```
{
 "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 assignment3 is to understand the flexibility and cost-effective options for storing data in cloud. \n",
    "Each VM 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 Amazon EBS), but it can be costly. Additionally, there is the option of utilizing object store (like S3 bucket) for storage. For instance, in a data analytics task, certain data should reside in EBS to ensure it is quickly available. However, archival data can be stored in S3, 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 S3 bucket. You have also the option of creating VM and S3 bucket inside a Virtual Private Cloud (VPC) 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 S3 bucket, while VM and S3 bucket are located inside a
```

```
VPC.\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.\n",
    "\n",
    "## Prerequisites:\n",
    "<b>Complete this Assignment in Zone: us-west-2</b>\n",
    " <br/>\n",
    "<br/>We have predefined cost tags. Therefore the only cost tags that will work on your account are: <br/> - Tag: cpsc436:cost Value: scenario1 <br/> - Tag: cpsc436:cost Value: scenario2 <br/><br/> Watch the following video to see how to track costs. <br/> <br/> <a href='https://www.loom.com/share/3dfeb5dae63f4b9ba17d363c5b091dcd?sid=77c96995-de51-4893-81e0-5b204d6616e4'>https://www.loom.com/share/3dfeb5dae63f4b9ba17d363c5b091dcd?sid=77c96995-de51-4893-81e0-5b204d6616e4 </a>"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "3c10e153",
   "metadata": {},
   "source": [
    "\n",
    "\n",
    "## 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 CIFAR10 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 EBS volume (1 GB). You can also mount an additional EBS volume to your current EC2 instance. 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 - Create a VM and Connect to VM's Terminal\n",
    "\n",
    "Create a VM with the same settings as the last assignment. Also ensure
```

```
it is at least t2.xLarge.\n",
    "\n",
    "We will access the VM's terminal without SSHing and view the terminal
through the internet:\n",
    "\n",
    " - Click on the dropdown \"Actions\" (top right corner). Then click on
\"Connect\"\n",
    "\n",
    " - You will be transported to a new screen. You should already be on
the tab \"EC2 Instance Connect\". If not Click on that tab .\n",
    "\n",
    " - Click the button \"Connect\" in the bottom right\n",
    "\n",
    " - You will now be able to directly interact with your VM's terminal
over the internet\n",
    "\n",
    "### 2 - Download the CIFAR10 dataset onto the VM\n",
    "\n",
    "The dataset can be downloaded and stored on the VM with the following
commands. \n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "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",
    "\n",
    "# Extract the files\n",
    "tar -xzf cifar-10-python.tar.gz"
   ]
  },
  {
   "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",
    "- To create the file on your VM run:"
```

```
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "78a51df7",
 "metadata": {},
 "outputs": [],
 "source": [
  "nano scenario1_image-recognition.py"
 ]
},
{
 "cell_type": "markdown",
 "id": "0690e7d0",
 "metadata": {},
 "source": [
  "- Copy the file into the text editor that is open"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "09be4b78",
 "metadata": {
  "scrolled": true
 },
 "outputs": [],
 "source": [
  "# scenario1_image-recognition.py\n",
  "\n",
  "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": "bf116c9d",
   "metadata": {},
   "source": [
    "Run the script with:\n"
   ]
  },
  {
```

```
   "cell_type": "code",
   "execution_count": null,
   "id": "3336bb12",
   "metadata": {},
   "outputs": [],
   "source": [
    "python scenario1_image-recognition.py"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "455fc6ca",
   "metadata": {},
   "source": [
    "You may have to wait a few minutes for the script to complete. Once it
is done, you will see the accuracy of the model on the test set."
   ]
  },
  {
   "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": [
    "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?
