# 7 Deployment

## Overview

The Software Project application is developed by JavaScript in the front-end and Python3 Django framework and MySQL databases in the back-end.

The steps for installing and deploying Software Project easily is as follows:

## 1.0 To start a front-end for developing:

1. install npm, can refer to https://www.npmjs.com/get-npm
2. Install the front-end project, in CLI: `npm install`
3. `npm start The front-end is now running on http://localhost:3000/`

## 2.0 To start a back-end for developing:

1. Install Python3.7 and MySQL
2. Install Understand tool. Follow the instructions in 7.2 Scitools Understand Deployment
3. Start MySQL server on localhost:3306 `sudo service mysql start`, and create a database named "sp90013", i.e., run "**mysql -u root -p**", input password, then "**`CREATE DATABASE sp90013;`**"
4. Clone git repository, change the working directory to COMP90082_Software_Project_Database_Backend
5. Install all packages needed "**`pip install -r requirements.txt`**" `(Do not install another version of packages which can help you miss many unwanted mistakes) If multiple version of python are installed, use python3 and pip3 or python3.x and pip3.x instead.`
6. Modify the MySQL username and password config in TeamSPBackend/settings/dev.py and TeamSPBackend/settings/prod.py (don't forget to modify 'DATABASES/default/TEST/PASSWORD' in prod.py)
7. Create MySQL tables "**`python manage.py migrate`**".
8. `Create a log folder` **`"mkdir ../logs"`**
9. Start server **"nohup `python manage.py runserver 0.0.0.0:18000 >> ../logs/1.log`",** the back-end is now running on http://127.0.0.1:18000/ . This command will store the output of the program from the command line to the 1.log file. And will keep the process alive in the background.

Reference

1. Sample Deployment Checklist: https://smartbear.com/blog/sample-deployment-checklist/
2. http://pages.cs.wisc.edu/~mikem/puddles/deployment
3. An example: https://docs.bmc.com/docs/ReleasePackageDeploy/50/using/example-of-a-web-application-deployment

# 7.1 Cloud Server Deployment

## Procedure

- Install MySQL
  - rpm -Uvh http://dev.mysql.com/get/mysql-community-release-el6-5.noarch.rpm
  - yum install mysql-server
- Start MySQL
  - service mysqld start
- Install pip
  - wget https://bootstrap.pypa.io/get-pip.py
  - python3 get-pip.py
- Install Git
  - sudo yum install git
- Install SSL Certification chain mitigation
  - (UAC/sudo required):

    ```
    python3.7 ca.py
    ```

  - or install manually:
    - get certificate directory by running:

    ```
    python3.7 -c 'import certifi; print(certifi.where())'
    ```

    - appending contents of chain.pem to the file cacert.pem in certificate directory
- Clone Git Repository
- Install requirement and edit setting File as 7 Deployment
- Run server
  - python3 manage.py runserver 0.0.0.0:18000

## SSL Certificate Issue:

### Description

The unimelb Atlassian stack runs a mis-configured server which sends an incomplete SSL certificate chain missing intermediate certificate QuoVadis Global SSL ICA G3, which is present on most browsers but not in urllib3(openssl).

We've provided one of the client side mitigations with ca.py that installs complete certificate chain exported from Chrome.

You can learn more about other possible solutions: https://urllib3.readthedocs.io/en/latest/advanced-usage.html

It is **highly recommended to configure the server to send the complete certificate chain**, as client side mitigation is only temporary, the installation script will warn about the certificate expiration date (2022-04-16 01:55:00).

### Export and generate the certification chain manually

If the certificate chain we provided expired and the server continues to behave this way, the Atlassian subsystems in our product **will not run correctly**.

