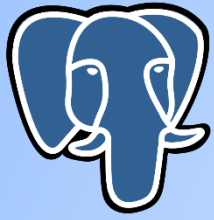




PostgreSQL

PostgreSQL

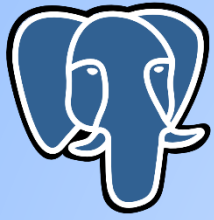
Configuration and Server Control



PostgreSQL

Pg_ctl utility

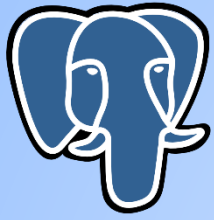
- ▶ Pg_ctl is the utility for initializing a postgresql database cluster, starting, stopping, or restarting the postgresql server, or displaying the status of running server
 - ▶ In start mode a new server is launched
 - ▶ In stop mode, the server that is running in specified data directory is shut down
 - ▶ Restart mode effectively executes a stop followed by a start
 - ▶ Status mode to check whether a server is running in specified mode or not



PostgreSQL

Starting the Database Server

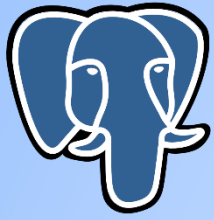
- ▶ Before anyone can access the database , you must start the database server.
- ▶ The database server program is called as *postgres*
- ▶ The simplest way to start the server is :
 - ▶ **\$ /usr/pgsql-12/bin/pg_ctl -D /var/lib/pgsql/12/data start -l logfile**



PostgreSQL

Starting And Stopping The Database Server

- ▶ The word "server" refers to the database server and its processes. The word "service" refers to the operating system wrapper by which the server gets called.
- ▶ Specific command to start the server:
 - ▶ **pg_ctl -D /var/lib/pgsql/data start**
- ▶ Specific command to stop the server using fast mode:
 - ▶ **pg_ctl -D /var/lib/pgsql/data stop**
 - ▶ When you do a stop, all users have their transactions aborted and all connections are disconnected.



PostgreSQL

There's more...

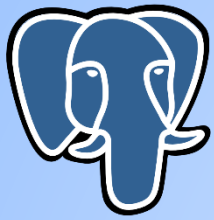
- ▶ Specific command to stopping the server in an emergency:
 - ▶ **`pg_ctl -D /var/lib/pgsql/data stop -m immediate`**
 - ▶ When you do an immediate stop, all users have their transactions aborted and all connections are disconnected. There is no clean shutdown, nor is there politeness of any kind.
 - ▶ An immediate mode stop is similar to a database crash.



PostgreSQL

Running Multiple Servers On One System

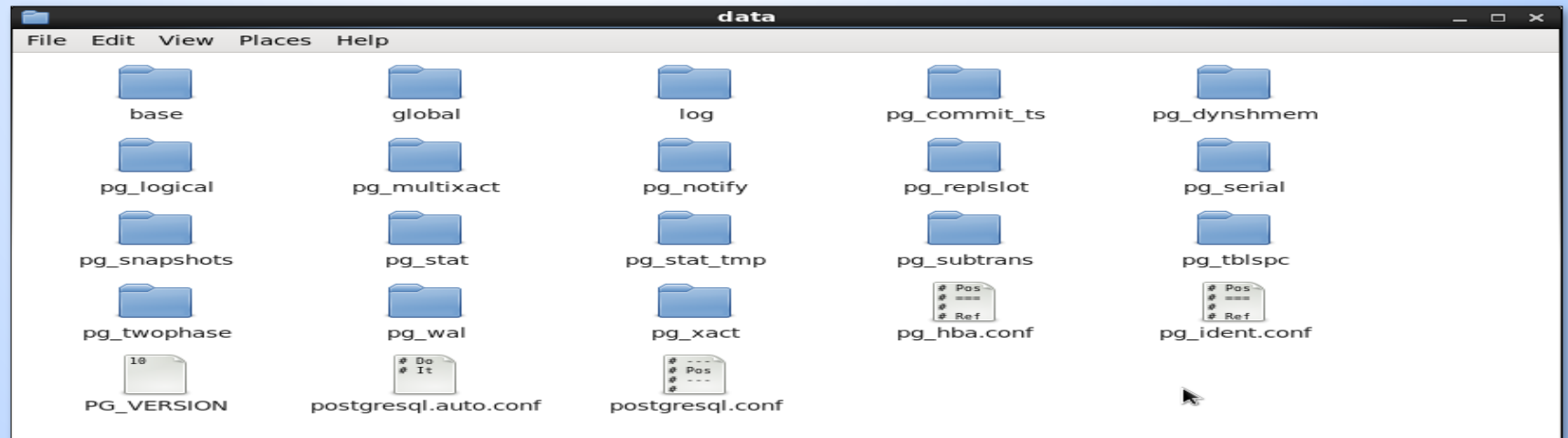
- ▶ Running multiple PostgreSQL servers on one physical system is possible if it is convenient for your needs.
- ▶ Create new data directories for this second instance and let postgres user own them
 - ▶ **mkdir /var/lib/pgsql/12/data2**
 - ▶ **chown postgres.postgres /var/lib/pgsql/12/data2**
 - ▶ Initialize your data directory
 - ▶ Modify the port parameter in the postgresql.conf file and start
- ▶ PostgreSQL servers are controlled using pg_ctl. Everything else is a wrapper of some kind around this utility