"
   ]
```

```
  },
  {
   "cell_type": "markdown",
   "id": "33429888",
   "metadata": {},
   "source": [
    "## Scenario 2: Establishing VM and S3 communication inside a Virtual Private Cloud (VPC)\n",
    " \n",
    "A VPC can be used for communication between a VM and an S3 bucket. While a VPC is not strictly required for such communication, it is generally recommended to use a VPC to provide an additional layer of security and isolation for your resources.\n",
    "\n",
    "When you create a VPC, you can specify the IP address range for the VPC, subnets, and routing tables. You can then launch your EC2 instance within a subnet in the VPC and configure the necessary security groups and network ACLs to control inbound and outbound traffic to and from the instance.\n",
    "\n",
    "By using a VPC, you can also enable VPC endpoints for S3, which provide a secure and private connection to S3 from within your VPC. VPC endpoints for S3 allow you to route traffic to S3 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 VPC and create the VM and S3 bucket inside the VPC. Secondly, you will run and test the image recognition application using data stored on S3 inside the VPC. Finally, you evaluate the cost and performance.  \n",
    "\n",
    "### <b>1- Setup S3 Bucket </b>\n",
    "### 1.0- Create an S3 bucket\n",
    " \n",
    "- Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.\n",
    "\n",
    "- Click the \"Create bucket\" button.\n",
    "\n",
    "- Select the region you want to create your bucket in\n",
    "\n",
    "- In the \"Create bucket\" dialog box, enter a unique name for your bucket\n",
    "\n",
    "- Ensure public access is blocked (Block all public access is checked)\n",
```

```
    "\n",
    "- Click \"Create Bucket\".\n",
    "\n",
    "### 1.1- Manually Download CIFAR-10 Dataset\n",
    "\n",
    "Manually download the CIFAR-10 dataset through your local machine so
you can upload it to the S3 bucket.\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 folder named \"cifar-10-batches-py\" that
contains the CIFAR-10 dataset in Python format.\n",
    "\n",
    "### 1.2- Store your dataset on S3 bucket\n",
    "\n",
    "- After your bucket is created, click on it in the S3 console to open
the bucket page.\n",
    "\n",
    "- Click on the \"Upload\" button to upload the CIFAR-10 dataset to the
bucket.\n",
    "\n",
    "- In the \"Upload\" dialog box, click on the \"Add files\" button and
select the CIFAR-10 dataset folder \n",
    "\n",
    "- Click the \"Upload\" button to start the upload process.\n",
    "\n",
    "- After the files have been uploaded, you can view them in the S3
console by clicking on the bucket and navigating to the folder where you
uploaded the files.\n",
    "\n",
    "You have manually created an S3 bucket and stored the CIFAR-10 dataset
on it. You can now access the dataset from your virtual machine or any
other machine that has access to your S3 bucket. \n",
    "<br/>   \n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "### Additional Information (no action required) - You can create an S3
```

```
bucket with Code \n",
    "You can also create the S3 bucket, and store CIFAR10 dataset
automatically through a code and use it. To do so, you can use the AWS SDK
for Python (Boto3), or any other S3 client library."
   ]
  },
  {
   "cell_type": "raw",
   "id": "3450a007",
   "metadata": {},
   "source": [
    "!pip install boto3"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "a27b9257",
   "metadata": {},
   "outputs": [],
   "source": [
    "import boto3\n",
    "\n",
    "# Replace <bucket-name> with the name you want to give to the
bucket\n",
    "bucket_name = \"<my-first-bucket>\"\n",
    "\n",
    "s3 = boto3.client(\"s3\")\n",
    "s3.create_bucket(Bucket=bucket_name)\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "4d6b039a",
   "metadata": {},
   "source": [
    "After you have successfully created your S3 bucket, you need to
download and store the CIFAR-10 dataset on the S3 bucket. To do so, you can
use the awscli command-line tool along with the os module in Python:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "0def88f9",
   "metadata": {},
```

```
   "outputs": [],
   "source": [
    "import os\n",
    "\n",
    "# Download the CIFAR-10 dataset\n",
    "os.system(\"wget
https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz\")\n",
    "\n",
    "# Replace <path-to-awscli> with the path to the awscli executable\n",
    "# Replace <bucket-name> with the name of the S3 bucket you created\n",
    "os.system(\"<path-to-awscli> s3 cp cifar-10-python.tar.gz
s3://<bucket-name>/cifar-10-python.tar.gz\")\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "862f2ea8",
   "metadata": {},
   "source": [
    "### <b>2- Setup Virtual Private Network </b>\n",
    "### 2.0- Create a VPC\n",
    "\n",
    "- Open the Amazon VPC console at
https://console.aws.amazon.com/vpc/.\n",
    "- Choose \"Your VPCs\" from the navigation pane.\n",
    "- Choose \"Create VPC\", and enter the following details:\n",
    "  - Name tag: A descriptive name for your VPC.\n",
    "  - IPv4 CIDR block: A range of IP addresses for your VPC (e.g.
