



**Cloud Development, 2021 fall.
Lab 5**

Report

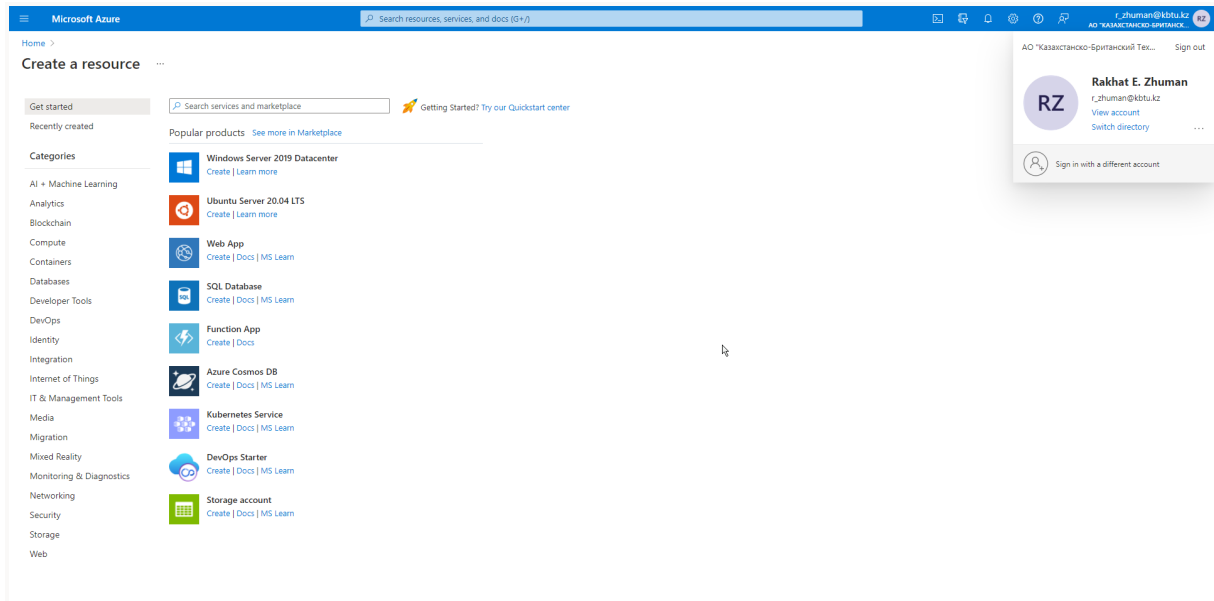
Deploying compute workloads by using images and containers

