Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
File Locations									
config_file	string	ConfigDir/postgresql.conf	default					Sets the server's main configuration file.	Can only be changed via command-line switch for obvious reasons. Useful primarily for testing different configuration options, or for automated restart with different configuration options.
data_directory	string	ConfigDir	default				postmaster	Sets the server's data directory.	Supports the ability to distribute files according to sysadmin or operating system defined schemes, or for launching multiple postmaster instances using the same binaries. Most of the time, it's better to use configuration options to define these locations so that all PostgreSQL binaries default to the correct paths.
hba_file	string	ConfigDir/pg_hba.conf	default				postmaster	Sets the server's "hba" configuration file.	
ident_file	string	ConfigDir/pg_ident.conf	default				postmaster	Sets the server's "ident" configuration file.	
external_pid_file	string	None	default				postmaster	Writes the postmaster PID to the specified file.	Creates an extra copy of the process ID. Used for server administration tools which need a copy of the process ID in a specific directory.
Connections and Authentication									
Connection Settings									
listen_addresses	list	localhost	localhost,1 address				postmaster	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
port	integer	5432	default	1	65535		postmaster	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, it's often better to compile in the port number.
max_connections	integer	100	under 1000	1	INT_MAX		postmaster	Sets the maximum number of concurrent connections.	Should be set to the maximum number of connections which you expect to need at peak load. Note that each connection uses shared_buffer memory, as well as additional non-shared memory, so be careful not to run the system out of memory. In general, if you need more than 1000 connections, you should probably be making more use of connection pooling.
superuser_reserved_connections	integer	3	default	0	INT_MAX		postmaster	Sets the number of connection slots reserved for superusers.	You should have at least one superuser connection open for troubleshooting at all times. So if you run more than two concurrent regular administrative tasks, you'll need more reserved connections. Note that this number is taken from max_connections, not in addition to it.
unix_socket_directory	string	tmpdir					postmaster	Sets the directory where the Unix-domain socket will be created.	By default, the PostgreSQL unix socket is world-writable, which allows some kinds of man-in-the-middle attacks. Move it to an alternate directory and set its permissions to be group-restricted to make this kind of attack more difficult.

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
unix_socket_group	string		postgres				postmaster	Sets the owning group of the Unix-domain socket. The owning user of the socket is always the user that starts the server.	
unix_socket_permissions	integer	0777	770	0	777		postmaster	Sets the access permissions of the Unix-domain socket. Unix-domain sockets use the usual Unix file system permission set. The parameter value is expected to be an numeric mode specification in the form accepted by the chmod and umask system calls. (To use the customary octal format the number must start with a 0 (zero).)	
bonjour_name	string	machine name	default				postmaster	Sets the Bonjour broadcast service name.	Bonjour support must be compiled in and activated on the host machine to be live. You'll want alternate names if you have several instances of PostgreSQL on the same machine.
Connection Persistence									
authentication_timeout	time	60	decrease	1	600	s	sighup	Sets the maximum allowed time to complete client authentication.	For production databases, it's important that this value be synchronized with the timeout on the application server side. Most web applications will want a shorter timeout, like 20s.
tcp_keepalives_count	integer	0	default	0	INT_MAX		user	Maximum number of TCP keepalive retransmits. This controls the number of consecutive keepalive retransmits that can be lost before a connection is considered dead. A value of 0 uses the system default.	The three tcp_keepalive settings help manage a system which tends to have "undead" connection/query processes. For systems which support them, you can regulate checking that connections are still "live" end-to-end to kill them off. Not needed if you're not having a problem.
tcp_keepalives_idle	integer	0	default	0	INT_MAX	s	user	Time between issuing TCP keepalives. A value of 0 uses the system default.	
tcp_keepalives_interval	integer	0	default	0	INT_MAX	s	user	Time between TCP keepalive retransmits. A value of 0 uses the system default.	
Security and Authentication									
db_user_namespace	bool	off	do not use				sighup	Enables per-database user names.	This setting is a hack to work around the lack of per-database users in PostgreSQL. Unless you desperately need it, avoid this setting as it will eventually be replaced by something more maintainable.
password_encryption	bool	on	on				user	Encrypt passwords. When a password is specified in CREATE USER or ALTER USER without writing either ENCRYPTED or UNENCRYPTED, this parameter determines whether the password is to be encrypted.	There is no good reason for this to be set to “off”.

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
ssl	bool	off	varies				postmaster	Enables SSL connections.	One of several different settings to turn on SSL connections for PostgreSQL. SSL is a very good idea for highly secure setups. In addition, you must compile in SSL support and set SSL connections in pg_hba.conf, as well as configuring SSL itself.
krb_caseins_users	bool	off	varies				postmaster	Sets whether Kerberos and GSSAPI user names should be treated as case-insensitive.	
krb_realm	string		varies				postmaster	Sets realm to match Kerberos and GSSAPI users against.	
krb_server_hostname	string		varies				postmaster	Sets the hostname of the Kerberos server.	
krb_server_keyfile	string		varies				postmaster	Sets the location of the Kerberos server key file.	
krb_srvname	string	postgres	varies				postmaster	Sets the name of the Kerberos service.	Speak with your sysadmin or network security about how to set the various kerberos settings to match your local kerberos setup. Kerberos support must be compiled in to PostgreSQL, and set in pg_hba.conf.
Resource Usage									
Memory									
shared_buffers	memory	varies 512kB to 8MB	AvRAM / 4	64kB	8192GB	8kB	postmaster	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. However, since PostgreSQL also needs free RAM for file system buffers, sorts and maintenance operations, it is not advisable to set shared_buffers to a majority of RAM. Note that increasing shared_buffers often requires you to increase some system kernel parameters, most notably SHMMAX and SHMALL. See Operating System Environment: Managing Kernel Resources in the PostgreSQL documentation for more details. Also note that shared_buffers over 2GB is only supported on 64-bit systems.
work_mem	integer	1MB	(AvRAM / max_connections) OR (AvRAM / 2 * max_connections)	64kB	2GB	kB	user	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.	Sets the limit for the amount of non-shared RAM available for each query operation, including sorts and hashes. This limit acts as a primitive resource control, preventing the server from going into swap due to overallocation. Note that this is non-shared RAM per operation, which means large complex queries can use multple times this amount. Also, work_mem is allocated by powers of two, so round to the nearest binary step. The second formula is for reporting and DW servers which run a lot of complex queries.

