



中原大學 雲端計算平台實務

11/19-作業報告

Administer containers in Azure

資訊四乙 10727211 林彥輝

授課教師：鍾武君 教授

中華民國一一〇年十一月

1. Model Intro

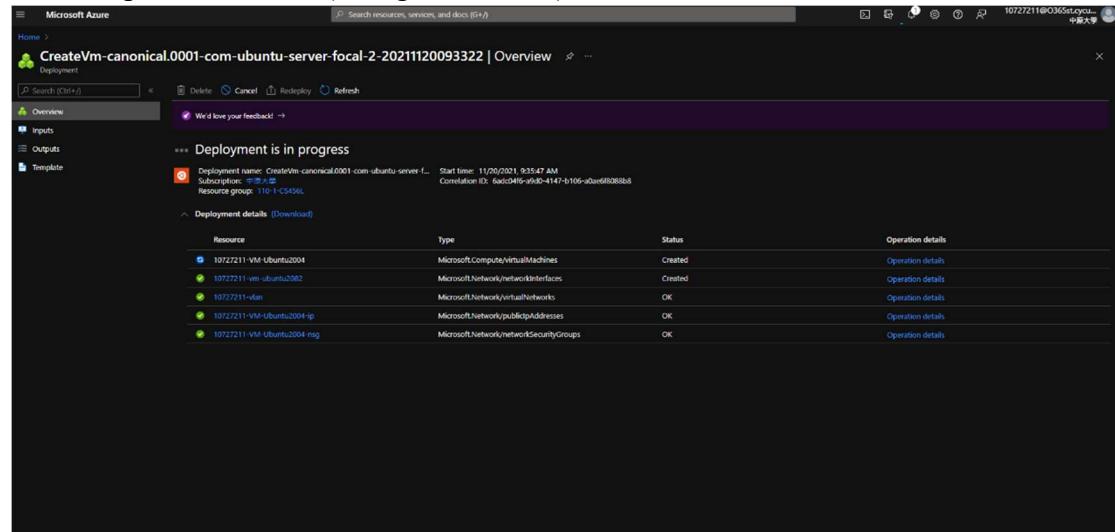
Administer containers in Azure

<https://docs.microsoft.com/en-us/learn/parts/administer-containers-in-azure/>

2. Summary Homework Assignment

Model 2:Build a containerized web application with Docker

1. Set up Environment (Using Azure VM)



2. Install Docker and check Status

```
azuser@10727211-VM-Ubuntu2004: ~
Setting up runc (1.0.1~ubuntu2~20.04.1) ...
Setting up libsystemd0 (2.48-1ubuntu19.5~2892) ...
Setting up libidn11-openssl (1.33-2.1ubuntu0.2) ...
Setting up bridge-utils (1.6.2~ubuntu1) ...
Setting up pigz (2.4-1) ...
Setting up containerd (1.5.5~ubuntu3~20.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker.io (20.10.7~ubuntu5~20.04.2) ...
Adding group 'docker' (GID 123) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up dnsmasq-base (2.80-1.1ubuntu1.4) ...
Setting up libsystemd-journal (2.48-1.1ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Processing triggers for systemd (245.4~ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16~ubuntu2.1) ...
Processing triggers for libsystemd-dbus (2.48-1.1ubuntu1) ...
azuser@10727211-VM-Ubuntu2004: ~$ service docker status
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2021-11-20 01:39:42 UTC; 14s ago
TriggeredBy: ● docker.socket
  Main PID: 2482 (dockerd)
    Tasks: 8
   Memory: 51.0M
      CGroup: /system.slice/docker.service
              └─2482 /usr/bin/dockerd -H fd:// - -containerd=/run/containerd/containerd.sock

Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.142882996Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.143040762Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.144343967Z" level=info msg="Loading containers: start."
Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.378797807Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.1/16, IPv4 link local address 172.17.0.1/24. You must not change this unless you know what you are doing."
Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.757119511Z" level=warning msg="Not enough memory available for overlay2, this may cause problems with large containers."
Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.758052117Z" level=info msg="Docker daemon" commit="20.10.7~ubuntus5~20.04.2" graphDriver="overlay2"
Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.758333917Z" level=info msg="Daemon has completed initialization"
Nov 20 01:39:42 10727211-VM-Ubuntu2004 system[1]: Started Docker Application Container Engine.
Nov 20 01:39:42 10727211-VM-Ubuntu2004 dockerd[2482]: time="2021-11-20T01:39:42.902006619Z" level=info msg="API listen on /run/docker.sock"
Lines 1-21/21 (END)
```

3. After docker run app, setting NSG port-forward (8080 port)

The screenshot shows the Azure portal interface for managing a virtual machine named '10727211-VM-Ubuntu2004'. On the left, the 'Networking' section is selected under 'Settings'. In the center, the 'Inbound port rules' tab is active, showing a table of rules. A new rule is being added via a modal dialog titled 'Add inbound security rule'. The 'Destination port ranges' field is set to '8080' and is highlighted with a red box. The 'Protocol' dropdown shows 'TCP' selected. The 'Action' dropdown shows 'Allow' selected.

4. open url in browser : <VM_ip>:8080

The screenshot shows a browser window with the URL 'http://20.210.124.84:8080'. The page title is 'Home page - aspnetapp'. The content of the page is the standard '.NET Welcome' page. Below the welcome message, there is a table of system information:

.NET version	.NET 6.0.0-rtm.21522.10
Operating system	Linux 5.11.0-1021-azure #22~20.04.1-Ubuntu SMP Fri Oct 29 01:11:25 UTC 2021
Processor architecture	X64
CPU cores	1
Containerized	true
Memory, total available GC memory	914.64 MiB
cgroup memory usage	41.93 MiB
cgroup memory limit	8589934592.00 GiB
Host name	04e9cfb2d4e2
Server IP address	172.17.0.2

At the bottom of the browser window, the taskbar shows various icons for system monitoring and connectivity.

1. Create Container Registry

The screenshot shows the Microsoft Azure portal interface for a Container Registry named "10727211ContainerRegistry". The left sidebar has a tree view with "Overview" selected. The main content area displays the "Essentials" section with details such as Resource group (110-1-CS456), Location (Southeast Asia), Creation date (11/20/2021, 10:07 AM GMT +8), SKU (Standard), and Provisioning state (Succeeded). It also shows "Usage" (100 GB included in SKU, 0.00 GB used, 0.00 GB additional storage) and "ACR Tasks" (Build, Run, Push and Patch containers in Azure with ACR tasks. Tasks supports Windows, Linux and ARM with QEMU). There is a "Container security integrations" section with "Azure Security Center" listed.

2. Push Image (cmd)

```
azureuser@10727211-VM-Ubuntu2004:~/mslearn-hotel-reservation-system/src$ sudo docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
reservationsystem  latest   b7e2ea087d4d  8 minutes ago  1.87GB
mcr.microsoft.com/dotnet/core/samples  aspnetapp  1926dcade061  2 days ago   216MB
mcr.microsoft.com/dotnet/core/sdk    2.2       2357b6798b9d  23 months ago  1.74GB
azureuser@10727211-VM-Ubuntu2004:~/mslearn-hotel-reservation-system/src$ docker tag reservationsystem:latest 10727211ContainerRegistry.azurecr.io/reservationsystem:latest
azureuser@10727211-VM-Ubuntu2004:~/mslearn-hotel-reservation-system/src$ docker login 10727211ContainerRegistry.azurecr.io
Username: 10727211ContainerRegistry
Password:
WARNING! Your password will be stored unencrypted in /home/azureuser/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
azureuser@10727211-VM-Ubuntu2004:~/mslearn-hotel-reservation-system/src$ docker push 10727211ContainerRegistry.azurecr.io/reservationsystem:latest
The push refers to repository [10727211ContainerRegistry.azurecr.io/reservationsystem]
54507bde05dd: Pushed
3bda4e430e77: Pushed
2e5c9c761905: Pushed
b7e44761466e: Pushed
6cb0513e99f6: Pushed
33d126946484: Pushed
5aae01c967c: Pushed
bb60fb1e0b76: Pushed
bbf1566001c0: Pushed
99964051b3c: Pushed
be1e19d7c3a: Pushed
1f59ab2e7e06: Pushed
0caf7f405c0: Pushed
eb9bae013034: Pushed
latest: digest: sha256:a62154ebd920a50066f51a0644018a220c002fd979289196eb4f942df31fe135 size: 3263
azureuser@10727211-VM-Ubuntu2004:~/mslearn-hotel-reservation-system/src$
```

3. Push Image (Web)

The screenshot shows the Microsoft Azure portal interface for the same Container Registry. The left sidebar has a tree view with "Repositories" selected. The main content area displays the "reservationsystem" repository under the "Repository" section. It shows 1 tag (latest) and 1 manifest. The "Tags" section lists "latest".

4. Create Container Instance

The screenshot shows the Microsoft Azure portal interface for creating a Container Instance. The main title bar says "Microsoft Azure" and "10727211@O365ct.you... 中文大". Below it, the URL is "Home > Microsoft.ContainerInstances-20211120101449 > 10727211-container-instance". The left sidebar has sections for Overview, Activity log, Access control (IAM), Tags, Settings, Containers, Identity, Properties, Locks, Monitoring, Metrics, Alerts, Automation, Tasks (preview), and Export template. The "Overview" tab is selected. The "Essentials" panel on the right shows the following details:

- Resource group (Move) : 110-1-C54561
- Status : Running
- Location : Southeast Asia
- Subscription (Move) : 中原大學
- Subscription ID : ba5f5e30-1488-49db-bae0-9026dacc1b0
- Tags (Edit) : 1072/211 :
- OS type : Linux
- IP address (Public) : 20.212.89.1
- FQDN : dns10727211.southeastasia.azurecontainer.io
- Container count : 1

Below this, there are two charts: "CPU" and "Memory". The CPU chart shows usage from 0% to 100% over time from 9:00 AM to 10:00 AM UTC+08:00. The Memory chart shows usage from 0B to 1000B over the same period.

5. Result1 (using FQDN navigate)

The screenshot shows a browser window with the URL "dns10727211.southeastasia.azurecontainer.io/api/reservations/1". The response body is a JSON object representing a reservation:

```
{"reservationID": "1", "customerID": "343706200", "hotelID": "Hotel_1671073415", "checkin": "2021-11-30T02:19:58", "checkout": "2022-01-03T17:07:55", "numberofGuests": 1, "reservationComments": "stsvxg1xxkxe12/z30182GKsknkv1j6louw2fxzNhtkf/h0q1jXm1klongecURc9702pOH0sCt1xJ3pR1ktxz1fkz0WvsoYnh1+kucb2CSP01pb7Vt+sxs1COKR6J5w081+1sgKvJS+K63S+V1Gn9/tDzccs0axqq1mlmuh0wG0/PkLirY0tp454rR1lQ1pVknxtXHFc35ew9u42qjh/glh17A/s5wz2lqjvz520p0pg1lByqc+vc0a1r/b/cRM0FV5j0hbyvP0elhGrnts1tP17Yb0454sC009ah0Bgn+26aCscQgev0TC59ruIUT13+SpNu9a4v1u70h3lipVaBBjCUf8+QdkybIr0hQ013cTrxLL4s1fHbX850mER28r1u7HCGrnsk+4nUpPXV8HBCL1d2kr+90o2mHr/lzcCE1lMAJ1ZM2AZltHu82Yw+j1Dy+LIZW0xpZK2+Je8dH2H8n/ZAK3s50gXn+O+p/4V5zyzeme0BNQhLvc=s"}
```

6. Result2 (using IP navigate)

The screenshot shows a browser window with the URL "20.212.89.71/api/reservations/1". The response body is a JSON object representing a reservation:

