```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "fb054efb-c141-45dd-9274-17d34933da90",
   "metadata": {},
   "source": [
    "# Assignment3 : VM Storage Vs. S3 for Running an Image Recognition Application\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "ea1a76b3-60bc-42ec-84a6-c44abc041d4d",
   "metadata": {},
   "source": [
    "## Description\n",
    "\n",
    "The goal of this assignment is to understand the flexibility and cost-effective options for storing data in the cloud. Each VM in Azure, like in AWS, is equipped with ephemeral storage, which is directly attached to the VM but gets deleted upon VM termination. On the other hand, there is a persistent storage that can be attached to a VM (like Azure Disk Storage), but it can be costly. Additionally, there is the option of utilizing object store (like Azure Blob Storage) for storage. For instance, in a data analytics task, certain data should reside in Disk Storage to ensure it is quickly available. However, archival data can be stored in Blob Storage, prioritizing accessibility over performance. This assignment aims to help you understand the trade-off between data access, performance, and cost, allowing you to evaluate which factors are more crucial based on the requirements of your application.\n",
    "\n",
    "You are a data scientist who is tasked with running an ML application that uses a large dataset for training and testing. You have the option to use a VM to run the application, and you must decide whether to read/train on data stored locally from the VM or from data stored in an Azure Blob Storage. You also have the option of creating VM and Blob Storage inside an Azure Virtual Network (VNet) or in public cloud, and you should be able to choose the proper option based on the application requirements.\n",
    "\n",
    "In this assignment, you will compare the three following scenarios:\n",
    "\n",
    "- Running the target application on a VM while reading/training on the dataset stored locally in the VM.\n",
    "- Running the target application on a VM while reading/training on the
```

```
dataset stored in an Azure Blob Storage.\n",
    "- Running the target application on a VM while reading/training on the
dataset stored in an Azure Blob Storage, while VM and Blob Storage are
located inside a VNet.\n",
    "\n",
    "You will be required to write a report that compares the three
scenarios in terms of cost, scalability, data management, security, and
performance."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "3c10e153",
   "metadata": {},
   "source": [
    "## Scenario 1: Use of VM Storage for storing data \n",
    "\n",
    "In the first scenario, you need to run and test an image recognition
program on the VM with the dataset in VM storage. For simplicity, you can
use the same image recognition application that you used for assignment 2.
The application trains a DNN on the CIFAR-10 dataset, which consists of
60,000 32x32 color images, each belonging to one of 10 classes. Before
running the application, you need to store the CIFAR-10 dataset on the VM.
Make sure to choose an instance type that provides enough storage capacity
to accommodate the CIFAR-10 dataset. For example, you can select an
instance type with a large Disk Storage volume (1 GB). You can also mount
an additional Disk Storage volume to your current Azure VM. For simplicity,
we ignore the option of mounted persistent storage. You may want to store
your dataset in a mounted persistent storage to evaluate the cost overhead.
However, the rest of steps will be the same. In the next step, you should
store your dataset on VM storage, run the application and measure the
performance and cost.\n",
    "\n",
    "### 1- Download the CIFAR-10 dataset onto the VM\n",
    "\n",
    "The dataset can be downloaded and stored on the VM in two different
ways:\n",
    " \n",
    "A) Manually through the website\n",
    "\n",
    "- Go to the CIFAR-10 website at
https://www.cs.toronto.edu/~kriz/cifar.html\n",
    "\n",
    "- Scroll down to the \"Download\" section and click on the \"CIFAR-10
python version\" link.\n",
    "\n",
```

```
    "- This will download a file called \"cifar-10-python.tar.gz\" to your computer. Extract the contents of this file to a directory on your computer.\n",
    "\n",
    "- You should now have a directory named \"cifar-10-batches-py\" that contains the CIFAR-10 dataset in Python format.\n",
    "\n",
    "B) Automatically through commands\n",
    "Alternatively, you can use the following commands to download and extract the CIFAR-10 dataset:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 11,