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
temp_buffers	memory	8MB	default	800kB	8192GB	8kB	user	Sets the maximum number of temporary buffers used by each session.	Currently used only for holding temporary tables in memory. If your application requires heavy use of temporary tables (many proprietary reporting engines do) then you might want to increase this substantially. However, be careful because this is non-shared RAM which is allocated per session. Otherwise, the default is fine.
max_prepared_transactions	integer	5	0 or max_connections	0	INT_MAX		postmaster	Sets the maximum number of simultaneously prepared transactions.	Most applications do not use XA prepared transactions, so should set this parameter to 0. If you do require prepared transactions, you should set this equal to max_connections to avoid blocking. May require increasing kernel memory parameters.
Kernel Resources									
max_files_per_process	integer	1000	default	25	INT_MAX		postmaster	Sets the maximum number of simultaneously open files for each server process.	
max_stack_depth	memory	2MB	default	100kB	2GB	kB	superuser	Sets the maximum stack depth, in kilobytes.	
shared_preload_libraries	string		default				postmaster	Lists shared libraries to preload into server.	Primarily used for custom C libraries (data types, stored procedures) which you expect your application to use heavily. Trades memory overhead for these libraries against load time, so really should only be used for libraries you expect most queries to require.
Maintenance									
Memory									
maintenance_work_mem	integer	16MB	(AvRAM / 8)	1MB	2GB	kB	user	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. Setting it to a moderately high value will increase the efficiency of vacuum and other operations. Applications which perform large ETL operations may need to allocate up to 1/4 of RAM to support large bulk vacuums.
Free Space Map									
max_fsm_pages	integer	76800	((DBSize / 8kB) / 8 to 16)	1000	INT_MAX		postmaster	Sets the maximum number of disk pages for which free space is tracked.	Sets the maximum number of data pages with free space which the Postmaster will track. Setting this too low can lead to table bloat and need for VACUUM FULL. Should be set to the maximum number of data pages you expect to be updated or deleted between vacuums. Increasing this setting may require increasing system kernel parameters. Additionally, the recommended formula is based on the default autovacuum settings; if you change the autovacuum parameters, then you may need to adjust this setting to match. Large databases with a lot of historical rows won't require as many FSM pages.

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
max_fsm_relations	integer	1000	default	100	INT_MAX		postmaster	Sets the maximum number of tables and indexes for which free space is tracked.	The default setting is plenty for most installations. If you have an application that requires thousands of tables, however, make sure that this setting is at least as high as the total number of tables in all databases plus 20 per database for system tables.
Manual Vacuum									
vacuum_cost_delay	integer	0		0	1000	ms	user	Vacuum cost delay in milliseconds.	Most of the time, you will want manual vacuum to execute without vacuum_delay, especially if you're using it as part of ETL. If for some reason you can't use autovacuum on an OLTP database, however, you may want to increase this to 20ms to decrease the impact vacuum has on currently running queries. Will cause vacuum to take up to twice as long to complete.
vacuum_cost_limit	integer	200	default	1	10000		user	Vacuum cost amount available before napping.	
vacuum_cost_page_dirty	integer	20	default	0	10000		user	Vacuum cost for a page dirtied by vacuum.	
vacuum_cost_page_hit	integer	1	default	0	10000		user	Vacuum cost for a page found in the buffer cache.	
vacuum_cost_page_miss	integer	10	default	0	10000		user	Vacuum cost for a page not found in the buffer cache.	
vacuum_freeze_min_age	integer	100000000	default	0	1000000000		user	Minimum age at which VACUUM should freeze a table row.	For unusually high-volume databases, you may want to increase this setting to 500million. Maximum setting is 1/2 of autovaccuum_max_freeze_age
Autovacuum									
autovacuum	bool	on	on				sighup	Starts the autovacuum subprocess.	Starts the daemon which cleans up your tables and indexes, preventing bloat and poor response times. The only reason to set it to "off" is for databases which regularly do large batch operations like ETL. Note that you can adjust the frequency or stop autovacuum on individual tables by adding rows to the pg_autovacuum system table.
autovacuum_analyze_scale_factor	real	0.1	default	0	100		sighup	Number of tuple inserts, updates or deletes prior to analyze as a fraction of reltuples.	
autovacuum_analyze_threshold	integer	50	default	0	INT_MAX		sighup	Minimum number of tuple inserts, updates or deletes prior to analyze.	

