DeepDive installation guide

Quick installation

We provide a simple installation method for the following supported systems:

- GNU/Linux: Debian (7, 8, or later) and Ubuntu (12.04LTS, 14.04LTS, 16.04LTS, or later)
    - Docker
    - VirtualBox
    - AWS EC2
- macOS with Homebrew

You can install DeepDive and all its dependencies with a single command.

1. Open your terminal and run this:

    ```
    bash <(curl -fsSL git.io/getdeepdive)
    ```

2. Select `deepdive` or `deepdive_from_release` when asked. Choose the latter option if you simply want to install DeepDive without any of its runtime dependencies.

    ```
    ### DeepDive installer for Mac
    1) deepdive                  5) jupyter_notebook
    2) deepdive_docker_sandbox    6) postgres
    3) deepdive_example_notebook  7) run_deepdive_tests
    4) deepdive_from_release      8) spouse_example
    # Install what (enter to repeat options, a to see all, q to quit,
    or a number)? 1
    [...]
    ```

    Here are some more details of each options:

    - For installation with the `deepdive` option, **All runtime dependencies are installed**. Note that some steps may ask your password.
    - If you don't have permission to install the dependencies, you may want to use the `deepdive_from_release` option and ask the system administrator to install DeepDive's runtime dependencies with the following command.

    ```
    bash <(curl -fsSL git.io/getdeepdive) _deepdive_runtime_deps
    ```

```
* For installation with `deepdive_from_source` option, extra build depen
dencies are installed, and DeepDive source tree is cloned at `./deepdive
`, then executables are installed under `~/local/bin/`.
```

~~~
You can run tests in DeepDive's source tree to make sure everything will
run fine.
See the [developer's guide](developer#build-test) for more details.
    ```bash
    cd ./deepdive
    make test
    ```
~~~

1. To use the `deepdive` command on a regular basis, it is recommended to add the following line to your `~/.bash_profile`. Otherwise, you need to always type its full path: `~/local/bin/deepdive`.**(在 Home 文件夹下，Ctrl+H 显示隐藏文件，在.profile 文件中添加如下一行命令，重启后生效)**

```
export PATH=~/local/bin:"$PATH"
```

2. Since DeepDive needs a database installation to run correctly. You should use one of the provided installer options:

   - `postgres` DeepDive works with most of the recent versions of PostgreSQL. However, 9.3+ is recommended to use all functionality.
   - `postgres_xl` DeepDive works with current release of PostgreSQL-XL, which is based on PostgreSQL 9.2. PL/Python extension is required.
   - `greenplum` DeepDive works with recent releases of Greenplum, which is based on PostgreSQL 8 that may lack some features that are required by some advanced DeepDive functionality. PL/Python extension is required.
   - `mysql` DeepDive provides minimal support for MySQL and MySQL Cluster, but **PostgreSQL-based databases are strongly recommended.**

3. You can verify whether your installation is correct using the `run_deepdive_tests` option in the installer. It downloads all examples and tests for the DeepDive release and runs the tests using the installed one. To only download the example applications, use the `deepdive_examples_tests` option.

Congratulations! DeepDive is now installed on your system, and you can proceed to the next steps.

Installing from source

Using the quick installation method is recommended unless you want to use a development branch or modify DeepDive engine itself. If you still want to build DeepDive from source code yourself, follow the developer's guide.

Installing PostgreSQL

Deepdive 提供了 postgres 的安装选项，与常规安装相比，deepdive 还自动执行了一些设置：

1. 版本：9.5
2. 函数脚本路径：/var/lib/postgresql/
3. 创建了一个数据库的 superuser，为当前用户，不用密码
4. 设置了 pg_hba.conf 配置文件，路径为/etc/postgresql/9.5/main/pg_hba.conf
5. 重启 postgresql service

测试：在当前用户下，能正常运行 createdb 和 dropdb，以及 psql 命令

由于默认情况下，PostgreSQL 不允许 TCP/IP 连接，需要进行更改：

1. 在 root 权限下，编辑/etc/postgresql/9.5/main/postgresql.conf 文件，将

```
#listen_addresses = 'localhost'
```

改为

listen_addresses = 'localhost' 或 listen_addresses = '*'

2. 重启服务

```
sudo service postgresql restart
```

3. 切换回当前用户，测试 TCP/IP 连接

```
psql -h 127.0.0.1 -p 5432 -U postgres
```

如能进入 psql 界面，说明连接成功

Installing pgAdmin III

安装 sudo apt-get install pgadmin3

然后点击搜索，输入 pgadmin，运行搜索到的该 application

程序运行后，可连接至目标数据库，通过可视化的界面来查看该数据库