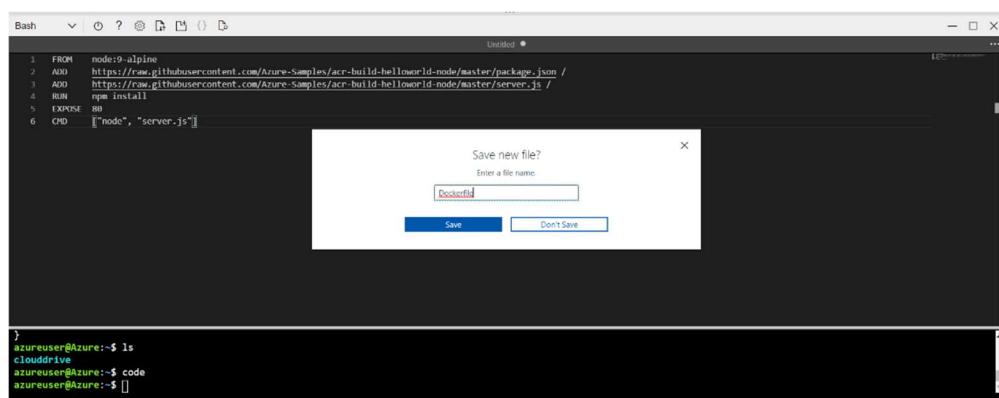
```
{"reservationID": "1", "customerID": "251752993", "hotelID": "Hotel_1343311918", "checkin": "2021-11-30T02:19:46", "checkout": "2022-01-26T02:47:01", "numberofGuests": 2, "reservationComments": "w1lp0xwQ0ve1lUAYu1lyKkp1d0XGPM6RFh171st1lypjl2kwawWl0b0wAD794DVt+0DX/8e3PF3EP2hu59n2xC8e+K1GJQcV20bjf/Sb3j7d5j0P3b/1491bh2Pckm0vep59QYy117s5x259hers/AwUszT8Rq+sigp0vQ0Lbpi450Qcr2uM8biqkvnX75nCh0PhBhrnyg2j3jKxP1fHrvnQbdk1Vm13jFC00BF0sH67Cafeu5FfrY2x2dYubPh1s6G4v73Dk4q7fQ7GSu1adov1j1711x4fQL18o1y2zdfBb9+37119E31fVv/k+277fm021zqN0K5B1012JjN24q1148Or1Tm13lk546Lw0X71LmWv8Y03asxe1BTHD13p1St07r/GV042Jv8j39C1xK64zKaksYSRT2Wlzuq78enGT91V22neC009K98ElzJ2i0SKSEv92n5BPrkGutfYNUtYfAIYQu+6uFtp+KQdQ115yzPCPs1kGUQ2okKyTDcvf0h7QHr-FTW3s550s3Cew9rdw1VwJZ988HM2blwM93jle/w/QrycWty0d1vObcH097MK2zrML7thuy7Q1z4s="}
```

Model 3:Build and store container images with Azure

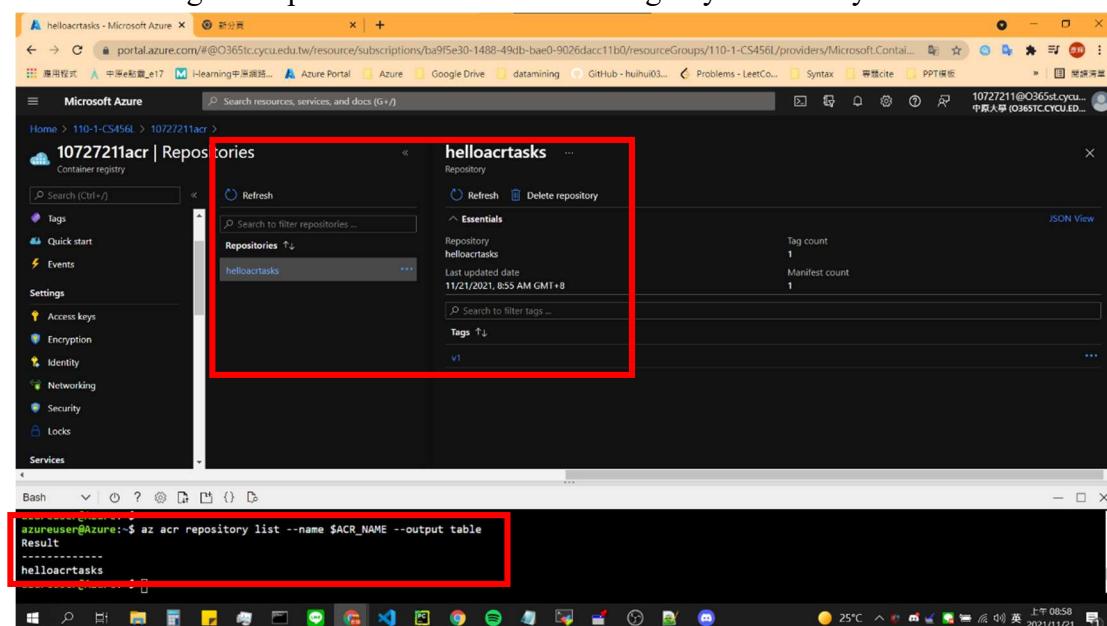
Container Registry

1. Create Azure Container Registry (using Azure Cloud Shell)

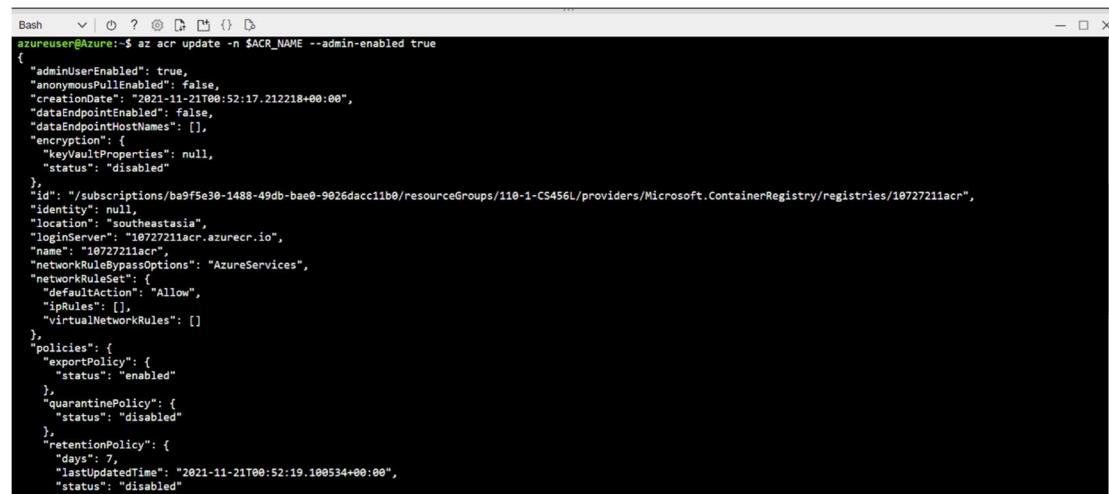
2. Write Dockerfile



3. Build Image then push to Azure Container Registry && Verify

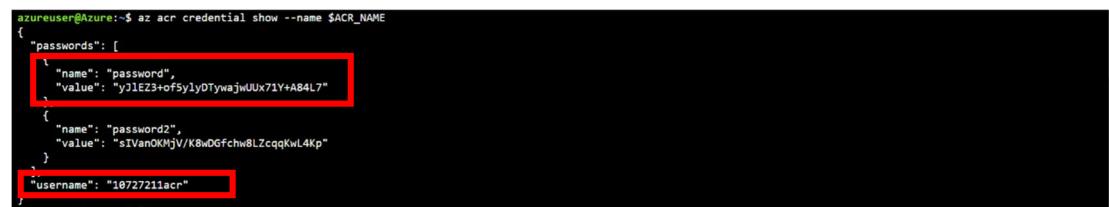


4. Enable the registry admin account (using Azure Cloud Shell)



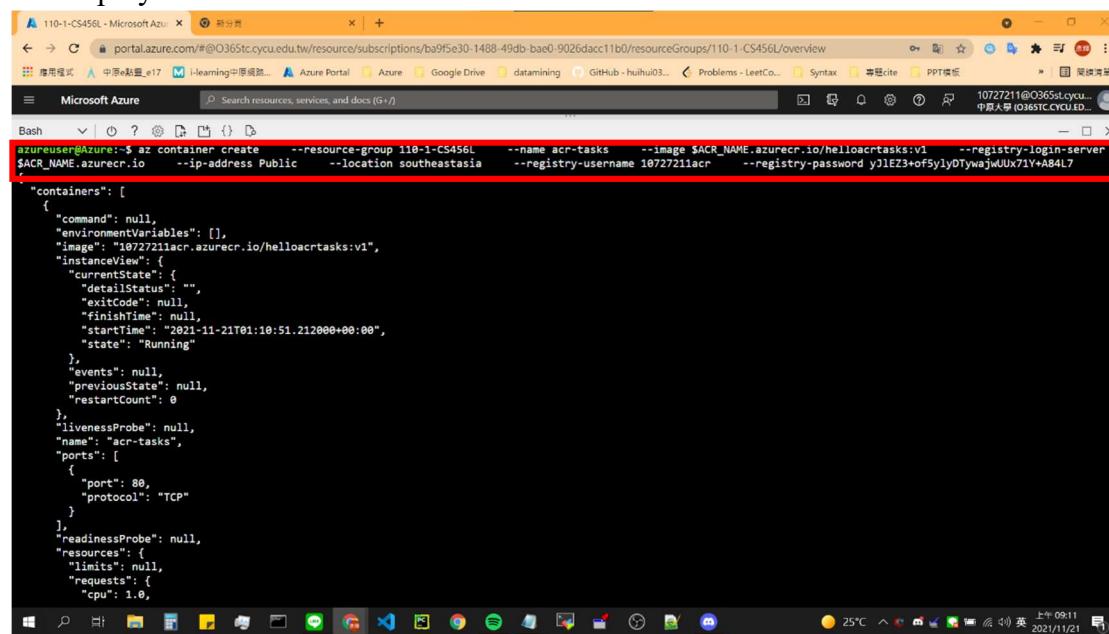
```
Bash azureuser@Azure:~$ az acr update -n $ACR_NAME --admin-enabled true
{
  "adminUserEnabled": true,
  "anonymousPullEnabled": false,
  "creationDate": "2021-11-21T00:52:17.212218+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
  "id": "/subscriptions/ba9f5e30-1488-49db-bae0-9026dacc11b0/resourceGroups/110-1-CS456L/providers/Microsoft.ContainerRegistry/registries/10727211acr",
  "identity": null,
  "location": "southeastasia",
  "loginServer": "10727211acr.azurecr.io",
  "name": "10727211acr",
  "networkRuleBypassOptions": "AzureServices",
  "networkRuleSet": {
    "defaultAction": "allow",
    "ipRules": [],
    "virtualNetworkRules": []
  },
  "policies": {
    "exportPolicy": {
      "status": "enabled"
    },
    "quarantinePolicy": {
      "status": "disabled"
    },
    "retentionPolicy": {
      "days": 7,
      "lastUpdatedTime": "2021-11-21T00:52:19.100534+00:00",
      "status": "disabled"
    }
  }
}
```

5. show credential (using Azure Cloud Shell)



```
azureuser@Azure:~$ az acr credential show --name $ACR_NAME
{
  "passwords": [
    {
      "name": "password",
      "value": "yJlEZ3+of5ylyDTwajwUx71Y+A84L7"
    },
    {
      "name": "password2",
      "value": "SIVanOKMjV/K8wDGfchw8LZcqKwL4Kp"
    }
  ],
  "username": "10727211acr"
}
```

6. Deploy a container



Azure portal screenshot showing the creation of a container named 'acr-tasks' in resource group '110-1-CS456L'. The command run in the terminal is:

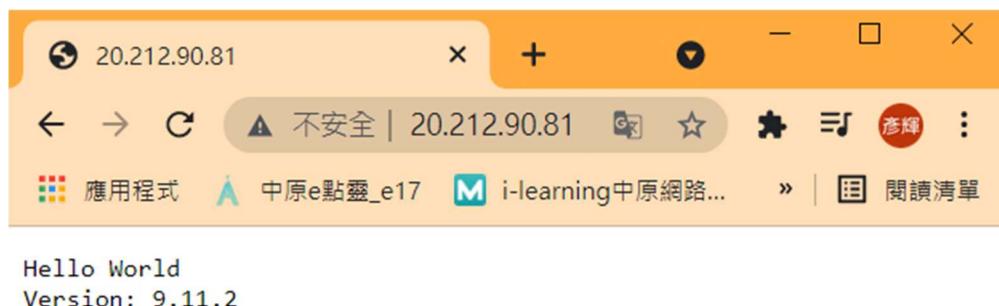
```
az container create --resource-group 110-1-CS456L --name acr-tasks --image $ACR_NAME.azurecr.io/helloacrtasks:v1 --registry-login-server $ACR_NAME.azurecr.io --ip-address Public --location southeastasia --registry-username 10727211acr --registry-password yJlEZ3+of5ylyDTwajwUx71Y+A84L7
```

The terminal output shows the container has been successfully deployed with the name 'acr-tasks'.