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
autovacuum_freeze_max_age	integer	200000000	default	100000000	2000000000		postmaster	Age at which to autovacuum a table to prevent transaction ID wraparound.	Triggers autovacuum automatically if a table is about to suffer from XID rollover. For very large / high-volume databases this setting might be low and need to be increased to 1000000000. Do not increase it any further, though, or you risk data loss.
autovacuum_max_workers	integer	3	default	1	536870911		postmaster	Sets the maximum number of simultaneously running autovacuum worker processes.	If you have an installation with many tables (100's to 1000's) or with some tables which autovacuum takes hours to process, you may want to add additional autovacuum workers so that multiple tables can be vacuumed at once. Be conservative, though, as each autovacuum worker will utilize a separate CPU core, memory and I/O.
autovacuum_naptime	integer	60	default	1	2147483	s	sighup	Time to sleep between autovacuum runs.	
autovacuum_vacuum_cost_delay	integer	20	default	-1	1000	ms	sighup	Vacuum cost delay in milliseconds, for autovacuum.	If autovacuum is having too much of a performance impact on running queries, you might want to increase this setting to 50ms. However, this will also cause individual vacuum tasks to take longer.
autovacuum_vacuum_cost_limit	integer	-1	default	-1	10000		sighup	Vacuum cost amount available before napping, for autovacuum.	
autovacuum_vacuum_scale_factor	real	0.2	default	0	100		sighup	Number of tuple updates or deletes prior to vacuum as a fraction of reltuples.	
autovacuum_vacuum_threshold	integer	50	default	0	INT_MAX		sighup	Minimum number of tuple updates or deletes prior to vacuum.	
WAL and Checkpoints									
Memory									
wal_buffers	integer	8	8MB	4	INT_MAX	8kB	postmaster	Sets the number of disk-page buffers in shared memory for WAL.	The default setting is small enough to cause choke on SMP machines; increasing to 8MB prevents that. There is no demonstrated benefit to increasing wal_buffers further. Also defines the maximum amount of data lost if synchronous_commit = off and the database shuts down.
Synch to Disk									
fsync	bool	on	on				sighup	Forces synchronization of updates to disk. The server will use the fsync() system call in several places to make sure that updates are physically written to disk. This insures that a database cluster will recover to a consistent state after an operating system or hardware crash.	Never turn off. Setting fsync=off is the equivalent of saying “I don't care about my data, I can recreate the database from scratch if I have to.” If synch activity is a performance concern, see synchronous_commit.

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
full_page_writes	bool	on	on				sighup	Writes full pages to WAL when first modified after a checkpoint. A page write in process during an operating system crash might be only partially written to disk. During recovery, the row changes stored in WAL are not enough to recover. This option writes pages when first modified after a checkpoint to WAL so full recovery is possible.	This is PostgreSQL's triple-check on transaction log integrity. Leave it on unless you have enough in-depth knowledge of your filesystem and hardware to be certain that torn page writes of log segments are completely prevented. Solaris/ZFS users claim to be able to turn this off, but that has not been destruction-tested.
synchronous_commit	bool	on	on				user	Sets immediate fsync at commit.	If data integrity is less important to you than response times (for example, if you are running a social networking application or processing logs) you can turn this off, making your transaction logs asynchronous. This can result in up to wal_buffers or wal_writer_delay * 2 worth of data in an unexpected shutdown, but your database will not be corrupted. Note that you can also set this on a per-session basis, allowing you to mix “lossy” and “safe” transactions, which is a better approach for most applications.
wal_sync_method	string	OS-dependent	default				sighup	Selects the method used for forcing WAL updates to disk.	On install, PostgreSQL figures out the best method for your OS. It's pretty good at this point; don't change the default. Note that the value of "fsync" shown in your postgresql.conf file is not necessarily the setting the server is using; try SHOW instead.
wal_writer_delay	integer	200	default	1	10000	ms	sighup	WAL writer sleep time between WAL flushes.	Defines the maximum data (in time) that can be lost if synchronous_commit=off and the database shuts down. Because of long transactions, actual data lost can be up to twice this time. Has no effect if synchronous_commit=on. If you are going to turn synchronous_commit=off server-wide, you should probably lower this to prevent too much data loss.
Checkpoints									
checkpoint_completion_target	real	0.5	default	0	1		sighup	Time spent flushing dirty buffers during checkpoint, as fraction of checkpoint interval.	Defines the fraction of one checkpoint_interval over which to spread checkpoints. There currently isn't any performance research to define the benefits of a lower or higher setting.
checkpoint_segments	integer	3	16 to 128	1	INT_MAX		sighup	Sets the maximum distance in log segments between automatic WAL checkpoints.	For most high-volume OTLP databases and DW you will want to increase this setting significantly. Alternately, just wait for checkpoint warnings in the log before increasing it. Increasing this setting can make recovery in the event of unexpected shutdown take longer. Maximum disk space required is (checkpoint_segments * 2 + 1) * 8MB, so make sure you have that much available before setting it.
checkpoint_timeout	integer	300	default	30	3600	s	sighup	Sets the maximum time between automatic WAL checkpoints.	If you do really large ETL batches, you may want to increase this setting to the maximum length of a batch run.
checkpoint_warning	integer	30	default	0	INT_MAX	s	sighup	Enables warnings if checkpoint segments are filled more frequently than this. Write a message to the server log if checkpoints caused by the filling of checkpoint segment files happens more frequently than this number of seconds. Zero turns off the warning.	

