

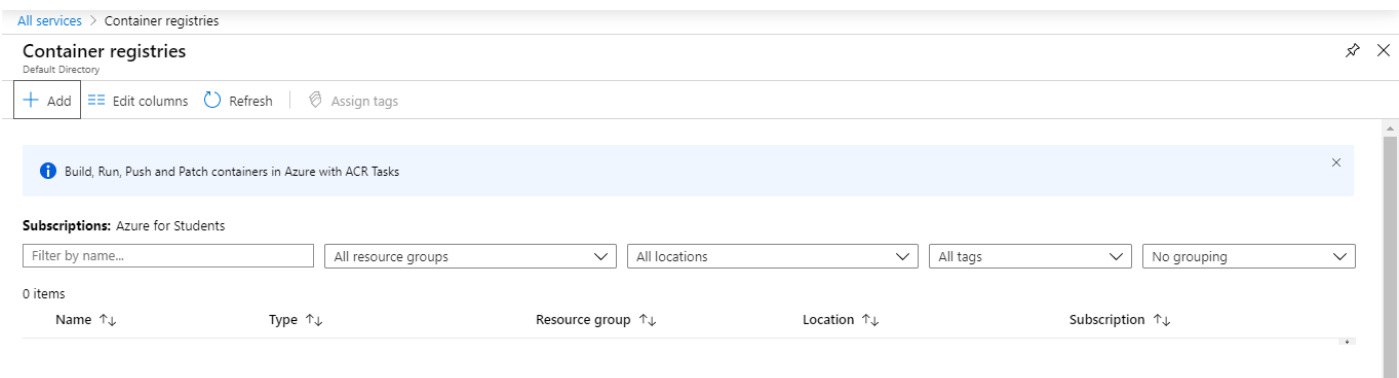
Azure 104 Module 5: Hands-on:2

Publishing and Automating Image Deployment to Azure Container Registry


Step 1: Deploy an Ubuntu virtual machine and install the Azure CLI and Docker.

```
newuser@vm1: ~  
  
sudo snap install microk8s --channel=1.18/beta --classic  
  
* Multipass 1.1 adds proxy support for developers behind enterprise  
  firewalls. Rapid prototyping for cloud operations just got easier.  
  
https://multipass.run/  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
newuser@vm1:~$
```

Step 2: Search for **Container registries** and click on it; and then, click on **Add**



Step 3: Enter details such as the resource group and the name of the registry and switch the Admin user to **Enable**. Then, click on **Next: Encryption**

 **Create container registry**

code updates, updates to a container's base image, or timers. [Learn more](#)

Project details

Subscription *
Azure for Students

Resource group *
intellipaat
[Create new](#)

Instance details

Registry name *
demoreg3121
.azurecr.io

Location *
(US) Central US


Admin user * ⓘ
☒ Enable ☐ Disable

SKU * ⓘ
Standard

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

Step 4: Here, enter the Encryption details, if any, and then click on **Review + create**

[All services](#) > [Container registries](#) > Create container registry

 **Create container registry**

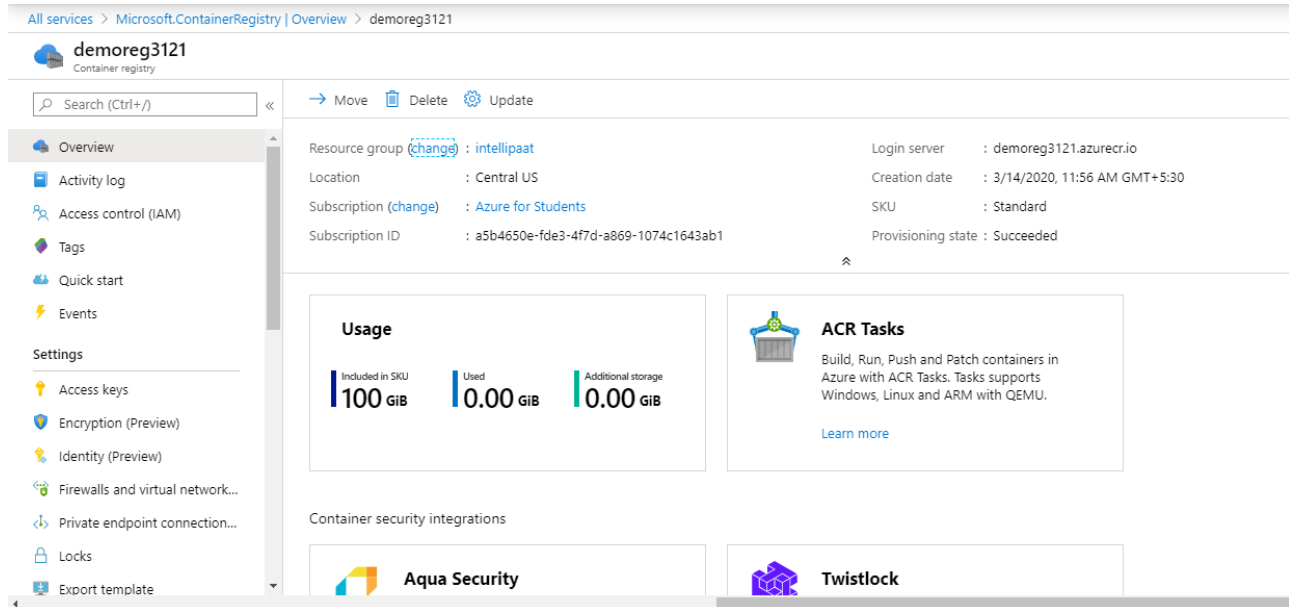
[Basics](#) * [Encryption](#) [Tags](#) [Review + create](#)

