Azure Practical Task

> ## Practical task
>
> 1. Write one or several bash scripts (using az cli) which:
>   a. Create Resource Group (RG), VNet, Subnet, Azure Container Registry (ACR), VM, public IP address for VM, Network Security group (NSG)
>   b. Push a java spring-petlinic container image you built in “Docker” module to ACR
>   c. ACR should be private
>   d. Install Docker on the VM
>   e. Run container on the VM
> 2. Verify that application is accessible in your browser
> 3. Remove the resource group

1. Write one or several bash scripts (using az cli) which:

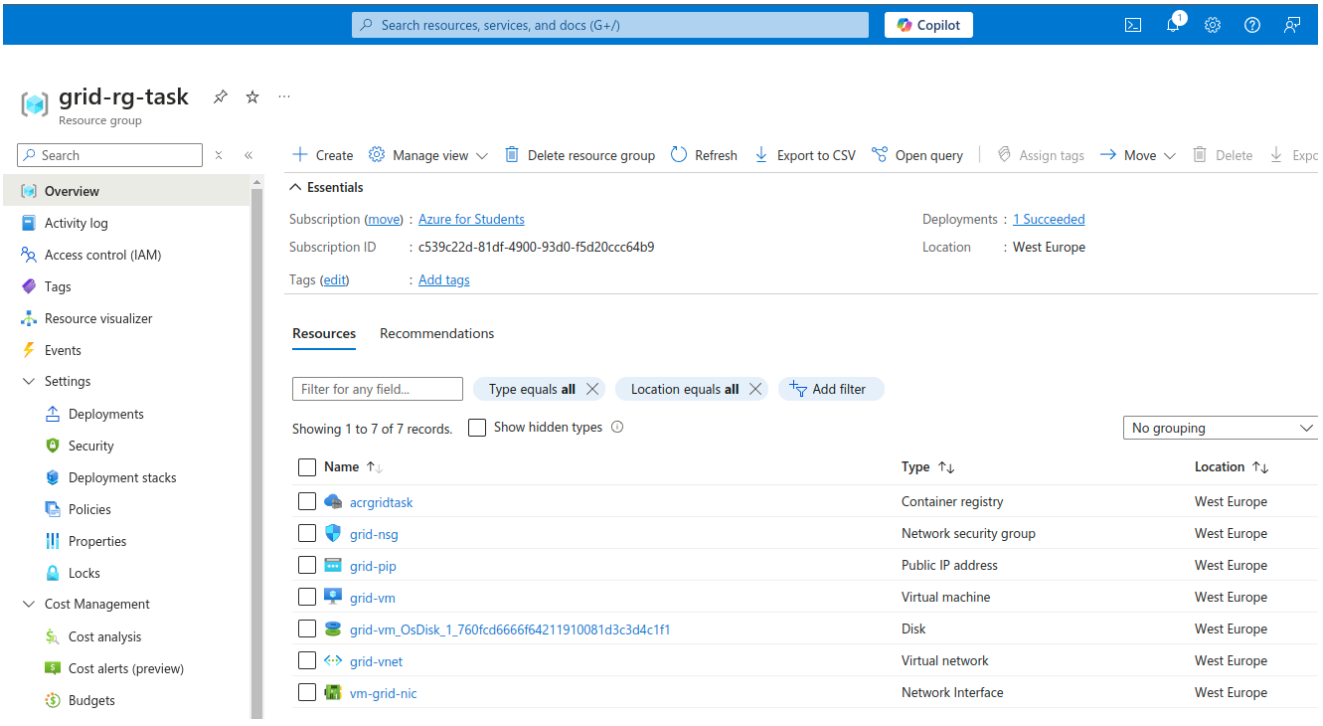
a. Create Resource Group (RG), VNet, Subnet, Azure Container Registry (ACR), VM, public IP address for VM, Network Security group (NSG)

Used script:

```
#!/bin/bash

RG_NAME=grid-rg-task
LOCATION=westeurope
VNET_NAME=grid-vnet
SUBNET_NAME=grid-subnet
NSG_NAME=grid-nsg
PUB_IP_NAME=grid-pip
VM_NIC_NAME=vm-grid-nic
VM_NAME=grid-vm
VNET_NET="10.0.0.0/16"
SUBNET_NET="10.0.1.0/24"
VM_PRIV_IP="10.0.1.4"
ACR_NAME=acrgridtask
ACR_SKU=basic


az group create \
    --name "$RG_NAME" \
    --location "$LOCATION"
```

