Control System Management and Deployment at MAX IV

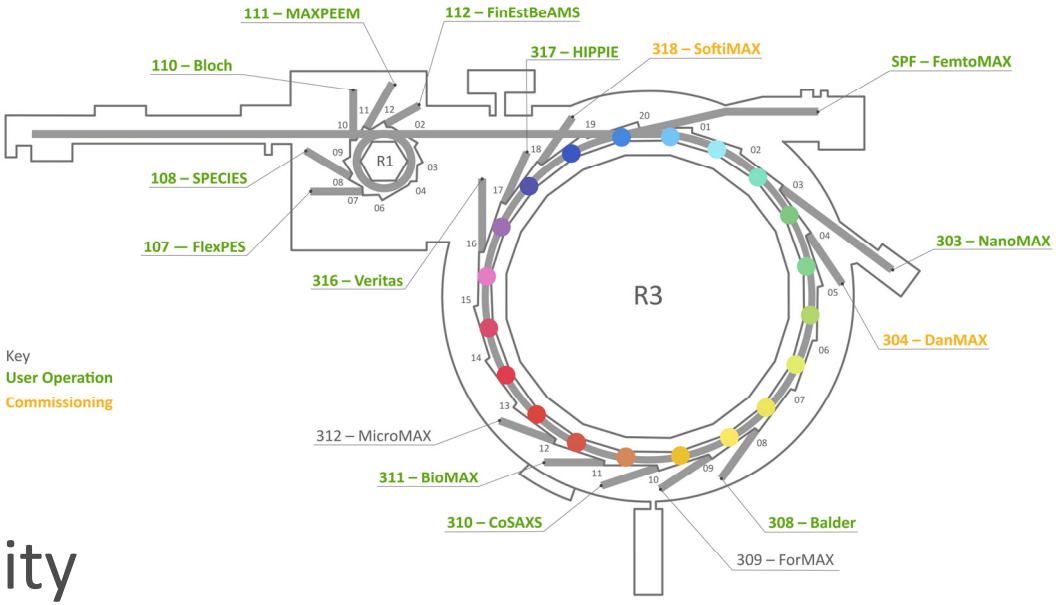
ICALEPCS 2021 – THBL01

Benjamin Bertrand

Outline

- Introduction
- RPM
- Conda
- Ansible

Introduction



- Synchrotron facility
- 16 beamlines
- Control System based on Tango
- Hundreds of Virtual and Physical machines


rpm

RPM

RPM Package Manager (originally Red Hat Package Manager)

- Red Hat Enterprise Linux
- Fedora
- **CentOS**
- Oracle Linux

Large number of high quality packages (CentOS repository and EPEL)

RPM SPEC file

```
Summary:    Tango device for Linkam T96 heater
Name:       tangods-linkamt96
Version:    1.2.0
Release:    1%{?dist}.maxlab
License:    GPL
URL:        http://www.maxiv.lu.se
Source:     %{name}-%{version}.tar.gz
Requires:   lib-maxiv-linkam-t96
Requires:   linkam-sdk
Requires:   lib-maxiv-common-cpp >= 4.0.0
Requires:   libtango9
BuildRequires: lib-maxiv-linkam-t96-devel
BuildRequires: linkam-sdk-devel
BuildRequires: lib-maxiv-common-cpp-devel >= 4.0.0
BuildRequires: libtango9-devel
# for pogo Makefile templates:
BuildRequires: tango-java

%description
Tango device for Linkam T96 heater

%prep
%setup -q
```

```
%build
make

%install
[ -z %{buildroot} ] || rm -rf %{buildroot}

# install bins
pushd bin > /dev/null
for f in *; do
    install -D -m755 $f %{buildroot}%{_bindir}/$f
done
popd > /dev/null

%files
%defattr (-,root,root,755)
%{_bindir}/*
```

C++ GitLab CI

```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.cpp-ci.yml'
```

C++ GitLab CI

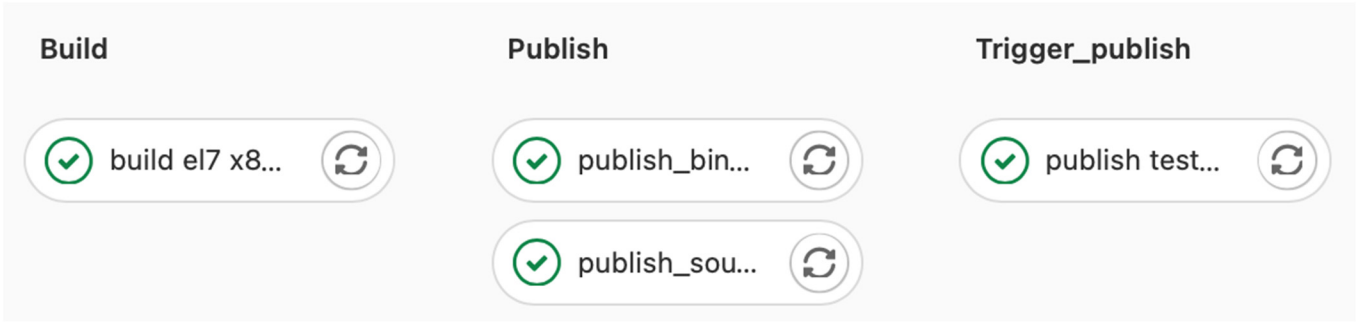
```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.cpp-ci.yml'
```

```yaml
.build-template:
  stage: build
  image: harbor.maxiv.lu.se/kits-sw/centos-build:7
  tags: [docker-privileged]
  script:
    - echo "This is a build stage for $RELEASE/$DIST/$ARCH"
    - |
      export PROFILE=maxlab-${RELEASE}-${DIST}-${ARCH}

      echo "Make build for $PROFILE"
      maxpkg --dist=${DIST} sources --create-archive
      maxpkg --dist=${DIST} mockbuild --root=$PROFILE
```

