Yakubyshyn Anatolii

Containerization and Orchestration

Practical Task 1: Deploy a Docker Container to Azure Container Instances (ACI) via Azure Portal

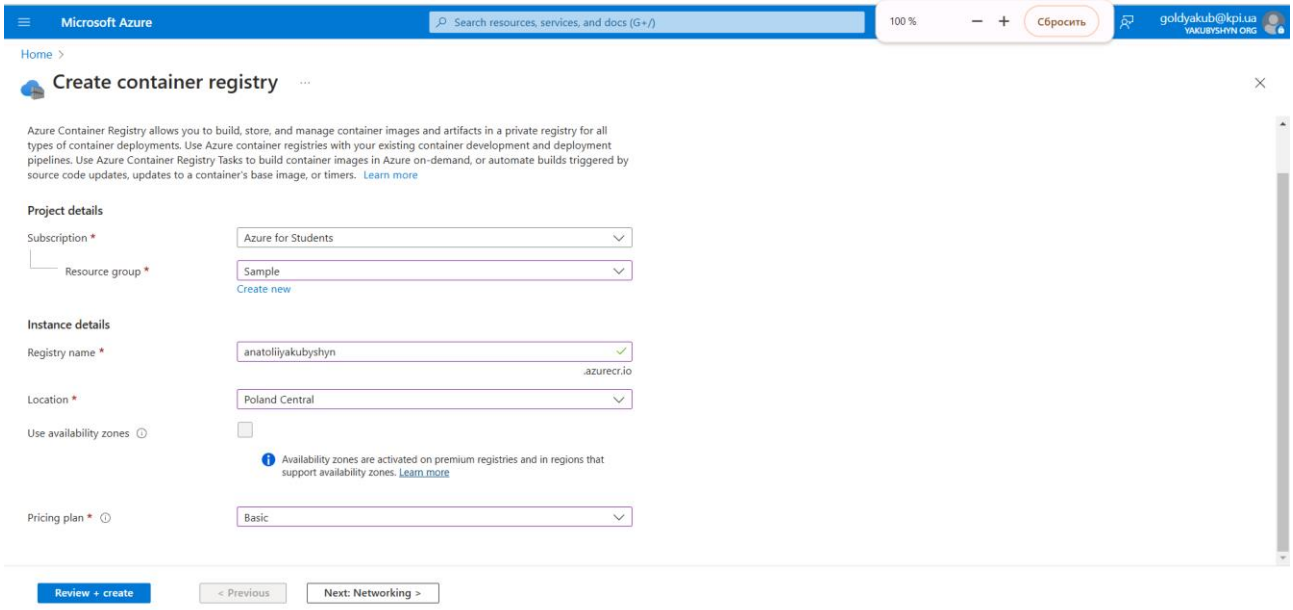
1. Create a lightweight Docker image for a simple web application (e.g., a Python Flask app) with minimal dependencies to reduce resource usage.

Dockerfile

```
FROM adoptopenjdk:11-jre-hotspot-focal@sha256:eac1c6cff5fded2dd35fc94bb23e7862a08277bd71f9b352a99df5bc740459c3

RUN apt update
RUN apt-get install firefox -y


ARG JAR_FILE=target/back-end-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} back.jar
CMD ["java","-jar","/back.jar"]
EXPOSE 8080

```

Docker image for Silpo price parser

2. Push the Docker image to Azure Container Registry (ACR) using a low-cost storage option.



Microsoft Azure

Home >

Create container registry

Running final validation

Basics | Networking | Encryption | Tags | **Review + create**

Registry details

Basics

Registry name	anatoliiyakubyshyn
Subscription	Azure for Students
Resource Group	Sample
Location	Poland Central
Availability zones	Disabled
Pricing plan	Basic

Networking

Public network access	Yes

Encryption

Customer-Managed Key	Disabled
Identity	None
Key Vault	None
Encryption key	None
Version	None

```
PS D:\Azure\task4\ButterAndBread-main\rest> docker build . -t silpoparser
[+] Building 166.1s (9/9) FINISHED                                                              docker:desktop-linu
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 317B
 => [internal] load metadata for docker.io/library/adoptopenjdk:11-jre-hotspot-focal@sha256:eac1c6cff5fded2dd35fc94bb23e7862a08277bd71f9b352a99df5bc740459c3
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/adoptopenjdk:11-jre-hotspot-focal@sha256:eac1c6cff5fded2dd35fc94bb23e7862a08277bd71f9b352a99df5bc740459c3
 => => resolve docker.io/library/adoptopenjdk:11-jre-hotspot-focal@sha256:eac1c6cff5fded2dd35fc94bb23e7862a08277bd71f9b352a99df5bc740459c3
 => => sha256:eac1c6cff5fded2dd35fc94bb23e7862a08277bd71f9b352a99df5bc740459c3 953B / 953B
 => => sha256:2c57fb3bc67b729194ef8e9542c2c8b66da35e790584faca402273f084e5193b 4.72kB / 4.72kB
 => => sha256:f3ef4ff62e0da0ef761ec1c8a578f3035bef51043e53ae1b13a20b3e03726d17 28.57MB / 28.57MB
 => => sha256:706b9b9c1c443d1cd3d83fa6ecf9e3a4ae13b711dcc6b3aa34e9603c1cb8a930 16.03MB / 16.03MB
 => => sha256:76205aac4d5a82ee8fb973ad9341726fd98259a7c0aed9e47c173ae3a194931f 43.16MB / 43.16MB
 => => extracting sha256:f3ef4ff62e0da0ef761ec1c8a578f3035bef51043e53ae1b13a20b3e03726d17
 => => extracting sha256:706b9b9c1c443d1cd3d83fa6ecf9e3a4ae13b711dcc6b3aa34e9603c1cb8a930
 => => extracting sha256:76205aac4d5a82ee8fb973ad9341726fd98259a7c0aed9e47c173ae3a194931f
 => [internal] load build context
 => => transferring context: 50.93MB
 => [2/4] RUN apt update
 => [3/4] RUN apt-get install firefox -y
 => [4/4] COPY target/back-end-0.0.1-SNAPSHOT.jar back.jar
 => exporting to image
 => => exporting layers
 => => writing image sha256:862936a21937574fa2a412d276ad672ba1c0cbdec9d3575bf415dfc69e466694
 => => naming to docker.io/library/silpoparser

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/kjcklsgcx8yxpqrqr1njy68v7

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Then installed azure cli

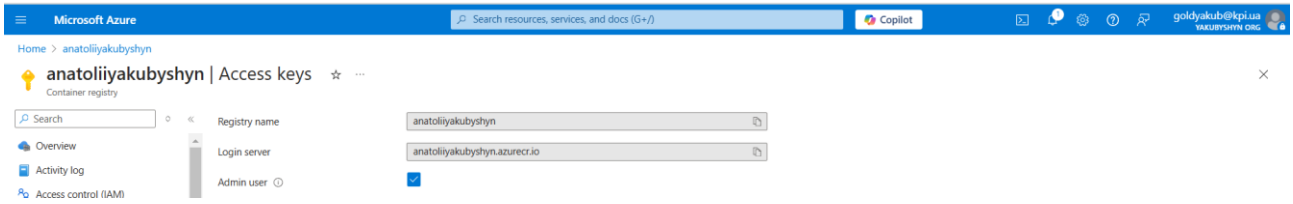
```
PS C:\Users\Anatoly> az login --tenant be098e76-f2f0-41f0-8292-f151f67b6729
Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136

Retrieving subscriptions for the selection...

[Tenant and subscription selection]

No     Subscription name    Subscription ID                       Tenant
-----  -------------------  ------------------------------------  ------------------------------------
[1] *  Azure for Students   3a612e70-8e22-4425-b3ea-29f6acf32428  be098e76-f2f0-41f0-8292-f151f67b6729

The default is marked with an *; the default tenant is 'be098e76-f2f0-41f0-8292-f151f67b6729' and subscription is 'Azure for Students' (3a612e70-8e22-4425-b3ea-29f6acf32428).

Select a subscription and tenant (Type a number or Enter for no changes):

Tenant: be098e76-f2f0-41f0-8292-f151f67b6729
Subscription: Azure for Students (3a612e70-8e22-4425-b3ea-29f6acf32428)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.
```

