

Automated user and access management in DBaaS environments

Who we are

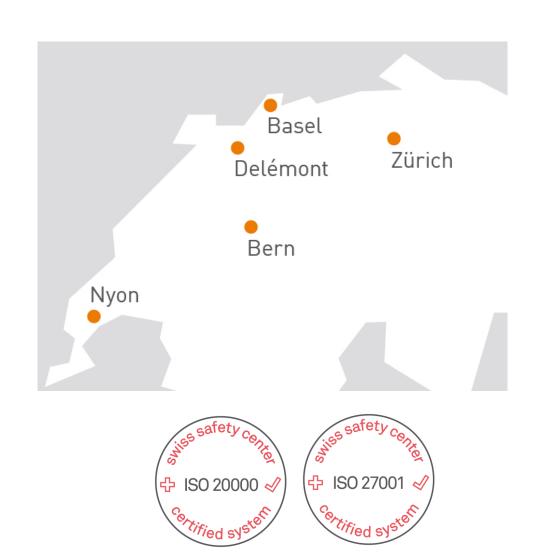


The Company

- > Founded in 2010
- > More than 90 specialists
- > Specialized in the Middleware Infrastructure
- > The invisible part of IT
- > Customers in Switzerland and all over Europe

Our Offer

- > Consulting
- > Service Level Agreements (SLA)
- > Trainings
- > License Management



About me



Karsten Lenz

Open Infrastructure Consultant

+41 78 799 0146

karsten.lenz[at]dbi-services.com

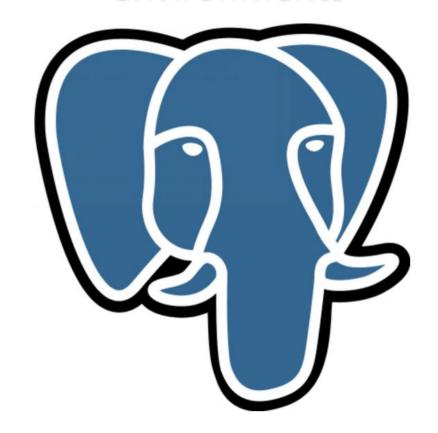


Agenda

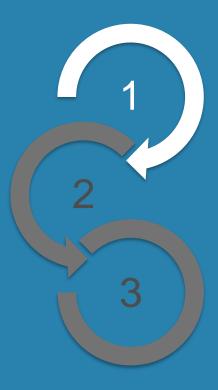


- > Automated user and access management in DBaaS environments
- > What is needed
- > How we implment

Automated user and access management in DBaaS environments







Automated user and access management in DBaaS environments



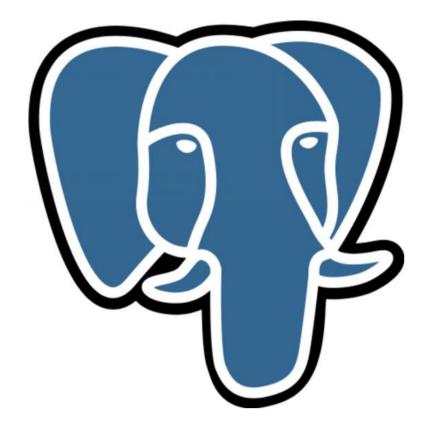
- > The task was how to implement a automated user and access mamanment.
- > Technical base are virtual machines using Linux (In this case SLES).
- > Central usable by Swiss Re developed Web Frontend.
- Use of PotsgreSQL functionality as much as possible.

PostgreSQL default functionality

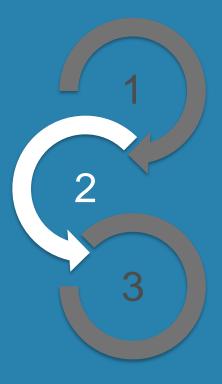


- > Using pg_hba_file_rules to read out existing configuration
- Table pg_hba has the structure of pg_hba_file_rules.
- > Data is written back using the structure of pg_hba_file_rules.
- > Functionality written in plpgsql.