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
Background Writer									
bgwriter_delay	time	200	default	10	10000	ms	sighup	Background writer sleep time between rounds.	Thanks to bgwriter autotuning, it should no longer be necessary for most users to touch the bgwriter settings. Only modify these if you have a demonstrated issue shown by checkpoint spikes and monitoring pg_stat_bgwriter. Laptop PostgreSQL users may want to increase bgwriter_delay to 60s to decrease I/O activity, since it is no longer possible to turn the bgwriter off.
bgwriter_lru_maxpages	integer	100	default	0	1000		sighup	Background writer maximum number of LRU pages to flush per round.	
bgwriter_lru_multiplier	real	2	default	0	10		sighup	Background writer multiplier on average buffers to scan per round.	
Archiving									
archive_command	string	(disabled)	varies				sighup	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. Please see the Backup and Restore section for more information.
archive_mode	bool	off	varies				postmaster	Allows archiving of WAL files using archive_command.	
archive_timeout	integer	0	varies	0	INT_MAX	s	sighup	Forces a switch to the next xlog file if a new file has not been started within N seconds.	
Planner Cost Constants									
commit_delay	integer	0	0	0	100000		user	Sets the delay in microseconds between transaction commit and flushing WAL to disk.	commit_delay and commit_siblings are legacy, and will probably be dropped from future versions of PostgreSQL. You should use synchronous_commit and related settings for batch commits instead.
commit_siblings	integer	5	1	1	1000		user	Sets the minimum concurrent open transactions before performing commit_delay.	
Query Tuning									
Planner Cost Constants									
effective_cache_size	integer	65536	(AvRAM * 0.75)	1	INT_MAX	8kB	user	Sets the planner's assumption about the size of the disk cache. That is, the portion of the kernel's disk cache that will be used for PostgreSQL data files. This is measured in disk pages, which are normally 8 kB each.	Tells the PostgreSQL query planner how much RAM is estimated to be available for caching data, in both shared_buffers and in the filesystem cache. This setting just helps the planner make good cost estimates; it does not actually allocate the memory.

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
seq_page_cost	real	1.0	default	0	REAL_MAX		user	Sets the planner's estimate of the cost of a sequentially fetched disk page.	The main reason to modify seq_page_cost is to try to get planner costs to more-or-less indicate execution times in milleseconds. All other costs change relative to this cost automatically.
cpu_tuple_cost	real	0.0100	0.0030	0	REAL_MAX		user	Sets the planner's estimate of the cost of processing each tuple (row).	
cpu_index_tuple_cost	real	0.0050	0.0010	0	REAL_MAX		user	Sets the planner's estimate of the cost of processing each index entry during an index scan.	
cpu_operator_cost	real	0.0025	0.0005	0	REAL_MAX		user	Sets the planner's estimate of the cost of processing each operator or function call.	The various cpu_* costs don't take into account improvements in CPU speed. Until we can set a new default level, I suggest setting them to the levels shown here.
random_page_cost	real	4.0	default	0	REAL_MAX		user	Sets the planner's estimate of the cost of a nonsequentially fetched disk page.	There's a lot of conventional wisdom about tinkering with random_page_cost. However, for most databases if the other planner cost constants are set correctly, and ANALYZE is run correctly, it's generally not necessary to touch random_page_cost. This setting is really a last resort to be lowered if indexes aren't being preferred often enough.
Planner Method Configuration									
enable_bitmapscan	bool	on	on				user	Enables the planner's use of bitmap-scan plans.	
enable_hashagg	bool	on	on				user	Enables the planner's use of hashed aggregation plans.	
enable_hashjoin	bool	on	on				user	Enables the planner's use of hash join plans.	
enable_indexscan	bool	on	on				user	Enables the planner's use of index-scan plans.	
enable_mergejoin	bool	on	on				user	Enables the planner's use of merge join plans.	
enable_nestloop	bool	on	on				user	Enables the planner's use of nested-loop join plans.	The various enable_* settings are intended for interactive query debugging, and not as a way to control the overall server. If you need to turn any of these off in your .conf file, then something else is seriously wrong with your setup.

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
enable_seqscan	bool	on	on				user	Enables the planner's use of sequential-scan plans.	
enable_sort	bool	on	on				user	Enables the planner's use of explicit sort steps.	
enable_tidscan	bool	on	on				user	Enables the planner's use of TID scan plans.	
Other Planner Options									
constraint_exclusion	bool	off	varies				user	Enables the planner to use constraints to optimize queries. Child table scans will be skipped if their constraints guarantee that no rows match the query.	Turn this on if you expect to use table partitioning. Otherwise, leave it off.
default_statistics_target	integer	10	varies	1	1000		user	Sets the default statistics target. This applies to table columns that have not had a column-specific target set via ALTER TABLE SET STATISTICS.	Most applications can use the default of 10. If you have run into bad rowcount estimates in your application already, try increasing it to 100 or 200. Data warehousing applications generally need to use 500 to 1000.
from_collapse_limit	integer	8	default	1	INT_MAX		user	Sets the FROM-list size beyond which subqueries are not collapsed. The planner will merge subqueries into upper queries if the resulting FROM list would have no more than this many items.	While it's probably true that newer CPUs could support higher collapse_limits, there's not much incremental benefit to just raising either collapse_limit to 10 or 11.
join_collapse_limit	integer	8	default	1	INT_MAX		user	Sets the FROM-list size beyond which JOIN constructs are not flattened. The planner will flatten explicit JOIN constructs into lists of FROM items whenever a list of no more than this many items would result.	If for some reason you wanted to explicitly declare the join order for all of your queries, you could set this to 1. That is not recommended, though.
Genetic Query Optimizer									
geqo	bool	on	default				user	Enables genetic query optimization. This algorithm attempts to do planning without exhaustive searching.	
geqo_threshold	integer	12	14	2	INT_MAX		user	Sets the threshold of FROM items beyond which GEQO is used.	With new, faster processors it's tempting to raise the geqo_threshold a lot. However, it's still true that exhaustively planning queries with 22 or more tables will take the rest of your life.
geqo_effort	integer	5	default	1	10		user	GEQO: effort is used to set the default for other GEQO parameters.	

