DEPLOYING TO CERNVM-FS VIA GITLAB-CI

CernVM Users Workshop 2018

Marko Petrič

On behalf of the CLICdp collaboration

Geneva, 30 January 2018
CLIC Detector and Physics

CLICdp: 30 institutes
clicdp.cern.ch

Focus of CLIC-specific studies on:
Physics prospects and simulation studies
Detector optimisation + R&D for CLIC

▫ First stage at 380 GeV (affordable within CERN budget)
▫ Upgradable to 3 TeV $e^+e^-$
Usage of CernVM-FS

Grid Software

- public API for users and prod managers to submit jobs
- CernVM-FS deployed at all sites

Offline Software

- shared with ILC users
- sim and reco job
- user jobs
- continuous integration

Marko Petrič (CERN) marko.petric@cern.ch

Deploying to CernVM-FS via Gitlab-CI
Where we started 1/2

- iLCSoft hosted on GitHub
  - SW composed of individual sub-packed that can be build against a release
  - No need for CERN credentials
- Use Travis-CI to build PR and run tests
- Use VM with sudo rights → install CVMFS directly on node
- Use Docker with mounted CVMFS
- Build against nightly on CVMFS

+ Good performance
  - Wait for nightly, hard to test dependencies and API changes
Where we started 2/2

1. Run builds in containers on builder nodes
2. Tar build and deploy to EOS
3. On stratum0 cURL the files and deploy
   - Long feedback time
   - Not clear which of the PRs broke the build

Where we wanted to go:
- Rebuild SW stack on demand for merges to certain packages
- Unify nightly and release builds
Can we do it with GitLab?

- Obviously it’s possible via Jenkins, but can it be easier?

Requirements:
1. Way to talk from GitLab to `stratum0` → `gitlab-runner`
2. Store build results → Gitlab-CI artifacts
3. Move builds into sudo env. and interact with server
   - CERN-IT instance
GitLab to stratum0 connection

- Cannot use shared runners provided by IT, since they only support docker executors
- Use `gitlab-runner` on dedicated machine/VM and use SSH executor
  - allows execution of CI command on third party machine

```json
[[runners]]
  name = "clicdpci13.cern.ch"
  url = "https://gitlab.cern.ch/"
  token = "05jf6p56jguq5ieos7usg05jf6p56j"
  executor = "ssh"
  builds_dir = "/tmp"
  [runners.ssh]
    user = "clicdpcs"
    host = "cvmfs-clicdp.cern.ch"
    port = "22"
    identity_file = "/__path__/ssh/id_rsa"
  [runners.cache]
```
GitLab to `stratum0` connection

- Authenticate via private key
- Slight hurdle with AFS accounts
- Copy public key to `~/public`
- `ln -s public/.ssh/authorized_keys .ssh/authorized_keys`
- Repeat for `.bash_profile` and `.bashrc`
  - Missing login script return code 1

Now you are able to execute commands in `/tmp on stratum0`
Storage of build results

- Use GitLab-CI to store build results and logs as artifacts of CI pipelines
- gitlab-runner with SSH executor does not support automatic download of artifacts
- Use GitLab API to find information about rights jobs
  
  ```
  curl -g --header "PRIVATE-TOKEN:$API_TOKEN"
  ```
- Parse `.json` and download artifacts

```python
import os
import json

with open('pipeline_info.json') as json_data:
    data = json.load(json_data)

# Find last passed GCC build
for i in range(0, len(data)):
    if data[i]['name'] == 'build:GCC6':
        print "Downloading GCC build"
        os.system("curl --0 https://gitlab.cern.ch/CLICdp/SoftwareConfigurations/iLCSoft/~/jobs/%s/artifacts/raw/ilcsoft_gcc.tar" % data[i]['id'])
        break
```
Deployment

- Move files from /tmp to sudo env. and deploy
  
  ```shell
  export RUNNER_LOCATION=$(pwd)
  sudo -u cvclicdp -i cp $RUNNER_LOCATION/ilcsoft_gcc.tar .
  sudo -u cvclicdp -i ./scripts/deploy.sh ilcsoft_gcc.tar $PATH
  ```

- Helper script checks if transaction in progress and deploys to desired path

- interactive sudo not possible via gitlab-runner
Final Setup

- Putting everything together

- Trigger this pipeline each time a PR is merged on GitHub
- Use the same approach for iLCDIRAC and AllPix$^2$
- Install for each project gitlab-runner on the same machine
- Allow only one concurrent runner → pipeline of CernVM-FS publish jobs
Summary

- CLICdp very happy with CernVM-FS performance
- Successfully integrated publishing into GitLab including pipelining
- For details on the implementation check repo https://gitlab.cern.ch/CLICdp/SoftwareConfigurations/iLCSoft/
- Wish list:
  - Means to know when new transaction replicated to all sites