# Databand Installation on Docker

## Overview

This document provides step-by-step instructions for installing *Databand* in Docker*.* Official installation instructions are available in [documentation](https://docs.databand.ai/docs/docker-compose-deployment). Technical sellers and business partners can use this document to practice installation of Databand on docker. Please make sure to use official documentation for installation in a customer environment.

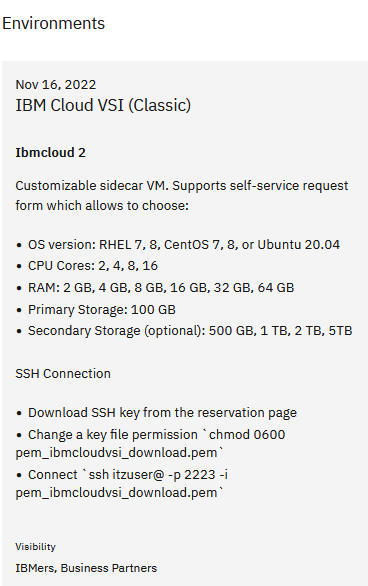
When working on a customer engagement, we should ask the customer to complete prerequisites specified in documentation, including installation of *Docker* and *Docker Compose*, as well as installation and configuration of the *Postgres* database.

In this lab we will complete all prerequisites prior to installing Databand.

## Step 1: Provision a Linux VM

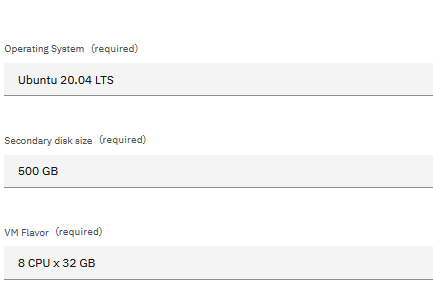
In this section we will provision a Linux VM (*Ubuntu*) using an **IBM TechZone** image. If you’re using your own infrastructure, you can select any supported Linux distro.

1. Log in to **TechZone** and navigate to the [Base Images collection](https://techzone.ibm.com/collection/base-images).
2. Select the following image:



1. Click **Reserve** and provide the following values for VM configuration.

*Note: You can reserve the VM in any geo.*



1. You will receive an email when the VM has been provisioned (may take up to an hour). After you have received the email, navigate to the TechZone reservation, open the **Details** page, and log in to the VM:

* When logging in, make sure to use the *login credentials/ssh* key provided in the reservation details
* Please see **Appendix A** if you need detailed instructions for connecting to the VM.

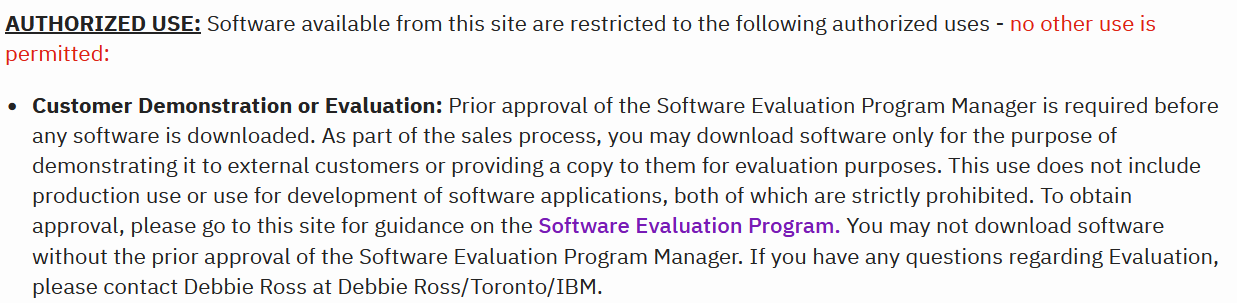
1. Test connectivity to the VM.

## Step 2: Download software for installation

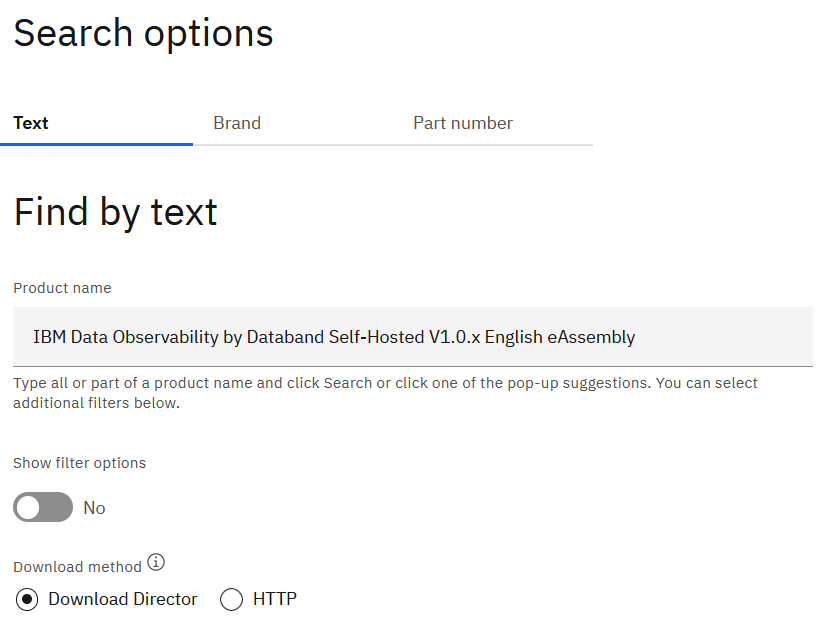
In this section we will download Databand installation files.

