Instructions for Bellevue Big Data Class (Google Cloud)

Signing up and Getting Free Credits on Google Cloud

1. Go to the Google Cloud website.
2. Click `Get Started for Free`.
3. Sign in with your Google account or create one.
4. Follow the prompts to create your new Google Cloud account. You'll need to provide your credit card details for verification purposes, but you won't be charged unless you upgrade your account.
5. After setting up, you should have $300 in free credits.

Remember, the Google Cloud free tier credits expire after 90 days or when they are all used. Always monitor your usage to avoid unexpected charges. Be sure to stop your instance when not in use to conserve your credits.

Creating an SSH Key

Before you can add an SSH key to your Google Cloud instance, you need to generate one. This process differs slightly depending on your operating system.

On macOS

1. Open Terminal.

2. Enter the following command and replace "your_email@example.com" with your email address:

   ```
   ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
   ```

3. When asked to "Enter a file in which to save the key," press Enter to use the default location.

4. At the prompt, type a secure passphrase.

This command generates a new SSH key, using the provided email as a label. Your public key will be saved in the file ~/.ssh/id_rsa.pub and your private key will be saved in the file ~/.ssh/id_rsa.

On Windows

1. Visit the official PuTTY download page: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
2. Under "Package files", download `putty-xx.xx-installer.msi` (where `xx.xx` is the latest version number).
3. Run the downloaded installer and follow the on-screen instructions to install PuTTY.
4. Search for and open `PuTTYgen` from the Start menu.

5. Click on the `Generate` button.
6. Move your mouse randomly over the blank area to generate some randomness until the progress bar fills up.
7. Once the key has been generated, you'll see the key displayed in the text field.
8. Save the private key by clicking `Save private key`. Save it somewhere safe and remember the location; you'll need it for authentication.

   Note do NOT enter a password for the Key Phrase. Keep it blank.

9. Examine the public key under the section "Public key for pasting into OpenSSH authorized_keys file" in PuttyGen. Copy this into a notepad. Your public key will look like this:

   ssh-rsa ABC……XYZ **rsa-key-20230926**

   Note do NOT directly click `Save publickey` as Google Cloud will not accept this default format.

10. At the end of the key (after the long series of characters), you'll see a comment which is often the username and/or hostname of the machine where the key was generated. In my case it is **rsa-key-20230926.** In your case this may also be u**sername@hostname**. Replace this with your name. In my case the public key would now look like:

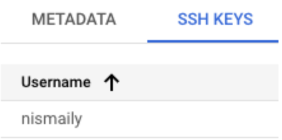
    ssh-rsa ABC……XYZ **nasheb**

    Note the user you enter will be the username used to SSH into your virtual machine in Google Cloud.

Adding an SSH Key to Your Google Cloud Instance

1. Go to the metadata page in the Google Cloud Console by clicking on `Compute Engine` -> `Metadata`.
2. Click on the `SSH Keys` tab.
3. Click on `Edit`, then `Add item`.
4. Open your public key file with a text editor, copy the content, and paste it into the box.
5. Click `Save`.

Note the `Username` that appears after the upload. You will use this username for SSH and for port-forwarding later.

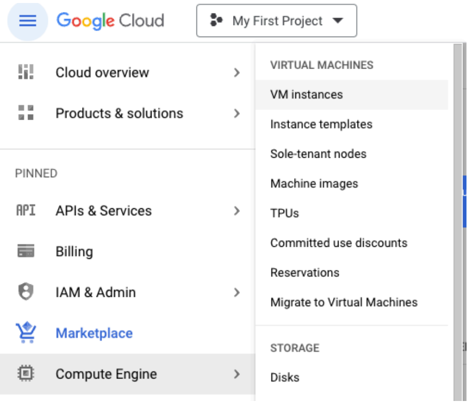


Username ↑
nismaily

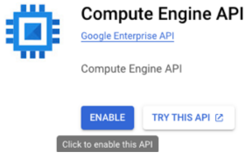
Now, you can use SSH to connect to your instance using the associated private key. Make sure to keep your private key safe and do not share it.

