contents

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your Instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recent | **My AMIs** | Quick Start

Don't include in launch template Owned by me Shared with me

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

CNV-lab-AWS-img
ami-0cdfa7707cae2a336
2024-04-23T20:18:24.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Web server for the CNV AWS lab

Architecture x86_64 AMI ID ami-0cdfa7707cae2a336

Summary

Software Image (AMI)
Web server for the CNV AWS lab...read more
ami-0cdfa7707cae2a336

Virtual server type (Instance type)
t2.micro

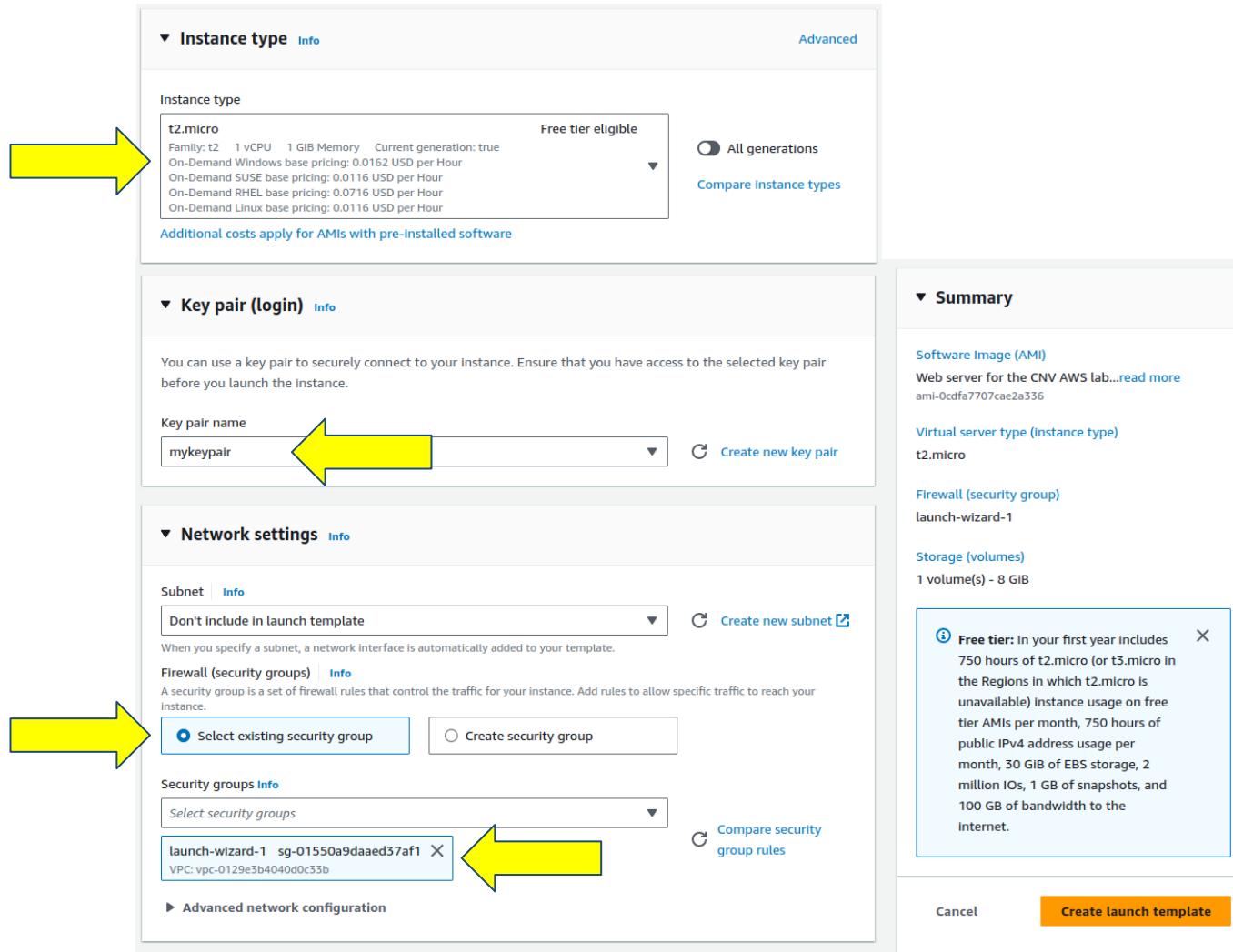
Firewall (security group)
-

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) Instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) **Create launch template**

