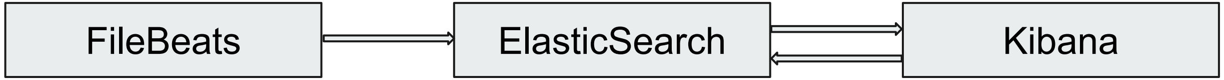
SRE Final Project Presentation

Team Unreliables
Members: Adya, Aman, Deepa, Divya, Utsav, Vinaya

Monitoring

Monitoring

Set up the following up Elastic Stack:



Created Dedicated VM for monitoring, so as not to overload the main server

- ElasticSearch - Monitoring VM
- Kibana - Monitoring VM http://34.82.254.169:5601/
- Filebeats - Main VM and Postgres_Standby VM

ElasticSearch

In the **Monitoring VM**, installed and configured ElasticSearch

- Memory Profiling: By default, ElasticSearch uses 1GB heap memory. Set it to use 512M at max
  config/jvm.options -> -Xmx512m -Xms512m
  To prevent overloading of RAM
- Configured ElasticSearch to allow other VMs to access it by changing it's binding IP address.
- ElasticSearch came with an option to daemonise the process, therefore started it in the background.

Kibana

In the **Monitoring VM**, installed and configured Kibana

- Set environment variable NODE_OPTIONS = “max-old-space-size=400”
(due to issue that by conjecture, was being caused by Kibana’s optimise/compile processes taking up more memory than the default heaph that Kibana specifies, and the low memory setting with which the VM was created. Therefore set a cap on it.)
- Configured Kibana to allow other VMs to access it by changing its binding IP address.
- Specified the ElasticSearch host: localhost
- Kibana was started as nohup process to allow it to run in the background.

Filebeats

In the **Main VM** (that hosts Nginx, Web App and Postgres-main server):

- Configured:
  1) ElasticSearch url
  2) Kibana url

- Enabled: Postgres, Nginx in Filebeat, to enable logging for both
  ~/modules.d/ -> ./filebeat modules enable nginx postgres

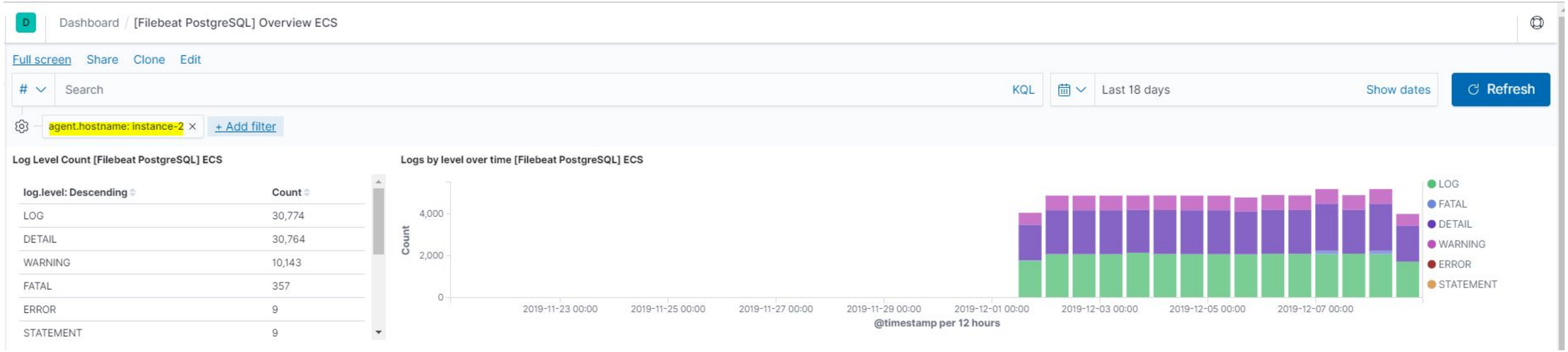
In the **Postgres_Standby VM** (that hosts the standby Postgres): similar configuration and enabled Postgres logging.

Filebeat doesn’t have default option to daemonize. Therefore, wrote python script to:

1. Get the pid of the currently running/previously ran Filebeat process
2. Check if the pid is still in running state
3. If not, launch filebeat
4. And store the newly launched process’ pid (referred to in step 1)

Crontab the above script to run once every hour.

POSTGRES: *How the system differentiates between log from Main VM and Postgres_Standby VM*

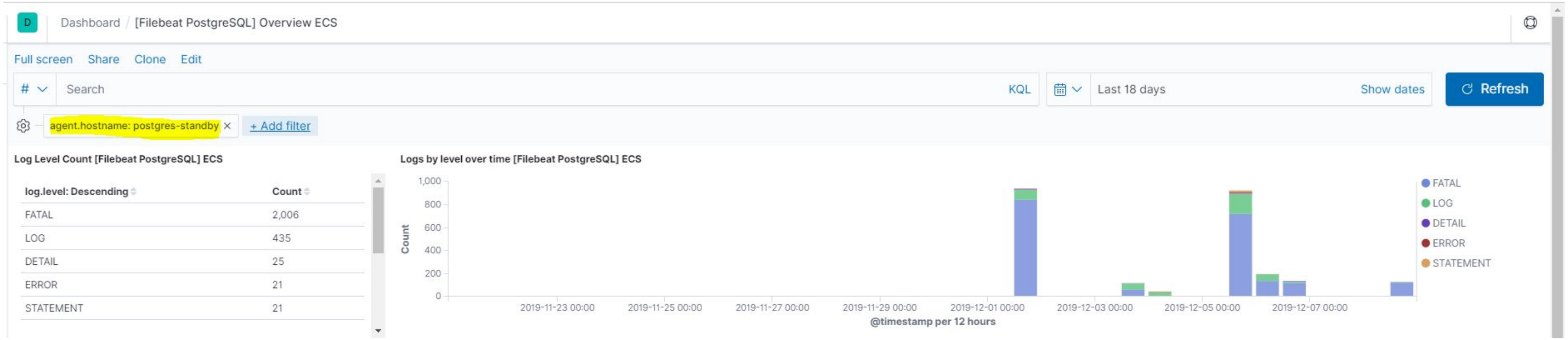


agent.hostname: instance-2

Log Level Count [Filebeat PostgreSQL] ECS

log.level: Descending	Count
LOG	30,774
DETAIL	30,764
WARNING	10,143
FATAL	357
ERROR	9
STATEMENT	9