```
PS C:\Users\Anatoly> az acr login --name anatoliiyakubyshyn.azurecr.io
The login server endpoint suffix '.azurecr.io' is automatically omitted.
Login Succeeded
PS C:\Users\Anatoly> docker push anatoliiyakubyshyn.azurecr.io/silpoparser
Using default tag: latest
The push refers to repository [anatoliiyakubyshyn.azurecr.io/silpoparser]
92c863acf07a: Pushed
488882c9d87e: Pushed
07b79d4282a0: Pushed
83b767b06655: Pushed
14fbd8039ba4: Pushed
da55b45d310b: Pushed
latest: digest: sha256:29e38ac4f0ccf908d42177eb67ede0e628c110d3af453563a5ecbf931cb42d34 size: 1590
PS C:\Users\Anatoly>
```

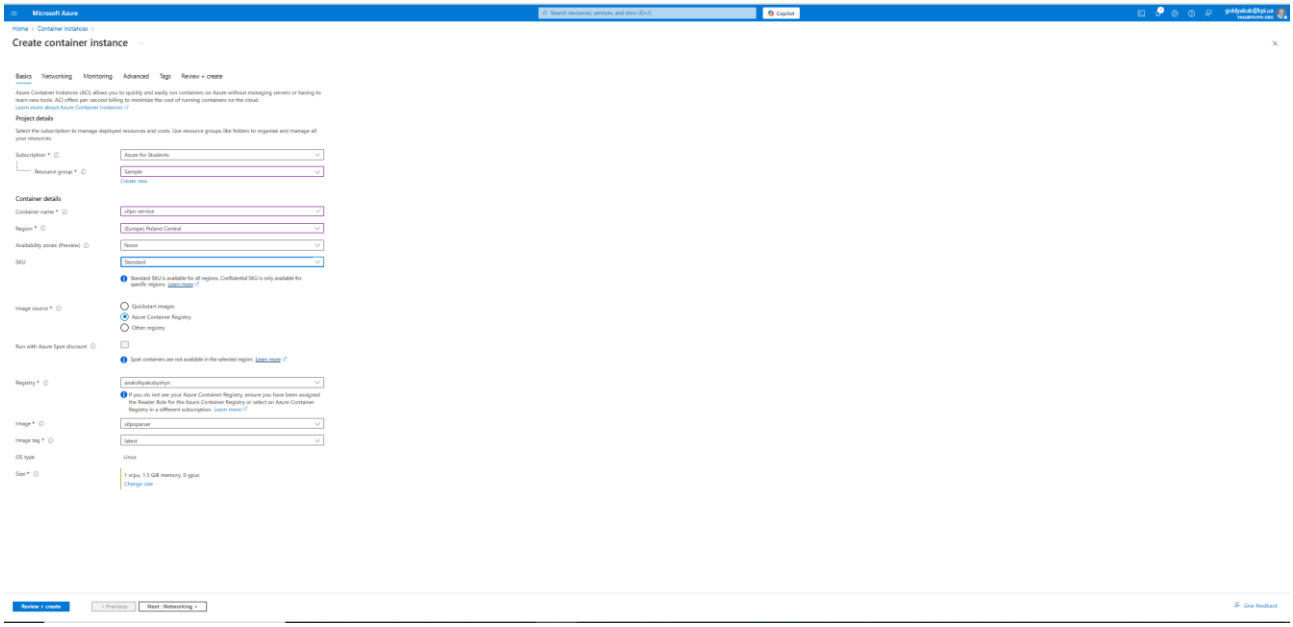
```
PS C:\Users\Anatoly> docker tag silpoparser anatoliiyakubyshyn.azurecr.io/silpoparser
PS C:\Users\Anatoly>
```

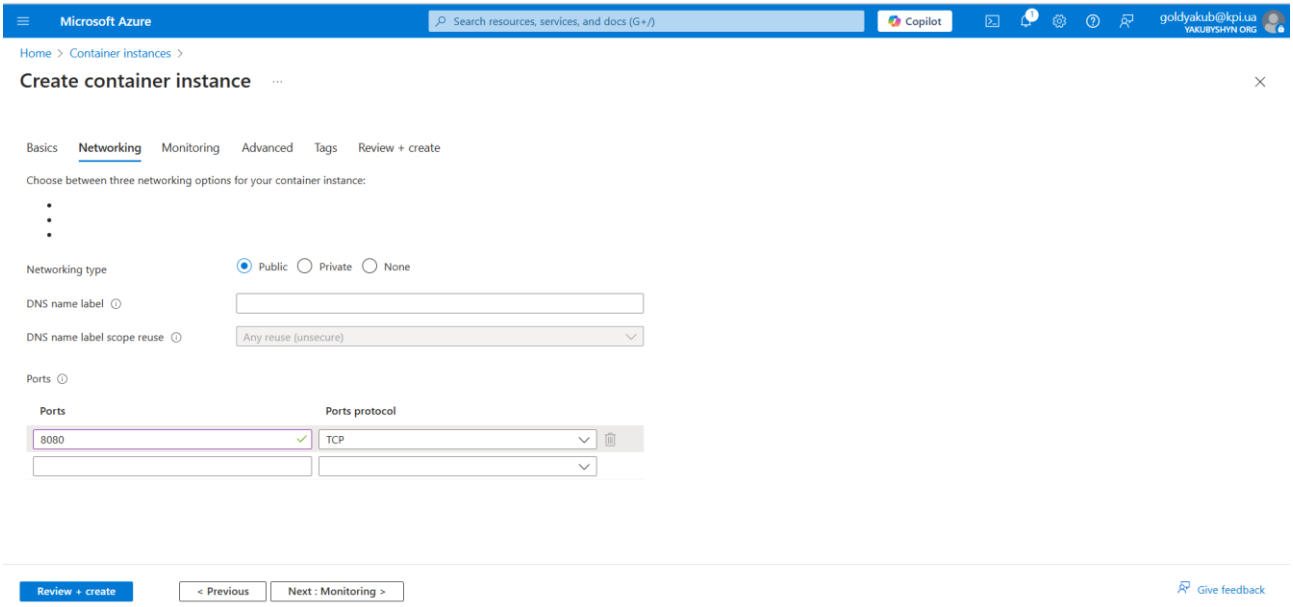
```
PS C:\Users\Anatoly> az acr login --name anatoliiyakubyshyn.azurecr.io
The login server endpoint suffix '.azurecr.io' is automatically omitted.
Login Succeeded
PS C:\Users\Anatoly> docker push anatoliiyakubyshyn.azurecr.io/silpoparser
Using default tag: latest
The push refers to repository [anatoliiyakubyshyn.azurecr.io/silpoparser]
92c863acf07a: Pushed
488882c9d87e: Pushed
07b79d4282a0: Pushed
83b767b06655: Pushed
14fbd8039ba4: Pushed
da55b45d310b: Pushed
latest: digest: sha256:29e38ac4f0ccf908d42177eb67ede0e628c110d3af453563a5ecbf931cb42d34 size: 1590
PS C:\Users\Anatoly>
```

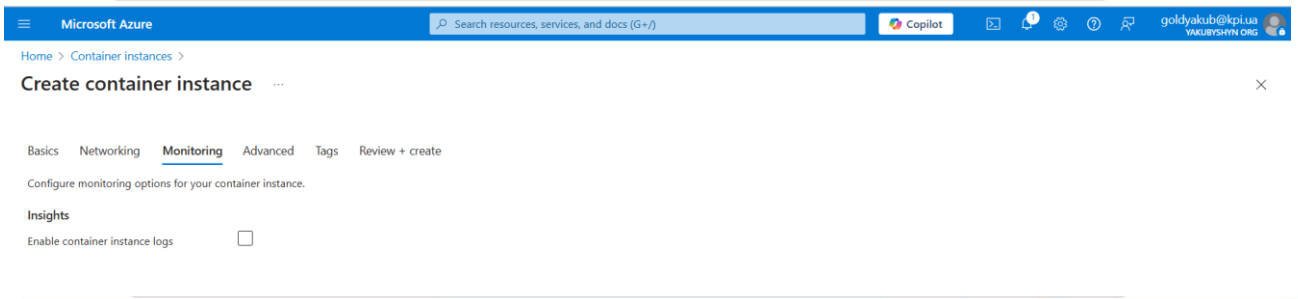
3. Use a lightweight ACI instance (e.g., B1s) to deploy the Docker container from ACR.

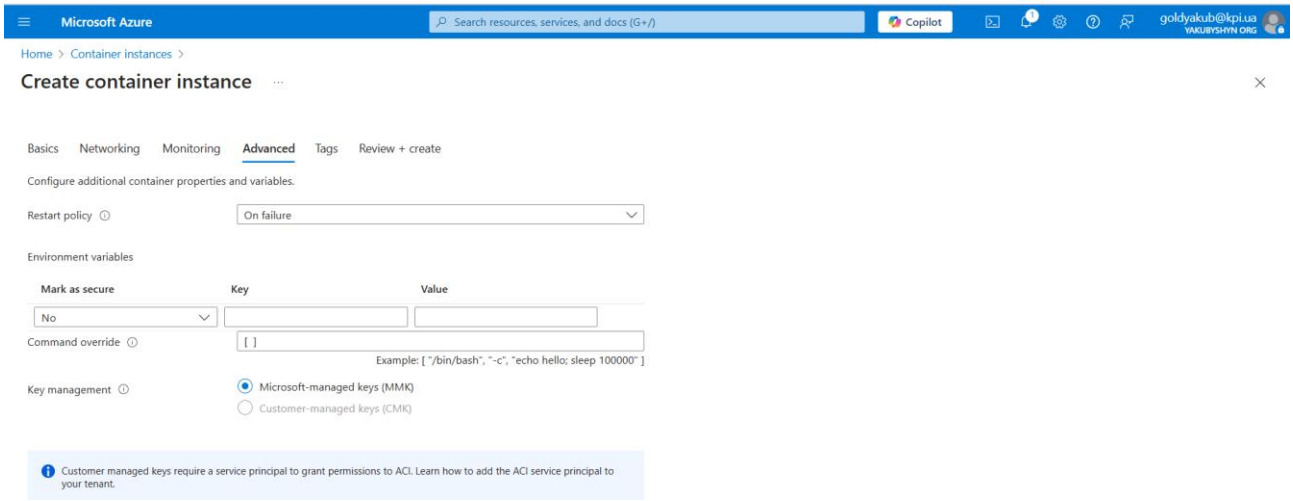


Enabled admin user




Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
goldyakub@kpi.ua
YAKUBYSHYN ORG
Home > Container instances >
Create container instance
Basics Networking Monitoring Advanced Tags Review + create
Choose between three networking options for your container instance:
Networking type
Public Private None
DNS name label
DNS name label scope reuse
Any reuse (unsecure)
Ports
Ports
Ports protocol
8080
TCP
Review + create
< Previous
Next : Monitoring >
Give feedback


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
goldyakub@kpi.ua
YAKUBYSHYN ORG
Home > Container instances >
Create container instance
Basics Networking Monitoring Advanced Tags Review + create
Configure monitoring options for your container instance.
Insights
Enable container instance logs

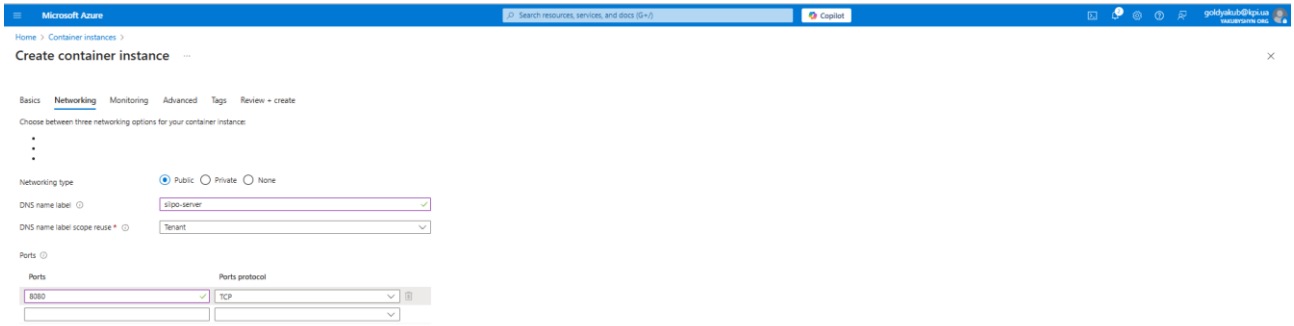

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
goldyakub@kpi.ua
YAKUBYSHYN ORG
Home > Container instances >
Create container instance
Basics Networking Monitoring Advanced Tags Review + create
Configure additional container properties and variables.
Restart policy
On failure
Environment variables
Mark as secure
Key
Value
No
Command override
[ ]
Example: ["/bin/bash", "-c", "echo hello; sleep 100000"]
Key management
Microsoft-managed keys (MMK)
Customer-managed keys (CMK)
Customer managed keys require a service principal to grant permissions to ACI. Learn how to add the ACI service principal to your tenant.

4. Verify the deployment by accessing the web application via the public IP address provided by ACI.

At first, I fixed bug in code.

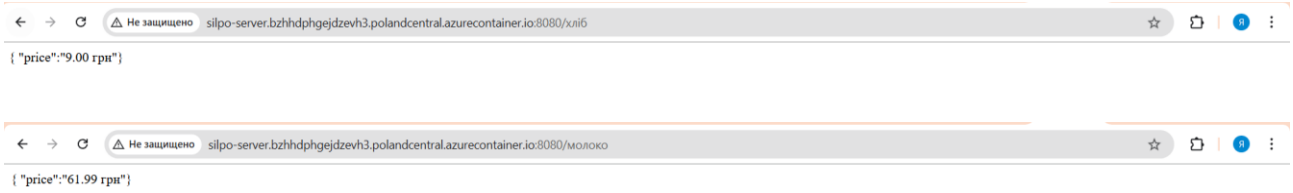
And pushed again to the ACR

Restarted container instance

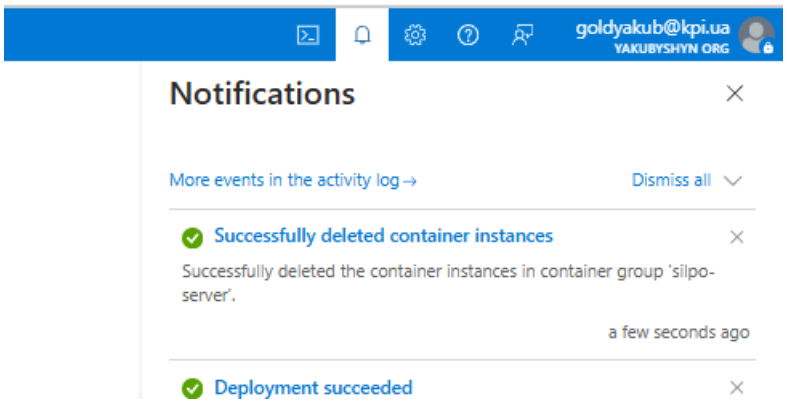


creating ACI again

enabled logs this time and DNS name



5. Remove the ACI container after verifying the deployment to stop billing.



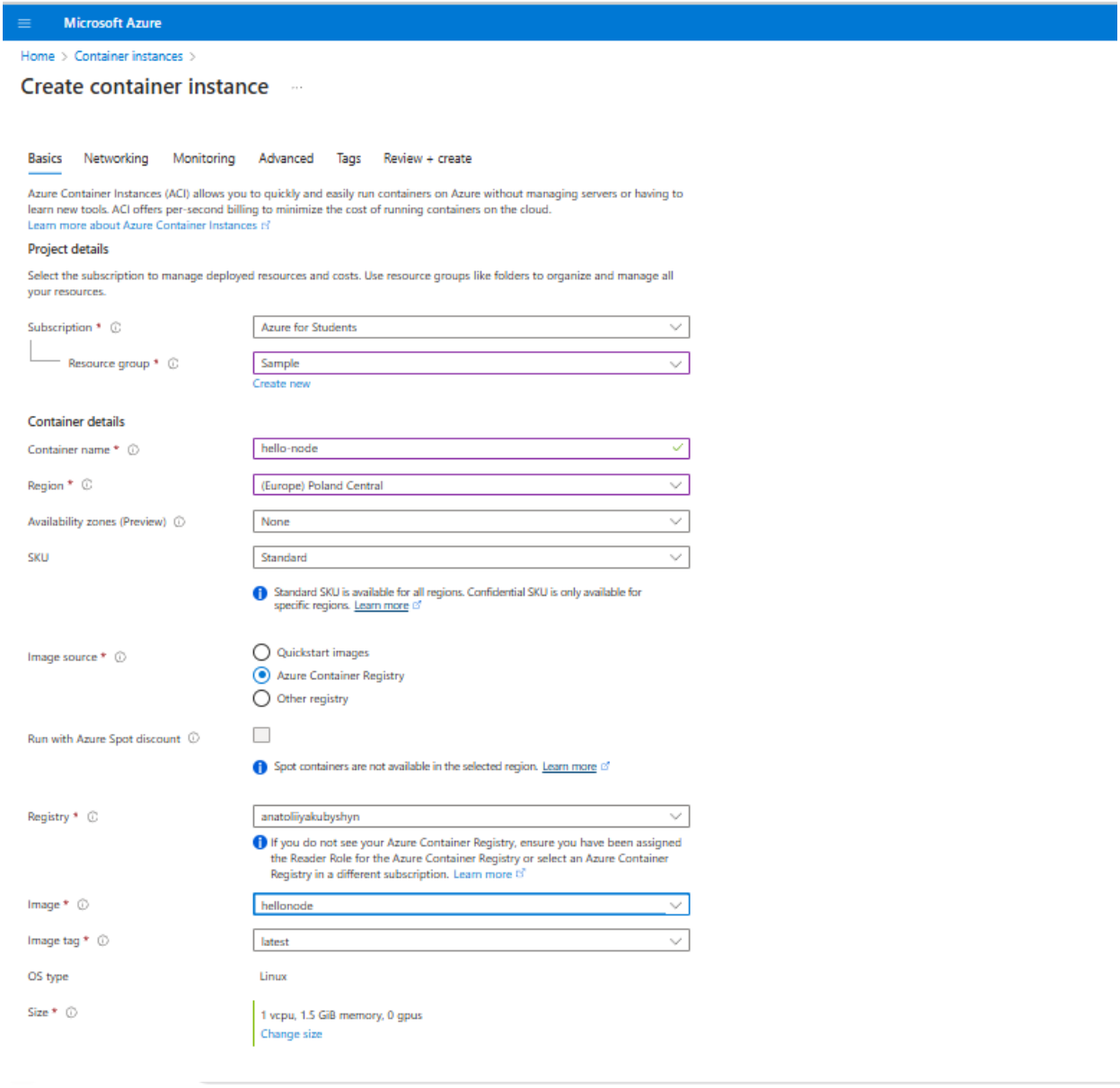
Practical Task 2: Configure Environment Variables in ACI via Azure Portal

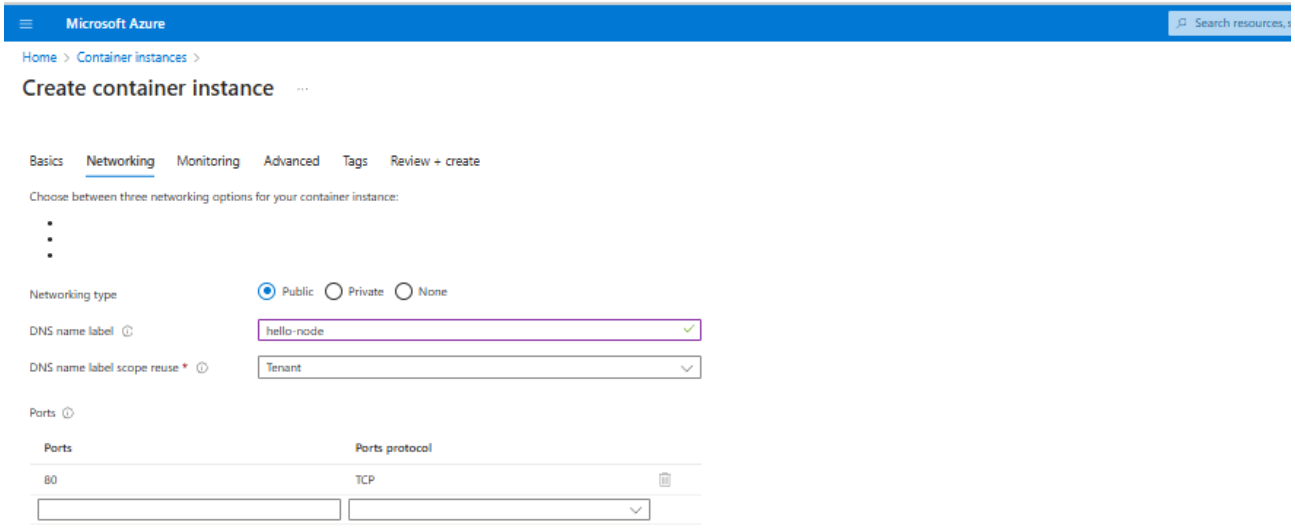
```
# Use official Node.js image
FROM node:18

# Set working directory
WORKDIR /app

# Copy package.json and install dependencies
COPY package.json package-lock.json ./
RUN npm install

# Copy app files
COPY main.js .

# Expose port
EXPOSE 80

# Set default environment variable
ENV NAME="Docker User"

# Run the app
CMD ["node", "main.js"]
```

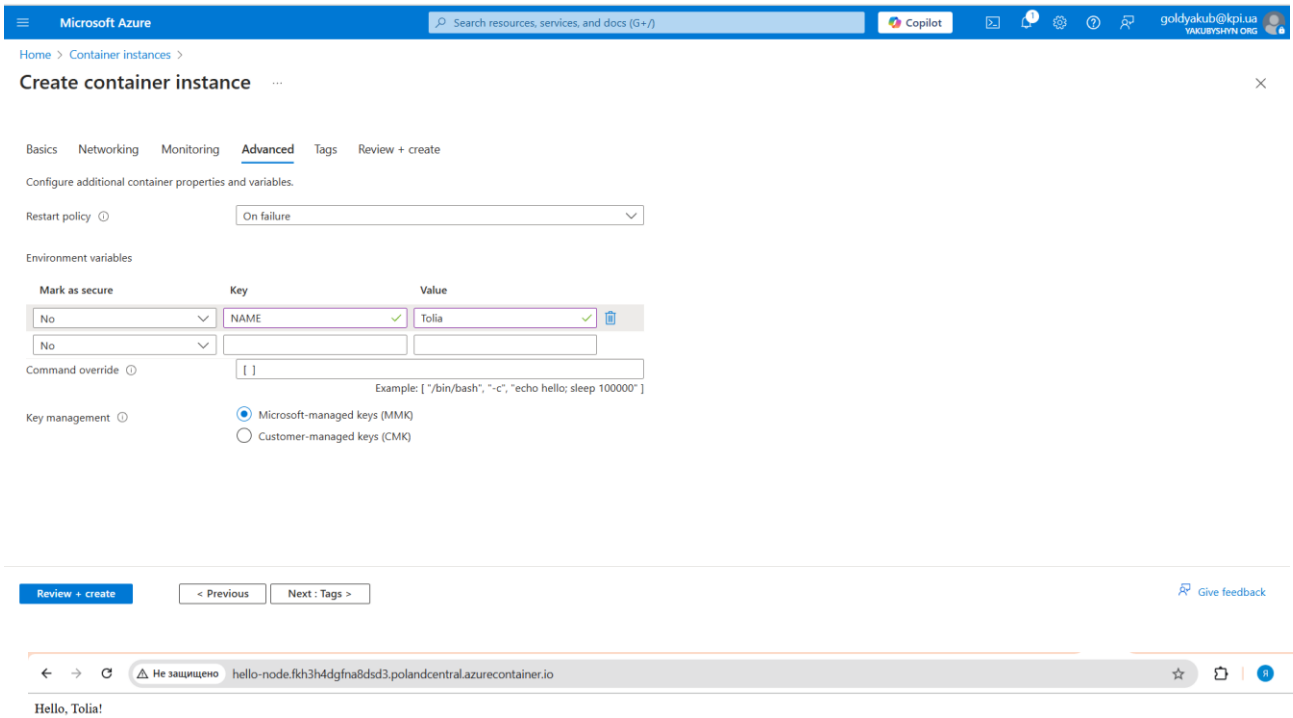
```
const express = require("express");
const app = express();

const PORT = process.env.PORT || 80;
const NAME = process.env.NAME || "World";

app.get("/", (req, res) => {
    res.send(`Hello, ${NAME}!`);
});

app.listen(PORT, () => {
    console.log(`Server running on port ${PORT}`);
});
```

```
PS D:\Azure\task4\app2> docker build -t hellonode .
[+] Building 154.9s (10/10) FINISHED                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                  0.1s
 => => transferring dockerfile: 397B                                                                  0.0s
 => [internal] load metadata for docker.io/library/node:18                                            2.0s
 => [internal] load .dockerignore                                                                     0.1s
 => => transferring context: 2B                                                                       0.0s
 => [1/5] FROM docker.io/library/node:18@sha256:720eeea325b3da50e108ba34dde0fd69feeb3c59485199c5e22b0ea49a792aa5   133.3s
 => => resolve docker.io/library/node:18@sha256:720eeea325b3da50e108ba34dde0fd69feeb3c59485199c5e22b0ea49a792aa5   0.1s
 => => sha256:2880d96377d4a21f8684ebf7c3549a7b2c07ca93ee1c3196cf862861e97930c1 2.49kB / 2.49kB       0.0s
 => => sha256:c0d22c846dfba5c68261e7acaea5596f6a12a32802ed0645254e31e5bb9297d5 6.39kB / 6.39kB       0.0s
```

