

## AGENDA – AZURE CONTAINER REGISTRY (ACR)

### 1) SEARCH = Container registries

Microsoft Azure

Upgrade

Search resources, services, and docs (G+/)

Home >

Container registries

Default Directory (drgauravraj1993@gmail.onmicrosoft.com)

Create

Manage view

Refresh

Export to CSV

Open query

Assign tags

Filter for any field...

Subscription equals all

Resource group equals all

Location equals all

Add filter

Showing 0 to 0 of 0 records.

Name ↑↓	Type ↑↓	Resource group
---------	---------	----------------

Home > Container registries >

Create container registry

...

Azure Container Registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments. Use Azure container registries with your existing container development and deployment pipelines. Use Azure Container Registry Tasks to build container images in Azure on-demand, or automate builds triggered by source code updates, updates to a container's base image, or timers. [Learn more](#)

Project details

Subscription \*  
Resource group \*  
Create new

Free Trial  
rgjan

Instance details

Registry name \*  
Location \*  
Use availability zones ⓘ  
Pricing plan \* ⓘ

highlysecureregistry  
Poland Central  
☐  
Standard

Availability zones are activated on premium registries and in regions that support availability zones. [Learn more](#)

Review + create

< Previous

Next: Networking >

Home > Container registries >

## Create container registry

Basics **Networking** Encryption Tags Review + create


### Network connectivity

You can connect to this registry either publicly, via public IP addresses, or privately, using a private endpoint. [Learn more](#)



Connectivity configuration

☒ Public access (all networks)





☐ Private access (Recommended)

 Private access (Recommended) is only available for Premium pricing plans.

[Review + create](#) [< Previous](#) [Next: Encryption >](#)

HOME > **highlysecureregistry**  

Container registry

Search    Move  Delete

**Overview**

- Activity log
- Access control (IAM)
- Tags
- Quick start
- Events
- Settings
- Services
- Repository permissions
- Policies
- Monitoring
- Automation
- Help

**Essentials**

Resource group (move)	: rgjan	Login server	: highlysecureregistry.azurecr.io
Location	: Poland Central	Creation date	: 1/9/2025, 10:44 PM GMT+5:30
Subscription (move)	: Free Trial	Provisioning state	: Succeeded
Subscription ID	: 48f88df7-0d53-4866-a66f-82eb0ac469e3	Pricing plan	: Standard
Soft delete (Preview)	: Disabled		
Tags (edit)	: Add tags		

[Get started](#) [Monitoring](#) [Capabilities \(9\)](#) [Tutorials](#)

**Simplify container lifecycle management**

Container registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments. [Learn more](#)

+++++

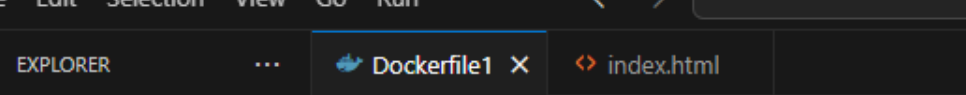
## AGENDA – Making image in local

-  ChatGPT 4o mini ▾

Sure! Here's a simple calculator app using HTML, CSS, and JavaScript all in one `index.html` file.

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
```

- 
- VS Code interface showing the Dockerfile editor. The Explorer sidebar on the left shows the file structure with 'Dockerfile1' selected. The Dockerfile editor on the right shows the following content:
- ```

