

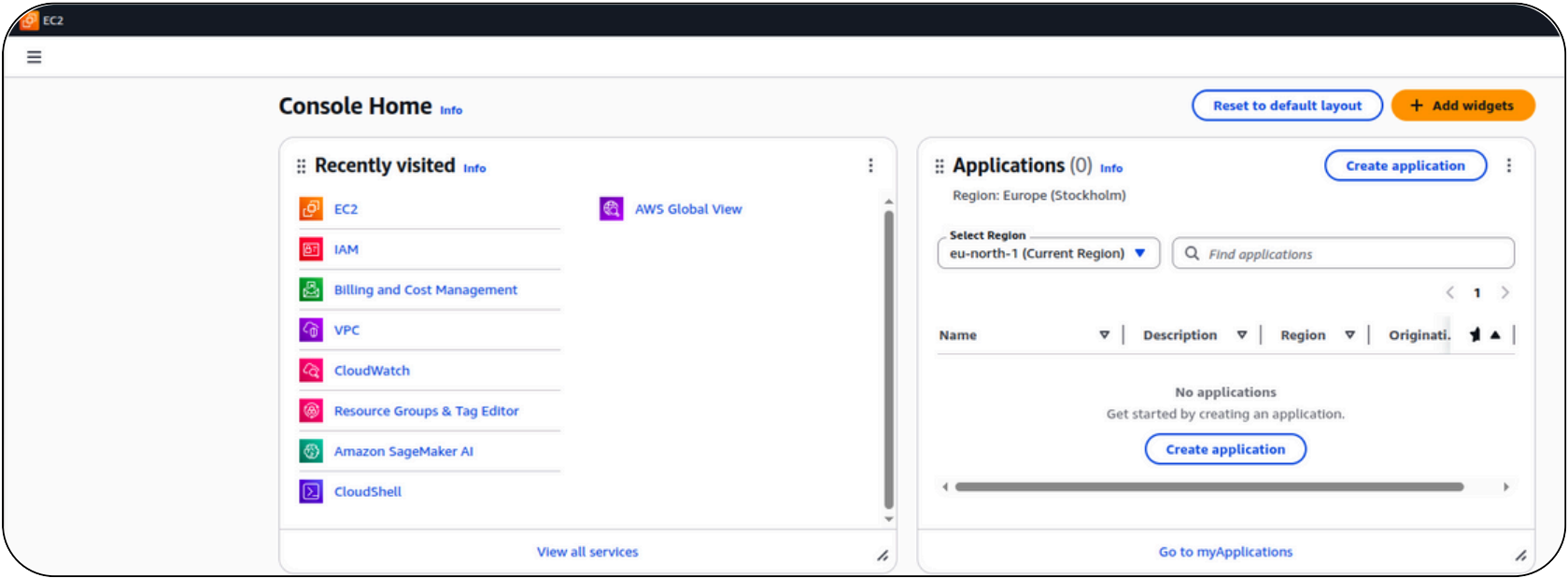
 **Jafar Khalaf**

 [LinkedIn](#)