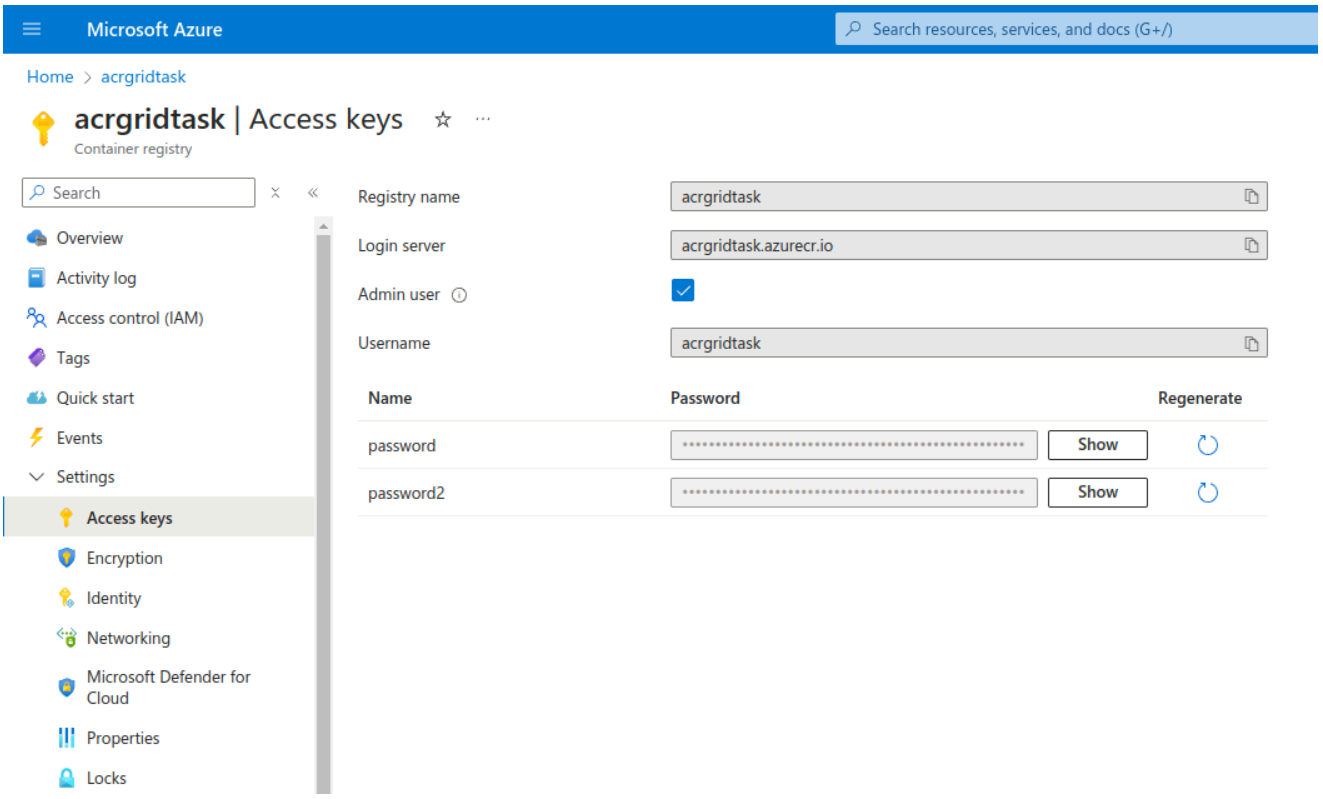
```
az network vnet create \
    --resource-group "$RG_NAME" \
    --name "$VNET_NAME" \
    --address-prefixes "$VNET_NET"

az network nsg create \
    --resource-group "$RG_NAME" \
    --name "$NSG_NAME"

az network nsg rule create \
    --resource-group "$RG_NAME" \
    --nsg-name "$NSG_NAME" \
    --name allow-SSH \
    --access allow \
    --protocol tcp \
    --priority 1000 \
    --destination-port-ranges 22

az network nsg rule create \
    --resource-group "$RG_NAME" \
    --nsg-name "$NSG_NAME" \
    --name allow-HTTP \
    --access allow \
    --protocol tcp \
    --priority 1001 \
    --destination-port-ranges 80

az network vnet subnet create \
    --resource-group "$RG_NAME" \
    --vnet-name "$VNET_NAME" \
    --name "$SUBNET_NAME" \
    --address-prefixes "$SUBNET_NET" \
    --network-security-group "$NSG_NAME"

az network public-ip create \
    --resource-group "$RG_NAME" \
    --name "$PUB_IP_NAME"

az network nic create \
    --resource-group "$RG_NAME" \
    --name "$VM_NIC_NAME" \
    --vnet-name "$VNET_NAME" \
    --subnet "$SUBNET_NAME" \
    --public-ip-address "$PUB_IP_NAME" \
    --network-security-group "$NSG_NAME" \
    --private-ip-address "$VM_PRIV_IP"

az vm create \
    --resource-group "$RG_NAME" \
    --name "$VM_NAME" \
```

```
    --image Ubuntu2404 \
    --nics "$VM_NIC_NAME" \
    --admin-username azureuser \
    --ssh-key-values "./ssh-keys"

az acr create \
    -g "$RG_NAME" \
    -n "$ACR_NAME" \
    --sku "$ACR_SKU"
```



