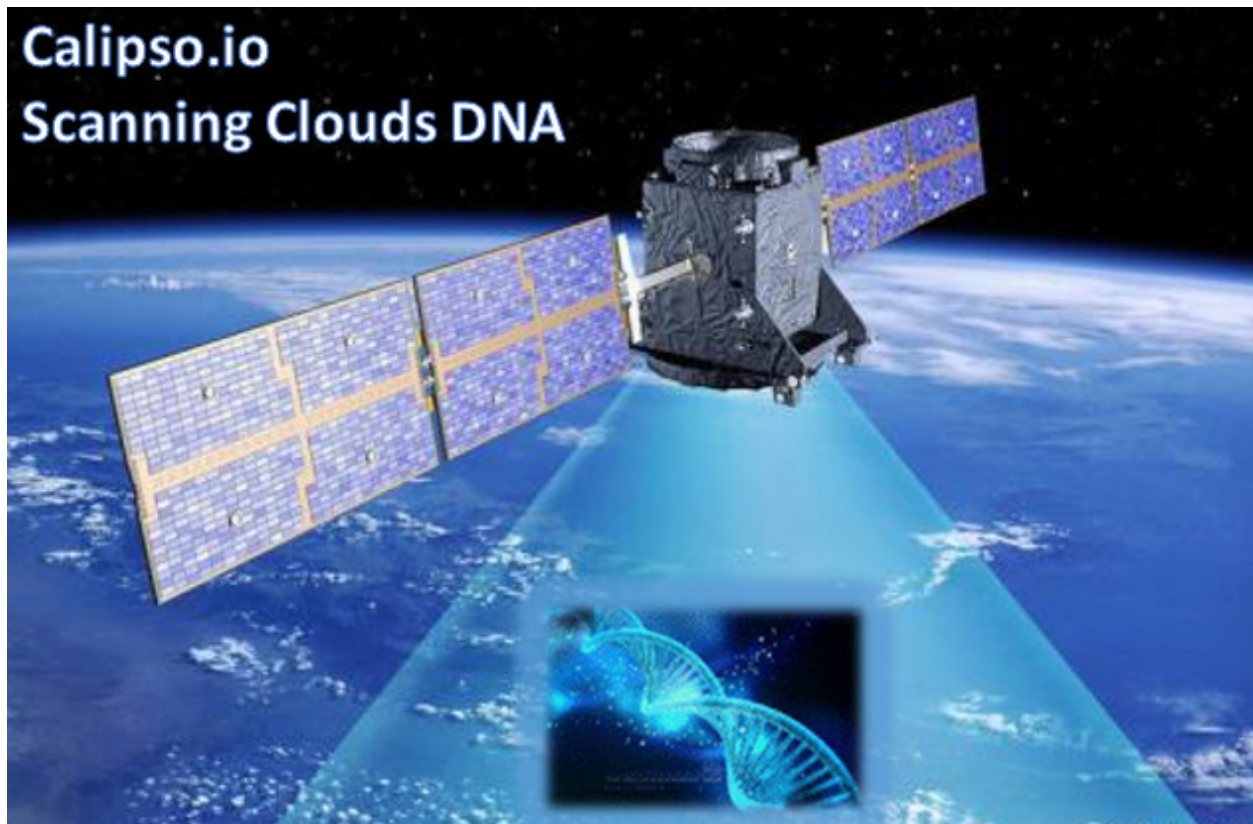


Calipso.io Installation Guide



Project “Calipso” tries to illuminate complex virtual networking with real time operational state visibility for large and highly distributed Virtual Infrastructure Management (VIM).

We believe that Stability is driven by accurate Visibility.

Calipso provides visible insights using smart discovery and virtual topological representation in graphs, with monitoring per object in the graph inventory to reduce error vectors and troubleshooting, maintenance cycles for VIM operators and administrators.