* Download [site](https://w3south-limited-use.cpc.ibm.com/software/xl/download/ticket.wss) for IBMers
* Business partners should use Passport Advantage.

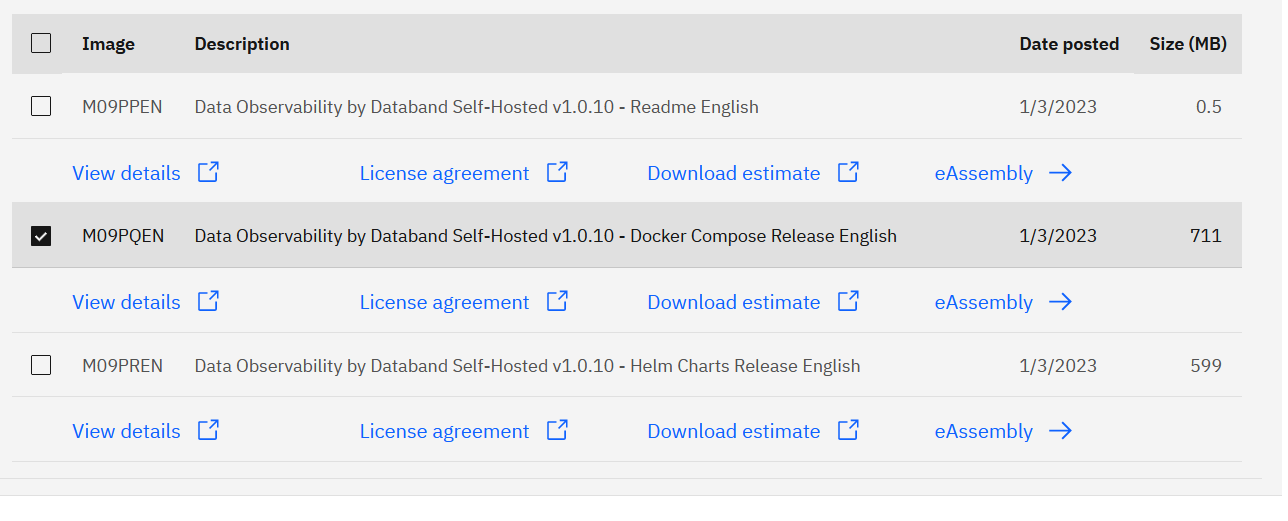
*Note: As stated on the internal downloads site, IBMers can use the downloaded files for PoCs after completing the required software evaluation* [*documentation*](https://ibm.seismic.com/app?ContentId=a37adba3-fec8-4365-94ab-e03cc0d22a00#/doccenter/5477419a-9474-4c51-94af-b442e9169fab/doc/%252Fdd98c5a3df-6b7c-1d77-6f07-d12e63954c78%252FdfOTRiYmU4NTQtNWY4NC03Y2QyLWZjYWUtOGIxYmFmZjkyZThk%252CPT0%253D%252CU2VsbGVyIGVuYWJsZW1lbnQ%253D%252Flf36e674a4-af29-44c4-8f05-000521f0b2ba/grid/?anchorId=5d4d533d-3b46-4773-b292-89e24053a2dc)*.*



* Download Databand installation files from IBM internal downloads or Passport Advantage:



For this section we need the *Docker compose* tar file, but if you wish, you can download all files in the Databand assembly.



*Note: If you did not install the Download Director, then select the HTTP option for download. The HTTP option supports downloading 1 file at a time, while the Download Director can download multiple files simultaneously*.

* Copy the installation files to the VM, if you need detailed instructions for copying files to VM please refer to **Appendix A**.

## Step 3: Install and configure prerequisite software

In this section we will install and configure prerequisite software as described in [documentation](https://docs.databand.ai/docs/docker-compose-deployment). When working on a customer engagement, we recommend that the prerequisites are completed by the customer. The customer should be familiar with Postgres and docker configuration.

*Postgres* is included with the installation files, but IBM recommends that it’s only used for testing and workshops. Customers should use external Postgres for production installations.

