```
{
 "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 platforms, gaining insights into the differences between running it on a VM versus a local host."
   ]
  },
  {
   "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": [
    "# image-recognition.py\n",
    "\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()"
   ]
  },
  {
   "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"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "02c6ad96",
   "metadata": {},
   "source": [
    "### 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"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "bf3dcdd3",
   "metadata": {},
   "source": [
    "### 3- Analyze the result\n",
    "\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",
    " "
   ]
  },
  {
   "cell_type": "markdown",
   "id": "33429888",
   "metadata": {},
   "source": [
    " ##  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 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",
```

```
    " \n",
    "### 1- Containerize the application\n",
    "\n",
    "NAME YOUR DOCKER INSTANCE: image-recognition\n",
    " \n",
    "Containerize the application in the same format that you would have if you were deploying to play with docker in the previous assignment.\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",
    " \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 AWS, Azure, or GCP. 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 AWS.\n",
    " \n",
    "   \n",
    "  - Sign in your AWS account.\n",
    "\n",
    "  - In the top right click on your current zone (might currently say Canada Central). Change your zone to \"US West Oregon   US-West-2\". Ensure it is in this zone for the whole assignment.\n",
    "\n",
    "  - Create an Amazon Machine Image (AMI): For this code, you can use the Deep Learning AMI provided by AWS, which includes popular deep learning frameworks such as TensorFlow, PyTorch, and MXNet. <b>If available try: Deep Learning OSS Nvidia Driver AMI GPU TensorFlow 2.13.</b>\n",
    "\n",
    "  - Choose an instance type: t2.micro. We have limited permissions and can only create free tier EC2 instances on this screen.\n",
    "\n",
    "  - Create Key Pair Login. Choose RSA and PEM options. Once created it will automatically download into your browser. Save it in your active coding directory for later use. Don't put it in the same folder as your dockerfile.\n",
    "\n"
   ]
```

```
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "03fe048e",
   "metadata": {},
   "outputs": [],
   "source": [
    "#Go to the directory where your keypair is located and run the
command. This allows you to utilize your keypair to connect to the
instance.\n",
    "chmod 400 \"yourKeyPairLogin.pem\""
   ]
  },
  {
   "cell_type": "markdown",
   "id": "5fe762f4",
   "metadata": {},
   "source": [
    "\n",
    "  - Launch your EC2 instance\n",
    "### 2.1-Upgrade Your EC2 Instance - t2.micro -> t2.xlarge\n",
    "\n",
    "  Unfortunately we have to use a backdoor to use a better instance of
EC2 on our current servers.\n",
    "\n",
    "  - Once your instance is created - click on it and go to that
instance's dashboard\n",
    "\n",
    "  - Click on the dropdown \"Instance State\" (top right corner). Then
click on \"Stop Instance\"\n",
    "\n",
    "  - Click on the dropdown \"Actions\" (top right corner). Then click
on \"Instance Settings\". Then click on \"Change Instance Type\".\n",
    "\n",
    "  - You will be transported to another dashboard where you can choose
a better instance type. Choose t2.xlarge. Observe the higher costs. Then
press \"Apply\".\n",
    "  \n",
    "  - Click on the dropdown \"Instance State\" (top right corner). Then
click on \"Start Instance\"\n",
    "\n",
    "  - Now you are ready for the next step...\n",
    "  "
   ]
  },
```

```
  {
   "cell_type": "markdown",
   "id": "09993990-0d0a-4cff-8631-33032169cd62",
   "metadata": {},
   "source": [
    " ### 3-Deploy the application on VM\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "ce74a687",
   "metadata": {},
   "source": [
    " - Click on the dropdown \"Actions\" (top right corner). Then click on
\"Connect\"\n",
    " - You will be transported to a new screen. Click on the tab \"
Client\"\n",
    " - At the bottom of the panel their will be an \"Example\". <b>Copy
your example</b>. It should be similar to:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "fc1d80c3",
   "metadata": {},
   "outputs": [],
   "source": [
    "# DO NOT USE THIS ONE. \n",
    "# THERE IS AN EXAMPLE ON THE PAGE THAT AWS HAS PROVIDED\n",
    "ssh -i \"yourKeyPairLogin.pem\"
ec2-user@ec2-89-182-37-39.ca-central-1.compute.amazonaws.com"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "7a3389dc",
   "metadata": {},
   "source": [
    "- Ensure that your url (ec2-user@...) starts with 'ec2-user'. If it
starts with 'root', replace it with 'ec2-user' \n",
    "\n",
    "- In your terminal, navigate to the directory that
\"yourKeyPairLogin.pem\" is saved.\n",
