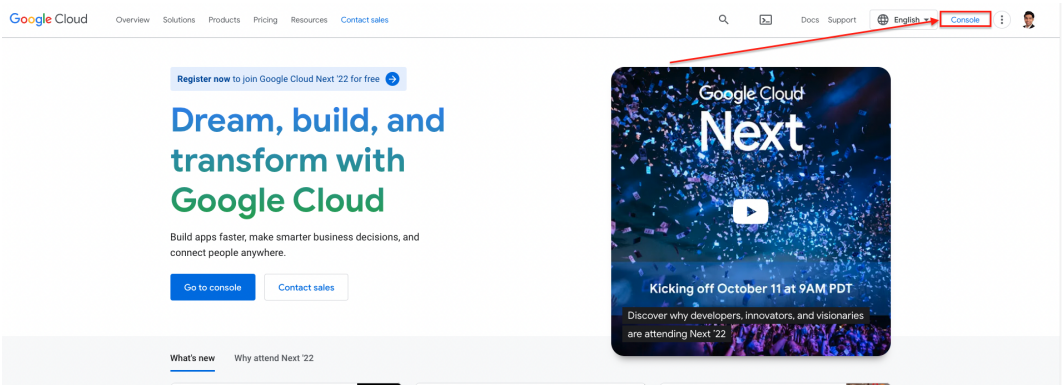
COMS 6998 - High Performance Machine Learning

Cloud and MKL Setup Instructions

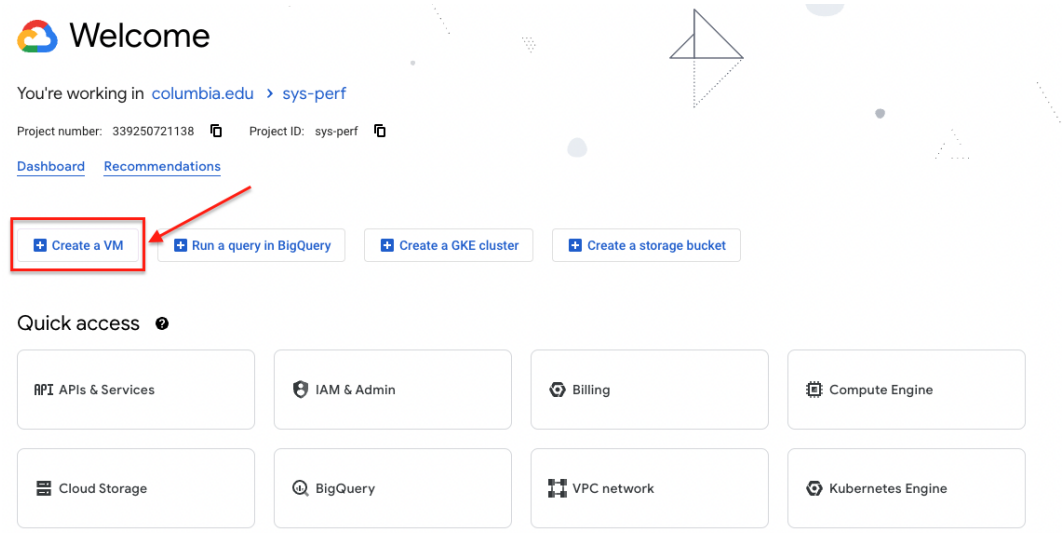
1 Google Cloud Setup

This setup assumes that you have a functional Google Cloud account and have used the credits provided by the course to create a billing account. You should also have a project linked to the billing account in which you can create VM instances. Refer the following link on how to create a project with a billing account **Creating Managing Projects**.

