😊 LAB: PUBLISHING TO AZURE CONTAINER REGISTRY

- In the last lab, we had seen how to democratize an application. I now want to publish that image that we have on the Linux VM that has our application onto the Azure Container registry that we created earlier on.
- See if you want, let's say other developers or other team members now have the ability to download the image that you have created for the application. So, they could also go and now create containers out of that image. We need to have that image located somewhere.
- We can make use of this private registry that's known as the Azure Container Registry.
- For that we are going to install Azure Command Line Interface (CLI) on to our virtual machine. This is a command line utility that you can use to work with your Azure based services. Hence, using this command line utility, I want to publish that image from my local machine onto the Azure Container Registry.
- For this, the first step is to go ahead and install the Azure CLI on our Linux based VM.
- In terms of installing the CLI I've used Microsoft Documentation. A copy of the first set of commands. This is to get the packages needed for the installation process.
- https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-linux?pivots=apt
- Use the above link to go through the commands that you are going to use to install CLI. I would suggest you to use option 2 to install CLI.

Option 2: Step-by-step installation instructions

If you prefer a step-by-step installation process, complete the following steps to install the Azure CLI.

1. Get packages needed for the installation process:

```bash
sudo apt-get update
sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg
```

2. Download and install the Microsoft signing key:

```bash
sudo mkdir -p /etc/apt/keyrings
curl -sLS https://packages.microsoft.com/keys/microsoft.asc |
    gpg --dearmor |
    sudo tee /etc/apt/keyrings/microsoft.gpg > /dev/null
sudo chmod go+r /etc/apt/keyrings/microsoft.gpg
```

3. Add the Azure CLI software repository:

```bash
AZ_DIST=$(lsb_release -cs)
echo "deb [arch=`dpkg --print-architecture` signed-by=/etc/apt/keyrings/microsoft.gpg] https://packages
    sudo tee /etc/apt/sources.list.d/azure-cli.list
```

4. Update repository information and install the `azure-cli` package:

```bash
sudo apt-get update
sudo apt-get install azure-cli
```

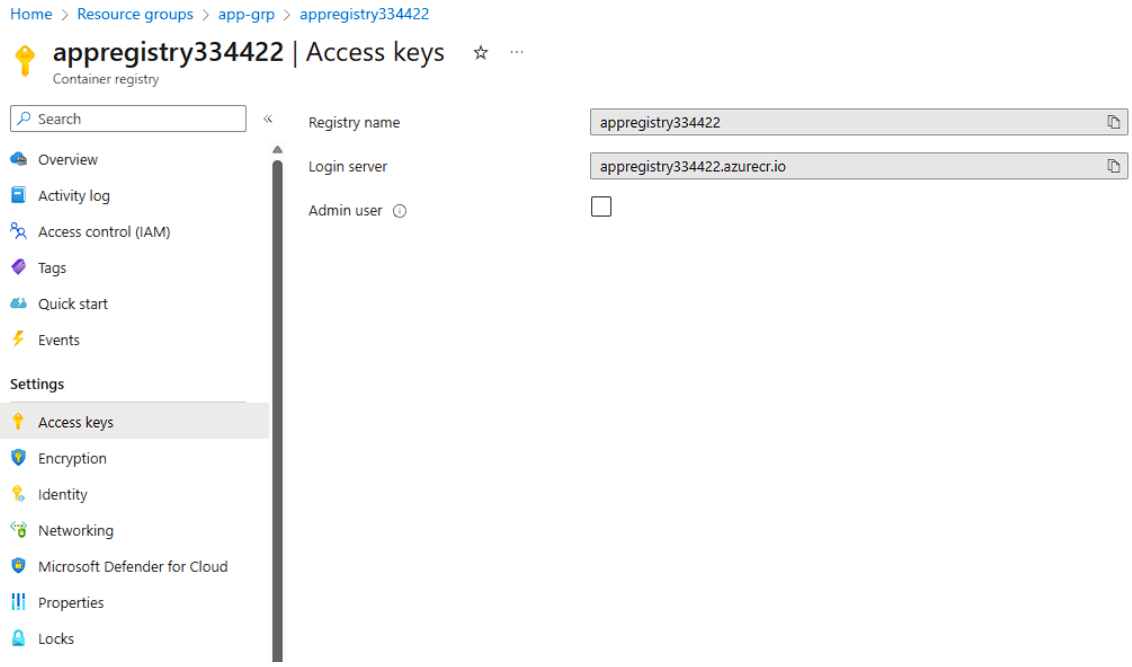
```
demousr@dockervm:~/publish$ sudo apt-get update
sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg
Hit:1 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:6 https://packages.microsoft.com/repos/azure-cli jammy InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu4).
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.15).
gnupg is already the newest version (2.2.27-3ubuntu2.1).
apt-transport-https is already the newest version (2.4.11).
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
demousr@dockervm:~/publish$
```

```
demousr@dockervm:~/publish$ sudo mkdir -p /etc/apt/keyrings
curl -sLS https://packages.microsoft.com/keys/microsoft.asc |
    gpg --dearmor |
    sudo tee /etc/apt/keyrings/microsoft.gpg > /dev/null
sudo chmod go+r /etc/apt/keyrings/microsoft.gpg
demousr@dockervm:~/publish$
```