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
geqo_generations	integer	0	default	0	INT_MAX		user	GEQO: number of iterations of the algorithm. Zero selects a suitable default value.	
geqo_pool_size	integer	0	default	0	INT_MAX		user	GEQO: number of individuals in the population. Zero selects a suitable default value.	
geqo_selection_bias	real	2	default	1.5	2		user	GEQO: selective pressure within the population.	
Reporting and Logging									
Where to Log									
log_destination	string	csvlog	varies				sighup	Sets the destination for server log output. Valid values are combinations of "stderr", "syslog", "csvlog", and "eventlog", depending on the platform.	Your choice of log destination depends on your system administration plans and the status of your server. “syslog” or “eventlog” (Windows) are good choices for most development servers, because they can support centralized log monitors. For development and testing, however, “csvlog” is probably the most useful, as it allows you to run queries against the log contents.
logging_collector	bool	on	on				postmaster	Start a subprocess to capture stderr output and/or csvlogs into log files.	Only relevant for “csvlog” and “stderr”.
log_directory	string	pg_log	varies				sighup	Sets the destination directory for log files. Can be specified as relative to the data directory or as absolute path.	If you are having PostgreSQL keep its own activity logs on a production server, it's probably a good idea to locate them on separate storage from the database and transaction log.
log_filename	string	postgresql-%Y-%m-%d_%H%M%S.log	varies				sighup	Sets the file name pattern for log files.	If you want your logs to rotate automatically without needing a cron job to delete old logs, try naming them after the days of the week or the month so they overwrite automatically (i.e. 'postgresql-%a' or 'postgresql-%d'). This also helps with log analysis.
log_rotation_age	integer	1440	default	0	35791394	min	sighup	Automatic log file rotation will occur after N minutes.	
log_rotation_size	integer	10240	0	0	2097151	kB	sighup	Automatic log file rotation will occur after N kilobytes.	Generally a good idea to turn off (0) because it makes log segment names hard to predict, and thus hard to clean up or automatically import for analysis.
log_truncate_on_rotation	bool	off	on				sighup	Truncate existing log files of same name during log rotation.	
syslog_facility	string	LOCAL0	varies				sighup	Sets the syslog "facility" to be used when syslog enabled. Valid values are LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5,	For production servers, it's often wise from both and administration and security point of view to use a centralized logserver. If so, the facility name for that logserver interface would go here.

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
								LOCAL6, LOCAL7.	
syslog_ident	string	postgres	POSTGRES_$HOST				sighup	Sets the program name used to identify PostgreSQL messages in syslog.	If using a centralized logserver, you probably want to identify your postgresql instance by hostname.
What to Log									
debug_pretty_print	bool	off	on				user	Indents parse and plan tree displays.	
debug_print_parse	bool	off	default				user	Prints the parse tree to the server log.	The three debug_print_* settings are really for doing careful query debugging for an isolated test. Do not turn them on for a production server as they produce enormous amounts of log output.
debug_print_plan	bool	off	default				user	Prints the execution plan to server log.	
debug_print_rewritten	bool	off	default				user	Prints the parse tree after rewriting to server log.	
log_checkpoints	bool	off	varies				sighup	Logs each checkpoint.	When doing performance analysis, it's often a good idea to turn on most of the logging options and log them to a CSVlog.
log_connections	bool	off	varies				backend	Logs each successful connection.	Useful for performance analysis.
log_disconnections	bool	off	varies				backend	Logs end of a session, including duration.	Useful for performance analysis.
log_duration	bool	on	varies				superuser	Logs the duration of each completed SQL statement.	Useful for performance analysis.
log_hostname	bool	off	off				sighup	Logs the host name in the connection logs. By default, connection logs only show the IP address of the connecting host. If you want them to show the host name you can turn this on, but depending on your host name resolution setup it might impose a non-negligible performance penalty.	As this setting requires resolution of each connecting hostname, it's pretty much always too expensive to have on, even when troubleshooting.
log_line_prefix	string		default				sighup	Controls information prefixed to each log line. If blank, no prefix is used.	Primarily useful for providing extra information when logging to syslog or eventlog. Try "%h:%d:%u:%c %t" for this.
log_lock_waits	bool	off	varies				superuser	Logs long lock waits.	Useful for performance analysis.
log_statement	ENUM	none	varies				superuser	Sets the type of statements logged. Valid values are "none", "ddl", "mod", and "all".	For exhaustive performance analysis on test systems, set to 'all'. Most production setups will just want to use 'ddl' to make sure to record database-altering actions, but very secure setups may want to use 'mod' or even 'all'. Can produce a lot of log volume.