Azure Container Registry service encryption protects your data at rest. Azure Container Registry encrypts your images and other artifacts when they're pushed to your registry and automatically decrypts when you pull them. [Learn more](#)

Customer-Managed Key
☐ Enabled ☒ Disabled
Customer-Managed Key is only available for Premium SKU.

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

Step 5: Once the resource has been deployed, copy the name of the container registry

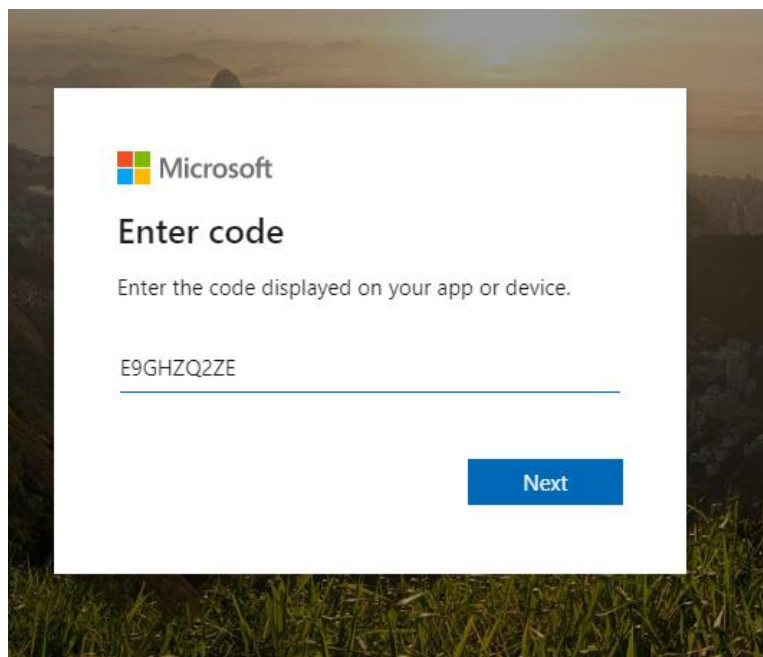


The screenshot shows the Azure Portal interface for a Container Registry resource named 'demoreg3121'. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Quick start, Events, and Settings. The main content area shows the Overview page with a search bar, action buttons (Move, Delete, Update), and a list of properties including Resource group (intellipaat), Location (Central US), Subscription (Azure for Students), and Subscription ID. A usage section shows 100 GiB included in the SKU, 0.00 GiB used, and 0.00 GiB additional storage. There are also sections for ACR Tasks and Container security integrations (Aqua Security, Twistlock).

Step 6: Go to the Ubuntu machine and type the command **az login** to log into Azure Portal. It will provide you with a code to authenticate

```
root@vml:/home/newuser# az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin
and enter the code E9GHZQ2ZE to authenticate.
^Croot@vml:/home/newuser#
```

Step 7: Enter the code to authenticate



Step 8: Connect to the Azure container registry with the command:

```
az acr login --name <name of the container registry>
```

```
root@vml:/home/newuser# az acr login --name demoreg3121
Login Succeeded
root@vml:/home/newuser#
```

Step 9: Pull an image 'hello-world' using the command **docker pull hello-world** and tag it using the docker tag command:

```
docker tag <image name><destination address of the image>
```

Now, follow these commands to push the image into the container registry and run it

```
> docker push demoreg3121.azurecr.io/hello-world:v1
> docker run demoreg3121.azurecr.io/hello-world:v1
```

```
root@vml: /home/newuser
root@vml:/home/newuser# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:f9dfddf63636d84ef479d645ab5885156ae030f611a56f3a7ac7f2fdd86d7e4e
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
root@vml:/home/newuser# docker tag hello-world demoreg3121.azurecr.io/hello-world:v1
root@vml:/home/newuser# docker push demoreg3121.azurecr.io/hello-world:v1
The push refers to repository [demoreg3121.azurecr.io/hello-world]
af0b15c8625b: Pushed
v1: digest: sha256:92c7f9c92844bbbb5d0a101b22f7c2a7949e40f8ea90c8b3bc396879d95e899a size: 524
root@vml:/home/newuser# docker run demoreg3121.azurecr.io/hello-world:v1

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
```

Step 10: Go to the Azure container registry in Azure Portal and click on **Repositories**. Refresh it to be able to see the repositories that have been pushed by you