Description

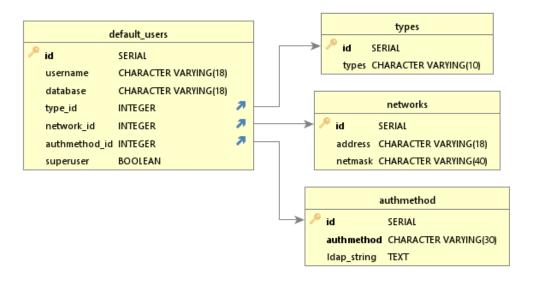






Datamodel





pg_hba line_number SERIAL TEXT TEXT database TEXT user_name TEXT address netmask TEXT auth_method TEXT options TEXT TEXT error

log
log_timestamp TIMESTAMP(6) WITHOUT TIME ZONE
error_text TEXT
error_message TEXT

Description



- > default users are defined users for administration.
- > types, allowed types are local, host, hostssl, hostnossl, checked by function.
- > networks, definition of allowed networks for connection, checked by function...
- > authmethod, scram-sha-256, ldap, peer etc. checked by function.
- > log, error logging.
- > pg_hba, table to copy pg_hba.conf using pg_hba_file_rules.

Functions



- > init_authent, initialization of default administration connectivity .
- > set_authent, setting authentication for user and database.
- > revoke_authent, revoking a defined access rule for a user and database.
- > clean_authent_database, revoking all access rules and users for one database.
- > clean_authent_user, revoking all access rules to all databases for one user.

All functions check if the authmethod, the type and the network are configured or not, if not user change or add will not be done!

