**Cloud COPASI Server Deployment guide**

This document describes the steps in sequence to setup cloud copasi on “linux CENTOS 7 server”.

NOTE: This guide is prepared and the steps are verified on the following server.

**Host:** CCAM UConn Health

**Server Name:** cloud-copasi-new

**Server address:** cloud-copasi-new.cam.uchc.edu

**Requirements**

1. Linux CentOS 7
2. Python 3.6
3. Django 3.x
4. Boto - Python interface to AWS (current version 2.49.0)
5. Cycler (current version 0.10.0)
6. Django-extensions (current version 3.0.8)
7. LXML (to process xml and html files in python) (current version 4.5.2)
8. matplotlib (current version 3.2.1)
9. Psycopgy2 – It is the most popular PostgreSQL database adapter for python programming language. (current version 2.8.5)
10. Python-dateutil – A built-in date time module which is used for manipulating dates and times from simple to complex ways. (current version 2.8.1)
11. subprocess32 – allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. (current version 3.5.4)
12. typing - Type Hints for Python (current version 3.7.4.3)

Pre installations

sudo yum update

sudo yum install yum-utils

sudo yum groupinstall development

sudo yum install epel-release

sudo yum install python-pip

sudo yum install python36

**Installation**

**User Setup**

For security reasons, Cloud-COPASI should run as its own user.

adduser cloudcopasi

su - cloudcopasi

**Creating a virtual environment “venv” and enabling it**

python3 -m venv venv

source venv/bin/activate

**Upgrading pip**

pip install --upgrade pip

**Installation of dependencies**

pip3 install django

pip3 install boto

pip3 install cycler

pip3 install django-extensions

pip3 install lxml

pip3 install matplotlib

python3 -m pip install psycopg2-binary

pip3 install subprocess32

pip3 install typing

**Installation of Database**

POSTGRESQL v13, Red Hat Enterprise, CentOS, Scientific or Oracle version 7

sudo yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm

sudo yum install -y postgresql13-server

Due to policies for Red Hat family distributions, the PostgreSQL installation will not be enabled for automatic start or have the database initialized automatically. To make your database installation complete, you need to perform the following steps, based on your distribution. For CentOS 7,

sudo postgresql-setup --initdb

sudo systemctl enable postgresql.service

sudo systemctl start postgresql.service

**Setting-up the database**

sudo -i -u postgres

psql

CREATE DATABASE cloud\_copasi\_db\_New;

CREATE USER cloud\_copasi\_user\_new WITH PASSWORD 'password';

ALTER ROLE cloud\_copasi\_user\_new SET client\_encoding TO 'utf8';

ALTER ROLE cloud\_copasi\_user\_new SET default\_transaction\_isolation TO 'read committed';

ALTER ROLE cloud\_copasi\_user\_new SET timezone TO 'UTC';

GRANT ALL PRIVILEGES ON DATABASE cloud\_copasi\_db\_New TO cloud\_copasi\_user\_new;

\q

logout

**Downloading and installing HT Condor**

sudo wget <http://parrot.cs.wisc.edu//symlink/20200806145602/8/8.9/8.9.8/09473020f227ee0f44174e6d7b0ae8c0/condor-8.9.8-x86_64_CentOS7-stripped.tar.gz>

tar -xvzf condor-8.9.8-x86\_64\_CentOS7-stripped.tar.gz

cd condor-8.9.8-x86\_64\_CentOS7-stripped

./bosco\_install

This, by default, appears to install bosco into the cloudcopasi user's home directory (~/bosco). This is the default expected location in settings.py.EXAMPLE.

It looks like Bosco needs to have some environment variables loaded, and to be started, if you want to test.

source bosco/bosco\_setenv

bosco\_start

**FOR MAC**

wget <https://research.cs.wisc.edu/htcondor/tarball/8.9/8.9.8/release/condor-8.9.8-x86_64_MacOSX-stripped.tar.gz>

same steps as mentioned above

**Downloading and installing COPASI**

mkdir -p copasi/bin

cd copasi

sudo wget https://github.com/copasi/COPASI/releases/download/Build-228/COPASI-4.29.228-AllSE.tar.gz

tar -xzvf COPASI-4.29.228-AllSE.tar.gz

cd bin

ln -s ../COPASI-4.29.228-AllSE/Linux64/CopasiSE

cd

**Creating folders to store log, user files and ssh keypairs**

mkdir log user-files instance\_keypairs

**Clone the new repository of Cloud Copasi**

git clone <https://github.com/hasanbaig/CloudCOPASI-New>

Copy settings.py.EXAMPLE to settings.py and fill in the details for the database, file locations. Eg.

