

✔ You pushed to [csed516-2022au](#) just now



[Upload New File](#)

[JACK LEE KHUU](#) authored just now

f98dafd6

↩ **section7.md** 5.81 KiB

Installation instructions

We recommend installing Vertica via a precompiled Docker image. If you don't already have Docker installed, a good option is to launch an EC2 instance (make sure the instance has enough disk space if you are using AWS Academy, e.g., a good disk space is 100 GB. You can set it when you launch your instance.) and run it there. On the other hand, Docker is fantastic and having the ability to use it locally is great.

1. Install Docker

(If you already have Docker installed, you can skip to Step 2.)

Docker is available for most major platforms, including Windows, Mac and Linux. Installation instructions can be found [here](#). For those interested in learning more, the [Docker Docs](#) are a good resource, especially this [Overview](#). This is also a nice [command line reference](#).

If you're running Ubuntu (possibly on an EC2 instance), you can install Docker via Aptitude:

```
sudo apt update
sudo apt install docker.io
```

During the process, if you see errors like,

```
E: Could not get lock /var/lib/dpkg/lock-frontent - open (11: Resource temporarily unavailable)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is another process using it?
```

Refer to <https://askubuntu.com/questions/1109982/e-could-not-get-lock-var-lib-dpkg-lock-frontent-open-11-resource-temporari> and follow the steps in the answer.

When installing docker.io, if you see an error like,

```
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:
```

```
The following packages have unmet dependencies:
```

```
docker.io : Depends: containerd (>= 1.2.6-0ubuntu1~)
0 upgraded, 2 newly installed, 2 to remove and 220 not upgraded.
```

Install containerd before installing docker.io by,

```
sudo apt install containerd
```

Now you should be able to install docker.io by,

```
sudo apt install docker.io
```

2. Download and run the Vertica image

Vertica is available on [Docker Hub](#), which is a public registry of Docker repositories. There are several Vertica repositories available, but we will be using [this one](#). The full installation instructions can be found by following the link, but the following steps should suffice.

Depending on how you installed Docker, you may need to run the following commands using `sudo`

Retrieve the image:

```
sudo docker pull sumitchawla/vertica
```

Launch a container

The `-p` flag maps host port 5433 to container port 5433, which is the default Vertica port.

```
sudo docker run -d -p 5433:5433 sumitchawla/vertica
```

You can check the status of running containers by issuing:

```
sudo docker ps
```

You should see that your Vertica container is now running as an active container.

Connection parameters:

- Default DB Name: `docker`
- Default User: `dbadmin`
- Default Password: (None)

3. Install the clients

Vertica Python Library

There are several different clients available to communicate with a Vertica database that should now be running on your computer on port 5433.

Perhaps the easiest to use is Vertica's Python library. This client library is available on [Github](#) and can be installed through a simple `pip` installation. Note that some functionality of the client depends on the Python PostgreSQL library, `psycopg2`, which you may need to install as well (if you don't have it already):

```
sudo apt install python3-pip # If not already installed

sudo pip install vertica-python
```

Using the Vertica Client Library:

Basic usage of the Python client is demonstrated in [sample.py](#). Try running this script to make sure everything works. Compare the output of the script with the [sample_data.csv](#) file:

```
wget https://courses.cs.washington.edu/courses/csed516/19au/sections/sample.py
wget https://courses.cs.washington.edu/courses/csed516/19au/sections/sample_data.csv

python3 sample.py
```

Interactive shell: `vsql`

The `vsql` client is almost identical to the `psql` client for Postgres. [Go here](#) to download the `vsql` client appropriate to your system, or issue the following commands if you're running on Ubuntu:

```
wget https://www.vertica.com/client_drivers/9.3.x/9.3.0-0/vertica-client-9.3.0-0.x86_64.tar.gz
tar xzf vertica-client-9.3.0-0.x86_64.tar.gz
```

Otherwise, if you are using the Docker image from above, download client drivers version 9.0.x. On OSX the download package will contain a binary `vsql` --- copy it into your working directory (preferably this one) and try to run it with `./vsql`.

To connect to a running Vertica database (like the one you've installed above), be sure that the database is running and connect with

```
cd opt/vertica/bin # Or wherever vsql is located
./vsql -d docker dbadmin
```

You can then execute queries interactively. Try querying the table you ingested via the `sample.py` script:

```
select * from sample_table;
```

Timing Queries using `VSQL`

This can be done by either turning timing on using the flag `\timing` by simply execute the following,

```
\timing
```

or by querying the relation `v_monitor.query_requests` (using `\timing` is the easiest!),

```
select request_duration_ms, request from v_monitor.query_requests limit 10;
```

Exit VSQL

Simply `Ctrl+D`.