b. Push a java spring-petlinic container image you built in “Docker” module to ACR

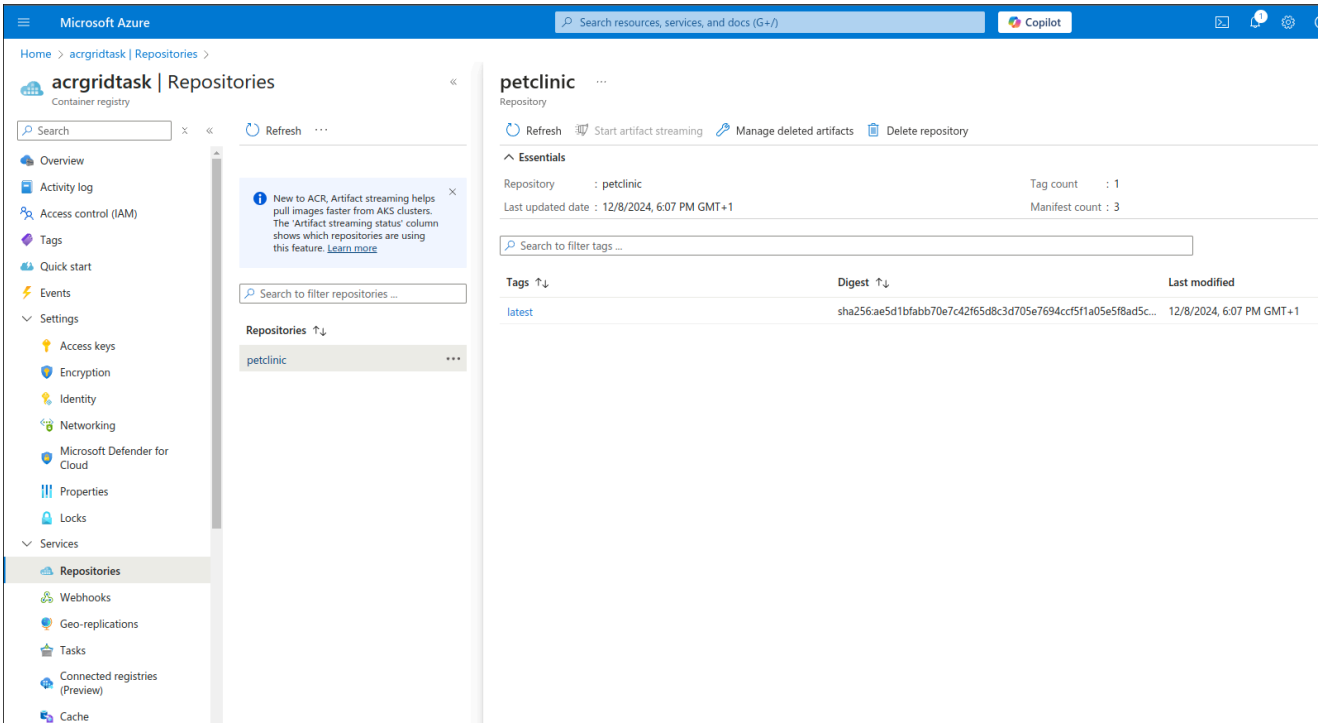
First authorize Docker to access created ACR. ACR credentials can be found in `Access keys` setting for ACR.




```
06:02:43 adrwal@olek-desktop-pc ~ → docker login acrgridtask.azurecr.io
Username: acrgridtask
Password:
WARNING! Your password will be stored unencrypted in /home/adrwal/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
06:05:39 adrwal@olek-desktop-pc ~ →
```