In this section we provide instructions that are specific to configuration in IBM environment.

We will complete the following steps:

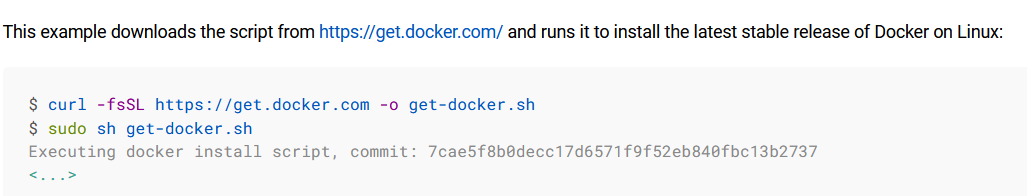
* Install docker
* Install docker compose
* Verify that rsync is installed
* Install and configure Postgres (optional).

First, we will install *docker* and *docker compose*.

*Note: Make sure to run all commands with* ***sudo*** *because Databand must be installed with root privileges.*

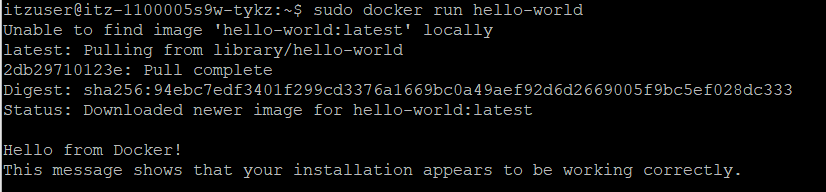
*We created a* [*file*](https://ibm.box.com/s/0kruuvdlcwdtnetywqxaly6vdqt3ef81) *(Installation\_commands) from which you can copy/paste Linux commands that are listed in the following section. You can also watch this* [*video*](https://ibm.box.com/s/vt5lnmi5xrdylhbbtk8qogjjel8g08vc)*.*

* For installing docker, we will follow [these instructions](https://docs.docker.com/engine/install/ubuntu/), specifically the **Convenience script** section



* After installation, test docker install:

sudo docker run hello-world



You have finished installing *docker*.

Next, we will install *docker compose*. In our experience, the easiest way to install *docker compose* is with instructions from [this blog](https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-20-04). If you would like to use official *docker compose* installation instructions, you can find them [here](https://docs.docker.com/compose/install/linux/#install-the-plugin-manually).

* Install and test docker compose installation. Run the following commands:

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

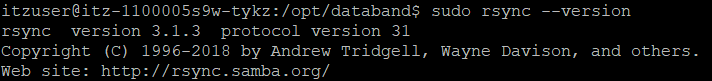
sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

You have finished installing docker compose.

* rsync is installed on the VM by default. If you wish, you can verify that it’s installed:

sudo rsync –version

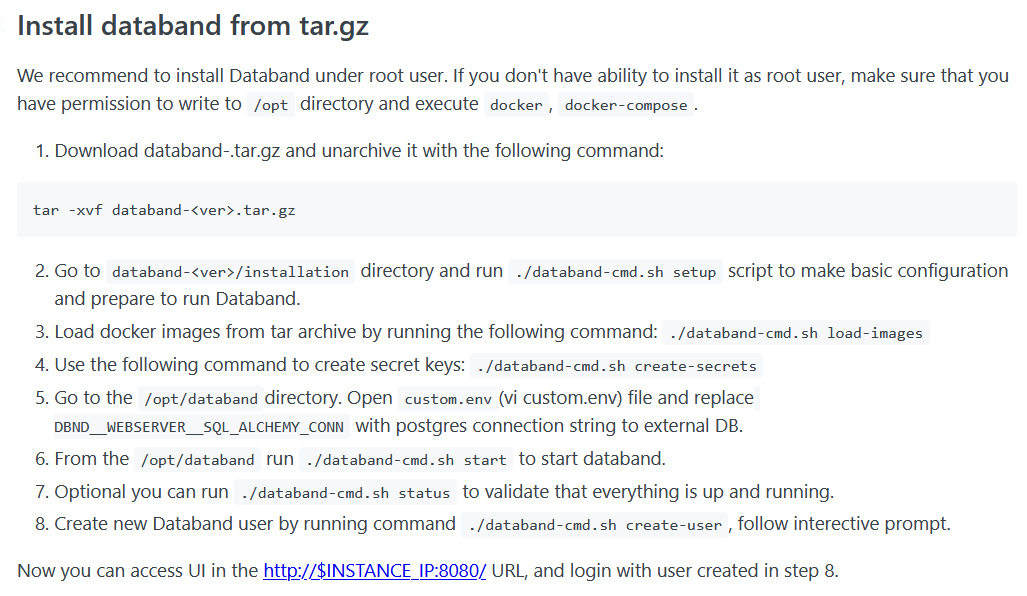


* In this lab, we will use internal Postgres. If you would like to install external Postgres, then follow instructions in **Appendix B**.

