FILL IN : Lab 7

*Lovepreet Singh*

Questions from the lab

*In the lab assignment, you’ll see several questions in red boxes. Paste those questions and their respective answers below. Make sure your answer is concise and well-formatted. You may submit this as e.g. a screenshot of a filled-out cell in a copy of the Notion document (e.g. with code, so that code formatting is maintained).*

**Q1 :** Write down at least 2 benefits of "Infrastructure as Code (IaC)”. Use online blogposts as your reference.

**A1 :**

1. Consistency and Repeatability
2. Faster and Automated Deployments

**Q2 :** Compute Start/Stop Cronjob schedule Give me a similar Schedule that starts the machines at 8.30 in the morning, except on Fridays when it should only start at 10 in the morning.

**A2 :**schedules:

compute\_start\_stop:

- action: start

state: enabled

trigger:

expression: 30 8 \* \* 1-4

time\_zone: UTC

type: cron

- action: start

state: enabled

trigger:

expression: 0 10 \* \* 5

time\_zone: UTC

type: cron

**Q3 :**

1. What is the other option next to type: cron for this “compute\_start\_stop” trigger.
2. Do we need to enter a time\_zone for the triggers?

**A3 :**

1. Type: recurrence (for schedule-based tasks).
2. No, not strictly required but recommended.

**Q4 :** Cleanup az ml output

**A4 :**

idle\_time\_before\_shutdown: PT30M

idle\_time\_before\_shutdown\_minutes: 30

**last\_operation:** # Can be deleted – operation status, not needed for IaC

operation\_name: Create

operation\_status: Succeeded

operation\_time: '2025-05-26T23:28:13.089Z'

operation\_trigger: User

**location: germanywestcentral** # Optional – region is usually set elsewhere

name: cli-created-machine-ls

**network\_settings:** # Can be deleted – auto-generated network info

private\_ip\_address: 10.0.0.4

public\_ip\_address: 135.220.42.38

**os\_image\_metadata:** # Can be deleted – auto-generated OS info

current\_image\_version: 25.04.23

is\_latest\_os\_image\_version: false

latest\_image\_version: 25.04.07

**provisioning\_state: Succeeded** # Can be deleted – deployment status, not needed

**release\_quota\_on\_stop: false** # Can be deleted – runtime property

**resourceGroup: mlops-demo** # Optional – usually set in deployment, not in YAML

schedules:

compute\_start\_stop:

- action: stop

**provisioning\_state: Completed** # Can be deleted – schedule status, not needed

schedule\_id: 7ee63ae1-5005-48c0-981f-4df6d618e3d8 # Can be deleted – auto-generated

state: enabled

trigger:

expression: 00 20 \* \* 1,2,3,4,5

start\_time: '2024-05-14T11:00:00'

time\_zone: UTC

type: cron

**services:** # Can be deleted – auto-generated service endpoints

- display\_name: Jupyter

endpoint\_uri: https://cli-created-machine-ls.germanywestcentral.instances.azureml.ms/tree/

- display\_name: Jupyter Lab

endpoint\_uri: https://cli-created-machine-ls.germanywestcentral.instances.azureml.ms/lab

size: Standard\_E2ds\_v4

**ssh\_public\_access\_enabled: false** # Can be deleted – runtime property

**ssh\_settings:** # Can be deleted – often managed elsewhere

admin\_username: azureuser

ssh\_port: '50000'

**state: Running** # Can be deleted – current state, not needed for IaC

type: computeinstance

**Q5 :**

1. Create your own Training Environment YAML file
2. Create your own Training Component YAML file

**A5 :**

1. **Tensorflow.yaml**

**$schema: https://azuremlschemas.azureedge.net/latest/environment.schema.json**

**description: Custom environment for Image Processing (with Pillow)**

**name: aml-Tensorflow-cli**

**version: 0.1.0**

**conda\_file: ../components/training/conda.yaml**

**image: mcr.microsoft.com/azureml/openmpi4.1.0-ubuntu20.04:latest**

**os\_type: linux**

**tags:**

**Pillow: 10.0.1**

**Tensorflow: 2.6.0**

1. **Training.yaml**

**$schema: https://azuremlschemas.azureedge.net/latest/commandComponent.schema.json**

**version: 0.1.0**

**type: command**

**name: training\_cli**

**display\_name: Training an AI model**

**description: Trains an AI model by inputting a lot of training and testing data.**

**code: ./code**

**environment: azureml:aml-Tensorflow-cli:0.1.0**

**command: 'python train.py**

**--training\_folder ${{inputs.training\_folder}} --testing\_folder ${{inputs.testing\_folder}}**

**--output\_folder ${{outputs.output\_folder}}**

**--epochs ${{inputs.epochs}}**

**'**

**inputs:**

**epochs:**

**optional: false**

**type: number**

**testing\_folder:**

**optional: false**

**type: uri\_folder**

**training\_folder:**

**optional: false**

**type: uri\_folder**

**outputs:**

**output\_folder:**

**type: uri\_folder**

**Q6 :** Output passing in YAML files + Pipeline input

1. How do we pass the output from the dataprep to the data\_split components?
2. How do we use the input defined in the pipeline input (train\_test\_split\_factor and epochs)

**A6 :**

**data\_split\_job:**

**component: ./components/dataprep/data-split.yaml**

**inputs:**

**animal\_1: ${{jobs.dataprep\_job.outputs.output\_data}} # Pass output from dataprep**

**animal\_2: <your\_animal\_2\_input>**

**animal\_3: <your\_animal\_3\_input>**

**train\_test\_split\_factor: ${{parent.inputs.train\_test\_split\_factor}} # Use pipeline input**

**outputs:**

**testing\_data: ${{outputs.testing\_data}}**

**training\_data: ${{outputs.training\_data}}**

**Q7 :** What is the full command to use?

**A7 :** az ml model download --name animal-classification --version 1 --resource-group mlops-demo --workspace-name singh-lovepreet-ml

**Q8 :** Can you find the line inside of the Training script/component/environment/pipeline … where the AI Model is being saved with the name animal-cnn.