Now we can push and pull to ACR.

```
06:05:39 adrwal@olek-desktop-pc ~ → docker tag my-petclinic:latest acrgridtask.azurecr.io/petclinic
06:07:02 adrwal@olek-desktop-pc ~ → docker push acrgridtask.azurecr.io/pet
acrgridtask.azurecr.io/petclinic         acrgridtask.azurecr.io/petclinic:latest
06:07:02 adrwal@olek-desktop-pc ~ → docker push acrgridtask.azurecr.io/petclinic
Using default tag: latest
The push refers to repository [acrgridtask.azurecr.io/petclinic]
5664b15f108b: Pushed
701c983262e9: Pushed
86fd78f09988: Pushed
e3269fcfc82e: Pushed
c44269e277af: Pushed
a62778643d56: Pushed
9aafee56e35e: Pushed
a3d717a89751: Pushed
0cb5c07f8edd: Pushed
0baecf37abee: Pushed
4aa0ea1413d3: Pushed
0bab15eea81d: Pushed
c83c31ce41af: Pushed
8ffb3c3cf71a: Pushed
f716db4e3c8d: Pushed
51a849027c78: Pushed
3214acf345c0: Pushed
1acfb4a268db: Pushed
bfb59b82a9b6: Pushed
8bd314445e66: Pushed
02fd24817d0f: Pushed
7c12895b777b: Pushed
cc249665630b: Pushed
9aee425378d2: Pushed
a52506876bfa: Pushed
57c7c63c9c91: Pushed
52210cfa5d4d: Pushed
f58f8cd53aa0: Pushed
da7816fa955e: Pushed
84845657fbc5: Pushed
latest: digest: sha256:ae5d1bfabb70e7c42f65d8c3d705e7694ccf5f1a05e5f8ad5c5271c054507625 size: 856
06:07:50 adrwal@olek-desktop-pc ~ →
```