Spinning up an Ubuntu 22.04.2 Instance

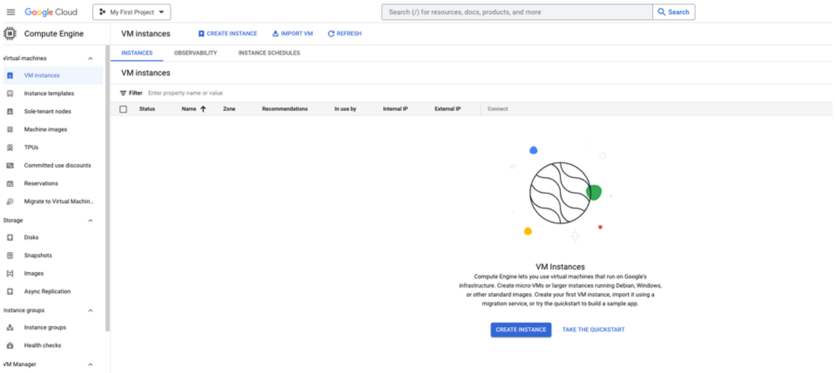
1. In the Google Cloud Console, go to the VM Instances page. Click on `Compute Engine -> VM Instances`.



2. Click Enable

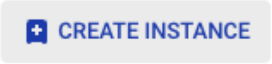


3. After the Engine has been enabled you will be taken here:

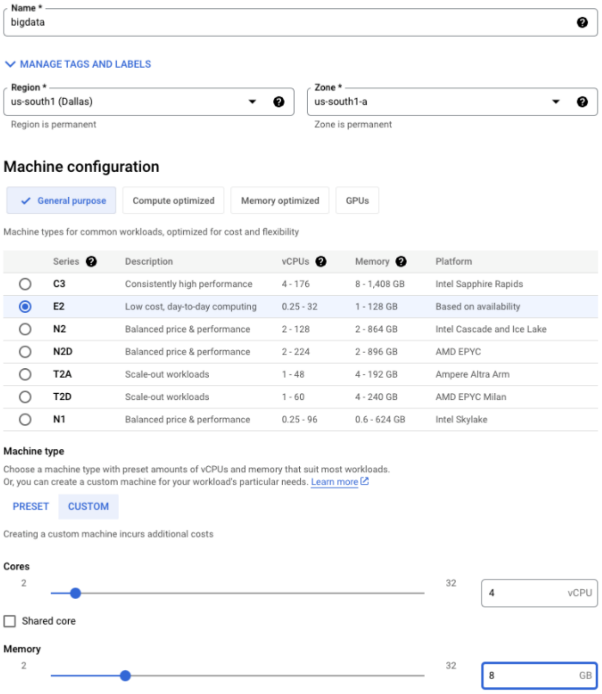


Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a migration service, or try the quickstart to build a sample app.

4. Click on `Create Instance`.



5. In the `Name` field, input a name for your instance.
6. Choose a region and a zone of your preference.
7. In the `Machine configuration` section, select `Custom` and set the number of CPUs to 4 and the memory to 8 GB.



Name * bigdata

Region * us-south1 (Dallas)

Region is permanent

Zone * us-south1-a

Zone is permanent

Machine configuration

Machine types for common workloads, optimized for cost and flexibility

	Series	Description	vCPUs	Memory	Platform
○	C3	Consistently high performance	4 - 176	8 - 1,408 GB	Intel Sapphire Rapids
●	E2	Low cost, day-to-day computing	0.25 - 32	1 - 128 GB	Based on availability
○	N2	Balanced price & performance	2 - 128	2 - 864 GB	Intel Cascade and Ice Lake
○	N2D	Balanced price & performance	2 - 224	2 - 896 GB	AMD EPYC
○	T2A	Scale-out workloads	1 - 48	4 - 192 GB	Ampere Altra Arm
○	T2D	Scale-out workloads	1 - 60	4 - 240 GB	AMD EPYC Milan
○	N1	Balanced price & performance	0.25 - 96	0.6 - 624 GB	Intel Skylake