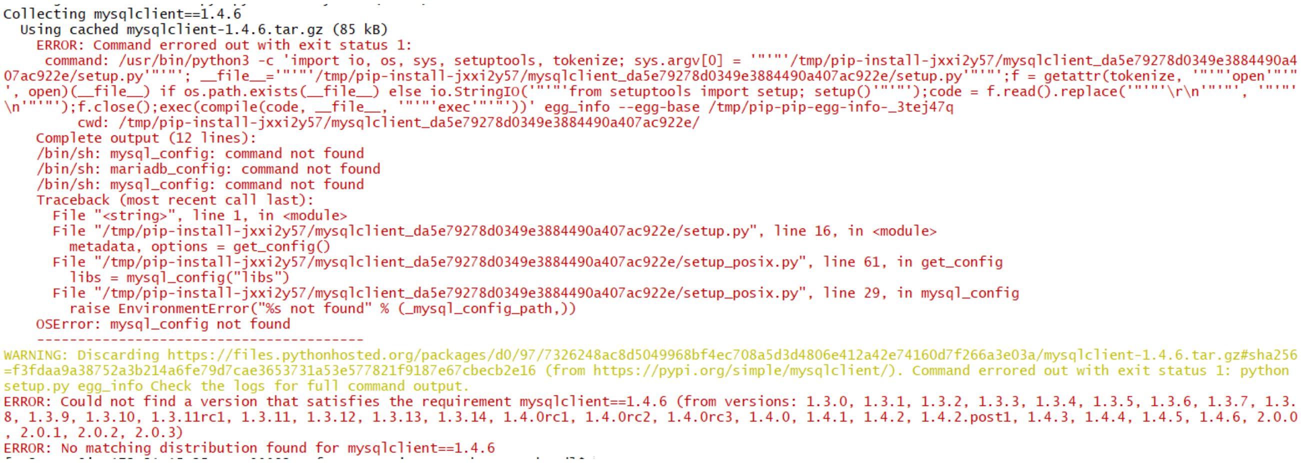
- Go to https://jira.cis.unimelb.edu.au:8444/ or https://confluence.cis.unimelb.edu.au:8443/ with Chromium browsers
- Click the lock icon in the address bar
- Click Certificate
- Click Details
- You can see the certificates used in the chain in Certificate Hierachy
- Export the each and every certificate, **starting from the bottom one**
- Create your own chain.pem file and enter these certificates **in that order**
- Run the ca.py script on your chain.pem file or paste them manually to the CA file

You can export certificates in similar ways with Firefox, Edge, Opera etc.

## Errors and Solution

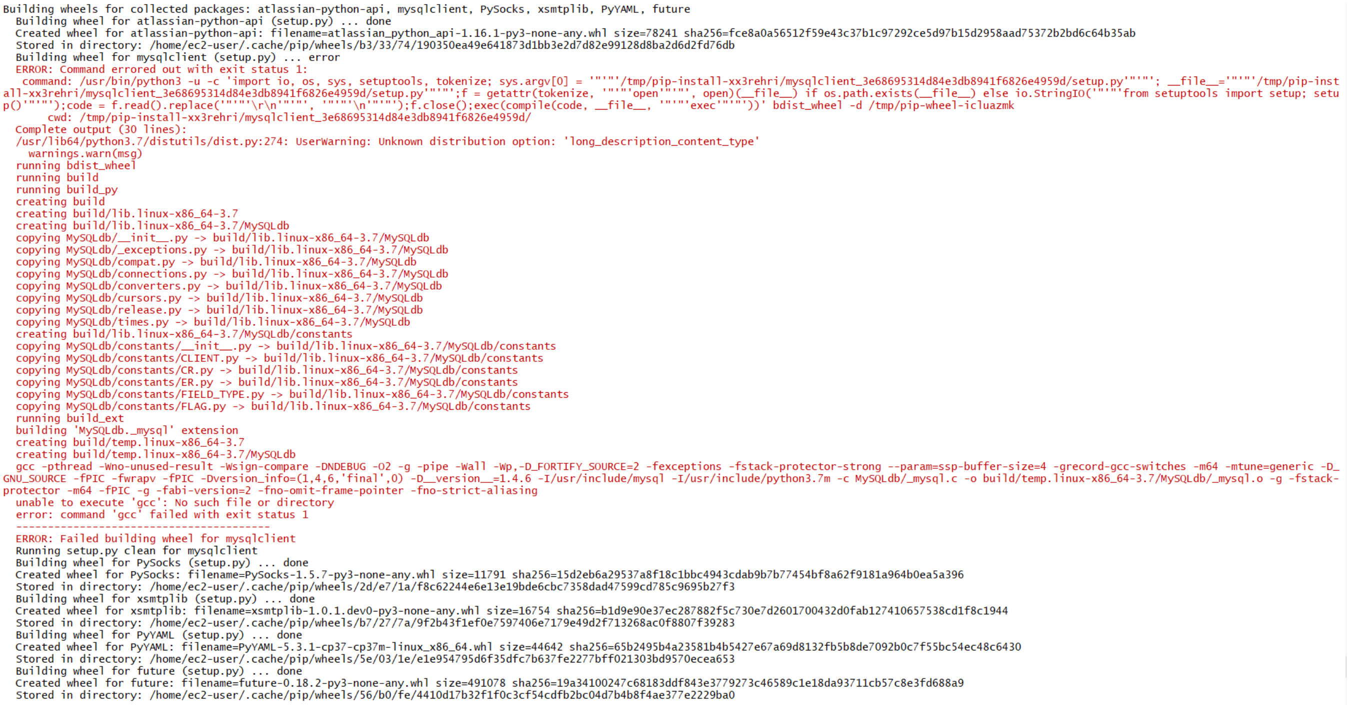
When installing mysqlclient on Linux, some problems happened.