1 FROM nginx
2 COPY index.html /usr/share/nginx/html

```

- +++++

### 1) Create VM

Home > CreateVm-canonical.ubuntu-24\_04-Its-server-20250109230028 | Overview >

**vmjan**  
Virtual machine

[Help me copy this VM in any region](#)

[Overview](#)

[Activity log](#)  
[Access control \(IAM\)](#)  
[Tags](#)  
[Diagnose and solve problems](#)  
[Connect](#)  
[Networking](#)  
[Settings](#)  
[Availability + scale](#)  
[Security](#)  
[Backup + disaster recovery](#)  
[Operations](#)  
[Monitoring](#)  
[Automation](#)  
[Help](#)

[Connect](#)
[Start](#)
[Restart](#)
[Stop](#)
[Hibernate](#)
[Capture](#)
[Delete](#)
[Refresh](#)
[Open in mobile](#)
[Feedback](#)
[CLI / PS](#)

### Essentials

Resource group

(move) : [rgjan](#)

Status

: Running

Location

: Japan East

Subscription

(move) : [Free Trial](#)

Subscription ID

: 48f88df7-0d53-4866-a66f-82eb0ac469e3

Tags

(edit) : [Add tags](#)

JSON View

Operating system

: Linux (ubuntu 24.04)

Size

: Standard D2s v3 (2 vcpus, 8 GiB memory)

Public IP address

: [74.176.58.204](#)

Virtual network/subnet

: [vmjan-vnet/default](#)

DNS name

: [Not configured](#)

Health state

: -

Time created

: 1/9/2025, 5:32 PM UTC

[Properties](#)
[Monitoring](#)
[Capabilities \(7\)](#)
[Recommendations](#)
[Tutorials](#)

Virtual machine

Computer name

vmjan

Operating system

Linux (ubuntu 24.04)

VM generation

V2

VM architecture

x64

Networking

Public IP address

[74.176.58.204](#) ( Network interface [vmjan358](#) )

Public IP address (IPv6)

-

Private IP address

10.0.0.4

Private IP address (IPv6)

-

```
azureuser@vmjan:~$ cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-color: #f4f4f9;
      margin: 0;
    }
    .calculator {
      background-color: #fff;
      padding: 20px;
```

azureuser@vmjan: ~

GNU nano 7.2

```
FROM nginx
COPY index.html /usr/share/nginx/html
```

#### 10) cat Dockerfile

```
azureuser@vmjan:~$ cat Dockerfile1
FROM nginx
COPY index.html /usr/share/nginx/html
azureuser@vmjan:~$
```

#### 11) sudo apt install nginx = INSTALL NGINX

sudo systemctl status nginx =

```
nginx [emerg] still could not bind()
root@vmjan:/home/azureuser/ACR# sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-01-09 17:57:20 UTC; 1min 33s ago
     Docs: man:nginx(8)
  Process: 4263 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 4264 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 4266 (nginx)
    Tasks: 3 (limit: 9459)
  Memory: 2.4M (peak: 2.5M)
     CPU: 21ms
   CGroup: /system.slice/nginx.service
           └─4266 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─4267 "nginx: worker process"
               └─4268 "nginx: worker process"

Jan 09 17:57:20 vmjan systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Jan 09 17:57:20 vmjan systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
root@vmjan:/home/azureuser/ACR#
```

11) Keep name as Dockerfile instead of Dockerfile1 as it is case sensitive and can cause issue

#### 12) docker build -t calculator .

