

Improving Control System Software Deployment at MAX IV

ICALEPCS 2023 – MO4BCO04

B.Bertrand, Á. Freitas, A.F. Joubert

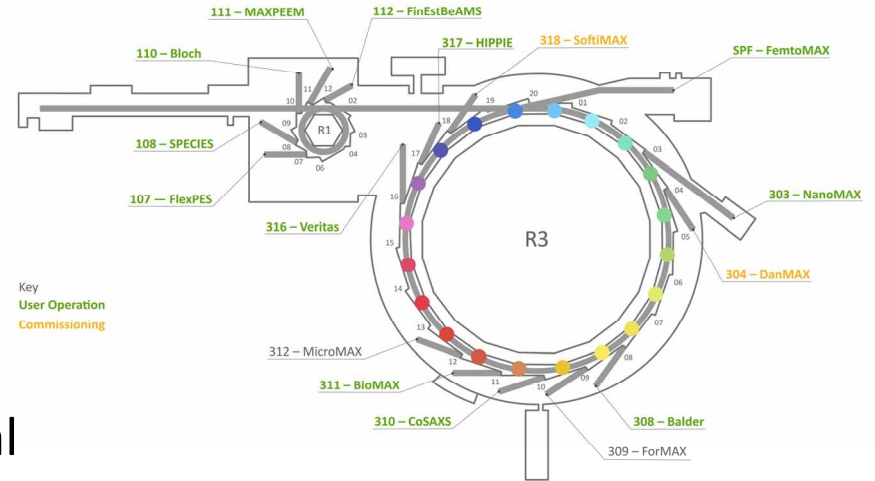
Outline

- Introduction
- Package Management
- Deployment
- Monitoring

Introduction



- Synchrotron facility
- 16 beamlines
- Control System based on Tango
- More than 500 Virtual and Physical machines
- 24k Tango devices with 134k configurable properties
- Using Ansible for 10 years



Package Management

RPM & conda

Have been using **RPM** for many years

- RPM Package Manager (originally Red Hat Package Manager)
- Distribution package manager



Switched to **conda**


- Package, dependency and environment management for any language
- 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



Conda switch: how?

Initially created a cookiecutter template to generate a conda recipe:

- Required to add a recipe to every repository
- One more file to maintain (switching to setuptools-scm had impact on the recipe)
- Slow adoption

Solution? Grayskull 
Recipe generator for Conda

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

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

source:
  path: ..

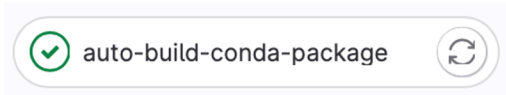
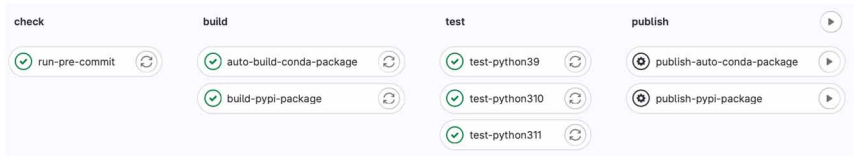
build:
  number: 0
  noarch: python
  script: {{ PYTHON }} -m pip install . -vv
  entry_points:
    - tango_exporter = tango_exporter:main

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.
```

Grayskull



auto-build-conda-package:

```
extends: .conda_build
before_script:
```

```
# Generate recipe with grayskull from local sdist
```

```
- /grayskull/bin/python -m build -s
```

```
- mkdir recipe
```

```
- /grayskull/bin/grayskull pypi -m KITS -o recipe dist/*.tar.gz
```

```
# Many entry points (like taurusgui) don't have a --help option... Skip entry point test...
```

```
- sed -i "/ --help/d" recipe/*/meta.yaml
```

```
- cat recipe/*/meta.yaml
```

```
- |
```

```
grep -q "noarch: python" recipe/*/meta.yaml || { echo "Recipe isn't noarch. Should script be replaced by entry_point? Aborting."; exit 1; }
```