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. Learn more

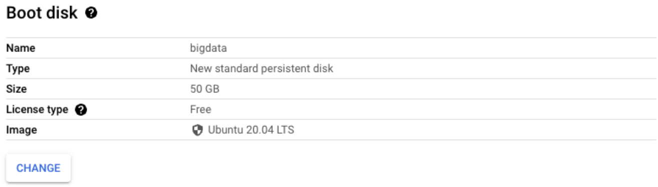
Creating a custom machine incurs additional costs

Cores 2 32 4 vCPU

□ Shared core

Memory 2 32 8 GB

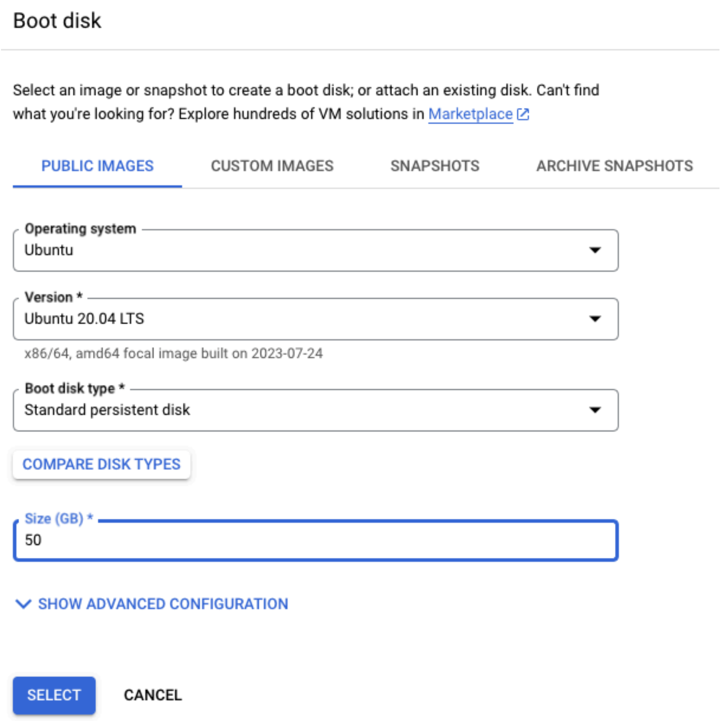
8. In the `Boot disk` section, click on `Change`.



Boot disk

Name	bigdata
Type	New standard persistent disk
Size	50 GB
License type	Free
Image	Ubuntu 20.04 LTS

9. Select `Ubuntu` from the OS images and choose `Ubuntu 22.04 LTS` from the list.
10. Change the boot disk to Standard persistent disk.
11. Change the disk size to 50 GB.



Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in Marketplace

Operating system: Ubuntu

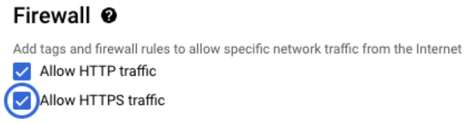
Version *: Ubuntu 20.04 LTS

x86/64, amd64 focal image built on 2023-07-24

Boot disk type *: Standard persistent disk

Size (GB) *: 50

12. Click Select.
13. Make sure to allow HTTP and HTTPS traffic by checking the boxes under the `Firewall` section.



Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

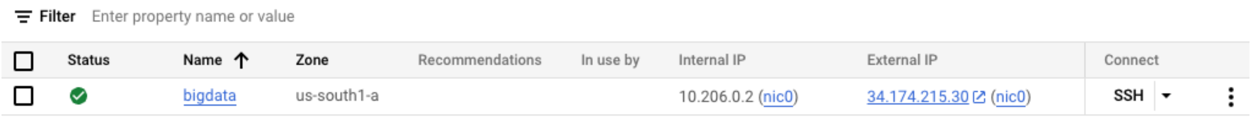
☑ Allow HTTP traffic

☑ Allow HTTPS traffic

14. Click `Create` to create the instance.

Setting Up Your Ubuntu Full Desktop Image

1. After your instance is set up, click the `SSH` button in the instances list.



□	Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	bigdata	us-south1-a			10.206.0.2 (nic0)	34.174.215.30 (nic0)	SSH