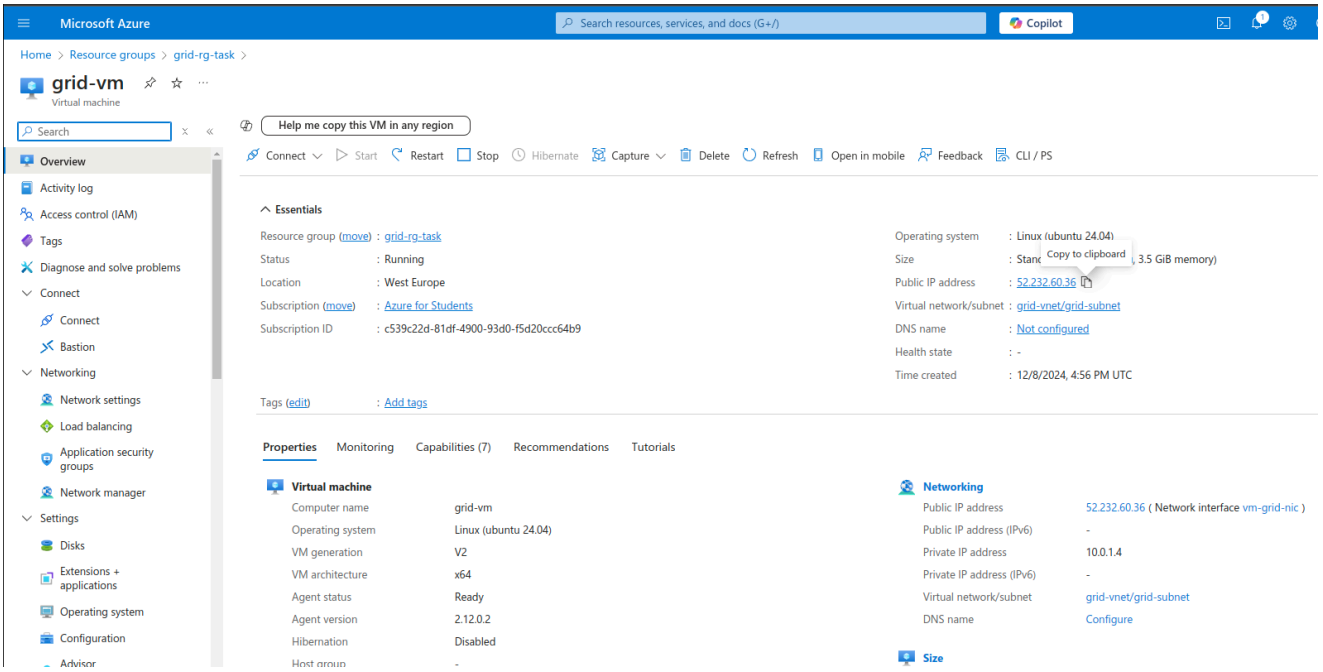
c. ACR should be private

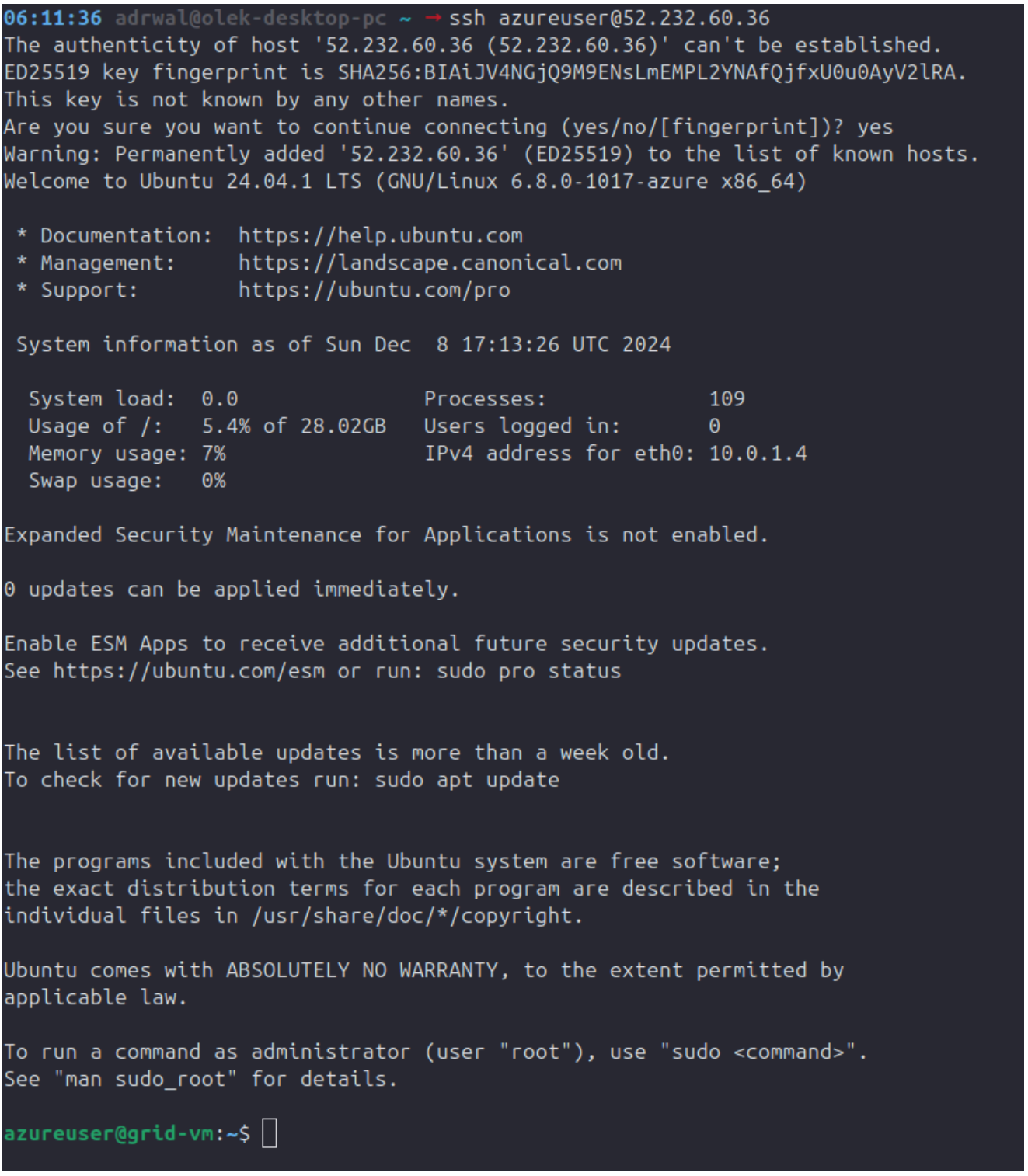
To verify that ACR is private we can try to run the image after removing previously saved credentials to ACR in Docker.

```
06:10:49 adrwal@olek-desktop-pc ~ → docker logout acrgridtask.azurecr.io
Removing login credentials for acrgridtask.azurecr.io
06:11:04 adrwal@olek-desktop-pc ~ → docker rmi acrgridtask.azurecr.io/petclinic
Untagged: acrgridtask.azurecr.io/petclinic:latest
06:11:13 adrwal@olek-desktop-pc ~ → docker run -it --rm acrgridtask.azurecr.io/petclinic
Unable to find image 'acrgridtask.azurecr.io/petclinic:latest' locally
docker: Error response from daemon: failed to resolve reference "acrgridtask.azurecr.io/petclinic:latest": failed to authorize: failed to fe
quest to https://acrgridtask.azurecr.io/oauth2/token?scope=repository%3Apetclinic%3Apull&service=acrgridtask.azurecr.io: 401 Unauthorized.
See 'docker run --help'.
06:11:18 adrwal@olek-desktop-pc ~ →
06:11:35 adrwal@olek-desktop-pc ~ →
06:11:36 adrwal@olek-desktop-pc ~ →
06:11:36 adrwal@olek-desktop-pc ~ →
06:11:36 adrwal@olek-desktop-pc ~ →
06:11:36 adrwal@olek-desktop-pc ~ →
06:11:36 adrwal@olek-desktop-pc ~ →
```