made by Zhuman Rakhat

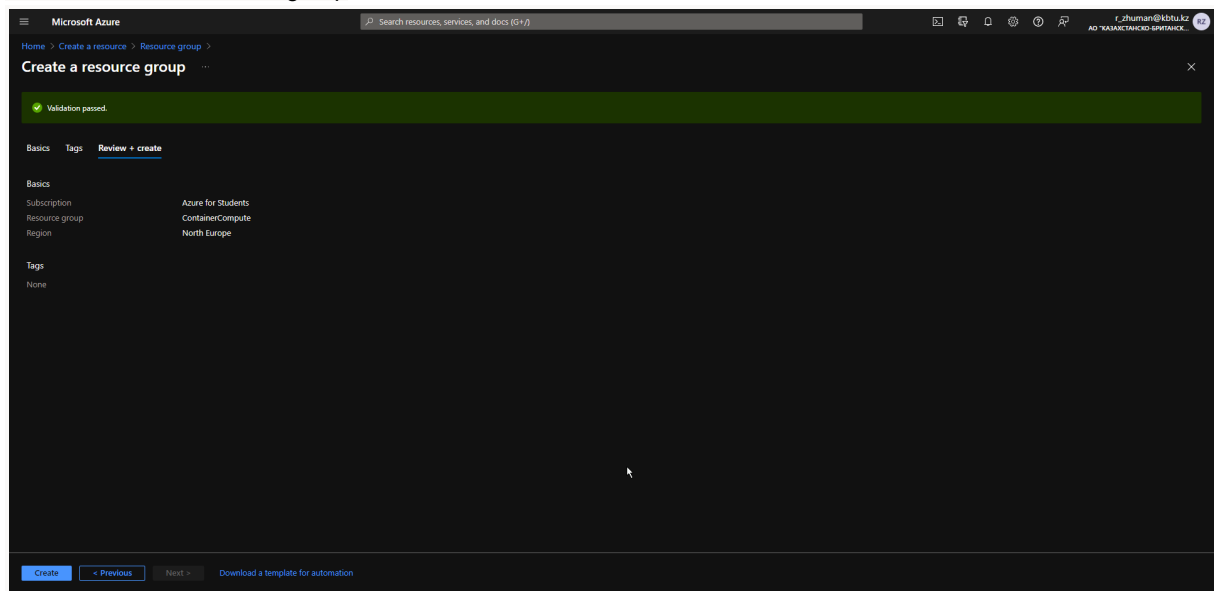
Almaty 2021

Exercise 1: Create a VM by using the Azure CLI

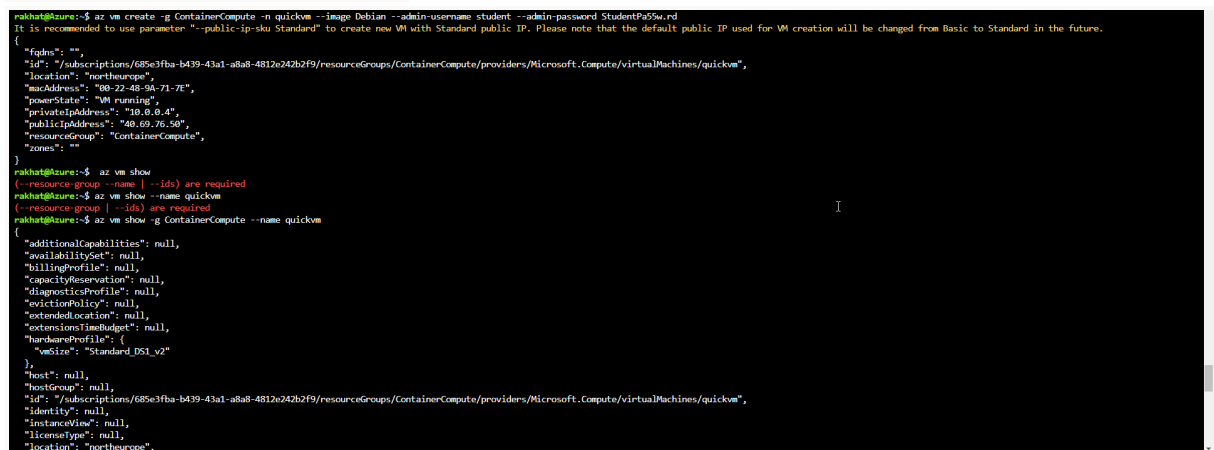
Task 1: Open the Azure portal



Task 2: Create a resource group



Task 4: Use the Azure CLI commands



```

rakhat@azure:~$ az vm list-ip-addresses --resource-group ContainerCompute --name quickvm
[
  {
    "virtualMachine": {
      "name": "quickvm",
      "network": {
        "privateIpAddresses": [
          "10.0.0.4"
        ],
        "publicIpAddresses": [
          {
            "id": "/subscriptions/685e3fba-b39-43a1-a8a8-4812e242b2f9/resourceGroups/ContainerCompute/providers/Microsoft.Network/publicIpAddresses/quickvmPublicIP",
            "ipAddress": "40.69.76.50",
            "ipAllocationMethod": "Dynamic",
            "name": "quickvmPublicIP",
            "resourceGroup": "ContainerCompute"
          }
        ]
      },
      "resourceGroup": "ContainerCompute"
    }
  }
]
rakhat@azure:~$

rakhat@azure:~$ az vm list-ip-addresses --resource-group ContainerCompute --name quickvm --query '[].(ip.virtualMachine.network.publicIpAddresses[0].ipAddress)' --output tsv
40.69.76.50
rakhat@azure:~$ echo tsv
tsv
rakhat@azure:~$ ls
cloudrive
rakhat@azure:~$ ipAddress=$(az vm list-ip-addresses --resource-group ContainerCompute --name quickvm --query '[].(ip.virtualMachine.network.publicIpAddresses[0].ipAddress)' --output tsv)
rakhat@azure:~$ echo $ipAddress
40.69.76.50
rakhat@azure:~$ ssh student@$ipAddress
The authenticity of host '40.69.76.50 (40.69.76.50)' can't be established.
ECDSA key fingerprint is SHA256:pbSoraWQ76s/hyexdh2HfJ8N8DXz36MyrUE66o.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.69.76.50' (ECDSA) to the list of known hosts.
student@40.69.76.50's password:
Permission denied, please try again.
student@40.69.76.50's password:
Linux quickvm 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
student@quickvm:~$ uname -a
Linux quickvm 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64 GNU/Linux
student@quickvm:~$ exit
logout
Connection to 40.69.76.50 closed.
rakhat@azure:~$