pg_hba.conf and Table pg_hba



Second S	maintenance=#	select * ·	from procedure	s.pg hba ord	er by line numb	per:			
1							auth_method	options	error
2	 0	# TYPE	+ DATABASE	+ USER	+ IP-ADDRESS	IP-MASK	+ METHOD	-+ OPTIONS	· +
3	1	local	all	all	İ		peer	i i	i
A	2	host	all	all	127.0.0.1	255.255.255.255	scram-sha-256	İ	İ
S	3	host	all	all	0.0.0.0	0.0.0.0	trust	İ	İ
A	4	host	all	all	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff	scram-sha-256		İ
7	5	local	replication	all	İ		peer		İ
8 local replication repmgr	6	host	replication	all	127.0.0.1	255.255.255.255	scram-sha-256		1
9	7	host	replication	all	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff	scram-sha-256		
10	8	local	replication	repmgr			scram-sha-256	1	1
11 local	9	hostssl	replication	repmgr	127.0.0.1	255.255.255.255	scram-sha-256	1	1
12 hostssl repmgrdb repmgr 127.0.0.1 255.255.255 scram-sha-256		hostssl	replication	repmgr	192.168.0.0	255.255.0.0	scram-sha-256	1	
13 hostssl repmgrdb repmgr	11	local	repmgrdb	repmgr			scram-sha-256	1	1
14 local all postgres peer map=postgres-map	12	hostssl		repmgr	127.0.0.1	255.255.255.255	scram-sha-256	1	
To	13	hostssl	repmgrdb	repmgr	192.168.0.0	255.255.0.0	scram-sha-256		
maintenance=# \q [postgres@pgconf ~]\$ cat /pgdata/14/data/pg_hba.conf # TYPE DATABASE USER IP-ADDRESS IP-MASK METHOD OPTIONS local all all peer host all all 0.0.0.0 0.0.0 0.0.0 host all all ::1 fffffffffffffffffffffffffffffffff		local		postgres				map=postgres	-map
maintenance=# \q [postgres@pgconf ~]\$ cat /pgdata/14/data/pg_hba.conf # TYPE DATABASE USER IP-ADDRESS IP-MASK METHOD oPTIONS local all all 27.0.0.1 255.255.255 scram-sha-256 host all all ::1 ffff:ffff:ffff:ffff:ffff:ffff:fff	17	host	all	all	0.0.0.0	0.0.0.0	reject		
Docal Al			t /pgdata/14/d	ata/pg_hba.c	onf				
host all all 127.0.0.1 255.255.255.255 scram-sha-256 host all all sil sil scram-sha-256 local replication all 127.0.0.1 255.255.255.255 scram-sha-256 host replication all 127.0.0.1 255.255.255.255 scram-sha-256 host replication repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl replication repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl replication repmgr 192.168.0.0 255.255.255.0.0 scram-sha-256 hostssl repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 local all postgres host all 0.0.0.0 0.0.	# TYPE	DATABAS		I	P-ADDRESS	IP-MASK		METHOD	OPTIONS
host all all 0.0.0.0 0.0.0.0 trust host all all ::1 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff	local							peer	
host all all ::1 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff	host			1	27.0.0.1	255.255.255		scram-sha-256	
local replication all 127.0.0.1 255.255.255.255 scram-sha-256 host replication all ::1 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff				0	.0.0.0			trust	
host replication all 127.0.0.1 255.255.255.255 scram-sha-256 host replication all ::1 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff					:1	ffff:ffff:ffff:ffff:ffff:fff	f:ffff	scram-sha-256	
host replication all ::1 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff									
local replication repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl replication repmgr 192.168.0.0 255.255.255.255 scram-sha-256 local repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl repmgrdb repmgr 192.168.0.0 255.255.0.0 scram-sha-256 local all postgres peer map=postgres-map host all all 0.0.0.0 0.0.0.0 reject									
hostssl replication repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl replication repmgr 192.168.0.0 255.255.0.0 scram-sha-256 local repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl repmgrdb repmgr 192.168.0.0 255.255.255.255 scram-sha-256 local all postgres peer map=postgres-map host all all 0.0.0.0 0.0.0.0 reject					:1	ffff:ffff:ffff:ffff:ffff:fff			
hostssl replication repmgr 192.168.0.0 255.255.0.0 scram-sha-256 local repmgrdb repmgr 127.0.0.1 255.255.255.255 scram-sha-256 hostssl repmgrdb repmgr 192.168.0.0 255.255.255.255 scram-sha-256 local all postgres peer map=postgres-map host all 0.0.0.0 0.0.0.0 reject									
localrepmgrdbrepmgrhostsslrepmgrdbrepmgr127.0.0.1255.255.255.255scram-sha-256hostsslrepmgrdbrepmgr192.168.0.0255.255.0.0scram-sha-256localallpostgrespeermap=postgres-maphostall0.0.0.00.0.0.0reject									
hostsslrepmgrdbrepmgr127.0.0.1255.255.255.255scram-sha-256hostsslrepmgrdbrepmgr192.168.0.0255.255.0.0scram-sha-256localallpostgrespeermap=postgres-maphostall0.0.0.00.0.0.0reject					92.168.0.0	255.255.0.0			
hostssl repmgrdb repmgr 192.168.0.0 255.255.0.0 scram-sha-256 local all postgres peer map=postgres-map host all all 0.0.0.0 0.0.0.0 reject									
local all postgres peer map=postgres-map host all all 0.0.0.0 0.0.0.0 reject									
host all all 0.0.0.0 0.0.0.0 reject					92.168.0.0	255.255.0.0			
_									map=postgres-map
[postgres@pgconf ~]\$		_	all	0	.0.0.0	0.0.0.0		reject	