SSH into your VM

On macOS

1. Open Terminal.

2. Enter the following command. Replace USER with the username that appears when uploading your key to Google Cloud and replace External IP with the External IP from your Google Cloud VM.

```
ssh USER@EXTERNALIP
```

On Windows

1. Open PuTTY (the main program).
2. Under `Host Name (or IP address)`, enter the IP address of the Google Cloud Virtual Machine.
3. On the left pane, go to Connection -> SSH -> Auth.
4. Click on the `Browse` button and select the private key you saved in the previous step.
5. Return to the main `Session` section, enter a name for this session under `Saved Sessions` and click `Save`. This way, you won't need to repeat the above steps every time.
6. Click `Open` to initiate the SSH connection.
7. The first time you connect, you'll see a security alert. Click `Yes` to trust and add the host's key to your cache.
8. Log in with the username you've set up on the VM. Since you're using key authentication, you should not be prompted for a password (unless you set a passphrase for your SSH key).

Downloading and Running the Setup Script

1. Download the git repository for the class.

   ```
   git clone https://github.com/bellevue-university/dsc650-infra.git
   ```

2. `Change into the dsc650-infra directory.`

   ```
   cd dsc650-infra
   ```

3. Change the script's permissions to make it executable:

   ```
   chmod +x setup.sh
   ```

4. Run the script:

   ```
   sudo ./setup.sh
   ```

   This will install Docker and Docker Compose, and clone the Bellevue Big Data repository.

Running the Big Data Software

1. Type `cd bellevue-bigdata` and hit `Enter`
2. You should now see several directories: `hadoop-hive-spark-hbase`, `kafka`, `nifi`, and `solr`. Each contains a `docker-compose.yml` file except for `nifi` which contains the software binaries.

Follow these steps for the **`hadoop-hive-spark-hbase`** `and` **`solr`** `directores:`

1. Change into the directory with `cd <directory-name>`, replacing `<directory-name>` with the name of the directory.

2. Type *`docker-compose up -d`* and hit `Enter`. This will start up the software in that directory.
3. Verify that everything is healthy using *`docker ps`*.
4. Navigate to the user interface for each software component with the instructions provided in the next section, Accessing User Interfaces
5. Once you've verified that the user interfaces are working correctly, you can shut down the Docker containers for that directory with *`docker-compose down`*
6. Return to the parent directory with *`cd ..`* and move on to the next directory

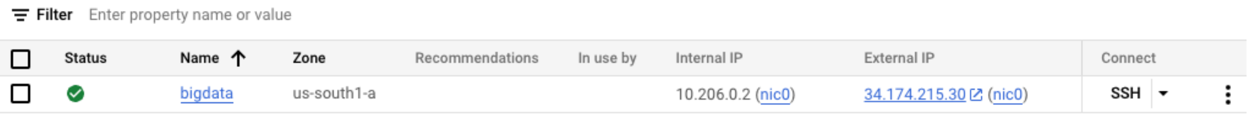
Follow these steps for the **`nifi`** `directory:`

1. Change into the nifi directory with `cd nifi`
2. Start NiFi using the command:
   `/bin/bash nifi-*/bin/nifi.sh start`
3. Navigate to the user interface for each software component with the instructions provided in the next section, Accessing User Interfaces
4. Stop NiFi using the command:
   `/bin/bash nifi-*/bin/nifi.sh stop`

Accessing User Interfaces

To access the user interfaces, you'll need to configure port forwarding for each component on your local machine. Use the SSH command, replacing `username` with the username from your SSH key setup and and `external_IP` with the external IP address of your Google Cloud VM instance. You can obtain the `external_IP from the compute engine page in Google Cloud.`

If you're using a Mac, then you can simply use a terminal session for port forwarding. If you're using putty. If you are using a PC, see the section `Port Forwarding with Putty` below.



