



Jafar Khalaf

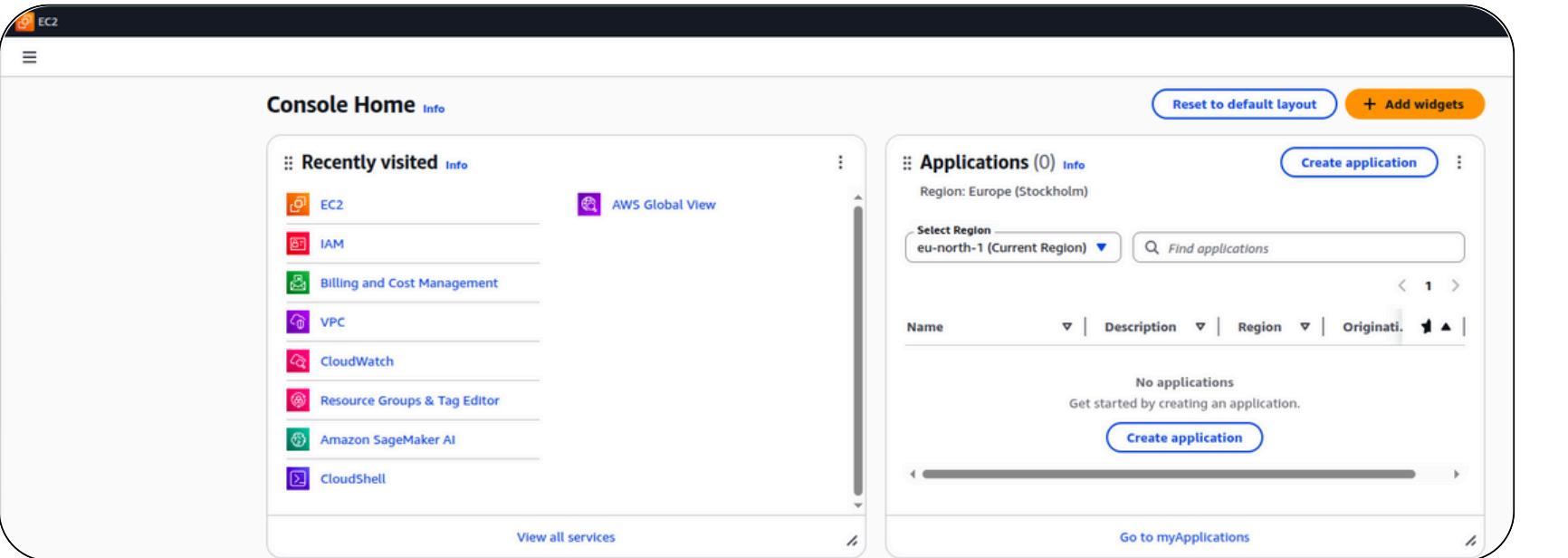


[LinkedIn](#)