C++ GitLab CI

```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.cpp-ci.yml'
```

```yaml
.build-template:
  stage: build
  image: harbor.maxiv.lu.se/kits-sw/centos-build:7
  tags: [docker-privileged]
  script:
    - echo "This is a build stage for $RELEASE/$DIST/$ARCH"
    - |
      export PROFILE=maxlab-${RELEASE}-${DIST}-${ARCH}

      echo "Make build for $PROFILE"
      maxpkg --dist=${DIST} sources --create-archive
      maxpkg --dist=${DIST} mockbuild --root=$PROFILE
```



Python: setup.py

```python
import io
from setuptools import setup

with io.open("README.md", "r", encoding="utf-8") as fh:
    long_description = fh.read()

tests_require = ["pytest", "pytest-cov", "Faker", "mock"]

setup(
    # Package
    name="dsconfig",
    version="1.6.0",
    author="KITS",
    author_email="kits-sw@maxiv.lu.se",
    url="https://gitlab.com/MaxIV/lib-maxiv-dsconfig",
    license="GPLv3",
    packages=["dsconfig", "dsconfig.appending_dict"],
    description="Library and utilities for Tango device configuration.",
    long_description=long_description,
    long_description_content_type="text/markdown",
    keywords="Tango",
    # Requirements
    python_requires=">=2.7",
    install_requires=["jsonpatch>=1.13", "jsonschema", "six", "xlrd", "pytango"],
    extras_require={"tests": tests_require},
    # Resources
    package_data={"dsconfig": ["schema/dsconfig.json", "schema/schema2.json"]},
    # Scripts
    entry_points={
        "console_scripts": [
            "xls2json = dsconfig.excel:main",
            "json2tango = dsconfig.json2tango:main",
        ]
    },
)
```

fpm

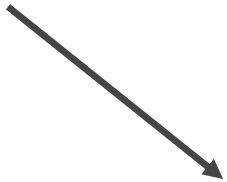
The goal of FPM is to be able to easily build platform-native packages (deb, rpm, etc)

```
fpm -s python -t rpm \
  --python-bin python3.6 \
  --rpm-dist el7 \
  -p results \
  --python-package-name-prefix python36 \
  --python-setup-py-arguments=--prefix=/usr \
  --no-python-downcase-dependencies \
  ${FPM_FLAGS} \
  --url $CI_PROJECT_URL --verbose .
```

fpm

The goal of FPM is to be able to easily build platform-native packages (deb, rpm, etc)

```
fpm -s python -t rpm \
  --python-bin python3.6 \
  --rpm-dist el7 \
  -p results \
  --python-package-name-prefix python36 \
  --python-setup-py-arguments=--prefix=/usr \
  --no-python-downcase-dependencies \
  ${FPM_FLAGS} \
  --url $CI_PROJECT_URL --verbose .
```



```
$ rpm -qp --requires python36-dsconfig-1.6.0-1.el7.noarch.rpm
python36-jsonpatch >= 1.13
python36-jsonschema
python36-six
python36-xlrd
python36-pytango
```

Python GitLab CI

```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.python-ci.yml'
```