   "id": "178ce028",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Download and extract the CIFAR-10 dataset\n",
    "!wget https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz\n",
    "!tar -xzf cifar-10-python.tar.gz"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "a361f252",
   "metadata": {},
   "source": [
    "This code will download the dataset and extract it to the current directory. "
   ]
  },
  {
   "cell_type": "markdown",
   "id": "70a85eab",
   "metadata": {},
   "source": [
    "### 2- Run an image recognition program on your VM\n",
    "\n",
    "In this step, you need to run an image recognition program on the VM and evaluate the performance and cost.\n"
   ]
  },
  {
   "cell_type": "code",
```

```
   "execution_count": null,
   "id": "09be4b78",
   "metadata": {
    "scrolled": true
   },
   "outputs": [],
   "source": [
    "import pickle\n",
    "import numpy as np\n",
    "import tensorflow as tf\n",
    "from tensorflow import keras\n",
    "\n",
    "# Load the dataset\n",
    "def unpickle(file):\n",
    "    with open(file, 'rb') as fo:\n",
    "        dict = pickle.load(fo, encoding='bytes')\n",
    "    return dict\n",
    "\n",
    "train_data = np.empty((0, 32*32*3))\n",
    "train_labels = []\n",
    "\n",
    "for i in range(1, 6):\n",
    "    batch = unpickle('cifar-10-batches-py/data_batch_' + str(i))\n",
    "    train_data = np.vstack((train_data, batch[b'data']))\n",
    "    train_labels += batch[b'labels']\n",
    "train_labels = np.array(train_labels)\n",
    "test_data = unpickle('cifar-10-batches-py/test_batch')[b'data']\n",
    "test_labels =
np.array(unpickle('cifar-10-batches-py/test_batch')[b'labels'])\n",
    "\n",
    "# Reshape and normalize the data\n",
    "train_data = train_data.reshape(-1, 32, 32, 3) / 255.0\n",
    "test_data = test_data.reshape(-1, 32, 32, 3) / 255.0\n",
    "\n",
    "# Define the model\n",
    "model = keras.Sequential([\n",
    "    keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu',
input_shape=(32, 32, 3)),\n",
    "    keras.layers.MaxPooling2D(pool_size=2),\n",
    "    keras.layers.Conv2D(filters=64, kernel_size=3,
activation='relu'),\n",
    "    keras.layers.MaxPooling2D(pool_size=2),\n",
    "    keras.layers.Flatten(),\n",
    "    keras.layers.Dense(units=128, activation='relu'),\n",
    "    keras.layers.Dense(units=10, activation='softmax')\n",
    "])\n",
```

```
    "\n",
    "model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])\n",
    "\n",
    "# Train the model\n",
    "model.fit(train_data, train_labels, epochs=10, batch_size=32,
validation_split=0.1)\n",
    "\n",
    "# Evaluate the model on the test data\n",
    "test_loss, test_acc = model.evaluate(test_data, test_labels)\n",
    "print('Test accuracy:', test_acc)\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "a5bc41b9",
   "metadata": {},
   "source": [
    "###  3- Evaluate the performance and cost\n",
    "\n",
    " The performance evaluation involves measuring the time taken to load
the data into memory as well as the time taken to train the model. You will
also need to calculate the cost of storing the data in the VM."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "bf3dcdd3",
   "metadata": {},
   "source": [
    "\n",
    "Answer the following questions:\n",
    "\n",
    "    A) How long did the execution (training and testing procedures)
take?\n",
    "\n",
    "    B) Did you face any issues running this program on your local
machine?\n",
    "    \n",
    "    c) How much did you pay for running and testing the
application?\n",
    " "
   ]
  },
  {
   "cell_type": "markdown",
```

```
   "id": "33429888",
   "metadata": {},
   "source": [
    "## Scenario 2: Use of Azure Blob Storage for storing data\n",
    "\n",
    "Next, you will create an Azure Blob Storage to store and access the CIFAR10 dataset on and from the Blob Storage. This can be done in two different ways:\n",
    "\n",
    "- Manually through the Azure portal\n",
    "- Automatically through your code\n",
    "\n",
    "To do so, you need to walk through the following steps:\n",
    "\n",
    "### 1- Create an Azure Blob Storage\n",
    "\n",
    "- Sign in to the Azure portal.\n",
    "\n",
    "- Click on \"Storage accounts\" from the left portal menu to display a list of your storage accounts.\n",
    "\n",
    "- Click on \"Create\" button to create a new storage account.\n",
    "\n",
    "- Provide the details for your storage account such as a unique name, performance tier, and redundancy configuration.\n",