Logs by level over time [Filebeat PostgreSQL] ECS



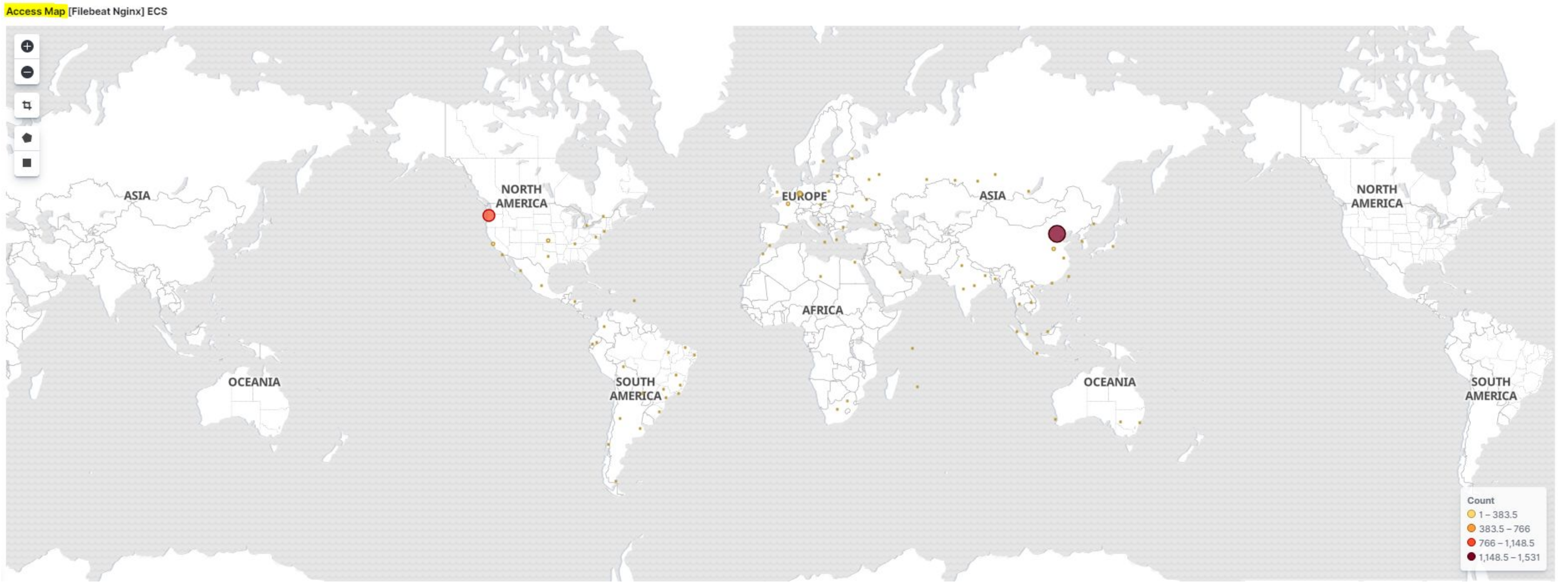
agent.hostname: postgres-standby

Log Level Count [Filebeat PostgreSQL] ECS

log.level: Descending	Count
FATAL	2,006
LOG	435
DETAIL	25
ERROR	21
STATEMENT	21

Logs by level over time [Filebeat PostgreSQL] ECS

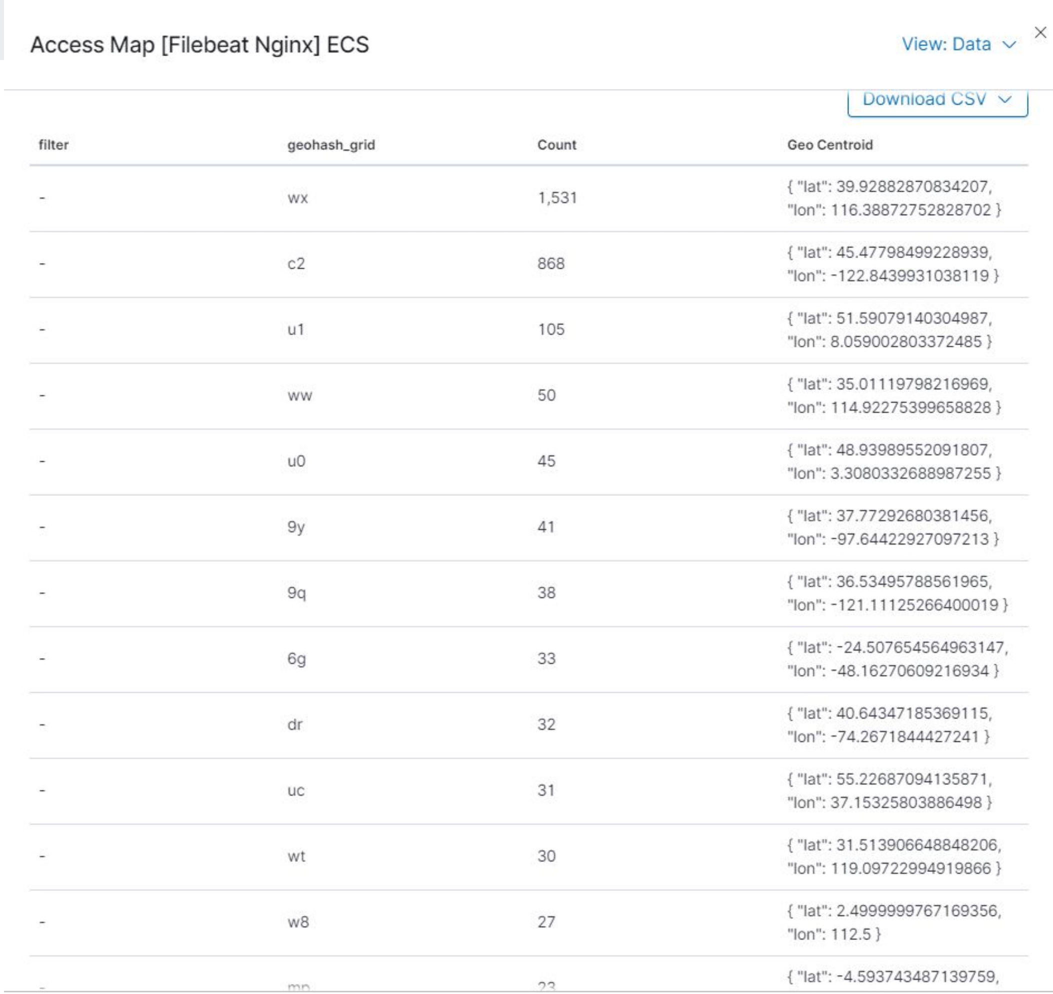
Nginx



Access Map [Filebeat Nginx] ECS

In numbers:

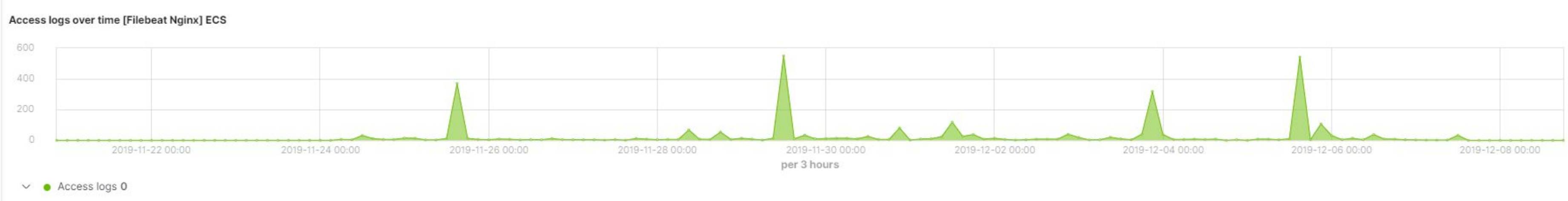
Total requests: 3199
Max Requests: Beijing, China



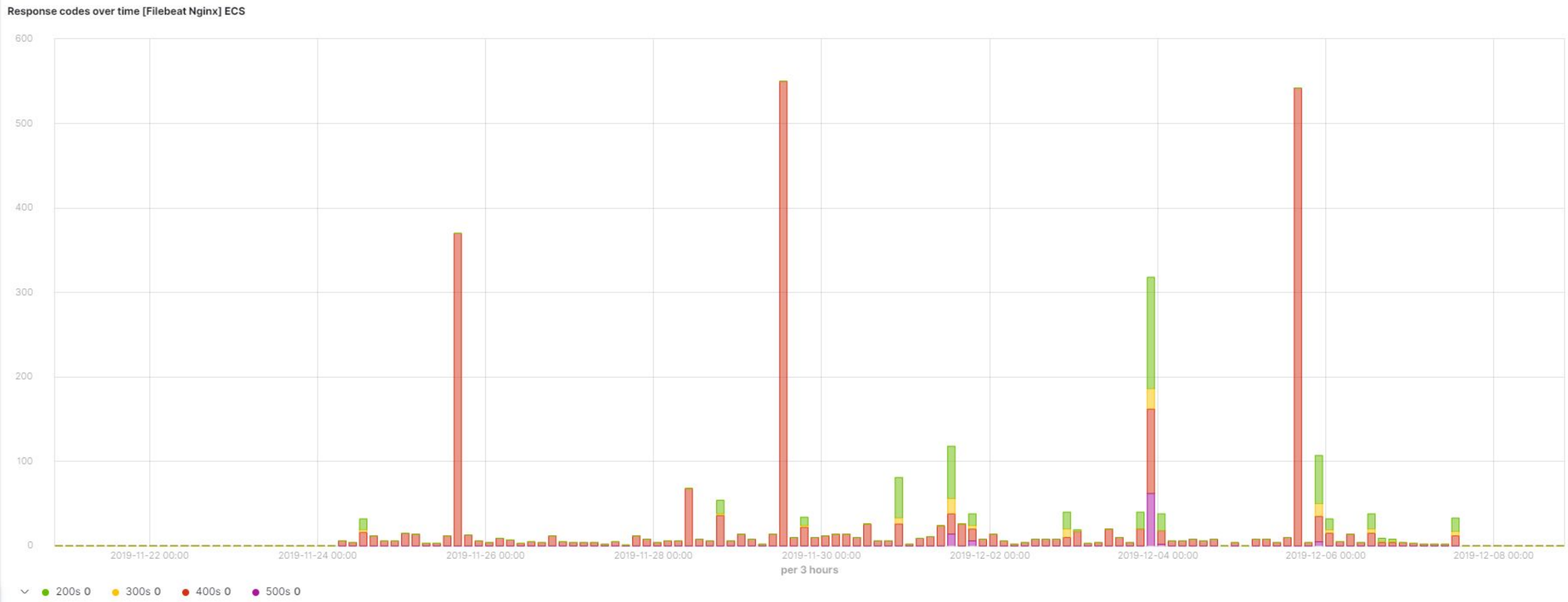
Access Map [Filebeat Nginx] ECS

filter	geohash_grid	Count	Geo Centroid
-	wx	1,531	{ "lat": 39.92882870834207, "lon": 116.38872752828702 }
-	c2	868	{ "lat": 45.47798499228939, "lon": -122.8439931038119 }
-	u1	105	{ "lat": 51.59079140304987, "lon": 8.059002803372485 }
-	ww	50	{ "lat": 35.01119798216969, "lon": 114.92275399658828 }
-	u0	45	{ "lat": 48.93989552091807, "lon": 3.3080332688987255 }
-	9y	41	{ "lat": 37.77292680381456, "lon": -97.64422927097213 }
-	9q	38	{ "lat": 36.53495788561965, "lon": -121.11125266400019 }
-	6g	33	{ "lat": -24.507654564963147, "lon": -48.16270609216934 }
-	dr	32	{ "lat": 40.64347185369115, "lon": -74.2671844427241 }
-	uc	31	{ "lat": 55.22687094135871, "lon": 37.15325803886498 }
-	wt	30	{ "lat": 31.513906648848206, "lon": 119.09722994919866 }
-	w8	27	{ "lat": 2.4999999767169356, "lon": 112.5 }
-	mn	23	{ "lat": -4.593743487139759,