Python GitLab CI

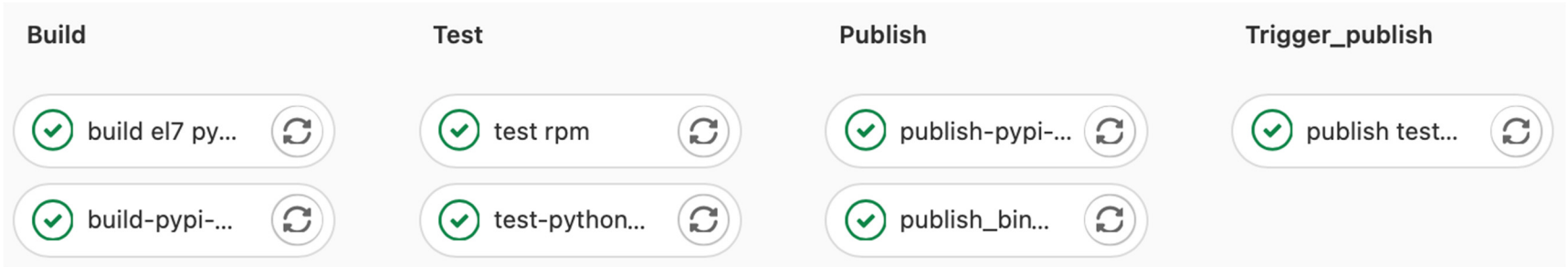
```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.python-ci.yml'
```

```yaml
.build-template:
  stage: build
  image: harbor.maxiv.lu.se/kits-sw/centos-build:7
  script:
    - echo "This is a build stage for $RELEASE/$DIST/$ARCH"
    - |
      export PROFILE=maxlab-${RELEASE}-${DIST}-${ARCH}

      echo "Make build for $PROFILE"
      mkdir -p results/$PROFILE
      if [ -z "${PACKAGE_NAME}" ]
      then
        fpm -s python -t rpm --python-bin ${PYTHON_VERSION} --rpm-dist ${DIST} -p results/${PROFILE}
--no-python-downcase-dependencies ${FPM_FLAGS} ${FPM_PREFIX} --url $CI_PROJECT_URL --verbose .
      else
        fpm -s python -t rpm --python-bin ${PYTHON_VERSION} --rpm-dist ${DIST} -p results/${PROFILE}
--no-python-downcase-dependencies --name ${PACKAGE_NAME} ${FPM_FLAGS} ${FPM_PREFIX} --url $CI_PROJECT_URL --verbose .
      fi
```

Python GitLab CI

```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.python-ci.yml'
```

```yaml
.build-template:
  stage: build
  image: harbor.maxiv.lu.se/kits-sw/centos-build:7
  script:
    - echo "This is a build stage for $RELEASE/$DIST/$ARCH"
    - |
      export PROFILE=maxlab-${RELEASE}-${DIST}-${ARCH}

      echo "Make build for $PROFILE"
      mkdir -p results/$PROFILE
      if [ -z "${PACKAGE_NAME}" ]
      then
        fpm -s python -t rpm --python-bin ${PYTHON_VERSION} --rpm-dist ${DIST} -p results/${PROFILE}
--no-python-downcase-dependencies ${FPM_FLAGS} ${FPM_PREFIX} --url $CI_PROJECT_URL --verbose .
      else
        fpm -s python -t rpm --python-bin ${PYTHON_VERSION} --rpm-dist ${DIST} -p results/${PROFILE}
--no-python-downcase-dependencies --name ${PACKAGE_NAME} ${FPM_FLAGS} ${FPM_PREFIX} --url $CI_PROJECT_URL --verbose .
      fi
```




CONDA

Conda

Package, dependency and environment management for any language – Python, R, Ruby, Lua, Scala, Java, Javascript, C/C++, FORTRAN

- Run on Windows, macOS and Linux
- Create isolated environments
- OS independent (conda Linux packages can be installed on any Linux distribution)
- Many packages available on conda-forge

https://conda.io

Quetz

Open source server for conda packages

- Local channels
- Proxy channels
- Mirror channels


```
channel_alias: https://conda.maxiv.lu.se/get
channels:
  - maxiv-kits
  - conda-forge
```

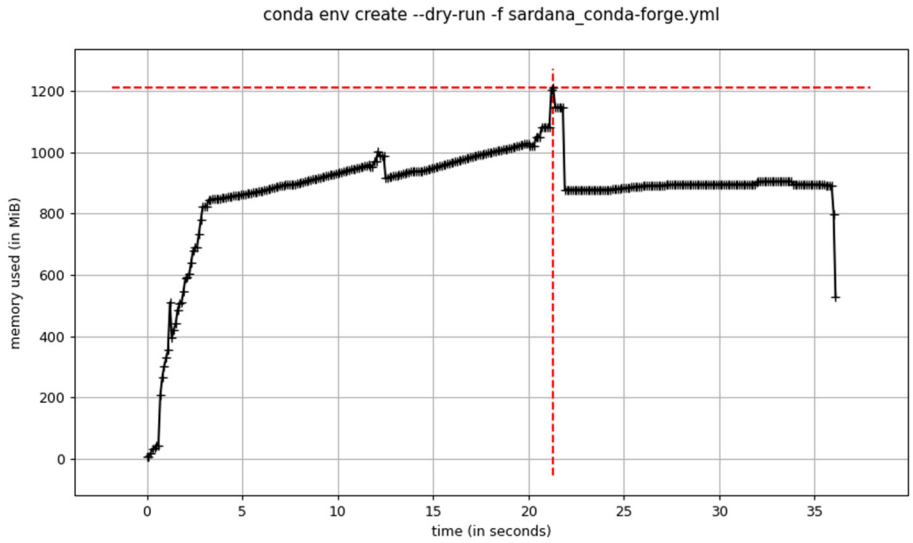
https://quetz.readthedocs.io

Conda performances

```yaml
name: sardana
channels:
  - conda-forge
dependencies:
  - python=3.9
  - sardana
```

Conda performances

```
name: sardana
channels:
  - conda-forge
dependencies:
  - python=3.9
  - sardana
```