OSError: mysql_config not found



```
Collecting mysqlclient==1.4.6
  Using cached mysqlclient-1.4.6.tar.gz (85 kB)
    ERROR: Command errored out with exit status 1:
     command: /usr/bin/python3 -c 'import io, os, sys, setuptools, tokenize; sys.argv[0] = '"'"'/tmp/pip-install-jxxi2y57/mysqlclient_da5e79278d0349e3884490a4
07ac922e/setup.py'"'"'; __file__='"'"'/tmp/pip-install-jxxi2y57/mysqlclient_da5e79278d0349e3884490a407ac922e/setup.py'"'"';f = getattr(tokenize, '"'"'open'"'"
', open)(__file__) if os.path.exists(__file__) else io.StringIO('"'"'from setuptools import setup; setup()'"'"');code = f.read().replace('"'"'\r\n'"'"', '"'"'
\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' egg_info --egg-base /tmp/pip-pip-egg-info-_3tej47q
         cwd: /tmp/pip-install-jxxi2y57/mysqlclient_da5e79278d0349e3884490a407ac922e/
    Complete output (12 lines):
    /bin/sh: mysql_config: command not found
    /bin/sh: mariadb_config: command not found
    /bin/sh: mysql_config: command not found
    Traceback (most recent call last):
      File "<string>", line 1, in <module>
      File "/tmp/pip-install-jxxi2y57/mysqlclient_da5e79278d0349e3884490a407ac922e/setup.py", line 16, in <module>
        metadata, options = get_config()
      File "/tmp/pip-install-jxxi2y57/mysqlclient_da5e79278d0349e3884490a407ac922e/setup_posix.py", line 61, in get_config
        libs = mysql_config("libs")
      File "/tmp/pip-install-jxxi2y57/mysqlclient_da5e79278d0349e3884490a407ac922e/setup_posix.py", line 29, in mysql_config
        raise EnvironmentError("%s not found" % (_mysql_config_path,))
    OSError: mysql_config not found
    ----------------------------------------
WARNING: Discarding https://files.pythonhosted.org/packages/d0/97/7326248ac8d5049968bf4ec708a5d3d4806e412a42e74160d7f266a3e03a/mysqlclient-1.4.6.tar.gz#sha256
=f3fdaa9a38752a3b214a6fe79d7cae3653731a53e577821f9187e67cbecb2e16 (from https://pypi.org/simple/mysqlclient/). Command errored out with exit status 1: python
setup.py egg_info Check the logs for full command output.
ERROR: Could not find a version that satisfies the requirement mysqlclient==1.4.6 (from versions: 1.3.0, 1.3.1, 1.3.2, 1.3.3, 1.3.4, 1.3.5, 1.3.6, 1.3.7, 1.3.
8, 1.3.9, 1.3.10, 1.3.11rc1, 1.3.11, 1.3.12, 1.3.13, 1.3.14, 1.4.0rc1, 1.4.0rc2, 1.4.0rc3, 1.4.0, 1.4.1, 1.4.2, 1.4.2.post1, 1.4.3, 1.4.4, 1.4.5, 1.4.6, 2.0.0
, 2.0.1, 2.0.2, 2.0.3)
ERROR: No matching distribution found for mysqlclient==1.4.6
```