    "\n",
    "- Once the storage account is created, expand it and select \"Blob Containers\", right-click and select \"Create Blob Container\". Provide a name and click \"OK\" to create the container.\n",
    "\n",
    "### 2- Store your dataset in Azure Blob Storage\n",
    "\n",
    "- After your Blob Container is created, click on it in the Azure portal to open the container page.\n",
    "\n",
    "- Click on the \"Upload\" button to upload the CIFAR-10 dataset to the container.\n",
    "\n",
    "- In the \"Upload blob\" dialog box, click on the \"Add files\" button and select the CIFAR-10 dataset files that you want to upload. You can select multiple files by holding down the Ctrl key.\n",
    "\n",
    "- After you have selected the files, click the \"Upload\" button to start the upload process.\n",
    "\n",
    "You have manually created an Azure Blob Storage and stored the
```

```
CIFAR-10 dataset in it. You can now access the dataset from your virtual
machine or any other machine that has access to your Blob Storage.\n",
    "\n",
    "You can also create the Azure Blob Storage, and store CIFAR10 dataset
automatically through a code and use it. To do so, you can use the Azure
SDK for Python, azure extension or any other Azure client library."
   ]
  },
  {
   "cell_type": "raw",
   "id": "3450a007",
   "metadata": {},
   "source": [
    "!pip install azure-mgmt-compute azure-mgmt-storage azure-mgmt-resource
azure-keyvault-secrets azure-storage-blob azure-identity"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "a27b9257",
   "metadata": {},
   "outputs": [],
   "source": [
    "from azure.storage.blob import BlobServiceClient, BlobClient,
ContainerClient\n",
    "from azure.identity import DefaultAzureCredential\n",
    "\n",
    "# Replace <storage-account-name> with the name of your storage
account\n",
    "# Replace <container-name> with the name you want to give to the
container\n",
    "storage_account_name = \"<storage-account-name>\"\n",
    "container_name = \"<container-name>\"\n",
    "\n",
    "# Get a credential object using CLI-based authentication\n",
    "credential = DefaultAzureCredential()\n",
    "\n",
    "# Construct the blob service client\n",
    "blob_service_client =
BlobServiceClient(account_url=f\"https://{storage_account_name}.blob.core.w
indows.net\", credential=credential)\n",
    "\n",
    "# Create a new container in the blob service\n",
    "container_client =
blob_service_client.create_container(container_name)"
```

```
   ]
  },
  {
   "cell_type": "markdown",
   "id": "4d6b039a",
   "metadata": {},
   "source": [
    "After you successfully created your Azure Blob Storage, you need to download and store the CIFAR-10 dataset on the Blob Storage. To do so, you can use the Azure SDK for Python along with the os module in Python.\n",
    "\n",
    "Note: You might have to add role \"Storage Blob Data Contributor\" to the storage account in order for the upload to work. You can do this by going to the storage account, clicking on \"Access Control (IAM)\" and then clicking on \"Add role assignment\". Select \"Storage Blob Data Contributor\" as the role and then select your user account (type in your email address) as the assignee. Click on \"Save\" to save the role assignment."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "0def88f9",
   "metadata": {},
   "outputs": [],
   "source": [
    "from azure.storage.blob import BlobServiceClient, BlobClient, ContainerClient\n",
    "import os\n",
    "\n",
    "# Replace <storage-account-name> and <container-name> with your storage account name and container name\n",
    "storage_account_name = \"<storage-account-name>\"\n",
    "container_name = \"<container-name>\"\n",
    "local_path = \"<local-path-to-dataset>\"\n",
    "local_file_name = \"<dataset-file-name>\"\n",
    "upload_file_path = os.path.join(local_path, local_file_name)\n",
    "\n",
    "# Create a blob client using the local file name as the name for the blob\n",
    "blob_client = blob_service_client.get_blob_client(container=container_name, blob=local_file_name)\n",
    "\n",
    "print(\"\\nUploading to Azure Storage as blob:\\n\\t\" +
```

```
local_file_name)\n",
    "\n",
    "# Upload the created file\n",
    "with open(upload_file_path, \"rb\") as data:\n",
    "    blob_client.upload_blob(data)\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "55abca35",
   "metadata": {},
   "source": [
    "To configure communication between your Azure VM and Blob Storage, you need to follow these steps:\n",