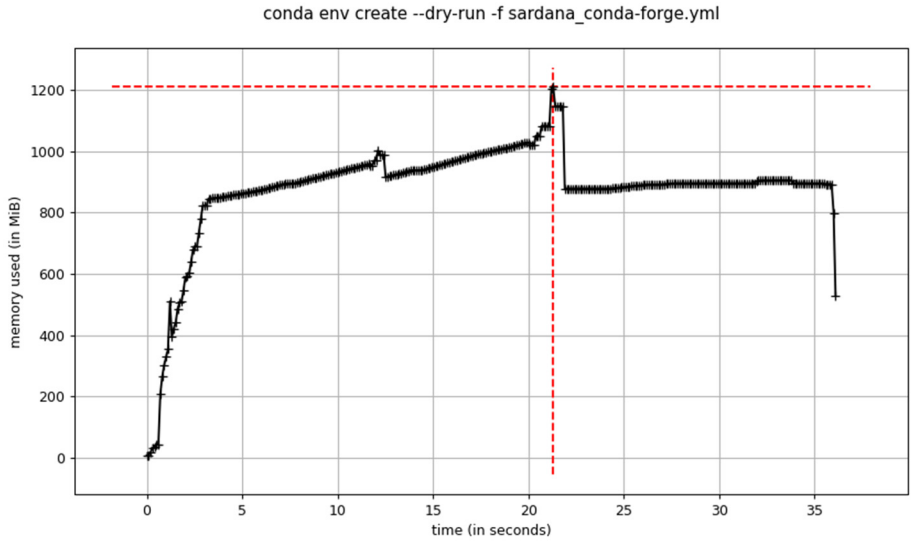


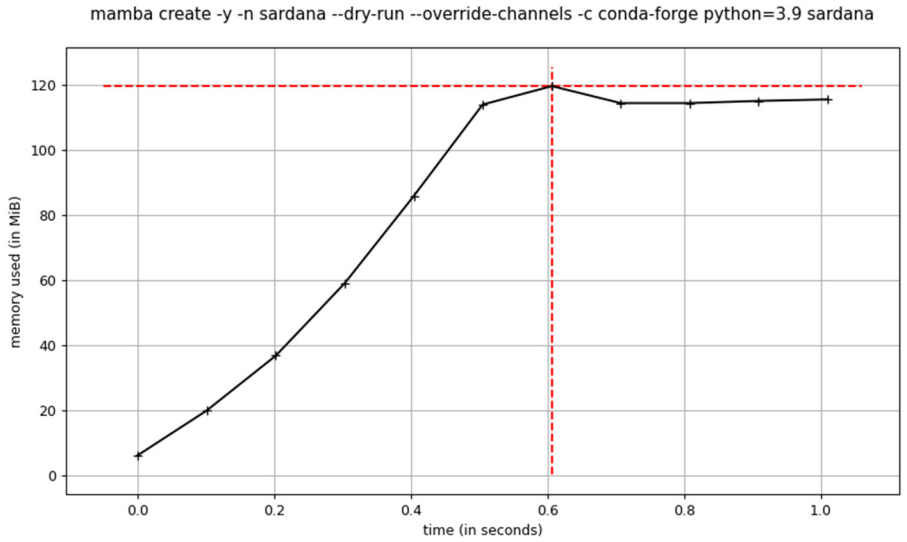
Conda performances

```
name: sardana
channels:
  - conda-forge
dependencies:
  - python=3.9
  - sardana
```

Mamba

Drop-in replacement for conda with some parts reimplemented in C++





MAX IV

mini-conda-forge channel

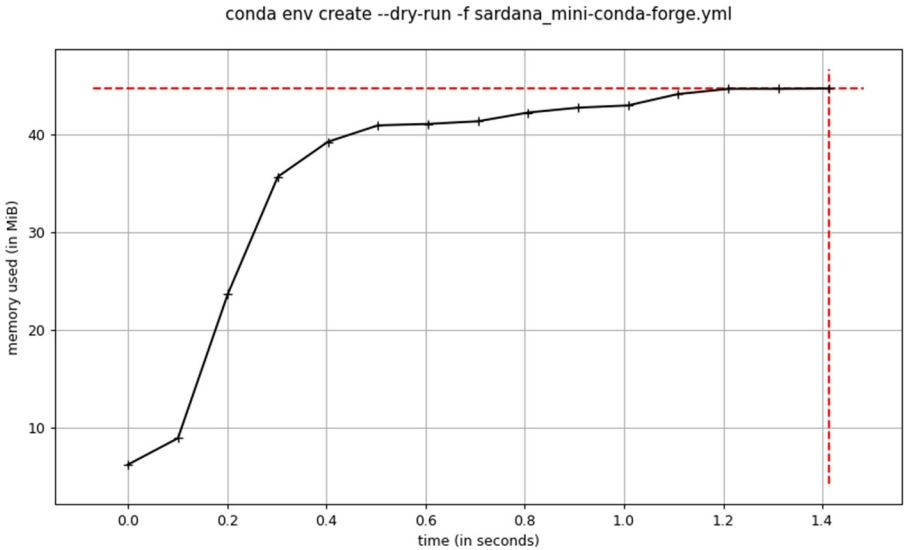
Local channel: small subset of conda-forge

```
name: sardana
channels:
  - mini-conda-forge
dependencies:
  - python=3.9
  - sardana
```