Table of Contents

Calipso.io Installation Guide	1
1 Pre Requisites.....	3
1.1 Pre Requisites for Calipso main application.....	3
1.2 Pre Requisites for Calipso UI application	3
2 Installation Options	4
2.1 Monolithic App.....	4
2.2 Micro Services App, single line install.....	4
2.3 Micro Services App, customized single line install	5
2.4 Micro Services App, customized interactive install	6
3 OPNFV Options	6
3.1 APEX scenarios	6
3.2 Fuel scenarios	6

1 Pre Requisites

1.1 Pre Requisites for Calipso main application

Calipso's main application is written with Python3.5 for Linux Servers, tested successfully on Centos 7.3 and Ubuntu 16.04. Here is a list of required software packages needed by the main scanning engines and the official steps required to install them:

1. Python3.5.x for Linux : <https://docs.python.org/3.5/using/unix.html#on-linux>
2. Pip for Python3 : <https://docs.python.org/3/installing/index.html>
3. Python3 packages :
 - sudo pip3 install falcon (>1.1.0)**
 - sudo pip3 install pymongo (>3.4.0)**
 - sudo pip3 install gunicorn (>19.6.0)**
 - sudo pip3 install ldap3 (>2.1.1)**
 - sudo pip3 install setuptools (>34.3.2)**
 - sudo pip3 install python3-dateutil (>2.5.3-2)**
 - sudo pip3 install bcrypt (>3.1.1)**
 - sudo pip3 install bson**
 - sudo pip3 install websocket**
 - sudo pip3 install datetime**
 - sudo pip3 install typing**
 - sudo pip3 install kombu**
 - sudo pip3 install boltons**
 - sudo pip3 install paramiko**
 - sudo pip3 install requests**
 - sudo pip3 install httpplib2**
 - sudo pip3 install mysql.connector**
 - sudo pip3 install xmltodict**
 - sudo pip3 install cryptography**
 - sudo pip3 install docker**
4. Git : <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
5. Docker : <https://docs.docker.com/engine/installation/>

1.2 Pre Requisites for Calipso UI application

Calipso UI is developed and maintained using Meteor Framework (<https://www.meteor.com/tutorials>). For stability and manageability reasons we decided to always build the latest Calipso UI as a Docker container pre-parameterized for stable behavior. The required steps for installing the Calipso UI with several options are listed below.

2 Installation Options

2.1 Monolithic App

For faster development use, one might require Calipso to be installed as a Monolithic Application, to do that all you need is a server installed with Calipso pre-requisites and cloning of Calipso's public repository, here are the required steps for this option:

1. Create a user named '**calipso**' and give it **sudo** access, login as '**calipso**' user
2. Create those directories as the '**calipso**' user : **mkdir -p log & mkdir log/calipso**
3. Clone calipso main application from current public repository:
git clone https://git.opnfv.org/calipso/
4. Move to the default install directory: **cd calipso**
5. Setup Python3 environment for calipso:
export PYTHONPATH=/home/calipso/calipso/app
6. Follow quick-start guide on how to use calipso modules for monolithic scenarios, and run each module manually.

2.2 Micro Services App, single line install

For most users, this will be the fastest and more reliable install option. We currently have Calipso divided into 7 major containers, those are installed using a single installer. The Calipso containers are pre-packaged and fully customized per our design needs. Here are the required steps for installation using this option:

1. Follow steps 1- 4 per section 2.1 above.
2. Although Calipso installer can download all needed containers, if they doesn't exist locally, we recommend doing this download manually for better control:
sudo docker login # use your DockerHub username and password to login.
sudo docker pull korenlev/calipso:scan # scan container used to scan VIM
sudo docker pull korenlev/calipso:listen # listen container to attach to VIM's BUS.
sudo docker pull korenlev/calipso:api # api container for application integration
sudo docker pull korenlev/calipso:sensu # sensu server container for monitoring
sudo docker pull korenlev/calipso:mongo # calipso mongo container
sudo docker pull korenlev/calipso:ui # downloads calipso ui container

sudo docker pull korenlev/calipso:ldap # downloads calipso ldap container

3. Check that all containers were downloaded and registered successfully:

sudo docker images

Expected results (As of Aug 2017):