```

root@vmjan:/home/azureuser/ACR# docker build -t calculator .
[+] Building 10.7s (7/7) FINISHED
==> [internal] load build definition from Dockerfile
==> => transferring dockerfile: 80B
==> [internal] load metadata for docker.io/library/nginx:latest
==> [internal] load .dockerignore
==> => transferring contest: 2B
==> [internal] load build context
==> => transferring contest: 3.69kB
==> [1/2] FROM docker.io/library/nginx:latest@sha256:42e017aaa1b5bb40dd0f6f774f857490ac7747d7ef73b391c774a41a8b994f15
==> resolve docker.io/library/nginx:latest@sha256:42e017aaa1b5bb40dd0f6f774f857490ac7747d7ef73b391c774a41a8b994f15
==> sha256:560e42bce4cd057dcab0090c002789af33bc8cbcbba95c8adb0d87285c85c75 43.84kB / 43.84kB
==> sha256:c375ae41bb29c0615ab18452822fedf1981f0ba7a7b00580b3d001h7ab682947 2.29kB / 2.29kB
==> sha256:fd674058ff8f0cfa7fba20c006fc0120541cbbad777f7f20df570d00f9e4d92 28.23kB / 28.23kB
==> sha256:2b95b9c5d9e5679c639357944d063c55c4c045e2cae21f64dfcfe5841e2ea59b 627B / 627B
==> sha256:42e017aaa1b5bb40dd0f6f774f857490ac7747d7ef73b391c774a41a8b994f15 10.27kB / 10.27kB
==> sha256:f876bf1cc63d905bb9c3ebc3ad39375bb0e22020959710d1a96e3f504060fa 8.58kB / 8.58kB
==> sha256:bd90674871f540eff8a4eaf3d4aa1ba504320ccbabfb0a217c4ea5c23b6144fd 957B / 957B
==> sha256:1e100dd2a0d75eb2ab2491daec5b300e99027ffdd528980612b03f3347b97e4 405B / 405B
==> sha256:da8cc133ff021c8b0ac7a6667e3a2e70ee0eb04f030e30600f59720017a069db 1.21kB / 1.21kB
==> extracting sha256:fd674058ff8f0cfa7fba20c006fc0120541cbbad777f7f20df570d00f9e4d92
==> sha256:c44f27300ea1a5e557aff07fbd5ec457d5cb05583f795b34f342d5550a00a5c 1.40kB / 1.40kB
==> extracting sha256:560e42bce4cd057dcab0090c002789af33bc8cbcbba95c8adb0d87285c85c75
==> extracting sha256:2b95b9c5d9e5679c639357944d063c55c4c045e2cae21f64dfcfe5841e2ea59b
==> extracting sha256:bd90674871f540eff8a4eaf3d4aa1ba504320ccbabfb0a217c4ea5c23b6144fd
==> extracting sha256:1e100dd2a0d75eb2ab2491daec5b300e99027ffdd528980612b03f3347b97e4
==> extracting sha256:da8cc133ff021c8b0ac7a6667e3a2e70ee0eb04f030e30600f59720017a069db
==> extracting sha256:c44f27300ea1a5e557aff07fbd5ec457d5cb05583f795b34f342d5550a00a5c
==> [2/2] COPY index.html /usr/share/nginx/html
==> exporting to image
==> => exporting layers
==> sha256:7654a5c1e88371050fe341207735d30fa23510442fa04b0ba2cc4b03abc7b0a5
==> now ready to docker.io/library/calculator

```

### 13) docker images