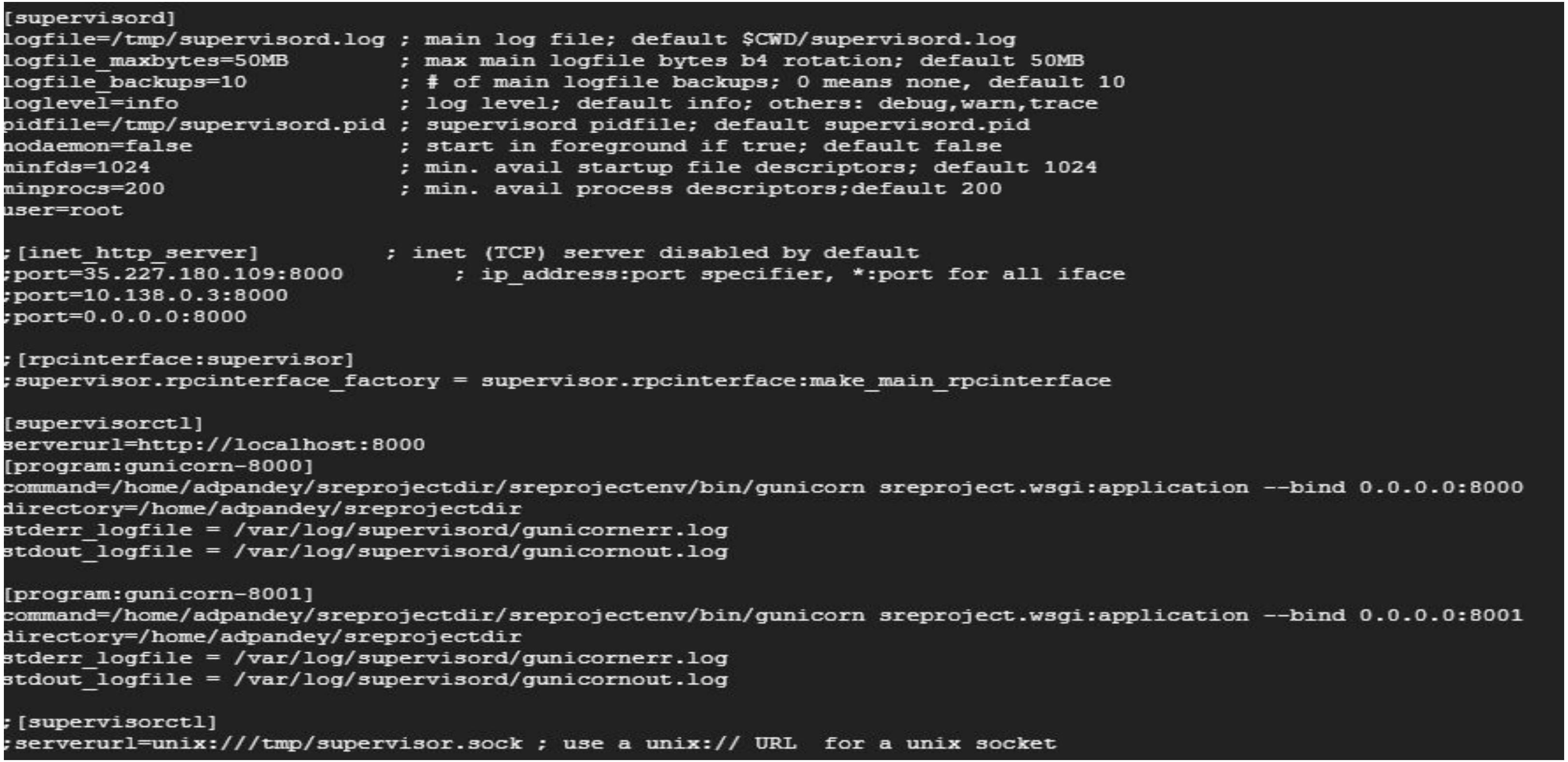
Access logs over time



Stats based on Response status codes:



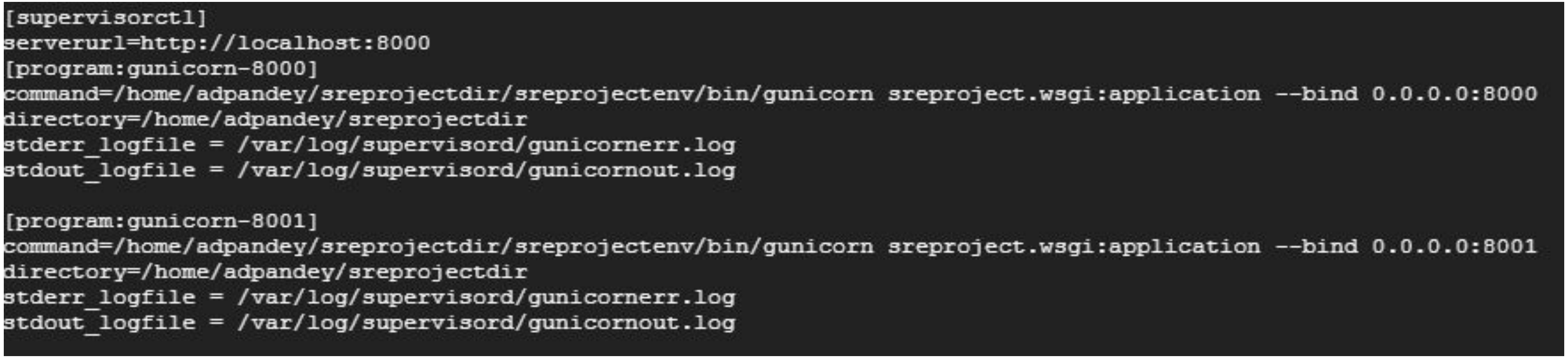
Gunicorn Redundancy



```
[supervisord]
logfile=/tmp/supervisord.log ; main log file; default $CWD/supervisord.log
logfile_maxbytes=50MB        ; max main logfile bytes b4 rotation; default 50MB
logfile_backups=10           ; # of main logfile backups; 0 means none, default 10
loglevel=info                ; log level; default info; others: debug,warn,trace
pidfile=/tmp/supervisord.pid ; supervisord pidfile; default supervisord.pid
nodaemon=false               ; start in foreground if true; default false
minfds=1024                  ; min. avail startup file descriptors; default 1024
minprocs=200                 ; min. avail process descriptors;default 200
user=root

;[inet_http_server]         ; inet (TCP) server disabled by default
;port=35.227.180.109:8000        ; ip_address:port specifier, *:port for all iface
;port=10.138.0.3:8000
;port=0.0.0.0:8000

;[rpcinterface:supervisor]
;supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface

[supervisorctl]
serverurl=http://localhost:8000
[program:gunicorn-8000]
command=/home/adpandey/sreprojectdir/sreprojectenv/bin/gunicorn sreproject.wsgi:application --bind 0.0.0.0:8000
directory=/home/adpandey/sreprojectdir
stderr_logfile = /var/log/supervisord/gunicornerr.log
stdout_logfile = /var/log/supervisord/gunicornout.log

[program:gunicorn-8001]
command=/home/adpandey/sreprojectdir/sreprojectenv/bin/gunicorn sreproject.wsgi:application --bind 0.0.0.0:8001
directory=/home/adpandey/sreprojectdir
stderr_logfile = /var/log/supervisord/gunicornerr.log
stdout_logfile = /var/log/supervisord/gunicornout.log

;[supervisorctl]
;serverurl=unix:///tmp/supervisor.sock ; use a unix:// URL  for a unix socket
```