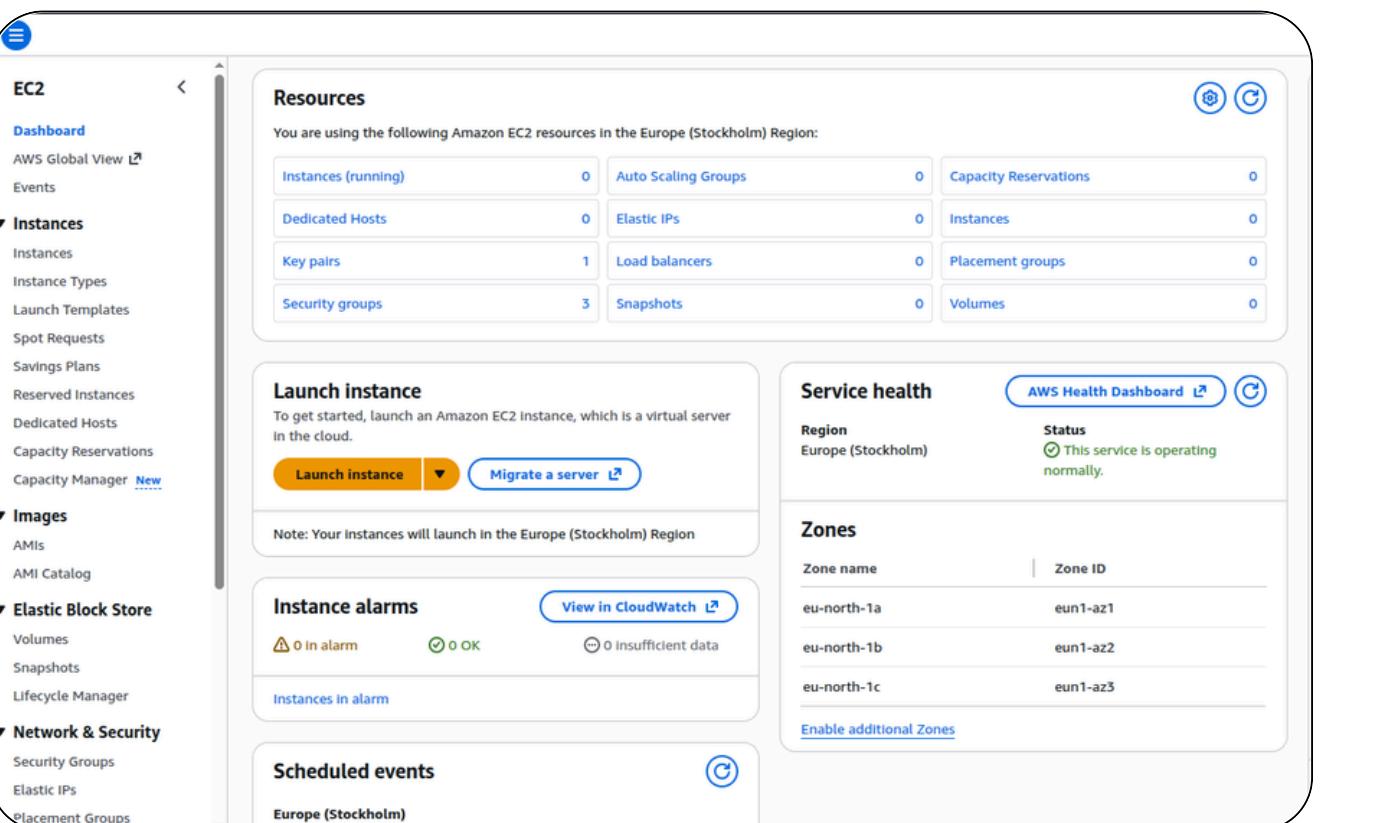
# Create Private Docker Registry



## Go to aws portal



## Go to EC2 service



# Create instance

**Name and tags** [Info](#)

**Name**

 [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](#).

[Recent](#) | [Quick Start](#)

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	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 [Free tier eligible](#)

ami-0fa91bc90632c73c9 (64-bit (x86)) / ami-0d14d7177686c6058 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

**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 <span>Verified provider</span>

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

**Instance type**

t3.micro [Free tier eligible](#)

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

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.

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** [Info](#)

**VPC - required** [Info](#)  
vpc-0229756b79c440e53 (default) ▾

**Subnet** [Info](#)  
subnet-0cc6ce80b1d748c4b VPC: vpc-0229756b79c440e53 Owner: 526056013916 Availability Zone: eu-north-1a (eu-n1-az1)  
Zone type: Availability Zone IP addresses available: 4091 CIDR: 172.31.16.0/20

**Auto-assign public IP** [Info](#)  
Enable

**Firewall (security groups)** [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.  
 Create security group  Select existing security group

**Security group name - required**  
Docker\_Registry\_Security\_Group

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-./@#%^&\_=;[]\$^

**Description - required** [Info](#)  
Docker\_Registry\_Security\_Group

**Inbound Security Group Rules**  
Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type [Info](#) Protocol [Info](#) Port range [Info](#)  
ssh TCP 22

# Summary

**▼ Summary**

**Number of instances** [Info](#)  
1

**Software Image (AMI)**  
Canonical, Ubuntu, 24.04, amd64... [read more](#)  
ami-0fa91bc90632c73c9

**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) <a href="#">Info</a>										
Last updated less than a minute ago										
<a href="#">Find Instance by attribute or tag (case-sensitive)</a> All states ▾										
	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS	Public IPv4 ... ▾	Elastic
	<input checked="" type="checkbox"/> docker_registry	i-0c53e29c47e70fb8a	<span>Running</span>	t3.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	eu-north-1a	ec2-51-20-96-136.eu-n...	51.20.96.136	-