```

Exercise 2: Create a Docker container image and deploy it to Container Registry

Task 1: Open the Cloud Shell and editor

```

rakhat@azure:~$ cd cloudrive
rakhat@azure:~/cloudrive$ ls
rakhat@azure:~/cloudrive$ mkdir ipcheck
rakhat@azure:~/cloudrive$ cd ipcheck/
rakhat@azure:~/cloudrive/ipcheck$ dotnet new console --output . --name ipcheck

Welcome to .NET Core 3.1!
-----
SDK Version: 3.1.403

Telemetry
The .NET Core tools collect usage data in order to help us improve your experience. The data is anonymous. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT
environment variable to '1' or 'true' using your favorite shell.

Read more about .NET Core CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Find out what's new: https://aka.ms/dotnet-whats-new
Learn about the installed HTTPS developer cert: https://aka.ms/aspnet-core-https
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli-docs
Write your first app: https://aka.ms/first-net-core-app
-----
Getting ready...
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on /ipcheck.csproj...
  Determining projects to restore...
  Restored /usr/cuser/cloudrive/ipcheck/ipcheck.csproj (in 166 ms).

Restore succeeded.

rakhat@azure:~/cloudrive/ipcheck$ touch Dockerfile
rakhat@azure:~/cloudrive/ipcheck$ ls
Dockerfile  ipcheck.csproj  Program.cs
rakhat@azure:~/cloudrive/ipcheck$

```

Task 2: Create and test a .NET application

```

Microsoft Azure
Search resources, services, and docs (GvF)

Bash
FILES
  obj
  Dockerfile
  ipcheck.csproj
  Program.cs
1  using System;
2
3  namespace ipcheck
4  {
5      public class Program
6      {
7          public static void Main(string[] args)
8          {
9              // Check if network is available
10             if (System.Net.NetworkInformation.NetworkInterface.GetIsNetworkAvailable())
11             {
12                 System.Console.WriteLine("Current IP Addresses:");
13                 // Get host entry for current hostname
14                 string hostname = System.Net.Dns.GetHostName();
15                 System.Net.IPHostEntry host = System.Net.Dns.GetHostEntry(hostname);
16                 // Iterate over each IP address and render their values
17                 foreach (System.Net.IPAddress address in host.AddressList)
18                 {
19                     System.Console.WriteLine($"{hostname}\t{address}");
20                 }
21             }
22             else
23             {
24                 System.Console.WriteLine("No Network Connection");
25             }
26         }
27     }
28 }

Running 'dotnet restore' on /ipcheck.csproj...
  Determining projects to restore...
  Restored /usr/cuser/cloudrive/ipcheck/ipcheck.csproj (in 166 ms).

Restore succeeded.

rakhat@azure:~/cloudrive/ipcheck$ touch Dockerfile
rakhat@azure:~/cloudrive/ipcheck$ ls
Dockerfile  ipcheck.csproj  Program.cs
rakhat@azure:~/cloudrive/ipcheck$ code .
rakhat@azure:~/cloudrive/ipcheck$ code .
rakhat@azure:~/cloudrive/ipcheck$ dotnet run
Current IP Addresses:
10.244.12.227
rakhat@azure:~/cloudrive/ipcheck$