PostgreSQL

Locating Database Server Files

- The initdb utility populates a given data directory.
- The path to data directory :
 - /var/lib/pgsql/12/data
- Once you've located the data directory, you can look for the files that comprise the PostgreSQL database server





Locating the Configurations Files

PostgreSQL

- Server files are initially stored in a location referred to as the data directory
- The default data directory location is - **`/var/lib/pgsql/R.r/data`**
- Execute the SQL query:
 - **`SELECT name, setting FROM pg_settings WHERE category = 'File Locations';`**

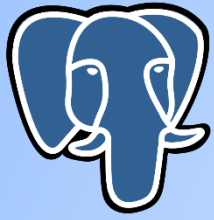
```
postgres=# select name, setting from pg_settings where category = 'File Locations';
 name          |          setting
-----+-----
 config_file   | /var/lib/pgsql/10/data/postgresql.conf
 data_directory | /var/lib/pgsql/10/data
 external_pid_file |
 hba_file      | /var/lib/pgsql/10/data/pg_hba.conf
 ident_file    | /var/lib/pgsql/10/data/pg_ident.conf
(5 rows)
```




Configuration file - postgresql.conf

- ▶ Parameters are case-insensitive
- ▶ Each line holds one parameter in 'key-value' pair separated by '=', example : `shared_buffers = 128MB`
- ▶ The setting value types can be the following:
 - ▶ Boolean
 - ▶ `enable_scan = on`
 - ▶ Integer
 - ▶ `max_connections = 100`
 - ▶ Enum
 - ▶ `listen_addresses` (comma-separated list of addresses)
 - ▶ Floating point
 - ▶ `cpu_operator_cost`
 - ▶ String
 - ▶ `archive_command`

PostgreSQL



PostgreSQL

Enabling access for network/remote users

- ▶ By default, PostgreSQL allows user if database user is the same as the system's username
- ▶ To enable other connections:
 - ▶ `listen_addresses = '*'`
 - ▶ The `listen_addresses` parameter specifies which IP addresses to listen to



PostgreSQL

Configuration file – pg_hba.conf

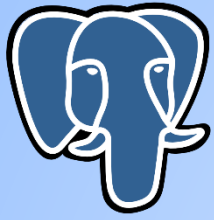
- ▶ Change the **host-based authentication (HBA)** file to refuse all incoming connections.

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
local all all reject
host all all 0.0.0.0/0 reject
```

If you still want superuser access, then try something like the following:

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
local all postgres peer
local all all reject
host all all 0.0.0.0/0 reject
```

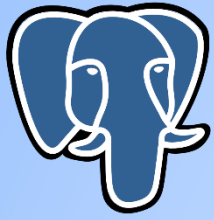
- ▶ The rules are specified in a file and applied by the postmaster process when connections are attempted.



