# Getting Started

CockroachDB has a sophisticated and modern architecture and is designed for global scale. However, that complexity and scalability doesn’t imply a steep learning curve or barrier to entry. In this chapter we’ll help you get started with a CockroachDB installation and introduce you to the basics of working with a CockroachDB system.

## Installation

CockroachDB can be installed on virtually any flavor of desktop operating system within a few minutes. Alternatively, you can create a free CockroachDB cloud system, or run CockroachDB within a Docker container or Kubernetes cluster.

### Installing CockroachDB software

In most scenarios, you’ll want to have the CockroachDB software installed on your desktop computer, so lets start with that. You’ll find a full list of CockroachDB binaries at <https://www.cockroachlabs.com/docs/releases/> . From there you can pick your operating system and download the most recent version, or pick a previous version.

#### Installation on Microsoft Windows

From <https://www.cockroachlabs.com/docs/releases/?filters=windows> , click on the link for the release you’d like to download. Once downloaded, unzip the archive into a directory and add the subdirectory containing the +cockroachdb.exe+ to your PATH

Alternatively, you can download the file directly from a PowerShell prompt. Here, we download the zip fie for version 20.2.7, unzip it to a directory `c:\tools\cockroachdb`:

PS >

wget <https://binaries.cockroachdb.com/cockroach-v20.2.7.windows-6.2-amd64.zip>

-OutFile crdb.zip   
PS > mkdir c:\tools\cockroachdb  
PS > Expand-Archive -Path crdb.zip -DestinationPath C:\tools\cockroachdb\

At time of writing, it was necessary to download the GOLANG timezone zipfile and add an environment variable ZONEINFO pointing to that file:

PS > wget [https://github.com/golang/go/raw/master/lib/time/zoneinfo.zip -OutFile zoneinfo.zip](https://github.com/golang/go/raw/master/lib/time/zoneinfo.zip%20-OutFile%20zoneinfo.zip)

PS > Get-Item -Path Env:\ZONEINFO

Name Value

---- -----

ZONEINFO C:\tools\cockroachdb\cockroach-v20.2.7.windows-6.2-amd64\zoneinfo.zip

Once that is done you can test your installation by issuing the +cockroach demo+ command:

PS > cockroach demo

#

# Welcome to the CockroachDB demo database!

#

# You are connected to a temporary, in-memory CockroachDB cluster of 1 node.

#

#

# Enter \? for a brief introduction.

#

root@127.0.0.1:57574/movr> show databases;

database\_name | owner

----------------+--------

defaultdb | root

movr | root

postgres | root

system | node

(4 rows)

Time: 2ms total (execution 1ms / network 0ms)

#### Installation on MacOS

If you have the +brew+ package manager installed, then that is probably the easiest way to get started installing CRDB on Mac. In fact, even if you don’t have +brew+ installed, its probably easier to install it and then install +CRDB+ than to install CRDB manually.

To install +brew+, issue the following command from a terminal window:

/bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"

Once brew is installed, you can install CRDB with the following command:

guyharrison@macos ~ % brew install cockroachdb/tap/cockroach

==> **Tapping cockroachdb/tap**

Cloning into '/usr/local/Homebrew/Library/Taps/cockroachdb/homebrew-tap'...

…

==> **Installing cockroach from cockroachdb/tap**

….

To have launchd start cockroachdb/tap/cockroach now and restart at login:

brew services start cockroachdb/tap/cockroach

Or, if you don't want/need a background service you can just run:

cockroach start-single-node --insecure

==> **Summary**

🍺 /usr/local/Cellar/cockroach/20.2.7: 134 files, 184.8MB, built in 8 seconds

One of the great things about brew is that it sets up CRDB as a service, so you can issue +brew services start cockroach+ to start a background instance of CRDB.