    "\n",
    "1. **Create a Network Security Group (NSG)**: An NSG contains a list of Access Control List (ACL) rules that allow or deny network traffic to your VM. You can create an NSG in the Azure portal.\n",
    "\n",
    "2. **Associate NSG to VM**: Once the NSG is created, you need to associate it with your VM. This can be done in the networking settings of your VM.\n",
    "\n",
    "3. **Add Inbound Security Rule**: In the NSG settings, add an inbound security rule to allow traffic from Azure Blob Storage. You need to specify the source as 'Any', destination as 'Any', destination port ranges as '443', and protocol as 'TCP'. This allows HTTPS traffic from Blob Storage.\n",
    "\n",
    "4. **Configure Azure Storage Firewalls and Virtual Networks**: In the settings of your Azure Storage account, go to the 'Networking' tab. Here, you can configure the network access to your storage account. You can choose to allow access from 'All networks', 'Selected networks', or 'Private endpoint'.\n",
    "\n",
    "5. **Create a Private Endpoint (Optional)**: If you want to secure traffic between your VM and Blob Storage over a private link, you can create a private endpoint for your storage account. This assigns a private IP address from your virtual network to the storage account.\n",
    "\n",
    "6. **Update Application Settings**: Finally, update your application running on the VM to use the Blob Storage account.\n",
    "\n",
    "Remember, modifying network security group rules will only affect new connections. Existing connections are not reevaluated with the new rules.\n",
```

```
    "\n",
    "Please note that these steps require you to have appropriate permissions in your Azure account. If you don't have the necessary permissions, please contact your Azure administrator."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "4df58a49",
   "metadata": {},
   "source": [
    " ### 4-Configure  and run the application code\n",
    " \n",
    "To configure your application code to read the dataset from Azure Blob Storage, you can use the Azure SDK for Python. Here’s an example of how to modify your code to load the training and testing data from Azure Blob Storage:\n",
    "\n",
    "Note: In the connection string url, provide the connection string of your storage account. You can find the connection string in the 'Access keys' section of your storage account. Alternatively you can also load the dataset in datastores (under machine learning extension of azure) and use it."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "f8464f55",
   "metadata": {},
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "import tensorflow as tf\n",
    "from tensorflow import keras\n",
    "from azure.storage.blob import BlobServiceClient\n",
    "import pickle\n",
    "import io\n",
    "\n",
    "# Create a BlobServiceClient object\n",
    "blob_service_client = BlobServiceClient.from_connection_string(\"<DefaultEndpointsProtocol...>\")\n",
    "\n",
    "\n",
    "# Define the container and blob names\n",
```

```
    "container_name = '<blob-container-name>'\n",
    "\n",
    "train_data_blob_name = 'train_data.pkl'\n",
    "train_labels_blob_name = 'train_labels.pkl'\n",
    "test_data_blob_name = 'test_data.pkl'\n",
    "test_labels_blob_name = 'test_labels.pkl'\n",
    "\n",
    "# Load the dataset from Azure Blob Storage\n",
    "def load_data(container_name, blob_name):\n",
    "    blob_client = blob_service_client.get_blob_client(container_name,
blob_name)\n",
    "    blob_data = blob_client.download_blob().readall()\n",
    "    return pickle.loads(blob_data)\n",
    "\n",
    "train_data = load_data(container_name, train_data_blob_name)\n",
    "train_labels = load_data(container_name, train_labels_blob_name)\n",
    "test_data = load_data(container_name, test_data_blob_name)\n",
    "test_labels = load_data(container_name, test_labels_blob_name)\n",
    "\n",
    "train_labels = np.array(train_labels)  # Convert train_labels to a
NumPy array\n",
    "\n",
    "# Reshape and normalize the data\n",
    "train_data = train_data.reshape(-1, 32, 32, 3) / 255.0\n",
    "test_data = test_data.reshape(-1, 32, 32, 3) / 255.0\n",
    "\n",
    "# Define the model\n",
    "model = keras.Sequential([\n",
    "    keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu',
input_shape=(32, 32, 3)),\n",
    "    keras.layers.MaxPooling2D(pool_size=2),\n",
    "    keras.layers.Conv2D(filters=64, kernel_size=3,
activation='relu'),\n",
    "    keras.layers.MaxPooling2D(pool_size=2),\n",