d. Install Docker on the VM

First SSH into VM. Public IP can be found on VM overview page.





Now download docker with apt.

```
azureuser@grid-vm:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease
Get:5 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Pack
Get:6 http://azure.archive.ubuntu.com/ubuntu noble/universe Translatio
Get:7 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Comp
Get:8 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-
Get:9 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 Pa
Get:10 http://azure.archive.ubuntu.com/ubuntu noble/multiverse Transla
```

```
azureuser@grid-vm:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite de
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base
0 upgraded, 8 newly installed, 0 to remove and 44 not
Need to get 80.1 MB of archives.
After this operation, 303 MB of additional disk space
Do you want to continue? [Y/n] Y
Get:1 http://azure.archive.ubuntu.com/ubuntu noble/un
```

```
azureuser@grid-vm:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
azureuser@grid-vm:~$
```

e. Run container on the VM

First we need to authenticate Docker to access ACR.

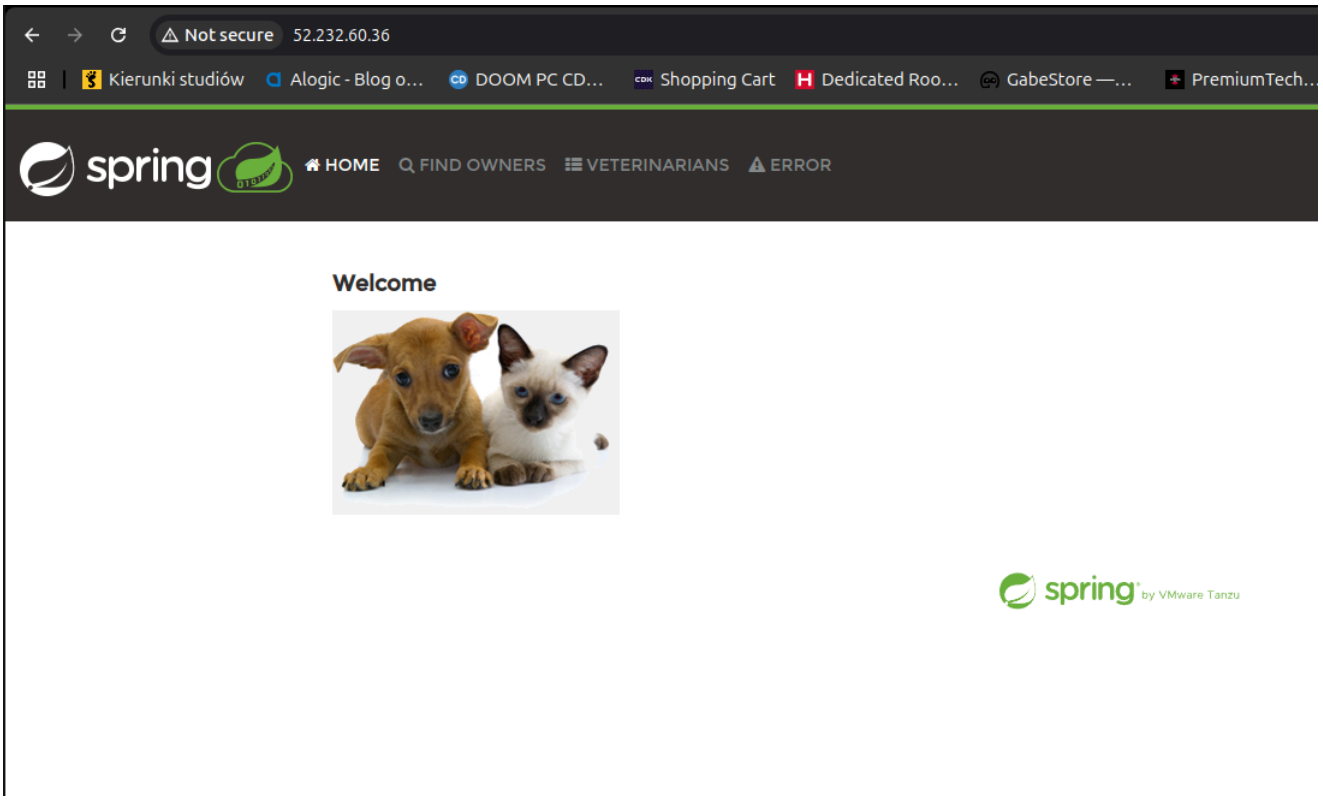
```
azureuser@grid-vm:~$ sudo docker login acrgridtask.azurecr.io
Username: acrgridtask
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Now we can run the `acrgridtask.azurecr.io/petclinic` container that we have previously pushed to ACR.