We completed all prerequisites, and now we’ll continue with instructions in documentation to install Databand.

## Step 4: Install Databand

In this section we will install databand as described in [documentation](https://docs.databand.ai/docs/docker-compose-deployment), specifically this section:

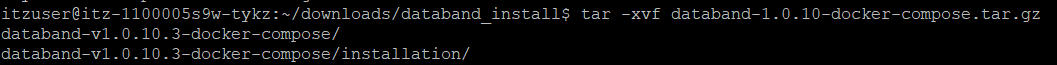


* Run all installation commands in documentation. Don’t forget to add *sudo* before each command.

**Notes:**

* If you’re using a Windows laptop and copy files with *FileZilla*, see **Appendix A** for connection instructions with the ssh key.
* After copying the databand tar file to your Linux VM, make sure to change permissions, for example:

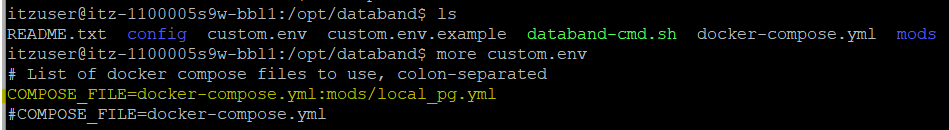
sudo chmod 777 databand-1.0.15-docker-compose.tar.gz



* Postgres:

If you **did not** install external Postgres, then

* + Do not complete **Step 5** as described in documentation. Instead, uncomment the top line (COMPOSE\_FILE value) in *custom.env* and edit it as shown in the screenshot. Then comment out the default COMPOSE\_FILE line below. This setting specifies which containers to use, and we are using the local Postgres in addition to Databand.



If you installed external Postgres:

* + When changing Postgres connection information in *custom.env*, use the following syntax:

postgres+psycopg2://<user>:<password>@<hostname>:<port>/<database-name>

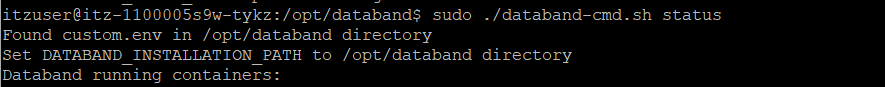
* + Example for VM config/lab instructions:

postgresql+psycopg2://databand:postgres4databand@<your\_VM\_ip>:5432/databand

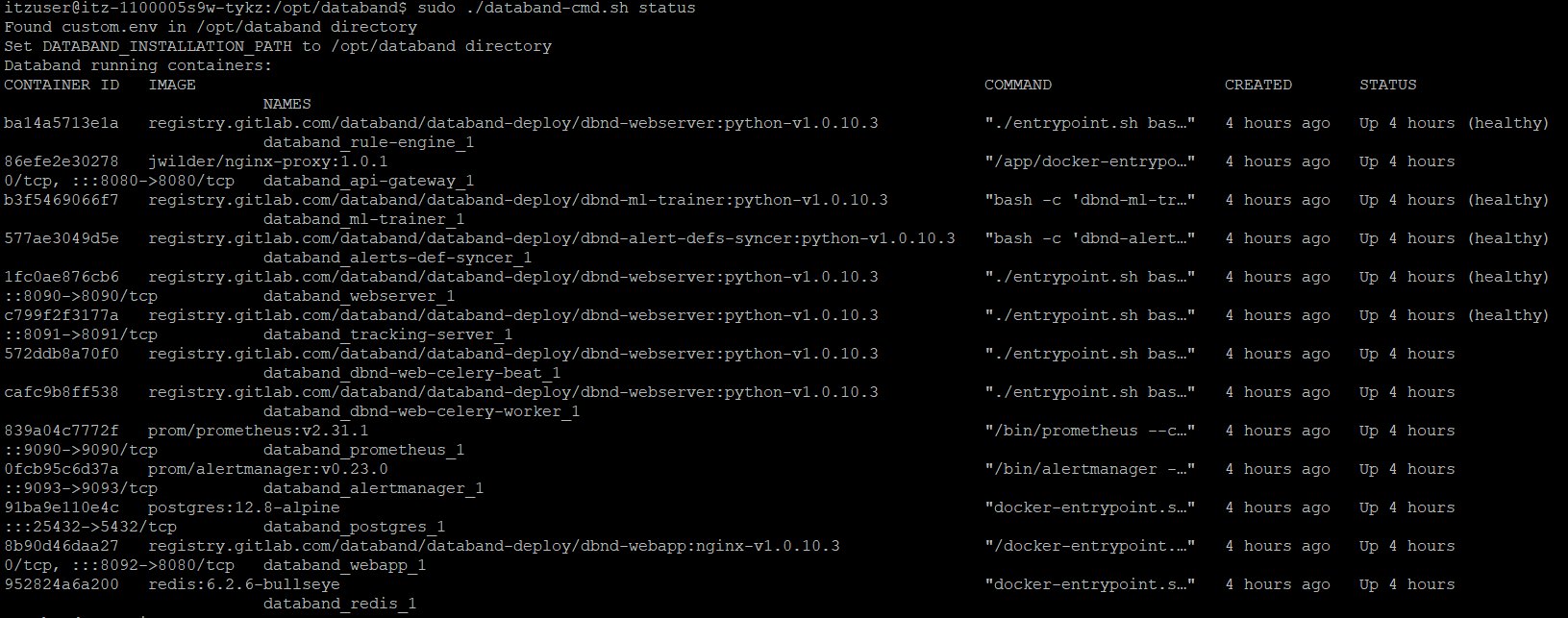


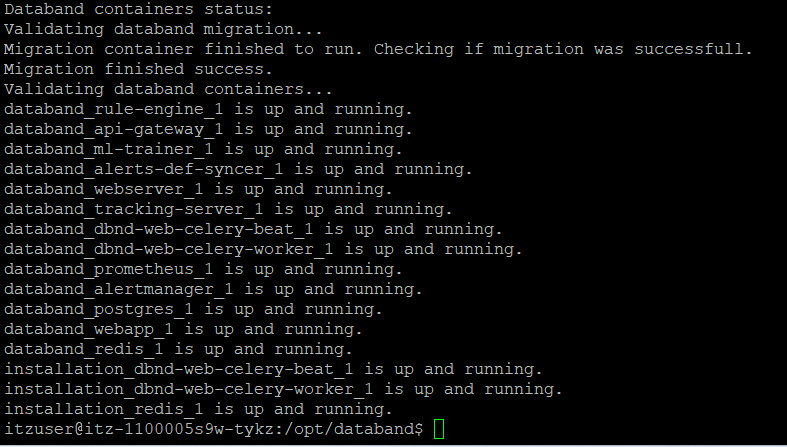
* Check if Databand is running after completing the last installation command

sudo ./databand-cmd.sh status



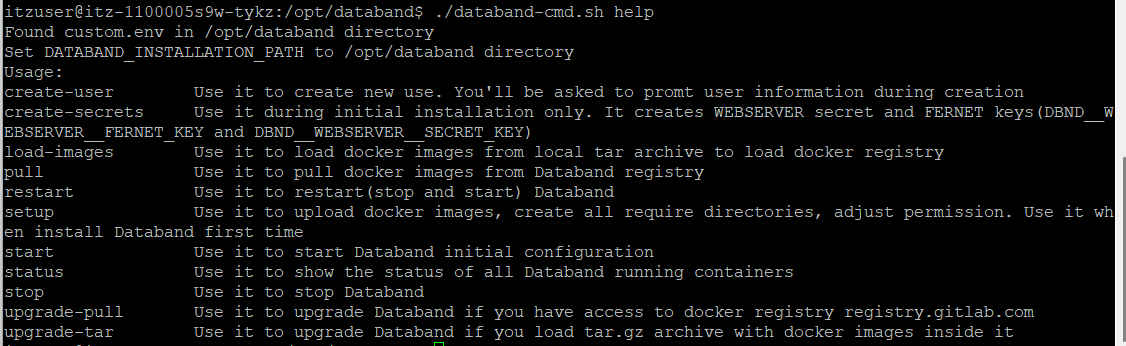
If Databand installed without issues, all pods will be healthy (no pods in *Starting* status):





* If you need to restart if Databand, use the *databand-cmd script* with *stop*/*start* or *restart*

sudo ./databand-cmd.sh **help**

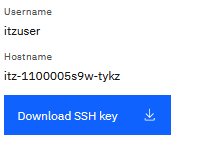


* When you log in to Databand after installation, all tabs with the exception of Integrations will be disabled until you start monitoring pipelines.
* If you wish, test data observability with one of the Python scripts that we’ve used in other labs.

**You have finished installing Databand.**

## Appendix A: Connect to a Linux VM in TechZone

When you review the details of your Ubuntu VM reservation in TechZone, you will notice that it provides the ssh key for authentication. Download the key.