    "    keras.layers.Flatten(),\n",
    "    keras.layers.Dense(units=128, activation='relu'),\n",
    "    keras.layers.Dense(units=10, activation='softmax')\n",
    "])\n",
    "\n",
    "model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])\n",
    "\n",
    "# Train the model\n",
    "model.fit(train_data, train_labels, epochs=10, batch_size=32,
validation_split=0.1)\n",
    "\n",
```

```
    "# Evaluate the model on the test data\n",
    "test_loss, test_acc = model.evaluate(test_data, test_labels)\n",
    "#print('Test accuracy:', test_acc)\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "97d95d62",
   "metadata": {},
   "source": [
    "### 5- Evaluate the performance and cost\n",
    "\n",
    "Answer the following questions:\n",
    "\n",
    "     A) How long did the execution (training and testing procedures) take on your Azure services?\n",
    "\n",
    "     B) How does the VM's storage capacity compare to the Azure Blob Storage's capacity?\n",
    "     \n",
    "     C) How much did you pay for running and testing the application using Azure Blob Storage? How does the cost of storage and data transfer for the VM compare to the cost of using Azure Blob Storage? To compare, calculate the performance-cost ratio in each case."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "498cf66c",
   "metadata": {},
   "source": [
    "## Scenario 3: Establishing VM and Blob Storage communication inside a Virtual Network (VNet)\n",
    "\n",
    "A VNet can be used for communication between a VM and a Blob Storage. While a VNet is not strictly required for such communication, it is generally recommended to use a VNet to provide an additional layer of security and isolation for your resources.\n",
    "\n",
    "When you create a VNet, you can specify the IP address range for the VNet, subnets, and routing tables. You can then launch your Azure VM within a subnet in the VNet and configure the necessary Network Security Groups (NSGs) to control inbound and outbound traffic to and from the instance.\n",
    "\n",
    "By using a VNet, you can also enable service endpoints for Blob
```

Storage, which provide a secure and private connection to Blob Storage from within your VNet. Service endpoints for Blob Storage allow you to route traffic to Blob Storage without going over the internet, which can help improve security and reduce data transfer costs.\n",
    "\n",
    "In this task, first you will set up a VNet and create the VM and Blob Storage inside the VNet. Secondly, you will run and test the image recognition application using data stored on Blob Storage inside the VNet. Finally, you evaluate the cost and performance.\n",
    "\n",
    "Here are the steps to create a VNet, an Azure VM, and a Blob Storage, along with the necessary configurations:"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "862f2ea8",
   "metadata": {},
   "source": [
    "### 1- Create a Virtual Network (VNet)\n",
    "A VNet is a fundamental building block for your private network in Azure. It provides isolation, segmentation, and other network layer benefits. It’s the first step because it provides the network infrastructure in which your other resources (like VMs and Blob Storage) will reside.\n",
    "\n",
    "- Open the Azure portal at https://portal.azure.com/.\n",
    "- In the left-hand menu, click on \"Create a resource\".\n",
    "- In the \"Search the Marketplace\" box, type \"Virtual Network\" and select it.\n",
    "- Click on \"Create\" and fill in the required details such as Name, Address space (e.g., 10.0.0.0/16), etc.\n",
    "- Click on \"Review + create\" and then \"Create\".\n",
    "\n",
    "### 2- Create a Subnet\n",
    "A subnet is a range within your VNet. You can have multiple subnets within a VNet, and each subnet can contain a different set of resources. This allows for further segmentation and isolation within your VNet.\n",
    "\n",
    "- In the Azure portal, navigate to the VNet you created.\n",
    "- Select \"Subnets\" from the left-hand menu.\n",
    "- Click on \"+ Subnet\" and fill in the Name and Address range (CIDR block) (e.g., 10.0.1.0/24).\n",
    "- Click on \"OK\".\n",
    "\n",
    "### 3- Create a Public IP\n",

```
    "A Public IP address is an IP address that’s reachable from the
internet. If your VM needs to be accessible from the internet (for example,
if it’s hosting a web server), it will need a Public IP address.\n",
    "\n",
    "- In the Azure portal, click on \"Create a resource\".\n",
    "- Search for \"Public IP address\" and select it.\n",