7. Get IP Address

```
azureuser@Azure:~$ az container show --resource-group 110-1-CS456L --name acr-tasks --query ipAddress.ip --output table
Result
-----
20.212.90.81
azureuser@Azure:~$
```

8. Result



9. Create a replicated region for an Azure Container Registry && List Azure Container Registry(using Azure Cloud Shell)

```
Bash
az acr replication create --registry $ACR_NAME --location japaneast
{
  "id": "/subscriptions/ba9f5e30-1488-49db-bae0-9026dacc11b0/resourceGroups/110-1-cs456l/providers/Microsoft.ContainerRegistry/registries/10727211acr/replications/japaneast",
  "location": "japaneast",
  "name": "japaneast",
  "provisioningState": "Succeeded",
  "regionEndpointEnabled": true,
  "resourceGroup": "110-1-cs456l",
  "status": {
    "displayStatus": "Ready",
    "message": null,
    "timestamp": "2021-11-21T01:15:12.925634+00:00"
  },
  "systemData": {
    "createdAt": "2021-11-21T01:14:50.732295+00:00",
    "createdBy": "10727211@0365st.cycu.edu.tw",
    "createdByType": "User",
    "lastModifiedAt": "2021-11-21T01:14:50.732295+00:00",
    "lastModifiedBy": "10727211@0365st.cycu.edu.tw",
    "lastModifiedByType": "User"
  },
  "tags": {},
  "type": "Microsoft.ContainerRegistry/registries/replications",
  "zoneRedundancy": "Disabled"
}
az acr replication list --registry $ACR_NAME --output table
NAME          LOCATION     PROVISIONING STATE   STATUS
japaneast      japaneast   Succeeded       Ready
southeastasia southeastasia Succeeded       Ready
```

10. List Azure Container Registry (using Web)

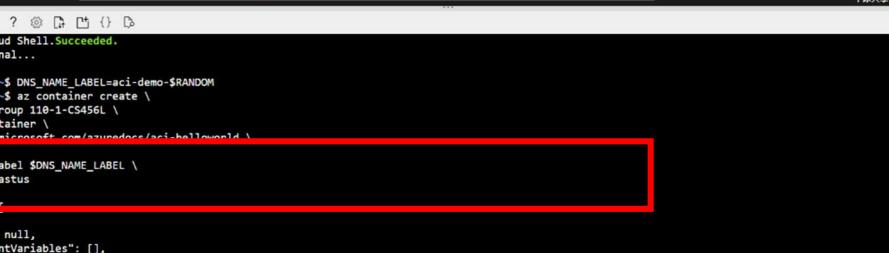
A screenshot of the Azure portal. The left sidebar shows the navigation path: Home > 110-1-CS456L > 10727211acr. The main area is titled "10727211acr | Replications". On the left, there is a sidebar with options like Webhooks, Replications, Tasks, and Connected registries (Preview). The main content area features a world map with green dots indicating replication locations. Below the map is a table with the following data:

Name	Location	Provisioning state	Status
japaneast	Japan East	Succeeded	Ready
southeastasia	Southeast Asia	Succeeded	Ready

Model 4:Run Docker containers with Azure Container

Instances

1. Create a container



The screenshot shows a Microsoft Azure Cloud Shell terminal window. The user is creating an Azure Container Instance (ACI) using the command `az container create`. A red box highlights the command line where the IP address of the container is being generated. The output shows the container has been successfully created with a public IP address.

```
Bash Requesting a Cloud Shell.Succeeded.
Connecting terminal...
azureuser@Azure:~$ DNS_NAME_LABEL=aci-demo-$RANDOM
azureuser@Azure:~$ az container create \
>   --resource-group 110-1-CS456L \
>   --name mycontavan \
>   --image mcr.microsoft.com/azuredocs/aci-helloworld \
>   --ports 88 \
>   --dns-name-label $DNS_NAME_LABEL \
>   --location eastus
{
  "ipAddress": "10.0.0.4",
  "command": null,
  "environmentVariables": [],
  "image": "mcr.microsoft.com/azuredocs/aci-helloworld",
  "instanceView": {
    "currentState": {
      "detailStatus": "",
      "exitCode": null,
      "finishTime": null,
      "startTime": "2021-11-21T01:24:39.868000+00:00",
      "state": "Running"
    },
    "events": [
      {
        "count": 1,
        "firstTimestamp": "2021-11-21T01:24:34+00:00",
        "lastTimestamp": "2021-11-21T01:24:34+00:00",
        "message": "Successfully pulled image \"mcr.microsoft.com/azuredocs/aci-helloworld@sha256:565dba8ce20ca1a311c2d9485089d7ddc935dd50140510050345a1b0ea4ffa6e\"",
        "name": "Pulled",
        "type": "Normal"
      }
    ]
  }
}
```

2. Check Container

The screenshot shows a Microsoft Azure interface with two main windows. The left window is a terminal session titled 'Bash' showing JSON configuration for a container instance named 'aci-demo-20547'. The right window is a browser displaying the 'Welcome to Azure Container Instances!' page, which includes a large blue cloud icon with a white arrow pointing up into a purple square containing three vertical bars.

```
dnsNameLabel": "aci-demo-20547",
"fqdn": "aci-demo-20547.eastus.azurecontainer.io",
"ip": "20.121.144.91",
"ports": [
  {
    "port": 80,
    "protocol": "TCP"
  }
],
"type": "Public",
},
"location": "eastus",
"name": "mycontainer",
"osType": "Linux",
"provisioningState": "Succeeded",
"resourceGroup": "110-1-CS456L",
"restartPolicy": "Always",
"sku": "Standard",
"subnetIds": null,
"tags": {},
"type": "Microsoft.ContainerInstance/containerGroups",
"volumes": null,
"zones": null
}
azurermuser@Azure:~$ az container show \
> --resource-group 110-1-CS456L \
> --name mycontainer \
> --query "[{QDN:ipAddress,fqn,ProvisioningState:provisioningState}]" \
> --output table
QDN                                ProvisioningState
aci-demo-20547.azurecontainer.io      Succeeded
azurermuser@Azure:~$
```

1. Create Container

```
azureuser@Azure:~$ az container create \
> --resource-group 110-1-CS456L \
> --name mycontainer-restart-demo \
> --image mcr.microsoft.com/azuredocs/aci-wordcount:latest \
> --restart-policy OnFailure \
> --location eastus
{
  "containers": [
    {
      "command": null,
      "environmentVariables": [],
      "image": "mcr.microsoft.com/azuredocs/aci-wordcount:latest",
      "instanceView": {
        "currentState": {
          "detailStatus": "Completed",
          "exitCode": 0,
          "finishTime": "2021-11-21T01:28:54.034000+00:00",
          "startTime": "2021-11-21T01:28:51.553000+00:00",
          "state": "Terminated"
        },
        "events": [
          {
            "count": 1,
            "firstTimestamp": "2021-11-21T01:28:46+00:00",
            "lastTimestamp": "2021-11-21T01:28:46+00:00",
            "message": "Successfully pulled image \"mcr.microsoft.com/azuredocs/aci-wordcount@sha256:f9bed361659288d722ce73be47e0777f583c48d1ad6ea5b2b242adf08805b473\"",
            "name": "Pulled",
            "type": "Normal"
          },
          {
            "count": 1,
            "firstTimestamp": "2021-11-21T01:28:46+00:00",
            "lastTimestamp": "2021-11-21T01:28:46+00:00",
            "message": "Container started \"mycontainer-restart-demo\""
          }
        ]
      }
    }
  ]
}
```

2. Check Status util 'Terminated' && View log

```
azureuser@Azure:~$ az container show \
> --resource-group 110-1-CS456L \
> --name mycontainer-restart-demo \
> --query "containers[0].instanceView.currentState.state"
"Terminated"
azureuser@Azure:~$ az container logs \
> --resource-group 110-1-CS456L \
> --name mycontainer-restart-demo
[("the", 990),
 ("and", 962),
 ("of", 656),
 ("to", 618),
 ("in", 544),
 ("you", 495),
 ("is", 453),
 ("my", 441),
 ("in", 399),
 ("HAMLET", 386)]
```

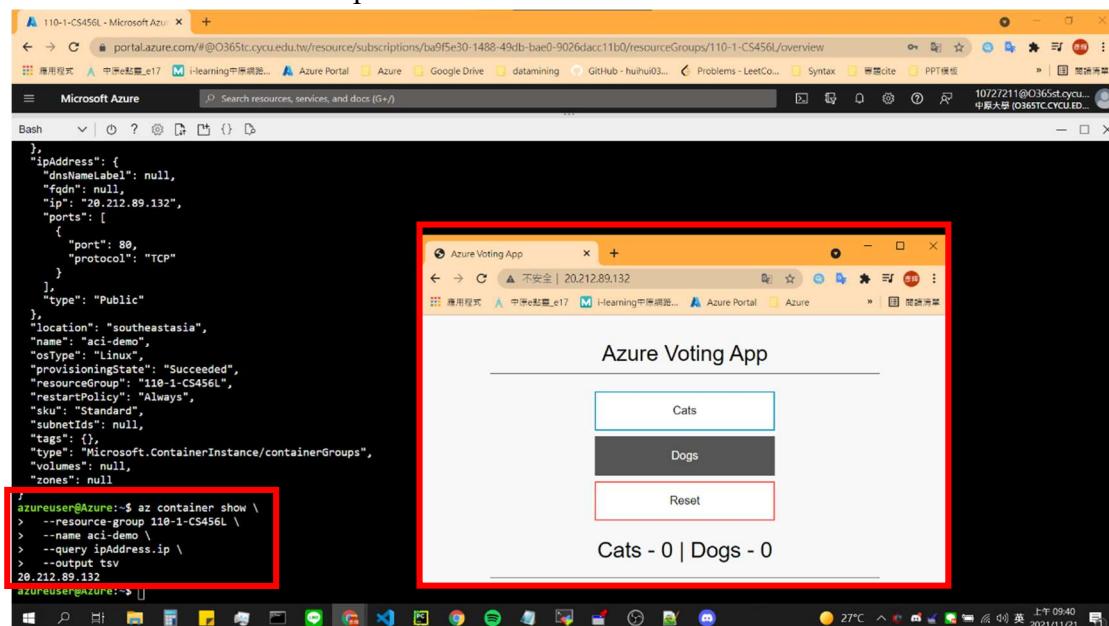
1. Create CosmosDB + save Bash variable && save key list

```
azureuser@Azure:~$ COSMOS_DB_ENDPOINT=$(az cosmosdb create \
> --resource-group 110-1-CS456L \
> --name $COSMOS_DB_NAME \
> --query documentEndpoint \
> --output tsv)
azureuser@Azure:~$ COSMOS_DB_MASTERKEY=$(az cosmosdb keys list \
> --resource-group 110-1-CS456L \
> --name $COSMOS_DB_NAME \
> --query primaryMasterKey \
> --output tsv)
azureuser@Azure:~$
```