```
azureuser@grid-vm:~$ sudo docker run -it --rm -p 80:8080 acrgridtask.azurecr.io/petclinic
Unable to find image 'acrgridtask.azurecr.io/petclinic:latest' locally
latest: Pulling from petclinic
0baecf37abee: Pull complete
bfb59b82a9b6: Pull complete
8ffb3c3cf71a: Pull complete
a62778643d56: Pull complete
7c12895b777b: Pull complete
3214acf345c0: Pull complete
5664b15f108b: Pull complete
0bab15eea81d: Pull complete
4aa0ea1413d3: Pull complete
da7816fa955e: Pull complete
9aee425378d2: Pull complete
```

```
Status: Downloaded newer image for acrgridtask.azurecr.io/petclinic:latest


              |\      _,,,--,,_
             /,`.-'`'   ._  \-;;,_
  _______ __|,4-  ) )_   .;.(__`'-'__     ___ __    _ ___ _______
 |       | '---''(_/._)-'(_\_)   |   |   |   |  |  | |   |       |
 |    _  |    ___|_     _|       |   |   |   |   |_| |   |       | __ _ _
 |   |_| |   |___  |   | |       |   |   |   |       |   |       | \ \ \ \
 |    ___|    ___| |   | |      _|   |___|   |  _    |   |      _|  \ \ \ \
 |   |   |   |___  |   | |     |_|       |   | | |   |   |     |_    ) ) ) )
 |___|   |_______| |___| |_______|_______|___|_|  |__|___|_______|  / / / /
 ==========================================================================/_/_/_/