    "\n",
    "- Paste your command (ssh -i ...) in your terminal and run it \n",
```

```
    "\n",
    "- Once you are connected, exit out of the SSH session. You will need to SSH into your VM later. "
   ]
  },
  {
   "cell_type": "markdown",
   "id": "e1cd45ca",
   "metadata": {},
   "source": [
    "#### Having Connection Issues?\n",
    "If you are having problems ensure that your public dns starts with \"ec2-user\" and not \"root\"\n",
    "- WRONG (eg. root@ec2-89-182-37-39.ca-central-1.compute.amazonaws.com)\n",
    "- RIGHT (eg. ec2-user@ec2-89-182-37-39.ca-central-1.compute.amazonaws.com)\n",
    "\n",
    "If you are having other authentication issues try:\n",
    "- aws configure (then put in your passwords)\n",
    "- aws configure set aws_session_token \"your session token without quotes\""
   ]
  },
  {
   "cell_type": "markdown",
   "id": "4b3fd485",
   "metadata": {},
   "source": [
    "### 3.1-Upload Docker File to VM"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "5f92502f",
   "metadata": {},
   "source": [
    " \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 into the /tmp directory on the VM.\n",
    "\n",
    " This might take 2 minutes to upload\n",
    "\n",
```

```
    " <b>MODIFY THE COMMAND BELOW</b>\n",
    " - replace ec2-user@12-34-46.ca-central-1.compute.amazonaws.com:/tmp/
with your info \n",
    " - replace yourKeyPairLogin.pem"
   ]
  },
  {
   "cell_type": "raw",
   "id": "230af40c",
   "metadata": {},
   "source": [
    "docker save image-recognition | gzip > image-recognition.tar.gz\n",
    "scp -i yourKeyPairLogin.pem image-recognition.tar.gz
ec2-user@12-34-46.ca-central-1.compute.amazonaws.com:/tmp/"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "b72e43c9",
   "metadata": {},
   "source": [
    "#### Issues Uploading?\n",
    "- Ensure you are in the same folder as your .pem key"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "096d302c",
   "metadata": {},
   "source": [
    "### 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",
    "- SSH into your VM like in step 3 (ssh -i ...)\n",
    "\n",
    "- Once you have SSH'ed into your VM make sure that Docker is installed
on the VM. You can check this by running the following command:"
   ]
  },
  {
   "cell_type": "raw",
   "id": "bdfd39d8",
   "metadata": {},
   "source": [
    "docker --version"
```

```
   ]
  },
  {
   "cell_type": "markdown",
   "id": "3acefa0f",
   "metadata": {},
   "source": [
    "If Docker is not installed, you will need to install it before
proceeding."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "ba452224",
   "metadata": {},
   "source": [
    "\n",
    "- Load the Docker image into Docker by running the following command
(change your file path based on the location of your file):"
   ]
  },
  {
   "cell_type": "raw",
   "id": "d3c2c969",
   "metadata": {},
   "source": [
    "docker load < /tmp/image-recognition.tar.gz"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "55645180",
   "metadata": {},
   "source": [
    "This will load the Docker image into Docker on the VM."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "da5e792b",
   "metadata": {},
   "source": [
    "- Run the Docker container by running the following command:"
   ]
  },
  {
```

```
   "cell_type": "raw",
   "id": "497f546b",
   "metadata": {},
   "source": [
    "docker run image-recognition"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "57825b14",
   "metadata": {},
   "source": [
    "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\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "85b88ac8",
   "metadata": {},
   "source": [
    "### 5- Analyze the result\n",
    "\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 you local
machine?\n",
    " \n",
    " C) How running the application on VM differs from local host?"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "31393e1d",
   "metadata": {},
   "source": [
    "## Deliverables:\n",
    "\n",
    "- A screenshot of the Dockerfile and a Docker image of the image
classification application.\n",
```

```
    "- A screenshot of the execution result of your program when running on the local host and on a VM (including the graph you will see on local host)\n",
    "- Document and compare application execution on the local machine and on the VM from a performance perspective. This also includes answering the questions in \"5-Analyze the result\"\n",
    "- A report detailing the advantages and disadvantages of running a heavy application on a local host and a VM.\n",
    "- Please delete your EC2 instances after you are complete - costs add up quickly with them"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "261c436e",
   "metadata": {},
   "source": [
    "\n",
    "\n",
    "# Good luck!"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "213e549a",
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "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": "3.11.5"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```