```

```
Microsoft Azure
Search resources, services, and docs (Ctrl)
i_zhuman@kbtu.kz
AD "KAZAKHSTANICHO SPITANICE..."

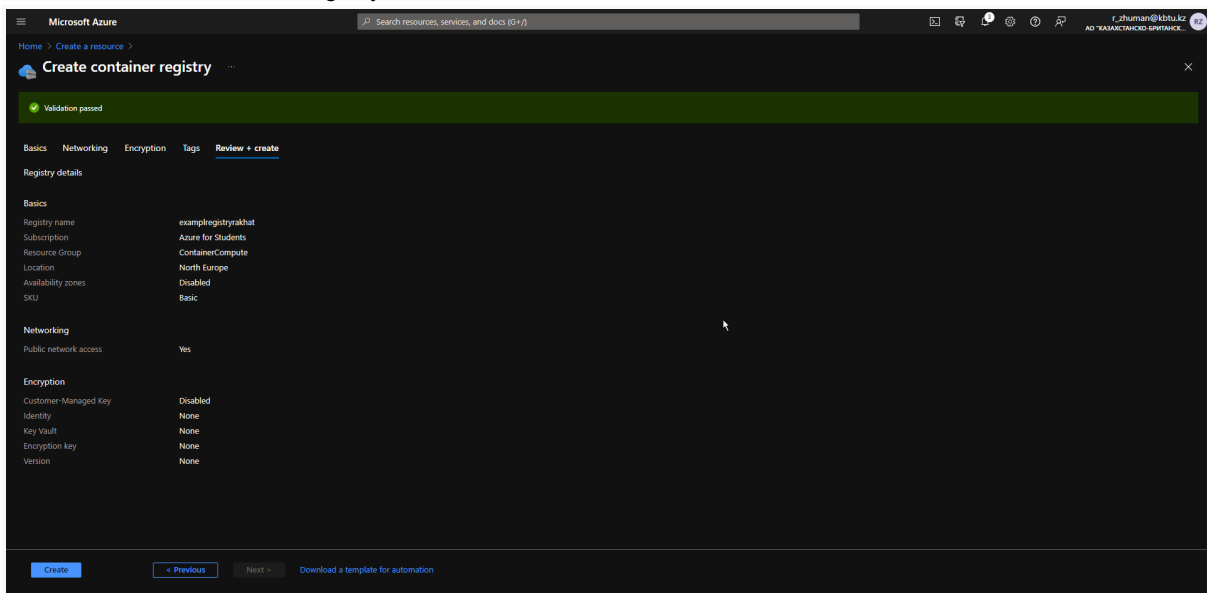
Bash
FILES
+ bin
+ obj
+ Dockerfile
+ ipcheck.csproj
+ Program.cs

1 # Start using the .NET Core 3.1 SDK container image
2 FROM mcr.microsoft.com/dotnet/sdk:3.1-alpine AS build
3 # Change current working directory
4 WORKDIR /app
5 # Copy existing files from host machine
6 COPY . ./
7 # Publish application to the "out" folder
8 RUN dotnet publish --configuration Release --output out
9 # Start container by running application DLL
10 ENTRYPOINT ["dotnet", "out/ipcheck.dll"]
11

Running 'dotnet restore' on /ipcheck.csproj...
Determining projects to restore...
Restored /usr/cuser/clooudrive/ipcheck/ipcheck.csproj (in 166 ms).
Restore succeeded.

rakhat@Azure:~/cloudrive/ipcheck$ touch Dockerfile
rakhat@Azure:~/cloudrive/ipcheck$ ls
Dockerfile  ipcheck.csproj  Program.cs
rakhat@Azure:~/cloudrive/ipcheck$ code .
rakhat@Azure:~/cloudrive/ipcheck$ dotnet run
Current IP Address:
10.244.12.227
rakhat@Azure:~/cloudrive/ipcheck$
```