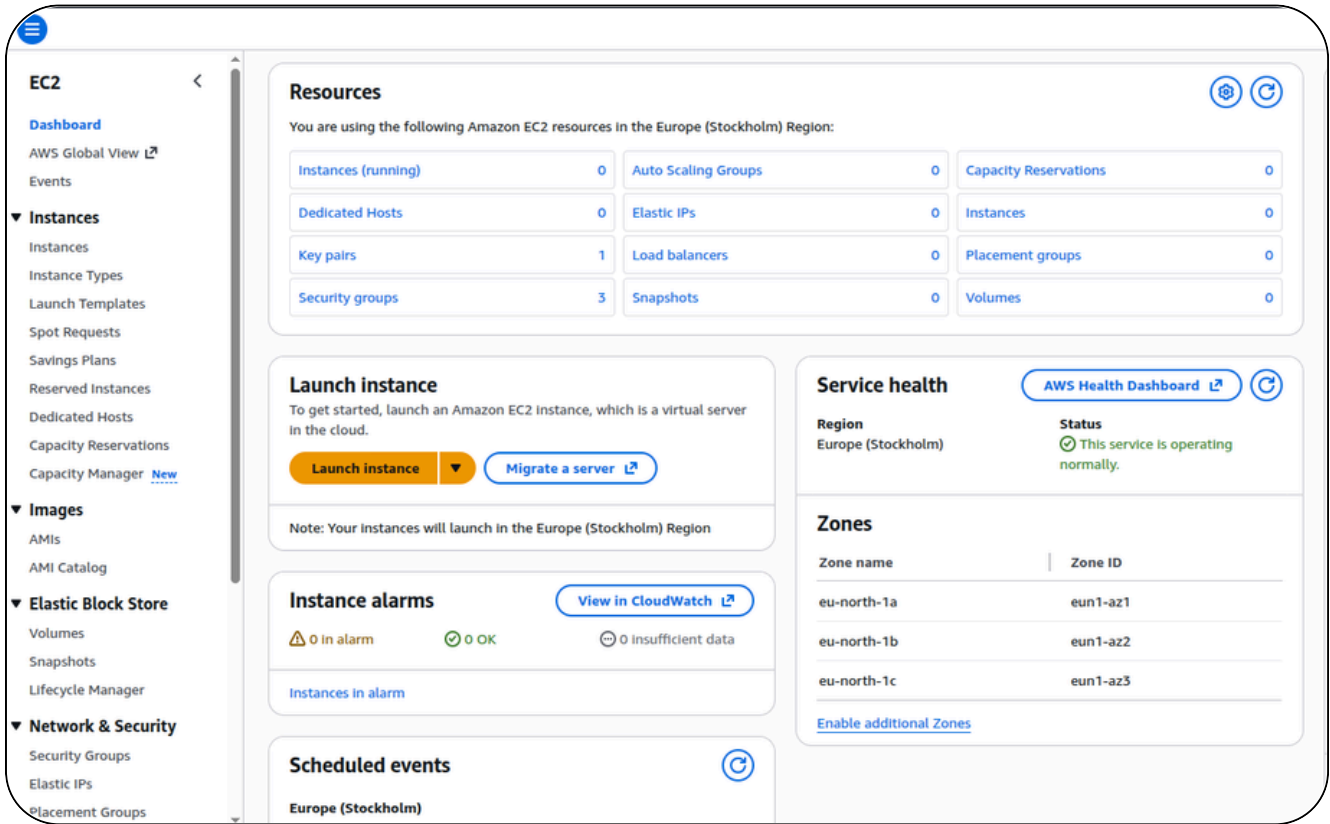
# Create Private Docker Registry



# Go to aws portal



# Go to EC2 service



# Create instance

## Name and tags [Info](#)

Name

docker\_registry

[Add additional tags](#)

## ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Recents

[Quick Start](#)

Amazon Linux  
aws

macOS  
Mac

Ubuntu  
ubuntu

Windows  
Microsoft

Red Hat  
Red Hat

SUSE Linux  
SUSE

Debian  
debian

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

### Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
ami-0fa91bc90632c73c9 (64-bit (x86)) / ami-0d14d7177686c6058 (64-bit (Arm))  
Virtualization: hvm    ENA enabled: true    Root device type: ebs

Free tier eligible

### Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	
64-bit (x86)	ami-0fa91bc90632c73c9	2025-10-22	ubuntu	<a href="#">Verified provider</a>

## ▼ Instance type [Info](#) | [Get advice](#)

### Instance type

t3.micro  
Family: t3    2 vCPU    1 GiB Memory    Current generation: true  
On-Demand Ubuntu Pro base pricing: 0.0143 USD per Hour    On-Demand RHEL base pricing: 0.0396 USD per Hour  
On-Demand SUSE base pricing: 0.0108 USD per Hour    On-Demand Linux base pricing: 0.0108 USD per Hour  
On-Demand Windows base pricing: 0.02 USD per Hour

Free tier eligible

☒ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

# Create private ssh key

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

docker\_registry\_key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA

RSA encrypted private and public key pair

ED25519

ED25519 encrypted private and public key pair

Private key file format

.pem

For use with OpenSSH

.ppk

For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel

Create key pair

# Add VPC , Subnet and Public ip

Network settings

VPC - required

vpc-0229756b79c440e53 (default)

Subnet

subnet-0cc6ce80b1d748c4b

Auto-assign public IP

Enable

Firewall (security groups)

Create security group

Security group name - required

Docker\_Registry\_Security\_Group

Description - required

Docker\_Registry\_Security\_Group

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type

ssh

Protocol

TCP

Port range

22

Remove

# Summary

Summary

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...read more

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

Launch instance

Preview code

# Show Instance

Instances (1)

Find Instance by attribute or tag (case-sensitive)