**Connecting on a Mac:**

* Use the following syntax (replace with the values for your VM):

*# change key file permission, otherwise you will see “UNPROTECTED PRIVATE KEY FILE” error*

sudo chmod 600 pem\_ibmcloudvsi\_download.cer

*# public ip address can be found in the reservation details page.*

ssh -i pem\_ibmcloudvsi\_download.cer itzuser@<VM-public-ip-address> -p 2223

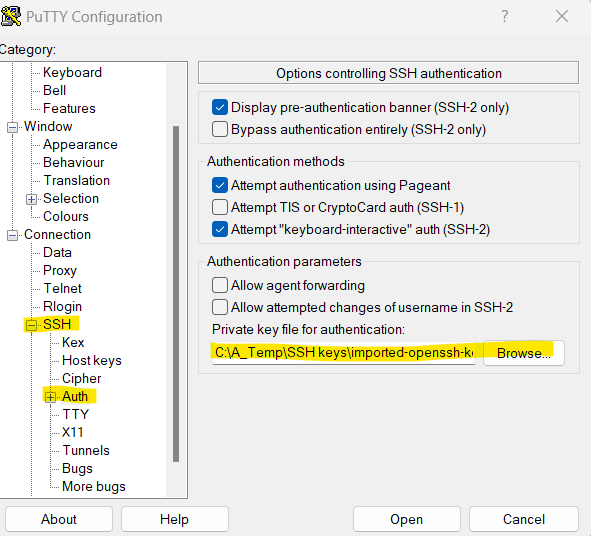
* For example, ssh -i pem\_ibmcloudvsi\_download.pem itzuser@158.176.198.108 -p 2223
* To copy installation files to the VM, use the following syntax:

scp -P 2223 -i pem\_ibmcloudvsi\_download.cer databand-1.0.15-docker-compose.tar.gz itzuser@@<VM-public-ip-address>:~/

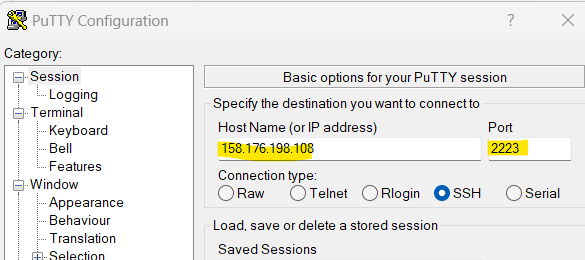
* If you want to use FileZilla on a Mac to transfer files, please refer to **Appendix C**.

**Connecting on a Windows Laptop (with PuTTY and FileZilla )**

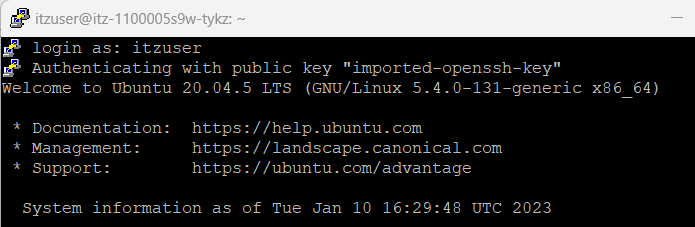
* Convert *.pem ssh* key file to *ppk* (ssh file format supported by PuTTY): <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>
* When configuring a connection in PuTTY, point to the generated ppk file in the **SSH - >Auth** section



* Specify *external IP address* and *port* for your VM.

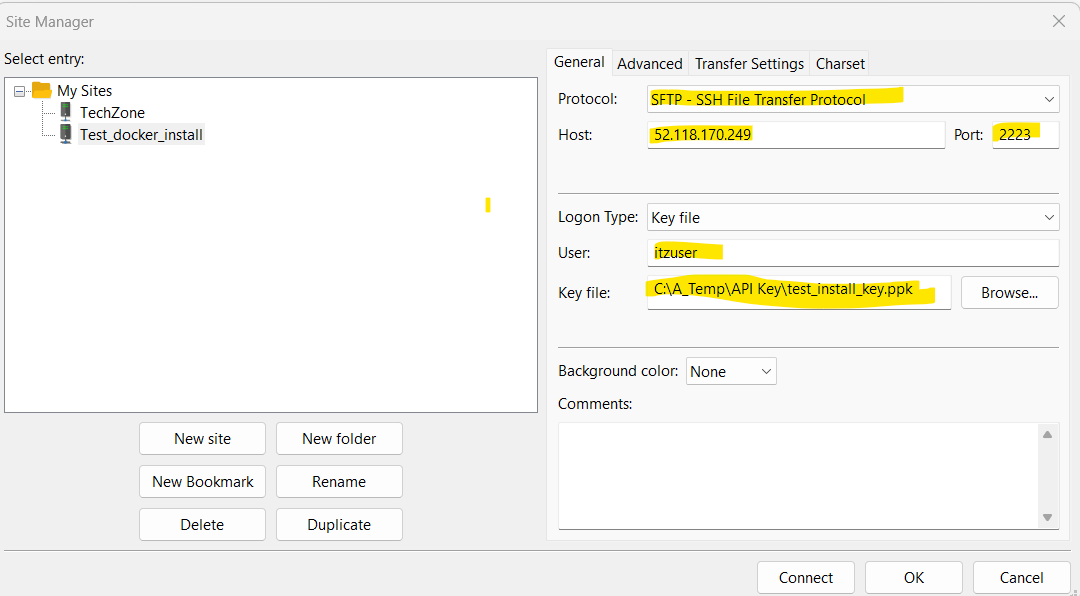


* When prompted to log in, make sure to specify the *itzuser* userid.