1. Go to the following link: cloud.google.com **and click on *Console* on the top right of the page.**

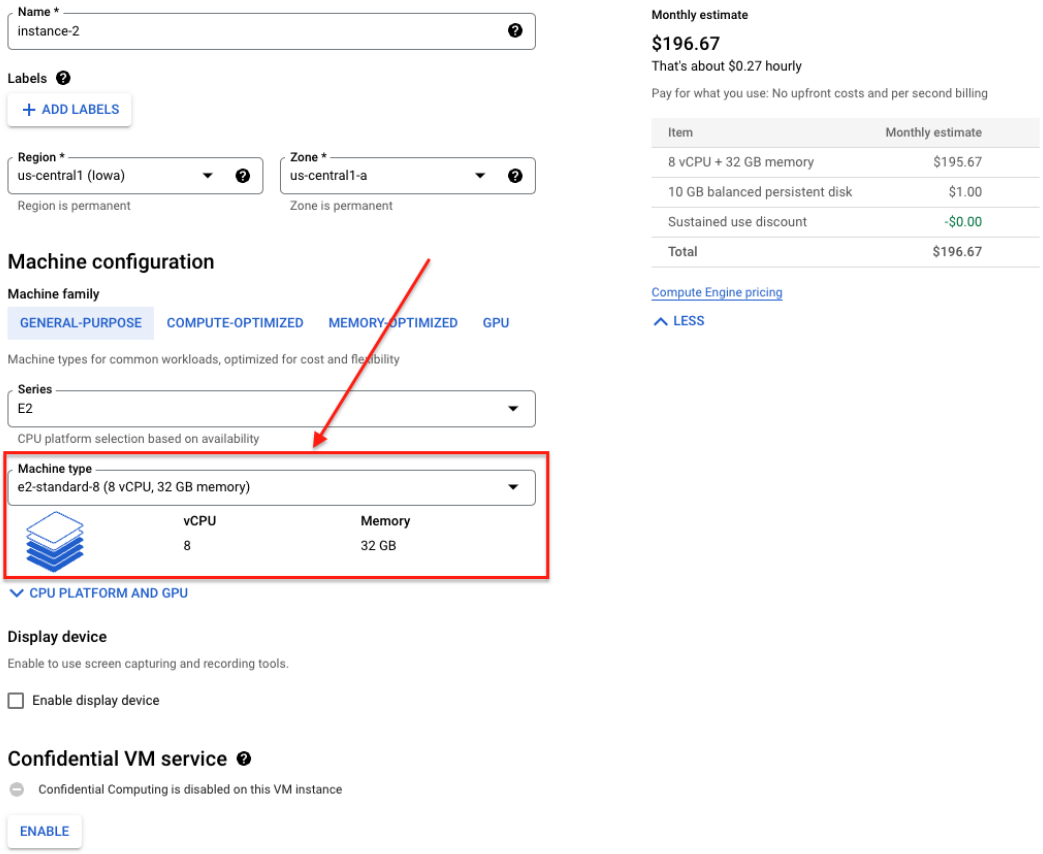


2. Click on *Create a VM* option.

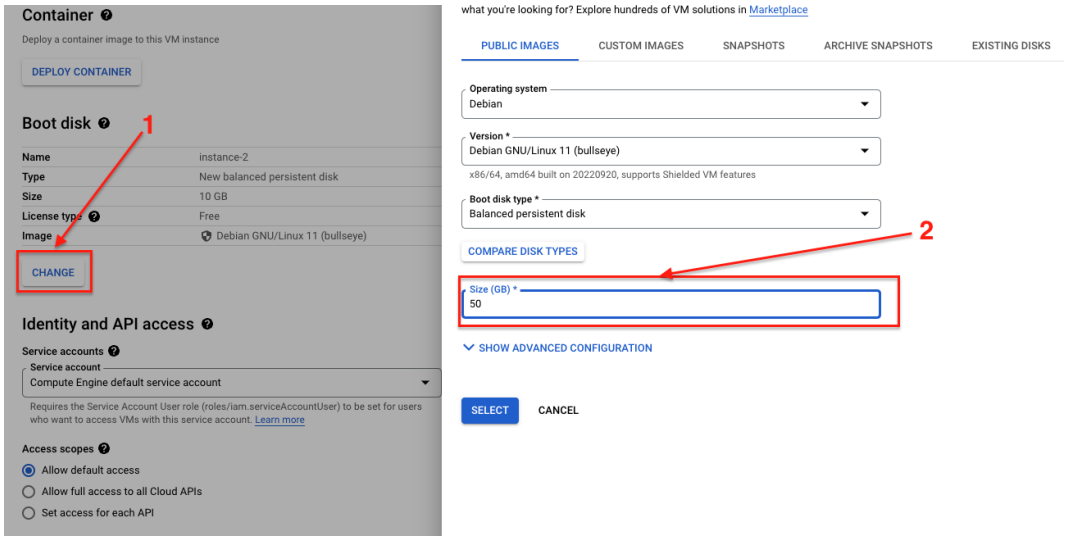


Make sure that you have the project with the billing account for the course selected.