Task 3: Create a Container Registry resource



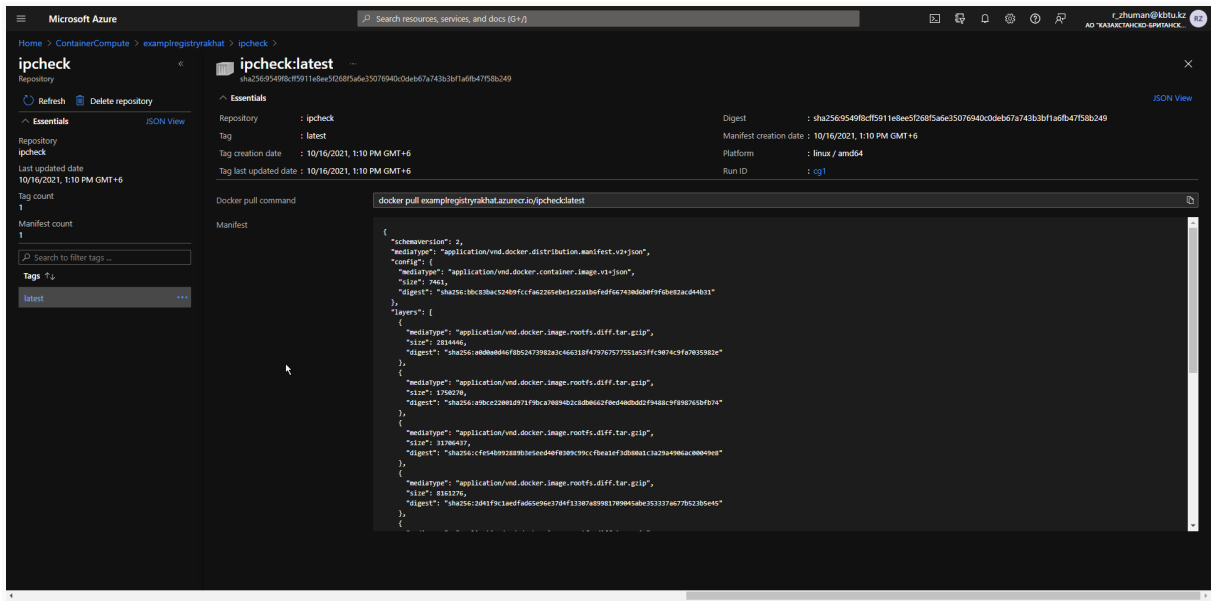
Task 4: Open Azure Cloud Shell and store Container Registry metadata

```
rakhat@Azure:~/cloudrive/ipcheck$ az acr list --query "max_by([], &creationDate).name" --output tsv
exampleregistryrakhat
rakhat@Azure:~/cloudrive/ipcheck$ az acr list --query "max_by([], &creationDate).name" --output tsv
exampleregistryrakhat
rakhat@Azure:~/cloudrive/ipcheck$ echo $acrName
exampleregistryrakhat
rakhat@Azure:~/cloudrive/ipcheck$
```

Task 5: Deploy a Docker container image to Container Registry

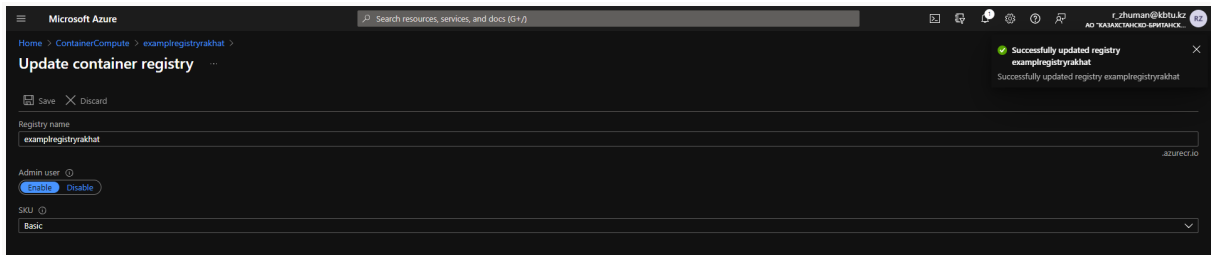
```
rakhat@Azure:~/cloudrive/ipcheck$ dir
bin Dockerfile ipcheck.csproj obj Program.cs
rakhat@Azure:~/cloudrive/ipcheck$ az acr build --registry $acrName --image ipcheck:latest .
Packing source code into tar to upload...
Uploading archived source code from '/tmp/build_archive_9c3f0aa346674baeb4468e848c999429.tar.gz'...
Sending context (99,752 KiB) to registry: exampleregistryrakhat...
Queued a build with ID: cal
Waiting for an agent...
2021/10/16 06:47:46 Downloading source code...
2021/10/16 06:47:47 Finished downloading source code
2021/10/16 06:47:47 Using acb_vol_5217f77e-f635-4743-be78-8b16484d189 as the home volume
2021/10/16 06:47:47 Setting up Docker configuration...
2021/10/16 06:47:48 Successfully set up Docker configuration
2021/10/16 06:47:48 Logging in to registry: exampleregistryrakhat.azurecr.io
2021/10/16 06:47:49 Successfully logged into exampleregistryrakhat.azurecr.io
2021/10/16 06:47:49 Executing step ID: build. Timeout(sec): 28800, Working directory: '', Network: ''
2021/10/16 06:47:49 Scanning for dependencies...
2021/10/16 06:47:49 Successfully scanned dependencies
2021/10/16 06:47:49 Launching container with name: build
Sending build context to Docker daemon 326.748
Step 1/5 : FROM mcr.microsoft.com/dotnet/sdk:3.1-alpine AS build
3.1-alpine: Pulling from dotnet/sdk
a0d8d4d6f8b: Already exists
a0c222081b9: Pulling fs layer
cfe40922889: Pulling fs layer
2d41f9c1aef: Pulling fs layer
ec4c85a0b7c6: Pulling fs layer
c688a6f81f02: Pulling fs layer
39690916a28: Pulling fs layer
ec4c85a0b7c6: Waiting
```