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
```

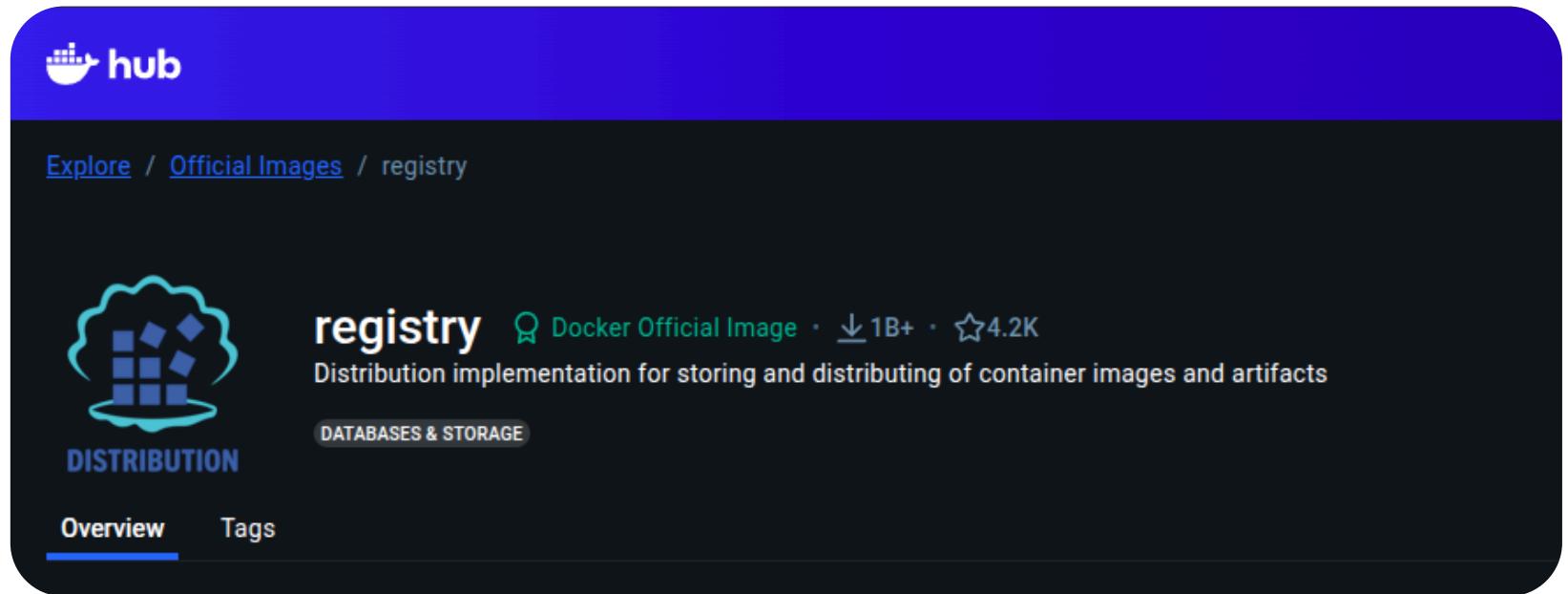
Install docker

```
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  
API version: 1.59
```

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

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



The screenshot shows the AWS CloudWatch Metrics interface. At the top, there are tabs for 'Metrics' (selected), 'CloudWatch Metrics Insights', 'Logs', and 'CloudWatch Metrics Insights'. Below the tabs, a search bar and a filter section are present. The main area displays a single metric named 'AWS Lambda' with a value of 1. The metric is shown in a table with columns: Metric Name, Value, Unit, and Last Value. The last value is displayed in a large green box.

**Inbound rules (1)**

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	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.

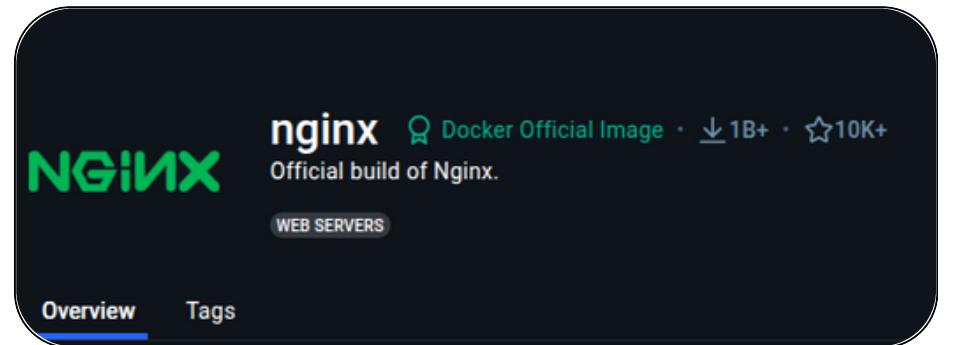
Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-00eaf92866e59b57f	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	5000	Anywhere	This port for docker registry

**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

```
Ubuntu@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
52bc359bcd7: Pull complete
e2f8e296d9df: Pull complete
Digest: sha256:1beed3ca46acebe9d3fb62e9067f03d05d5bfa97a00f30938a0a3580563272ad
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

show images

```
Ubuntu@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
Ubuntu@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)
      Tasks: 1
     Memory: 1.9M
        CPU: 0.000 CPU(s)
```

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"]}
```