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
korenlev/calipso	listen	12086aaedbc3	6 hours ago	1.05GB
korenlev/calipso	api	34c4c6c1b03e	6 hours ago	992MB
korenlev/calipso	scan	1ee60c4e61d5	6 hours ago	1.1GB
korenlev/calipso	sensu	a8a17168197a	6 hours ago	1.65GB
korenlev/calipso	mongo	17f2d62f4445	22 hours ago	1.31GB
korenlev/calipso	ui	ab37b366e812	11 days ago	270MB
korenlev/calipso	ldap	316bc94b25ad	2 months ago	269MB

4. Run the calipso installer using single line arguments:

python3 calipso/app/install/calipso-installer.py--command start-all --copy q

This should launch all calipso modules in sequence along with all needed configuration files in /home/calipso.

2.3 Micro Services App, customized single line install

Calipso app includes the following directory in its default structure (as of Aug 2017): **app/install/db**, this directory holds the initial Database scheme and files needed as an initial data for starting Calipso application.

Calipso Database container (calipso-mongo) comes pre-packaged with the necessary initial scheme and files, but in some development cases might not be synchronized with the latest ones supported. For this reason, the installer has an option to copy files from the above directory into the Database after runtime.

You can run calipso installer using the following single line arguments:

1. **--command start-all | stop-all**

This will either start (docker run) or stop (docker kill and remove) all calipso containers, a mandatory attribute for a single line install option.

2. **--copy q | c**

This will either copy all files from app/install/db into mongoDB or skip that step, a mandatory attribute for a single line install option.

3. **--hostname**

Allows to enter an IP address or hostname where container will be deployed, an optional argument, default IP 172.17.0.1 (docker0 default) is deployed if not used.

4. **--webport**

Allows to enter a TCP port to be used for calipso UI on the host, an optional argument, default 80 (http default) is deployed if not used.

5. **--dbport**

Allows to enter a TCP port to be used for mongoDB port on the host, an optional argument, default 27017 (mongo default) is deployed if not used.

6. --dbuser

Allows to enter a username to be used for mongoDB access on the host, an optional argument, default 'calipso' (calipso-mongo container's default) is deployed if not used.

7. --dbpassword

Allows to enter a password to be used for mongoDB access on the host, an optional argument, default 'calipso_default' (calipso-mongo container's default) is deployed if not used.

2.4 Micro Services App, customized interactive install

Calipso's application containers can be initiated and stopped individually for testing purposes, this option is available through interactive install, run calipso-installer.py with no argument to kickstart the interactive process, with the following steps:

1. **Action? (stop, start, or 'q' to quit):**
2. **Container? (all, calipso-mongo, calipso-scan, calipso-listen, calipso-ldap, calipso-api, calipso-sensu, calipso-ui or 'q' to quit):**
3. **create initial calipso DB ? (copy json files from 'db' folder to mongoDB - 'c' to copy, 'q' to skip):**

Note: based on the arguments input (or defaults), calipso installer automatically creates and place 2 configuration files under `/home/calipso`: `ldap.conf` and `calipso_mongo_access.conf`, those are mandatory configuration files used by calipso containers to interact with each other.

3 OPNFV Options

Although calipso is designed for any VIM and for enterprise too, service providers may use additional capability to install calipso

3.1 APEX scenarios

When using apex to install OPNFV, the Triple-O based OpenStack is installed automatically and calipso installation can be initiated automatically after apex completes the VIM installation process for a certain scenario.

In this case calipso-configuration.py can be used for creating a new environment automatically into mongoDB and UI of Calipso (instead of using the calipso UI to do that as typical user would do), then detailed scanning can start immediately, the following options are available for calipso-configuration.py:

3.2 Fuel scenarios