Coding in plpsql



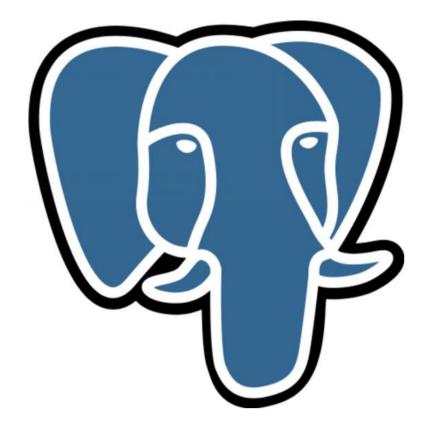
```
1 FREATE OR REPLACE FUNCTION "procedures"."init_authent" () RETURNS integer 2 VOLATILE
3 SECURITY DEFINER
4 AS $body$
5 -- by Karsten Lenz dbi services SA 2021.03.04
5 declare
         -- define variables --
         _type varchar(18);
         _type_id int;
         _address_id int;
         _authmethod_id int;
         _database varchar(50);
         _user_name varchar(18);
         _auth_method varchar(25);
          _address varchar(18);
         _netmask varchar(18);
          _ldap_string text;
          _log_timestamp timestamp := now();
          _error_text text;
         _log_table varchar(25) = 'procedures.log';
          _error_message text;
          _user_exists text;
         _pg_hba varchar(50) := setting from pg_settings where name like '%hba%';
         _check int = 0;
         _return int = 0;
          _count int = 0;
         _id int = 0;
         _position int;
         _reject_id int;
         _max_line int;
         _exist text;
          _superuser boolean;
7 BEGIN
         --lock table in exclusive lock
          _query = 'lock table procedures.pg_hba IN ACCESS EXCLUSIVE MODE';
         raise notice 'lock query : %',_query;
                 execute _query;
                          exception when others then
                                --log timestamp on error
                                 _log_timestamp := now();
                                 --error message
                                 _error_text = ''||_query||' failed. Errorcode = ' || SQLSTATE || ' Message = ' || SQLERRM || '';
                                 --raise warnina
                                 raise warning 'ERROR: %', _error_text;
                                 --error message
                                 _error_message = '' || SQLSTATE || ' Message = ' || SQLERRM || '';
                                 --build log query
                                 _query = 'insert into '||log_table||' (log_timestamp, error_text, error_message) values ('''||log_timestamp||''','''||_error_text||''','''||_error_message||''');';
                                 --raise query
                                 raise notice 'insert log query : %',_query;
                                 --execute query
                                 execute _query;
                                 return 1; -- lock table faild
                         end;
         end;
          --truncate table for reload
         _query = 'truncate table procedures.pg_hba;';
         raise notice 'truncate table pg_hba : %',_query;
         begin
                 execute _query;
                          exception when others then
                          begin
                                 --log timestamp on error
                                 _log_timestamp := now();
                                 --error message
                                 _error_text = ''||_query||' failed. Errorcode = ' || SQLSTATE || ' Message = ' || SQLERRM || '';
```

Error handling

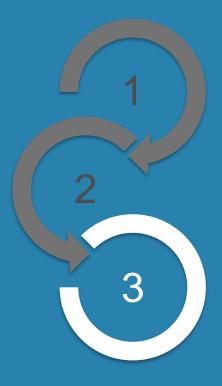


- > Each error is logged in log table.
- > Return codes are given back to web frontend for error handling.
- > Table pg_hba is exclusive locked that only one change can be done at time.

Demo







Creating user kle for all databases using ident = superuser



```
maintenance=# select procedures.set authent ('hostssl', 'all', 'kle', '192.168.198.0', 'ident', true);
NOTICE: type
                  : hostssl
NOTICE: database : all
NOTICE: user
                  : kle
NOTICE: network : 192.168.198.0
NOTICE: method
                  : ident
NOTICE: address : 192.168.198.0
NOTICE: netmask : 255.255.255.255
NOTICE: path pg_hba.conf : /pgdata/14/data/pg_hba.conf
NOTICE: lock query : lock table procedures.pg hba IN ACCESS EXCLUSIVE MODE
NOTICE: truncate query : truncate table procedures.pg_hba;
NOTICE: reload query : insert into procedures.pg hba (TYPE, DATABASE, user name, address, netmask, auth method, OPTIONS, error) (SELECT type, database, user name, coalesce(address,''), coa
lesce(netmask,''), auth_method, coalesce(options,''), error FROM procedures.pg_hba_file_read) order by line_number;
NOTICE: position of reject : 15
NOTICE: add data line query : insert into procedures.pg hba (line number, type, database, user name, address, netmask, auth method, options) VALUES (0, '# TYPE', 'DATABASE', 'USER', 'IP-ADD
RESS', 'IP-MASK', 'METHOD', 'OPTIONS');
NOTICE: add data line query : insert into procedures.pg hba (type, database, user_name, address, netmask, auth_method, options) values ('hostssl','all','kle','192.168.198.0','255.255.255.25
5','ident','');
NOTICE: add data line query : insert into procedures.pg_hba (type, database, user_name, address, netmask, auth_method) VALUES ('host', 'all', 'all', '0.0.0.0', '0.0.0.0', 'reject');
NOTICE: write pg hba.conf query : copy (SELECT FORMAT('%-8s',type), FORMAT('%-15s',database), FORMAT('%-15s',user name), FORMAT('%-20s',address), FORMAT('%-40s',netmask), FORMAT('%-15s',aut
h method), options from procedures.pg hba order by line number) to '/pgdata/14/data/pg hba.conf' WITH (NULL'');
NOTICE: user exits: 0
NOTICE: create user query : create user kle;
NOTICE: alter user query : alter user kle with superuser;
 set authent
(1 row)
maintenance=#
```