□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	bigdata	us-south1-a			10.206.0.2 (nic0)	34.174.215.30 (nic0)	SSH

- HDFS:
  - Run the command: `ssh -L 9870:localhost:9870 username@external_IP`
  - Then, open your web browser and go to: http://localhost:9870
- YARN:
  - Run the command: `ssh -L 8088:localhost:8088 username@external_IP`
  - Then, open your web browser and go to: http://localhost:8088
- Spark Master:
  - Run the command: `ssh -L 8080:localhost:8080 username@external_IP`
  - Then, open your web browser and go to: http://localhost:8080

- Spark History:
  - Run the command: `ssh -L 18080:localhost:18080 username@external_IP`
  - Then, open your web browser and go to: http://localhost:18080
- HBASE:
  - Run the command: `ssh -L 16010:localhost:16010 username@external_IP`
  - Then, open your web browser and go to: http://localhost:16010
- Solr:
  - Run the command: `ssh -L 8983:localhost:8983 username@external_IP`
  - Then, open your web browser and go to: http://localhost:8983
- NIFI:
  - Run the command: `ssh -L 8443:localhost:8443 username@external_IP`
  - Then, open your web browser and go to: https://localhost:8443/nifi

    1. Your browser may show a warning about the website's security certificate. This is expected because we are using a self-signed certificate for the NiFi instance. To proceed, click on "Advanced" and then "Accept the Risk and Continue" (the wording may vary depending on your browser).

    2. To log in, you will need a username and password. These are generated when the NiFi instance is started and can be found in the instance's logs.

    3. On your VM terminal, go into the nifi directory and run:

       ```
       grep Generated nifi-*/logs/*
       ```

    4. Look for the username and password in the output. They will be inside square brackets. For example:

       ```
       Generated Username […]
       Generated Password […]
       ```

    5. Use these credentials to log in to the NiFi user interface.

Remember, these URLs will only be accessible when the respective command for port forwarding is running in your terminal, and the appropriate services are running on your Google Cloud instance.

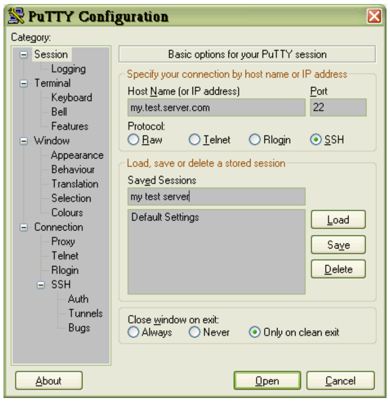
Port Forwarding with Putty

This option is for PC users only. To port forward with Putty do the following:

1. Open PuTTY.EXE, configure your host name, and select SSH for port.

2. Type the name you wish to use for the saved connection. In this example it is my.test.server. You will replace this with the `External IP` of your google cloud virtual machine. Do not save this yet; we must configure the ports for tunneling.

3. Click on the path to reach Tunnels (Connection > SSH >Tunnels):



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): my.test.server.com

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

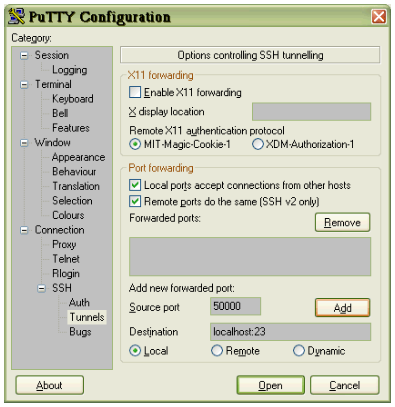
Load, save or delete a stored session

Saved Sessions: my test server

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

4. In the Port forwarding section, the Source Port is the source TCP/IP address you want assigned to your local host connection. The Destination is the connection on your remote SSH machine. localhost:23 in this example will get you a Telnet connection. For your tunnels, you will need to use the ports in the `Accessing User Interfaces` section above. Select both `Local ports accept connections from other hosts` and `Remote ports do the same`.