```

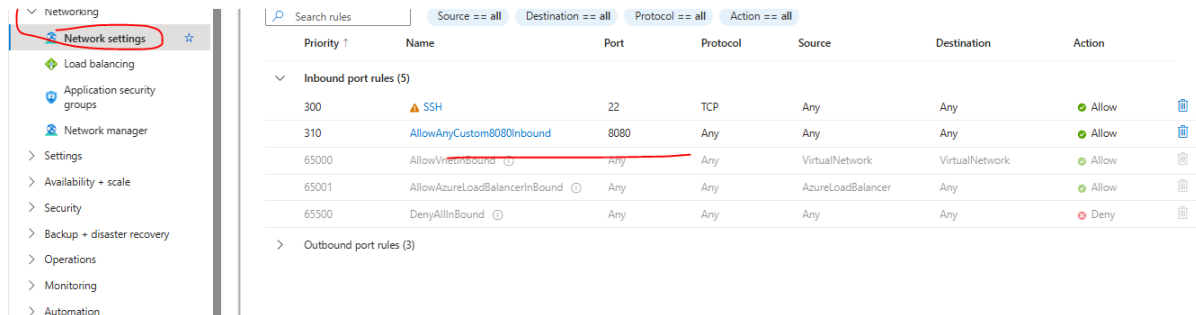
root@vmjan:/home/azureuser/ACR# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
calculator     latest    7654a5c1e883   9 seconds ago  192MB
root@vmjan:/home/azureuser/ACR#

```

14) **docker run -d -p 8080:80 calculator** = now run image

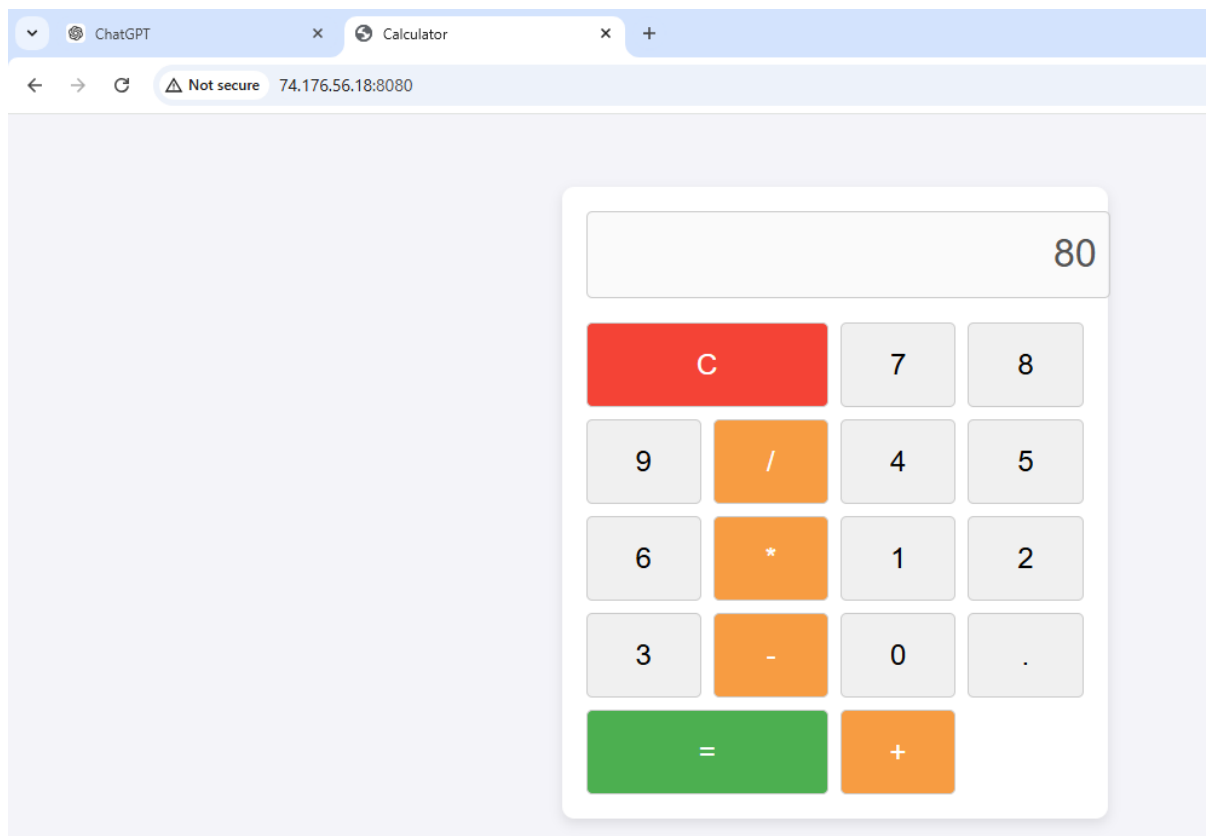
```
root@vmtabla:/home/azureuser/acr# docker run -d -p 8080:80 calculator  
15503b34f6c6fbd4dd5674495cb94b0aee2c399b34865e569e2b18971c91d9ae3  
root@vmtabla:/home/azureuser/acr#
```

15) open 8080 port on vm in nsg



Priority ↑	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	AllowAnyCustom8080Inbound	8080	Any	Any	Any	Allow
65000	AllowVirtualNetworkInbound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInbound	Any	Any	Any	Any	Deny

16) **74.176.56.18:8080** = public ip of vm:8080



+++++

### AGENDA – Put docker created image in ACR

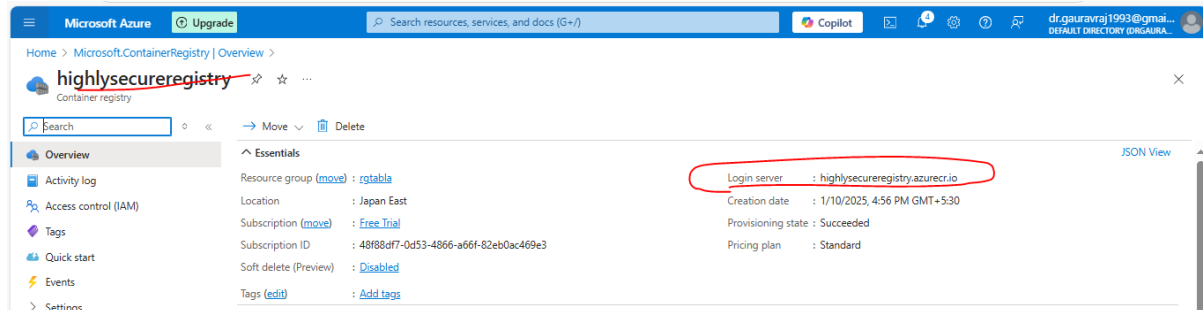
1) **docker images** = so we have a tag