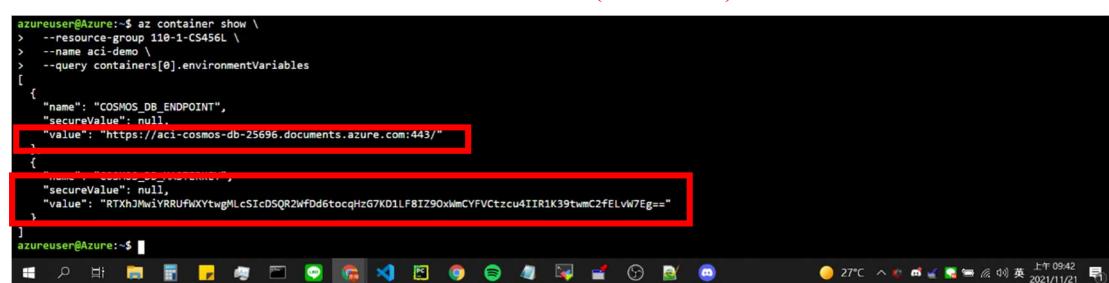
2. Deployment a container that works with CosmosDB

```
azureuser@Azure:~$ az container create \
> --resource-group 110-1-CS456L \
> --name aci-demo \
> --image mcr.microsoft.com/azuredocs/azure-vote-front:cosmosdb \
> --ip-address Public \
> --location southeastasia \
> --environment-variables \
>   COSMOS_DB_ENDPOINT=$COSMOS_DB_ENDPOINT \
>   COSMOS_DB_MASTERKEY=$COSMOS_DB_MASTERKEY
{
  "containers": [
    {
      "command": null,
      "environmentVariables": [
        {
          "name": "COSMOS_DB_ENDPOINT",
          "secureValue": null,
          "value": "https://aci-cosmos-db-25696.documents.azure.com:443/"
        },
        {
          "name": "COSMOS_DB_MASTERKEY",
          "secureValue": null,
          "value": "RTXhJMWiYRRUWXYtwgMLcS1cDSQR2WFd6tocQHzG7KD1LF8IZ90xwmCYFVCtzcu4IIR1K39twmC2fELvW7Eg=="
        }
      ],
      "image": "mcr.microsoft.com/azuredocs/azure-vote-front:cosmosdb",
      "instanceView": {
        "currentState": {
          "detailStatus": "",
          "exitCode": null,
          "finishTime": null,
          "startTime": "2021-11-21T01:38:36.330000+00:00",
          "state": "Running"
        }
      }
    }
  ]
}
```

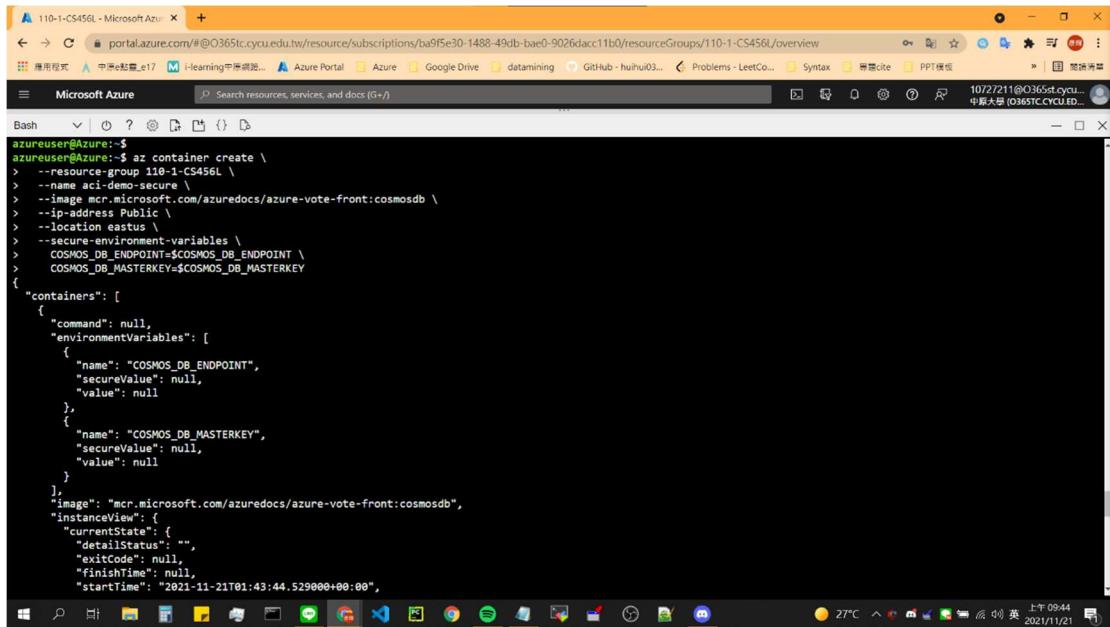
3. show IP address and Open Browser



4. Show Container's environment variables (Viewable)

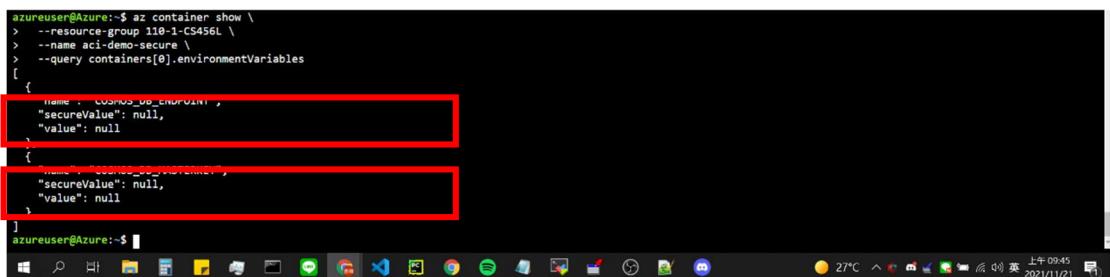


5. Create second container (using secure-environment-variables)



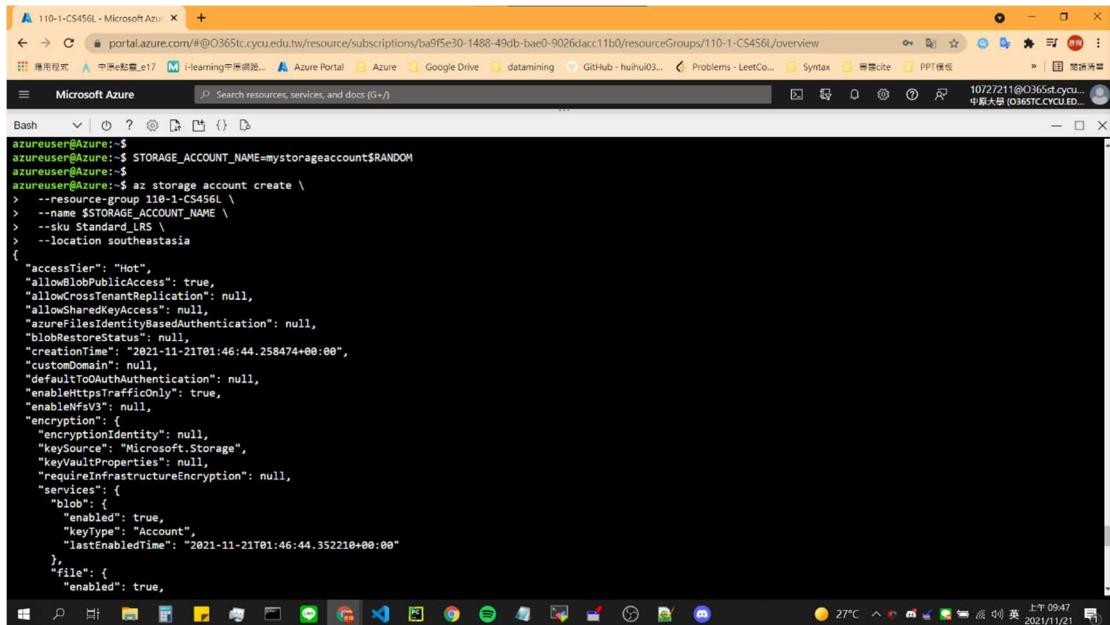
```
azureuser@Azure:~$ az container create \
> --resource-group 110-1-CS456L \
> --name aci-demo-secure \
> --image mcr.microsoft.com/azuredocs/azure-vote-front:cosmosdb \
> --ip-address Public \
> --location eastus \
> --secure-environment-variables \
> COSMOS_DB_ENDPOINT=$COSMOS_DB_ENDPOINT \
> COSMOS_DB_MASTERKEY=$COSMOS_DB_MASTERKEY
{
  "containers": [
    {
      "command": null,
      "environmentVariables": [
        {
          "name": "COSMOS_DB_ENDPOINT",
          "secureValue": null,
          "value": null
        },
        {
          "name": "COSMOS_DB_MASTERKEY",
          "secureValue": null,
          "value": null
        }
      ],
      "image": "mcr.microsoft.com/azuredocs/azure-vote-front:cosmosdb",
      "instanceView": {
        "currentState": {},
        "detailStatus": "",
        "exitCode": null,
        "finishTime": null,
        "startTime": "2021-11-21T01:43:44.529000+00:00",
        "state": "Running"
      }
    }
  ]
}
```

6. Show Container's environment variables (UnViewable)



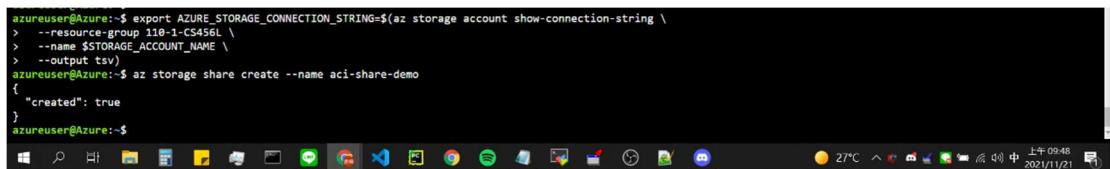
```
azureuser@Azure:~$ az container show \
> --resource-group 110-1-CS456L \
> --name aci-demo-secure \
> --query containers[0].environmentVariables
[
  {
    "name": "COSMOS_DB_ENDPOINT",
    "secureValue": null,
    "value": null
  },
  {
    "name": "COSMOS_DB_MASTERKEY",
    "secureValue": null,
    "value": null
  }
]
```

1. Create Azure storage



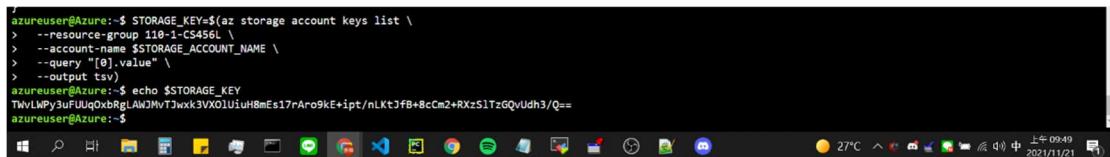
```
azureuser@Azure:~$ az storage account create \
>   --resource-group 110-1-CS456L \
>   --name $STORAGE_ACCOUNT_NAME \
>   --sku Standard_LRS \
>   --location southeastasia
{
  "accessTier": "Hot",
  "allowBlobPublicAccess": true,
  "allowCrossTenantReplication": null,
  "allowSharedKeyAccess": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2021-11-21T01:46:44.258474+00:00",
  "customDomain": null,
  "defaultToOAuthAuthentication": null,
  "enableHttpsTrafficOnly": true,
  "enableIdfsV3": null,
  "encryption": {
    "encryptionIdentity": null,
    "keySource": "Microsoft.Storage",
    "keyVaultProperties": null,
    "requireInfrastructureEncryption": null,
    "services": {
      "blob": {
        "enabled": true,
        "keyType": "Account",
        "lastEnabledTime": "2021-11-21T01:46:44.352210+00:00"
      },
      "file": {
        "enabled": true,
```