Gunicorn configuration

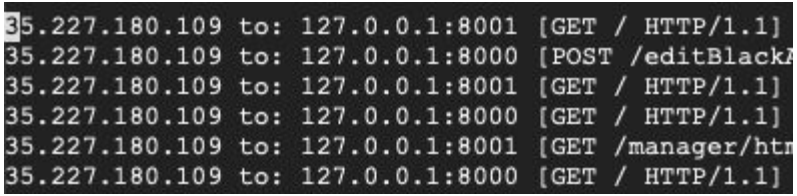
- Making some changes to the existing supervisord configuration
- Running gunicorn as the application server on ports 8000 and 8001
- Checked the logs to see these ports were up



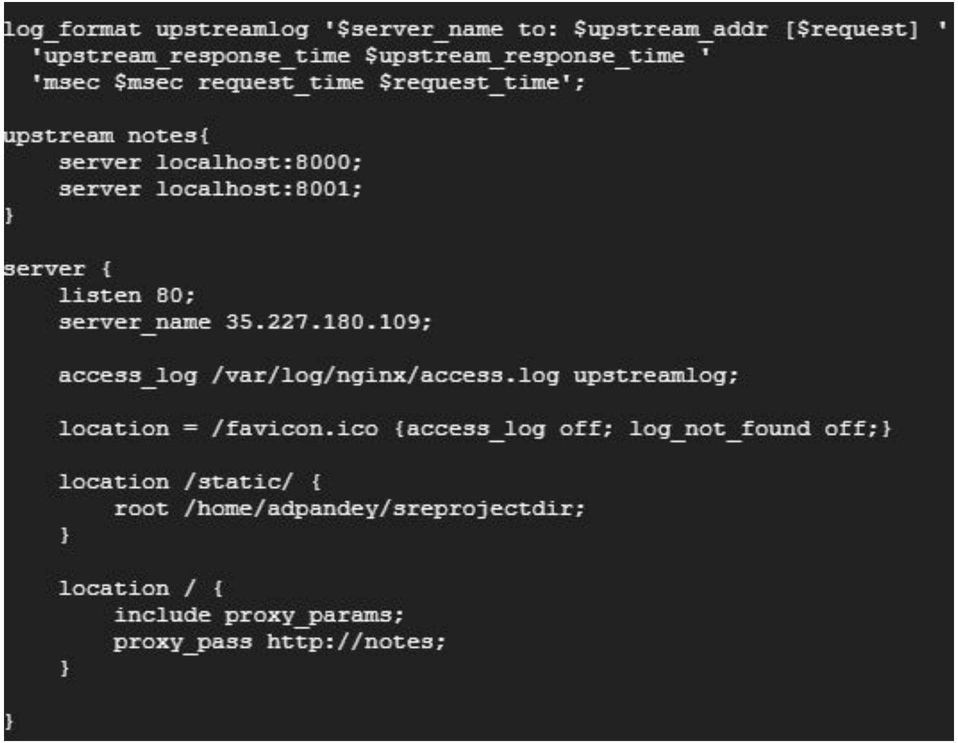
```
[supervisorctl]
serverurl=http://localhost:8000
[program:gunicorn-8000]
command=/home/adpandey/sreprojectdir/sreprojectenv/bin/gunicorn sreproject.wsgi:application --bind 0.0.0.0:8000
directory=/home/adpandey/sreprojectdir
stderr_logfile = /var/log/supervisord/gunicornerr.log
stdout_logfile = /var/log/supervisord/gunicornout.log

[program:gunicorn-8001]
command=/home/adpandey/sreprojectdir/sreprojectenv/bin/gunicorn sreproject.wsgi:application --bind 0.0.0.0:8001
directory=/home/adpandey/sreprojectdir
stderr_logfile = /var/log/supervisord/gunicornerr.log
stdout_logfile = /var/log/supervisord/gunicornout.log
```

Nginx configuration

- Redirect traffic to different servers both serving on localhost, port 8000 and 8001
- Fairly simple
- Logs below show traffic being forwarded correctly



```
35.227.180.109 to: 127.0.0.1:8001 [GET / HTTP/1.1]
35.227.180.109 to: 127.0.0.1:8000 [POST /editBlackA
35.227.180.109 to: 127.0.0.1:8001 [GET / HTTP/1.1]
35.227.180.109 to: 127.0.0.1:8000 [GET / HTTP/1.1]
35.227.180.109 to: 127.0.0.1:8001 [GET /manager/htm
35.227.180.109 to: 127.0.0.1:8000 [GET / HTTP/1.1]
```



```
log_format upstreamlog '$server_name to: $upstream_addr [$request] '
  'upstream_response_time $upstream_response_time '
  'msec $msec request_time $request_time';

upstream notes{
    server localhost:8000;
    server localhost:8001;
}

server {
    listen 80;
    server_name 35.227.180.109;

    access_log /var/log/nginx/access.log upstreamlog;

    location = /favicon.ico {access_log off; log_not_found off;}

    location /static/ {
        root /home/adpandey/sreprojectdir;
    }

    location / {
        include proxy_params;
        proxy_pass http://notes;
    }

}
```