All states

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

< 1 >

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
<input type="checkbox"/>	docker_registry	i-0c53e29c47e70fb8a	Running	t3.micro	Initializing	View alarms +	eu-north-1a	ec2-51-20-96-136.eu-n...	51.20.96.136	-

© 2025 AWS. All rights reserved. | Privacy Policy | Terms of Service

Go to private key location

```
jafar22@jafar22:~/docker-registry$ ls  
docker_registry_key.pem
```

Access ec2

```
jafar22@jafar22:~/docker-registry$ sudo ssh -i docker_registry_key.pem ubuntu@51.20.96.136
```

EC2

```
ubuntu@ip-172-31-27-59:~$
```

Update and Upgrade

```
ubuntu@ip-172-31-27-59:~$ sudo apt update -y && apt upgrade -y
```

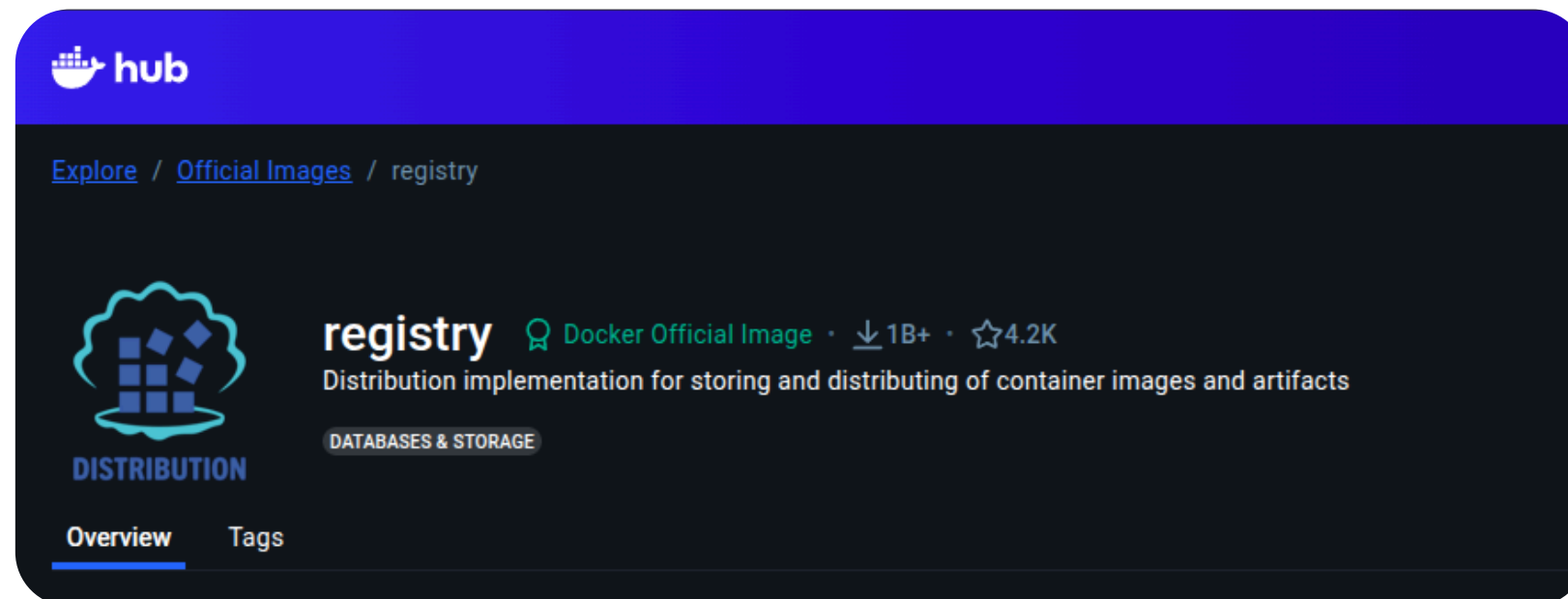
Install docker

```
ubuntu@ip-172-31-27-59:~$ sudo apt install docker.io -y
```

Check

```
ubuntu@ip-172-31-27-59:~$ docker version  
Client:  
Version:      28.2.2
```

We will use this image to be private registry from docker hub



Create directory for volume to save all images in host machine

```
ubuntu@ip-172-31-27-59:~$ sudo mkdir -p /opt/registry/data
```

pull and run registry image and configure it

```
ubuntu@ip-172-31-27-59:~$ sudo docker run -d --name registry -p5000:5000 --restart=always -v /opt/registry/data:/var/lib/registry registry:2
```