2. Get Connect String & Create File Share



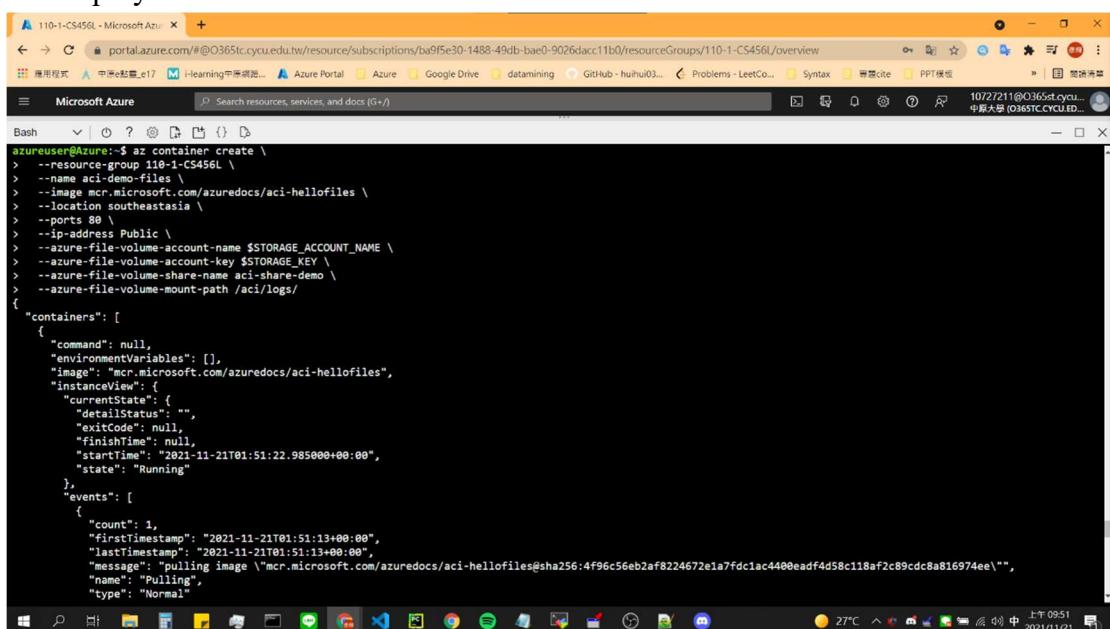
```
azureuser@Azure:~$ export AZURE_STORAGE_CONNECTION_STRING=$(az storage account show-connection-string \
>   --resource-group 110-1-CS456L \
>   --name $STORAGE_ACCOUNT_NAME \
>   --output tsv)
azureuser@Azure:~$ az storage share create --name aci-share-demo
{
  "created": true
}
azureuser@Azure:~$
```

3. Get storage credentials



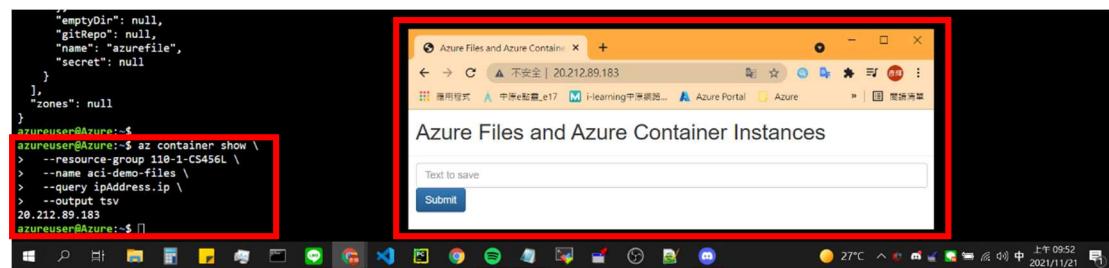
```
azureuser@Azure:~$ STORAGE_KEY=$(az storage account keys list \
>   --resource-group 110-1-CS456L \
>   --account-name $STORAGE_ACCOUNT_NAME \
>   --query "[0].value" \
>   --output tsv)
azureuser@Azure:~$ echo $STORAGE_KEY
TwvLWPy3ufUUqbxRgLwJHvTJwxk3VXO1UiUh8mEs17rAro9kE+ipt/nLkJfB+8cCm+RXzSlTzGQvUdh3/Q==
azureuser@Azure:~$
```

4. Deploy a container and mount the file share



```
azureuser@Azure:~$ az container create \
>   --resource-group 110-1-CS456L \
>   --name aci-demo-files \
>   --image mcr.microsoft.com/azuredocs/aci-hellofiles \
>   --location southeastasia \
>   --ports 80 \
>   --ip-address Public \
>   --azure-file-volume-account-name $STORAGE_ACCOUNT_NAME \
>   --azure-file-volume-account-key $STORAGE_KEY \
>   --azure-file-volume-share-name aci-share-demo \
>   --azure-file-volume-mount-path /aci/logs/
{
  "containers": [
    {
      "command": null,
      "environmentVariables": [],
      "image": "mcr.microsoft.com/azuredocs/aci-hellofiles",
      "instanceView": {
        "currentState": {
          "detailsStatus": "",
          "exitCode": null,
          "finishTime": null,
          "startTime": "2021-11-21T01:51:22.985000+00:00",
          "state": "Running"
        },
        "events": [
          {
            "count": 1,
            "firstTimestamp": "2021-11-21T01:51:13+00:00",
            "lastTimestamp": "2021-11-21T01:51:13+00:00",
            "message": "pulling image \\"mcr.microsoft.com/azuredocs/aci-hellofiles@sha256:4f96c56eb2af8224672e1a7fdc1ac4400eadf4d58c118af2c89cdc8a816974ee\\\"",
            "name": "Pulling",
            "type": "Normal"
          }
        ]
      }
    }
  ],
  "osType": "Linux"
}
```

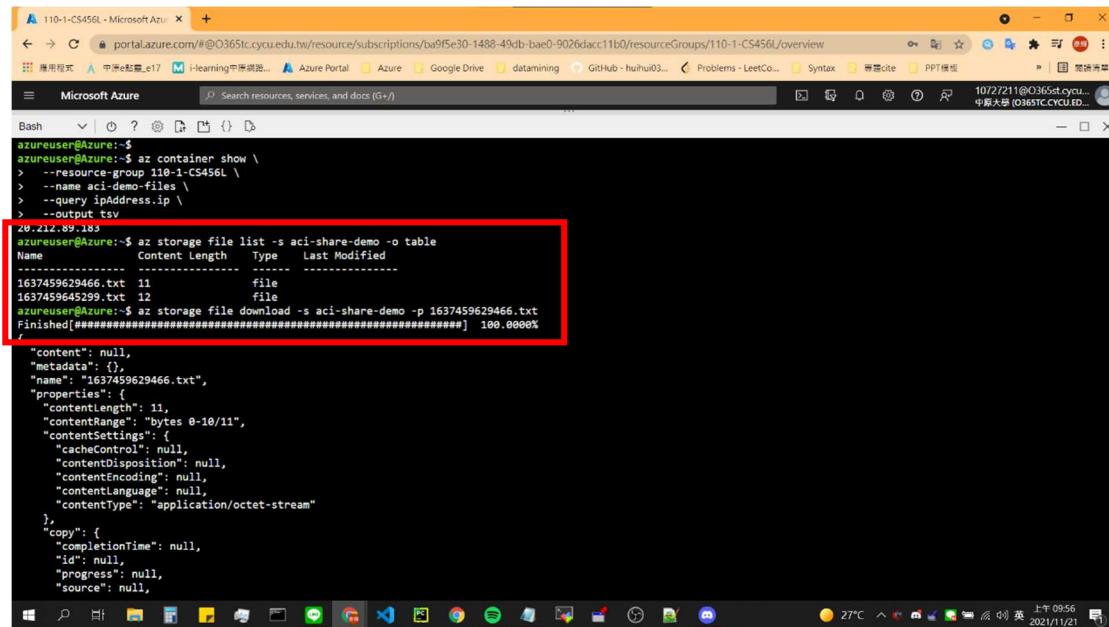
5. Get IP Address



```
        "emptyDir": null,
        "gitRepo": null,
        "name": "azurefile",
        "secret": null
    },
    "zones": null
}
azreuser@Azure:~$ az container show \
> --resource-group 110-1-CS456L \
> --name aci-demo-files \
> --query ipAddress.ip \
> --output tsv
20.212.89.183
azreuser@Azure:~$
```

Azure Files and Azure Container Instances
Text to save
Submit

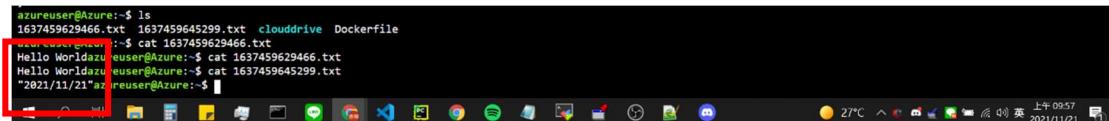
6. After Insert Text in Browser, Look & Download File in Azure Cloud Shell.



```
azreuser@Azure:~$ az storage file list -s aci-share-demo -o table
Name          Content Length  Type   Last Modified
-----        -----        ----   -----
1637459629466.txt 11          file
1637459645299.txt 12          file
azreuser@Azure:~$ az storage file download -s aci-share-demo -p 1637459629466.txt
Finished[=====] 100.0000%
```

```
"/
"content": null,
"metadata": {},
"name": "1637459629466.txt",
"properties": {
    "contentLength": 11,
    "contentRange": "bytes 0-10/11",
    "contentSettings": {
        "cacheControl": null,
        "contentDisposition": null,
        "contentEncoding": null,
        "contentLanguage": null,
        "contentType": "application/octet-stream"
    },
    "copy": {
        "completionTime": null,
        "id": null,
        "progress": null,
        "source": null,
    }
}
```

7. Show text



```
azreuser@Azure:~$ ls
1637459629466.txt 1637459645299.txt  clouddrive Dockerfile
azreuser@Azure:~$ cat 1637459629466.txt
Hello World
azreuser@Azure:~$ cat 1637459629466.txt
Hello World
azreuser@Azure:~$ cat 1637459645299.txt
"2021/11/21"azreuser@Azure:~$
```

1. Get logs from your container instance

```
Bash < | ? | > | < > | D

azureuser@Azure:~$ az container logs \
> --resource-group 110-1-CS456L \
> --name aci-demo
sh: NGINX_WORKER_CONNECTIONS: unknown operand
sh: NGINX_WORKER_OPEN_FILES: unknown operand
Checking for script in /app/prestart.sh
Running script /app/prestart.sh
Running inside /app/prestart.sh, you could add migrations to this file, e.g.:

#!/usr/bin/env bash

# Let the DB start
sleep 10;
# Run migrations
alembic upgrade head

/usr/lib/python2.7/site-packages/supervisor/options.py:298: UserWarning: Supervisor is running as root and it is searching for its configuration file in default locations (including its current working directory); you probably want to specify a "-c" argument specifying an absolute path to a configuration file for improved security.
    'Supervisor is running as root and it is searching '
2021-11-21 01:38:36,067 CRIT Supervisor running as root (no user in config file)
2021-11-21 01:38:36,068 INFO Included extra file "/etc/supervisor.d/supervisord.ini" during parsing
2021-11-21 01:38:36,076 INFO RPC interface 'supervisor' initialized
2021-11-21 01:38:36,076 CRIT Server 'unix_http_server' running without any HTTP authentication checking
2021-11-21 01:38:36,077 INFO supervisord started with pid 17
2021-11-21 01:38:37,079 INFO spawned: 'nginx' with pid 25
2021-11-21 01:38:37,082 INFO spawned: 'uwsgi' with pid 26
[uWSGI] getting INI configuration from /app/uwsgi.ini
[uWSGI] getting INI configuration from /etc/uwsgi/uwsgi.ini
*** Starting uwsgi 2.0.17 (64bit) on [Sun Nov 21 01:38:37 2021] ***
compiled with version: 6.4.0 on 27 March 2018 12:43:27
os: Linux-5.4.81-microsoft-standard #1 SMP Thu Dec 3 23:47:24 UTC 2020
node name: SandboxHost-637730554929238345
machine: x86_64
```

2. Get Container events

3 Execute a command in container

```
^Cazureuser@Azure:~$ az container exec \
>   --resource-group 110-1-CS456L \
>   --name aci-demo \
>   --exec-command /bin/sh
/app
/app # ls
._pycache_ config_file.cfg main.py prestart.sh static supervisord.pid templates uwsgi.ini
/app #
```