    "- Click on \"Create\" and fill in the required details.\n",
    "- Click on \"Review + create\" and then \"Create\".\n",
    "\n",
    "### 4- Configure Routing\n",
    "Routing controls how traffic flows between subnets within your VNet,
between VNets, and between your VNet and the internet. Proper routing
configuration is crucial for network functionality and security.\n",
    "\n",
    "To create a route table:\n",
    "- In the Azure portal, click on “Create a resource”.\n",
    "- Search for “Route table” and select it.\n",
    "- Click on “Create” and fill in the required details.\n",
    "- Click on “Review + create” and then “Create”.\n",
    "\n",
    "Now configure the route table:\n",
    "- In the Azure portal, navigate to the VNet you created.\n",
    "- Select \"Subnets\" from the left-hand menu.\n",
    "- Click on the subnet you created.\n",
    "- In the \"Route table\" field, select the route table.\n",
    "- Click on \"+ Associate\" and select the route table you created to
associate with this subnet.\n",
    "\n",
    "\n",
    "### 5- Create a Network Security Group (NSG)\n",
    "An NSG contains a list of security rules that allow or deny network
traffic to resources connected to Azure VNets. These rules can be based on
source and destination IP address, port, and protocol. This step is
important for controlling access to your VM and Blob Storage.\n",
    "\n",
    "- In the Azure portal, click on \"Create a resource\".\n",
    "- Search for \"Network Security Group\" and select it.\n",
    "- Click on \"Create\" and fill in the required details.\n",
    "- Click on \"Review + create\" and then \"Create\".\n",
    "\n",
    "### 6- Launch an Azure VM\n",
    "The VM is where your application will run. It needs to be connected to
the VNet (and specifically to one of the subnets) so that it can
communicate with other resources like Blob Storage. The VM also needs a
Public IP if it should be reachable from the internet, and it should be
associated with the NSG to ensure it’s protected by the security rules.\n",
```

```
    "\n",
    "- In the Azure portal, click on \"Create a resource\".\n",
    "- Search for \"Virtual Machine\" and select it.\n",
    "- Click on \"Create\" and fill in the required details such as Image, Size, Username, Password, etc.\n",
    "- Under \"Networking\", select the VNet, Subnet, Public IP, and NSG you created earlier.\n",
    "- Click on \"Review + create\" and then \"Create\".\n",
    "\n",
    "### 7- Create a Blob Storage\n",
    "Blob Storage is where you'll store your data. Like the VM, it should be associated with the VNet for connectivity and with the NSG for security.\n",
    "\n",
    "- In the Azure portal, click on \"Create a resource\".\n",
    "- Search for \"Storage account\" and select it.\n",
    "- Click on \"Create\" and fill in the required details.\n",
    "- Click on \"Review + create\" and then \"Create\".\n",
    "\n",
    "### 8 - Run your image recognition program on your VM using data stored in Blob Storage\n",
    "\n",
    "### 9 - Evaluate performance and cost\n",
    "Azure provides various tools to monitor performance such as Azure Monitor and Log Analytics. You can use these tools to track metrics like CPU usage, disk operations, network traffic etc."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "f418cf54",
   "metadata": {},
   "source": [
    "As always, remember to turn off your instances when not in use."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "197a49ae",
   "metadata": {},
   "source": [
    "## Deliverables:\n",
    "\n",
    "A report comparing the three scenarios that includes the following:\n",
    "- An introduction that outlines the task and background
```

```
information\n",
    "- A comparison of the cost of local storage versus an Azure Blob
Storage\n",
    "- A comparison of the performance of the application when
reading/training on data stored locally on the VM versus stored in Azure
Blob Storage\n",
    "- A comparison of the network security features of a VM-Blob Storage
communications inside and outside a VNet and the associated cost.\n",
    "- A conclusion that summarizes the findings and makes a recommendation
for which scenario to use for the image recognition application from
scalability, performance, cost, and security perspectives."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "67bef2f9",
   "metadata": {},
   "source": [
    "# Good luck!"
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.13"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```