
postgraas*serverDocumentation*

Sebastian Neubauer

Aug 17, 2018

Contents

1	Contents	3
1.1	Installation	3
1.2	Usage	5
1.3	Configuration	6
1.4	Troubleshooting	6
1.5	License	7
1.6	Developers	10
1.7	Changelog	11
1.8	postgraas_server	14
2	Indices and tables	15

This is the documentation of **postgraas_server**.

1.1 Installation

1.1.1 Prerequisites

- Tested on Debian/Ubuntu
- Python tested with 2.7
- Any wsgi server, tested with gunicorn
- recent pip & setuptools
- Docker Engine >= 1.12.0 (for the `docker` backend, see `postgraas-backends`)
- PostgreSQL Server (for the `pg_cluster` backend, see `postgraas-backends`)

Install via pip:

```
pip install postgraas_server
```

1.1.2 Docker

Docker is the default backend of Postgraas. To use it, pull the official postgres 9.4 docker [image](#)

```
docker pull postgres:9.4
```

We need a postgres for the service to persist the information about the created instances and where do we get one? Yes, we create a docker postgres instance just like all other instances.

First we need a config file where we set the credentials for the meta database and this need to be called *postgraas_server.cfg* and must be located in the CWD:

```
[metadb]
db_name = postgraas
db_username = postgraas
db_pwd = very_secure_postgraas
host = localhost
port = 5432
```

For the creation of the meta db container and creation of the schema, there is a convenience entry point:

```
postgraas_init
```

With this, everything is prepared and we can start the postgraas server, for example with gunicorn:

```
gunicorn -w 4 -b 0.0.0.0:8080 postgraas_server.postgraas_api:app
```

This will start the server, listening on port 8080 and all incoming interfaces (DISCLAIMER: please don't use anything like this in a production like environment)

Ready to serve your postgres instances!

1.1.3 PostgreSQL Cluster

In case you don't want to use Docker and just use a plain Postgres cluster instance, you have to add another section to your config to change the default backend:

```
[backend]
type = pg_cluster
host = localhost
port = 5432
database = postgres
username = postgres
password = S3cr3t
```

The given user has to be able to create new roles and databases on the specified cluster.

In case you need to log in with usernames of the form <username@hostname>, you can use the optional `server` config option:

```
[backend]
type = pg_cluster
host = localhost
port = 5432
database = postgres
username = postgres
password = S3cr3t
server = postgraas-instance-1
```

Note: A big difference to the Docker backend is that with the `pg_cluster` backend users only get single database, while for the docker backend they get a whole Postgres instance.

1.2 Usage

Intentionally the usage is a simple stupid REST'ish api, in fact there is only *create* a new instance, *delete* an existing instance, list all instances and *read/get* the details of an instance.

1.2.1 Create a postgraas Instance

We need to send all the required parameters for the creation as an http request. This is quite convenient by creating a file e.g. *my_postgraas.json*:

```
{
  "postgraas_instance_name": "my_postgraas",
  "db_name": "my_db",
  "db_username": "db_user",
  "db_pwd": "secret"
}
```

and making a POST request to the collection resource with *curl*:

```
curl -H "Content-Type: application/json" -X POST --data @my_postgraas.json http://
↳localhost:8080/api/v2/postgraas_instances
```

now your instance is created and as a response you get the details of your instance:

```
{
  "postgraas_instance_id": 1,
  "container_id": "193f0d94d49fa26626fdbdb583e9453f923468b01eac59207b4852831a105c03
↳",
  "db_pwd": "secret",
  "host": "not imlemented yet",
  "db_name": "my_db",
  "db_username": "db_user",
  "port": 54648
}
```

We are now able to connect to the database for exaple with *psql*:

```
psql -h localhost -p 54648 -U db_user my_db
```

Awesome, isn't it?