```
PS C:\Users\Anatoly> docker tag hellonode anatoliiyakubyshyn.azurecr.io/hellonode
PS C:\Users\Anatoly> docker push anatoliiyakubyshyn.azurecr.io/hellonode
Using default tag: latest
The push refers to repository [anatoliiyakubyshyn.azurecr.io/hellonode]
74362c881faa: Pushed
fd975ca59459: Pushed
231ea03fb497: Pushed
b41cdddf76f7: Pushed
207fd040c49e: Pushed
0149b9feb6a3: Pushed
7e517a081f3d: Pushed
d3244fce0fd3: Pushed
ebad64620a59: Pushed
f379f6005525: Pushed
0e5c23e041ee: Pushed
397f1b2e2505: Pushed
latest: digest: sha256:6fac5f5995f4a73dd95bd1e437bfdff763639b67f62a26745f16cde5380af824 size: 2836
PS C:\Users\Anatoly>
```


Microsoft Azure
Home > Container instances >
Create container instance
Basics Networking Monitoring Advanced Tags Review + create
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 per-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 * Sample
Create new
Container details
Container name * hello-node
Region * (Europe) Poland Central
Availability zones (Preview) None
SKU Standard
Standard SKU is available for all regions. Confidential SKU is only available for specific regions. Learn more
Image source * Quickstart images
Azure Container Registry
Other registry
Run with Azure Spot discount
Spot containers are not available in the selected region. Learn more
Registry * anatoliiyakubyshyn
If you do not see your Azure Container Registry, ensure you have been assigned the Reader Role for the Azure Container Registry or select an Azure Container Registry in a different subscription. Learn more
Image * hellonode
Image tag * latest
OS type Linux
Size * 1 vcpu, 1.5 GiB memory, 0 gpus
Change size


Microsoft Azure
Search resources,
Home > Container instances >
Create container instance
Basics Networking Monitoring Advanced Tags Review + create
Choose between three networking options for your container instance:
Networking type Public Private None
DNS name label hello-node
DNS name label scope reuse * Tenant
Ports
Ports Ports protocol
80 TCP

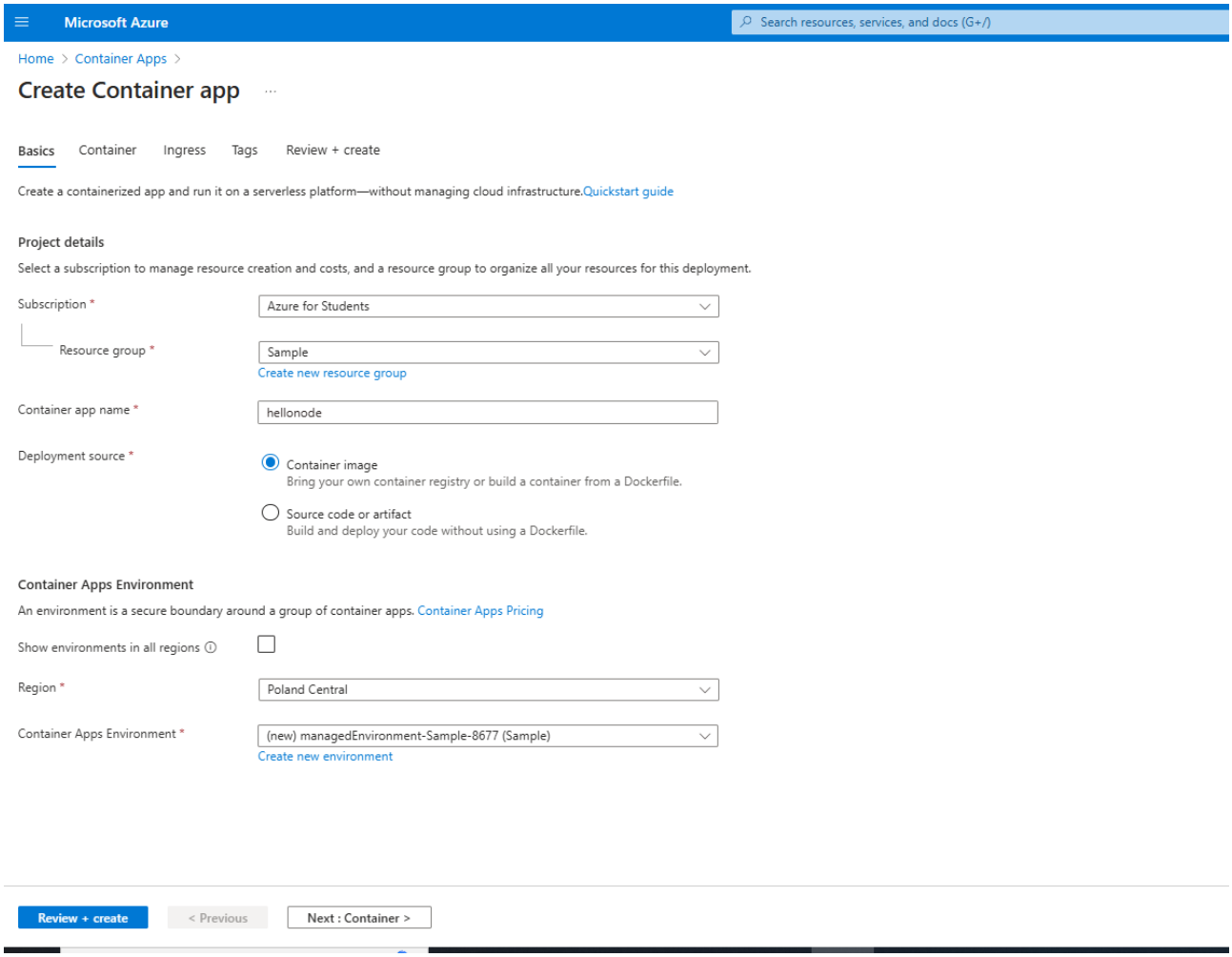

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
goldyakub@kpi.ua
YAKUBYSHYN ORG
Home > Container instances >
Create container instance
Basics Networking Monitoring Advanced Tags Review + create
Configure additional container properties and variables.
Restart policy
On failure
Environment variables
Mark as secure
Key
Value
No
NAME
Tolia
No
Command override
[ ]
Example: ["/bin/bash", "-c", "echo hello; sleep 100000"]
Key management
Microsoft-managed keys (MMK)
Customer-managed keys (CMK)
Review + create
< Previous
Next : Tags >
Give feedback

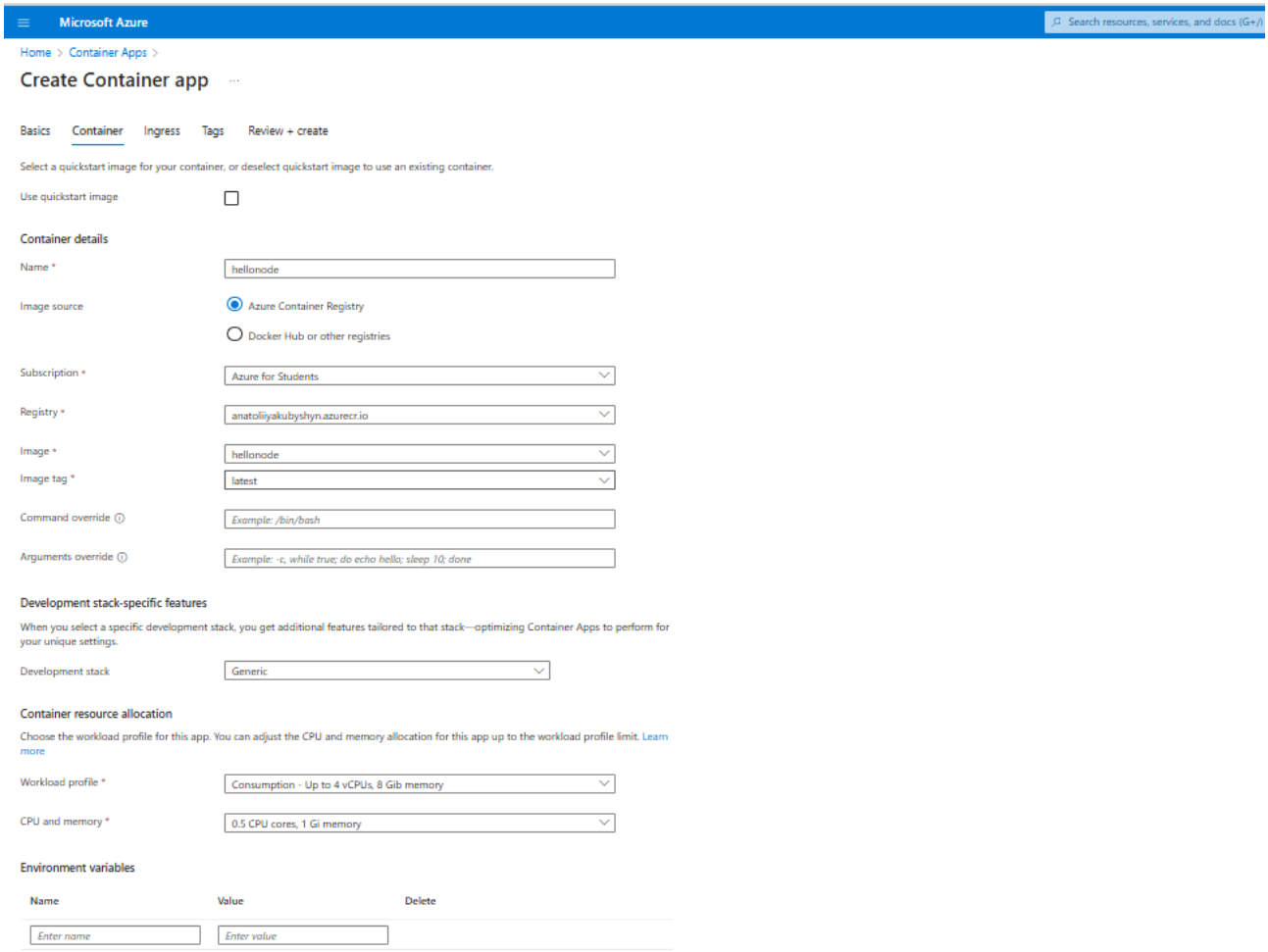
Не защищено
hello-node.fkh3h4dgfna8dsd3.polandcentral.azurecontainer.io

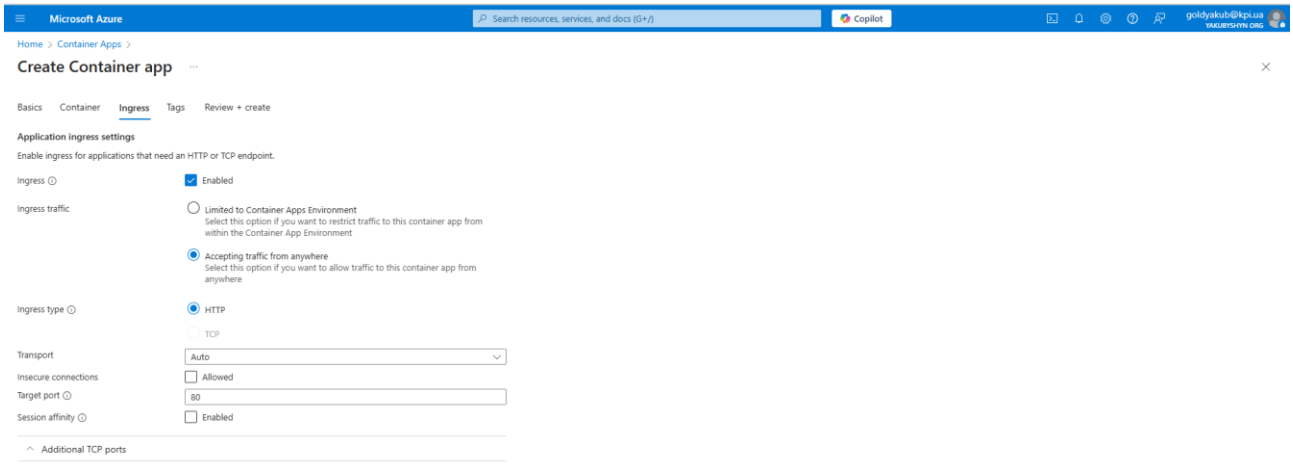
Hello, Tolia!

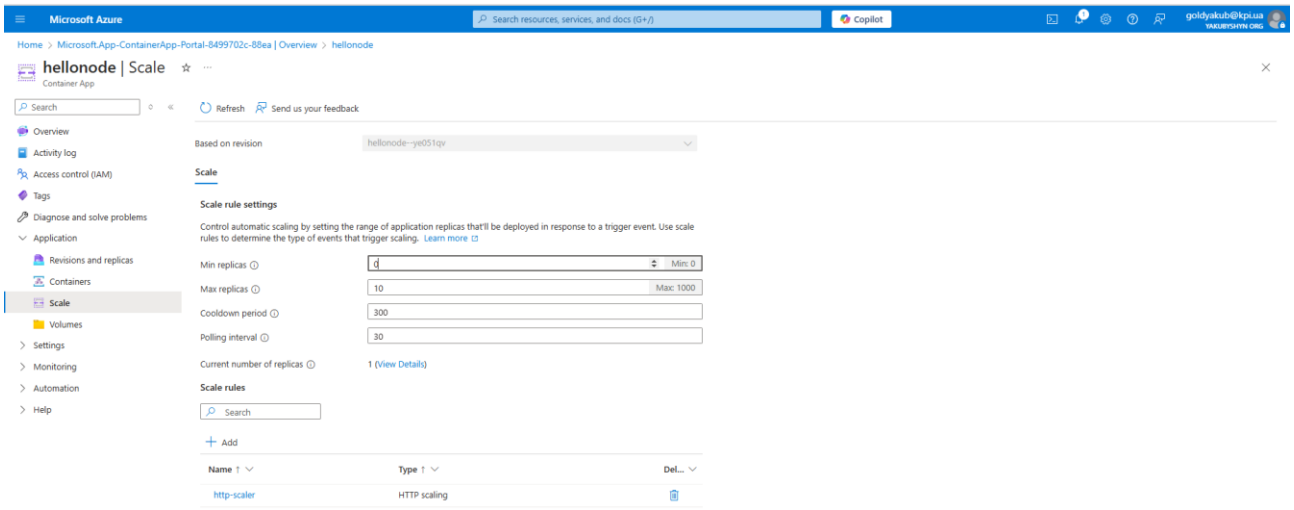
Practical Task 3: Scale Out with Azure Container Instances via Azure Portal

I suppose here it is a typo -> Azure Container Instances -> **Azure Container Apps**




Microsoft Azure
Search resources, services, and docs (G+/)
Home > Container Apps >
Create Container app
Basics Container Ingress Tags Review + create
Select a quickstart image for your container, or deselect quickstart image to use an existing container.
Use quickstart image
Container details
Name *
hellonode
Image source
Azure Container Registry
Docker Hub or other registries
Subscription *
Azure for Students
Registry *
anatoliiyakubyshyn.azurecr.io
Image *
hellonode
Image tag *
latest
Command override
Example: /bin/bash
Arguments override
Example: -c, while true; do echo hello; sleep 10; done
Development stack-specific features
When you select a specific development stack, you get additional features tailored to that stack—optimizing Container Apps to perform for your unique settings.