10.0.0.0/16).\n",
    "- Once finished, click on \"Create VPC\".\n",
    "\n",
    "### 2.1- Create a Subnet:\n",
    "\n",
    "- Choose \"Subnets\" from the left navigation pane.\n",
    "- Choose \"Create subnet\", and enter the following details:\n",
    "  - Name tag: A descriptive name for your subnet.\n",
    "  - VPC: Select the VPC you created in step 1.\n",
    "  - Availability Zone: Select an availability zone in which to create
the subnet.\n",
    "  - IPv4 CIDR block: A range of IP addresses for your subnet (e.g.
10.0.1.0/24).\n",
    "- Once finished, click on \"Create subnet\".\n",
    "\n",
    "### 2.2- Create an Internet Gateway:\n",
    "\n",
    "- Choose \"Internet Gateways\" from the left navigation pane.\n",
```

```
    "- Enter a name for your gateway, and click on \"Create internet
gateway\".\n",
    "- Choose \"Internet Gateways\" from the left navigation pane, again.
There you will see your new internet gateway.\n",
    "- Click on the row corresponding to your VPC, it will be highlighted
and marked as selected.\n",
    "- In the top right click on the dropdown \"Actions\" and choose
\"Attach to VPC\", then select the VPC that you have created in step 1.\n",
    "\n",
    "### 2.3- Configure Routing:\n",
    "\n",
    "- Choose \"Route Tables\" from the left navigation pane.\n",
    "- You will be transported to a page with table of routes\n",
    "- In the table, you'll see a column titled \"VPC.\" Drag the edges of
this column to enlarge it, which will help you find the row that
corresponds to your VPC\n",
    "- After locating and clicking on the row corresponding to your VPC, it
will be highlighted and marked as selected.\n",
    "- After clicking on the row corresponding to your VPC, details about
the route will be displayed below.\n",
    "- Click on the \"Routes\" tab, then choose \"Edit routes\".\n",
    "- Add a new route with the following settings:\n",
    "  - Destination: 0.0.0.0/0\n",
    "  - Target: Choose Internet Gateway. Then select the internet gateway
created in step 3.\n",
    "- When finished, click on \"Save\".\n",
    "\n",
    "### 2.4- Create a Security Group:\n",
    "You will be adding a new rules to allow traffic from your VM to your
S3 bucket. The rule should allow traffic from the IP range used by the S3
service (52.95.0.0/16 and 52.219.0.0/16). You should also specify the
protocol and port(s) that you want to allow (for example, TCP on port 443
for HTTPS traffic).\n",
    "\n",
    "- Choose \"Security Groups\" from the left navigation pane.\n",
    "- Choose \"Create security group\", and enter the following
details:\n",
    "  - Name tag: A descriptive name for your security group.\n",
    "  - Description: A brief description of your security group.\n",
    "  - VPC: Select the VPC you created in step 1.\n",
    "  - Create Inbound Rules:\n",
    "    - Rule 1:  Type: SSH,        Port Range: 22,      IP(magnifying
glass): 0.0.0.0/0\n",
    "    - Rule 2: Type: Custom TCP, Port Range: 0 - 444, IP:
52.95.0.0/16\n",
    "    - Rule 3: Type: Custom TCP, Port Range: 0 - 444, IP:
```

```
52.219.0.0/16\n",
    "<br/>\n",
    "- When finished, click on \"Create security group\"."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "d9eb9440",
   "metadata": {},
   "source": [
    "### <b>3- Setup EC2 Instance </b>\n",
    "### 3.0- Launch an EC2 Instance:\n",
    "\n",
    "- Choose \"Instances\" from the navigation pane.\n",
    "- Choose \"Launch instance\"\n",
    "\n",
    "- Choose the AMI and instance type:\n",
    "  - The AMI should be the same as the one you used earlier in the
assignment (same as previous assignment as well)\n",
    "  - Choose an instance type of t2.micro and later you will need to
upgrade it to t2.xLarge (if unsure, follow steps from the previous
assignment to upgrade from t2.micro -> t2.xlarge)\n",
    "\n",
    "\n",
    "- On the \"Network Settings\" Tab, click on Edit\n",
    "  - Select the VPC you've previously created\n",
    "  - Select the Subnet you've previously created\n",
    "  - !!!VERIFY!!! -> Auto-assign public IP: Enable\n",
    "  - Choose \"Select existing security group\" and select the security
group that you have previously created\n",
    "\n",
    "\n",
    "- Launch the instance.\n",
    "\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "4df58a49",