Again pg_hba.conf and table pg_hba with user kle



ne_number	type	database	user_name	address	netmask	auth_method	option	ıs	error
0	+ # TYPE	DATABASE	USER	IP-ADDRESS	IP-MASK	METHOD	OPTIONS		
1	local	all	all			peer	İ	ĺ	
2	host	all	all	127.0.0.1	255.255.255.255	scram-sha-256	5	ĺ	
3	host	all	all	0.0.0.0	0.0.0.0	trust		ĺ	
4	host	all	all	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff	scram-sha-256	5		
5	local	replication	all			peer			
6	host	replication	all	127.0.0.1	255.255.255	scram-sha-256	5		
7	host	replication	all	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff	scram-sha-256	5		
8	local	replication	repmgr			scram-sha-256	5		
9	hostssl	replication	repmgr	127.0.0.1	255.255.255	scram-sha-256	5		
10	hostssl	replication	repmgr	192.168.0.0	255.255.0.0	scram-sha-256	5		
11	local	repmgrdb	repmgr			scram-sha-256	5		
12	hostssl	repmgrdb	repmgr	127.0.0.1	255.255.255	scram-sha-256	5		
13	hostssl	repmgrdb	repmgr	192.168.0.0	255.255.0.0	scram-sha-256	5		
14	local	all	postgres			peer	map=postgr	es-map	
16	hostssl	all	kle	192.168.198.0	255.255.255	ident		- 1	
17	host	all	all	0.0.0.0	0.0.0.0	reject		- 1	
rows)						, 5	•	,	
rows) tenance=# tgres@pgc	\q onf ~]\$ ca [.]	t /pgdata/14/da F USER		onf			THOD	OPTIONS	
rows) tenance=# tgres@pgc PE	\q onf ~]\$ ca [.] DATABASI	E USER			IP-MASK	ME	THOD	OPTIONS	
rows) tenance=# tgres@pgco PE 1	\q onf ~]\$ ca DATABAS all	E USER all	I	onf P-ADDRESS	IP-MASK	ME pe	eer	OPTIONS	
rows) tenance=# tgres@pgco PE 1	\q onf ~]\$ ca [.] DATABASI	E USER	II	onf	IP-MASK 255.255.255	ME pe so		OPTIONS	
rows) tenance=# tgres@pgco PE 1	√q onf ~]\$ ca DATABASI all all all	E USER all all all	11 0	onf P-ADDRESS 27.0.0.1 .0.0.0	IP-MASK 255.255.255.255 0.0.0.0	ME pe so tr	eer cram-sha-256 rust	OPTIONS	
rows) tenance=# tgres@pgco PE 1	√q onf ~]\$ ca DATABASI all all	E USER all all all all all	11 0	onf P-ADDRESS 27.0.0.1	IP-MASK 255.255.255	ME pe so tr ffff so	eer cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1	\q onf ~]\$ ca DATABASI all all all all	E USER all all all all tion all	17 0 :	onf P-ADDRESS 27.0.0.1 .0.0.0	IP-MASK 255.255.255.255 0.0.0.0	ME pe so tr ffff so pe	eer cram-sha-256 cust cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1	\q onf ~]\$ ca DATABASI all all all all replica	E USER all all all all tion all tion all	17 9 :	onf P-ADDRESS 27.0.0.1 .0.0.0	IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f	ME pe so tr ffff so pe so	eer cram-sha-256 rust cram-sha-256 eer	OPTIONS	
rows) tenance=# tgres@pgco PE 1	\q onf ~]\$ ca DATABASI all all all all replica replica	E USER all all all tion all tion all	17 0 : 12	onf P-ADDRESS 27.0.0.1 .0.0.0 :1	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so tr ffff so pe so ffff so	eer cram-sha-256 cust cram-sha-256 eer cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1	\q onf ~]\$ ca DATABASI all all all replica replica replica	E USER all all all tion all tion all tion all tion all	17 0 : 12 :	onf P-ADDRESS 27.0.0.1 .0.0.0 :1	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so tr ffff so so so so so so	eer cram-sha-256 cram-sha-256 eer cram-sha-256 cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1 1 1 1	`\q onf ~]\$ ca DATABASI all all all replica replica replica replica	E USER all all all tion all tion all tion all tion repmgnution repmgnution	17 0 12 12 13	onf P-ADDRESS 27.0.0.1 .0.0.0 :1 27.0.0.1 :1	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so trefff so so so so so so	eer cram-sha-256 cust cram-sha-256 eer cram-sha-256 cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1 1 ssl ssl	\q onf ~]\$ ca DATABASI all all all replica replica replica replica replica	E USER all all all tion all tion all tion all tion repmgn tion repmgn tion repmgn	17 0 12 12 13 14 15 17	onf P-ADDRESS 27.0.0.1 .0.0.0 :1 27.0.0.1 :1	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so trefff so so so so so so so so so so so so so	eer cram-sha-256 cram-sha-256 eer cram-sha-256 cram-sha-256 cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1 1 ssl ssl ssl	\q onf ~]\$ ca DATABASI all all all replica replica replica replica replica replica	E USER all all all tion all tion all tion repmgn tion repmgn tion repmgn tion repmgn	17 07 17 17 19 19	onf P-ADDRESS 27.0.0.1 .0.0.0 :1 27.0.0.1 :1 27.0.0.1 92.168.0.0	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so tr pe so so so so so so so so so so so so so	eer cram-sha-256 cram-sha-256 eer cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1 1 ssl ssl ssl	\q onf ~]\$ ca' DATABASI all all all replica' replica' replica' replica' replica' replica' replica' replica'	E USER all all all tion all tion all tion repmged tion repmged tion repmged tion repmged tion repmged tion repmged tion repmged tion repmged tion repmged tion repmged tion repmged tion repmged	17 0 17 17 18 19 19 19	onf P-ADDRESS 27.0.0.1 .0.0.0 :1 27.0.0.1 :1 27.0.0.1 92.168.0.0	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so tr pe so so so so so so so so so so so so so	eer cram-sha-256 cram-sha-256 eer cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256	OPTIONS	
rows) tenance=# tgres@pgco PE 1 1 ssl ssl ssl ssl ssl	\q onf ~]\$ ca' DATABASI all all all replica' replica' replica' replica' replica' replica' replica' replica' replica' replica' reprodica' repmgrdi repmgrdi all	E USER all all all tion all tion all tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge	12 0 12 13 14 15 15 16 17 17 18 19 19	onf P-ADDRESS 27.0.0.1 .0.0.0 :1 27.0.0.1 :1 27.0.0.1 92.168.0.0	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so so so so so so so so so so so so so	eer cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256	OPTIONS	
rows) ntenance=#	\q onf ~]\$ ca' DATABASI all all all replica' replica' replica' replica' replica' replica' replica' replica' replica' reprodica'	E USER all all all tion all tion all tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge tion repmge	12 0 12 13 14 15 15 16 17 17 18 19 19	onf P-ADDRESS 27.0.0.1 .0.0.0 :1 27.0.0.1 :1 27.0.0.1 92.168.0.0	<pre>IP-MASK 255.255.255.255 0.0.0.0 ffff:ffff:ffff:ffff:ffff:ffff:f</pre>	ME pe so so so so so pe	eer cram-sha-256 cram-sha-256 cer cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256 cram-sha-256		