Task 6: Validate your container image in Container Registry

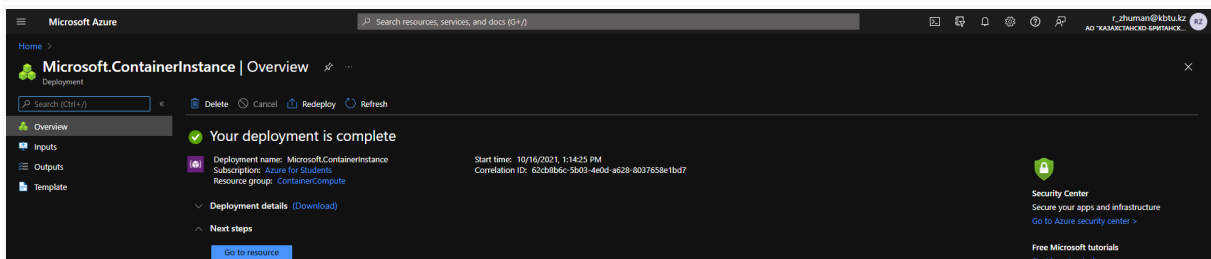
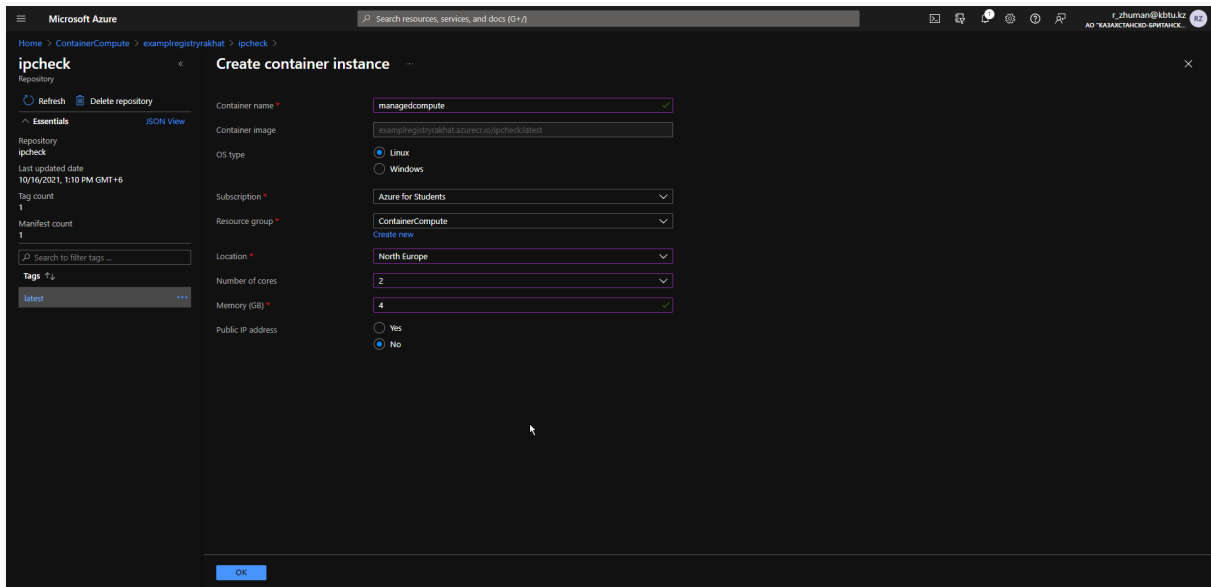