Development stack
Generic
Container resource allocation
Choose the workload profile for this app. You can adjust the CPU and memory allocation for this app up to the workload profile limit. Learn more
Workload profile *
Consumption - Up to 4 vCPUs, 8 Gib memory
CPU and memory *
0.5 CPU cores, 1 Gi memory
Environment variables
Name
Value
Delete
Enter name
Enter value


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
goldyakub@kpi.ua
YAKUBYSHYN.ORG
Home > Container Apps >
Create Container app
Basics Container Ingress Tags Review + create
Application ingress settings
Enable ingress for applications that need an HTTP or TCP endpoint.
Ingress
Enabled
Ingress traffic
Limited to Container Apps Environment
Select this option if you want to restrict traffic to this container app from within the Container App Environment
Accepting traffic from anywhere
Select this option if you want to allow traffic to this container app from anywhere
Ingress type
HTTP
TCP
Transport
Auto
Insecure connections
Allowed
Target port
80
Session affinity
Enabled
Additional TCP ports



Refresh Send us your feedback

Based on revision hellonode--ye051qv

Scale

Scale rule settings

Control automatic scaling by setting the range of application replicas that'll be deployed in response to a trigger event. Use scale rules to determine the type of events that trigger scaling. Learn more

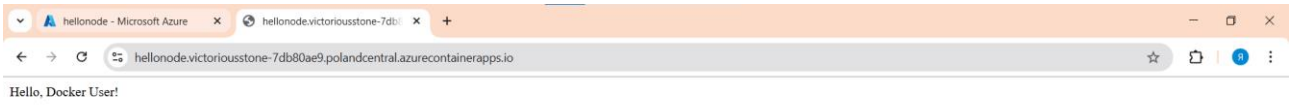
Min replicas	1	Min: 0
Max replicas	3	Max: 1000
Cooldown period	300	
Polling interval	30	
Current number of replicas	1 (View Details)	

Scale rules

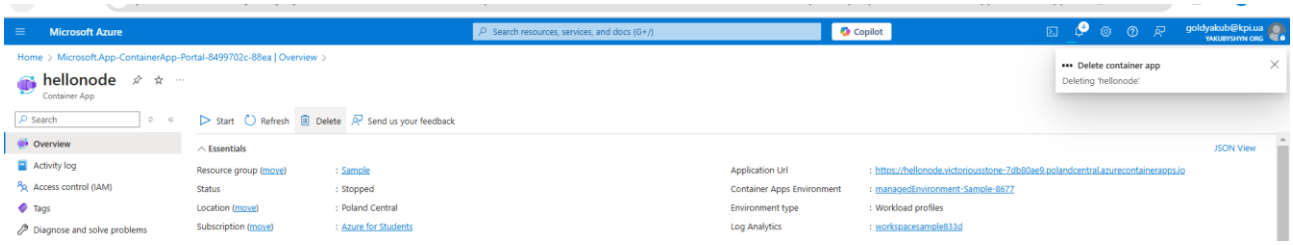
Search

Add

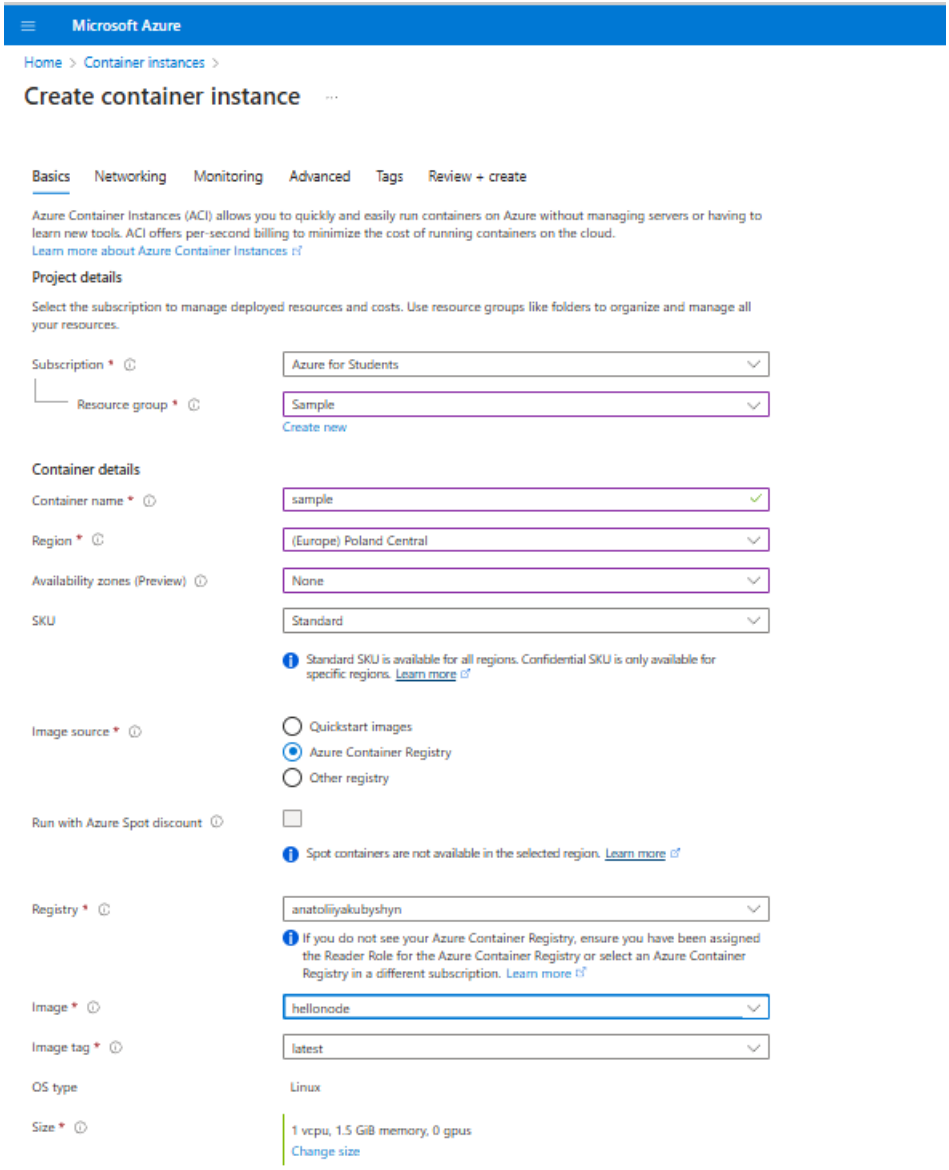
Name	Type	Del...
http-scaler	HTTP scaling	

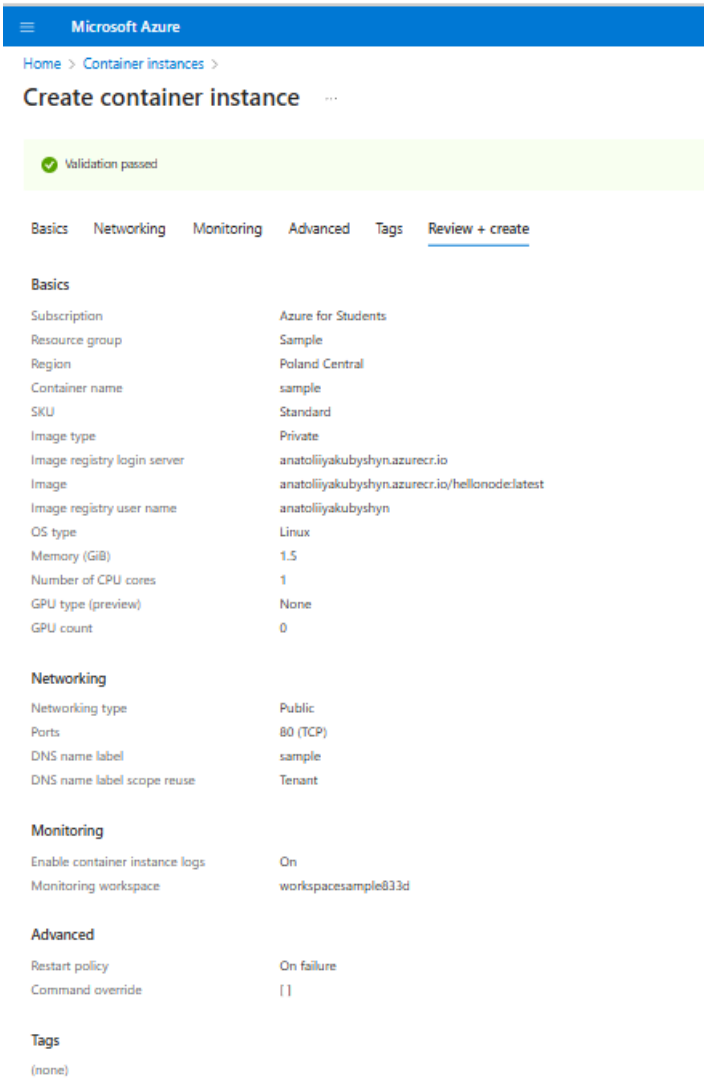


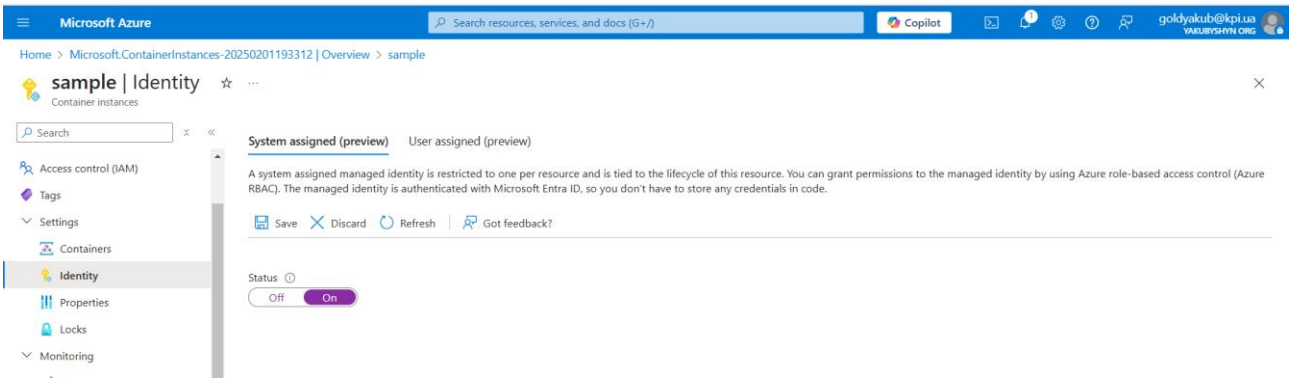
(Default parameter for name)

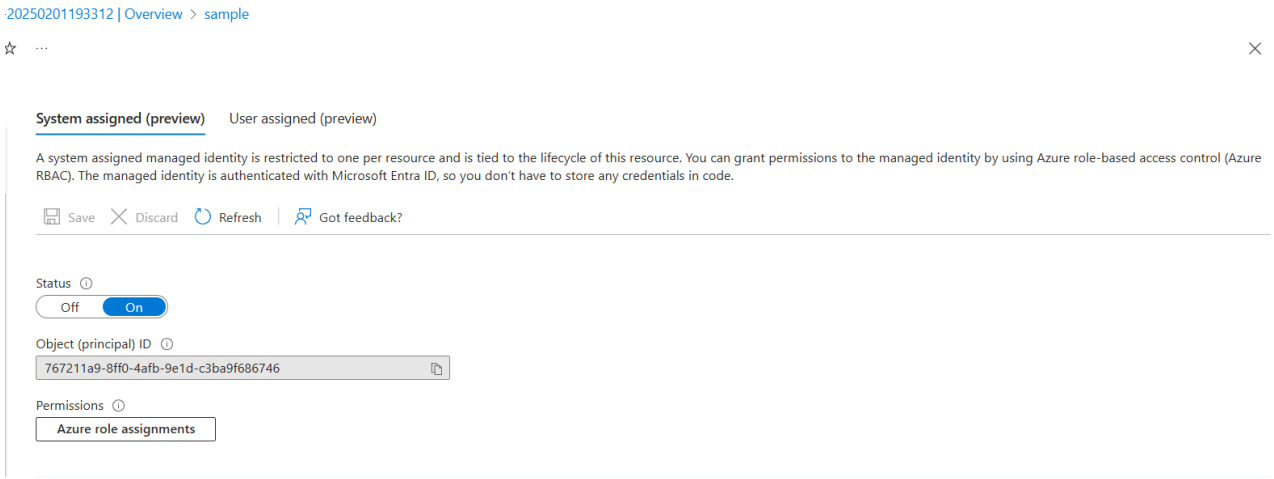


Practical Task 4: Secure a Docker Container in ACI with Managed Identity via Azure Portal

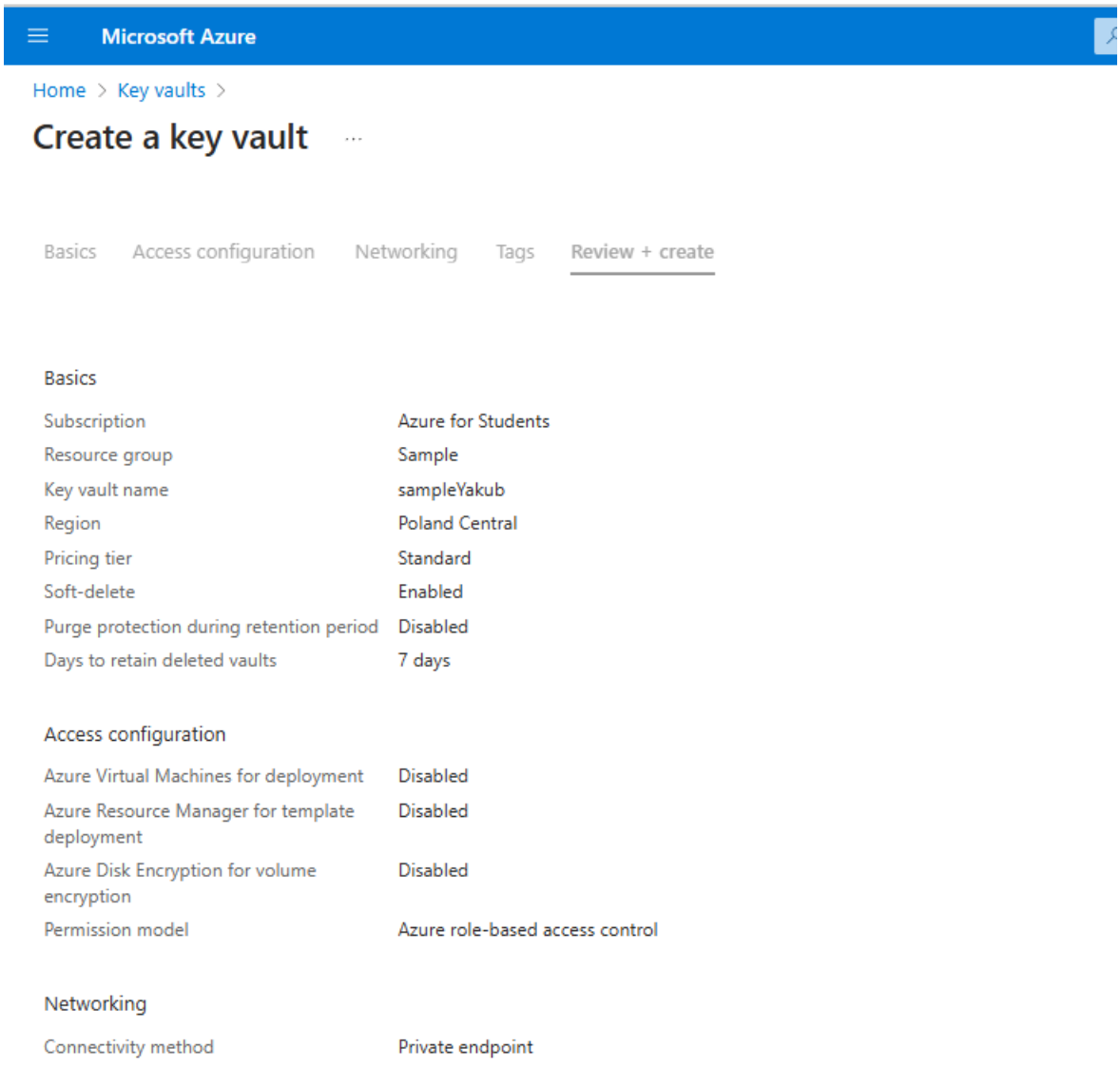



Microsoft Azure
Home > Container instances >
Create container instance
Validation passed
Basics Networking Monitoring Advanced Tags Review + create
Basics
Subscription Azure for Students
Resource group Sample
Region Poland Central
Container name sample
SKU Standard
Image type Private
Image registry login server anatoliiyakubyshyn.azurecr.io
Image anatoliiyakubyshyn.azurecr.io/hellonode:latest
Image registry user name anatoliiyakubyshyn
OS type Linux
Memory (GiB) 1.5
Number of CPU cores 1
GPU type (preview) None
GPU count 0
Networking
Networking type Public
Ports 80 (TCP)
DNS name label sample
DNS name label scope reuse Tenant
Monitoring
Enable container instance logs On
Monitoring workspace workspacesample833d
Advanced
Restart policy On failure
Command override [ ]
Tags
(none)

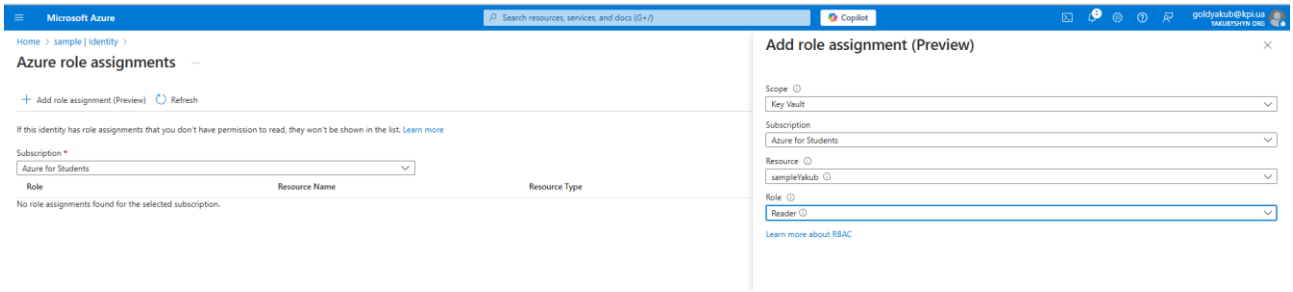

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
goldyakub@kpi.ua
YAKUBYSHYN ORG
Home > Microsoft.ContainerInstances-20250201193312 | Overview > sample
sample | Identity
Container instances
Search
Access control (IAM)
Tags
Settings
Containers
Identity
Properties
Locks
Monitoring
System assigned (preview)
User assigned (preview)
A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. You can grant permissions to the managed identity by using Azure role-based access control (Azure RBAC). The managed identity is authenticated with Microsoft Entra ID, so you don't have to store any credentials in code.
Save
Discard
Refresh
Got feedback?
Status
Off
On

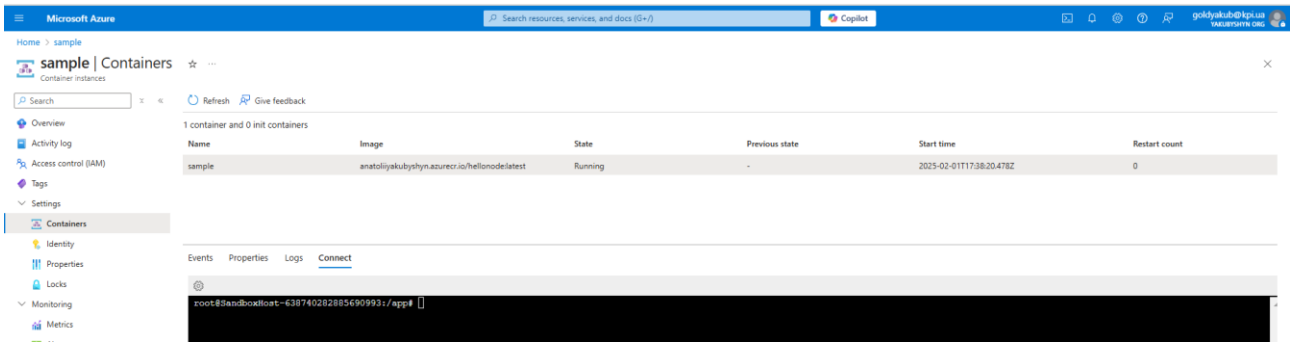


Click azure role assignments



Created secret there



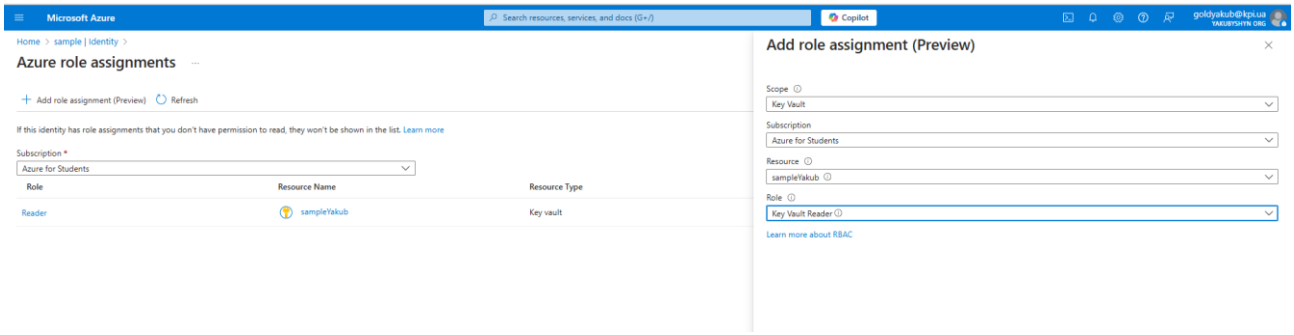