PostgreSQL

Changing Parameters

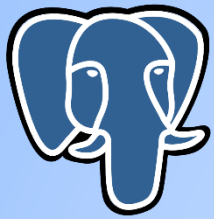
- ▶ PostgreSQL allows you to set some parameter settings for each session or transaction.
 - ▶ You can change the value of a setting during your session, This value will then be used for every future transaction, like this:
 - ▶ **SET work_mem = '16MB';**
 - ▶ You can also change it only for the duration of the "current transaction":
 - ▶ **SET LOCAL work_mem = '16MB';**
 - ▶ The setting will last until you issue this command:
 - ▶ **RESET work_mem;**



PostgreSQL

Finding the Current Configuration Settings

- ▶ We can use the **SHOW** command like this:
 - ▶ **postgres=# SHOW work_mem;**
- ▶ Another way of finding the current settings is to access a PostgreSQL catalog view named **pg_settings**:
 - ▶ **SELECT * FROM pg_settings WHERE name = 'work_mem';**
- ▶ Thus, you can use the SHOW command to retrieve the value for a setting, or you can access the full details via the catalog table.



PostgreSQL

Parameters Are At Non-default Settings

- ▶ We write a SQL query that lists all parameter values, excluding those whose current value is either the default or set from a configuration file:
 - ▶ `postgres=# SELECT name, source, setting FROM pg_settings WHERE source != 'default' AND source != 'override' ORDER by 2, 1;`

name	source	setting
application_name	client	psql
client_encoding	client	UTF8
DateStyle	configuration file	ISO, DMY
default_text_search_config	configuration file	pg_catalog.english
dynamic_shared_memory_type	configuration file	posix
lc_messages	configuration file	en_GB.UTF-8
lc_monetary	configuration file	en_GB.UTF-8
lc_numeric	configuration file	en_GB.UTF-8
lc_time	configuration file	en_GB.UTF-8
log_timezone	configuration file	Europe/Rome
max_connections	configuration file	100
port	configuration file	5460
shared_buffers	configuration file	16384
TimeZone	configuration file	Europe/Rome
max_stack_depth	environment variable	2048



PostgreSQL

Updating The Parameter File

- ▶ All the parameters can be set in the parameter file, which is known as **postgresql.conf**.
 - ▶ Some of the parameters take effect only when the server is first started. Example- shared Buffers
 - ▶ . To Alter:
 - ▶ **ALTER SYSTEM SET shared_buffers = '1GB';**
 - ▶ After changing the required parameters, we issue a reload operation to the server, forcing PostgreSQL to reread the postgresql.conf file. Example : **pg_ctl reload**
 - ▶ Some other parameters require a restart of the server for changes to take effect, for instance, max_connections, listen_addresses etc.



PostgreSQL – Memory Management

- ▶ Parameters recommended for memory management
- ▶ Mainly the following parameters:
 - ▶ About **shared_buffers**:
 - ▶ Below 2GB, set to 20% of total system memory.
 - ▶ Below 32GB, set to 25% of total system memory.
 - ▶ Above 32GB, set to 8GB
 - ▶ About **work_mem**
 - ▶ Start low: 32-64MB
 - ▶ About **maintenance_work_mem**
 - ▶ 10% of system memory, up to 1GB
 - ▶ Maybe even higher if you are having VACUUM problems

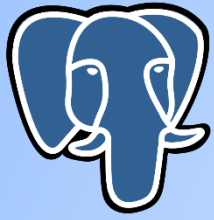
PostgreSQL



PostgreSQL

PostgreSQL parameters

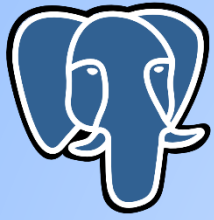
- ▶ Listen_addresses (list) - Sets the host name or IP address(es) to listen to. Set your listen_address as restrictively as possible; '*' should only be used for development machines
- ▶ Shared memory is accessible by all postgres server processes.
 - ▶ The biggest chunk of memory is shared_buffers. Postgres suggests to use 25% of RAM
 - ▶ The WAL buffers are normally much smaller, 1/32 of shared buffers is default
 - ▶ Ideally, your tables are not too large and your RAM is not too small, so you can afford setting **autovacuum_work_mem** to reflect your smallest table size
 - ▶ No more than **autovacuum_max_workers** workers, each uses **maintenance_work_mem** or **autovacuum_work_mem** of RAM