1. Monitor CPU

```
azureuser@Azure:~$ az container show \
> --resource-group 110-1-CS456L \
> --name aci-demo \
> --query id \
> --output tsv
azuser@Azure:~$ az monitor metrics list \
> --resource $CONTAINER_ID \
> --metrics CPUUsage \
> --output table
Timestamp      Name      Average
-----
2021-11-21 01:04:00 CPU Usage    0.0
2021-11-21 01:05:00 CPU Usage    0.0
2021-11-21 01:06:00 CPU Usage    0.0
2021-11-21 01:07:00 CPU Usage    0.0
2021-11-21 01:08:00 CPU Usage    0.0
2021-11-21 01:09:00 CPU Usage    0.0
2021-11-21 01:10:00 CPU Usage    0.0
2021-11-21 01:11:00 CPU Usage    0.0
2021-11-21 01:12:00 CPU Usage    0.0
2021-11-21 01:13:00 CPU Usage    0.0
2021-11-21 01:14:00 CPU Usage    0.0
2021-11-21 01:15:00 CPU Usage    0.0
2021-11-21 01:16:00 CPU Usage    0.0
2021-11-21 01:17:00 CPU Usage    0.0
2021-11-21 01:18:00 CPU Usage    0.0
2021-11-21 01:19:00 CPU Usage    0.0
2021-11-21 01:20:00 CPU Usage    0.0
2021-11-21 01:21:00 CPU Usage    0.0
2021-11-21 01:22:00 CPU Usage    0.0
2021-11-21 01:23:00 CPU Usage    0.0
2021-11-21 01:24:00 CPU Usage    0.0
2021-11-21 01:32:00 CPU Usage    0.0
2021-11-21 01:33:00 CPU Usage    0.0
2021-11-21 01:34:00 CPU Usage    0.0
2021-11-21 01:35:00 CPU Usage    0.0
2021-11-21 01:36:00 CPU Usage    0.0
2021-11-21 01:37:00 CPU Usage    0.0
2021-11-21 01:38:00 CPU Usage    0.0
2021-11-21 01:39:00 CPU Usage    0.0
2021-11-21 01:40:00 CPU Usage    0.0
2021-11-21 01:41:00 CPU Usage    1.0
2021-11-21 01:42:00 CPU Usage    0.0
2021-11-21 01:43:00 CPU Usage    0.0
2021-11-21 01:44:00 CPU Usage    0.0
2021-11-21 01:45:00 CPU Usage    0.0
2021-11-21 01:46:00 CPU Usage    0.0
2021-11-21 01:47:00 CPU Usage    0.0
2021-11-21 01:48:00 CPU Usage    1.0
2021-11-21 01:49:00 CPU Usage    0.5
2021-11-21 01:50:00 CPU Usage    0.0
2021-11-21 01:51:00 CPU Usage    0.0
2021-11-21 01:52:00 CPU Usage    0.0
2021-11-21 01:53:00 CPU Usage    0.0
2021-11-21 01:54:00 CPU Usage    0.0
2021-11-21 01:55:00 CPU Usage    0.0
2021-11-21 01:56:00 CPU Usage    0.0
2021-11-21 01:57:00 CPU Usage    0.0
2021-11-21 01:58:00 CPU Usage    0.0
2021-11-21 01:59:00 CPU Usage    0.0
2021-11-21 02:00:00 CPU Usage    0.0
2021-11-21 02:01:00 CPU Usage    0.0
2021-11-21 02:02:00 CPU Usage    0.0
2021-11-21 02:03:00 CPU Usage    0.0
2021-11-21 02:04:00 CPU Usage    0.0
azuser@Azure:~$
```

2. Monitor Memory

```
2021-11-21 02:03:00 CPU Usage 0.0
azuser@Azure:~$ az monitor metrics list \
> --resource $CONTAINER_ID \
> --metrics MemoryUsage \
> --output table
Timestamp      Name      Average
-----
2021-11-21 01:04:00 Memory Usage 55767040.0
2021-11-21 01:05:00 Memory Usage 62144512.0
2021-11-21 01:06:00 Memory Usage 68431872.0
2021-11-21 01:07:00 Memory Usage 68431872.0
2021-11-21 01:08:00 Memory Usage 68431872.0
2021-11-21 01:09:00 Memory Usage 68431872.0
2021-11-21 01:10:00 Memory Usage 68431872.0
2021-11-21 01:11:00 Memory Usage 68431872.0
2021-11-21 01:12:00 Memory Usage 68431872.0
2021-11-21 01:13:00 Memory Usage 69541888.0
2021-11-21 01:14:00 Memory Usage 76639616.0
2021-11-21 01:15:00 Memory Usage 76631424.0
2021-11-21 01:16:00 Memory Usage 76631424.0
2021-11-21 01:17:00 Memory Usage 76631424.0
2021-11-21 01:18:00 Memory Usage 76631424.0
2021-11-21 01:19:00 Memory Usage 76631424.0
2021-11-21 01:20:00 Memory Usage 76631424.0
2021-11-21 01:21:00 Memory Usage 76631424.0
2021-11-21 01:22:00 Memory Usage 76631424.0
2021-11-21 01:23:00 Memory Usage 76631424.0
2021-11-21 01:24:00 Memory Usage 76631424.0
2021-11-21 01:25:00 Memory Usage 76631424.0
2021-11-21 01:26:00 Memory Usage 76631424.0
2021-11-21 01:27:00 Memory Usage 76631424.0
2021-11-21 01:28:00 Memory Usage 76631424.0
2021-11-21 01:29:00 Memory Usage 76631424.0
2021-11-21 01:30:00 Memory Usage 76631424.0
2021-11-21 01:33:00 Memory Usage 76631424.0
2021-11-21 01:34:00 Memory Usage 76631424.0
2021-11-21 01:35:00 Memory Usage 76631424.0
2021-11-21 01:36:00 Memory Usage 76631424.0
2021-11-21 01:37:00 Memory Usage 76631424.0
2021-11-21 01:38:00 Memory Usage 76631424.0
2021-11-21 01:39:00 Memory Usage 76631424.0
2021-11-21 01:40:00 Memory Usage 76631424.0
2021-11-21 01:41:00 Memory Usage 76631424.0
2021-11-21 01:42:00 Memory Usage 76631424.0
2021-11-21 01:43:00 Memory Usage 76631424.0
2021-11-21 01:44:00 Memory Usage 76631424.0
2021-11-21 01:45:00 Memory Usage 76631424.0
2021-11-21 01:46:00 Memory Usage 76631424.0
2021-11-21 01:47:00 Memory Usage 76631424.0
2021-11-21 01:48:00 Memory Usage 76631424.0
2021-11-21 01:49:00 Memory Usage 76631424.0
2021-11-21 01:50:00 Memory Usage 76631424.0
2021-11-21 01:51:00 Memory Usage 76631424.0
2021-11-21 01:52:00 Memory Usage 76631424.0
2021-11-21 01:53:00 Memory Usage 76631424.0
2021-11-21 01:54:00 Memory Usage 76631424.0
2021-11-21 01:55:00 Memory Usage 76631424.0
2021-11-21 01:56:00 Memory Usage 76631424.0
2021-11-21 01:57:00 Memory Usage 76631424.0
2021-11-21 01:58:00 Memory Usage 76631424.0
2021-11-21 01:59:00 Memory Usage 76631424.0
2021-11-21 02:00:00 Memory Usage 76631424.0
2021-11-21 02:01:00 Memory Usage 76631424.0
2021-11-21 02:02:00 Memory Usage 76631424.0
2021-11-21 02:03:00 Memory Usage 76631424.0
2021-11-21 02:04:00 Memory Usage 76631424.0
2021-11-21 02:05:00 Memory Usage 76631424.0
2021-11-21 02:06:00 Memory Usage 76631424.0
2021-11-21 02:07:00 Memory Usage 76631424.0
2021-11-21 02:08:00 Memory Usage 76631424.0
2021-11-21 02:09:00 Memory Usage 76631424.0
2021-11-21 02:10:00 Memory Usage 76631424.0
2021-11-21 02:11:00 Memory Usage 76631424.0
2021-11-21 02:12:00 Memory Usage 76631424.0
2021-11-21 02:13:00 Memory Usage 76631424.0
2021-11-21 02:14:00 Memory Usage 76631424.0
2021-11-21 02:15:00 Memory Usage 76631424.0
2021-11-21 02:16:00 Memory Usage 76631424.0
2021-11-21 02:17:00 Memory Usage 76631424.0
2021-11-21 02:18:00 Memory Usage 76631424.0
2021-11-21 02:19:00 Memory Usage 76631424.0
2021-11-21 02:20:00 Memory Usage 76631424.0
2021-11-21 02:21:00 Memory Usage 76631424.0
2021-11-21 02:22:00 Memory Usage 76631424.0
2021-11-21 02:23:00 Memory Usage 76631424.0
2021-11-21 02:24:00 Memory Usage 76631424.0
2021-11-21 02:25:00 Memory Usage 76631424.0
2021-11-21 02:26:00 Memory Usage 76631424.0
2021-11-21 02:27:00 Memory Usage 76631424.0
2021-11-21 02:28:00 Memory Usage 76631424.0
2021-11-21 02:29:00 Memory Usage 76631424.0
2021-11-21 02:30:00 Memory Usage 76631424.0
2021-11-21 02:31:00 Memory Usage 76631424.0
2021-11-21 02:32:00 Memory Usage 76631424.0
2021-11-21 02:33:00 Memory Usage 76631424.0
2021-11-21 02:34:00 Memory Usage 76631424.0
2021-11-21 02:35:00 Memory Usage 76631424.0
2021-11-21 02:36:00 Memory Usage 76631424.0
2021-11-21 02:37:00 Memory Usage 76631424.0
2021-11-21 02:38:00 Memory Usage 76631424.0
2021-11-21 02:39:00 Memory Usage 76631424.0
2021-11-21 02:40:00 Memory Usage 76631424.0
2021-11-21 02:41:00 Memory Usage 76631424.0
2021-11-21 02:42:00 Memory Usage 76631424.0
2021-11-21 02:43:00 Memory Usage 76631424.0
2021-11-21 02:44:00 Memory Usage 76631424.0
2021-11-21 02:45:00 Memory Usage 76631424.0
2021-11-21 02:46:00 Memory Usage 76631424.0
2021-11-21 02:47:00 Memory Usage 76631424.0
2021-11-21 02:48:00 Memory Usage 76631424.0
2021-11-21 02:49:00 Memory Usage 76631424.0
2021-11-21 02:50:00 Memory Usage 76631424.0
2021-11-21 02:51:00 Memory Usage 76631424.0
2021-11-21 02:52:00 Memory Usage 76631424.0
2021-11-21 02:53:00 Memory Usage 76631424.0
2021-11-21 02:54:00 Memory Usage 76631424.0
2021-11-21 02:55:00 Memory Usage 76631424.0
2021-11-21 02:56:00 Memory Usage 76631424.0
2021-11-21 02:57:00 Memory Usage 76631424.0
2021-11-21 02:58:00 Memory Usage 76631424.0
2021-11-21 02:59:00 Memory Usage 76631424.0
2021-11-21 03:00:00 Memory Usage 76631424.0
azuser@Azure:~$
```

3. Container Instances Overview

The screenshot shows the Azure portal interface for the 'aci-demo' container instance. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main content area displays the container instance details and two performance charts.

Container Instance Details:

- Name:** aci-demo
- Subscription (Move):** 中原大學
- Subscription ID:** ba9f5e30-1488-49db-bae0-9026dacc11b0
- Tags (Edit):** Click here to add tags
- Location:** Southeast Asia
- FQDN:** ...
- Container count:** 1

Performance Metrics:

- CPU:** A line chart showing CPU usage over time, with a sharp peak around 9:45 AM UTC+08:00.
- Memory:** A line chart showing memory usage over time, showing a step increase at approximately 9:45 AM UTC+08:00.

