FILL IN : Lab 6

*Lovepreet Singh*

Questions from the lab

A screenshot of a computer

AI-generated content may be incorrect.*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).*

A screenshot of a computer

AI-generated content may be incorrect.

**Q1 :** What resources have been created, and what do they do according to the Azure documentation? Explain in your own words.

**A1 :** This is what was created

1. Storage account
   1. A place to store files and data in the cloud, like documents, images, logs, or even backups.
2. Key vault
   1. A secure locker for sensitive things like passwords, API keys, or connection strings. Apps can use these without hardcoding them.
3. Application insight
   1. A monitoring tool for your app. It shows if your app is working well, catches errors, and tracks usage.
4. Container registry
   1. A private storage for your Docker container images. Think of it as your own version of Docker Hub in Azure.

A screenshot of a computer

AI-generated content may be incorrect.This is the cheapest option possible that i could select.

A screenshot of a computer

AI-generated content may be incorrect.

Same for dogs and cats

**Q2 :** Can you give some extra benefits of using this Cloud Compute Instance?

**A2 :**

1. Pay Only When You Use It – No need to buy hardware.
2. Easy to Scale – Add more power or machines when needed.
3. Secure – Built-in protection for your data and apps.
4. Flexible – Choose your OS, tools, and customize it your way.
5. Works with Other Azure Services – Easy to connect with storage, databases, etc.
6. Remote Access – Use it from anywhere.

**Q3 :** Why is it a good idea to already use Environment variables, and not just variables to set in code? HINT: Think about reusability of your code, like in Docker Images.

**A3 :**

1. Reusable Code – You can run the same code in different places (like dev, test, or production) without changing it.
2. Safer – Keeps secrets (like passwords or API keys) out of your code.
3. Easier Setup – Just change settings outside the code; no need to touch the code itself.

**Q4 :** How to configure the Idle Shutdown into the Compute Instance created in the SDK?

**A4 :** You can change the idle shutdown in the compute tab and click on schedules, then you can configure the time or you can add the following code.

from azure.ai.ml.entities import ComputeInstance, AmlCompute

ci\_basic\_name = "cpu-lovepreet-auto"

ci\_basic = ComputeInstance(

name=ci\_basic\_name,

size="Standard\_E2ds\_v4",

idle\_time\_before\_shutdown="30" # Sets idle shutdown to 30 minutes

)

ml\_client.begin\_create\_or\_update(ci\_basic).result()

**Q5 :** Can you paste the content of this conda.yaml file into the right directory using Python code? Or using Magic Cells in Jupyter?

**A5 :**

Python:  
yaml\_content = """name: aml-Pillow

channels:

- conda-forge

dependencies:

- python=3.8

- numpy=1.21.2

- pip=21.2.4

- scikit-learn=0.24.2

- scipy=1.7.1

- pandas>=1.1,<1.2

- pip:

- inference-schema[numpy-support]==1.3.0

- xlrd==2.0.1

- mlflow==1.26.1

- azureml-mlflow==1.42.0

- Pillow==10.0.1

"""

# Write to file

with open('conda.yaml', 'w') as f:

f.write(yaml\_content)

print("conda.yaml file created successfully!")

**Magic cell:**

%%writefile conda.yaml

name: aml-Pillow

channels:

- conda-forge

dependencies:

- python=3.8

- numpy=1.21.2

- pip=21.2.4

- scikit-learn=0.24.2

- scipy=1.7.1

- pandas>=1.1,<1.2

- pip:

- inference-schema[numpy-support]==1.3.0

- xlrd==2.0.1

- mlflow==1.26.1

- azureml-mlflow==1.42.0

- Pillow==10.0.1

**Q6 :** Can you explain in your own words what this file (dataprep.py) does? What is the use of the bottom part ?

**A6 :** This script resizes all JPG images in a given input folder to a fixed size (64x64 pixels) and saves them to an output folder.

**Q7 :** Can you try to create a component using the following instructions?

**A7 :**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Q8 :** Submit this pipeline using a similar Python code we already used before. I did a few extra’s to make it as dynamic as we can possible make it at this point. You don’t need this, but it’s fun

**A8 :**

**Q9 :** Can you find the logs of the training AI component? Can you find the location of the saved AI Model?

**A9 : Logs of the training AI component**

1. Go to your Azure ML workspace
2. Navigate to "Jobs" → Select your pipeline job
3. Click on the specific "data\_split\_job" component
4. Select "Outputs + logs" tab
5. Look for these files:
   * std\_log.txt: Standard output/error logs
   * azureml-logs/\*.txt: Additional execution logs

**Saved model location**

1. In the datastorage

**Q10 :** Add a dynamic way to set the Seed, Initial Learning Rate, Batch Size, Patience and Model Name as extra inputs into the Training component. You can do so in the 'argparser', the 'component inputs', and the pipeline inputs.

**A10 :**

import argparse

def parse\_args():

parser = argparse.ArgumentParser()

# Existing arguments

parser.add\_argument("--training\_data", type=str, help="Path to training data")

parser.add\_argument("--validation\_data", type=str, help="Path to validation data")

parser.add\_argument("--output\_dir", type=str, help="Output directory for model and logs")

# New dynamic parameters

parser.add\_argument("--seed", type=int, default=42, help="Random seed for reproducibility")

parser.add\_argument("--initial\_learning\_rate", type=float, default=0.001,

help="Initial learning rate for the optimizer")

parser.add\_argument("--batch\_size", type=int, default=32,

help="Batch size for training and evaluation")

parser.add\_argument("--patience", type=int, default=5,

help="Number of epochs to wait before early stopping")

parser.add\_argument("--model\_name", type=str, default="custom\_model",

help="Name of the model for saving/identification")

return parser.parse\_args()

def main():

args = parse\_args()

# Use the parameters in your training logic

print(f"Training with configuration:")

print(f"Seed: {args.seed}")

print(f"Learning Rate: {args.initial\_learning\_rate}")

print(f"Batch Size: {args.batch\_size}")

print(f"Patience: {args.patience}")

print(f"Model Name: {args.model\_name}")

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 work with Azure CLI and GUI.
2. Model creation.
3. Working with cloud environments.

**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. Create a compute, pipeline and component using cli only.
2. How can you edit idle shutdown in a python file.
3. Give some upside and downside of cloud computing.

**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.