ERROR: Failed building wheel for mysqlclient



```
Building wheels for collected packages: atlassian-python-api, mysqlclient, PySocks, xsmtplib, PyYAML, future
  Building wheel for atlassian-python-api (setup.py) ... done
  Created wheel for atlassian-python-api: filename=atlassian_python_api-1.16.1-py3-none-any.whl size=78241 sha256=fce8a0a56512f59e43c37b1c97292ce5d97b15d2958aad75372b2bd6c64b35ab
  Stored in directory: /home/ec2-user/.cache/pip/wheels/b3/33/74/190350ea49e641873d1bb3e2d7d82e99128d8ba2d6d2fd76db
  Building wheel for mysqlclient (setup.py) ... error
  ERROR: Command errored out with exit status 1:
   command: /usr/bin/python3 -u -c 'import io, os, sys, setuptools, tokenize; sys.argv[0] = '"'"'/tmp/pip-install-xx3rehri/mysqlclient_3e68695314d84e3db8941f6826e4959d/setup.py'"'"'; __file__='"'"'/tmp/pip-inst
all-xx3rehri/mysqlclient_3e68695314d84e3db8941f6826e4959d/setup.py'"'"';f = getattr(tokenize, '"'"'open'"'"', open)(__file__) if os.path.exists(__file__) else io.StringIO('"'"'from setuptools import setup; setu
p()'"'"');code = f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' bdist_wheel -d /tmp/pip-wheel-icluazmk
       cwd: /tmp/pip-install-xx3rehri/mysqlclient_3e68695314d84e3db8941f6826e4959d/
  Complete output (30 lines):
  /usr/lib64/python3.7/distutils/dist.py:274: UserWarning: Unknown distribution option: 'long_description_content_type'
    warnings.warn(msg)
  running bdist_wheel
  running build
  running build_py
  creating build
  creating build/lib.linux-x86_64-3.7
  creating build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/__init__.py -> build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/_exceptions.py -> build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/compat.py -> build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/connections.py -> build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/converters.py -> build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/cursors.py -> build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/release.py -> build/lib.linux-x86_64-3.7/MySQLdb
  copying MySQLdb/times.py -> build/lib.linux-x86_64-3.7/MySQLdb
  creating build/lib.linux-x86_64-3.7/MySQLdb/constants
  copying MySQLdb/constants/__init__.py -> build/lib.linux-x86_64-3.7/MySQLdb/constants
  copying MySQLdb/constants/CLIENT.py -> build/lib.linux-x86_64-3.7/MySQLdb/constants
  copying MySQLdb/constants/CR.py -> build/lib.linux-x86_64-3.7/MySQLdb/constants
  copying MySQLdb/constants/ER.py -> build/lib.linux-x86_64-3.7/MySQLdb/constants
  copying MySQLdb/constants/FIELD_TYPE.py -> build/lib.linux-x86_64-3.7/MySQLdb/constants
  copying MySQLdb/constants/FLAG.py -> build/lib.linux-x86_64-3.7/MySQLdb/constants
  running build_ext
  building 'MySQLdb._mysql' extension
  creating build/temp.linux-x86_64-3.7
  creating build/temp.linux-x86_64-3.7/MySQLdb
  gcc -pthread -Wno-unused-result -Wsign-compare -DNDEBUG -O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic -D_
GNU_SOURCE -fPIC -fwrapv -fPIC -Dversion_info=(1,4,6,'final',0) -D__version__=1.4.6 -I/usr/include/mysql -I/usr/include/python3.7m -c MySQLdb/_mysql.c -o build/temp.linux-x86_64-3.7/MySQLdb/_mysql.o -g -fstack-
protector -m64 -fPIC -g -fabi-version=2 -fno-omit-frame-pointer -fno-strict-aliasing
  unable to execute 'gcc': No such file or directory
  error: command 'gcc' failed with exit status 1
  ----------------------------------------
  ERROR: Failed building wheel for mysqlclient
  Running setup.py clean for mysqlclient
  Building wheel for PySocks (setup.py) ... done
  Created wheel for PySocks: filename=PySocks-1.5.7-py3-none-any.whl size=11791 sha256=15d2eb6a29537a8f18c1bbc4943cdab9b7b77454bf8a62f9181a964b0ea5a396
  Stored in directory: /home/ec2-user/.cache/pip/wheels/2d/e7/1a/f8c62244e6e13e19bde6cbc7358dad47599cd785c9695b27f3
  Building wheel for xsmtplib (setup.py) ... done
  Created wheel for xsmtplib: filename=xsmtplib-1.0.1.dev0-py3-none-any.whl size=16754 sha256=b1d9e90e37ec287882f5c730e7d2601700432d0fab127410657538cd1f8c1944
  Stored in directory: /home/ec2-user/.cache/pip/wheels/b7/27/7a/9f2b43f1ef0e7597406e7179e49d2f713268ac0f8807f39283
  Building wheel for PyYAML (setup.py) ... done
  Created wheel for PyYAML: filename=PyYAML-5.3.1-cp37-cp37m-linux_x86_64.whl size=44642 sha256=65b2495b4a23581b4b5427e67a69d8132fb5b8de7092b0c7f55bc54ec48c6430
  Stored in directory: /home/ec2-user/.cache/pip/wheels/5e/03/1e/e1e954795d6f35dfc7b637fe2277bff021303bd9570ecea653
  Building wheel for future (setup.py) ... done
  Created wheel for future: filename=future-0.18.2-py3-none-any.whl size=491078 sha256=19a34100247c68183ddf843e3779273c46589c1e18da93711cb57c8e3fd688a9
  Stored in directory: /home/ec2-user/.cache/pip/wheels/56/b0/fe/4410d17b32f1f0c3cf54cdfb2bc04d7b4b8f4ae377e2229ba0
```