Exercise 3: Deploy an Azure container instance

Task 1: Enable the admin user in Container Registry



Task 2: Automatically deploy a container image to an Azure container instance



Task 3: Manually deploy a container image to Container Instance

Microsoft Azure

Home > Create a resource >

Create container instance

Azure Container Instances (ACI) allows you to quickly and easily run containers on Azure without managing servers or having to learn new tools. ACI offers pre-second billing to minimize the cost of running containers on the cloud.
[Learn more about Azure Container Instances](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription: Azure for Students

Resource group: ContainerCompute

Container details

Container name: manualcompute

Region: (Europe) North Europe

Image source: ☒ Quickstart images ☒ Azure Container Registry ☐ Docker Hub or other registry

Registry: exampleregistryrakhat

Image: ipcheck

Image tag: latest

OS type: Linux

Size: 1 vcpu, 1.5 GiB memory, 0 gpus

[Change size](#)

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

Microsoft Azure

Home > Microsoft.ContainerInstances-20211016131548 | Overview

Deployment

[Delete](#) [Cancel](#) [Redeploy](#) [Refresh](#)

[Go to resource](#) [Pin to dashboard](#)

Deployment succeeded

Deployment "Microsoft.ContainerInstances-20211016131548" to resource group "ContainerCompute" was successful.

Your deployment is complete

Deployment name: Microsoft.ContainerInstances-20211016131548 Start time: 10/16/2021, 1:16:57 PM
Subscription: Azure for Students Correlation ID: 661de182-123d-470e-b2cf-61bcd6077f7ee
Resource group: ContainerCompute

[Deployment details \(Download\)](#)

[Go to resource](#)

Security Center

Secure your apps and infrastructure
[Go to Azure security center >](#)

Free Microsoft tutorials

[Start learning today >](#)

Task 4: Validate that the container instance ran successfully

Microsoft Azure

Home > Microsoft.ContainerInstances-20211016131548 > manualcompute

manualcompute | Containers

Container instances

[Refresh](#)

1 container

Name	Image	State	Previous state	Start time	Restart count
manualcompute	exampleregistryrakhat.azurecr.io/ipcheckstat...	Terminated	-	2021-10-16T07:17:33.182Z	0

Events **Properties** **Logs** **Connect**

Display time zone: ☒ Local time ☐ UTC

Name	Type	First timestamp	Last timestamp	Message	Count
Killing	Normal	10/16/2021, 01:17 PM GMT+6	10/16/2021, 01:17 PM GMT+6	Killing container with id b69c1c7ba04b126f...	1
Started	Normal	10/16/2021, 01:17 PM GMT+6	10/16/2021, 01:17 PM GMT+6	Started container	1
Pulled	Normal	10/16/2021, 01:17 PM GMT+6	10/16/2021, 01:17 PM GMT+6	Successfully pulled image "exampleregistryr...	1
Pulling	Normal	10/16/2021, 01:17 PM GMT+6	10/16/2021, 01:17 PM GMT+6	pulling image "exampleregistryrakhat.azurec...	1

Microsoft Azure

Home > Microsoft.ContainerInstances-20211016131548 > manualcompute

manualcompute | Containers

Container instances

[Refresh](#)

1 container

Name	Image	State	Previous state	Start time	Restart count
manualcompute	exampleregistryrakhat.azurecr.io/ipcheckstat...	Terminated	-	2021-10-16T07:17:33.182Z	0

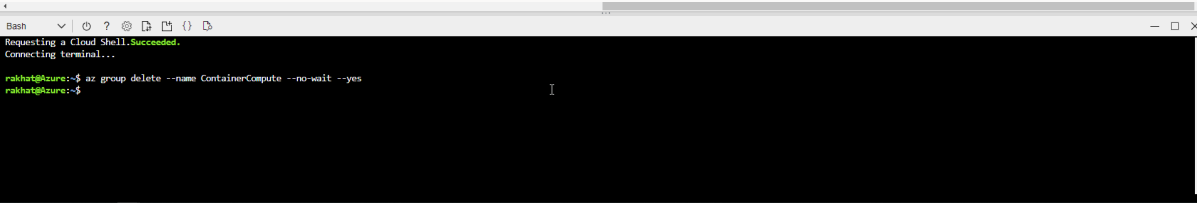
Events **Properties** **Logs** **Connect**

Current IP Addresses:

172.17.0.1

Exercise 4: Clean up your subscription

Task 2: Delete resource groups



```
4
Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

rakhot@Azure:~$ az group delete --name ContainerCompute --no-wait --yes
rakhot@Azure:~$
```