mini-conda-forge channel

Local channel: small subset of conda-forge

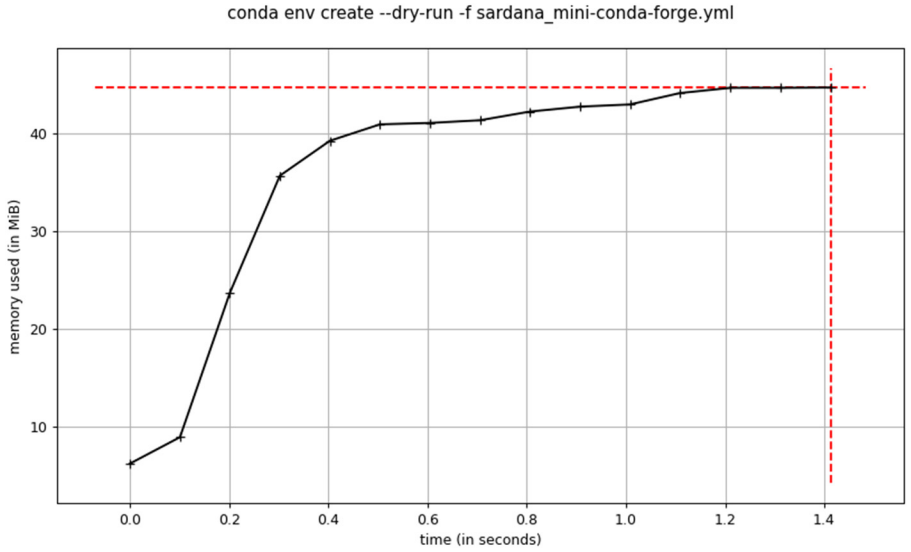
```
name: sardana
channels:
  - mini-conda-forge
dependencies:
  - python=3.9
  - sardana
```

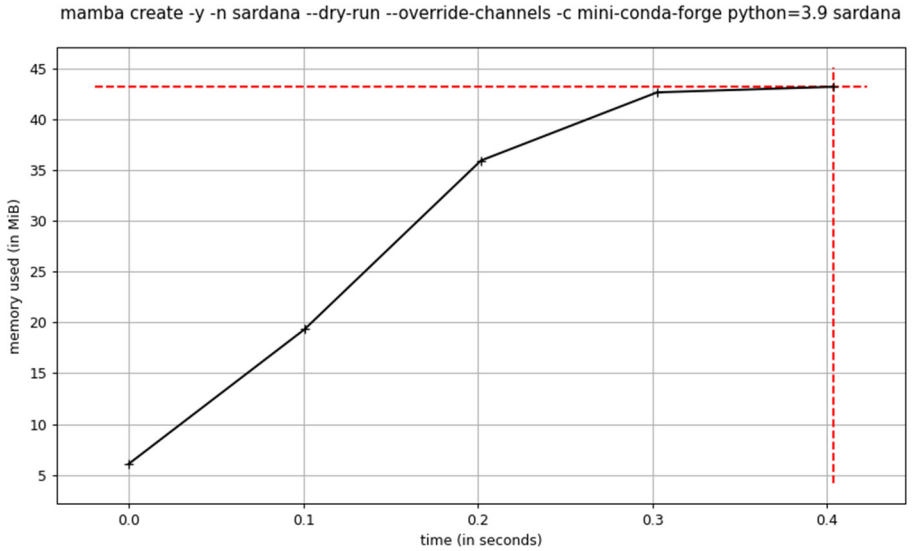


mini-conda-forge channel

Local channel: small subset of conda-forge

```
name: sardana
channels:
  - mini-conda-forge
dependencies:
  - python=3.9
  - sardana
```





Sync mini-conda-forge

```
$ cat maxiv_packages_specs.txt
# We want to be able to install conda and mamba
conda
mamba
# Default packages for a sardana environment
python=3.9 sardana bitshuffle matplotlib spyder pytango taurus taurus_pyqtgraph h5py hdf5plugin icepap sockio
...
# Tango packages
tango-test tango-database tango-admin jtango
```