```

root@vmtabla:/home/azureuser/acr# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
calculator           latest          7d48a9f2cd38   18 minutes ago 192MB
root@vmtabla:/home/azureuser/acr#

```

2) in created registry we have login server



url = **highlysecureregistry.azurecr.io**

3) **docker tag** command is used to rename the image

**docker tag calculator:latest highlysecureregistry.azurecr.io/calculator:latest**

4) **docker images**

```

root@vmtabla:/home/azureuser/acr# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
calculator           latest          7d48a9f2cd38   35 minutes ago 192MB
highlysecureregistry.azurecr.io/calculator latest          7d48a9f2cd38   35 minutes ago 192MB
root@vmtabla:/home/azureuser/acr#

```

5) SEARCH = azure container registry

<https://learn.microsoft.com/en-us/azure/container-registry/container-registry-get-started-azure-cli>

Filter by title

Azure Container Registry documentation

Container registries

Overview

Quickstarts

Create container registry - CLI

Create container registry - Portal

Create container registry - PowerShell

Create container registry - ARM template

Create container registry - Bicep

Send events to Event Grid - CLI

Manage content with client libraries

Build and push Java images - CLI

## Log in to registry

Before pushing and pulling container images, you must log in to the registry. To do so, use the **az acr login** command. Specify only the registry resource name when logging in with the Azure CLI. Don't use the fully qualified login server name.

Azure CLI

Copy

```
az acr login --name <registry-name>
```

Example:

Azure CLI

Copy

```
az acr login --name mycontainerregistry
```

The command returns a **Login Succeeded** message once completed.

## AGENDA – Install az cli in VM

**curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash**



- Azure CLI
- > Overview
- > Install, update and run
  - Overview
  - Install - Windows
  - Install - macOS
  - Install - Linux**
  - Run Azure CLI in a Docker container
  - Run Azure Cloud Shell
  - Update the Azure CLI
  - Azure CLI endpoints
- > Sign in
- > What's new

There are two options to install the Azure CLI on your system. You can download an install script that runs the install commands for you, or you can execute the install commands yourself in a step-by-step process. Both methods are provided here:

## Option 1: Install with one command

The easiest way to install the Azure CLI is through a script maintained by the Azure CLI team. This script runs all installation commands in one step. This script is downloaded via `curl` and piped directly to `bash` to install the CLI.

If you wish to inspect the contents of the script yourself before executing, download the script first using `curl` and inspect it in your favorite text editor.