Database Redundancy

Database redundancy

We have Postgres Database running on separate Virtual Machines:

- **A primary server runs the active database. This database accepts connections from django app and permits read-write operations.**
- **Standby server runs a copy of the active database in Hot Standby Mode.**

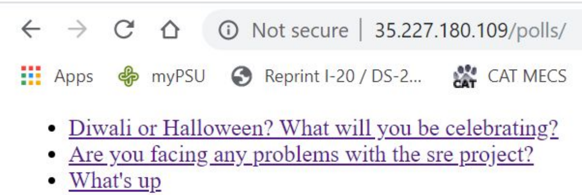
Primary Server Configurations:

1. Created ***repuser*** for replication activities.
2. Created a directory to store archive files.
3. Made changes to configuration files.
   (pg_hba.conf and postgresql.conf)
4. Restarted the primary server

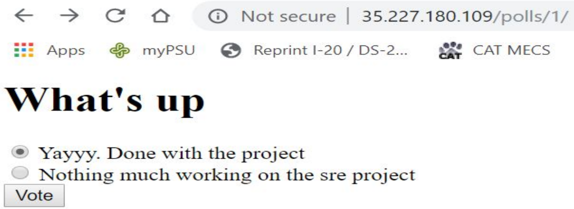
StandBy Server Configurations (Streaming Replication):

1. Stopped the standby server.
2. Ran the backup utility to copy files from the data directory on the primary server to standby server
3. Created recovery.conf file.
4. Started the StandBy Server.

Database redundancy



- Diwali or Halloween? What will you be celebrating?
- Are you facing any problems with the sre project?
- What's up



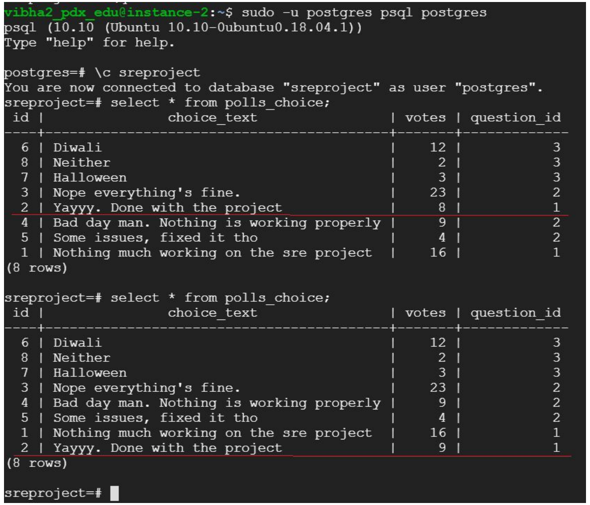
What's up

● Yayyy. Done with the project

○ Nothing much working on the sre project

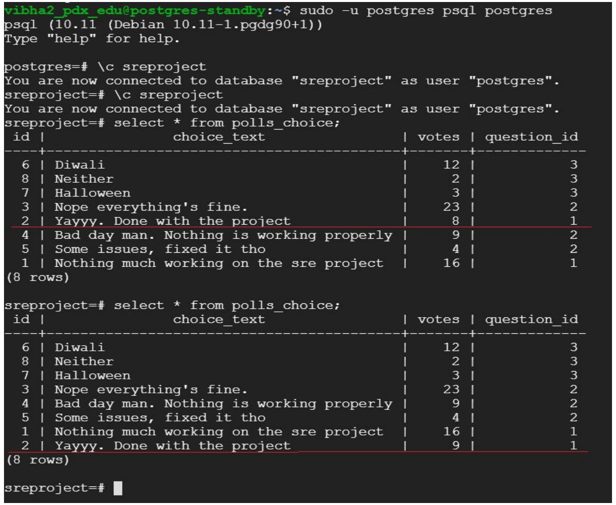
Vote

Primary Database



```
vibha2_pdx_edu@instance-2:~$ sudo -u postgres psql postgres
psql (10.10 (Ubuntu 10.10-0ubuntu0.18.04.1))
Type "help" for help.

postgres=# \c sreproject
You are now connected to database "sreproject" as user "postgres".
sreproject=# select * from polls_choice;
 id |               choice_text                | votes | question_id
----+------------------------------------------+-------+-------------
  6 | Diwali                                   |    12 |           3
  8 | Neither                                  |     2 |           3
  7 | Halloween                                |     3 |           3
  3 | Nope everything's fine.                  |    23 |           2
  2 | Yayyy. Done with the project             |     8 |           1
  4 | Bad day man. Nothing is working properly |     9 |           2
  5 | Some issues, fixed it tho                |     4 |           2
  1 | Nothing much working on the sre project  |    16 |           1
(8 rows)

sreproject=# select * from polls_choice;
 id |               choice_text                | votes | question_id
----+------------------------------------------+-------+-------------
  6 | Diwali                                   |    12 |           3
  8 | Neither                                  |     2 |           3
  7 | Halloween                                |     3 |           3
  3 | Nope everything's fine.                  |    23 |           2
  4 | Bad day man. Nothing is working properly |     9 |           2
  5 | Some issues, fixed it tho                |     4 |           2
  1 | Nothing much working on the sre project  |    16 |           1
  2 | Yayyy. Done with the project             |     9 |           1
(8 rows)

sreproject=# 
```