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
log_temp_files	memory	-1	varies	-1	INT_MAX	kB	user	Log the use of temporary files larger than this number of kilobytes. Zero logs all files. The default is -1 (turning this feature off).	This logger is used for troubleshooting sorts and other activities which are spilling to disk. If you use it at all, it's probably good to set it a something low like 1kB so that you know each query that spilled to disk, since any disk spill at all causes a dramatic slowdown in the query. Can be used to see if you need more work_mem, temp_mem or maintenance_work_mem.
log_timezone	string	per ENV	local timezone				sighup	Sets the time zone to use in log messages.	To avoid confusion, it's often useful to log to the timezone where the DBA or sysadmin lives.
When to Log									
client_min_messages	ENUM	notice	default				user	Sets the message levels that are sent to the client. Valid values are DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, LOG, NOTICE, WARNING, and ERROR. Each level includes all the levels that follow it. The later the level, the fewer messages are sent.	Unless doing interactive debugging, then you want it set to DEBUG1-5. If you have a client application which is confused by some of PostgreSQL's WARNINGs then you may want to set this to ERROR.
log_min_messages	ENUM	notice	default				superuser	Sets the message levels that are logged. Valid values are DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, INFO, NOTICE, WARNING, ERROR, LOG, FATAL, and PANIC. Each level includes all the levels that follow it.	Unless doing serious troubleshooting. If you want to output parses and plans, set to DEBUG1.
log_min_error_statement	ENUM	error	default				superuser	Causes all statements generating error at or above this level to be logged. All SQL statements that cause an error of the specified level or a higher level are logged.	Logs SQL statements which error. If you have an application which routinely generates errors and can't fix it, then raise the level to FATAL or PANIC.
log_error_verbosity	ENUM	default	default				superuser	Sets the verbosity of logged messages. Valid values are "terse", "default", and "verbose".	Unless doing intensive debugging. Alternately, set to TERSE if managing log volume is becoming a problem.
log_min_duration_statement	integer	-1	1min	-1	2147483	ms	superuser	Sets the minimum execution time above which statements will be logged. Zero prints all queries. -1 turns this feature off.	Possibly the most generally useful log setting for troubleshooting performance, especially on a production server. Records only long-running queries for analysis; since these are often your "problem" queries, these are the most useful ones to know about.
log_autovacuum_min_duration	integer	-1	1min (tune)	-1	2147483	ms	sighup	Sets the minimum execution time above which autovacuum actions will be logged. Zero prints all actions. -1 turns autovacuum logging off.	Logs all autovacuum actions which take more than the specified time. Useful for figuring out if autovacuum is bogging down your system or blocking.
silent_mode	bool	off	varies				postmaster	Runs the server silently. If this parameter is set, the server will automatically run in the background and any controlling terminals are dissociated.	Set to "on" if using syslog or eventlog. Otherwise, do not dare use.
Statistics									
Query and Index Statistics Collector									

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
track_activities	bool	on	default				superuser	Collects information about executing commands. Enables the collection of information on the currently executing command of each session, along with the time at which that command began execution.	These options have been renamed for 8.3 to underline the fact that they work differently than they used to. For one thing, their overhead is now very low, so there's absolutely no reason to turn any of the statistics collector options off.
track_counts	bool	on	default				superuser	Collects statistics on database activity.	Needed for autovacuum to work properly. Do not turn off.
update_process_title	bool	on	default				superuser	Updates the process title to show the active SQL command. Enables updating of the process title every time a new SQL command is received by the server.	Updates the process title on OSes which support this. Very useful for checking resource usage by currently running queries.
Monitoring									
log_executor_stats	bool	off	default				superuser	Writes executor performance statistics to the server log.	Used for profiling the query executor.
log_parser_stats	bool	off	default				superuser	Writes parser performance statistics to the server log.	Used for profiling the query parser.
log_planner_stats	bool	off	default				superuser	Writes planner performance statistics to the server log.	Used for profiling the query planner.
log_statement_stats	bool	off	default				superuser	Writes cumulative performance statistics to the server log.	Used for full query path profiling. Exclusive of the other three options.
Lock Management									
deadlock_timeout	integer	1000	5s	1	2147483	ms	sighup	Sets the time to wait on a lock before checking for deadlock.	Unless you believe that you need to troubleshoot deadlocks, this should be set to be longer than 80% of your transactions.
max_locks_per_transaction	integer	64	default	10	INT_MAX		postmaster	Sets the maximum number of locks per transaction. The shared lock table is sized on the assumption that at most max_locks_per_transaction * max_connections distinct objects will need to be locked at any one time.	Some databases with very complex schema or with many long-running tranactions need a higher amount. This is rare though.
Locale & Formatting									
Display									

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
DateStyle	list	ISO, MDY	default				user	Sets the display format for date and time values. Also controls interpretation of ambiguous date inputs.	Should be set according to the format in which you expect to receive date information.
TimeZone	string	per ENV	default				user	Sets the time zone for displaying and interpreting time stamps.	To avoid a lot of confusion, make sure this is set to your local timeszone. If the server covers multiple time zones, then this should be set on a ROLE or connection basis.
extra_float_digits	integer	0	default	-15	2		user	Sets the number of digits displayed for floating-point values. This affects real, double precision, and geometric data types. The parameter value is added to the standard number of digits (FLT_DIG or DBL_DIG as appropriate).	Only significant for applications which do a lot of float calculations, like scientific databases.
timezone_abbreviations	string	Default	default				user	Selects a file of time zone abbreviations.	See appendencies for alternatives.
Locale									
client_encoding	string	per ENV	default				user	Sets the client's character set encoding.	Should match server_encoding unless you have a really good reason why not.
lc_collate	string	as compiled	N/A				internal	Shows the collation order locale.	Set at initdb time. Displayed for information only.
lc_ctype	string	as compiled	N/A				internal	Shows the character classification and case conversion locale.	Set at initdb time. Displayed for information only.
lc_messages	string	as compiled	default				superuser	Sets the language in which messages are displayed.	
lc_monetary	string	as compiled	default				user	Sets the locale for formatting monetary amounts.	
lc_numeric	string	as compiled	default				user	Sets the locale for formatting numbers.	
lc_time	string	as compiled	default				user	Sets the locale for formatting date and time values.	
server_encoding	string	per ENV	N/A				internal	Sets the server (database) character set encoding.	Set at initdb time. Displayed for information only.
Other Settings & Defaults									
Default Locations									