1.2.2 List all instances

With a GET request to the collection resource you get a list of all instances (not only the one you created...):

```
curl -H "Content-Type: application/json" -X GET http://localhost:8080/api/v2/
↳postgraas_instances
```

And the response is the list:

```
[
  {
    "username": "db_user",
    "container_id":
↳"193f0d94d49fa26626fdbdb583e9453f923468b01eac59207b4852831a105c03",
```

(continues on next page)

(continued from previous page)

```
    "db_name": "my_db",
    "postgraas_instance_name": "my_postgraas",
    "password": "",
    "creation_timestamp": "2016-07-06T23:38:38.367493",
    "id": 1,
    "hostname": "not imlemented yet",
    "port": "54648"
  }
]
```

1.2.3 Delete an instance

With a DELETE request to the uri of the instance you can delete an instance:

```
curl -H "Content-Type: application/json" -X DELETE --data @pwd.json http://
↪localhost:8080/api/v2/postgraas_instances/1
```

where the *pwd.json* file contains the postgres password as “db_pwd” key, e.g.:

```
{
  "db_pwd": "secret"
}
```

You will get a response like this:

```
{
  "status": "success",
  "msg": "deleted postgraas instance"
}
```

That’s it at the moment!

1.3 Configuration

Besides the already seen backend config options, it is also possible to configure the application itself. You can for example change the behaviour of *sqlalchemy* or integrate *sentry*:

```
[application]
SQLALCHEMY_POOL_RECYCLE = 120
SQLALCHEMY_POOL_SIZE = 1
SENTRY_DSN = https://<key>:<secret>@sentry.io/<project>
SENTRY_ENVIRONMENT = production
```

1.4 Troubleshooting

1.4.1 Errors while installing the requirements

‘Error: pg_config executable not found.’

You need to install the dev version of *libpq*. On Debian Jessie for example, you can do this with:

```
apt-get install libpq-dev
```

‘/psycopg/psycopg.h:30:20: fatal error: Python.h: No such file or directory’

You need to install the dev version of Python. On Debian Jessie for example, you can do this with:

```
apt-get install python-dev
```

1.5 License

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

(continues on next page)

(continued from previous page)

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

(continues on next page)

(continued from previous page)

- excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

(continues on next page)

(continued from previous page)

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2016 Blue Yonder GmbH

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.6 Developers

- Patrick Muehlbauer <patrick.muehlbauer@blue-yonder.com>
- Sebastian Neubauer <sebineubauer@gmail.com>

1.7 Changelog

1.7.1 UNRELEASED

- Fix log messages of Gunicorn and Flask not ending up in the root logger.

1.7.2 v2.0.2

- return proper HTTP status codes, when deleting postgraas instances

1.7.3 v2.0.1

- prevent creation of databases with an empty password as those cannot be removed.
- return http status code 409 instead of 200 in case of a name conflict during creation of a new instance.

1.7.4 v2.0.0

- security: Harden the Postgres cluster backend against SQL injections.
- hardened the server against leaking of database connections and transactions.

1.7.5 v2.0.0b1

- breaking change: config not ini anymore, but json
- add support for secure-config (experimental)

1.7.6 v1.0.0b3

- add support for simple Postgres Cluster as backend
- add support for sentry (<https://sentry.io>)
- bugfix: varchar(50) field in the DB too small for hostname

1.7.7 v0.1.9

- added authentication to delete resource

1.7.8 v0.1.8

- added init script
- fixed postgraas to 9.4

1.7.9 v0.1.7

- fixed docker version 1.12

1.7.10 v0.1.6

- bugfix: password was not set in instance

1.7.11 v0.1.5

- automated release process via travis
- fixed meta db initialization

1.7.12 v0.1.4

- removed postgraas instance one and fixed the tests

1.7.13 v0.1.3

- Introduced completely new API v2

1.7.14 v0.1.0

- First running version

1.8 postgraas_server

1.8.1 postgraas_server package

Subpackages

postgraas_server.backends package

Subpackages

postgraas_server.backends.docker package

Submodules

postgraas_server.backends.docker.create_db module

postgraas_server.backends.docker.postgres_instance_driver module

Module contents

postgraas_server.backends.postgres_cluster package

Submodules

postgraas_server.backends.postgres_cluster.postgres_cluster_driver module

Module contents

Submodules

postgraas_server.backends.exceptions module

Module contents

Submodules

postgraas_server.configuration module

postgraas_server.create_app module

postgraas_server.init_db module

postgraas_server.management_database module

postgraas_server.management_resources module

postgraas_server.postgraas_api module

postgraas_server.prometheus_app module

postgraas_server.utils module

Module contents

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`