Cleaning user kle for all databases using ident = superuser



```
maintenance=# select procedures.clean_authent('kle', true);
NOTICE: user
                  : kle
NOTICE: complete : t
NOTICE: path pg_hba.conf : /pgdata/14/data/pg_hba.conf
NOTICE: lock query : lock table procedures.pg_hba IN ACCESS EXCLUSIVE MODE
NOTICE: reload query : insert into procedures.pg hba (TYPE, DATABASE, user name, address, netmask, auth method, OPTIONS, error) (SELECT type, database, user name, coalesce(address,''), coa
lesce(netmask,''), auth_method, coalesce(options,''), error FROM procedures.pg_hba_file_read) order by line_number;
NOTICE: reject id : 16
NOTICE: delete reject line query : delete from procedures.pg_hba where line_number = 16;
NOTICE: id of user record : 15
NOTICE: add data line query : insert into procedures.pg hba (type, database, user name, address, netmask, auth method, options) VALUES ('host', 'all', 'all', '0.0.0.0', '0.0.0.0', 'reject'
NOTICE: write pg_hba.conf query : copy (SELECT FORMAT('%-8s',type), FORMAT('%-15s',database), FORMAT('%-15s',user_name), FORMAT('%-20s',address), FORMAT('%-40s',netmask), FORMAT('%-15s',aut
h_method), options from procedures.pg_hba order by line_number) to '/pgdata/14/data/pg_hba.conf' WITH (NULL'');
NOTICE: drop user query : drop user kle;
clean_authent
(1 row)
maintenance=#
```