**A8 :**

File name : train.py

Line number : 89

Extra explanation : The ModelCheckpoint callback saves the best model during training to the path specified by model\_path, which is constructed using model\_name = 'animal-cnn'. This ensures the model is saved in a directory or file named animal-cnn inside the output folder.

**Q9 :** Share your Dockerfile**.** What’s the size of the created Docker Image?Any optimisation recommendations to reduce the size of the image?

**A9 :**

File :

FROM python:3.9-slim

# Set environment variables for Python

ENV PYTHONDONTWRITEBYTECODE=1 \

    PYTHONUNBUFFERED=1 \

    PIP\_NO\_CACHE\_DIR=1 \

    PIP\_DISABLE\_PIP\_VERSION\_CHECK=1 \

    PIP\_DEFAULT\_TIMEOUT=100

# Set the working directory

WORKDIR /app

# Install system dependencies (for Pillow and TensorFlow)

RUN apt-get update && \

    apt-get install -y --no-install-recommends \

        build-essential \

        libglib2.0-0 \

        libsm6 \

        libxext6 \

        libxrender-dev \

        && rm -rf /var/lib/apt/lists/\*

# Copy only requirements first for better cache usage

COPY requirements.txt .

# Install Python dependencies

RUN pip install --upgrade pip && pip install -r requirements.txt

# Copy the rest of the application code

COPY . .

# Expose the port the app runs on

EXPOSE 8004

# Use environment variable for model path (improves reusability)

ENV MODEL\_PATH=/app/animal-classification/INPUT\_model\_path/animal-cnn

# Entrypoint for running the FastAPI app with reload option for development

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8004"]

Image size : 1.2 GB

Optimizations : Use smaller base image, remove build dependencies after install, multi-stage builds, remove pip cache.

**Q10 :** Provide a better solution using checks and if-statements.

**A10 :** Ensure the compute is only created if it does not already exist and only started if it exists. It also avoids redundant extension installs and configuration.

- name: Azure -- Create compute if not exists

        if: steps.check\_compute.outputs.exists == 'false'

        uses: azure/CLI@v2.1.0

        with:

          azcliversion: 2.64.0

          inlineScript: |

            az ml compute create --file ./environment/compute.yaml

      - name: Azure -- Start compute (cold-start compliant)

        # Always try to start, ignore error if already running

        uses: azure/CLI@v2.1.0

        with:

          azcliversion: 2.64.0

          inlineScript: |

            az ml compute start --name $COMPUTE\_NAME

        continue-on-error: true

      - name: Azure -- Register combined environment

        uses: azure/CLI@v2.1.0

        with:

          azcliversion: 2.64.0

          inlineScript: |

            az ml environment create --file ./environment/combined.yaml

**Q11 :**

1. What is the purpose of the needs: azure-pipeline ?
2. What’s the point of the actions/upload-artifact@v4.3.3 ?

**A11 :**

* needs: azure-pipeline  
  This makes sure the download job only starts after the azure-pipeline job finishes successfully.
* actions/[upload-artifact@v4.3.3](vscode-file://vscode-app/c:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)  
  This step uploads files (here, the inference folder) so you can use them later or in other jobs.

**Q12 :**

1. Paste the YAML file for your Kubernetes deployments
2. Paste the GitHub Actions pipeline step with the Kubectl commands

**A12 :**

1.

apiVersion: apps/v1

kind: Deployment

metadata:

name: animals-api

labels:

app: animals-api

spec:

replicas: 1

selector:

matchLabels:

app: animals-api

template:

metadata:

labels:

app: animals-api

spec:

containers:

- name: animals-api

image: ghcr.io/singhlovepreet/mlops-animals-api:latest

ports:

- containerPort: 8080

env:

- name: MODEL\_PATH

value: /model

volumeMounts:

- name: model-volume

mountPath: /model

volumes:

- name: model-volume

persistentVolumeClaim:

claimName: animals-model-pvc

---

apiVersion: v1

kind: Service

metadata:

name: animals-api-service

spec:

selector:

app: animals-api

ports:

- protocol: TCP

port: 80

targetPort: 8080

type: LoadBalancer

**2.**

- name: Set up Kubeconfig

uses: azure/aks-set-context@v3

with:

creds: ${{ secrets.AZURE\_CREDENTIALS }}

cluster-name: <your-aks-cluster-name>

resource-group: ${{ env.GROUP }}

- name: Deploy to Kubernetes

run: |

kubectl apply -f k8s/deployment.yaml

kubectl rollout status deployment/animals-api

Questions to answer for every lab

**What did you learn?**

*Fill in your three take aways that you learned during this lesson.*

1. How to build an end-to-end MLOps pipeline in github actions.
2. How to use Github secrets for variables.
3. How to utilize multi-stage workflows.

**Givethree interesting exam questions about the contents of the lab and/or the theory**.

*Thinking about this will make sure you remember the key take-aways and important details better and longer.*

1. Why would you use a combined environment.
2. What does the continue-on-error: true option do in a GitHub Actions step?
3. How does the workflow ensure that the correct version of the trained model is downloaded and included in the Docker image for deployment?

**Check the following:**

* I have made the entire lab assignment (be careful, some labs consist out of two or more Notion documents!).
* I have answered all the questions from the lab assignment.
* I have submitted my code as a zip file and/or as a link to a *public* Git repository.
* <For labs on Azure> I have shut down any resources that are in use, in order to avoid unexpected costs.