# Instructions for Bellevue Big Data Class (Google Cloud)

This option is designed for those who are unable to use their local machines due to hardware limitations or incompatible architectures (e.g., M1 and M2 Mac processors).

## Signing up and Getting Free Credits on Google Cloud

1. Go to the [Google Cloud](https://cloud.google.com) 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"

1. When asked to “Enter a file in which to save the key,” press Enter to use the default location.
2. 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.

## 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 port-forwarding later.

A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated

1. Click Enable

A screenshot of a computer application

Description automatically generated

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

A screenshot of a web page

Description automatically generated

1. Click on Create Instance.



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

A screenshot of a computer

Description automatically generated

1. In the Boot disk section, click on Change.

A screenshot of a computer

Description automatically generated

1. Select Ubuntu from the OS images and choose Ubuntu 22.04 LTS from the list.
2. Change the boot disk to Standard persistent disk.
3. Change the disk size to 50 GB.

A screenshot of a computer

Description automatically generated

1. Click Select.
2. Make sure to allow HTTP and HTTPS traffic by checking the boxes under the Firewall section.

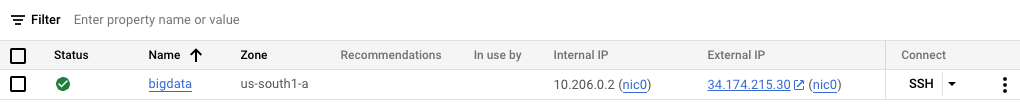
A white background with black text

Description automatically generated

1. 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.



## 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

1. Change into the dsc650-infra directory.

cd dsc650-infra

1. Change the script’s permissions to make it executable:

* chmod +x setup.sh

1. 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:

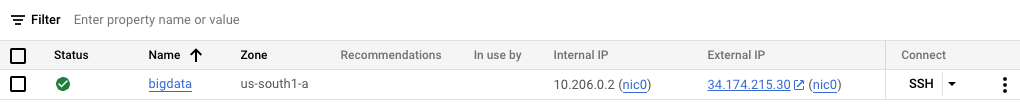
/bin/bash nifi-\*/bin/nifi.sh start

1. Navigate to the user interface for each software component with the instructions provided in the next section, Accessing User Interfaces
2. 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.



* 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/\*

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

**Generated** Username […]

**Generated** Password […]

1. 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.

## 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.

A screenshot of a computer

Description automatically generated

**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.**