Automation tools in AV workflows

Building digital pipelines using standard IT tools.

Jonáš Svatoš
Národní filmový archiv, CZ

No Time To Wait! #3
London, UK. 25.-26.10.2018
Various steps when working with AV content

- Ingest
- Verification
- Storage
- Transcoding
- Packaging
- Delivery
- ..
Workflow management?

- DAMS
- Digipres systems like Archivematica, Preservica, Adlib
- Manual flow
Scripts are good, but ..

- Individual scripts do not serve a generic purpose
- Complex workflows must chain many steps together
- Logging anyone?
- Auditing the logs anyone?
Is this deterministic?
Automation to the rescue!

- Define inputs
- Implement steps by reusing existing tools and scripts
- Set validations and conditions for eventual branching
- Assign notifications
- Read the notifications :-)
- Audit the logs (GUI!)
There are generic tools available

- Jenkins
- Ansible
- Rundeck
- Buildbot
- ..
Jenkins

- Build automation server
- Web GUI, written in Java
- As it is generic, we can use it for whatever purpose
- Multi-tenant
- Plugin ecosystem
Jenkins jobs overview
Jenkins job parameters

Project MAKE-BAGIT
This build requires parameters:

- **SOURCE**: /mnt/central/TEMP_JONAS/SPC_zvuk
  - What is bag
- **INTERNAL_DESCRIPTION**: Please see GENERAL_INFORMATION.pdf in the bag
  - Technical description
- **INTERNAL_IDENTIFIER**: 4001054
- **NODE**: cl1-render-1
Jenkins job parameters config
Jenkins build log

Finished: SUCCESS
Jenkins job tasks

```
Execute shell
Command
# /bin/bash
set +x
virtualenv-3.7 venv
source venv/bin/activate
pip install --quiet bagit
export PYTHONUNBUFFERED=1

See the list of available environment variables
```

```
Execute shell
Command
# /bin/bash
set +x
source venv/bin/activate
ORGANIZATION="Národní filmový archiv"
ORGANIZATION_ADDRESS="Malolická 12, Praha 3, 130 00, Czech Republic"
CONTACT_NAME="Digital laboratory"
CONTACT_EMAIL="digilab@nf.a.cz"

bagit.py --source-organization "$ORGANIZATION" --organization-address "$ORGANIZATION_ADDRESS"

See the list of available environment variables
```
Jenkins past build detail
Jenkins job chaining
Jenkins slaves
Jenkins slave tag matching

The screen shows a section of the Jenkins slave configuration within a project named MAKE-SCREENER. The specific settings highlighted include:

- Label Expression: The label expression is set to "linux & cuda", indicating that the slave node must match this label to execute the build.
- Source Code Management: The option selected is Git, and the repository URL is set to "https://git-digitlab.nfa.cz/digitlab/fmpeg-builder.git".

The configuration includes options for adding parameters, throttling builds, managing environment, and defining build triggers. The screen also shows a save and apply button at the bottom.
Jenkins as a cron
Jenkins job config in Git
Conclusions

- Standardized workflow needs deterministic steps
- As little human intervention as possible
- There are generic automation servers available
- Log retention!
- Jenkins is great
Thanks! Questions?

jonas.svatos@nfa.cz

github.com/NFAcz