```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "fb054efb-c141-45dd-9274-17d34933da90",
   "metadata": {},
   "source": [
    "# Assignment2 : VM Deployment for Image Recognition\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "ea1a76b3-60bc-42ec-84a6-c44abc041d4d",
   "metadata": {},
   "source": [
    "## Description\n",
    "  \n",
    "The goal of assignment2 is to learn how to deploy and run an application on a Virtual Machine (VM). Why not containers? Containers are lightweight, efficient, and fast to start, making them ideal for running microservices and other distributed applications. However, there are situations where it makes sense to move from containers to virtual machines. One reason is when you need stronger security and isolation between different applications or services. Virtual machines offer hardware-level isolation, which makes it harder for attackers to access sensitive data or applications. Another reason to use virtual machines is when you need to run legacy applications or those that require specific versions of operating systems (like MS Windows). In this case, virtual machines can provide a consistent environment for running these applications without having to worry about compatibility issues. \n",
    "  \n",
    "In this assignment, you will work with a Convolutional Neural Network (CNN) for image recognition that has complex dependencies and utilizes a large dataset. In the first step, you will attempt to run the application on your local host. A CNN model typically requires a large amount of data and computational power for training, making it difficult for many users to run them on their local machines. In the next step, you will explore how to deploy and run the same application on a virtual machine on a cloud platform, gaining insights into the differences between running it on a VM versus a local host. In this case, you will be using Azure Virtual Machines for the virtual machine architecture."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "3c10e153",
```

```
   "metadata": {},
   "source": [
    "## A. Run and test an image recognition program on your local machine\n",
    "\n",
    "\n",
    "    "For simplicity, a python script has been provided for you in this assignment. Here's a sample code for a convolutional neural network (CNN) for image recognition. This code trains a CNN on the CIFAR-100 dataset, which contains 50,000 training images. We repeat each image 10 times to increase the dataset size, resulting in a total of 500,000 training images. This larger dataset will require more memory for training.\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "4d5cf896",
   "metadata": {},
   "outputs": [],
   "source": [
    "\n",
    "import tensorflow as tf\n",
    "from tensorflow.keras import datasets, layers, models\n",
    "import matplotlib.pyplot as plt\n",
    "\n",
    "# Load the dataset\n",
    "(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()\n",
    "\n",
    "# Normalize pixel values to be between 0 and 1\n",
    "train_images, test_images = train_images / 255.0, test_images / 255.0\n",
    "\n",
    "# Define the CNN architecture\n",
    "model = models.Sequential()\n",
    "model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))\n",
    "model.add(layers.MaxPooling2D((2, 2)))\n",
    "model.add(layers.Conv2D(64, (3, 3), activation='relu'))\n",
    "model.add(layers.MaxPooling2D((2, 2)))\n",
    "model.add(layers.Conv2D(64, (3, 3), activation='relu'))\n",
    "model.add(layers.Flatten())\n",
    "model.add(layers.Dense(64, activation='relu'))\n",
    "model.add(layers.Dense(10))\n",
    "\n",
```

```
    "# Compile the model\n",
    "model.compile(optimizer='adam',\n",
    "
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),\n",
    "              metrics=['accuracy'])\n",
    "\n",
    "# Train the model\n",
    "history = model.fit(train_images, train_labels, epochs=10, \n",
    "                    validation_data=(test_images, test_labels))\n",
    "\n",
    "# Evaluate the model\n",
    "test_loss, test_acc = model.evaluate(test_images,  test_labels,
verbose=2)\n",
    "print('Test accuracy:', test_acc)\n",
    "\n",
    "# Plot the training and validation accuracy over time\n",
    "plt.plot(history.history['accuracy'], label='accuracy')\n",
    "plt.plot(history.history['val_accuracy'], label = 'val_accuracy')\n",
    "plt.xlabel('Epoch')\n",
    "plt.ylabel('Accuracy')\n",
    "plt.ylim([0.5, 1])\n",
    "plt.legend(loc='lower right')\n",
    "plt.show()\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "a5bc41b9",
   "metadata": {},
   "source": [
    "### 1- Set up the environment and dependencies\n",
    "\n",
    "- Install the required dependencies for the image recognition program,
such as tensorflow and matplotlib.\n",
    "- Set up the environment variables if required.\n",
    "\n",
    "### 2- Run the program on a local host\n",
    "\n",
    "- Open the terminal and navigate to the directory where the image
classification program is stored.\n",
    "- Run the program on your local host. \n",
    "- Verify the output generated by the program.\n",
    "\n",
    "##### Note for M1 Mac Users: \n",
    "If you are using a Mac with an M1 chip, you may encounter warnings
with the tensorflow installation. Something like \"The TensorFlow library
```

```
was compiled to use AVX instructions, but these aren't available on your machine\". This is because the current version of tensorflow is not optimized for the M1 chip. You can ignore these warnings for now, as they won't affect the execution of the program. If you want to get rid of this warning, you can change your tensorflow version to 2.6.0.\n",
    "\n",
    "### 3- Analyze the result\n",
    "Answer the following questions:\n",
    "     A) How long did the execution (training and testing procedures) take?\n",