Step 3: Create a Launch Template



Step 3: Create a Launch Template

Advanced details [Info](#)

IAM Instance profile [Info](#)
Don't include in launch template [Create new IAM profile](#)

Hostname type [Info](#)
Don't include in launch template

DNS Hostname [Info](#)
 Enable resource-based IPv4 (A record) DNS requests
 Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery [Info](#)
Don't include in launch template

Shutdown behavior [Info](#)
Don't include in launch template

Stop - Hibernate behavior [Info](#)
Don't include in launch template

Termination protection [Info](#)
Don't include in launch template

Stop protection [Info](#)
Don't include in launch template

Detailed CloudWatch monitoring [Info](#)
Enable

Elastic GPU [Info](#)
Don't include in launch template

Credit specification [Info](#)
Don't include in launch template

Placement group [Info](#)
Don't include in launch template [Create new placement group](#)

Summary

Software Image (AMI)
Web server for the CNV AWS lab...[read more](#)
ami-0cdfa7707cae2a336

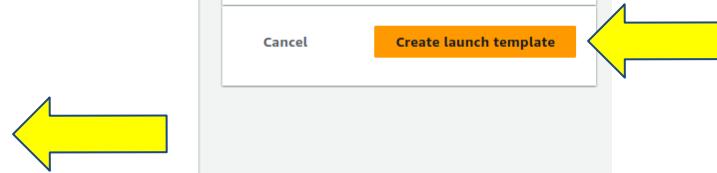
Virtual server type (Instance type)
t2.micro

Firewall (security group)
launch-wizard-1

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the Internet.

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



Step 3: Create a Launch Template

[EC2](#) > [Launch templates](#) > Create launch template

 Success
Successfully created [cnv-launchconfig\(lt-05cc37d3c12824c12\)](#).

[Actions log](#)

Next Steps

Launch an instance
With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.
[Launch Instance from this template](#)

Create an Auto Scaling group from your template
Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 Instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.
[Create Auto Scaling group](#)

Create Spot Fleet
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot Instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.
[Create Spot Fleet](#)

[View launch templates](#)

Step 4: Create an Auto Scaling Group

The screenshot shows the AWS EC2 Auto Scaling Groups page. The left sidebar lists various AWS services like Launch Templates, Spot Requests, and Auto Scaling Groups (which is currently selected). A large central panel titled "Amazon EC2 Auto Scaling" explains its purpose: "helps maintain the availability of your applications". Below this is a diagram titled "How it works" showing an "Auto Scaling group" containing four squares, with labels for "Minimum size", "Desired capacity", and "Maximum size". A callout box on the right says "Create Auto Scaling group" with a yellow arrow pointing to it. The bottom of the page includes links for "Pricing", "Getting started", and "FAQ", along with standard AWS footer links.

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations New

Images

AMIs

AMI Catalog

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

Trust Stores New

Auto Scaling

Auto Scaling Groups ←

CloudShell Feedback

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变化.

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

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 4: Create an Auto Scaling Group

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Choose instance launch options

Step 3 - optional
Configure advanced options

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name
Enter a name to identify the group.
cnv-scaling

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

Launch template [Info](#) [Switch to launch configuration](#)

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

cnv-launchconfig

[Create a launch template](#)

Version
Default (1) [Create a launch template version](#)