Options controlling SSH tunnelling

X11 forwarding

☐ Enable X11 forwarding

X display location

Remote X11 authentication protocol: ● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

Port forwarding

☑ Local ports accept connections from other hosts

☑ Remote ports do the same (SSH v2 only)

Forwarded ports:

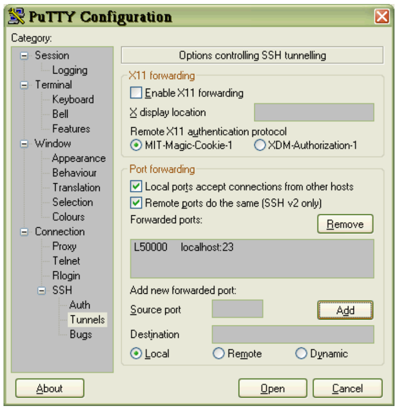
Add new forwarded port:

Source port: 50000

Destination: localhost:23

● Local ○ Remote ○ Dynamic

5. Click the Add button to place your tunnel configuration in the Forwarded ports window.



Options controlling SSH tunnelling

X11 forwarding

☐ Enable X11 forwarding

X display location

Remote X11 authentication protocol: ● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

Port forwarding

☑ Local ports accept connections from other hosts

☑ Remote ports do the same (SSH v2 only)

Forwarded ports:

L50000 localhost:23

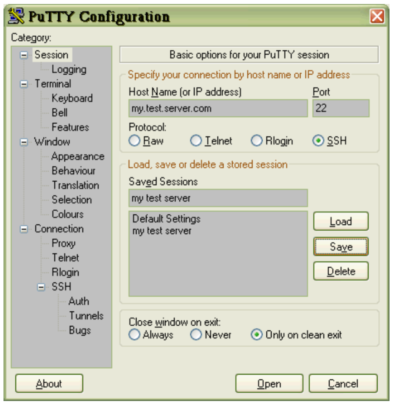
Add new forwarded port:

Source port

Destination

● Local ○ Remote ○ Dynamic

6. In the left pane, click on Session to bring up the following window. Click on the Save button:



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): my.test.server.com

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Load, save or delete a stored session

Saved Sessions: my test server

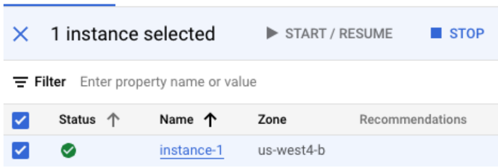
Default Settings
my test server

Close window on exit: ○ Always ○ Never ● Only on clean exit

7. On the left pane, go to Connection -> SSH -> Auth.
8. Click on the `Browse` button and select the private key you saved in the previous step.
9. Return to the main `Session` section, enter a name for this tunnel session under `Saved Sessions` and click `Save`. This way, you won't need to repeat the above steps every time.
10. Now you can launch your session and sign in to the secure shell. After you are signed in, you must leave this window open to keep your tunnel active.

Shutting Down

- Ensure all Docker containers are turned off with `docker-compose down` for each directory.
- Ensure NiFi is stopped using `/bin/bash nifi-*/bin/nifi.sh stop`
- You can then stop your Google Cloud instance.



☑	Status ↑	Name ↑	Zone	Recommendations
☑	✓	instance-1	us-west4-b	

Remember, the Google Cloud free tier credits expire after 90 days or when they are all used. Always monitor your usage to avoid unexpected charges. Be sure to stop your instance when not in use to conserve your credits.