:: Built with Spring Boot :: 3.3.5


2024-12-08T17:19:32.845Z  INFO 1 --- [           main] o.s.s.petclinic.PetClinicApplicatio
 by root in /)
2024-12-08T17:19:32.862Z  INFO 1 --- [           main] o.s.s.petclinic.PetClinicApplicatio
2024-12-08T17:19:36.230Z  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDel
2024-12-08T17:19:36.406Z  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDel
2024-12-08T17:19:38.795Z  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebSe
2024-12-08T17:19:38.837Z  INFO 1 --- [           main] o.apache.catalina.core.StandardServ
2024-12-08T17:19:38.839Z  INFO 1 --- [           main] o.apache.catalina.core.StandardEngi
2024-12-08T17:19:39.145Z  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]
2024-12-08T17:19:39.149Z  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationCo
2024-12-08T17:19:40.396Z  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource
2024-12-08T17:19:41.015Z  INFO 1 --- [           main] com.zaxxer.hikari.pool.HikariPool
user=SA
2024-12-08T17:19:41.020Z  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource
```

2. Verify that application is accessible in your browser



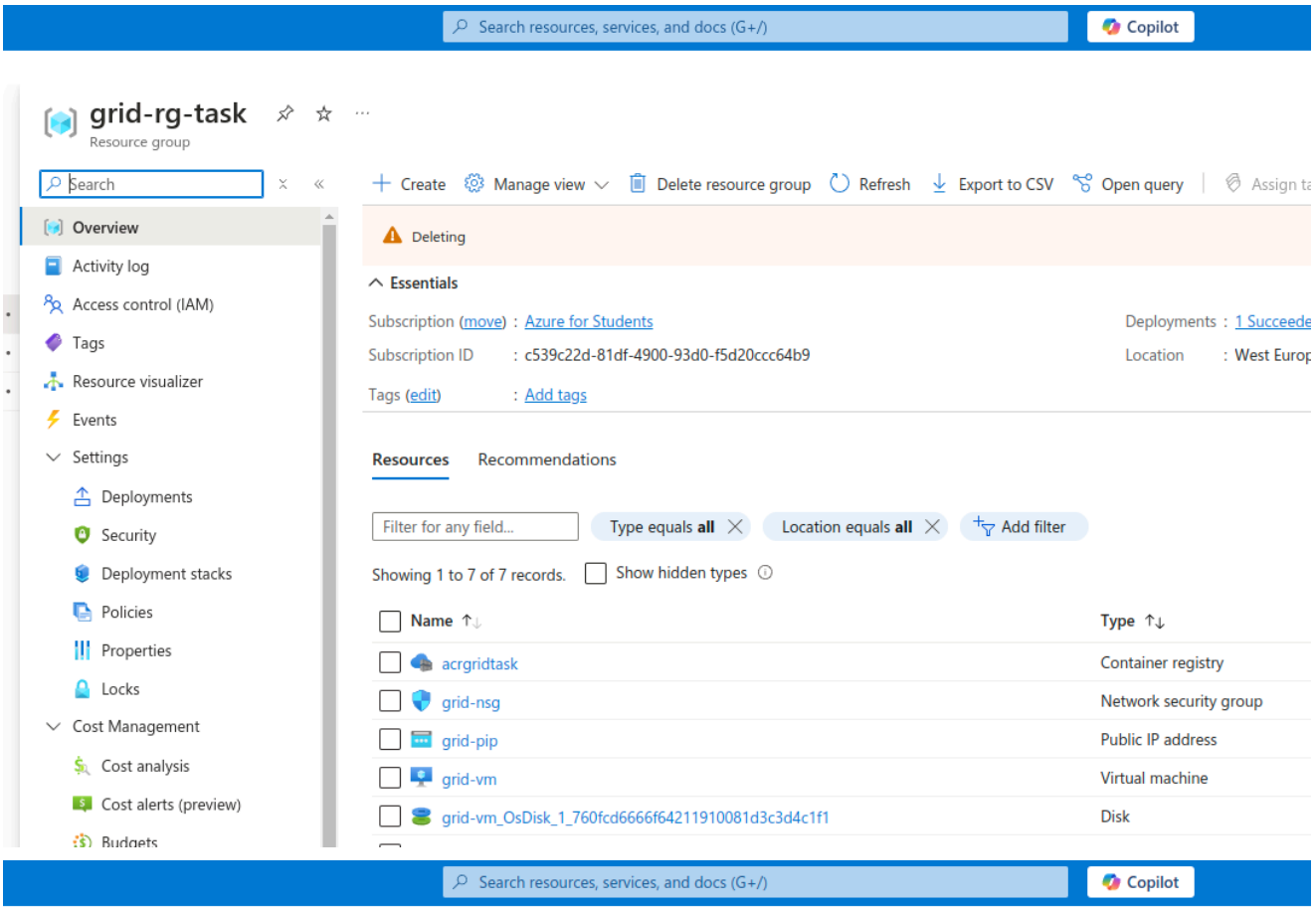
3. Remove the resource group

We can remove resource group with one `az` command.

```
az group delete \
    --name grid-rg-task \
    --yes \
    --no-wait
```

```
06:24:43 adrwal@olek-desktop-pc ~ → az group delete \
    --name grid-rg-task \
    --yes \
    --no-wait
06:24:49 adrwal@olek-desktop-pc ~ →
```

Now resource group is marked with "Deleting" warning and after about a minute it has been successfully deleted with all of the created resources.



Resource not found

Get support Perform self-diagnostics

Summary

Session ID
feb6d59dd5ea4345bfdd7890ec493fa8

Resource ID
/subscriptions/c539c22d-81df-4900-93d0-f5d20ccc64b...

Extension
HubsExtension

Content
ResourceMenuBlade

Error code
404

Details
The resource was not found, it may have been deleted. If this was launched from a pinned tile on the dashboard, it should be removed.

Resource ID: /subscriptions/c539c22d-81df-4900-93d0-f5d20ccc64b9/resourceGroups/grid-rg-task

Status Code: 404