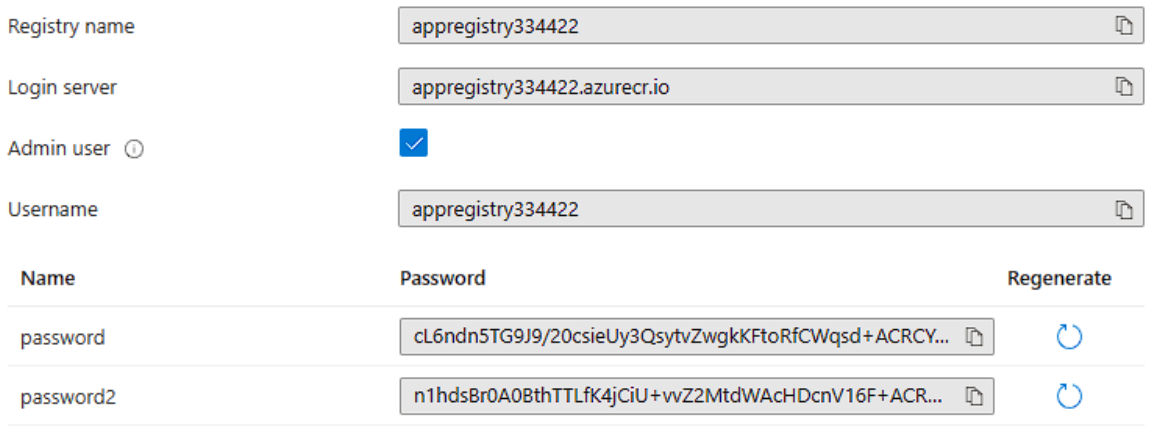
```
demousr@dockervm:~/publish$ sudo mkdir -p /etc/apt/keyrings
curl -sLS https://packages.microsoft.com/keys/microsoft.asc |
    gpg --dearmor |
    sudo tee /etc/apt/keyrings/microsoft.gpg > /dev/null
sudo chmod go+r /etc/apt/keyrings/microsoft.gpg
demousr@dockervm:~/publish$ AZ_DIST=$(lsb_release -cs)
echo "deb [arch=`dpkg --print-architecture` signed-by=/etc/apt/keyrings/microsoft.gpg] https://packages.microsoft.com/repos/azure-cli/ $AZ_DIST main" |
    sudo tee /etc/apt/sources.list.d/azure-cli.list
deb [arch=amd64 signed-by=/etc/apt/keyrings/microsoft.gpg] https://packages.microsoft.com/repos/azure-cli/ jammy main
demousr@dockervm:~/publish$
```

```
demousr@dockervm:~/publish$ sudo apt-get update
sudo apt-get install azure-cli
Hit:1 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://packages.microsoft.com/repos/azure-cli jammy InRelease
Hit:6 https://download.docker.com/linux/ubuntu jammy InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
azure-cli is already the newest version (2.55.0-1~jammy).
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
demousr@dockervm:~/publish$
```