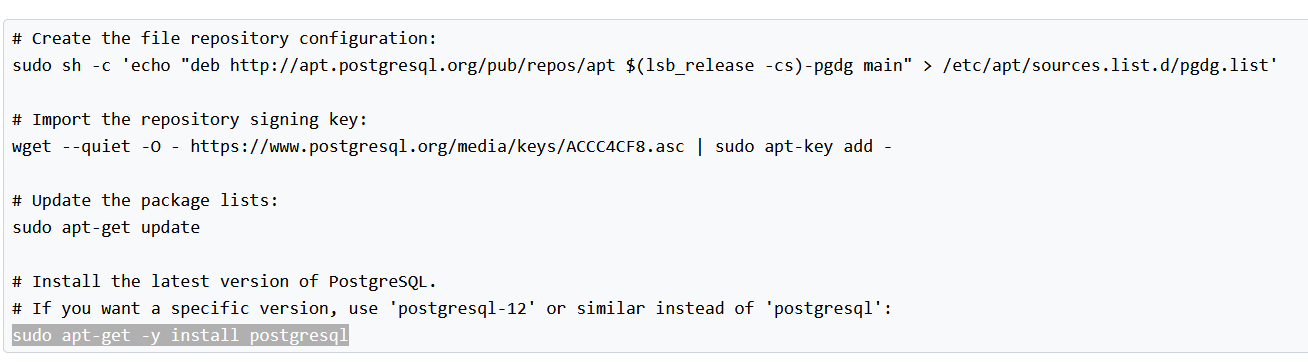
Later in the lab you will need to copy installation files to the VM. If you use *FileZilla*, you will need to configure it to use the ssh key. You can complete this section now or come back to it later when you have to copy the files)

* In Filezilla, create a *new site* (**File -> Site Manager**) using IP address of your cluster and the converted SSH key.



## Appendix B: Postgres installation and configuration

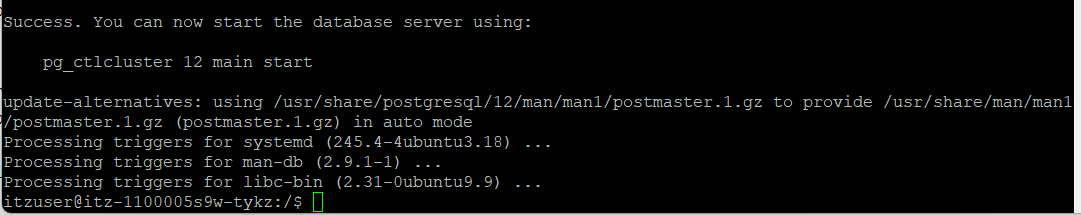
* For Postgres installation, we will follow instructions on [this page](https://www.postgresql.org/download/linux/ubuntu/).
* Run the following commands:



* + Note: Please install **postgresql-12**:

sudo apt-get -y install postgresql-12

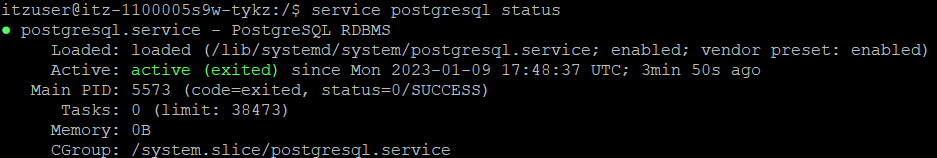
* Output of the last command



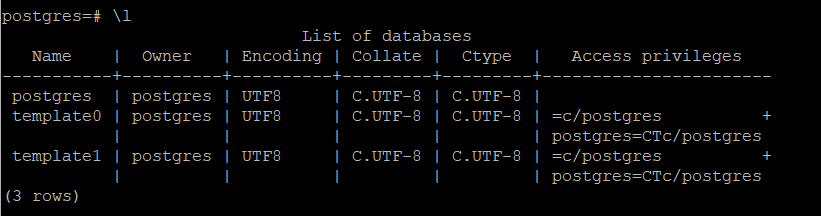
* Start Postgres: sudo pg\_ctlcluster 12 main start



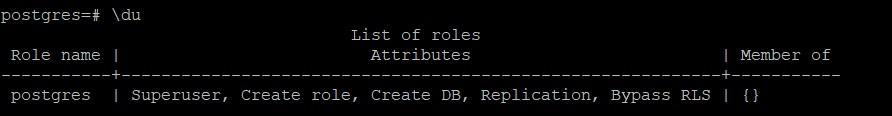
* Check that the database has started: service postgresql status



* Other useful commands
  + Start command line UI: sudo -u postgres psql
    - \l (list databases)



* + - \du (list users)



Next, we will create a database and a userid for Databand. These steps are captured in [documentation](https://docs.databand.ai/docs/external-db-for-databand-metadata), but you can follow instructions in this document.