Show the container running

```
ubuntu@ip-172-31-27-59:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
9f0e96b568fe   registry:2    "/entrypoint.sh /etc..." 13 seconds ago Up 13 seconds 0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp registry
```

Edit inbound rules to open port 5000 to make users access to registry service

Inbound rules									
Inbound rules (1)									
Search									
<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description	
<input type="checkbox"/>	-	sgr-00eaf92866e59b57f	IPv4	SSH	TCP	22	0.0.0.0/0	-	

### Edit inbound rules

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

#### Inbound rules

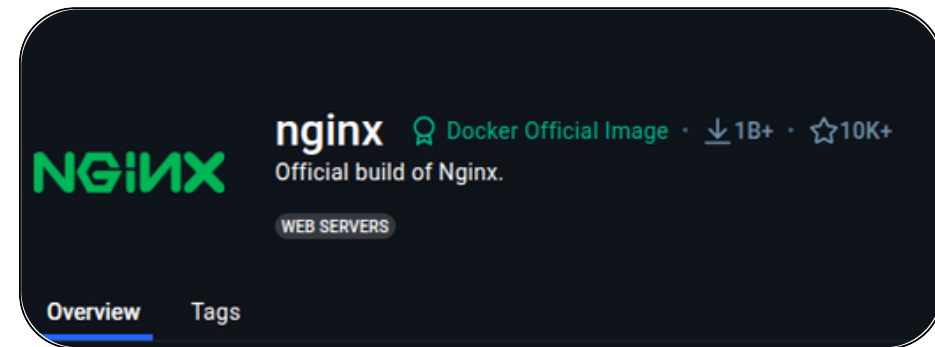
Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-00eaf92866e59b57f	SSH	TCP	22	Custom		Delete
-	Custom TCP	TCP	5000	Anywhe...	This port for docker registry	Delete

Add rule

Check if registry service running by this command and this is result

```
ubuntu@ip-172-31-27-59:~$ curl http://51.20.96.136:5000/v2/_catalog
{"repositories":[]}
```

To test private docker registry by using nginx image



Firstly pull image from docker hub

```
buntu@ip-172-31-27-59:~$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
d7ecded7702a: Pull complete
266626526d42: Pull complete
320b0949be89: Pull complete
d921c57c6a81: Pull complete
9def903993e4: Pull complete
52bc359bcbd7: Pull complete
e2f8e296d9df: Pull complete
Digest: sha256:1beed3ca46acebe9d3fb62e9067f03d05d5bfa97a00f30938a0a3580563272ad
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

show images

```
buntu@ip-172-31-27-59:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    d261fd19cb63   6 days ago    152MB
registry      2         26b2eb03618e   2 years ago    25.4MB
buntu@ip-172-31-27-59:~$
```

change image name

```
ubuntu@ip-172-31-27-59:~$ sudo docker tag nginx:latest 51.20.96.136:5000/nginx_test:v1
```

Try to push but there is issue because docker need https connection not http. For now we need to http

```
ubuntu@ip-172-31-27-59:~$ sudo docker push 51.20.96.136:5000/nginx_test:v1
The push refers to repository [51.20.96.136:5000/nginx_test]
Get "https://51.20.96.136:5000/v2/": http: server gave HTTP response to HTTPS client
```

Go to this file and add this script

```
ubuntu@ip-172-31-27-59:~$ sudo nano /etc/docker/daemon.json
```

```
{
  "insecure-registries": ["51.20.96.136:5000"]
}
```

Restart docker

```
ubuntu@ip-172-31-27-59:~$ sudo systemctl restart docker
```

Check

```
ubuntu@ip-172-31-27-59:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-11-10 10:52:49 UTC; 13s ago
 TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 2706 (dockerd)
```

Try again push

```
ubuntu@ip-172-31-27-59:~$ sudo docker push 51.20.96.136:5000/nginx_test:v1
The push refers to repository [51.20.96.136:5000/nginx_test]
d7217c60dca4: Pushed
d81df94f8d07: Pushed
99cd1b1b6a43: Pushed
2ced4cd78a7b: Pushed
8feb164cd673: Pushed
6e19587ac541: Pushed
36d06fe0cbc6: Pushing [=====> ] 75.53MB/78.62MB
```

Check from images by this command

```
ubuntu@ip-172-31-27-59:~$ curl http://51.20.96.136:5000/v2/_catalog
{"repositories":["nginx_test"]}
```