Sync mini-conda-forge

```
$ cat maxiv_packages_specs.txt
# We want to be able to install conda and mamba
conda
mamba
# Default packages for a sardana environment
python=3.9 sardana bitshuffle matplotlib spyder pytango taurus taurus_pyqtgraph h5py hdf5plugin icepap sockio
...
# Tango packages
tango-test tango-database tango-admin jtango
```

```
$ sync-channel --file maxiv_package_specs.txt
Processing list of specs for linux-64
Checking existing packages for mini-conda-forge/linux-64
Downloading mini-conda-forge/noarch repodata
Downloading mini-conda-forge/linux-64 repodata
Processing 'conda' for linux-64
Run 'mamba create --dry-run --json -n xxx --override-channels -c https://conda.maxiv.lu.se/get/conda-forge conda' for linux-64
Processing 'mamba' for linux-64
Run 'mamba create --dry-run --json -n xxx --override-channels -c https://conda.maxiv.lu.se/get/conda-forge mamba' for linux-64
Processing 'python=3.9 sardana bitshuffle matplotlib spyder pytango taurus taurus_pyqtgraph h5py hdf5plugin icepap sockio' for linux-64
Run 'mamba create --dry-run --json -n xxx --override-channels -c https://conda.maxiv.lu.se/get/conda-forge python=3.9 sardana bitshuffle matplotlib spyder pytango taurus
taurus_pyqtgraph h5py hdf5plugin icepap sockio' for linux-64
...
Processing 'tango-test tango-database tango-admin jtango' for linux-64
Run 'mamba create --dry-run --json -n xxx --override-channels -c https://conda.maxiv.lu.se/get/conda-forge tango-test tango-database tango-admin jtango' for linux-64
8 new packages to upload for linux-64 list of specs
Downloading https://conda.maxiv.lu.se/get/conda-forge/linux-64/omniorb-4.2.4-py39hff7568b_5.tar.bz2
Downloading https://conda.maxiv.lu.se/get/conda-forge/linux-64/cpptango-9.3.4-hf7cf922_4.tar.bz2
Downloading https://conda.maxiv.lu.se/get/conda-forge/linux-64/omniorb-libs-4.2.4-h812cca2_5.tar.bz2
Downloading https://conda.maxiv.lu.se/get/conda-forge/linux-64/python-3.8.12-hb7a2778_2_cpython.tar.bz2
Downloading https://conda.maxiv.lu.se/get/conda-forge/linux-64/tango-test-3.3-h4bd325d_1.tar.bz2
Uploading cpptango-9.3.4-hf7cf922_4.tar.bz2 to mini-conda-forge
Uploading tango-test-3.3-h4bd325d_1.tar.bz2 to mini-conda-forge
Uploading omniorb-libs-4.2.4-h812cca2_5.tar.bz2 to mini-conda-forge
Uploading omniorb-4.2.4-py39hff7568b_5.tar.bz2 to mini-conda-forge
Downloading https://conda.maxiv.lu.se/get/conda-forge/linux-64/tango-admin-1.16-h4bd325d_1.tar.bz2
Downloading https://conda.maxiv.lu.se/get/conda-forge/linux-64/libnsl-2.0.0-h7f98852_0.tar.bz2
Uploading libnsl-2.0.0-h7f98852_0.tar.bz2 to mini-conda-forge
```

conda-forge recipe

```
{% set name = "dsconfig" %}
{% set version = "1.6.0" %}

package:
  name: {{ name|lower }}
  version: {{ version }}

source:
  url: https://pypi.io/packages/source/{{ name[0] }}/{{ name }}/dsconfig-{{ version }}.tar.gz
  sha256: e7709b0fa920b9a41460c0613ea2925aed089b8852cc8fa10ed2c3056ef82b56

build:
  number: 0
  noarch: python
  entry_points:
    - xls2json = dsconfig.excel:main
    - json2tango = dsconfig.json2tango:main
  script: {{ PYTHON }} -m pip install . -vv

requirements:
  host:
    - pip
    - python >=3.6
  run:
    - jsonpatch >=1.13
    - jsonschema
    - pytango
    - python >=3.6
    - six
    - xlrd

test:
  imports:
    - dsconfig
    - dsconfig.appending_dict
  commands:
    - pip check
    - xls2json --help
    - json2tango --help
  requires:
    - pip

about:
  home: https://gitlab.com/MaxIV/lib-maxiv-dsconfig
  summary: Library and utilities for Tango device configuration.
  license: GPL-3.0-or-later
  license_file: LICENSE.txt
  description: |
    This is a command line tool for managing configuration of Tango device servers.
    Tango-Controls is a software toolkit for building control systems.
  dev_url: https://gitlab.com/MaxIV/lib-maxiv-dsconfig

extra:
  recipe-maintainers:
    - beenje
```

Local recipe

```
{% set data = load_setup_py_data(setup_file="../setup.py",
  from_recipe_dir=True) %}

package:
  name: tango_exporter
  version: {{ data.get('version') }}

source:
  path: ..

build:
  number: 0
  noarch: python
  script: {{ PYTHON }} -m pip install . -vv
  entry_points:
    - tango_exporter = tango_exporter:main

requirements:
  host:
    - pip
    - python >=3.6
  run:
    - python >=3.6
{% for dep in data['install_requires'] %}
    - {{ dep.lower() }}
{% endfor %}

test:
  imports:
    - tango_exporter
  requires:
    - pip
  commands:
    - pip check
    - tango_exporter --help

about:
  home: https://gitlab.maxiv.lu.se/kits-maxiv/app-maxiv-tangoexporter
  license: GPL-3.0-or-later
  license_file: ../LICENSE.txt
  summary: Prometheus exporter for a Tango control system.
```