```
$ cat recipe/*/meta.yaml
{% set name = "tangods-massoft" %}
{% set version = "1.1.0" %}
package:
  name: {{ name|lower }}
  version: {{ version }}
source:
  url: file:///builds/kits-maxiv/dev-maxiv-massoft/dist/tangods-massoft-1.1.0.tar.gz
  sha256: 3cf9bb8e3ba1dfd498ccf4cd663736c8476121f9a3821448c6115df5a9990d8
build:
  entry_points:
    - MASsoft=massoftds.server:main
  noarch: python
  script: {{ PYTHON }} -m pip install . -vv
  number: 0
requirements:
  host:
    - python >=3.9
    - setuptools-scm
    - pip
  run:
    - python >=3.9
    - pytango
    - massoftlib >=1.1.0
test:
  imports:
    - massoftds
  commands:
    - pip check
  requires:
    - pip
```

Benefits

- Separate the deployment from the OS packaging and system Python version
- Migrated from CentOS 7 to Rocky Linux 8 using the same conda packages
- Use modern Python: 3.9 for a while – moving to 3.11
- conda-forge ecosystem
- Easy to develop locally in a conda env (Linux, macOS, Windows)

Constraints

conda build creates a package, installs it in a clean env and runs some tests.
Build part is slow 😞

auto-build-conda-package CI job: around 5 minutes

Future?

A new recipe format – part 1 [↗](#)

Title	A new recipe format
Status	Proposed
Author(s)	Wolf Vollprecht <wolf@prefix.dev>
Created	May 23, 2023
Updated	May 23, 2023
Discussion	
Implementation	https://github.com/prefix-dev/rattler-build



<https://github.com/conda-incubator/ceps/pull/54>

conda mambabuild recipe: 3 minutes 18 seconds

rattler-build build -r recipe: 23 seconds

Deployment

Ansible inventory: client computers

```
conda_envs_extra:
- "{{ conda_env_snapshot }}"
- "{{ conda_env_luxviewer }}"
- "{{ conda_env_silxliveview }}"
- "{{ conda_env_ctfrontends }}"
- "{{ conda_env_mapping_scan_tool }}"
- env_name: ctformaxsynoptic
  dependencies:
    python: default
    pytango: default
    taurus: default
    svgsynoptic2: default
    taurusgui-formaxsynoptic: default
  wrappers:
    - ctformaxsynoptic
desktop-menus:
- name: ForMAX Synoptic
  exec: ctformaxsynoptic
```

Ansible inventory: servers

```
packages_stable:
  lima-basler: 1.7.2
  lima-core: 1.7.2
  tangods-limaccds: 1.5.0
  tangods-basler: default
  tangods-pathfixer: default

tango_devices:
  - device: b112a-0a05/dia/cam-03
    server: Basler
    class: Basler
    instance: B112A-0A05-3
    properties:
      camera_ip: b112a-0a05-dia-cam-03
      inter_packet_delay: "4000"
      packet_size: "1500"
      max_push_event_frequency: '2'
  - device: b112a-ob04/dia/cam-01
    server: Basler
    class: Basler
    instance: B112A-0B04-1
    properties:
      camera_ip: b112a-ob04-dia-cam-01
      inter_packet_delay: "4000"
      packet_size: "1500"
      max_push_event_frequency: '2'
```

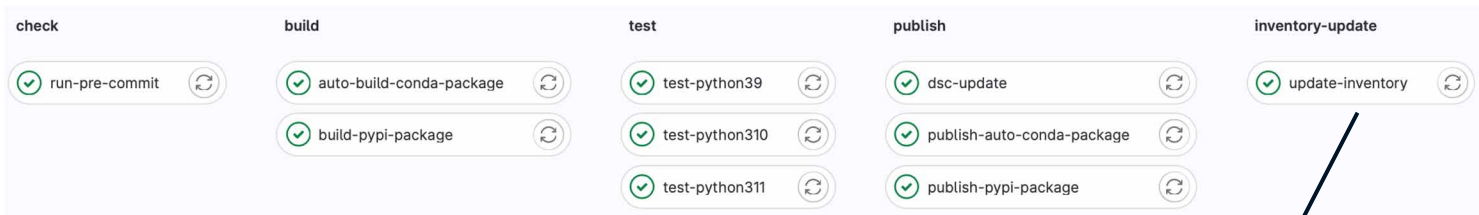
Old Ansible inventory