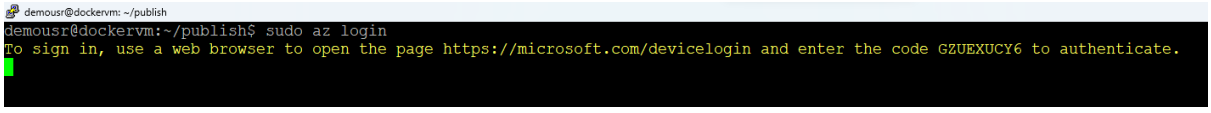
- Once the CLI has installed, now we are going to make use of access keys which we can get from Container Registry.
- Go back to Azure Portal, there on the left side of the screen, look for access keys.
- In the access keys you can see your registry name and log in server.

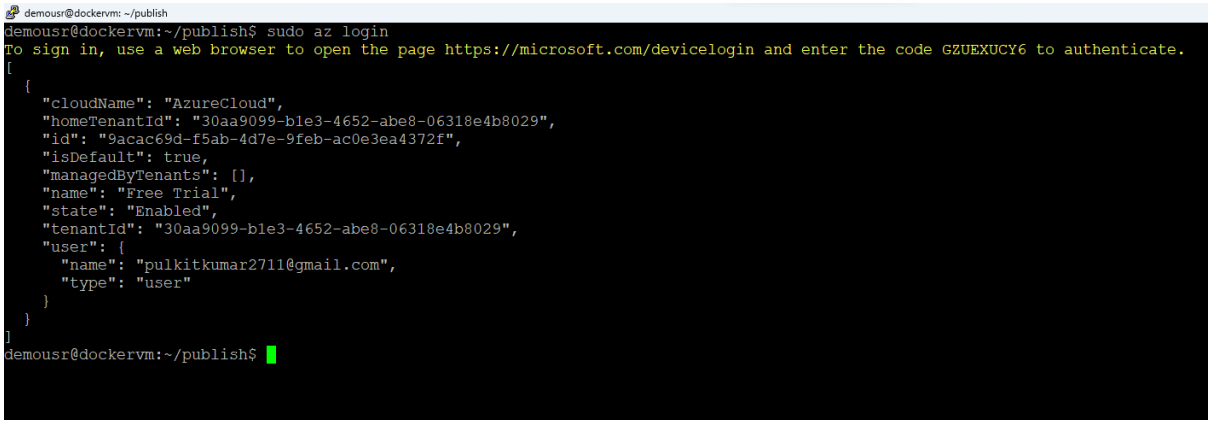


- Now you need to click on Admin user, it will give you access to the user's name and password.



- Then after go back to your virtual machine and type a command to log in.
- **sudo az login**
- Then it will give you a code and ask you to use that code on the Microsoft official website and then you will have to log in.
- For that open the website mentioned and then enter the code there, after that you need to log into your account.
- Then you will get this script written which means that you have successfully login.





- Now you need to write this code.
- **sudo az acr login --name --username –password**
- After the code has successfully executed then move forward to write another code and execute that.

- Now write this code and execute that.

- **sudo docker tag sqlapp appregistry10002313.azurecr.io/sqlapp**
- **sudo docker push appregistry10002313.azurecr.io/sqlapp**
- Remember to change the registry name with yours.

```
demousr@dockervm:~/publish$ sudo docker tag sqlapp appregistry334422.azurecr.io/sqlapp
demousr@dockervm:~/publish$ sudo docker push appregistry334422.azurecr.io/sqlapp
Using default tag: latest
The push refers to repository [appregistry334422.azurecr.io/sqlapp]
d240617eac2e: Pushed
f47e20252927: Pushed
4d2310e1c9fa: Pushed
48dac439e1e6: Pushed
12f0532081f1: Pushed
d47be6633206: Pushed
e6cd39d4cea4: Pushed
latest: digest: sha256:fdf60c2233214faad2613fd207f7fba7ec78436a5e6cca6c9949d356ca582496 size: 1788
demousr@dockervm:~/publish$
```

- Once every code is executed go to your Container Registry and then from the left the side go to repositories.
- There you will see that a repository is saved here.