cd cloud\_copasi/cloud\_copasi

cp settings.py.EXAMPLE settings.py

nano settings.py

cd

Copy the Copasi model file used in the cluster test into the copasi directory.

cp cloud-copasi/brusselator\_scan\_test.cps copasi/

Replace the file in your bosco installation (~/bosco/bin/bosco\_cluster) with the customised file in cloud-copasi/bosco, and ensure that the copasise variable points to the location of the CopasiSE binary.

cd bosco/bin

mv bosco\_cluster bosco\_cluster.orig

cp ../../cloud-copasi/bosco/bosco\_cluster .

cd

Add the Cloud-COPASI source folder to the python path, and set the Django setting module:

export PYTHONPATH=$PYTHONPATH:/home/cloudcopasi/cloud-copasi/

export DJANGO\_SETTINGS\_MODULE=cloud\_copasi.cloud\_copasi.settings

While in the the ~/cloud-copasi/cloud\_copasi directory run migrate, to create/update the Django website's data "model" in the database.

python manage.py migrate

And create the static files directory

python manage.py collectstatic

**Installation of Webserver (Apache)**

sudo yum install -y httpd httpd-devel

Downloading, compiling and installing a mod\_wsgi package by doing this:

$ cd /opt

$ sudo wget https://files.pythonhosted.org/packages/25/d8/1df4ba9c051cd88e02971814f0867274a8ac821baf983b6778dacd6e31f7/mod\_wsgi-4.6.8.tar.gz

$ sudo tar -zxvf mod\_wsgi-4.6.8.tar.gz

$ cd mod\_wsgi-4.6.8

$ sudo ./configure --with-apxs=/bin/apxs --with-python=/usr/local/bin/python3.8

# where path to apxs and python3.8 can be determined using “which apxs” and “which #python3.8” respectively

$ sudo LD\_RUN\_PATH=/usr/local/lib make

$ cd /etc/

Add a new entry to /etc/ld.so.conf

$ cd /etc/

$ sudo nano ld.so.conf

#Add the following lines in this file

include ld.so.conf.d/\*.conf

include /usr/local/lib

Verify if mod\_wsgi is linked properly

$ ldd /usr/lib64/httpd/modules/mod\_wsgi.so

Configure the httpd site config by adding a new file (cloud-copasi.conf) to /etc/httpd/conf.d/:

$ sudo nano /etc/httpd/conf.d/

Add the following configuration to this file

<VirtualHost \*:80>

ServerName cloud-copasi-new.cam.uchc.edu

ServerAlias cloudcopasi

DocumentRoot /home/cloudcopasi/cloud-copasi-new/

Alias /static /home/cloudcopasi/cloud-copasi-new/cloud\_copasi/web\_interface/templates/static-all/

<Directory /home/cloudcopasi/cloud-copasi-new/cloud\_copasi/web\_interface/templates/static-all/>

Options FollowSymLinks

Order allow,deny

Allow from all

Require all granted

</Directory>

Alias /admin/static /home/cloudcopasi/cloud-copasi-new/cloud\_copasi/web\_interface/templates/static-all/admin-media/

<Directory /home/cloudcopasi/cloud-copasi-new/cloud\_copasi/web\_interface/templates/static-all/admin-media/>

Options FollowSymLinks

Order allow,deny

Allow from all

Require all granted

</Directory>

#WSGIPassAuthorization On

WSGIDaemonProcess cloud-copasi-new user=cloudcopasi group=cloudcopasi threads=5 python-path=/home/cloudcopasi/cloud-copasi-new/ python-home=/home/cloudcopasi/cloud-copasi-new/ccEnv/

WSGIProcessGroup cloud-copasi-new

WSGIScriptAlias / /home/cloudcopasi/cloud-copasi-new/cloud\_copasi/cloud\_copasi/wsgi.py

<Directory /home/cloudcopasi/cloud-copasi-new/cloud\_copasi/>

Require all granted

</Directory>

ErrorLog /var/log/httpd/error.log

# Possible values include: debug, info, notice, warn, error, crit,

# alert, emerg.

LogLevel debug

CustomLog /var/log/httpd/access.log combined

</VirtualHost>

Now, start the httpd server

sudo systemctl start httpd

NOTE: for debugging, you can check the error log at   
sudo nano /var/log/httpd/error.log

BACKGROUND DAEMON Installation (see the old deployment guide on github)