

How to generate artefacts for SDC catalog using Tosca Lab Tool

This page describes how to generate artefacts for the SDC catalog using the Tosca Lab Tool. These artefacts are required for on-boarding a microservice (see [MicroServices Onboarding in ONAP](#)).

The steps below are carried out using the 3GPP PM Mapper's (DCAE microservice) component spec as input.

```
# Checkout the tosca lab repo:
root@node:~/tosca_lab# git clone https://gerrit.onap.org/r/sdc/dcae-d/tosca-lab
Cloning into 'tosca-lab'...
remote: Counting objects: 2, done
remote: Finding sources: 100% (2/2)
remote: Total 292 (delta 0), reused 292 (delta 0)
Receiving objects: 100% (292/292), 162.28 KiB | 0 bytes/s, done.
Resolving deltas: 100% (104/104), done.
Checking connectivity... done.
root@node:~/tosca_lab# ls
tosca-lab

# Change directory to app:
root@node:~/tosca_lab# cd tosca-lab/
root@node:~/tosca_lab/tosca-lab# ls
INFO.yaml LICENSE.TXT app docker mvn-phase-script.sh nginx pom.xml scripts simple_compose_stdin.py tox.
ini version.properties
root@node:~/tosca_lab/tosca-lab# cd app/
root@node:~/tosca_lab/tosca-lab/app# ls
README.md __init__.py data map_sup_enrich_compose.py model_create.py policy_create.py requirements.
txt simple_compose.py tests tosca_server.py toscalib version.py web

# Create tosca model artifacts from component spec:
root@node:~/tosca_lab/tosca-lab/app# python model_create.py -h
OPTIONS:
  -h|--help: print this help message
  -i|--input: The PATH to spec file
  -o|--output: the folder for the output model
  -n|--name: the name of the service
  -t|--import: the PATH to import file
  -m|--meta: the PATH to meta model file (default: ./data/meta_model/meta_tosca_schema.yaml)
root@node:~/tosca_lab/tosca-lab/app# python model_create.py -i /home/ubuntu/tosca-lab/app/pmmapper-component-
spec.json -o /home/ubuntu/tosca-model/
WARNING:root:Interface namecloudify.interfaces.lifecyclehas been definend in parenty type, overwritten here

# Create policy model artifact from component spec:
root@node:~/tosca_lab/tosca-lab/app# python policy_create.py -i /home/ubuntu/tosca-lab/app/pmmapper-component-
spec.json -o /home/ubuntu/tosca-model/policy.yaml

# Verify created tosca model artifacts :
root@node:~/tosca-model# cd /home/ubuntu/tosca-model/
root@node:~/tosca-model# ls
pmmapper-component-spec.json schema.yaml template.yaml translate.yaml
```

Troubleshooting:

If the Python YAML module is not installed (i.e you get the error message *ImportError: No module named yaml* when executing the *model_create.py* command) you will need to install PyYAML

You may first need to install pip # **apt install python-pip**

```
root@node:~/tosca_lab/tosca-lab/app# pip install PyYAML
The directory '/home/ubuntu/.cache/pip/http' or its parent directory is not owned by the current user and the
cache has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo,
you may want sudo's -H flag.
The directory '/home/ubuntu/.cache/pip' or its parent directory is not owned by the current user and caching
wheels has been disabled. check the permissions and owner of that directory. If executing pip with sudo, you
may want sudo's -H flag.
Collecting PyYAML
  Downloading https://files.pythonhosted.org/packages/9e/a3
/1d13970c3f36777c583f136c136f804d70f500168edc1e6daa6daa7200769/PyYAML-3.13.tar.gz (270kB)
    100% |aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa| 276kB 13.5MB/s
Installing collected packages: PyYAML
  Running setup.py install for PyYAML ... done
Successfully installed PyYAML-3.13
You are using pip version 18.1, however version 19.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

Input and Output Files

[schema.yaml](#) [template.yaml](#) [translate.yaml](#) [pmmapper-component-spec.json](#)