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
search_path	list	"$user",public	varies				user	Sets the schema search order for names that are not schema-qualified.	Most DBAs either use the default or set search_path on a ROLE or database object basis. The one reason to set it in postgresql.conf is if you are taking the security step of removing the special "public" schema in order to lock down your database.
default_tablespace	string		default				user	Sets the default tablespace to create tables and indexes in. An empty string selects the database's default tablespace.	Change this if you want a different tablespace for user-created tables. Generally, better set on a ROLE or session basis.
temp_tablespaces	list		default				user	Sets the tablespace(s) to use for temporary tables and sort files.	For applications which create lots of temporary objects, this setting can be used to put the temp space on a faster/separate device, or even a ramdisk. Because it accepts a list, it can even be used to load balance temp object creation among several tablespaces.
Statement Behavior									
default_transaction_isolation	ENUM	read committed	default				user	Sets the transaction isolation level of each new transaction. Each SQL transaction has an isolation level, which can be either "read uncommitted", "read committed", "repeatable read", or "serializable".	This can be used to default the entire database operation to SERIALIZABLE, if you need all statements serialized to support some middleware application. Otherwise not recommended to change.
default_transaction_read_only	bool	off	default				user	Sets the default read-only status of new transactions.	This setting is mainly useful for preventing yourself from accidentally changing data. It is not really a security setting, as anyone can revoke it on their own session. Better set on a session or ROLE level.
statement_timeout	time	0	varies	0	INT_MAX	ms	user	Sets the maximum allowed duration of any statement. A value of 0 turns off the timeout.	Defaults to 0, meaning no timeout. For most web applications, it's a good idea to set a default timeout, such as 60s to prevent runaway queries from bogging the server. If set, though, you need to remember to set (at the ROLE or session level) a higher statement_timeout for expected long-running maintenance or batch operations.
Libraries									
dynamic_library_path	string	$libdir	default				superuser	Sets the path for dynamically loadable modules. If a dynamically loadable module needs to be opened and the specified name does not have a directory component (i.e., the name does not contain a slash), the system will search this path for the specified file.	Primarily useful if you've written lots of custom C libraries for your installation and want to organize them into custom directories.
local_preload_libraries	string		default				backend	Lists shared libraries to preload into each backend.	This is largely a convenience setting, automatically loading libraries listed without needing an explicit load command. Has not effect on performance.
Replication									
session_replication_role	ENUM	origin	default				superuser	Sets the session's behavior for triggers and rewrite rules. Each session can be either "origin", "replica", or "local".	Only gets changed for databases which are taking part in a replication chain. In that case, "origin" servers fire replication (and other) triggers, and "replica" do not. Part of the generic replication hooks which are used by Slony and Bucardo.
XML									

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
xmlbinary	ENUM	base64	varies				user	Sets how binary values are to be encoded in XML. Valid values are BASE64 and HEX.	Set to whatever your client application supports.
xmloption	ENUM	content	default				user	Sets whether XML data in implicit parsing and serialization operations is to be considered as documents or content fragments. Valid values are DOCUMENT and CONTENT.	
TSearch									
gin_fuzzy_search_limit	integer	0	varies	0	INT_MAX		user	Sets the maximum allowed result for exact search by GIN.	If you're going to use GIN queries in a web application, it's generally useful to set a limit on how many rows can be returned from the index just for response times. However, the maximum number needs to depend on your application; what do users see as an acceptable expression of "many"?
default_text_search_config	string	per ENV	default				user	Sets default text search configuration.	Set to the most common language used by the users.
Other Defaults									
check_function_bodies	bool	on	default				user	Check function bodies during CREATE FUNCTION.	You only really want to turn this off to resolve circular dependancies, and that can be done on a per-session basis. In general, checking for syntax errors in PL/pgSQL functions is a very good idea.
explain_pretty_print	bool	on	default				user	Uses the indented output format for EXPLAIN VERBOSE.	
Version and Platform Compatibility									
Previous PostgreSQL Versions									
add_missing_from	bool	off	default				user	Automatically adds missing table references to FROM clauses.	Do not turn on! This setting is a frequent cause of runaway cartesian join queries. Fix your application's queries instead.
array_nulls	bool	on	default				user	Enable input of NULL elements in arrays. When turned on, unquoted NULL in an array input value means a null value; otherwise it is taken literally.	Provided for compatibility with 7.4 behavior.
backslash_quote	ENUM	safe_encoding	default				user	Sets whether "\'" is allowed in string literals. Valid values are ON, OFF, and SAFE_ENCODING.	If you have cleaned up your application code, you can set this to 'off' to help lock down the database. Older PHP applications will require the insecure setting of 'on'.