```
tango_ds:
  - name: Basler
    conda_packages:
      tangods-basler: default
    instances:
      - name: B112A-0A05-3
        devices:
          - name: b112a-0a05/dia/cam-03
            properties:
              camera_ip: b112a-0a05-dia-cam-03
              inter_packet_delay: '4000'
              packet_size: '1500'
              max_push_event_frequency: '2'
      - name: B112A-0B04-1
        devices:
          - name: b112a-ob04/dia/cam-01
            properties:
              camera_ip: b112a-ob04-dia-cam-01
              inter_packet_delay: '4000'
              packet_size: '1500'
              max_push_event_frequency: '2'
```

New definition

Inventory update

 Benjamin Bertrand @benber
-o- Pushed new tag 1.1.0



The CI pipeline consists of five stages:

- check**: run-pre-commit
- build**: auto-build-conda-package, build-pypi-package
- test**: test-python39, test-python310, test-python311
- publish**: dsc-update, publish-auto-conda-package, publish-pypi-package
- inventory-update**: update-inventory

```
inventory/group_vars/all.yml
```

	@@	-577,7	+577,7	@@	versions:
577	577				tangods-logger: 1.0.0-1.el7.centos.maxlab
578	578				tangods-magnet: 1.1.5
579	579				tangods-magnetcircuit: 3.0.9
580	-				tangods-massoft: 1.0.0
580	+				tangods-massoft: 1.1.0
581	581				tangods-mcs2: 1.0.6

KITS MAXIV > ansible-galaxy > cfg-maxiv-ansible > Merge requests > 13372

Update tangods-massoft to 1.1.0

Merged kits-sw-inventory-bot requested to merge update_tangods-massoft_to_1.1.0 into master 1 day ago

Overview 0 Commits 1 Pipelines 1 Changes 1

This MR was created by the inventory-update script.
It was triggered by Benjamin Bertrand when tagging dev-maxiv-massoft 1.1.0 by the job 390786.

You can check the list of merged MR.

0 thumbs up 0 thumbs down

Pipeline #129106 passed for 0f98a7b on update_tangods-massoft_to_1.1.0 1 day ago

8 Approval is optional

Merged by Benjamin Bertrand 1 day ago

Merge details

- Changes merged into master with 946c7792.
- Deleted the source branch.

Add a to do >>

Assignee Benjamin Bertrand Edit

0 Reviewers None - assign yourself Edit

Labels None Edit

Milestone None Edit

Time tracking No estimate or time spent +

Lock merge request Unlocked Edit

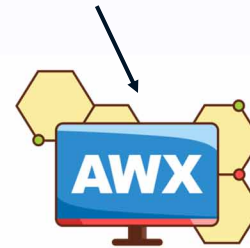
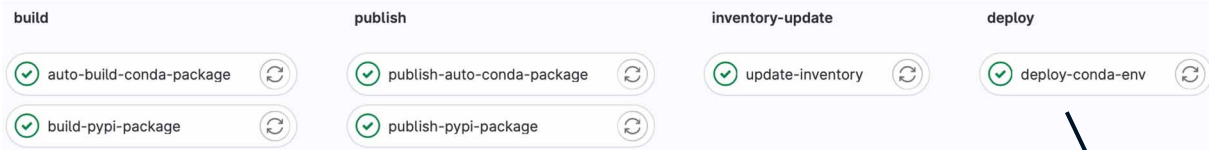
Notifications On Edit

2 Participants

Continuous deployment

Mirko Milas @mirmil
-> Pushed new tag 1.6.0

MR in the inventory is automatically merged



Only for specific use-case!
Synoptic is a typical one.
Beamline staff can update the app themselves.