```
root@SandboxHost-638740282885690993:/app# az
bash: az: command not found
root@SandboxHost-638740282885690993:/app# ls
main.js  node_modules  package-lock.json  package.json
root@SandboxHost-638740282885690993:/app# curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
bash: sudo: command not found
root@SandboxHost-638740282885690993:/app# curl -sL https://aka.ms/InstallAzureCLIDeb |  bash
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
```

```
root@SandboxHost-638740282885690993:/app# az version
{
  "azure-cli": "2.68.0",
  "azure-cli-core": "2.68.0",
  "azure-cli-telemetry": "1.1.0",
  "extensions": {}
}
```

```
}
root@SandboxHost-638740282885690993:/app# az login --identity
[
  {
    "environmentName": "AzureCloud",
    "homeTenantId": "be098e76-f2f0-41f0-8292-f151f67b6729",
    "id": "3a612e70-8e22-4425-b3ea-29f6acf32428",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "be098e76-f2f0-41f0-8292-f151f67b6729",
    "user": {
      "assignedIdentityInfo": "MSI",
      "name": "systemAssignedIdentity",
      "type": "servicePrincipal"
    }
  }
]
root@SandboxHost-638740282885690993:/app#
```

ɔting



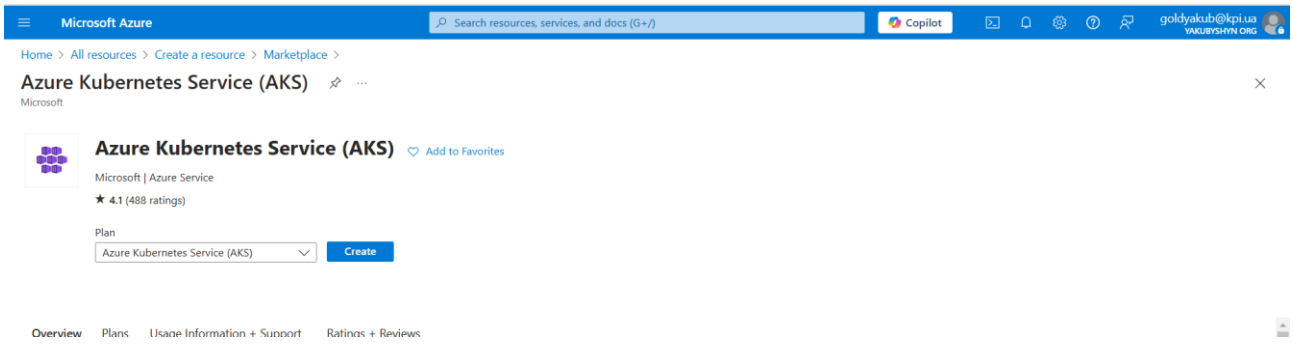
finally, have found needed RBAC role

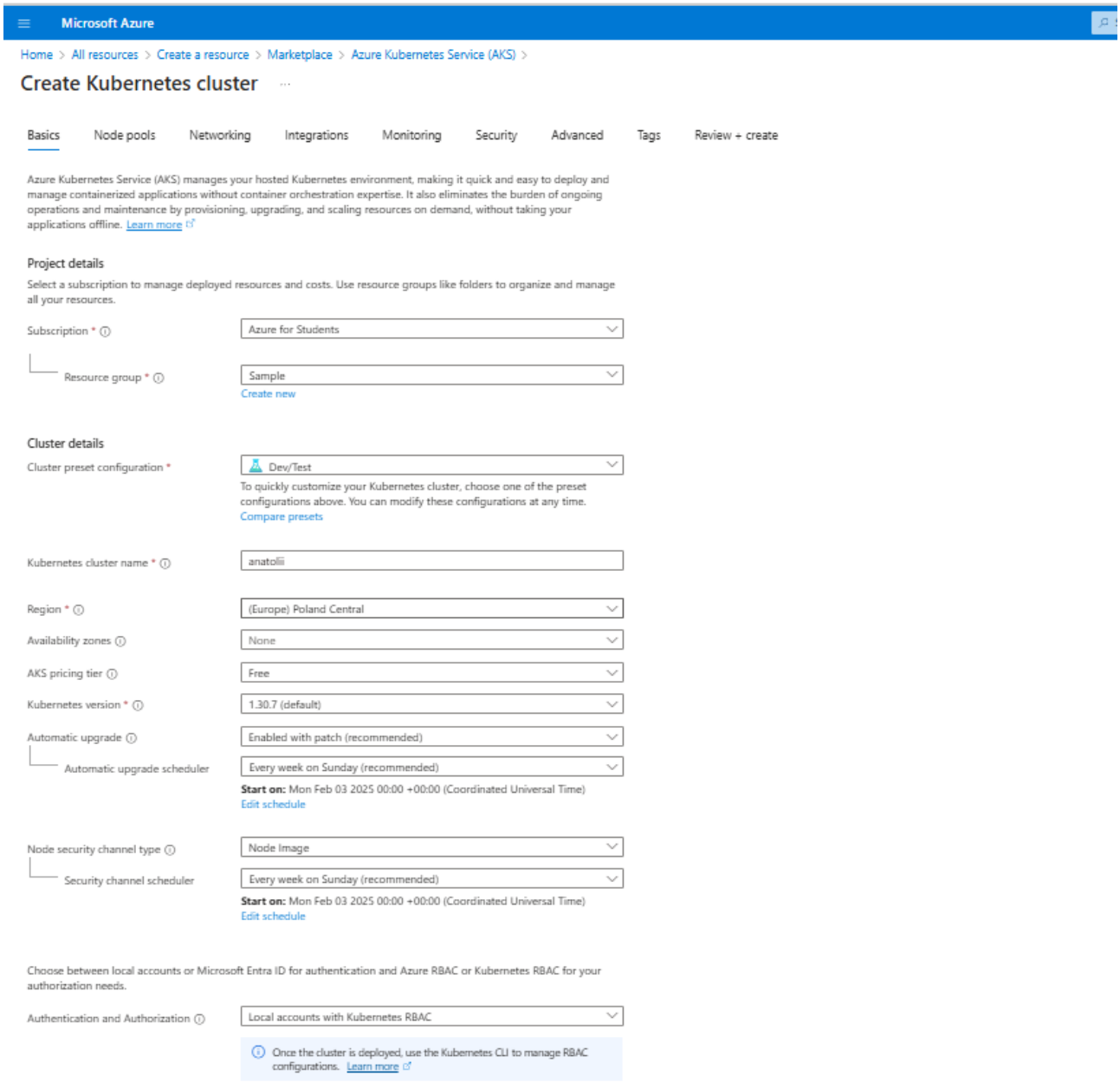
```
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root@SandboxHost-638740282885690993:/app# az login --identity
[
  {
    "environmentName": "AzureCloud",
    "homeTenantId": "be098e76-f2f0-41f0-8292-f151f67b6729",
    "id": "3a612e70-8e22-4425-b3ea-29f6acf32428",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "be098e76-f2f0-41f0-8292-f151f67b6729",
    "user": {
      "assignedIdentityInfo": "MSI",
      "name": "systemAssignedIdentity",
      "type": "servicePrincipal"
    }
  }
]
root@SandboxHost-638740282885690993:/app# az keyvault secret show --name "secret" --vault-name "sampleYakub" --query "value" -o tsv
secret
root@SandboxHost-638740282885690993:/app#
```

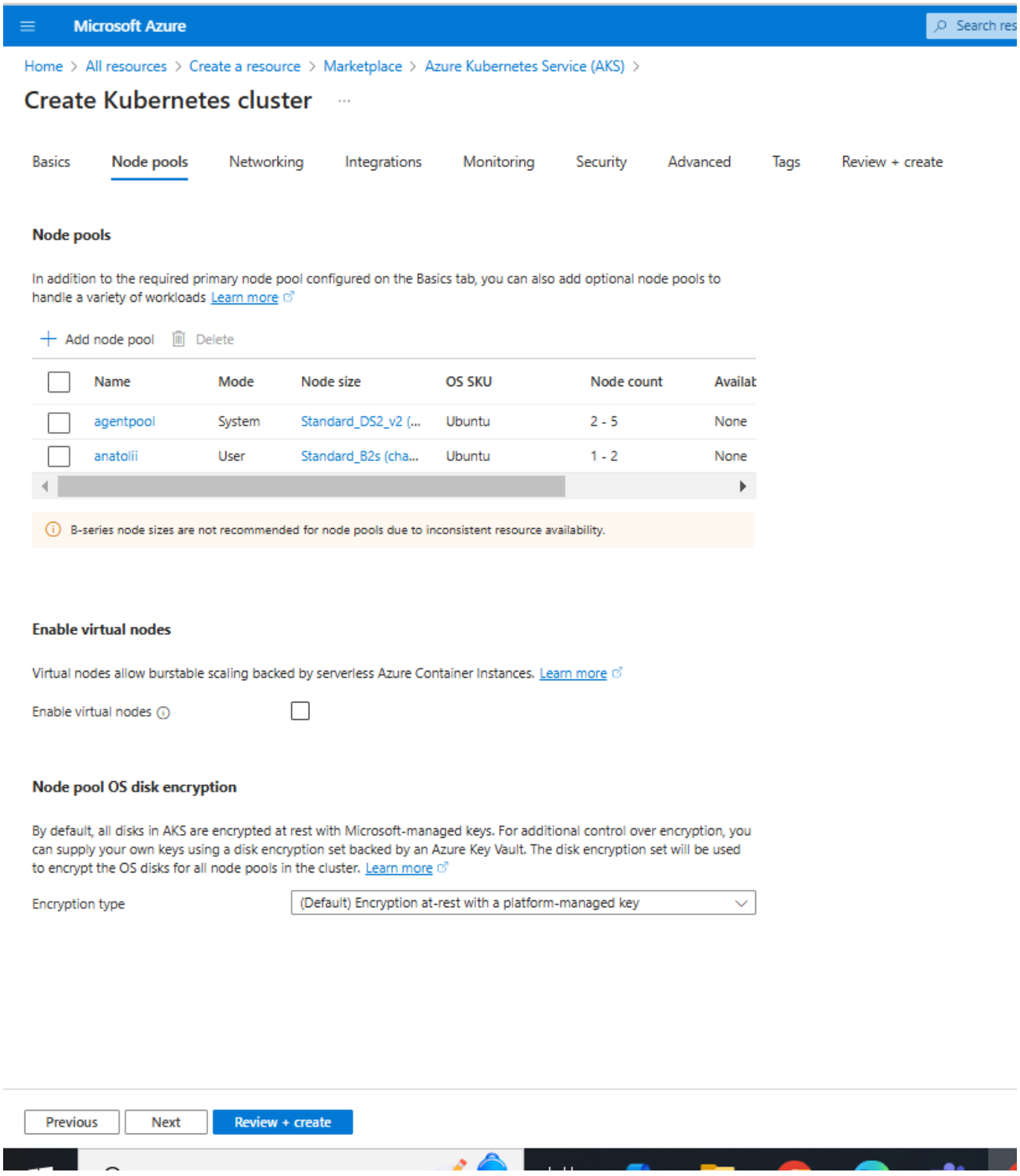
Deleted container instance

Practical Task 5: Deploy a Kubernetes Cluster with AKS via Azure Portal

1. Create an Azure Kubernetes Service (AKS) cluster with the smallest VM size (e.g., B2s) and the minimum number of nodes (e.g., 1–2).






Microsoft Azure
Search res
Home > All resources > Create a resource > Marketplace > Azure Kubernetes Service (AKS) >
Create Kubernetes cluster
Basics
Node pools
Networking
Integrations
Monitoring
Security
Advanced
Tags
Review + create
Node pools
In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads Learn more
Add node pool
Delete
Name
Mode
Node size
OS SKU
Node count
Availat
agentpool
System
Standard_DS2_v2 (...
Ubuntu
2 - 5
None
anatolii
User
Standard_B2s (cha...
Ubuntu
1 - 2
None
B-series node sizes are not recommended for node pools due to inconsistent resource availability.
Enable virtual nodes
Virtual nodes allow burstable scaling backed by serverless Azure Container Instances. Learn more
Enable virtual nodes
Node pool OS disk encryption
By default, all disks in AKS are encrypted at rest with Microsoft-managed keys. For additional control over encryption, you can supply your own keys using a disk encryption set backed by an Azure Key Vault. The disk encryption set will be used to encrypt the OS disks for all node pools in the cluster. Learn more
Encryption type
(Default) Encryption at-rest with a platform-managed key
Previous
Next
Review + create

Microsoft Azure

Home > All resources > Create a resource > Marketplace > Azure Kubernetes Service (AKS) >

Create Kubernetes cluster ...

Basics | Node pools | Networking | Integrations | Monitoring | Security | Advanced | Tags | Review + create

View automation template

Basics

Subscription	Azure for Students
Resource group	Sample
Region	Poland Central
Kubernetes cluster name	anatolii
Kubernetes version	1.30.7
Automatic upgrade	patch
Automatic upgrade scheduler	Every week on Sunday (recommended)
Node security channel type	NodeImage
Security channel scheduler	Every week on Sunday (recommended)

Node pools

Node pools	2
Enable virtual nodes	Disabled

Access

Resource identity	System-assigned managed identity
Local accounts	Enabled
Authentication and Authorization	Local accounts with Kubernetes RBAC
Encryption type	(Default) Encryption at-rest with a platform-managed key

Networking

Private cluster	Disabled
Authorized IP ranges	Disabled
Network configuration	Azure CNI Overlay
DNS name prefix	anatolii-dns
Network policy	None
Load balancer	Standard

Integrations

Container registry	None
Service mesh	Disabled
Azure Policy	Disabled

Monitoring

Enable Container Logs	Disabled
Enable Prometheus metrics	Disabled
Enable Grafana	Disabled
Alert rules	2 rules

Advanced

Infrastructure resource group	MC_Sample_anatolii_polandcentral

Enable Grafana	Disabled
Alert rules	2 rules

Advanced

Infrastructure resource group	MC_Sample_anatolii_polandcentral

Security

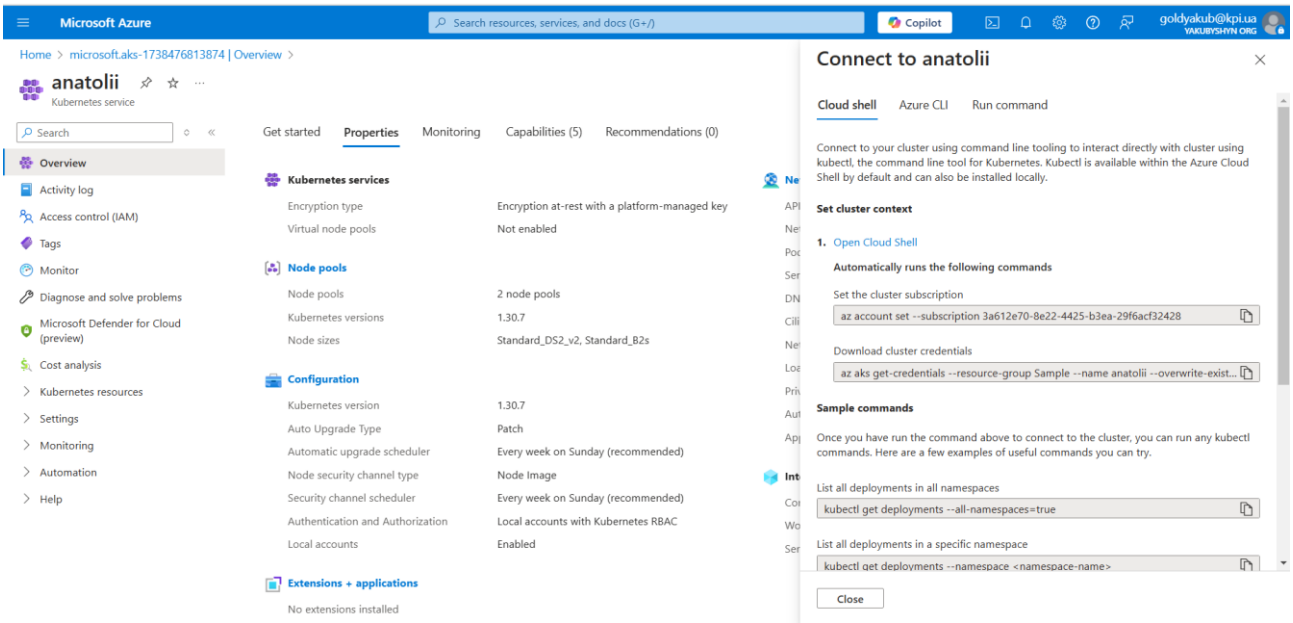
Microsoft Defender for Cloud	Free
OpenID Connect (OIDC)	Enabled
Workload Identity	Enabled
Image Cleaner	Enabled

Tags

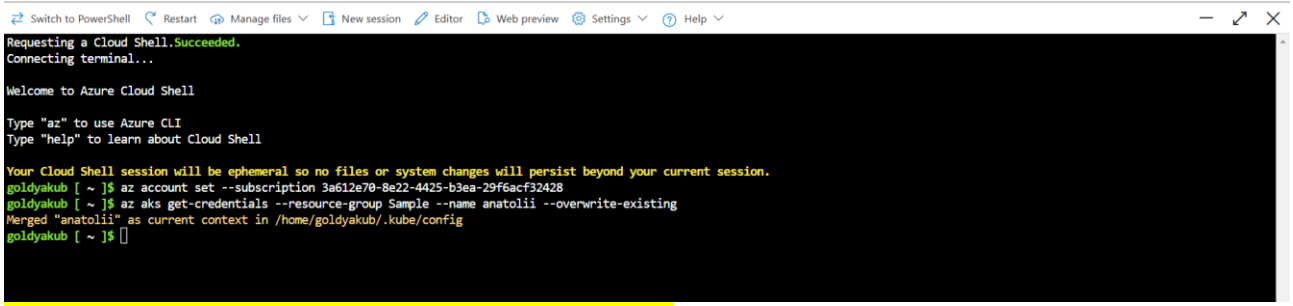
None

Previous | Next | Create

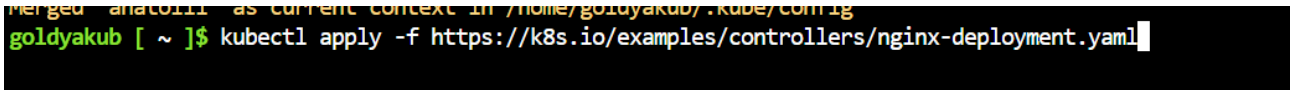
2. Connect to the AKS cluster using Azure Cloud Shell with kubectl.

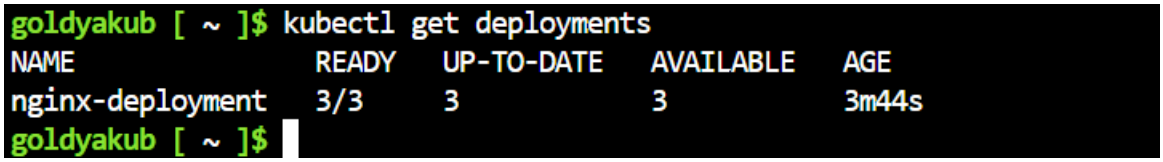


Click open cloudshell



3. Deploy a lightweight Nginx application for verification