```
Bash
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

## Option 2: Step-by-step installation instructions

+++++

## AGENDA = CONNECTING TO ACR and PUSHING Image to ACR

1) **az login** = Authentication will happen to VM

```
root@vmtabla:/home/azureuser/acr# az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code IPMLT3V3N to authenticate.
```

2) **az acr login --name highlysecureregistry** = we will login into acr registry

```
root@vmtabla:/home/azureuser/acr# az acr login --name highlysecureregistry
Login Succeeded
root@vmtabla:/home/azureuser/acr#
```

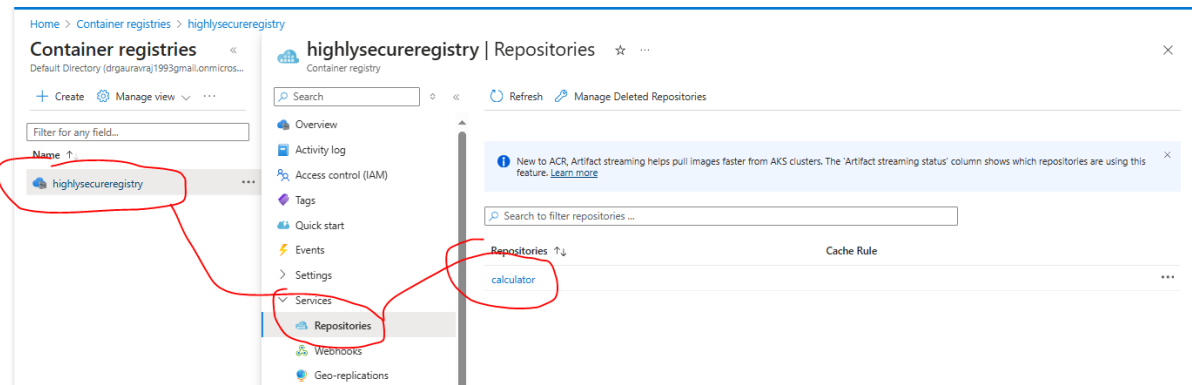
3) **docker images**

```
root@vmtabla:/home/azureuser/acr# docker images
REPOSITORY                                TAG      IMAGE ID      CREATED      SIZE
calculator                                latest   7d48a9f2cd38  2 hours ago  192MB
highlysecureregistry.azurecr.io/calculator latest   7d48a9f2cd38  2 hours ago  192MB
root@vmtabla:/home/azureuser/acr#
```

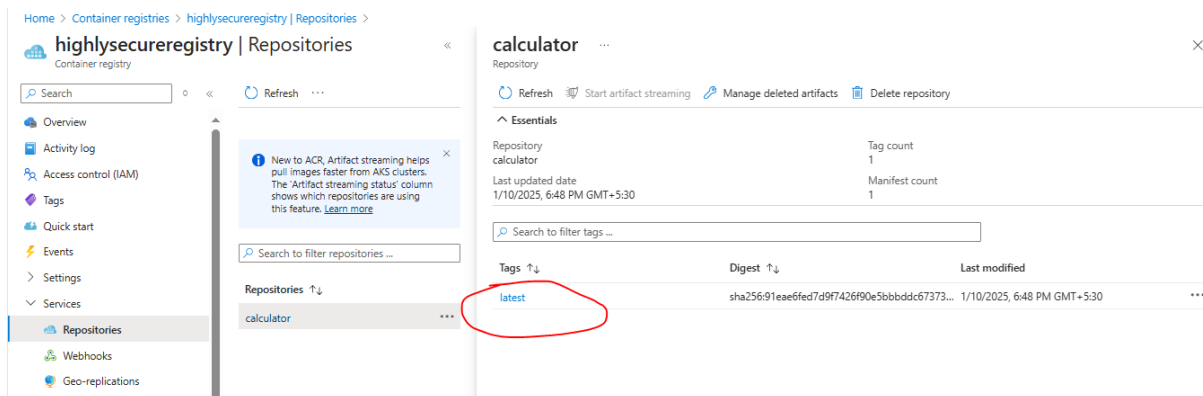
4) **docker push highlysecureregistry.azurecr.io/calculator** = now pushed image into acr

```
root@vmtabla:/home/azureuser/acr# docker push highlysecureregistry.azurecr.io/calculator
Using default tag: latest
The push refers to repository [highlysecureregistry.azurecr.io/calculator]
728e1c879f08: Pushed
af90855d8344: Pushed
ad206e285c61: Pushed
24aeff94f79e: Pushed
d567f5b4517e: Pushed
14a96b2ac595: Pushed
c4c8312766f1: Pushed
8b296f486960: Pushed
latest: digest: sha256:91eae6fed7d9f7426f90e5bbbddc6737374d81287348b72817fd60f036a59574 size: 1986
root@vmtabla:/home/azureuser/acr#
```