```
PLAY RECAP *****
b-micromax-cc-0      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-micromax-cc-1      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-micromax-cc-2      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-micromax-cc-3      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-micromax-cc-4      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-micromax-cc-5      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-micromax-cc-6      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-micromax-cc-7      : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
b-v-micromax-cc-0    : ok=4   changed=1  unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
```

Monday deployment

The screenshot shows a GitHub milestone page for 'release-2023-06-12'. The milestone is marked as 'Closed' and 'Milestone expired on Jun 12, 2023'. It has 21 merge requests, 0 issues, 0 participants, and 0 labels. The 'Merge requests' section is highlighted with a blue box. Below this, there are four columns representing different stages of merge requests: 'Work in progress (open and unassigned)' with 1 item, 'Waiting for merge (open and assigned)' with 0 items, 'Rejected (closed)' with 0 items, and 'Merged' with 20 items. The 'Merged' column is expanded to show a list of merged merge requests, including 'New species archiver gui', 'Update sherlock to 1.1.2', 'Remove unused RPMs at Balder', 'Update Balder to Sardana 3.4.0', 'CoSAXS: Pilatus and PandABox Sardana Ctrl', and 'Update tangods-scpowersupply to 1.1.4'.

- All *approved* Ansible MRs belonging to the corresponding milestone are merged by the Monday Deployment Crew.
- The Monday Deployment Crew deploys the updates by running the deploy playbook in AWX, limited to <beamlines>-ec and <beamlines>-cc groups

Important to keep the inventory in sync with what is deployed.
Playbook shall be run regularly.

Monitoring

Prometheus packages export

```
*/30 * * * * root nice -n 10 /usr/local/bin/prometheus_packages_export > /var/lib/node_exporter/packages.prom
```

```
package_conda{env="sardana",name="pytango",platform="linux-64",  
version="9.3.6",build_string="py39hf983217_2",channel="mini-conda-forge",  
requested="1",local_changes="0"} 1  
package_rpm{name="tango-java",platform="x86_64",version="9.3.5-24.el8.maxlab",  
source="maxiv-public",local_changes="0"} 1
```

packages.prom example

type	conda	package	pytango	project	maxpeem	env	sardana	
Installations								
name	project	env	version	build_string	channel	hostname	requested	local_changes
pytango	maxpeem	sardana	9.3.6	py39hf983217_2	mini-conda-forge	b-maxpeem-cc-0	1	0
pytango	maxpeem	sardana	9.3.6	py39hf983217_2	mini-conda-forge	b-maxpeem-cc-1	1	0
pytango	maxpeem	sardana	9.3.6	py39hf983217_2	mini-conda-forge	b-v-maxpeem-cc-0	1	0
pytango	maxpeem	sardana	9.3.6	py39hf983217_2	mini-conda-forge	b-v-maxpeem-ec-2	1	0

Grafana dashboard

Nox

```
☰ README.md 188 B [Code Icon] [File Icon] Edit ▾  
1 # tangods-massoft  
2  
3   
4  
5 Tango device server to interact with Hiden Analytical MASsoft over sockets.
```

📄 README.md

tangods-massoft

Name	Number of hosts	Source	Info
tangods-massoft	2		
Host	Version	Source	Info
b-v-finest-ec-3	1.2.1.dev5+gff9e16d	maxiv-kits-dev	Environment: MASsoft
b-v-flexpes-ec-2	1.2.0	maxiv-kits	Environment: MASsoft

Open Conda panel ↗
Open RPM panel ↗
Generated by <https://nox.apps.okd.maxiv.lu.se>

Tango device server to interact with Hiden Analytical MASsoft over sockets.

Conclusion

- Adopted conda as our primary packaging tool instead of RPM
- Automatic conda recipe creation thanks to grayskull
- Improved Ansible workflow
 - Easier way to define Tango Device Servers
 - Restart applications during deployment (if needed)
- Regular deployment process
- Monitoring tools with Prometheus

Contact

Benjamin BERTRAND

benjamin.bertrand@maxiv.lu.se