

Get a Cloud Virtual Machine (VM)

Most cloud providers (e.g., AWS, Azure, and GCP) offer VMs and education credits, and you can choose any of the providers. Here we select GCP (Google Cloud Platform) as the cloud provider. Instructions for getting free GCP credit can be found in the appendices.

- Register an account on GCP: <https://cloud.google.com/>
- Log into the console: <https://console.cloud.google.com>
- Create a VM with the following configuration
 - Instance type: `e2-standard-8` (8 vCPUs, 32 GB memory)
 - VM provisioning model: `Spot`
 - On VM termination: `Stop`
 - Note: Spot VMs are cheaper than regular VMs but can be reclaimed by the provider at any time on a short notice (the reclamation rate is actually quite low according to our experience). You can restart the VM if it's reclaimed. You can also select "Standard" to avoid the trouble at a higher price.
 - Boot disk -> Change -> Image: `Ubuntu 22.04 LTS (x86_64)`
 - Disks (under Advanced options): Add new disk
 - Disk type: `Balanced persistent disk`
 - Size: `100 GB`
 - Deletion rule: `Delete disk`

Once the VM is created, you can SSH into it through the GCP console. Clouds adopt the pay-as-you-go billing model, which means you pay for what you use (often at the minute granularity). The good practice is to start a VM when you use it and stop it when you don't.

Set up the VM

After SSH into the VM, you need to configure the disk to utilize it.

- Verify there is a 100 GB disk: `lsblk`, e.g., `sdb`
- Format it with the `ext4` file system: `sudo mkfs.ext4 /dev/sdb`
- Create a directory and mount it to the disk: `sudo mkdir /data; sudo mount /dev/sdb /data`
- Change the permissions of the directory: `sudo chmod 777 /data`

Install essential build tools for compiling PostgreSQL and your own applications.

- Update APT: `sudo apt update`
- Install build-essential: `sudo apt install build-essential`

Install PostgreSQL (version 16)

We show how to install PostgreSQL from the source code, mostly following the official instructions:

<https://www.postgresql.org/docs/16/installation.html>.

We use /data as the default directory.

- Install dependencies: `sudo apt install libicu-dev pkg-config libreadline-dev libz-dev flex bison`
- Create a directory where PostgreSQL will be installed: `mkdir /data/pg16`
- Get the source code from GitHub: `git clone https://github.com/postgres/postgres.git`
- Check out PostgreSQL 16: `cd postgres; git checkout tags/REL_16_3 -b v16`
- Configure PostgreSQL: `./configure --prefix=/data/pg16`
- Compile and install PostgreSQL: `make -j; make install`

On successful installation, there will be four directories under /data/pg16: bin/, include/, lib/, and share/.

Set up PostgreSQL

- Create a user group and a user called postgres: `sudo groupadd postgres; sudo useradd -g postgres postgres`
- Create a data directory for storing future database files and change the ownership to postgres: `mkdir /data/pg16/data; sudo chown postgres:postgres /data/pg16/data`
- Configure the postgres user
 - Create a home directory: `sudo mkdir /home/postgres; sudo chown postgres:postgres /home/postgres`
 - Set a password: `sudo passwd postgres`
 - Switch to the postgres user: `sudo su postgres`
 - Enter bash: `bash`
 - Create a .bash_profile file under the home directory (/home/postgres/) using, for example, Vim, with the following content

```
export PGHOME=/data/pg16
export PGDATA=/data/pg16/data
PATH=$PATH:$PGHOME/bin
```
 - Apply the profile: `source ~/.bash_profile`
- Initialize PostgreSQL: `initdb`
 - Check if files and directories are successfully generated under /data/pg16/data.
- Start PostgreSQL service: `pg_ctl start`
- Check service status: `pg_ctl status` (to stop the service, simply run `pg_ctl stop`)

Use PostgreSQL

The PostgreSQL database server is running in the background. You can now connect to the server and create databases (the default database account is `postgres` with password `postgres`). We demonstrate this using the PostgreSQL-provided SQL client.

- Enter the SQL client console: `psql`
- Create a database called `mydb`: `CREATE DATABASE mydb;`
- Connect to the database: `\c mydb;`
- Create a test table: `CREATE TABLE testtbl(name varchar(32), value int);`
- Insert a tuple into the table: `INSERT INTO testtbl(name, value) VALUES ('hello', 42);`
- Query the whole test table: `SELECT * FROM testtbl;`, which should show the following output:

```
name | value
-----+-----
hello |    42
(1 row)
```

- Quit from the SQL client console: `quit`

To develop applications with PostgreSQL, you can pick your favorite programming language. In the next two sections, we provide basic sample code in C++ and Java.

Program PostgreSQL in C++

Under the project folder, you can find the sample C++ code that embeds PostgreSQL under `Templates/CPP/pgsample.cpp`. It shows how to connect to the database, execute SQL queries, and display the result. Follow the steps below to compile and run the program:

- Upload the `CPP/` folder to `/data/` in the VM, e.g., using `scp` (you may need to set up SSH keys in order to access the VM from your local machine:
https://cloud.google.com/compute/docs/connect/add-ssh-keys#console_1).
- Install dependencies: `cd /data/CPP/; ./InstallDeps.sh`
- Compile the program with Make: `make`
- Run the program: `./pgsample`

Program PostgreSQL in Java

Under the project folder, you can find the sample Java code that embeds PostgreSQL under `Templates/Java/Main.java`. It shows how to connect to the database, execute SQL queries, and display the result. Follow the steps below to compile and run the program:

- Upload the `Java/` folder to `/data/` in the VM, e.g., using `scp` (you may need to set up SSH keys in order to access the VM from your local machine:
https://cloud.google.com/compute/docs/connect/add-ssh-keys#console_1).
- Install dependencies: `cd /data/Java/; ./InstallDeps.sh`
- Set up the Java project with Maven: `./SetupProject.sh`
- Compile the program and run it: `./BuildAndRun.sh`

Appendices

Create a free GCP account

You can register and create a free GCP account here: <https://cloud.google.com/free?hl=en>. As a new user, you should get \$300 USD free credit that you may use to develop the project.

Redeem additional \$50 USD credit on GCP

Every student in this course can get an additional \$50 USD credit on GCP. Here is the URL you will need to access in order to request a Google Cloud coupon. You will be asked to provide your school email address and name. An email will be sent to you to confirm these details before a coupon is sent to you.

[Student Coupon Retrieval Link](#)

You will be asked for a name and email address, which needs to match your school domain (*cs.toronto.edu*, *utoronto.ca*, or *mail.utoronto.ca*). A confirmation email will be sent to you with a coupon code.

You can request a coupon from the URL and redeem it until: 10/20/2024

Coupon valid through: 6/20/2025

You can only request ONE code per unique email address.