```
goldyakub [ ~ ]$ kubectl apply -f https://k8s.io/examples/controllers/nginx-deployment.yaml
deployment.apps/nginx-deployment created
goldyakub [ ~ ]$
```

4. Delete the AKS cluster immediately after testing to avoid additional VM and cluster costs.

Practical Task 6: Deploy a Containerized Application on AKS

Requirements:

1. Build a lightweight Docker image for a simple web application (e.g., a Node.js app with minimal dependencies) and push it to Azure Container Registry (ACR).
2. Reuse the AKS cluster from Task 5 to deploy the application using a Kubernetes deployment and service manifest file.
3. Test the application for a limited time and remove the deployment afterward.

hello-node-deplo

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-node
  labels:
    app: hello-node
spec:
  replicas: 3
  selector:
    matchLabels:
      app: hello-node
  template:
    metadata:
      labels:
        app: hello-node
    spec:
      imagePullSecrets:
        - name: acr-secret
      containers:
      - name: hello-node
        image: anatoliiyakubyshyn.azurecr.io/hellonode:latest
        ports:
        - containerPort: 80

---
```

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: hello-node-service
spec:
  type: NodePort
  selector:
    app: hello-node
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30000
```

```
goldyakub [ ~ ]$ code
goldyakub [ ~ ]$ ls
hello-node-deployment.yaml
goldyakub [ ~ ]$ kubectl apply -f hello-node-deployment.yaml
deployment.apps/hello-node created
The Service "hello-node-service" is invalid: spec.ports[0].nodePort: Forbidden: may not be used when `type` is 'ClusterIP'
goldyakub [ ~ ]$ kubectl apply -f hello-node-deployment.yaml
deployment.apps/hello-node unchanged
The Service "hello-node-service" is invalid: spec.ports[0].nodePort: Invalid value: 80: provided port is not in the valid range. The range of valid ports is 30000-32767
goldyakub [ ~ ]$ kubectl apply -f hello-node-deployment.yaml
deployment.apps/hello-node unchanged
service/hello-node-service created
goldyakub [ ~ ]$
```

Executed this command:

```
PS C:\Users\Anatoly> az aks update -n anatolii -g sample --attach-acr anatoliiyakubyshyn
```

To fix this issue

```
goldyakub [ ~ ]$ kubectl get pods
NAME                          READY   STATUS             RESTARTS   AGE
hello-node-57575f4c7-2l8qh    0/1     ImagePullBackOff   0          25m
hello-node-57575f4c7-b8qmk    0/1     ImagePullBackOff   0          25m
hello-node-57575f4c7-lpcvr    0/1     ImagePullBackOff   0          25m
hello-node-5c9b86c4c4-qnxs7   0/1     ImagePullBackOff   0          7m36s
goldyakub [ ~ ]$
```

Then:

```
goldyakub [ ~ ]$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-5c9b86c4c4-dxdzr   1/1     Running   0          3m4s
hello-node-5c9b86c4c4-qnxs7   1/1     Running   0          13m
hello-node-5c9b86c4c4-rlkqp   1/1     Running   0          3m3s
goldyakub [ ~ ]$
```

Changed type of Service:

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: hello-node-service
spec:
  type: LoadBalancer
  selector:
    app: hello-node
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

3. Test the application for a limited time and remove the deployment afterward.

Practical Task 7: Configure and Use ConfigMaps and Secrets in AKS

1. Create a ConfigMap to store non-sensitive configuration data with only the required keyvalue pairs for the application

Modified app to use 2 env variables

```
const express = require("express");
const app = express();

const PORT = process.env.PORT || 80;
const NAME = process.env.NAME || "World";

app.get("/", (req, res) => {
    res.send(`Hello, ${NAME} ${process.env.SURNAME}!`);
});

app.listen(PORT, () => {
    console.log(`Server running on port ${PORT}`);
});
```

```
ockerfile > ...
# Use official Node.js image
FROM node:18

# Set working directory
WORKDIR /app

# Copy package.json and install dependencies
COPY package.json package-lock.json ./
RUN npm install

# Copy app files
COPY main.js .

# Expose port
EXPOSE 80

# Set default environment variable
ENV NAME="Docker"

ENV SURNAME="USER"

# Run the app
CMD ["node", "main.js"]
```

```
PS C:\Users\Anatoly>
PS C:\Users\Anatoly> docker images
REPOSITORY                                    TAG          IMAGE ID       CREATED          SIZE
hellonamesurname                              latest       ff8f967b47f1   24 seconds ago   1.09GB
hellonode                                     latest       9ad024ed8dc8   22 hours ago     1.09GB
anatoliiyakubyshyn.azurecr.io/hellonod        latest       9ad024ed8dc8   22 hours ago     1.09GB
anatoliiyakubyshyn.azurecr.io/hellonode       latest       9ad024ed8dc8   22 hours ago     1.09GB
anatoliiyakubyshyn.azurecr.io/hellonode       <none>       b6b651fba471   22 hours ago     1.09GB
anatoliiyakubyshyn.azurecr.io/hellonode       <none>       06f9a1fbf684   22 hours ago     1.09GB
anatoliiyakubyshyn.azurecr.io/silpoparser     latest       7fb2d558b269   24 hours ago     792MB
silpoparser                                   latest       369222a50bc4   24 hours ago     792MB
<none>                                        <none>       b2b8bdab32cf   24 hours ago     792MB
<none>                                        <none>       a73acbc15237   24 hours ago     792MB
<none>                                        <none>       5e1dcd4e560f   24 hours ago     792MB
anatoliiyakubyshyn.azurecr.io/silpoparser     <none>       862936a21937   25 hours ago     792MB
tomcat                                        latest       9a15eef1e849   3 weeks ago      479MB
mcr.microsoft.com/mssql/server                2022-latest  142124bf195c   2 months ago     1.59GB
PS C:\Users\Anatoly> docker tag hellonamesurname anatoliiyakubyshyn.azurecr.io/hellonamesurname
PS C:\Users\Anatoly> docker push anatoliiyakubyshyn.azurecr.io/hellonamesurname
Using default tag: latest
The push refers to repository [anatoliiyakubyshyn.azurecr.io/hellonamesurname]
2546c74264e3: Pushed
fd975ca59459: Mounted from hellonode
231ea03fb497: Mounted from hellonode
b41cdddf76f7: Mounted from hellonode
207fd040c49e: Mounted from hellonode
0149b9feb6a3: Mounted from hellonode
7e517a081f3d: Mounted from hellonode
d3244fce0fd3: Mounted from hellonode
ebad64620a59: Mounted from hellonode
f379f6005525: Mounted from hellonode
0e5c23e041ee: Mounted from hellonode
397f1b2e2505: Mounted from hellonode
latest: digest: sha256:b2be08cd9cda1bc044783277e91d8cc4b4a5202a67df0af536447a7e9ee5a30b size: 2836
PS C:\Users\Anatoly>
```

2. Create a Kubernetes Secret to store sensitive data (e.g., API keys) with the least amount of information needed.

3. Update the application deployment to use the ConfigMap and Secrets

```
tolia@M515U:/mnt/d/Azure/task4/task7$ echo -n 'Yakubyshyn' | base64
WWFrdWJ5c2h5bg==
```

io.k8s.api.core.v1.Secret (v1@secret.json)

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-surname
type: Opaque
data:
  SURNAME: WWFrdWJ5c2h5bg==
```

io.k8s.api.core.v1.ConfigMap (v1@configmap.json)

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-name
data:
  NAME: Anatolii
```

io.k8s.api.apps.v1.Deployment (v1@deployment.json)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-name-surname
spec:
  selector:
    matchLabels:
      app: hello-name-surname
  template:
    metadata:
      labels:
        app: hello-name-surname
    spec:
      containers:
      - name: hello-name-surname
        image: anatoliiyakubyshyn.azurecr.io/hellonamesurname
        resources:
          limits:
            memory: "128Mi"
            cpu: "500m"
        ports:
        - containerPort: 80
        env:
        - name: NAME
          valueFrom:
            configMapKeyRef:
              name: app-name
              key: NAME
        - name: SURNAME
          valueFrom:
            secretKeyRef:
              name: app-surname
              key: SURNAME
```

helo-name-surname-deployment.yaml | service.yaml | secrets.yaml | configmap.yaml

service.yaml > {} spec > [ ] ports > {} 0

io.k8s.api.core.v1.Service (v1@service.json)

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-name-surname
spec:
  type: LoadBalancer
  selector:
    app: hello-name-surname
  ports:
  - port: 80
    targetPort: 80
```

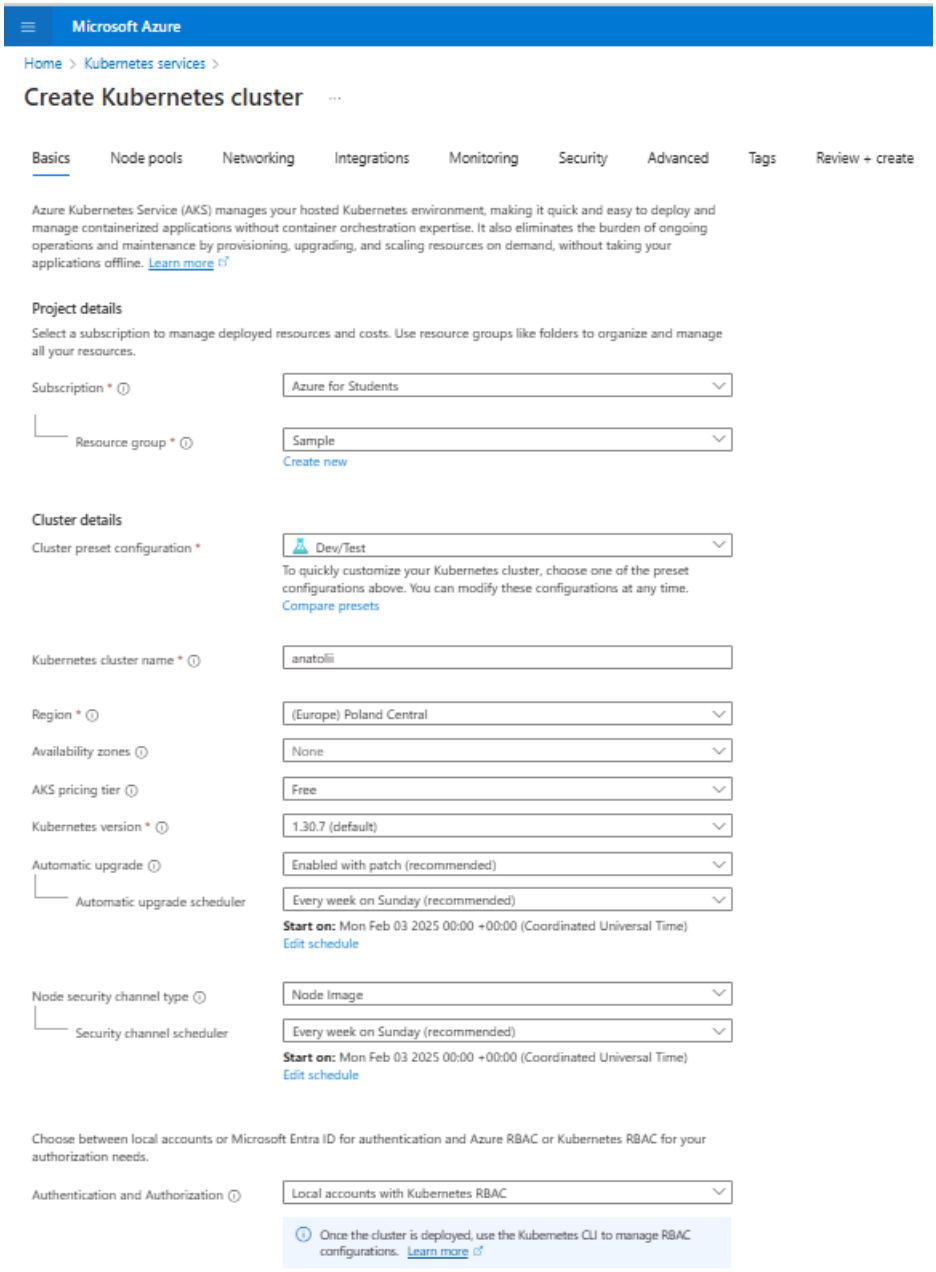
Hello, Anatolii Yakubyshyn!

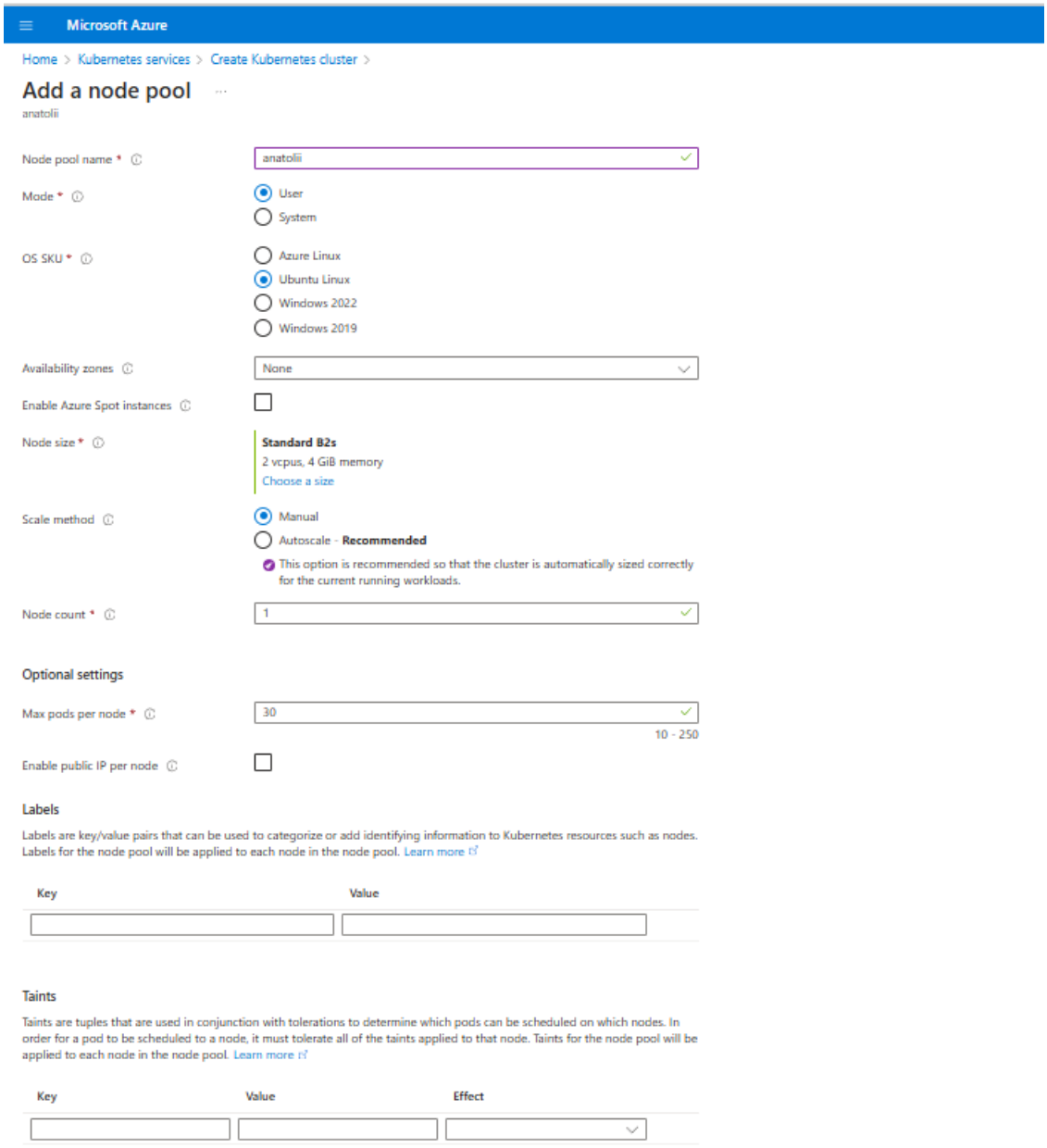
4. Remove the ConfigMap, Secret, and deployment after testing.

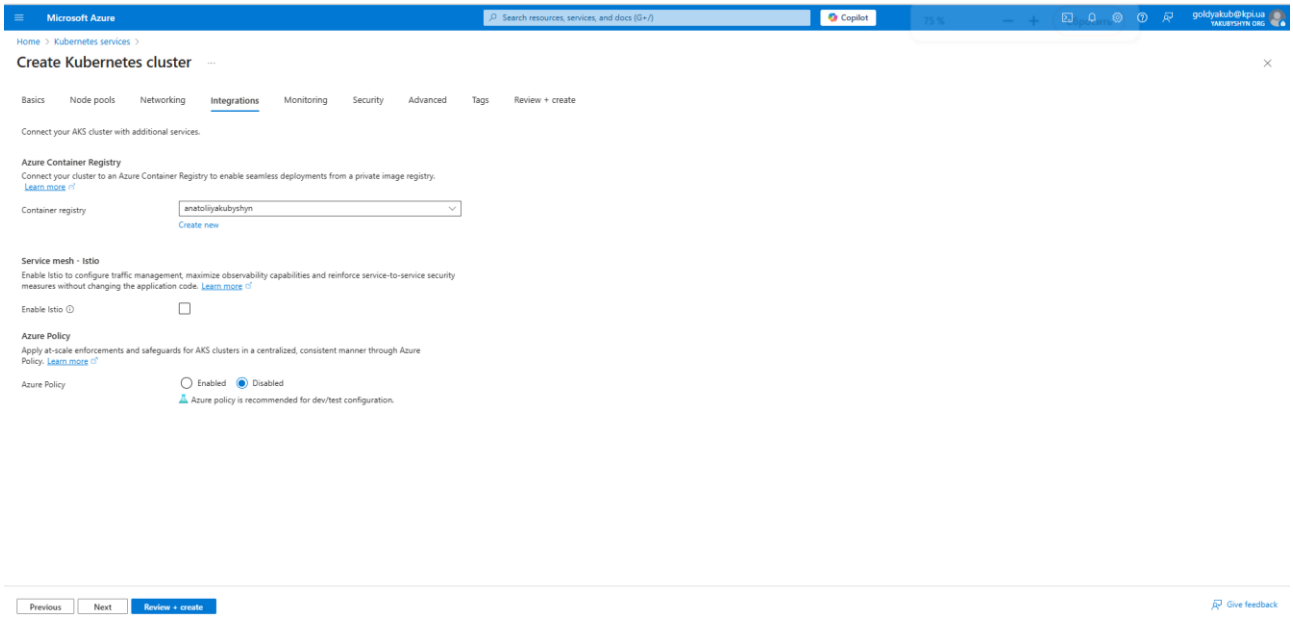
```
goldyakub [ ~ ]$ kubectl get deploy -A
NAMESPACE     NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
default       hello-name-surname                  1/1     1            1           6m49s
kube-system   azure-wi-webhook-controller-manager   2/2     2            2           33m
kube-system   coredns                             2/2     2            2           34m
kube-system   coredns-autoscaler                  1/1     1            1           34m
kube-system   eraser-controller-manager           1/1     1            1           33m
kube-system   konnectivity-agent                  2/2     2            2           34m
kube-system   metrics-server                      2/2     2            2           34m
goldyakub [ ~ ]$ kubectl delete deploy hello-name-surname
deployment.apps "hello-name-surname" deleted
goldyakub [ ~ ]$ kubectl delete secret app-surname
secret "app-surname" deleted
goldyakub [ ~ ]$ kubectl delete configmap app-name
configmap "app-name" deleted
```

Practical Task 8: Scale Applications in AKS

1. Deploy a stateless application to the AKS cluster using minimal resource specifications.




Microsoft Azure
Home > Kubernetes services > Create Kubernetes cluster >
Add a node pool
anatolii
Node pool name *
anatolii
Mode *
User
System
OS SKU *
Azure Linux
Ubuntu Linux
Windows 2022
Windows 2019
Availability zones
None
Enable Azure Spot instances
Node size *
Standard B2s
2 vcpus, 4 GiB memory
Choose a size
Scale method
Manual
Autoscale - Recommended
This option is recommended so that the cluster is automatically sized correctly for the current running workloads.
Node count *
1
Optional settings
Max pods per node *
30
10 - 250
Enable public IP per node
Labels
Labels are key/value pairs that can be used to categorize or add identifying information to Kubernetes resources such as nodes. Labels for the node pool will be applied to each node in the node pool. Learn more
Key
Value
Taints
Taints are tuples that are used in conjunction with tolerations to determine which pods can be scheduled on which nodes. In order for a pod to be scheduled to a node, it must tolerate all of the taints applied to that node. Taints for the node pool will be applied to each node in the node pool. Learn more
Key
Value
Effect



deployed the app from previous task

2. Use the kubectl scale command to manually scale the application to only 2–3 replicas for testing.