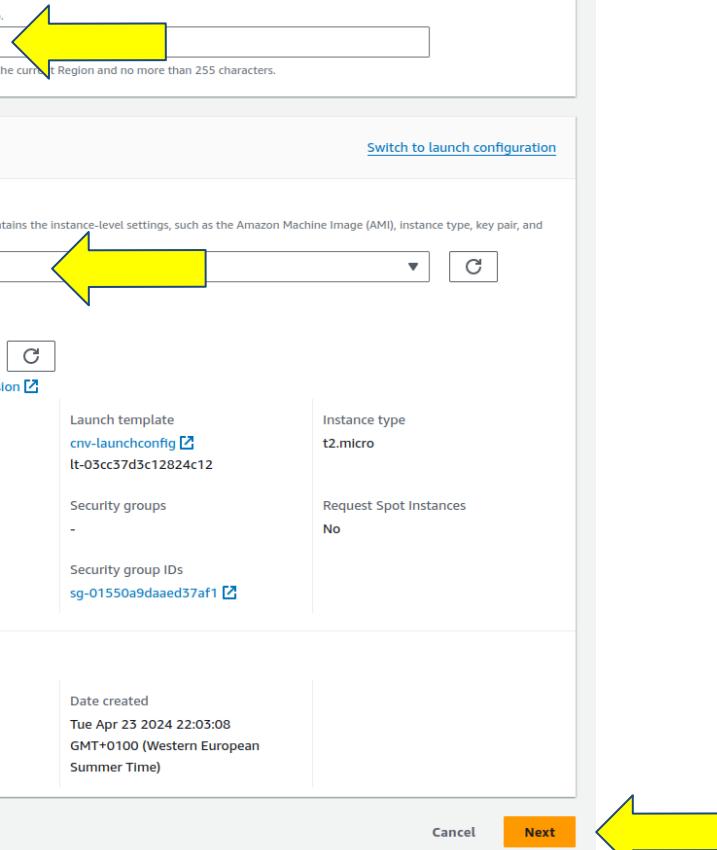
Description	Launch template	Instance type
-	cnv-launchconfig lt-05cc37d3c12824c12	t2.micro
AMI ID	ami-0cdfa7707cae2a336	Security groups
Key pair name	mykeypair	Security group IDs sg-01550a9daae37af1

Additional details

Storage (volumes)
-

Date created
Tue Apr 23 2024 22:03:08
GMT+0100 (Western European Summer Time)

[Cancel](#) [Next](#)



Step 4: Create an Auto Scaling Group

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
[Choose launch template or configuration](#)

Step 2
Choose instance launch options

Step 3 - optional
[Configure advanced options](#)

Step 4 - optional
[Configure group size and scaling](#)

Step 5 - optional
[Add notifications](#)

Step 6 - optional
[Add tags](#)

Step 7
[Review](#)

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
cnv-launchconfig	It-03cc37d3c12824c12	-

Instance type
t2.micro

Network Info

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

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-0129e3b4040d0c33b
172.31.0.0/16 Default

Create a VPC

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group uses.
Select Availability Zones and subnets
us-east-1a | subnet-0cb6b4781c26ac0e5
172.31.0.0/20 Default

Create a subnet

Should match the load balancer availability zone and subnet

Cancel Skip to review Previous Next

Step 4: Create an Auto Scaling Group

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
[Choose launch template or configuration](#)

Step 2
[Choose instance launch options](#)

Step 3 - optional
[Configure advanced options](#)

Step 4 - optional
[Configure group size and scaling](#)

Step 5 - optional
[Add notifications](#)

Step 6 - optional
[Add tags](#)

Step 7
[Review](#)

Configure advanced options - optional Info

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

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.

Attach to a new load balancer
Create a basic load balancer to attach to your Auto Scaling group.