Conda GitLab CI

```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.python-ci.yml'

variables:
  RPM_ENABLED: "false"
```

Conda GitLab CI

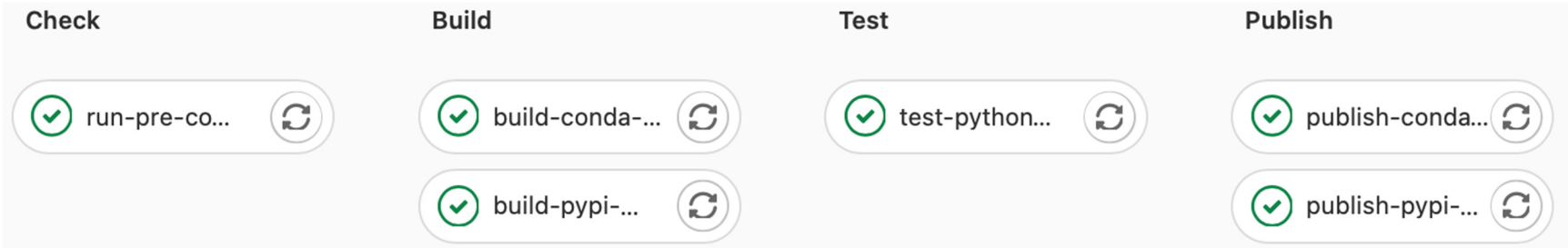
```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.python-ci.yml'

variables:
  RPM_ENABLED: "false"
```

```yaml
build-conda-package:
  stage: build
  image: harbor.maxiv.lu.se/kits-sw/conda-build:latest
  tags: [kubernetes]
  script:
    # Download conda_build_config.yaml here as the entrypoint isn't executed
    # by kubernetes executor: https://gitlab.com/gitlab-org/gitlab-runner/-/issues/4125
    - curl -Ls ${CONDA_BUILD_CONFIG_URL} > ${HOME}/conda_build_config.yaml
    - mkdir pkg
    - 'echo "CONDA_BUILD_EXTRA_ARGS: $CONDA_BUILD_EXTRA_ARGS"'
    - >
      conda build --error-overlinking
      --skip-existing $CONDA_BUILD_EXTRA_ARGS
      --output-folder pkg recipe
  artifacts:
    paths:
      - pkg/linux-64/*.tar.bz2
      - pkg/noarch/*.tar.bz2
    expire_in: 1 week
  rules:
    - if: '$CI_COMMIT_BRANCH != $CI_DEFAULT_BRANCH'
      variables:
        CONDA_BUILD_EXTRA_ARGS: "-c maxiv-kits-dev"
      exists:
        - recipe/meta.yaml
```

Conda GitLab CI

```yaml
include:
  - project: 'kits-maxiv/cfg-maxiv-gitlabci'
    file: '/.python-ci.yml'

variables:
  RPM_ENABLED: "false"
```

```yaml
build-conda-package:
  stage: build
  image: harbor.maxiv.lu.se/kits-sw/conda-build:latest
  tags: [kubernetes]
  script:
    # Download conda_build_config.yaml here as the entrypoint isn't executed
    # by kubernetes executor: https://gitlab.com/gitlab-org/gitlab-runner/-/issues/4125
    - curl -Ls ${CONDA_BUILD_CONFIG_URL} > ${HOME}/conda_build_config.yaml
    - mkdir pkg
    - 'echo "CONDA_BUILD_EXTRA_ARGS: $CONDA_BUILD_EXTRA_ARGS"'
    - >
      conda build --error-overlinking
      --skip-existing $CONDA_BUILD_EXTRA_ARGS
      --output-folder pkg recipe
  artifacts:
    paths:
      - pkg/linux-64/*.tar.bz2
      - pkg/noarch/*.tar.bz2
    expire_in: 1 week
  rules:
    - if: '$CI_COMMIT_BRANCH != $CI_DEFAULT_BRANCH'
      variables:
        CONDA_BUILD_EXTRA_ARGS: "-c maxiv-kits-dev"
      exists:
        - recipe/meta.yaml
```



MAX IV

Ansible RPM packages role

```yaml
- name: Ensure the required packages are present
  yum:
    name: "{{ packages_stable_present }}"
    disablerepo: maxiv-testing,maxiv-testing-noarch
    update_cache: yes
    state: present
  tags:
    - stable
    - packages
```

```yaml
versions:
  python-taurus: 4.5.1-7.el7
  python-dsconfig: 1.4.0-1.el7
  python-facadedevice: 1.0.3.dev1-1.el7
  python-fandango: 14.3.0-1.el7
  python36-sdm: 1.6.1
  tangods-pathfixer: 1.5.7
  python-pyicepap: 2.8.1-1.el7
  python36-sherlock: 1.0.4
```

inventory/group_vars/all.yml

```yaml
packages_stable:
  python-taurus: default
  python-dsconfig: default
  python-facadedevice: 0.9.0
  python-fandango: 10.9
  python36-sdm: default
  tangods-pathfixer: default
  python-pyicepap: default
  python36-sherlock: latest
```

inventory/host_vars/myhost.yml

Ansible conda modules

Custom modules (no builtin):

```
- name: install flask 2.0 and Python 3.9
  conda:
    name:
      - python=3.9
      - flask=2.0
    state: present
    environment: myapp
```

```
- name: create myenv environment
  conda_env:
    name: myenv
    state: present
    file: /tmp/environment.yml
```