```
kubernetes          ClusterIP      10.0.0.1       <none>          443/TCP          21m
goldyakub [ ~ ]$ kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
hello-name-surname-88f866cb5-kqcf8   1/1     Running   0          6m59s
goldyakub [ ~ ]$ kubectl scale --replicas=3 -f helo-name-surname-deployment.yaml
deployment.apps/hello-name-surname scaled
goldyakub [ ~ ]$ kubectl get pods
NAME                                   READY   STATUS              RESTARTS   AGE
hello-name-surname-88f866cb5-bcctq   1/1     Running             0          4s
hello-name-surname-88f866cb5-kqcf8   1/1     Running             0          8m40s
hello-name-surname-88f866cb5-tngnx   0/1     ContainerCreating   0          4s
goldyakub [ ~ ]$ kubectl get pods
NAME                                   READY   STATUS              RESTARTS   AGE
hello-name-surname-88f866cb5-bcctq   1/1     Running             0          11s
hello-name-surname-88f866cb5-kqcf8   1/1     Running             0          8m47s
hello-name-surname-88f866cb5-tngnx   0/1     ContainerCreating   0          11s
goldyakub [ ~ ]$ kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
hello-name-surname-88f866cb5-bcctq   1/1     Running   0          24s
hello-name-surname-88f866cb5-kqcf8   1/1     Running   0          9m
hello-name-surname-88f866cb5-tngnx   1/1     Running   0          24s
goldyakub [ ~ ]$
```

3. Set up Horizontal Pod Autoscaler (HPA) with reasonable CPU usage thresholds to minimize pod creation.

```
hpa.yaml > {} spec > [ ] metrics > {} 0 > {} resource > {} target
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: hello-name-surname
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: hello-name-surname
  minReplicas: 1
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 40
```

```
goldyakub [ ~ ]$ nano hpa.yaml
goldyakub [ ~ ]$ kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/hello-name-surname created
goldyakub [ ~ ]$
```

4. Simulate load on the application for a short duration and remove the deployment after observing the scaling behavior.

After apply:

```
goldyakub [ ~ ]$ kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
hello-name-surname-88f866cb5-tngnx     1/1     Running   0          14m
goldyakub [ ~ ]$
```

```
 GNU nano 6.0                                  hpa.yaml                                  Modified
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: hello-name-surname
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: hello-name-surname
  minReplicas: 1
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 5
```

```
goldyakub [ ~ ]$ kubectl top pods
NAME                                   CPU(cores)   MEMORY(bytes)
hello-name-surname-88f866cb5-tngnx     1m           39Mi
goldyakub [ ~ ]$ kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
hello-name-surname-88f866cb5-tngnx     1/1     Running   0          26m
```

After running ‘ddos.py’:

```
goldyakub [ ~ ]$ kubectl top pods
NAME                                   CPU(cores)   MEMORY(bytes)
hello-name-surname-88f866cb5-tngnx     28m          36Mi
goldyakub [ ~ ]$ kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
hello-name-surname-88f866cb5-tm972     1/1     Running   0          14s
hello-name-surname-88f866cb5-tngnx     1/1     Running   0          27m
goldyakub [ ~ ]$
```

ddos.py > ...

```python
import subprocess
from concurrent.futures import ThreadPoolExecutor

# Function to run curl command
def run_curl(url):
    for i in range(10):
        result = subprocess.run(['curl', '-o', '/dev/null', '-s', url], capture_output=True, text=True)
        result = subprocess.run(['curl', '-o', '/dev/null', '-s', url], capture_output=True, text=True)
        result = subprocess.run(['curl', '-o', '/dev/null', '-s', url], capture_output=True, text=True)
        result = subprocess.run(['curl', '-o', '/dev/null', '-s', url], capture_output=True, text=True)
        result = subprocess.run(['curl', '-o', '/dev/null', '-s', url], capture_output=True, text=True)
    result = subprocess.run(['curl', '-o', '/dev/null', '-s', url], capture_output=True, text=True)
    return result.returncode  # Return the status code of the curl request

# URL to send multiple requests
url = 'http://74.248.73.225/'

# Number of requests to send in parallel
num_requests = 1050

# Using ThreadPoolExecutor to send requests in parallel
with ThreadPoolExecutor() as executor:
    results = executor.map(run_curl, [url] * num_requests)

# Print results for each request
for i, result in enumerate(results, 1):
    print(f"Request {i} completed with exit code {result}")

```

deleted deployment:

```
goldyakub [ ~ ]$ ls
configmap.yaml  helo-name-surname-deployment.yaml  hpa.yaml  secrets.yaml  service.yaml
goldyakub [ ~ ]$ kubectl delete deploy  hello-name-surname
deployment.apps "hello-name-surname" deleted
goldyakub [ ~ ]$ kubectl top pods
NAME                                  CPU(cores)   MEMORY(bytes)
hello-name-surname-88f866cb5-2zf8p    1m           17Mi
hello-name-surname-88f866cb5-jkf65    1m           17Mi
hello-name-surname-88f866cb5-tm972    1m           53Mi
hello-name-surname-88f866cb5-tngnx    1m           54Mi
goldyakub [ ~ ]$ kubectl get pods
NAME                                  READY   STATUS        RESTARTS   AGE
hello-name-surname-88f866cb5-2zf8p    1/1     Terminating   0          2m39s
hello-name-surname-88f866cb5-jkf65    1/1     Terminating   0          2m39s
hello-name-surname-88f866cb5-tm972    1/1     Terminating   0          4m39s
hello-name-surname-88f866cb5-tngnx    1/1     Terminating   0          32m
goldyakub [ ~ ]$
```

but before I did it there have been already **4** pods.

Practical Task 9: Rolling Update of an Application in AKS

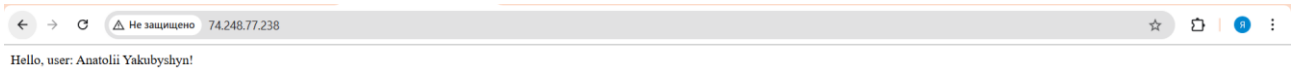
1. Deploy a lightweight version of your application to the AKS cluster.

First version of app. (As in the previous task)

2. Update the Docker image to a new version with minimal changes (e.g., color change).
3. Perform a rolling update using kubectl set image with minimal replicas to reduce resource usage.

4. Verify the update process quickly and remove the deployment after the update

```
goldyakub [ ~ ]$ kubectl set image deployment/hello-name-surname hello-name-surname=anatoliiyakubyshyn.azurecr.io/hellonamesurname:latest
deployment.apps/hello-name-surname image updated
goldyakub [ ~ ]$ kubectl rollout history deployment hello-name-surname
deployment.apps/hello-name-surname
REVISION  CHANGE-CAUSE
1         <none>
2         <none>

goldyakub [ ~ ]$ kubectl get svc
NAME                 TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)        AGE
hello-name-surname   LoadBalancer   10.0.244.119   74.248.77.238   80:31699/TCP   19m
kubernetes           ClusterIP      10.0.0.1       <none>          443/TCP        32m
```



Hello, user: Anatolii Yakubyshyn!

Bonus task.

GitOps with AKS Requirements:

1. Setup ArgoCD on Azure Kubernetes Services

2. Perform image update on cluster

```
goldyakub [ ~ ]$ kubectl create namespace argocd
namespace/argocd created
goldyakub [ ~ ]$ kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/applicationsets.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/appprojects.argoproj.io created
serviceaccount/argocd-application-controller created
```

```
networkpolicy.networking.k8s.io/argocd-server-network-policy created
goldyakub [ ~ ]$ kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
service/argocd-server patched
```

```
service/argocd-server patched
goldyakub [ ~ ]$ kubectl get svc -n argocd
NAME                                  TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)                      AGE
argocd-applicationset-controller      ClusterIP      10.0.144.74     <none>          7000/TCP,8080/TCP            88s
argocd-dex-server                     ClusterIP      10.0.110.13     <none>          5556/TCP,5557/TCP,5558/TCP   88s
argocd-metrics                        ClusterIP      10.0.70.189     <none>          8082/TCP                     88s
argocd-notifications-controller-metrics   ClusterIP      10.0.109.206    <none>          9001/TCP                     88s
argocd-redis                          ClusterIP      10.0.238.160    <none>          6379/TCP                     88s
argocd-repo-server                    ClusterIP      10.0.189.251    <none>          8081/TCP,8084/TCP            88s
argocd-server                         LoadBalancer   10.0.120.58     74.248.107.33   80:31277/TCP,443:30706/TCP   88s
argocd-server-metrics                 ClusterIP      10.0.67.40      <none>          8083/TCP                     88s
goldyakub [ ~ ]$
```

```
oldyakub [ ~ ]$ kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d; echo
```

To retrieve password

Windows

Download With PowerShell: Invoke-WebRequest ¶

You can view the latest version of Argo CD at the link above or run the following command to grab the version:

```
$version = (Invoke-RestMethod https://api.github.com/repos/argoproj/argo-cd/releases/lates
```

Replace `$version` in the command below with the version of Argo CD you would like to download:

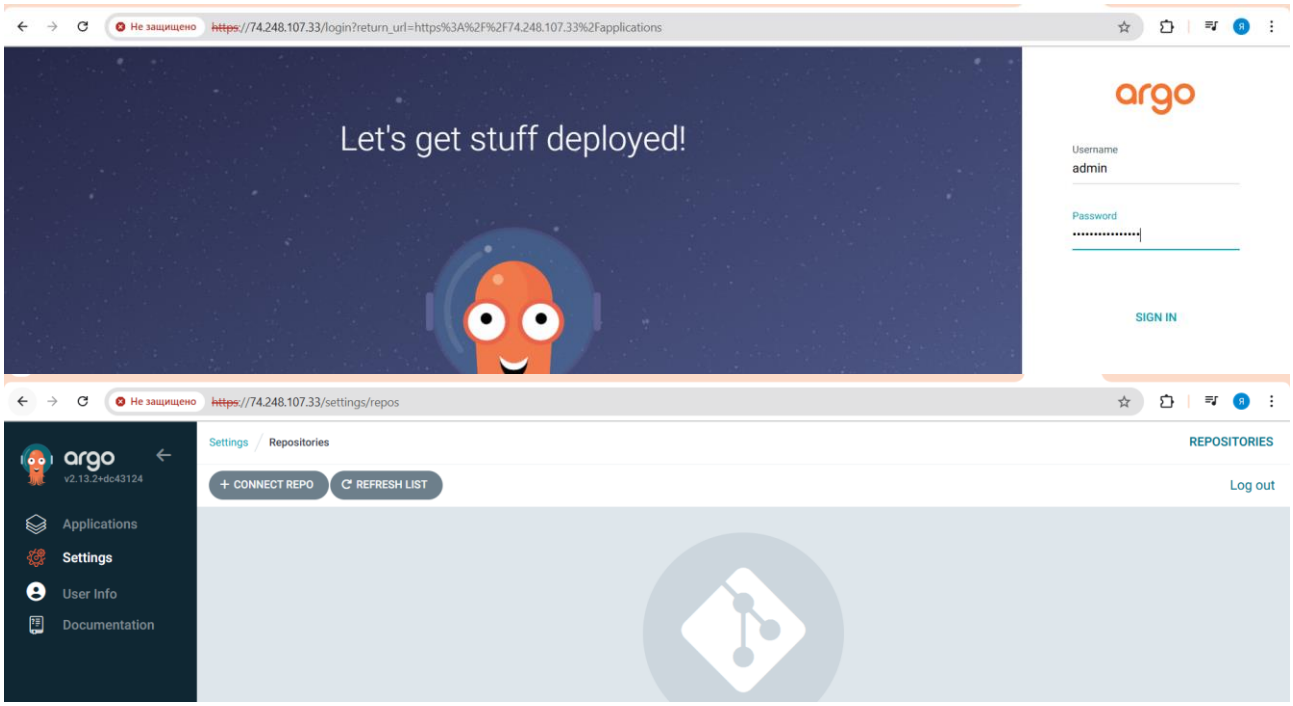
```
$url = "https://github.com/argoproj/argo-cd/releases/download/" + $version + "/argocd-wind
$output = "argocd.exe"