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
default_with_oids	bool	off	default				user	Create new tables with OIDs by default.	Provided for consistency with 7.3 behavior. Since this creates an OID for every row, can cause OID wraparound in large databases.
escape_string_warning	bool	on	off				user	Warn about backslash escapes in ordinary string literals.	Useful for providing warnings for interpreted-language applications which may be engaging in unsafe string escape behavior. Unless you are currently porting or upgrading such an application, though, these warnings are not useful and should be turned off.
regex_flavor	ENUM	advanced	default				user	Sets the regular expression "flavor". This can be set to advanced, extended, or basic.	Provided for compatibility with 7.3, or with strict spec-compliant client applications, which will want 'extended' or 'basic'. Most people want "advanced" though.
sql_inheritance	bool	on	default				user	Causes subtables to be included by default in various commands.	Turn off for 7.1-compatible behavior.
standard_conforming_strings	bool	off	on				user	Causes '...' strings to treat backslashes literally.	If you can clean up your application code, this disables use of \ as an escape character except in escaped (E' ') strings. This is both safer, and less likely to result in unexpected output for things like Windows filepaths.
synchronize_seqscans	bool	on	on				user	Enable synchronized sequential scans.	This new peformance enhancment can also cause rows to be returned in an order other than physical storage order. For poorly-written older applications, this may break application code; turn it off to disable.
Other Platforms and Clients									
transform_null_equals	bool	off	off				user	Treats "expr=NULL" as "expr IS NULL". When turned on, expressions of the form expr = NULL (or NULL = expr) are treated as expr IS NULL, that is, they return true if expr evaluates to the null value, and false otherwise. The correct behavior of expr = NULL is to always return null (unknown).	Provided for compatibility with Microsoft Access and similar broken applications which treat "= NULL" as the same as "IS NULL".
Customized Options									
custom_variable_classes	string		default				sighup	Sets the list of known custom variable classes.	Available to support PLs (procedural languages) which require additional configuration variables, most notably R and Java. See the instructions for those PLs for settings.
Developer Options									
allow_system_table_mods	bool	off	default				postmaster	Allows modifications of the structure of system tables.	Only available in single-user mode; this setting is for initdb and may be used in the future for upgrade-in-place.

Annotated GUCS Version 8.3 -- Draft 1

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
debug_assertions	bool	off	default				user	Turns on various assertion checks. This is a debugging aid.	Used for debugging PostgreSQL code problems; not for production use. Requires compile options.
ignore_system_indexes	bool	off	default				backend	Disables reading from system indexes. It does not prevent updating the indexes, so it is safe to use. The worst consequence is slowness.	Useful for salvaging data from a corrupted database.
post_auth_delay	integer	0	default	0	INT_MAX	s	backend	Waits N seconds on connection startup after authentication. This allows attaching a debugger to the process.	Primarily used for attaching debuggers to sessions.
pre_auth_delay	integer	0	default	0	60	s	sighup	Waits N seconds on connection startup before authentication. This allows attaching a debugger to the process.	Primarily used for attaching debuggers to sessions.
trace_notify	bool	off	default				user	Generates debugging output for LISTEN and NOTIFY.	The various TRACE options are for debugging specific behaviors interactively. Many of them require compile-time options. trace_notice is for debugging listen/notice.
trace_sort	bool	off	default				user	Emit information about resource usage in sorting.	For debugging sorts.
trace_locks	bool	off	default				user	Emit information about resource usage in sorting.	For debugging row locks.
trace_lwlocks	bool	off	default				user	Emit information about resource usage in sorting.	For debugging low-level locks.
trace_userlocks	bool	off	default				user	Emit information about resource usage in sorting.	For debugging USERLOCKS
trace_locks_oidmin	bool	off	default				user	Emit information about resource usage in sorting.	For debugging OIDs
trace_locks_table	bool	off	default				user	Emit information about resource usage in sorting.	For debugging table locks
debug_deadlocks	bool	off	default				user	Emit information about resource usage in sorting.	For debugging deadlocks
log_btree_build_stats	bool	off	default				user	Emit information about resource usage in sorting.	For debugging BTree indexes.
wal_debug	bool	off	default				user	Emit information about resource usage in sorting.	For debugging the Write Ahead Log
zero_damaged_pages	bool	off	off				superuser	Continues processing past damaged page headers. Detection of a damaged page header normally causes PostgreSQL to report an error, aborting the current transaction. Setting zero_damaged_pages to true causes the system to instead report a warning, zero out the damaged page, and continue	Used for salvaging data from a known-bad database. You should always make a binary backup before using this option, and it should not be used while users are allowed to connect. After damaged pages are erased, other kinds of data intergrity errors may persist (like broken PKs and FKs). ZDP should generally be used to get your DB to a stage where the data can be dumped and loaded into a new database.

Setting	Type	Default	Recommended	Min	Max	Unit	Context	Docs	Comments
								processing. This behavior will destroy data, namely all the rows on the damaged page.	
Preset Options									
block_size	integer	8192	N/A	8192	8192		internal	Shows the size of a disk block.	All of the Preset options are strictly informational data about compiled-in settings. None of these are adjustable without recompiling the server, and often rebuilding the database as well. Mostly, these are supplied so that administration scripts and applications can behave appropriately.
integer_datetimes	bool	off	N/A				internal	Datetimes are integer based.	
max_function_args	integer	100	N/A	100	100		internal	Shows the maximum number of function arguments.	
max_identifier_length	integer	63	N/A	63	63		internal	Shows the maximum identifier length.	
max_index_keys	integer	32	N/A	32	32		internal	Shows the maximum number of index keys.	
server_version	string	8.3.0	N/A				internal	Shows the server version.	
server_version_num	integer	80300	N/A	80300	80300		internal	Shows the server version as an integer.	