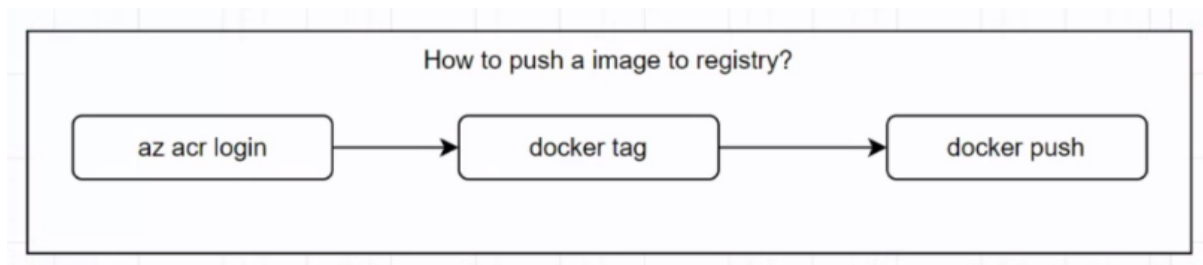
5) Go to portal acr



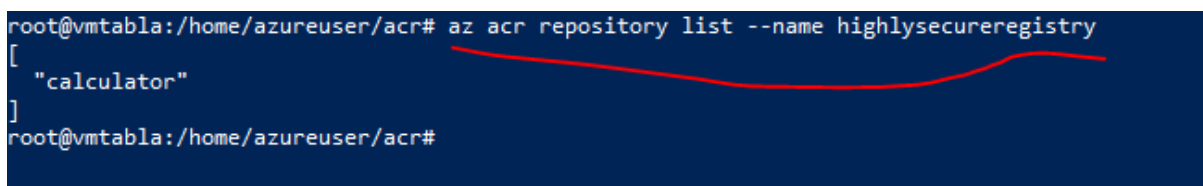
6) Now we can see the tag passed into it i.e. "latest"



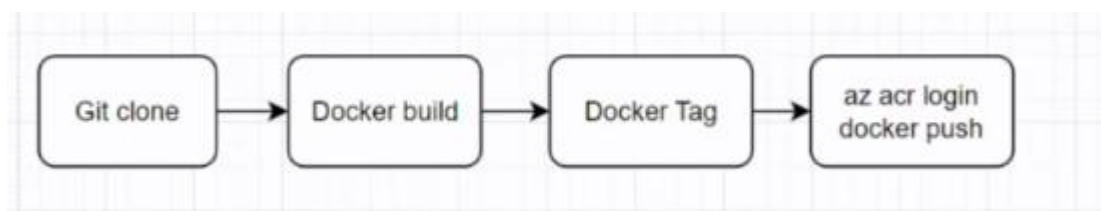
7) So whole steps were



8) **az acr repository list --name highlysecureregistry** = list how many images our registry has on cli



9)



10) Image