Ansible conda role

```yaml
- name: create or update the conda environment
  conda:
    name: "{{ item.dependencies }}"
    state: "{{ item.state | default('present') }}"
    environment: "{{ item.env_name }}"
    channels: "{{ item.channels | default(omit) }}"
    use_mamba: "{{ item.use_mamba | default(omit) }}"
  loop: "{{ conda_envs }}"
  tags: conda-env

- name: create the cli wrappers
  template:
    src: wrapper.j2
    dest: /usr/local/bin/{{ item[1] }}
    mode: 0755
    owner: root
    group: root
  loop: "{{ conda_envs | subelements('wrappers') }}"
  loop_control:
    label: "env:{{ item[0].env_name }}, wrapper:{{ item[1] }}"
  tags: conda-env
```

Ansible conda role

```yaml
- name: create or update the conda environment
  conda:
    name: "{{ item.dependencies }}"
    state: "{{ item.state | default('present') }}"
    environment: "{{ item.env_name }}"
    channels: "{{ item.channels | default(omit) }}"
    use_mamba: "{{ item.use_mamba | default(omit) }}"
  loop: "{{ conda_envs }}"
  tags: conda-env

- name: create the cli wrappers
  template:
    src: wrapper.j2
    dest: /usr/local/bin/{{ item[1] }}
    mode: 0755
    owner: root
    group: root
  loop: "{{ conda_envs | subelements('wrappers') }}"
  loop_control:
    label: "env:{{ item[0].env_name }}, wrapper:{{ item[1] }}"
  tags: conda-env
```

```yaml
conda_envs:
  - env_name: silx
    dependencies:
      - python=3.9
      - silx=0.15.1
    wrappers:
      - silx
```


MAX IV

Ansible conda role

```yaml
- name: create or update the conda environment
  conda:
    name: "{{ item.dependencies }}"
    state: "{{ item.state | default('present') }}"
    environment: "{{ item.env_name }}"
    channels: "{{ item.channels | default(omit) }}"
    use_mamba: "{{ item.use_mamba | default(omit) }}"
  loop: "{{ conda_envs }}"
  tags: conda-env

- name: create the cli wrappers
  template:
    src: wrapper.j2
    dest: /usr/local/bin/{{ item[1] }}
    mode: 0755
    owner: root
    group: root
  loop: "{{ conda_envs | subelements('wrappers') }}"
  loop_control:
    label: "env:{{ item[0].env_name }}, wrapper:{{ item[1] }}"
  tags: conda-env
```

```yaml
conda_envs:
  - env_name: silx
    dependencies:
      - python=3.9
      - silx=0.15.1
    wrappers:
      - silx
```

```
#!/bin/bash

# Some cli need the env to be activated
source /opt/conda/etc/profile.d/conda.sh
conda activate silx

/opt/conda/envs/silx/bin/silx "$@"
```

/usr/local/bin/silx

MAX IV

Ansible sardana role

```yaml
- name: create sardana environment.yml file
  copy:
    dest: /tmp/sardana.yml
    content: "{{ sardana_env | to_yaml }}"
    mode: 0644
    owner: root
    group: root
  tags: sardana-env

- name: create sardana environment
  conda_env:
    name: "{{ sardana_env_name }}"
    state: present
    file: /tmp/sardana.yml
  tags: sardana-env
```

Ansible sardana role

```yaml
- name: create sardana environment.yml file
  copy:
    dest: /tmp/sardana.yml
    content: "{{ sardana_env | to_yaml }}"
    mode: 0644
    owner: root
    group: root
  tags: sardana-env

- name: create sardana environment
  conda_env:
    name: "{{ sardana_env_name }}"
    state: present
    file: /tmp/sardana.yml
  tags: sardana-env
```

```yaml
channels: [maxiv-kits, mini-conda-forge]
dependencies:
- pytango=9.3.3
- taurus=4.7.1
- taurus_pyqtgraph
- h5py
- hdf5plugin
- icepap=3.6.2
- sockio=0.15.0
- taurusgui-maxpeemenergy=0.2.2
- python=3.9
- sardana=3.1.2
- bitshuffle
- matplotlib
- spyder
- pytest
- pip
- pip:
    - --trusted-host repo.maxiv.lu.se
    - '-i http://repo.maxiv.lu.se/devpi/maxiv/prod'
    - taurusgui-scangui3==1.1.3
    - taurusgui-quickgui3==2.1.3
name: sardana
```

Conclusion

Use a package manager!

Each package manager has its strengths and weaknesses

- RPM:
  - OS integration (users, systemd services)
- Conda
  - OS independent
  - Isolated environments (more freedom in the packages to install)

Ansible + RPM or conda: reliable and reproducible deployment