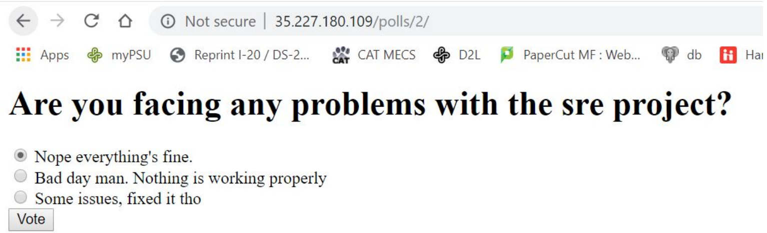
StandBy Database



```
vibha2_pdx_edu@postgres-standby:~$ sudo -u postgres psql postgres
psql (10.11 (Debian 10.11-1.pgdg90+1))
Type "help" for help.

postgres=# \c sreproject
You are now connected to database "sreproject" as user "postgres".
sreproject=# \c sreproject
You are now connected to database "sreproject" as user "postgres".
sreproject=# select * from polls_choice;
 id |               choice_text                | votes | question_id
----+------------------------------------------+-------+-------------
  6 | Diwali                                   |    12 |           3
  8 | Neither                                  |     2 |           3
  7 | Halloween                                |     3 |           3
  3 | Nope everything's fine.                  |    23 |           2
  2 | Yayyy. Done with the project             |     8 |           1
  4 | Bad day man. Nothing is working properly |     9 |           2
  5 | Some issues, fixed it tho                |     4 |           2
  1 | Nothing much working on the sre project  |    16 |           1
(8 rows)

sreproject=# select * from polls_choice;
 id |               choice_text                | votes | question_id
----+------------------------------------------+-------+-------------
  6 | Diwali                                   |    12 |           3
  8 | Neither                                  |     2 |           3
  7 | Halloween                                |     3 |           3
  3 | Nope everything's fine.                  |    23 |           2
  4 | Bad day man. Nothing is working properly |     9 |           2
  5 | Some issues, fixed it tho                |     4 |           2
  1 | Nothing much working on the sre project  |    16 |           1
  2 | Yayyy. Done with the project             |     9 |           1
(8 rows)

sreproject=# 
```

Chaos Testing

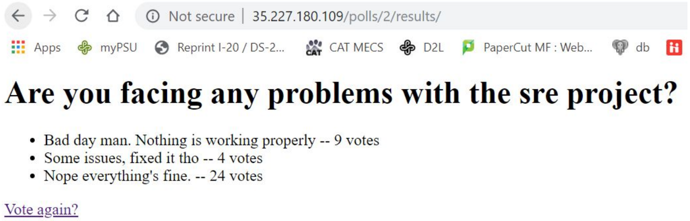
- Database Replication is in place but what happens if the Primary Database goes down ?
- Even though this scenario is very rare, we expect it to happen and want to be prepared.



Are you facing any problems with the sre project?

● Nope everything's fine.
○ Bad day man. Nothing is working properly
○ Some issues, fixed it tho

Vote

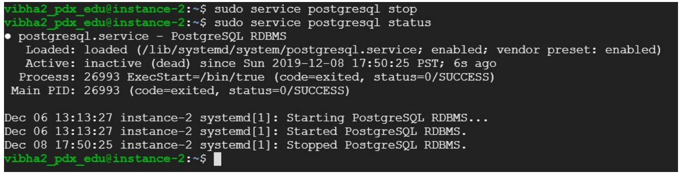


Are you facing any problems with the sre project?

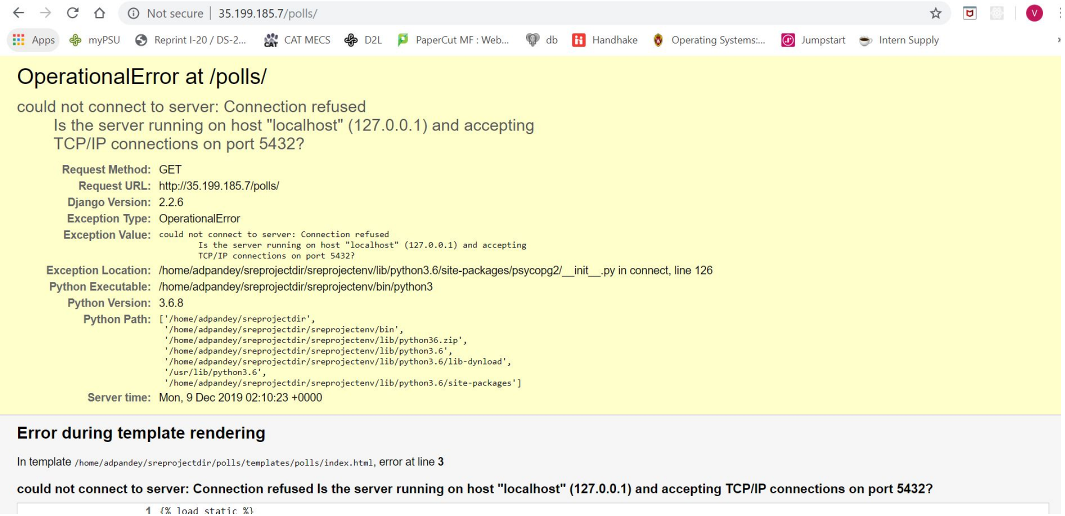
- Bad day man. Nothing is working properly -- 9 votes
- Some issues, fixed it tho -- 4 votes
- Nope everything's fine. -- 24 votes

Vote again?

Chaos Testing



```
vibha2_pdx_edu@instance-2:~$ sudo service postgresql stop
vibha2_pdx_edu@instance-2:~$ sudo service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sun 2019-12-08 17:50:25 PST; 6s ago
  Process: 26993 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 26993 (code=exited, status=0/SUCCESS)

Dec 06 13:13:27 instance-2 systemd[1]: Starting PostgreSQL RDBMS...
Dec 06 13:13:27 instance-2 systemd[1]: Started PostgreSQL RDBMS.
Dec 08 17:50:25 instance-2 systemd[1]: Stopped PostgreSQL RDBMS.
vibha2_pdx_edu@instance-2:~$ 
```



OperationalError at /polls/

```
could not connect to server: Connection refused
	Is the server running on host "localhost" (127.0.0.1) and accepting
	TCP/IP connections on port 5432?
```

Request Method: GET
Request URL: http://35.199.185.7/polls/
Django Version: 2.2.6
Exception Type: OperationalError
Exception Value:

```
could not connect to server: Connection refused
	Is the server running on host "localhost" (127.0.0.1) and accepting
	TCP/IP connections on port 5432?
```

Exception Location: /home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/site-packages/psycopg2/__init__.py in connect, line 126
Python Executable: /home/adpandey/sreprojectdir/sreprojectenv/bin/python3
Python Version: 3.6.8
Python Path:

```
['/home/adpandey/sreprojectdir',
 '/home/adpandey/sreprojectdir/sreprojectenv/bin',
 '/home/adpandey/sreprojectdir/sreprojectenv/lib/python36.zip',
 '/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6',
 '/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/lib-dynload',
 '/usr/lib/python3.6',
 '/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/site-packages']
```