However, If you don’t want to use brew, then you can download the CockroachDB binary directly, and copy the binary into your path. Visit <https://www.cockroachlabs.com/docs/releases/?filters=mac> to determine the path for the release you want, then use curl or wget to copy and decompress that binary:

curl https://binaries.cockroachdb.com/cockroach-v20.2.7.darwin-10.9-amd64.tgz | tar -xJ

You can then copy the binary into your PATH so you can execute cockroach commands from any shell:

cp -i cockroach-v20.2.7.darwin-10.9-amd64/cockroach /usr/local/bin/

Once you’ve installed CRDB either manually, or via brew, run the +cockroach demo+ command to start a demo instance and confirm that it is running:

guyharrison@macos ~ % cockroach demo

#

# Welcome to the CockroachDB demo database!

#

# You are connected to a temporary, in-memory CockroachDB cluster of 1 node.

#

# Enter \? for a brief introduction.

#

root@127.0.0.1:49418/movr> show databases;

database\_name | owner

----------------+--------

defaultdb | root

movr | root

postgres | root

system | node

(4 rows)

Time: 1ms total (execution 1ms / network 0ms)

root@127.0.0.1:49418/movr>

#### Installation on Linux

To perform a basic installation on Linux, visit <https://www.cockroachlabs.com/docs/releases/>? To obtain the latest release (or a specific version you are interested in), download it and unpack it. Of course, you can use curl or wget to obtain the tarball once you have determined it’s path:

$ wget https://binaries.cockroachdb.com/cockroach-v20.2.7.linux-amd64.tgz

2021-04-17 16:10:35 (1.83 MB/s) - 'cockroach-v20.2.7.linux-amd64.tgz’ saved [94338926/94338926]

$ sudo cp -i cockroach-v20.2.7.linux-amd64/cockroach

Once installed, run the +cockroach demo+ command to start a temporary local instance of CRDB and verify the installation

$ cockroach demo

#

# Welcome to the CockroachDB demo database!

#

# You are connected to a temporary, in-memory CockroachDB cluster of 1 node.

#

…

#

# Enter \? for a brief introduction.

#

root@127.0.0.1:44913/movr> show databases;

database\_name | owner

----------------+--------

defaultdb | root

movr | root

postgres | root

system | node

(4 rows)

Time: 1ms total (execution 1ms / network 0ms)

[root@127.0.0.1:44913/movr](mailto:root@127.0.0.1:44913/movr)>

### Starting a local single node server

As we’ve seen above, you can use the +cockroach demo+ command to start a temporary demo cluster. But if you want to start a single-node CockroachDB with persistent storage, you can use the +start=single-node+ option:

$ cockroach start-single-node --insecure --background

\*

\* WARNING: ALL SECURITY CONTROLS HAVE BEEN DISABLED!

\*

\* This mode is intended for non-production testing only.

\*

\* In this mode:

\* - Your cluster is open to any client that can access any of your IP addresses.

\* - Intruders with access to your machine or network can observe client-server traffic.

\* - Intruders can log in without password and read or write any data in the cluster.

\* - Intruders can consume all your server's resources and cause unavailability.

\*

\*

\* INFO: To start a secure server without mandating TLS for clients,

\* consider --accept-sql-without-tls instead. For other options, see:

\*

\* - https://go.crdb.dev/issue-v/53404/v20.2

\* - https://www.cockroachlabs.com/docs/v20.2/secure-a-cluster.html

\*

\*

\* WARNING: neither --listen-addr nor --advertise-addr was specified.

\* The server will advertise "mubuntu" to other nodes, is this routable?