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

VPC Lattice integration options Info

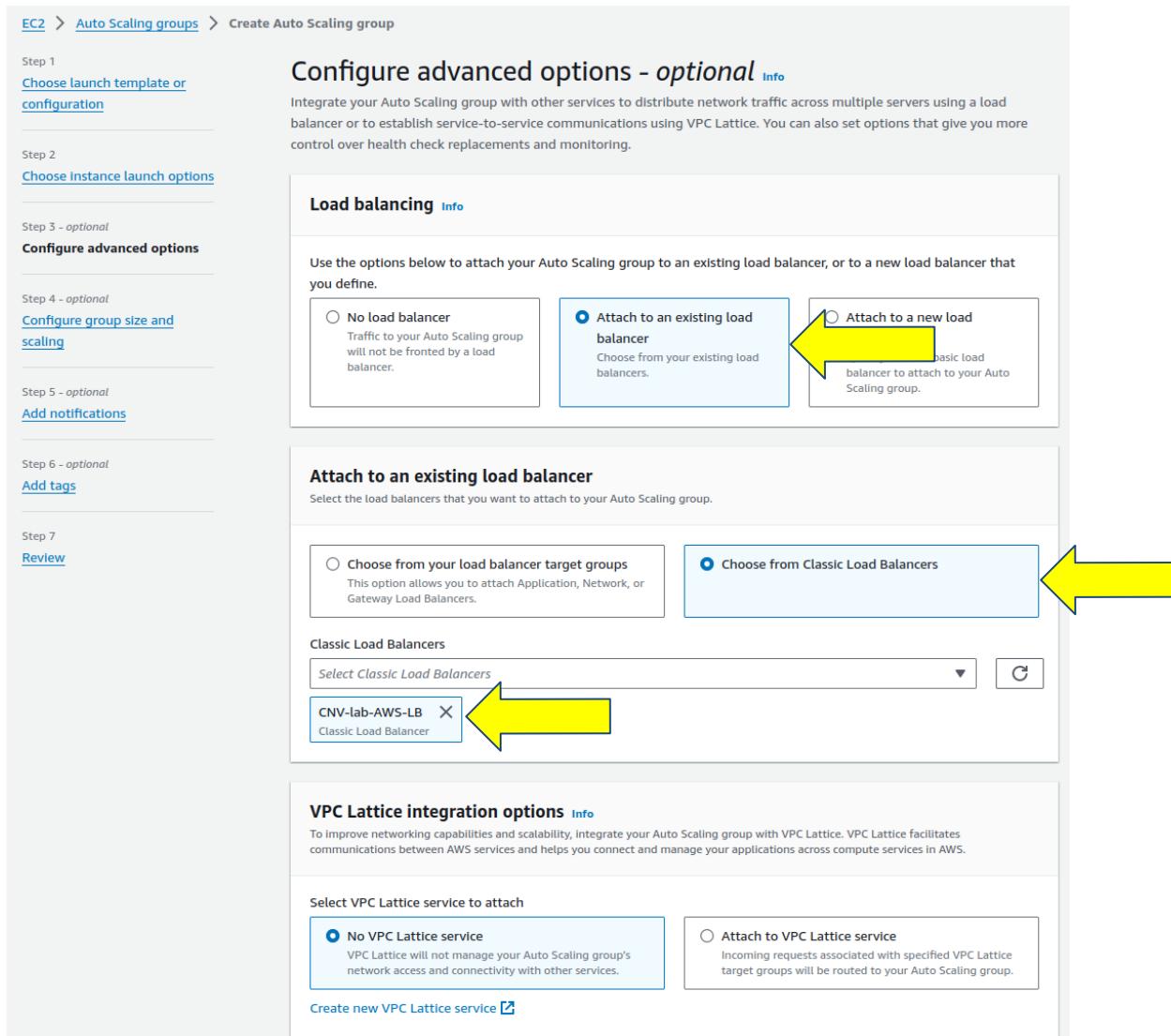
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service [+]



Step 4: Create an Auto Scaling Group

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

Always enabled

Additional health check types - optional | [Info](#)

Turn on Elastic Load Balancing health checks Recommended

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

(i) EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the [Load Balancer console](#).

Turn on VPC Lattice health checks

VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Health check grace period | [Info](#)

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

60 seconds

Additional settings

Monitoring | [Info](#)

Enable group metrics collection within CloudWatch

Default instance warmup | [Info](#)

The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

Enable default instance warmup

[Cancel](#) [Skip to review](#) [Previous](#) **Next**

Step 4: Create an Auto Scaling Group

The screenshot shows the AWS EC2 Auto Scaling Groups 'Create Auto Scaling group' wizard, specifically Step 4: Configure group size and scaling - optional. The 'Desired capacity' field is set to 1, and the 'Max desired capacity' field is set to 3. A yellow callout box highlights these settings with the text: "We are creating a scaling group that will grow up to 3 VMs".

Configure group size and scaling - optional

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Desired capacity

Specify your group size.

Scaling

You can resize your Auto Scaling group manually or automatically.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity: 1

Max desired capacity: 3

Automatic scaling - optional

Choose whether to use a target tracking policy

No scaling policies

Your Auto Scaling group will remain at its initial size and

Target tracking scaling policy

Choose a CloudWatch metric and target value and let the

Step 4: Create an Auto Scaling Group

Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity: 1 Equal or less than desired capacity

Max desired capacity: 3 Equal or greater than desired capacity

Automatic scaling - optional

Choose whether to use a target tracking policy | Info

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

No scaling policies for now...