Server time: Mon, 9 Dec 2019 02:10:23 +0000

Error during template rendering

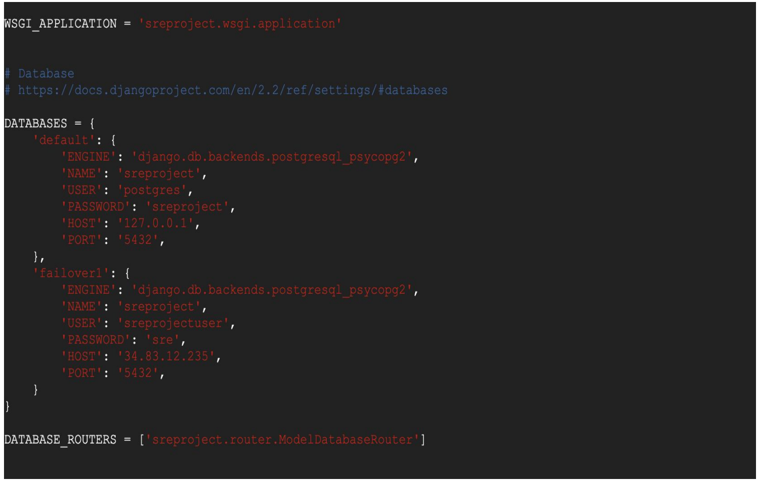
In template `/home/adpandey/sreprojectdir/polls/templates/polls/index.html`, error at line **3**

could not connect to server: Connection refused Is the server running on host "localhost" (127.0.0.1) and accepting TCP/IP connections on port 5432?

```
1 {% load static %}
```

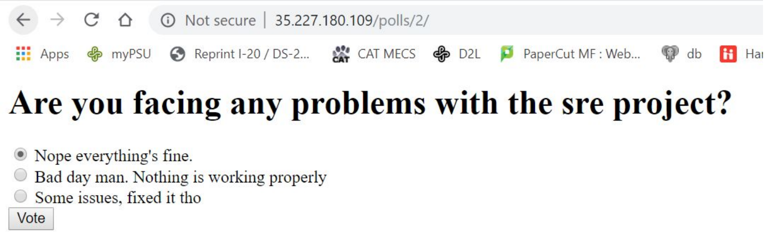
Chaos Testing

- One solution we thought of was to configure Django to handle this situation.



```
WSGI_APPLICATION = 'sreproject.wsgi.application'


# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'sreproject',
        'USER': 'postgres',
        'PASSWORD': 'sreproject',
        'HOST': '127.0.0.1',
        'PORT': '5432',
    },
    'failover1': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'sreproject',
        'USER': 'sreprojectuser',
        'PASSWORD': 'sre',
        'HOST': '34.83.12.235',
        'PORT': '5432',
    }
}

DATABASE_ROUTERS = ['sreproject.router.ModelDatabaseRouter']
```

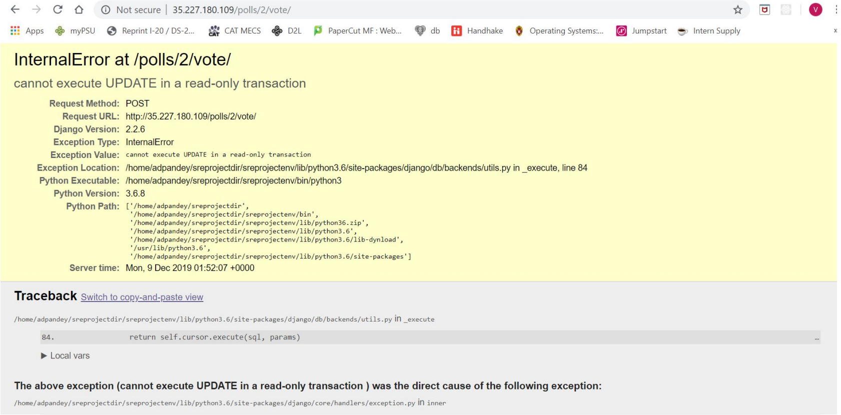
Chaos Testing



Are you facing any problems with the sre project?

- ● Nope everything's fine.
- ○ Bad day man. Nothing is working properly
- ○ Some issues, fixed it tho

Vote



```
InternalError at /polls/2/vote/
cannot execute UPDATE in a read-only transaction

   Request Method: POST
      Request URL: http://35.227.180.109/polls/2/vote/
   Django Version: 2.2.6
   Exception Type: InternalError
  Exception Value: cannot execute UPDATE in a read-only transaction
Exception Location: /home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/site-packages/django/db/backends/utils.py in _execute, line 84
Python Executable: /home/adpandey/sreprojectdir/sreprojectenv/bin/python3
   Python Version: 3.6.8
      Python Path: ['/home/adpandey/sreprojectdir',
                    '/home/adpandey/sreprojectdir/sreprojectenv/bin',
                    '/home/adpandey/sreprojectdir/sreprojectenv/lib/python36.zip',
                    '/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6',
                    '/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/lib-dynload',
                    '/usr/lib/python3.6',
                    '/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/site-packages']
      Server time: Mon, 9 Dec 2019 01:52:07 +0000

Traceback Switch to copy-and-paste view

/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/site-packages/django/db/backends/utils.py in _execute

    84.                 return self.cursor.execute(sql, params)                                  ...

    ▶ Local vars

The above exception (cannot execute UPDATE in a read-only transaction ) was the direct cause of the following exception:

/home/adpandey/sreprojectdir/sreprojectenv/lib/python3.6/site-packages/django/core/handlers/exception.py in inner
```

Django connects to StandBy server and tries to execute write operation.

Chaos Testing

- Postgres doesn't provide functionality to automatically fail over when the primary server fails.
- The Django application connects to the Standby Database however it cannot write to it since the Standby Database is still in hot standby mode.
- Handling database failover can either be done by
  - Executing ***pg_ctl promote*** command manually on StandBy server.
  - Using third-party solution such as repmgr.

Thank You