\*

\* Consider using:

\* - for local-only servers: --listen-addr=localhost

\* - for multi-node clusters: --advertise-addr=<host/IP addr>

This will start a single node CockroachDB cluster with no security controls and run it as a background process. To connect to this server we can use the +cockroach sql+ command with the default connection string:

$ cockroach sql --insecure

#

# Welcome to the CockroachDB SQL shell.

# All statements must be terminated by a semicolon.

# To exit, type: \q.

#

# Server version: CockroachDB CCL v20.2.4 (x86\_64-unknown-linux-gnu, built 2021/01/21 00:08:24, go1.13.14) (same version as client)

# Cluster ID: 848d8b85-4000-484a-b4ad-8f2c76c68221

#

# Enter \? for a brief introduction.

#

root@:26257/defaultdb> show databases;

database\_name | owner

----------------+--------

defaultdb | root

postgres | root

system | node

(3 rows)

Time: 3ms total (execution 2ms / network 0ms)

root@:26257/defaultdb>

.insecure mode

\*\*\*\*

The use of the +insecure+ flag when starting a CockroachDB server is convenient for quickly starting a CockroachDB server but it is absolutely not appropriate for a production system. Please see Chapter ?? for instructions on setting up a properly secured production system.

\*\*\*\*

### Starting up CockroachDB in a docker container

If you have docker, you can start a CockroachDB single node instance inside a docker container.

You’ll need a persistent volume for data, so let’s create that first:

docker volume create crdb1

Then, we invoke +docker run+ to pull and start the latest cockroachdb docker image and start the server in single node, insecure mode:

$ docker run -d \

> --name=crdb1 \

> --hostname=crdb1 \

> -p 26257:26257 -p 8080:8080 \

> -v "crdb1:/cockroach/cockroach-data" \

> cockroachdb/cockroach:latest start-single-node \

> --insecure \

>

Unable to find image 'cockroachdb/cockroach:latest' locally

latest: Pulling from cockroachdb/cockroach

a591faa84ab0: Pull complete

…

6913e7a5719b8cb705c32540523885f6592270cf091ac1013cca66914b1aafe8

The output of the docker run command is the container identifier for the CockroachDB container. Using that containerId, we can connect to that container using the +cockroach sql+ command.

$ docker exec -it 6913e7a5719b8cb705c32540523885f6592270cf091ac1013cca66914b1aafe8 cockroach sql --insecure

#

# Welcome to the CockroachDB SQL shell.

# All statements must be terminated by a semicolon.

# To exit, type: \q.

#

# Server version: CockroachDB CCL v20.2.7 (x86\_64-unknown-linux-gnu, built 2021/03/29 17:52:00, go1.13.14) (same version as client)

# Cluster ID: 8fcbb9bb-ec7c-40dc-afe0-90306c87f5d7

#

# Enter \? for a brief introduction.

#

root@:26257/defaultdb> show databases;

database\_name | owner

----------------+--------

defaultdb | root

postgres | root

system | node

(3 rows)

Time: 3ms total (execution 3ms / network 0ms)

### Starting up a secure server

In the previous examples, we’ve used the +--insecure+ mode to start the server without needing to configure secure communications. This is a quick way to setup a test server, but is catestropically dangerous for anything that contains valuable data.

We’ll cover CockroachDB security in depth within Chapter 12, but for now to setup a secure server, we need to create security certificates to encrypt the communications channel and authenticate the client and server.

The following commands create the certificates. The Certificate Authority key will be held in +my-safe-directory+; the certificates themselves will be held in the +certs+ directory:

$ mkdir certs my-safe-directory

$ # CA certificate and keypair

$ cockroach cert create-ca \

> --certs-dir=certs \

> --ca-key=my-safe-directory/ca.key

$ # certificate and keypair for localhost

$ cockroach cert create-node localhost `hostname` --certs-dir=certs \

> --ca-key=my-safe-directory/ca.key

$ # certificate for the root user

$ cockroach cert create-client root \

> --certs-dir=certs \

> --ca-key=my-safe-directory/ca.key

We can now start the server and specify the directory containing the certificates:

$ # start single node

$ cockroach start-single-node --certs-dir=certs --background

\*

\* WARNING: neither --listen-addr nor --advertise-addr was specified.

\* The server will advertise "mubuntu" to other nodes, is this routable?