3. Configure the VM with the required hardware



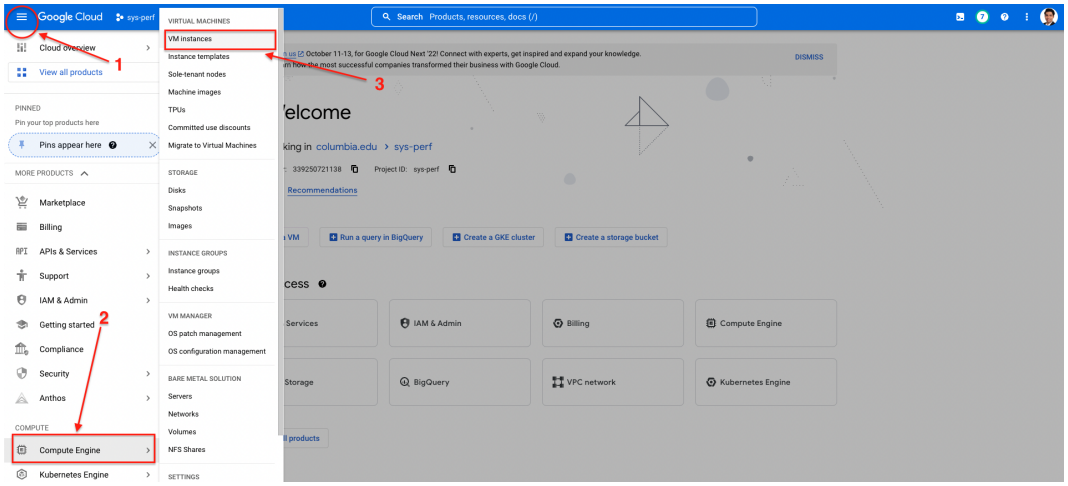
There are two things that need to be done for the first homework. This is first changing the machine configuration to a machine with higher RAM. In the above screenshot, you can see I have selected a machine with 32 GB RAM.



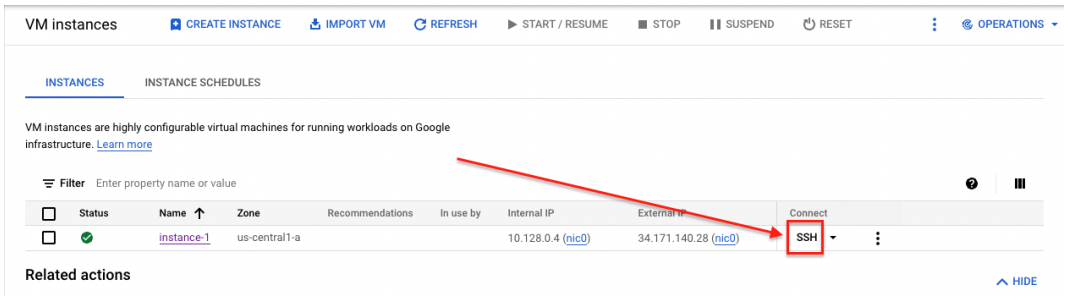
The second thing that needs to be done is to increase the storage space of the virtual machine from 10GB to at least 30GB. In the above screenshot, you can see I have increased the storage space to 50GB to be on the safer side.

Once you have configured your machine, scroll to the bottom of the page and click on the *Create* button to create an instance.