Instance maintenance policy Info

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Choose a replacement behavior depending on your availability requirements

Mixed behavior

No policy For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

Prioritize availability

Launch before terminating Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

Control costs

Terminate and launch Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

Flexible

Custom behavior Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Instance scale-in protection

Scale-in protection prevents newly launched instances from being terminated by scaling activities. Make sure to remove scale-in protection for the group or individual instances when instances are ready to be terminated.

Enable instance scale-in protection

Cancel Skip to review Previous Next

Step 4: Create an Auto Scaling Group

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
[Choose launch template or configuration](#)

Step 2
[Choose instance launch options](#)

Step 3 - optional
[Configure advanced options](#)

Step 4 - optional
[Configure group size and scaling](#)

Step 5 - optional
Add notifications

Step 6 - optional
[Add tags](#)

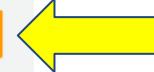
Step 7
[Review](#)

Add notifications - optional Info

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

Add notification

Cancel Skip to review Previous Next



No notifications

Step 4: Create an Auto Scaling Group

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
[Choose launch template or configuration](#)

Step 2
[Choose instance launch options](#)

Step 3 - optional
[Configure advanced options](#)

Step 4 - optional
[Configure group size and scaling](#)

Step 5 - optional
[Add notifications](#)

Step 6 - optional
[Add tags](#)

Step 7
Review

Add tags - optional 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

No tags

Cancel Previous Next

Step 4: Create an Auto Scaling Group

The screenshot shows the AWS Auto Scaling 'Create an Auto Scaling group' wizard, Step 4. The interface includes:

- Target tracking policy:** Step 1 of 3.
- Instance maintenance policy:** Step 2 of 3. It includes fields for Replacement behavior (No policy), Min healthy percentage (-), and Max healthy percentage (-).
- Instance scale-in protection:** Step 3 of 3. It includes a checkbox for Enable Instance protection from scale in (unchecked).
- Step 5: Add notifications:** Shows a 'Notifications' section with No notifications and an 'Edit' button.
- Step 6: Add tags:** Shows a 'Tags (0)' section with a table for Key and Value, and a 'Tag new instances' checkbox. It also shows a 'No tags' message.

A large blue callout box labeled "Review and create" covers the bottom right area. At the very bottom, there are buttons for Cancel, Previous, and a prominent orange "Create Auto Scaling group" button. A yellow arrow points from the "Create Auto Scaling group" button to the "Review and create" callout box.

Step 4: Create an Auto Scaling Group

The screenshot shows the AWS Auto Scaling Groups page. At the top, there is a search bar and navigation links for Services and EC2. The main heading is "Auto Scaling groups (1) Info". Below this is a search bar with placeholder text "Search your Auto Scaling groups". The main 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 | Version Default" as the launch configuration, 0 instances, and a status of "Updating capacity". A large callout box with a blue border and a yellow gradient background is overlaid on the page, containing the text "Preparing Auto Scaling Group...". At the bottom left, it says "0 Auto Scaling groups selected". The bottom navigation bar includes links for CloudShell, Feedback, and various AWS services like Lambda, S3, and CloudWatch.

aws Services Search [Alt+S]

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
cnv-scaling	cnv-launchconfig Version Default	0	Updating capacity	1	1	3	us-east-1a

Preparing Auto Scaling Group...

0 Auto Scaling groups selected

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 5: Create CloudWatch Alarms

The screenshot shows the AWS navigation interface. On the left, a sidebar lists various services under 'Recently visited' and 'All services'. A yellow arrow points from the 'CloudWatch' entry in the sidebar to its corresponding link in the 'Recently visited' dropdown menu. The main content area shows the CloudWatch Metrics dashboard with tabs for 'Launch configurations', 'Launch templates', 'Actions', and a prominent orange 'Create Auto Scaling group' button.

AWS Services Search [Alt+S] N. Virginia Serhiy Ivanenko

Recently visited

EC2 Virtual Servers in the Cloud

Console Home View resource insights, service shortcuts, and feature updates

Billing and Cost Management View and pay bills, analyze and govern your spending, and optimize your costs

Lambda Run code without thinking about servers

AWS Billing Conductor Simplifying your billing practice

CloudWatch Monitor Resources and Applications

Amazon EventBridge Serverless service for building event-driven applications.