    "     B) Did you face any issues running this program on your local machine? "
   ]
  },
  {
   "cell_type": "markdown",
   "id": "33429888",
   "metadata": {},
   "source": [
    "Here is the Azure version of your instructions:\n",
    "\n",
    "## B. Deploying the Program on a Virtual Machine\n",
    " \n",
    "When running this code on a local machine with limited resources, you might observe that the code runs into memory issues and may even crash due to insufficient memory to handle the increased dataset size. On the other hand, running the same code on a cloud-based virtual machine, which provides more generous resource allocations, may allow the code to execute successfully without running out of memory. In the next step, you will deploy and test the application on a virtual machine. To do so, you need to walk through the following steps:\n",
    "\n",
    "### 1- Containerize the application\n",
    " \n",
    "In the previous assignment, you gained an understanding of the benefits of containerizing a complex application, which not only provides more flexibility and control over the infrastructure, but also makes the application portable. To streamline the process of transferring the application to a virtual machine, you first need to containerize the application. Once the Docker image is created, it can be conveniently deployed on a virtual machine. \n",
    "\n",
    "### 2- Provision a virtual machine\n",
    "\n",
    "You need to provision a virtual machine with a supported operating system and Docker installed. This can be done on cloud platforms like
```

```
Azure. This step may be slightly different depending on the cloud platform that you work with. In the following, you will learn how to provision a VM on Azure.\n",
    "   \n",
    "- Sign in to your Azure account.\n",
    "- Create a Virtual Machine: For this code, you can use the Ubuntu Server 20.04 LTS image.\n",
    "- Choose an instance type: Azure offers a variety of instance types, each with varying amounts of CPU, memory, and storage resources. For this code, you can choose an Standard D8s v3 instance type, which has 8 vCPUs, 32 GB of memory.\n",
    "- Launch a VM instance: Once you have selected a VM and an instance type, launch a VM instance using the Azure portal or the Azure CLI.\n",
    "- Configure security groups: By default, VM instances are launched with restrictive security settings. Configure the security group to allow access to the required ports and protocols for your code to function correctly. For example, select the \"Allow selected ports\" option and add port 22 (SSH) and port 80 (HTTP) and port 443 (HTTPS) to the list of allowed ports.\n",
    "  \n",
    "### 3-Deploy the application on VM\n",
    " \n",
    "Transfer the Docker image to the virtual machine using a secure copy tool like SCP or SFTP. You can use the following command to save the Docker image as a tar file and transfer it to the virtual machine: For example, if the image file is named \"image-recognition.tar.gz\" and you copied it to the \"/tmp\" directory on the VM, you can use the following command:"
   ]
  },
  {
   "cell_type": "raw",
   "id": "230af40c",
   "metadata": {},
   "source": [
    "docker save image-recognition | gzip > 
/tmp/image-recognition.tar.gz\n",
    "scp /tmp/image-recognition.tar.gz <username>@<vm-ip>:/tmp"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "a7d2fd9b",
   "metadata": {},
   "source": [
    "Replace \"user\" with your username and \"vm_ip_address\" with the IP address of the VM.\n",
```

```
    "### 4- Run and test the application on VM\n",
    "\n",
    "To run a Docker image on a VM, you can follow these steps:\n",
    "\n",
    "- First, make sure that Docker is installed on the VM. You can check
this by running the following command:\n",
    "\n",
    "```\n",
    "docker --version\n",
    "```\n",
    "\n",
    "If Docker is not installed, you will need to install it before
proceeding. (https://docs.docker.com/engine/install/ubuntu/)\n",
    "- Open a terminal or command prompt on your local machine and type the
following command to connect to the virtual machine via SSH:\n",
    "```\n",
    "ssh user@<VM_IP_ADDRESS>\n",
    "```\n",
    "\n",
    "- Load the Docker image into Docker by running the following
command:\n",
    "\n",
    "```\n",
    "docker load < /tmp/image-recognition.tar.gz\n",
    "```\n",
    "\n",
    "This will load the Docker image into Docker on the VM.\n",
    "- Run the Docker container by running the following command:\n",
    "```\n",
    "docker run image-recognition\n",
    "```\n",
    "\n",
    "This will start a new Docker container based on the image you
loaded.\n",
    "\n",
    "- Test the application on the VM\n",
    "- Record the execution result"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "85b88ac8",
   "metadata": {},
   "source": [
    "### 5- Analyze the result\n",
    "\n",
```

```
    "Answer to 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 you local
machine?\n",
    " \n",
    " C) How running the application on VM differs from local host?\n",
    " \n",
    "## Deliverables:\n",
    "\n",
    "- A Dockerfile and a docker image of the image classification
application.\n",
    "\n",
    "- The execution result of your program when running on local host and
on a VM.\n",
    "\n",
    "- Document and compare application execution on local machine and on
the VM from performance perspective.\n",
    "- A report detailing the advantages and disadvantages of running a
heavy application on a local host and a VM.\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "261c436e",
   "metadata": {},
   "source": [
    "\n",
    "\n",
    "# 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": "3.9.16"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```