To solve the above two problems, install the dependencies of mysqlclient: `sudo yum install -y python3-devel mysql-devel gcc`

# 7.2 Scitools Understand Deployment

This document is a tutorial for deploying Understand (a powerful code analysis tool)  in Windows, Mac and Linux environment.

## I. Understand Environment Setting in Linux

1 make a directory
mkdir understand

2 download Understand

wget https://latest.scitools.com/Understand/Understand-6.0.1066-Linux-64bit.tgz

3 install Understand

tar xvfz Understand-6.0.1066-Linux-64bit.tgz

4 setting Environment Variables

4.1 open setting file
vi ~/etc/profile

4.2 add variables

export PYTHONPATH="${PYTHONPATH}:~/comp90082sp/understand/scitools/bin/linux64/Python"

export PATH="${PATH}:~/comp90082sp/understand/scitools/bin/linux64"

**export PATH="${LD_LIBRARY_PATH}:~/comp90082sp/understand/scitools/bin"**

For vm:

export PYTHONPATH="${PYTHONPATH}:~/spUnimelb/understand/scitools/bin/linux64/Python"

export PATH="${PATH}:~/spUnimelb/understand/scitools/bin/linux64"

**export PATH="${LD_LIBRARY_PATH}:~/spUnimelb/understand/scitools/bin"**

4.3 run setting file and make it valid
source /etc/profile

## II. python + und for getting metrics

### II.1 und command line mode:

1 Setting License:   und -setlicensecode XfA7YbMwUZ9OCYJd

2 create an und project and analyze its metrics:

there are 3 steps, for example:

```
2.1 und create -db KT.und -languages python C++ Java
2.2 und add ./Jerry-Shan-KT-blend-word-detection KT.und
2.3 und analyze KT.und
-->
Merge them togather:
und create -db KT.und -languages python C++ Java add ./Jerry-Shan-KT-blend-word-detection KT.und analyze
--> Template:
und create -db fileName.und -languages python C++ Java add filePath filename.und analyze
```

### II.2 Python API

```
import understand udb = understand.open('KT.und') metrics = udb.metric(udb.metrics()) return metrics
```

## II.3 Python API + und command line in Python Script

```
# using Understand for analyze Metrics
# For Mac
# UND_PATH = '/Applications/Understand.app/Contents/MacOS/'
# For Linux Server
UND_PATH = '~/comp90082sp/understand/scitools/bin/linux64/'
sys.path.append(UND_PATH)
sys.path.append(UND_PATH+'Python')
import understand

# set Understand License
UND_LICENSE = 'und -setlicensecode XfA7YbMwUZ9OCYJd'
os.system(UND_LICENSE)

# understand und command line for loading a git repo and generate metrics
UND_METRICS = UND_PATH + 'und create -db {} -languages python C++ Java add {} {} analyze'



und_file = construct_certification(repo, space_key) + '.und'
path = REPO_PATH + convert(repo)
und_metrics = UND_METRICS.format(und_file, path, und_file)
os.system(und_metrics)
# open a project und
udb = understand.open(und_file)
# get all project metrics
metrics = udb.metric(udb.metrics())
return metrics
```

# Reference

https://support.scitools.com/t/using-understand-from-the-command-line-with-und/79

https://support.scitools.com/t/getting-started-with-the-python-api/51