Model 5:Deploy and run a containerized web app with

Azure App Service

1. Download Code and build Images

```
azureuser@azure:~/1121_experiment/Model5_AzureAppService$ git clone https://github.com/MicrosoftDocs/mslearn-deploy-run-container-app-service.git
Cloning into 'mslearn-deploy-run-container-app-service'...
remote: Enumerating objects: 208, done.
remote: Counting objects: 100%, 86(86), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 208 (delta 20), reused 57 (delta 18), pack-reused 122
Receiving objects: 100% (208/208), 1.28 MiB | 22.96 MiB/s, done.
Resolving deltas: 100% (37/37), done.

azureuser@azure:~/1121_experiment/Model5_AzureAppService$ cd mslearn-deploy-run-container-app-service/
dotnet /git/ node
azureuser@azure:~/1121_experiment/Model5_AzureAppService$ cd mslearn-deploy-run-container-app-service/dotnet/
azureuser@azure:~/1121_experiment/Model5_AzureAppService/mslearn-deploy-run-container-app-service/dotnet$ ls
Dockerfile SammleWeb SammleWeb.sln

azureuser@azure:~/1121_experiment/Model5_AzureAppService/mslearn-deploy-run-container-app-service/dotnet$ az acr build --registry 10727211ContainerRegistry --image webimage

Packing source code into tar to upload...
Excluding '.gitignore' based on default ignore rules
Uploading archived source code from '/tmp/build/archive_bef48fb5c10b43d8a6beec905c6facb9.tar.gz'...
Sending context (914.58 KB) to registry: 10727211ContainerRegistry...
Quoted build with ID: cmi
Waiting for an agent...
2021/11/21 04:38:14 Downloading source code...
2021/11/21 04:38:15 Finished downloading source code
2021/11/21 04:38:15 Using acb_vol_08586da2-215b-43bc-9f26-145e52414d5e as the home volume
2021/11/21 04:38:15 Setting up Docker configuration...
2021/11/21 04:38:16 Successfully set up Docker configuration
2021/11/21 04:38:16 Logging in to registry: 10727211containerregistry.azurecr.io
2021/11/21 04:38:17 Successfully logged into 10727211containerregistry.azurecr.io
2021/11/21 04:38:17 Executing step ID: build. Timeout(sec): 28800, Working directory: '', Network: ''
2021/11/21 04:38:17 Scanning for dependencies...
2021/11/21 04:38:17 Successfully scanned dependencies
2021/11/21 04:38:17 Launching container with name: build
Sending build context to Docker daemon 4.38MB
```

2. Check Container Registry

The screenshot shows the Microsoft Azure portal interface for a Container Registry. The left sidebar includes 'Home', 'Container registry', 'Identity', 'Networking', 'Security', 'Locks', 'Services', and 'Repositories'. The 'Repositories' section is currently selected. The main area displays the '10727211ContainerRegistry' page with a search bar and a refresh button. Below the search bar is a dropdown menu for 'Search to filter repositories ...' containing 'Repositories ↑' and 'webimage ...'. The 'webimage' repository is highlighted. The repository details page shows the following information:

Repository	Tag count
webimage	1

Details: Last updated date 11/21/2021, 12:49 PM GMT+8. Manifest count 1.

Below the repository details, there is a 'Tags ↑' section with a 'latest' entry.

3. Create ‘Web App’ and Test

The screenshot shows two windows side-by-side. The left window is the Microsoft Azure portal for the app service '10727211-WebApp'. It displays the 'Overview' blade with the following details:

- Resource group (Move) : 110-1-CS456L
- Status : Running
- Location : Japan East
- Subscription (Move) : 中原大學
- Subscription ID : ba9f5e30-1488-49db-bae0-9026dacc11b0

The right section of the Azure window shows deployment settings:

URL	: https://10727211-webapp.azurewebsites.net
Health Check	: Not Configured
App Service Plan	: ASP-1101CS456L-024d (P1v2: 1)
FTP/deployment username	: No FTP/deployment user set
FTP hostname	: ftp://waws-prod-t1-047.ftp.azurewebsites.windows.net/site/www...
FTPS hostname	: https://waws-prod-t1-047.ftp.azurewebsites.windows.net/site/www...

The 'JSON View' button is located at the top right of the Azure window.

The right window is a Microsoft Edge browser displaying the deployed application at the URL https://10727211-webapp.azurewebsites.net. The page content includes:

Express

Welcome to Express

4. Configure continuous deployment and create a webhook

The screenshot shows the Azure Deployment Center interface. The left sidebar lists 'Deployment', 'Events (preview)', 'Settings', 'Configuration', 'Authentication', 'Application Insights', 'Identity', 'Backups', 'Custom domains', and 'TLS/SSL settings'. The main area is titled 'Registry settings' and contains fields for 'Container Type' (Single Container), 'Registry source' (Azure Container Registry), 'Subscription ID' (中原大學), 'Registry' (10727211ContainerRegistry), 'Image' (webimage), 'Tag' (latest), and 'Startup File' (index.js). A red box highlights the 'Continuous deployment' section, which has an 'On' radio button selected. Below it is a 'Webhook URL' field with a placeholder '.....'.

5. Update the web app

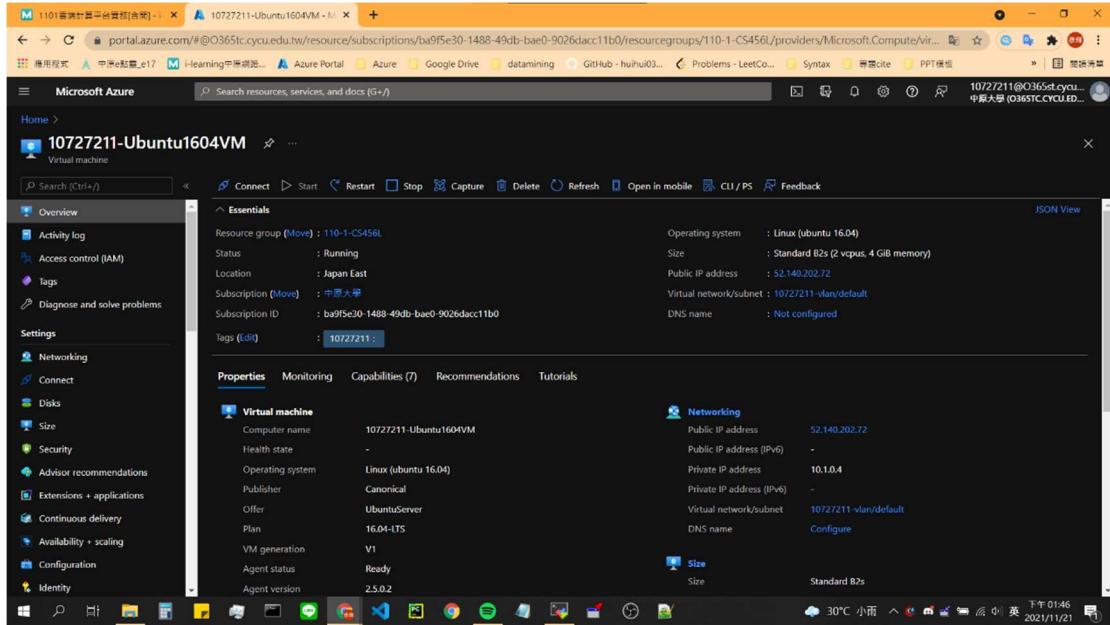
The screenshot shows a terminal window with Bash selected. It displays the contents of index.js:index.js
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function(req, res, next) {
6 res.render('index', { title: 'Midrosft Learn' });
7 });
8
9 module.exports = router;At the bottom of the terminal, the command `app.js bin Dockerfile package.json public routes view` is shown being run.

6. Check Webhook event & Test Web app

The screenshot shows the Azure Container Registry interface. The left sidebar lists 'Services', 'Repositories', 'Webhooks', and 'Replications'. The main area shows a table for a webhook named 'webapp10727211WebApp'. The table includes columns for 'Actions' (push), 'Scope' (webimageatest), 'Status' (On), and 'Provisioning state' (Succeeded). The last entry in the table is a push action from 'webimageatest' at 11:24 PM on November 21, 2021. Below this, a browser window shows the 'Midrosft Learn' website with the message 'Welcome to Midrosft Learn'.

Model 6:Introduction to Kubernetes

1. Environment (Azure VM)



2. Install MicroK8s

```
Reading package lists... Done
azuser@10727211-Ubuntu1604VM:~$ sudo snap install microk8s --classic
2021-11-21T09:48:04Z [INFO] Waiting for automatic snapd restart...
microk8s (1.22/stable) v1.22.3 from Canonical installed
azuser@10727211-Ubuntu1604VM:~$ sudo microk8s.status --wait-ready
microk8s is running
high-availability: no
datastore master nodes: 127.0.0.1:19001
datastore standby nodes: none
addons:
  enabled:
    ha-cluster      # Configure high availability on the current node
  disabled:
    ambassador     # Ambassador API Gateway and Ingress
    cilium         # SDN, fast with full network policy
    dashboard      # The Kubernetes dashboard
    dnsmasq        # DNSMasq
    fluentd       # Elasticsearch-Fluentd-Kibana logging and monitoring
    gpu            # Automatic enablement of Nvidia CUDA
    helm           # Helm 2 - the package manager for Kubernetes
    helm3          # Helm 3 - Kubernetes package manager
    host-access    # Host access Pod connecting to external services smoothly
    kubelet        # Kubernetes Kubelet
    metrics-server # Metrics Server
    multus         # Multus enables multiple network interfaces to pods
    openebs        # OpenEBS is an open-source storage solution for Kubernetes
    openfaas       # openfaas Serverless framework
    portainer     # Portainer UI for your Kubernetes cluster
    prometheus    # Prometheus operator for monitoring and logging
    rbac           # Role-Based Access Control for authorisation
    registry       # Private Image registry exposed on localhost:32000
    storage        # Storage class; allocates storage from host directory
    traefik        # Traefik Ingress controller for external access
```

3. Prepare the cluster

```
azuser@10727211-Ubuntu1604VM:~$ sudo microk8s.enable dns dashboard registry
Enabling DNS...
Annotating DNS...
serviceaccount/coredns created
configmap/coredns created
Warning: spec.template.metadata.annotations[scheduler.alpha.kubernetes.io/critical-pod]: non-functional in v1.16+; use the "priorityClassName" field instead
deployment.apps/coredns created
service/kube-dns created
clusterrole.rbac.authorization.k8s.io/coredns created
clusterrolebinding.rbac.authorization.k8s.io/coredns created
Restarting kubelet
DNS is enabled
Enabling Kubernetes Dashboard
Enabling Metrics Server
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/vbeta1.metrics.k8s.io created
clusterrolebinding.rbac.authorization.k8s.io/microk8s-admin created
Metrics Server is enabled
Applying manifest
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
```

4. Explore the Kubernetes cluster && check status

```
azureuser@10727211-Ubuntu1604VM:~ Terminal Sessions View X server Tools Games Settings Macros Help Session Servers Tools Games Sessions View Split MultiSec Tunneling Packages Settings Help Quick connect... https://mobaxterm.mobatek.net https://home/mobaxterm https://azuser@10727211-Ubuntu1604VM:~ Enabling default storage class deployment.apps/hostpath-provisioner created storageclass.storage.k8s.io/microk8s-hostpath created serviceaccount/microk8s-hostpath created clusterrolebinding/microk8s-hostpath created clusterrolebinding/microk8s-hostpath created Storage will be available soon Applying registry manifest namespace/container-registry created persistentvolumeclaim/registry-claim created deployment.apps/registry created service/registry created configmap/local-registry-hosting configured The registry is running! azuser@10727211-Ubuntu1604VM:~$ sudo snap alias microk8s.kubectl kubectl Added: - microk8s.kubectl as kubectl NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER _RUNTIME 10727211-ubuntu1604vm Ready <none> 3m46s v1.22.3-3+9ec7c40ec93c7 10.1.0.4 <none> Ubuntu 16.04.7 LTS 4.15.0-1113-azure container azuser@10727211-ubuntu1604VM:~$ sudo kubectl get nodes -o wide NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER _RUNTIME 10727211-ubuntu1604vm Ready <none> 4m26s v1.22.3-3+9ec7c40ec93c7 10.1.0.4 <none> Ubuntu 16.04.7 LTS 4.15.0-1113-azure container azuser@10727211-ubuntu1604VM:~$ sudo kubectl get services -o yaml --all-namespaces NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR default kubernetes ClusterIP 10.152.183.1 <none> 443/TCP 4044s <none> kube-system kube-dns ClusterIP 10.152.183.10 <none> 53/UDP,53/TCP,9153/TCP 2057s k8s-app=kube-dns kube-system metrics-server ClusterIP 10.152.183.112 <none> 443/TCP 2042s kds-app=metrics-server kube-system kubernetes-dashboard ClusterIP 10.152.183.59 <none> 443/TCP 2035s k8s-app=kubernetes-dashboard kube-system dashboard-metrics-scraper ClusterIP 10.152.183.210 <none> 8080/TCP 2035s kds-app=dashboard-metrics-scr per Container-registry registry NodePort 10.152.183.239 <none> 5000:32000/TCP 2032s app=registry > per Container-registry registry NodePort 10.152.183.239 <none> 5000:32000/TCP 2032s app=registry
```