\*

\* Consider using:

\* - for local-only servers: --listen-addr=localhost

\* - for multi-node clusters: --advertise-addr=<host/IP addr>

\*

\*

$ \*

\* INFO: Replication was disabled for this cluster.

\* When/if adding nodes in the future, update zone configurations to increase the replication factor.

\*

Now when connecting, we must specify the certificates directory. If we are connecting from a remote host then we would need to copy the certificates to that host.

$ cockroach sql --certs-dir=certs --certs-dir=certs

#

# Welcome to the CockroachDB SQL shell.

# All statements must be terminated by a semicolon.

# To exit, type: \q.

#

# Server version: CockroachDB CCL v20.2.4 (x86\_64-unknown-linux-gnu, built 2021/01/21 00:08:24, go1.13.14) (same version as client)

# Cluster ID: f908d29e-1fb6-40b8-9e1f-a2a0a3763603

#

# Enter \? for a brief introduction.

#

root@:26257/defaultdb>

### Remote connection

In the previous examples, we’ve connected to a server running on the same host as our client. This is pretty unusual in the real world, where we would normally be connecting to a server on another machine. Typically, we’d specify the url parameter to identify the server concerned. For instance to connect to a server on the +mubuntu+ server on the default port we could issue the following command:

$ cockroach sql --certs-dir=certs --url postgresql://root@mubuntu:26257/defaultdb

#

# Welcome to the CockroachDB SQL shell.

# All statements must be terminated by a semicolon.

# To exit, type: \q.

#

# Server version: CockroachDB CCL v20.2.4 (x86\_64-unknown-linux-gnu, built 2021/01/21 00:08:24, go1.13.14) (same version as client)

# Cluster ID: f908d29e-1fb6-40b8-9e1f-a2a0a3763603

#

# Enter \? for a brief introduction.

#

root@mubuntu:26257/defaultdb>

Note the +postgressql+ identifier in the url. CockroachDB uses the postgreSQL wire protocol, so we can use PostgreSQL compatible URLs. The full form or a URL is:

postgresql://[user[:passwd]@]host[:port]/[db][?parameters...]

At a minimum, the hostname must be specified.

### Creating a Cockroach Cloud cluster

### Creating a Kubernetes cluster

# Outline

The CockroachDB client is built into the CockroachDB binary which can be installed on any platform.

* Installing on Mac
* Installing on Windows
* Installing on Linux

#### PostgreSQL clients

Because CockroachDB is compatible with PostgreSQL, you may already have a compatible client installed. The `pg` client works fine with CockroachDB, although the CockroachDB client is preferred.

* Brief overview of the psql client

#### GUI Clients

Many GUI clients support Postgres and some support CockroachDB directly. Here we mention a few with DBEaver probably the recommended client (?)

### Connecting to the CockroachDB cloud

CockroachCloud is the simplest way to get a CockroachDB sever up and running. CockroachDB offers a Forever-free plan which is ideal for getting started.

### Installing a local CockroachDB server

If you want to setup a local CockroachDB server it’s easy to install a single node cluster on any platform.

* Installing on Mac
* Installing on Windows
* Installing on Linux
* Installing with Docker

### Installing a test cluster

If you want to work with a CockroachDB cluster, you might want to install a K8S hosted cluster on a K8S cluster you have lying around the house or (maybe) on minikube.

* Preview of Kubernetes installation using K8S operator

### Connecting to a cluster

Now that you have a server and a cluster, it’s time to connect!

NB: We’ll have shown connection as the last step in every install tutorial, but here we’ll talk about the CockroachDB connection string and some of it’s important options

## Exploring CockroachDB

Let’s get started with CockroachDB by:

* Installing demo data using CockroachDB command line
* Listing and describing databases and tables
* Creating a table
* Issuing SELECT queries
* Inserting, updating deleting data
* Exploring the information schema

## Working with programming languages

Here we will show the simplest possible example of a program that connects to CockroachDB and selects some data. We’ll do this for:

* NodeJS
* GoLang
* Java
* Python