
Nephos Documentation

Release 0.3.4

Alejandro (Sasha) Vicente Grabovetsky

Mar 25, 2019

Contents:

1	Nephos	3
1.1	Prerequisites	4
1.2	Installation	4
1.3	Testing	5
1.4	Usage	5
1.5	Examples	5
1.6	Further information	6
2	Indices and tables	9

CHAPTER 1

Nephos

Library to deploy [Hyperledger Fabric](#) projects to [Kubernetes](#)

Source resides at <https://github.com/hyperledger-labs/nephos>, originally developed at ****AID:Tech****

Documentation resides at <https://nephos.readthedocs.io>

- *Prerequisites*
- *Installation*
 - *Pip*
 - *Git repository*
 - * *Virtual environment*
 - * *Requirements*
- *Testing*
 - *Unit tests*
- *Usage*
- *Examples*
 - *Development*
 - *QA and Production*
 - *Minikube*
- *Further information*
 - *Helm charts*
 - *Educational material*

1.1 Prerequisites

This library requires an existing Kubernetes cluster.

For best results, use a real cluster (e.g. on a cloud like AWS, GCP, Azure, IBM Cloud, etc.). However, you may also use [Minikube](#).

Either way, you will need to have the following tools installed:

- [python 3.7.0](#) or above
- [kubectl](#)
- [helm](#)

1.2 Installation

1.2.1 Pip

You can install nephos from [PyPI](#) by running:

```
pip install nephos
```

1.2.2 Git repository

You can also download the git repository with:

```
git clone https://github.com/hyperledger-labs/nephos.git
```

And work locally by installing the following:

Virtual environment

This library currently only supports Python 3:

```
python3 -m venv ./venv  
source ./venv/bin/activate
```

Requirements

All python-related requirments are held in the requirements.txt file

```
pip install -r requirements.txt
```

You will also need to install an initialise [helm](#) for Kubernetes, as described [here](#)

Furthermore, you will need the Hyperledger Fabric utility binaries that can be installed with this [script](#) on Linux, or via Homebrew for the [Fabric tools](#) and [CA tools](#) on Mac OS X.

1.3 Testing

1.3.1 Unit tests

Once you have all requirements installed, all the unit tests should pass and provide full coverage:

```
PYTHONPATH=. pytest --ignore=./integration --cov=. --cov-report xml:coverage.xml --
↳cov-report term-missing
```

The integration tests should also pass:

```
PYTHONPATH=. pytest -x -s ./integration
```

1.4 Usage

To use *nephos*, run the `deploy.py` executable CLI script.

For instance, you can see available commands/options by running:

```
PYTHONPATH=. ./nephos/deploy.py --help
```

To install a full end-to-end fabric network, you can run:

```
PYTHONPATH=. ./nephos/deploy.py -f ./PATH_TO_YOUR_SETTINGS/file.yaml fabric
```

You can also upgrade a network:

```
PYTHONPATH=. ./nephos/deploy.py --upgrade -f ./PATH_TO_YOUR_SETTINGS/file.yaml fabric
```

1.5 Examples

1.5.1 Development

Example of development/QA/production(-ish) networks are provided in the examples folder.

To run the dev example from the git repository, use this command:

```
./nephos/deploy.py --verbose -f ./examples/dev/nephos_config.yaml fabric
```

Note: The `nephos_config.yaml` is by default set to point to the `minikube` context (even for the `prod` example) to prevent accidental deployments to production clusters. If your K8S context name is different, please update this file.

1.5.2 QA and Production

For the QA and production examples, you will need to replace the CA hostname to one pointing to your K8S cluster Ingress Controller (e.g. NGINX or Traefik) IP address.

In a real cluster, you will wish to install an ingress controller and a certificate manager. We include in the repository two example Cluster Issuers (you will need to modify the email field in them) for the `cert-manager` deployment:

```
helm install stable/nginx-ingress -n nginx-ingress --namespace ingress-controller
helm install stable/cert-manager -n cert-manager --namespace cert-manager
kubectl create -f ./examples/certManagerCI_staging.yaml
kubectl create -f ./examples/certManagerCI_production.yaml
```

To use the Composer examples, you will need a Cloud system capable of a “ReadWriteMany” policy (e.g. “azurefile” on Azure).

1.5.3 Minikube

Given that we may wish to test locally on Minikube, we will need to use a local ingress controller and ignore cert-manager in favour of self-cooked SSL certificates.

In `./examples` we include the `ca-nephos-local.*` self-signed certificates, created with OpenSSL as follows:

```
openssl req -new -newkey rsa:4096 -days 3650 -nodes -x509 -subj "/C=IE/ST=Dublin/
↪L=Dublin/O=AID:Tech/CN=ca.nephos.local" -keyout ca-nephos-local.key -out ca-nephos-
↪local.crt

openssl x509 -in ca-nephos-local.crt -out ca-nephos-local.pem -outform PEM

kubectl create ns cas

kubectl -n cas create secret tls ca--tls --cert=ca-nephos-local.crt --key=ca-nephos-
↪local.key
```

We can save them to the `cas` namespace as follows

```
cd ./examples

kubectl create ns cas

kubectl -n cas create secret tls ca--tls --cert=ca-nephos-local.crt --key=ca-nephos-
↪local.key
```

We can then enable the ingress on minikube and update `/etc/hosts` with the IP of minikube:

```
minikube addons enable ingress

echo "$(minikube ip) ca.nephos.local" | sudo tee -a /etc/hosts
```

1.6 Further information

For more information on how to deploy Hyperledger Fabric to Kubernetes, please see the following resources:

1.6.1 Helm charts

We have released a set of Helm Charts, currently living in two locations:

- The official Helm Chart [repository](#) and [KubeApps](#).

- The AID:Tech Helm Chart [repository](#).

1.6.2 Educational material

A [workshop](#) on the Hyperledger Global Forum, featuring [slides](#) and a [part 1](#) and [part 2](#) videos.

A(n older) [webinar](#) on deploying Hyperledger Fabric on Kubernetes.

We have also contributed the Composer chapter on the EdX course [Blockchain for Business](#).

And we have also released a course on [Packt](#), [Udemy](#) and [O'Reilly](#) called Hyperledger for Blockchain Applications.

CHAPTER 2

Indices and tables

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