5. Install a web server on cluster



```
azuser@10727211-Ubuntu1604VM: ~
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions
Quick connect...
kube-system kubernetes-dashboard ClusterIP 10.152.183.59 <none> 443/TCP 2m32s k8s-app=kubernetes-dashboard
kube-system dashboard-metrics-scraper ClusterIP 10.152.183.210 <none> 8080/TCP 2m35s k8s-app=dashboard-metrics-scraper
/home/azuser/azuser@10727211-Ubuntu1604VM: ~$ sudo kubectl create deployment nginx --image=nginx
azuser@10727211-Ubuntu1604VM: ~$ sudo kubectl get deployments
deployment.apps/nginx created
azuser@10727211-Ubuntu1604VM: ~$ sudo kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx    0/1     0           0            34s
azuser@10727211-Ubuntu1604VM: ~$ sudo kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
azuser@10727211-Ubuntu1604VM: ~$ sudo kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
azuser@10727211-Ubuntu1604VM: ~$ azuser@10727211-Ubuntu1604VM: ~$
```

6. Test the website installation

```
azuser@10727211-Ubuntu1604VM: ~$ sudo kubectl get pods -o wide
NAME          READY  STATUS    RESTARTS   AGE   IP           NODE
nginx-6799fc8808-qwdwt  1/1  Running   0          90s  10.1.68.8  10727211-ubuntu1604vm <none>        <none>
azuser@10727211-Ubuntu1604VM: ~$ wget 10.1.68.8
Connecting to 10.1.68.8:443... connected.
HTTP request sent. Waiting response... 200 OK
Length: 615 [text/html]
Saving to: 'index.html'

index.html          100%[=====]   615 --KB/s   in 0s
2021-11-21 05:56:44 (107 MB/s) - 'index.html' saved [615/615]
```

7. Scale a web server deployment on a cluster

```

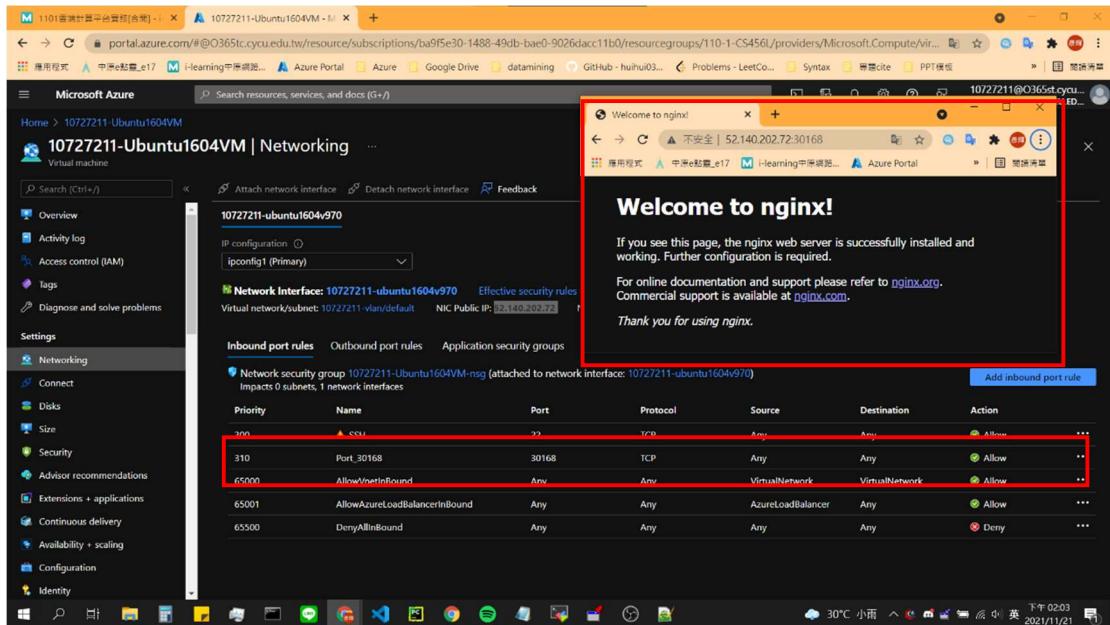
azureuser@10727211-Ubuntu16G4Vm:~$ sudo kubectl get all -o wide
NAME                                     READY   STATUS    RESTARTS   AGE     IP           NODE      NOMINATED-NODE  READINESS   GATES
pod/nginx-6799fc88d-4h9td    1/1     Running   0          5m45s   10.168.8.8   10727211-ubuntu16G4Vm  <none>        <none>
pod/nginx-6799fc88d-4h9td    1/1     Running   0          3m3s    10.168.9.8   10727211-ubuntu16G4Vm  <none>        <none>
pod/nginx-6799fc88d-25d12   1/1     Running   0          3m3s    10.168.10.8  10727211-ubuntu16G4Vm  <none>        <none>

NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE   SELECTOR
service/kubernetes  ClusterIP  10.152.181.1  <none>        443/TCP   11m  <none>
service/nginx     NodePort    10.152.181.222 <none>        80:30168/TCP 14s  app=nginx

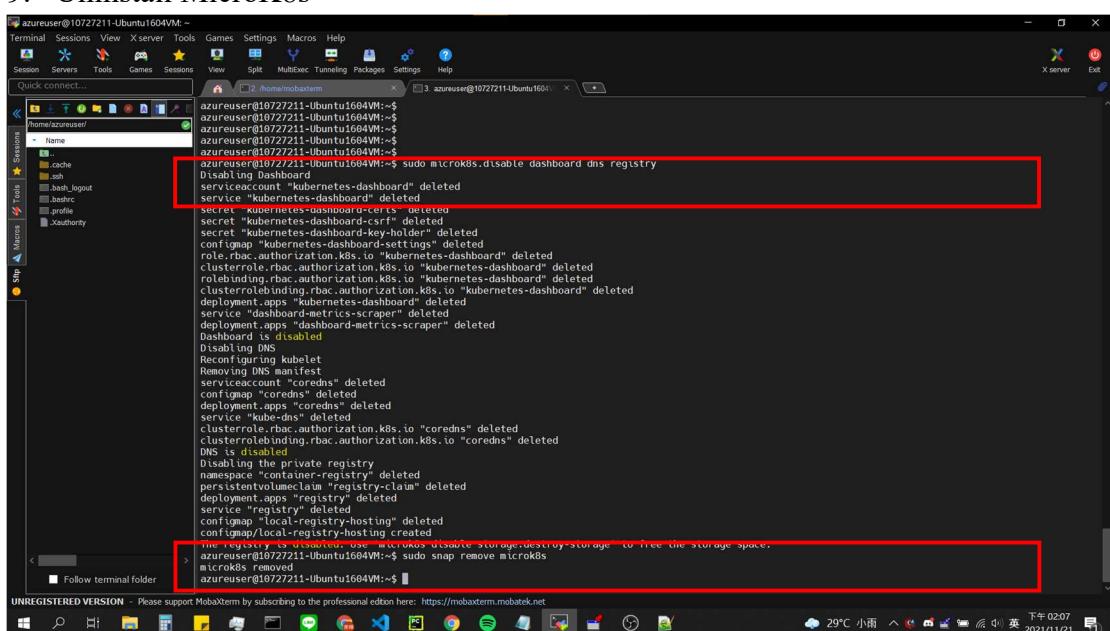
NAME            READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES   SELECTOR
deployment.apps/nginx  3/3     3           3           5m45s   nginx        nginx   app=nginx

NAME            DESIRED   CURRENT   READY   AGE     CONTAINERS   IMAGES   SELECTOR
replicaset.apps/nginx-6799fc88d8  3       3         3       5m45s   nginx        nginx   app=nginx,pod-template-hash=6799fc88d8
azureuser@10727211-Ubuntu16G4Vm:~$ 
```

8. Setting NSG & Test



9. Uninstall MicroK8s



Take screenshots of Badges and Trophies

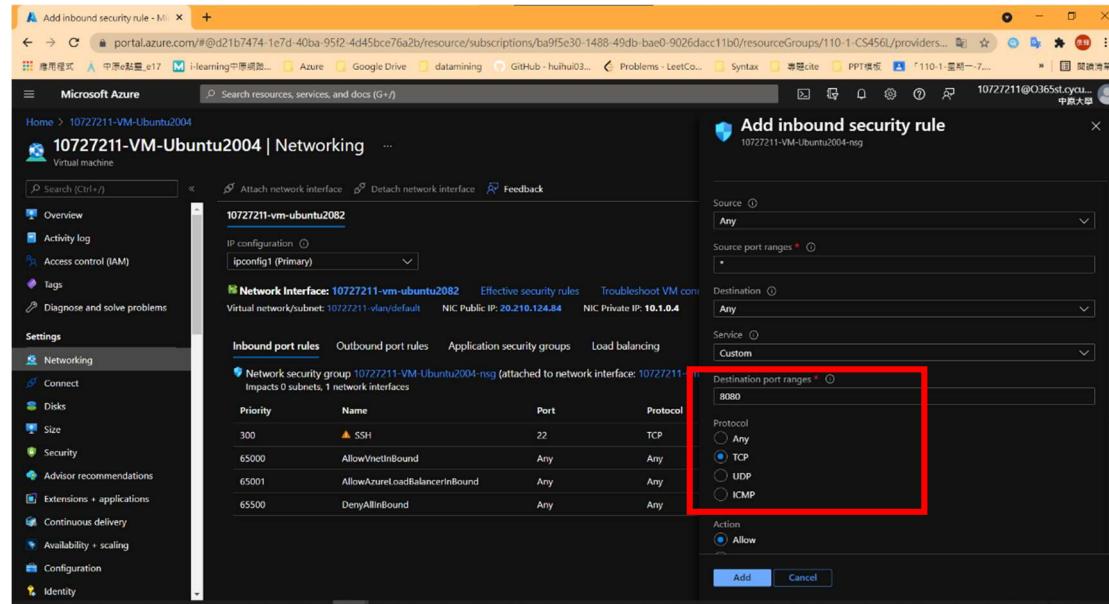
The screenshot shows the Microsoft Learn profile page for a user named Yan-Hui, Lin. At the top, there's a navigation bar with links to Microsoft Docs, Documentation, Learn, Q&A, and More. A search bar and a user profile icon are also present. Below the header, the user's profile picture and name are displayed, along with their email address: 10727211@O365st.cycu.edu.tw. A progress bar indicates the user is at LEVEL 6, with 29850 XP earned out of 32299 total. Below the progress bar, statistics are shown: 22 Badges, 2 Trophies, 1 Reputation, 0 Answers accepted, 0 Following, and 0 Followers accepted. The main content area is divided into sections: Activity, Trophies, Achievements, and Badges. The 'Achievements' section is currently selected and displays two completed achievements: 'Administer containers in Azure' (Completed on 11/21/2021) and 'Azure Fundamentals part 1: Describe core Azure concepts' (Completed on 9/18/2021). The 'Badges' section displays two completed badges: 'Introduction to Kubernetes' (Completed on 11/21/2021) and 'Deploy and run a containerized web app with Azure App Service' (Completed on 11/21/2021).

Learned from the Learning Path

在 Model1 簡單介紹了容器化的技術，透過容器化的技術可以使部署更加快速，相對起 VM，容器可以透過 Image 指定所需的環境，部署的步驟也相當簡易，使用容器化的技術可以減少資源的浪費。Model2-5 開始透過實作安裝 Docker，透過 Sample Code 簡單教導部署的流程，從中了解容器的軟體生命週期，更從中看到 Azure 相關的資源：Container Registry、Container Instances，在這些 Model 看到更多的範例教導我們如何使用這些資源，無論是 GUI 介面或是 CLI 介面，讓使用者學習更多不僅僅是容器的知識，更藉由 Azure 相關資源提供更多優秀的服務。最後提到了 AKS，從中學習到簡單 Cluster 的部署方式，但可惜此 Model 中使用 MicroK8s 進行實作，真正 AKS 的實作未包含在此學習路徑上。

3. Problems

整體實作上無太大困難，僅需要時間適應 Azure Portal 的介面，練習多了就能適應 Azure 的部署機制。在 NSG 在設定上曾經有出現障礙，下圖為其中一個例子，在網頁需要 Port-forward 的條件下，Protocol 的區塊需要額外指定‘TCP’，若選擇預設的‘Any’則無法成功 Port-Forward。



4. FeedBack

儘管這是一個相當基礎的 Model，但仍希望 Microsoft 官方可以針對此 Learning Path 提供 SandBox，否則針對想學習容器技術的初學者而言可能是個不友善的環境，其餘部分包含解說精細、範例程式都能正常 work，是個相當適合新手的教學。