Again pg_hba.conf and table pg_hba without user kle



maintenance=#	select *	from procedure	s.pg hba oro	ler by line_numb	per:			
line_number	type	database	user_name		netmask	auth_method	options	error
0	# TYPE	DATABASE	USER	IP-ADDRESS	IP-MASK	METHOD	OPTIONS	
1	local	all	all			peer		
2	host	all	all	127.0.0.1	255.255.255.255	scram-sha-256		
3	host	all	all	0.0.0.0	0.0.0.0	trust	1	
4	host	all	all	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff	scram-sha-256	1	
5	local	replication	all			peer	1	
6	host	replication	all	127.0.0.1	255.255.255.255	scram-sha-256	1	
7	host	replication	all	::1	ffff:ffff:ffff:ffff:ffff:ffff:ffff	scram-sha-256		
8	local	replication	repmgr			scram-sha-256	1	
9	hostssl	replication	repmgr	127.0.0.1	255.255.255.255	scram-sha-256		
10	hostssl	replication	repmgr	192.168.0.0	255.255.0.0	scram-sha-256		
11	local	repmgrdb	repmgr			scram-sha-256		
12	hostssl	repmgrdb	repmgr	127.0.0.1	255.255.255.255	scram-sha-256		
13	hostssl	repmgrdb	repmgr	192.168.0.0	255.255.0.0	scram-sha-256		
14	local	all	postgres			peer	map=postgres-	map
17 16 rows)	host	all	all	0.0.0.0	0.0.0.0	reject		
aintenance=# postgres@pgco		t /pgdata/14/d	ata/pg_hba.c	conf				
TYPE	DATABASI			P-ADDRESS	IP-MASK		METHOD	OPTIONS
ocal	all	all					peer	
ost	all	all	1	27.0.0.1	255.255.255.255		scram-sha-256	
st	all	all	6	0.0.0.0	0.0.0.0		trust	
ost	all	all	:	:1	ffff:ffff:ffff:ffff:ffff:fff	f:ffff	scram-sha-256	
ocal	replica [.]	tion all					peer	
ost	replica [.]	tion all	1	27.0.0.1	255.255.255		scram-sha-256	
ost	replica [.]	tion all	:	:1	ffff:ffff:ffff:ffff:ffff:fff	f:ffff	scram-sha-256	
ocal	replica [.]	tion repmg	r				scram-sha-256	
ostssl	replica [.]	tion repmg	r 1	27.0.0.1	255.255.255		scram-sha-256	
ostssl	replica [.]	tion repmg	r 1	92.168.0.0	255.255.0.0		scram-sha-256	
ocal	repmgrdl		r				scram-sha-256	
ostssl	repmgrdl			27.0.0.1	255.255.255.255		scram-sha-256	
ostssl	repmgrdl	b repmg	r 1	.92.168.0.0	255.255.0.0		scram-sha-256	
ocal	all	postg					peer	map=postgres-map
ost	all _	all	6	0.0.0.0	0.0.0.0		reject	
postgres@pgcc	onf ~]\$							

Some publications from my side



- > https://blog.dbi-services.com/recurring-postgresql-installations-using-rhel-8-and-clones/
- https://www.heise.de/ratgeber/PostgreSQL-installieren-mit-den-Community-Paketen-4877556.html

Question from my side



We would like to give the coding for the base installation (shell scripts) and the functionally we have presented here back to the community if there is any interest.



Any questions?

Please do ask!



We would love to boost your IT-Infrastructure

How about you?