IAM Manage access to AWS resources

DynamoDB Managed NoSQL Database

S3 Scalable Storage in the Cloud

Create Auto Scaling group

Desired capacity 1 Min 1 Max 3 Availability Zones us-east-1a

Analytics Application Integration Blockchain Business Applications Cloud Financial Management Compute Containers Customer Enablement Database Developer Tools End User Computing Front-end Web & Mobile Game Development Internet of Things Machine Learning Management & Governance Media Services Migration & Transfer Networking & Content Delivery Quantum Technologies Robotics

<https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1> © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 5: Create CloudWatch Alarms

The screenshot shows the AWS CloudWatch Alarms interface. On the left, a sidebar menu is open under the 'CloudWatch' heading. The 'Alarms' section is expanded, showing 'In alarm' (0), 'All alarms' (0), and 'Billing' options. A large yellow arrow points to the 'All alarms' link. On the right, the main 'Alarms (0)' page is displayed. At the top right of this page, there is a prominent orange 'Create alarm' button, which is also highlighted with a yellow arrow. The rest of the page shows search and filter fields, a table header for 'Name', 'State', 'Last state update', and 'Conditions', and a message stating 'No alarms'.

Step 5: Create CloudWatch Alarms

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

Specify metric and conditions

Metric

Graph
Preview of the metric or metric expression and the alarm threshold.

Select metric



Cancel Next

Step 5: Create CloudWatch Alarms

Note the path.
It may take a couple of minutes for the “By Auto Scaling Group” option to appear

Metrics (8)

AutoScalingGroupName	Metric name	Alarms
cnv-scaling	MetadataNotoken	No alarms
cnv-scaling	NetworkIn	No alarms
cnv-scaling	DiskWriteBytes	No alarms
cnv-scaling	DiskReadOps	No alarms
<input checked="" type="checkbox"/> cnv-scaling	CPUUtilization	No alarms
cnv-scaling	DiskWriteOps	No alarms
cnv-scaling	NetworkOut	No alarms
cnv-scaling	DiskReadBytes	No alarms

Select metric

Step 5: Create CloudWatch Alarms

CloudWatch > Alarms > Create alarm

Step 1
Specify metric and conditions

Step 2
Configure actions

Step 3
Add name and description

Specify metric and conditions

Metric

Graph

This alarm will trigger when the blue line goes above the red line for 1 datapoint within 1 minute.

Percent

50

25.3

0.69

20:00 21:00 22:00

CPUUtilization

Namespace: AWS/EC2

Metric name: CPUUtilization

AutoScalingGroupName: cnv-scaling

Statistic: Average

Period: 1 minute

Conditions

Threshold type:

Static: Use a value as a threshold

Anomaly detection: Use a band as a threshold

Whenever CPUUtilization is...:

Greater: > threshold

Greater/Equal: >= threshold

Lower/Equal: <= threshold

Lower: < threshold

than...:

Define the threshold value: 50

Must be a number

► Additional configuration

Next

Alarm goes off if avg CPU utilization over 1 min is >50%

Alarm goes off if avg CPU utilization over 1 min is >50%

► Additional configuration

Next

Step 5: Create CloudWatch Alarms

No actions/notifications needed
(remove if any)

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

Configure actions

Notification

Add notification

Lambda action

Add Lambda action

Auto Scaling action

Add Auto Scaling action

EC2 action

This action is only available for EC2 Per-Instance Metrics.

Add EC2 action

Systems Manager action Info

This action will create an Incident or OpsItem in Systems Manager when the alarm is **In alarm** state.

Add Systems Manager action

Cancel Previous Next

Step 5: Create CloudWatch Alarms

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

Add name and description

Name and description

Alarm name

A yellow arrow points from the right towards this input field.