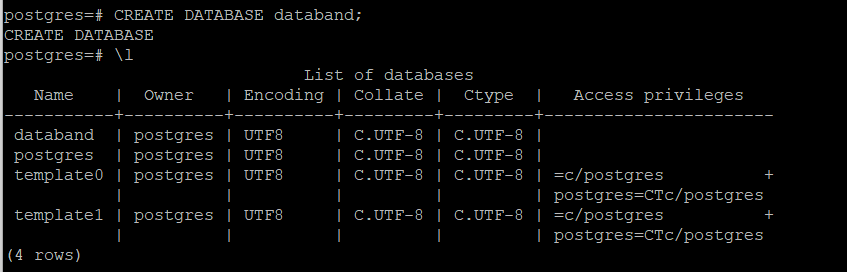
* From Postgres prompt run the following commands (run one command at a time)

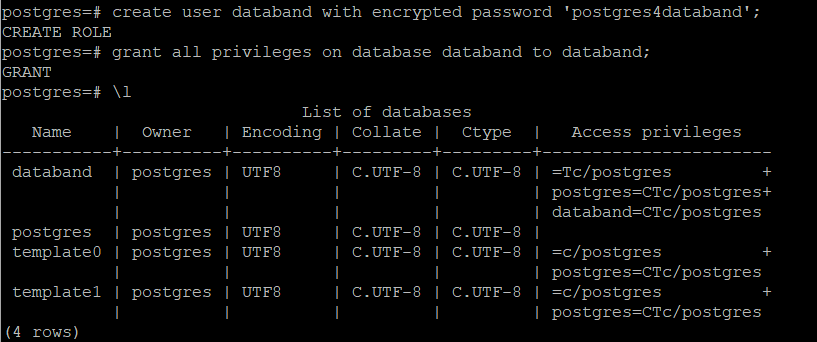
CREATE DATABASE databand;

create user databand with encrypted password 'postgres4databand';

grant all privileges on database databand to databand;

CREATE EXTENSION "uuid-ossp" SCHEMA pg\_catalog;





By default, Postgres is not enabled for access by external applications after the initial install. When working on a customer engagement, verify that they have *allowed remote access* to Postrges. Configuration instructions are not provided in Databand documentation because configuration is a part of standard Postgres setup.

We need to complete two steps to configure Postgres for external access:

* + Update the *postgresql.conf* file
  + Update the *pg\_hba.conf* file.

We are providing high-level instructions in this document, and if you need more information, you can review this [article](https://blog.logrocket.com/setting-up-a-remote-postgres-database-server-on-ubuntu-18-04/).

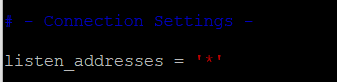
* Update the *postgres.conf* file
  + On the VM, change directory to */var/databand/runtime/postgres\_data/pgdata/*

sudo bash

**cd** /var/databand/runtime/postgres\_data/pgdata/



* Edit the file with vi or another editor
  + Change the *the listen\_addresses* value to *‘\*’*. Make sure it’s not commented out.



* Update the *pg\_hba.conf* file
  + Change directory to */etc/postgresql/12/main/*



* + Change the *IPV4* value to look like the following screenshot (replace *127.0.0.1/32* with *0.0.0.0/0*)



* Restart Postgres by running this command

sudo systemctl restart postgresql

You have finished installing and configuring *Postgres*.

## Appendix C: File Transfer using FileZilla on Mac:

1. You can download FileZilla for free [here](https://filezilla-project.org/download.php?type=client#close):

Table

Description automatically generated

1. Click the site manager button on the menu bar

Graphical user interface, application, Teams

Description automatically generated

1. Click new site button:
   * set protocol as SFTP – SSH File Transfer Protocol,
   * host is your VM’s public IP address,
   * port number is 2223,
   * Logon Type: select Key fille,
   * User is itzuser,
   * and browse and select the ssh key file you downloaded from the reservation details page (remember to change file permission by `sudo chmod 600 pem\_ibmcloudvsi\_download.cer` before this step if not yet)
   * click connect.

Graphical user interface, application

Description automatically generated

1. If you see the unknown host key dialog, choose always trust this host and click ok.

Graphical user interface, text, application

Description automatically generated

1. When the connection is established, you can transfer files from your desktop to the VM by dragging and dropping the file from the left to the right box.

Graphical user interface, application

Description automatically generated