PostgreSQL

PostgreSQL parameters

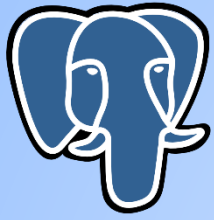
- ▶ Port (integer) - Sets the TCP port the server listens on. Alternate ports are primarily useful for running several versions, or instances, of PostgreSQL on one machine. However, if you're using an alternate port to support several versions
- ▶ max_connections(integer) - Sets the maximum number of concurrent connections. In general, if you need more than 1000 connections, you should probably be making more use of connection pooling
- ▶ shared_buffers(memory) - Sets the number of shared memory buffers used by the server. A memory quantity defining PostgreSQL's "dedicated" RAM, which is used for connection control, active operations, and more. Also note that shared_buffers over 2GB is only supported on 64-bit systems
- ▶ work_mem(integer) - Sets the maximum memory to be used for query workspaces. This much memory can be used by each internal sort operation and hash table before switching to temporary disk files



PostgreSQL parameters

PostgreSQL

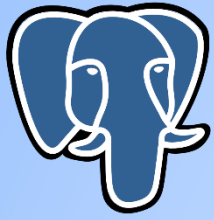
- ▶ `temp_buffers (memory)` - Sets the maximum number of temporary buffers used by each session. Currently used only for holding temporary tables in memory
- ▶ `maintenance_work_mem(integer)` - Sets the maximum memory to be used for maintenance operations. This includes operations such as `VACUUM` and `CREATE INDEX`. Sets the limit for the amount that autovacuum, manual vacuum, bulk index build and other maintenance routines are permitted to use
- ▶ `autovacuum(bool)` - Starts the autovacuum subprocess. Starts the daemon which cleans up your tables and indexes, preventing bloat and poor response times
- ▶ `autovacuum_naptime(integer)` - Time to sleep between autovacuum runs



PostgreSQL

PostgreSQL parameters

- `wal_buffers(integer)` - Sets the number of disk-page buffers in shared memory for WAL
- `wal_writer_delay(integer)` - WAL writer sleep time between WAL flushes.
- `checkpoint_timeout(integer)` - Sets the maximum time between automatic WAL checkpoints
- `bgwriter_delay(time)` - Background writer sleep time between rounds. Thanks to `bgwriter` autotuning, it should no longer be necessary for most users to touch the `bgwriter` settings
- `bgwriter_lru_maxpages(integer)` - Background writer maximum number of LRU pages to flush per round
- `archive_command(string)` - Sets the shell command that will be called to archive a WAL file. All of the Archiving settings are part of a Point In Time Recovery or Warm Standby configuration -- `archive_mode`
- `archive_mode(bool)` - Allows archiving of WAL files using `archive_command`.



PostgreSQL parameters

PostgreSQL

- ▶ `log_destination` (string) - Sets the destination for server log output. Valid values – `stderr`, `csvlog`
- ▶ `log_directory`(string) - Sets the destination directory for log files
- ▶ `log_filename`(string) - Sets the file name pattern for log files
- ▶ `log_rotation_age`(integer) - Automatic log file rotation will occur after N minutes
- ▶ `log_rotation_size` (integer) - Automatic log file rotation will occur after N kilobytes.
- ▶ `search_path`(list) - Sets the schema search order for names that are not schema-qualified



The Basic Server Configuration Checklist

PostgreSQL

- PostgreSQL arrives configured for use on a shared system, though many people want to run dedicated database systems
- or instance, PostgreSQL is able to store information to the disk when the available memory is too Small
- It's better to be conservative. It is good practice to set a low value in your postgresql.conf and increment slowly to ensure that you get the benefits from each change
- If you're doing heavy write activity, then you may want to set wal_buffers to a much higher value than the default.
- If your database has many large queries, you may wish to set work_mem to a value higher than the default.
- Ensure that autovacuum is turned on, unless you have a very good reason to turn it off.