4. Go to the dashboard to check out the created VM instance



5. SSH into the created VM instance



2 Intel MKL Library Installation

For this installation, we will be using the Intel OneAPI Basekit. The instructions for installation are given in the following link: **Installation using APT manager**. There are other options for installation as well, but we suggest following the APT manager installation.
Make sure you have installed **wget**.

```
sudo apt install wget
```

1. Download the key to system key ring

Copy and paste the following command into the SSH terminal of the VM instance:

```
wget -O-
https://apt.repos.intel.com/intel-gpg-keys/GPG-PUB-KEY-INTEL-SW-PRODUCTS.PUB
| gpg --dearmor | sudo tee /usr/share/keyrings/oneapi-archive-keyring.gpg
> /dev/null
```

The above command is a single line. Make sure there are no new lines in the command.

```
sb4539@instance-1:~$ wget -O- https://apt.repos.intel.com/intel-gpg-keys/GPG-PUB-KEY-INTEL
-SW-PRODUCTS.PUB | gpg --dearmor | sudo tee /usr/share/keyrings/oneapi-archive-keyring.gpg
 > /dev/null
```

2. Add signed entry to apt sources and configure the APT client to use Intel repository

```
echo "deb [signed-by=/usr/share/keyrings/oneapi-archive-keyring.gpg]
https://apt.repos.intel.com/oneapi all main" |
sudo tee /etc/apt/sources.list.d/oneAPI.list
```

The above command is also a single line. Make sure there are no new lines in the command.

```
sb4539@instance-1:~$ echo "deb [signed-by=/usr/share/keyrings/oneapi-archive-keyring.gpg]
https://apt.repos.intel.com/oneapi all main" | sudo tee /etc/apt/sources.list.d/oneAPI.lis
t
```

3. Update packages list and repository index

```
sudo apt update
```

4. Install MKL basekit

```
sudo apt install intel-basekit
```

This is going to take a while to install.

5. Set environment variables

```
. /opt/intel/oneapi/setvars.sh
```

```
sb4539@instance-1:~$  . /opt/intel/oneapi/setvars.sh

:: initializing oneAPI environment ...
   -bash: BASH_VERSION = 5.1.4(1)-release
   args: Using "$@" for setvars.sh arguments:
:: advisor -- latest
:: ccl -- latest
:: compiler -- latest
:: dal -- latest
:: debugger -- latest
:: dev-utilities -- latest
:: dnnl -- latest
:: dpcpp-ct -- latest
:: dpl -- latest
:: intelpython -- latest
:: ipp -- latest
:: ippcp -- latest
:: mkl -- latest
:: mpi -- latest
:: tbb -- latest
:: vpl -- latest
:: vtune -- latest
:: oneAPI environment initialized ::

sb4539@instance-1:~$
```

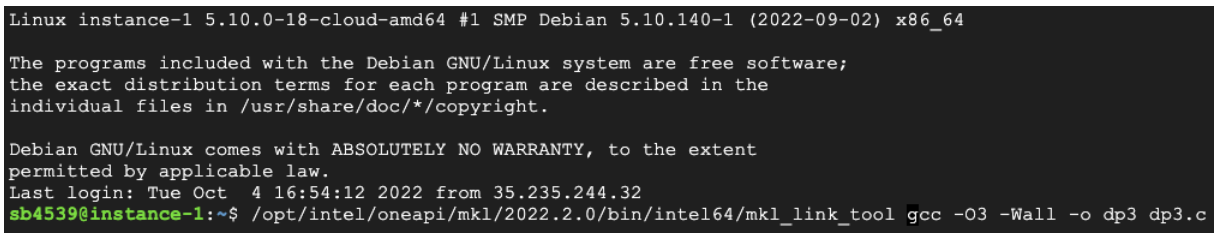
Now you are all set to run your code with the MKL library.

3 Running Code with MKL Linkage

This section assumes that you have a functioning code for $C3$. The commands given below are single line commands and need to be run with any new lines. Make sure to remove new lines before running the commands in the terminal.

Option 1: Using MKL_LINK_TOOL

```
/opt/intel/oneapi/mkl/2022.2.0/bin/intel64/mkl_link_tool
gcc -O3 -Wall -o dp3 dp3.c
```



Option 2: Using GCC Flags

```
gcc -O3 -Wall -o dp3
-I /opt/intel/oneapi/mkl/2022.2.0/include dp3.c
-L /opt/intel/oneapi/mkl/2022.2.0/lib -lmkl_rt
```