   "metadata": {},
   "source": [
    " ### 3.1- Run the image recognition program on your VM using data
stored in S3 bucket.\n",
    " \n",
    "- Connect to your VM terminal in the browser like done previously\n",
    "- Run \"aws configure...\" with the script below in your VM's
```

```
terminal\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "eda42990",
   "metadata": {
    "vscode": {
     "languageId": "shellscript"
    }
   },
   "outputs": [],
   "source": [
    "### run these on your VM\n",
    "aws configure \n",
    "### default region: (the one you're currently using eg. us-west-2)
\n",
    "### default output format: json\n",
    "\n",
    "aws configure set aws_session_token <your-access-token>"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "5624df71",
   "metadata": {},
   "source": [
    "- Upload the image recognition script below (that accesses your S3
bucket). Ensure to change the <b>bucket_name </b> variable to your actual
bucket name.\n",
    "- You may then run your script"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "f8464f55",
   "metadata": {},
   "outputs": [],
   "source": [
    "# scenario2_image-recognition.py\n",
    "\n",
    "import numpy as np\n",
    "import tensorflow as tf\n",
    "from tensorflow import keras\n",
```

```
    "import boto3\n",
    "import pickle\n",
    "\n",
    "# your-bucket-name\n",
    "# └──cifar-10-batches-py\n",
    "#     ├── batches.meta\n",
    "#     ├── data_batch_1\n",
    "#     ├── data_batch_2\n",
    "#     ├── data_batch_3\n",
    "#     ├── data_batch_4\n",
    "#     ├── data_batch_5\n",
    "#     ├── readme.html\n",
    "#     └── test_batch\n",
    "#\n",
    "# TODO: Change the bucket name\n",
    "bucket_name = 'your-bucket-name'\n",
    "\n",
    "\n",
    "s3 = boto3.client('s3')\n",
    "\n",
    "# Load the dataset from S3\n",
    "def load_data(bucket_name, file_name):\n",
    "    obj = s3.get_object(Bucket=bucket_name, Key=file_name)\n",
    "    data = obj['Body'].read()\n",
    "    return pickle.loads(data, encoding='bytes')\n",
    "\n",
    "train_data = np.empty((0, 32*32*3))\n",
    "train_labels = []\n",
    "\n",
    "# Load 5 batches of training data and one batch of test data from the
S3 bucket\n",
    "for i in range(1, 6):\n",
    "    batch = load_data(bucket_name, '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",
    "\n",
    "train_labels = np.array(train_labels)\n",
    "test_data = load_data(bucket_name,
'cifar-10-batches-py/test_batch')[b'data']\n",
    "test_labels = np.array(load_data(bucket_name,
'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": "97d95d62",
   "metadata": {},
   "source": [
    "### 3.2- Evaluate the performance and cost\n",
    " \n",
    " Answer the following questions:\n",
    "\n",
    "     A) How long did the execution (training and testing procedures)
take?\n",
    "\n",
    "     B) How does the VM's storage capacity compare to the S3 bucket's
storage capacity?\n",
    "\n",
    "     c) How much did you pay for running and testing the application
using S3? How does the cost of storage and data transfer for the VM compare
to the cost of using an S3 bucket? To compare, calculate the
performance-cost ratio in each case."
   ]
```

```
  },
  {
   "cell_type": "markdown",
   "id": "80092683",
   "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 2 scenarios that includes the following:\n",
    "- An introduction that outlines the task of the running the 2 scenarios and background information.\n",
    "- Screenshots showing the outcomes of scenario 1 and 2 after successfully executing the code in the terminal\n",
    "- A comparison of the cost and time to complete the function in local storage (scenario 1) versus an S3 bucket (scenario 2)\n",
    "- A comparison of the performance of the application when reading/training on data stored locally on the VM (scenario 1) versus stored in an S3 bucket (scenario 2)\n",
    "- A comparison of the network security features of a VM-S3 communications inside and outside a VPC and the associated cost (scenario 2).\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. \n"
   ]
  },
  {
   "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": "3.10.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```