Alarm description - optional [View formatting guidelines](#)

[Edit](#) [Preview](#)

> 50% on avg over the last 1 min

Up to 1024 characters (32/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel Previous Next

Give it a name

Step 5: Create CloudWatch Alarms

Preview and create

25.3

0.69

20:00 21:00 22:00

CPUUtilization

AutoScalingGroupName
cnv-scaling

Statistic
Average

Period
1 minute

Conditions

Threshold type
Static

Whenever CPUUtilization 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
> 50% on avg over the last 1 min

ⓘ Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel

Previous

Create alarm

Step 5: Create CloudWatch Alarms

The screenshot shows the AWS CloudWatch Alarms page. On the left, there's a navigation sidebar with options like Dashboards, Alarms (0), Logs, Metrics, X-Ray traces, Events, Application Signals, Network monitoring, and Insights. Below that are Settings, Getting Started, and What's new. The main content area shows a success message: "Successfully created alarm [LowCPUUtilizationAlarm](#)". The Alarms section lists two alarms:

Name	State	Last state update	Conditions	Actions
LowCPUUtilizationAlarm	Insufficient data	2024-04-23 22:11:36	CPUUtilization < 25 for 1 datapoints within 1 minute	No actions
HighCPUUtilizationAlarm	Insufficient data	2024-04-23 22:10:27	CPUUtilization > 50 for 1 datapoints within 1 minute	No actions

A callout box with a blue arrow points from the text "Repeat the same steps for low CPU utilization alarm (e.g., <25%)" to the "LowCPUUtilizationAlarm" row.

Repeat the same steps for low CPU utilization alarm (e.g., <25%)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 6: Setup scaling policies

The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, etc. The main area shows one Auto Scaling group named "cnv-scaling". A yellow arrow points from the "Automatic scaling" tab in the cnv-scaling group details to the "Create dynamic scaling policy" button at the bottom of the page. A callout box says "Go back to your Auto Scaling Group".

Auto Scaling groups (1/1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
cnv-scaling	cnv-launchconfig Version Default	2	-	1	1	3	us-east-1a

Auto Scaling group: cnv-scaling

Automatic scaling

Scaling policies let you customize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 Instances require more time to initialize.

Dynamic scaling policies (0) Info

No dynamic scaling policies have been created

Dynamic scaling policies use real-time data to scale your group based on configurable metrics.

Create dynamic scaling policy

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 6: Setup scaling policies

EC2 > Auto Scaling groups > cnv-scaling

Create dynamic scaling policy

Policy type: Step scaling

Scaling policy name: IncreaseGroupSize

CloudWatch alarm: HighCPUUtilizationAlarm

The previously created alarm

Take the action:

- Add step: 1 capacity units when CPUUtilization <= 50 < infinity
- Add step

Instance warmup: 300 seconds

Cancel Create

HighCPUUtilizationAlarm

AutoScalingGroupName = cnv-scaling

Take the action

Add step

300 seconds

Create

Step 6: Setup scaling policies

The screenshot shows the AWS Auto Scaling Groups page with a success message: "Dynamic scaling policy created or edited successfully." The main table lists one Auto Scaling group named "cnv-scaling". A callout box points to the "Dynamic scaling policies" section with the instruction: "Repeat the previous step to decrease your group size". Another callout box points to the "DecreaseGroupSize" policy with the note: "Note ‘Remove’ here".

Auto Scaling groups (1/1) [Info](#)

Search your Auto Scaling groups

Actions ▾ [Create Auto Scaling group](#)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
cnv-scaling	cnv-launchconfig Version Default	2	-	1	1	3	us-east-1a

Auto Scaling group: cnv-scaling

Dynamic scaling policies (4) [Info](#)

Actions ▾ [Create dynamic scaling policy](#)

DecreaseGroupSize Step scaling Enabled

LowCPUUtilizationAlarm breaches the alarm threshold: CPUUtilization < 25 for 1 consecutive periods of 60 seconds for the metric dimensions:
AutoScalingGroupName = cnv-scaling

Remove 1 capacity units when 25 >= CPUUtilization > -infinity

IncreaseGroupSize Step scaling Enabled

HighCPUUtilizationAlarm breaches the alarm threshold: CPUUtilization > 50 for 1 consecutive periods of 60 seconds for the metric dimensions:
AutoScalingGroupName = cnv-scaling

Add 1 capacity units when 50 <= CPUUtilization < +infinity
300 seconds to warm up after each step

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now you have scaling policies set up for your Auto Scaling Group. Try them!

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

Generate (lots of) accesses to your web server cluster.
For example, open several tabs, or run:

```
ab -c 1000 -n 500000 http://<load-balancer-address>/test  
(Adjust this command to your needs)
```

Attention: Before you leave

Shut your Auto-Scaler down!

Don't waste credits!

Good work in the Cloud!