Invoke-WebRequest -Uri $url -OutFile $output
```

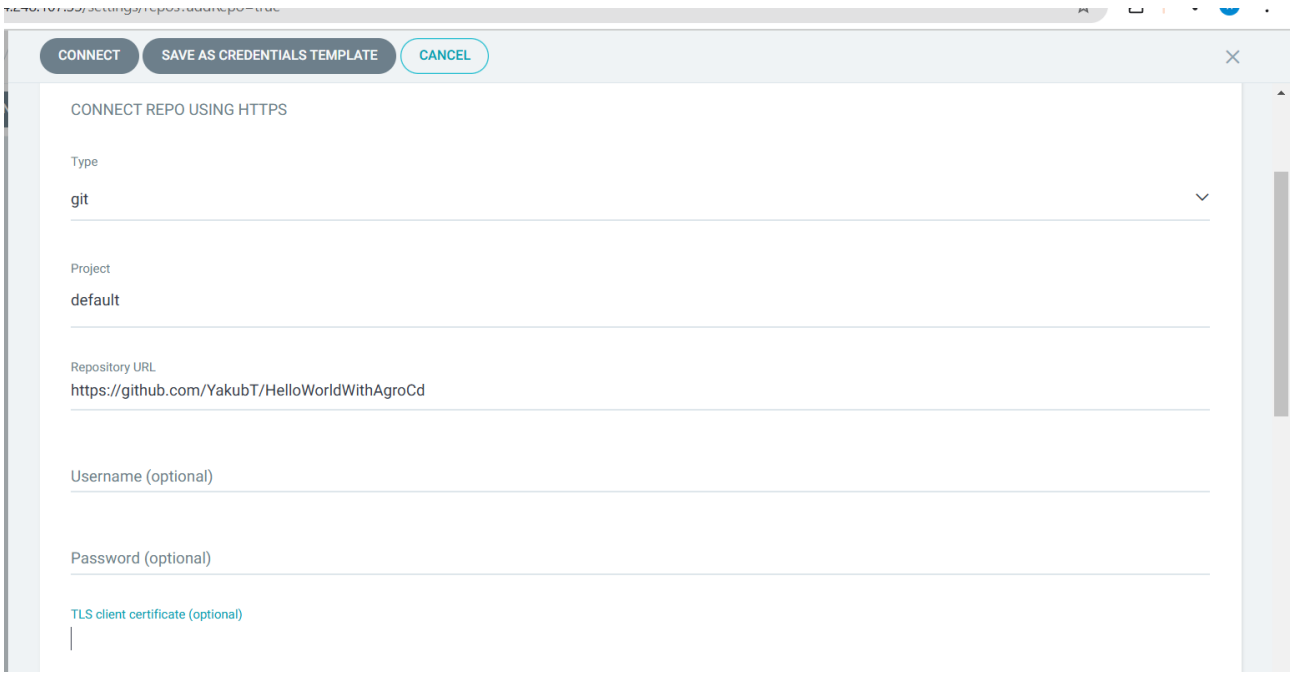
Also please note you will probably need to move the file into your PATH. Use following command to add Argo CD into environment variables PATH

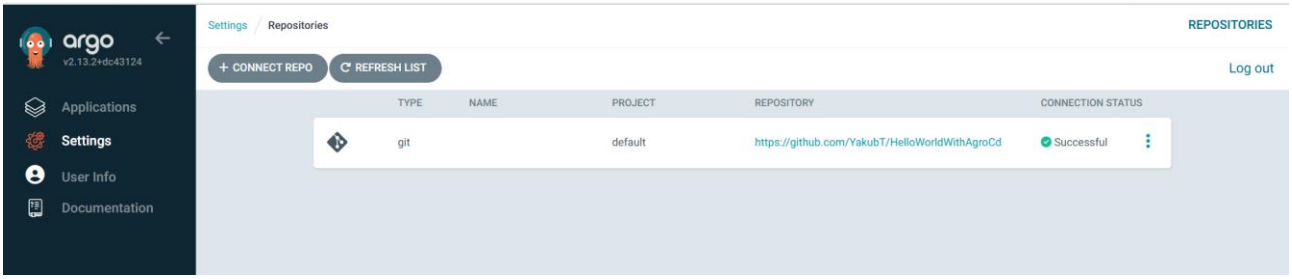
```
[Environment]::SetEnvironmentVariable("Path", "$env:Path;C:\Path\To\ArgoCD-CLI", "User")
```

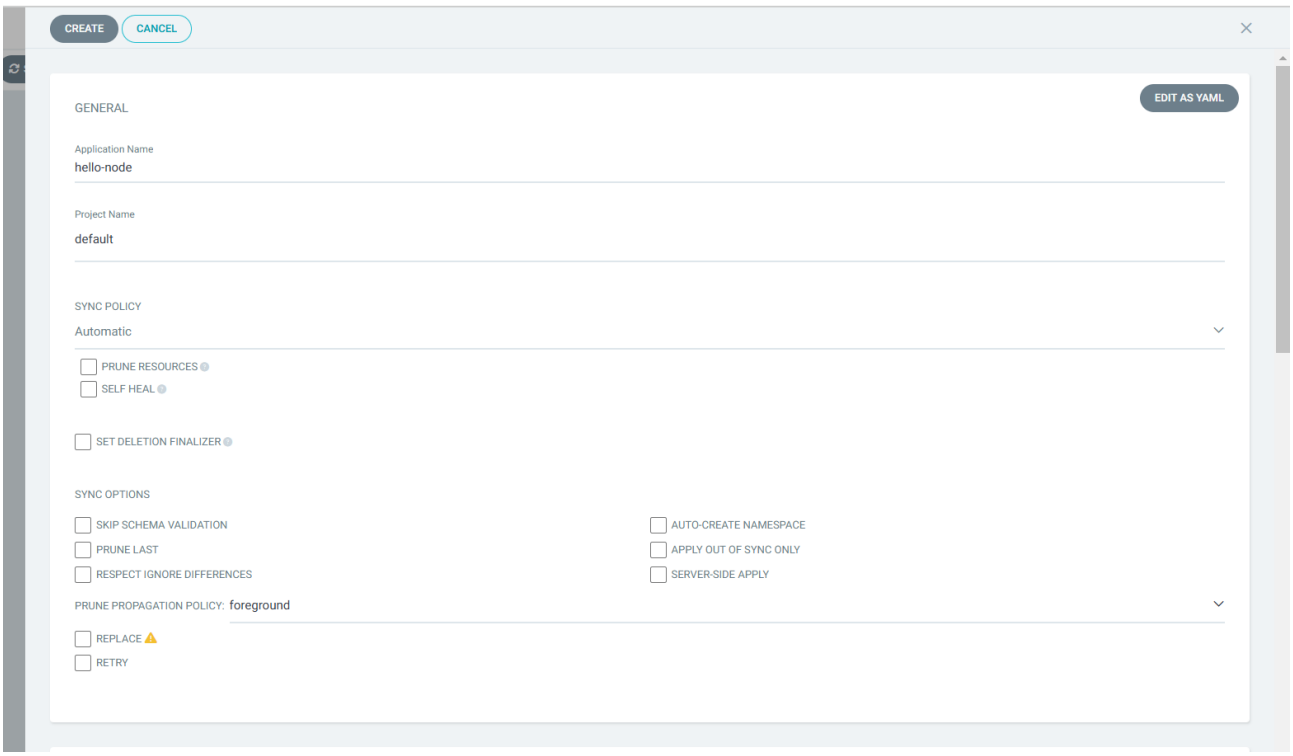


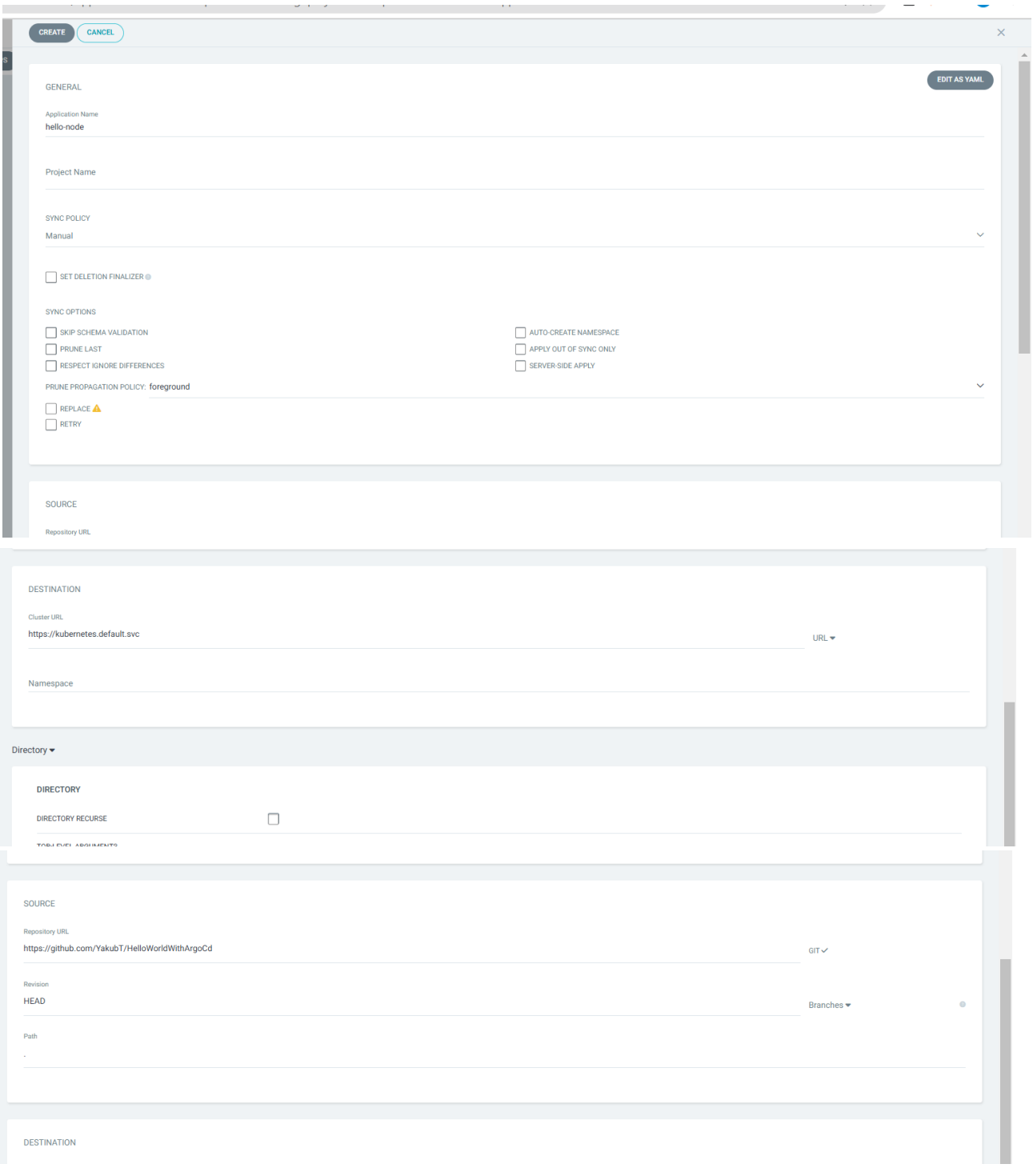
```
D:\Azure\task4>argocd login 74.248.107.33 --username admin --password EjLVJrlwJibcpVju --insecure
'admin:login' logged in successfully
Context '74.248.107.33' updated

D:\Azure\task4>
```




CONNECT
SAVE AS CREDENTIALS TEMPLATE
CANCEL
CONNECT REPO USING HTTPS
Type
git
Project
default
Repository URL
https://github.com/YakubT/HelloWorldWithAgroCd
Username (optional)
Password (optional)
TLS client certificate (optional)

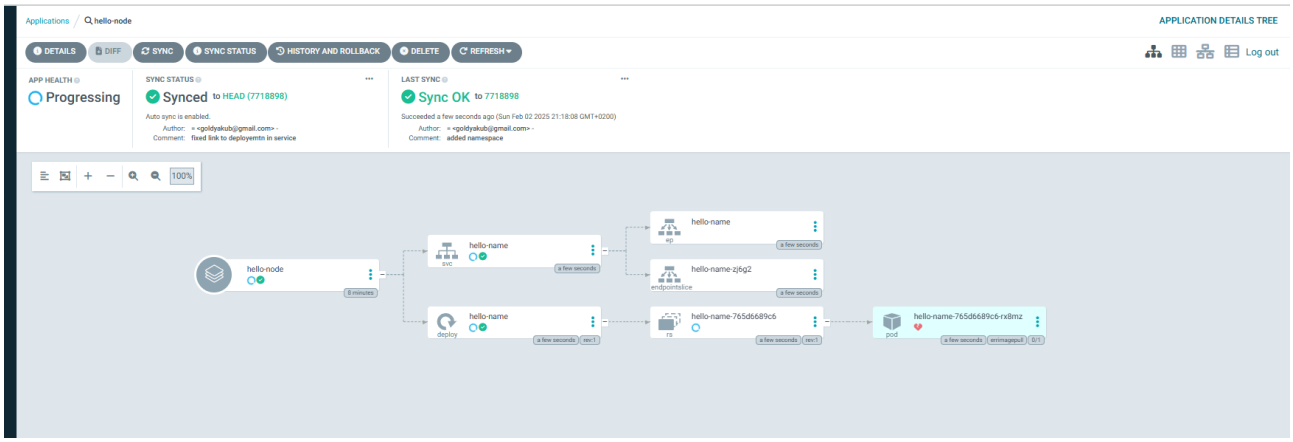

argo
v2.13.2+dc43124
Applications
Settings
User Info
Documentation
Settings / Repositories
REPOSITORIES
+ CONNECT REPO
REFRESH LIST
Log out
TYPE
NAME
PROJECT
REPOSITORY
CONNECTION STATUS
git
default
https://github.com/YakubT/HelloWorldWithAgroCd
Successful


CREATE
CANCEL
EDIT AS YAML
GENERAL
Application Name
hello-node
Project Name
default
SYNC POLICY
Automatic
PRUNE RESOURCES
SELF HEAL
SET DELETION FINALIZER
SYNC OPTIONS
SKIP SCHEMA VALIDATION
PRUNE LAST
RESPECT IGNORE DIFFERENCES
AUTO-CREATE NAMESPACE
APPLY OUT OF SYNC ONLY
SERVER-SIDE APPLY
PRUNE PROPAGATION POLICY: foreground
REPLACE
RETRY

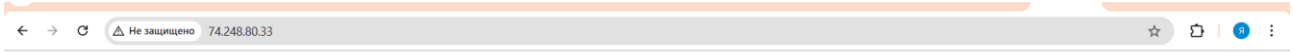

CREATE
CANCEL
GENERAL
EDIT AS YAML
Application Name
hello-node
Project Name
SYNC POLICY
Manual
SET DELETION FINALIZER
SYNC OPTIONS
SKIP SCHEMA VALIDATION
AUTO-CREATE NAMESPACE
PRUNE LAST
APPLY OUT OF SYNC ONLY
RESPECT IGNORE DIFFERENCES
SERVER-SIDE APPLY
PRUNE PROPAGATION POLICY: foreground
REPLACE
RETRY
SOURCE
Repository URL

DESTINATION
Cluster URL
https://kubernetes.default.svc
URL
Namespace
Directory
DIRECTORY
DIRECTORY RECURSE

SOURCE
Repository URL
https://github.com/YakubT/HelloWorldWithArgoCd
GIT
Revision
HEAD
Branches
Path
.
DESTINATION




```
goldyakub [ ~ ]$ kubectl get svc
NAME          TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)        AGE
hello-name    LoadBalancer   10.0.130.199    74.248.80.33   80:30732/TCP   46s
kubernetes    ClusterIP      10.0.0.1        <none>         443/TCP        41m
goldyakub [ ~ ]$
```

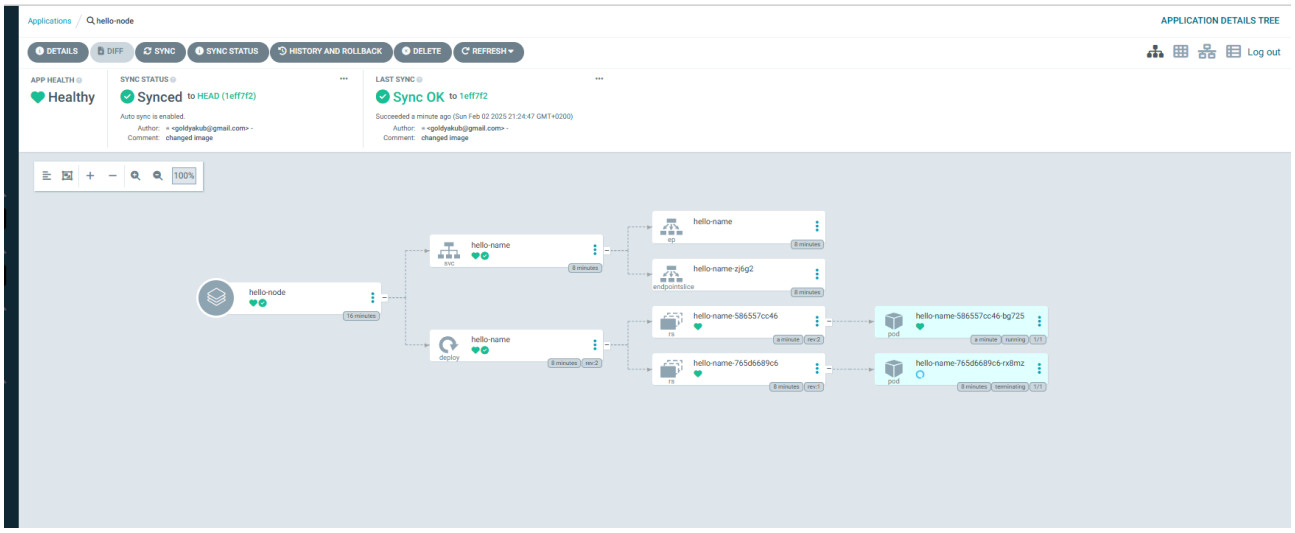


Hello, Docker User!

```
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-name
  namespace: default
spec:
  selector:
    matchLabels:
      app: hello-name
  template:
    metadata:
      labels:
        app: hello-name
    spec:
      containers:
      - name: hello-name
        image: anatoliiyakubyshyn.azurecr.io/hellonamesurname
        resources:
          limits:
            memory: "128Mi"
            cpu: "500m"
        ports:
        - containerPort: 80
```

changed image

```
PS D:\Azure\task4\agrocd> git add -A
PS D:\Azure\task4\agrocd> git commit -m "changed image"
[main 1eff7f2] changed image
 1 file changed, 1 insertion(+), 1 deletion(-)
PS D:\Azure\task4\agrocd> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 331 bytes | 331.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote: This repository moved. Please use the new location:
remote:   https://github.com/YakubT/HelloWorldWithArgoCd.git
To https://github.com/YakubT/HelloWorldWithAgroCd.git
   7718898..1eff7f2  main -> main
PS D:\Azure\task4\agrocd>
```



Deleted ACR and cluster