



# State of the new build scripts

- **Work in progress**

# INTRODUCTION TO LCG RELEASES



- Provide LHCb, ATLAS, BE, SWAN with externals and HEP software
- Two releases per year
- Build many different configurations
  - ROOT *stable* and *HEAD*
  - Python 2 and 3
  - GeantV, ...
- Test latest changes and integration

## Continuous Integration:

- <https://epsft-jenkins.cern.ch/>
- <https://lcgapp-services.cern.ch/jenkins/>

## Monitoring:

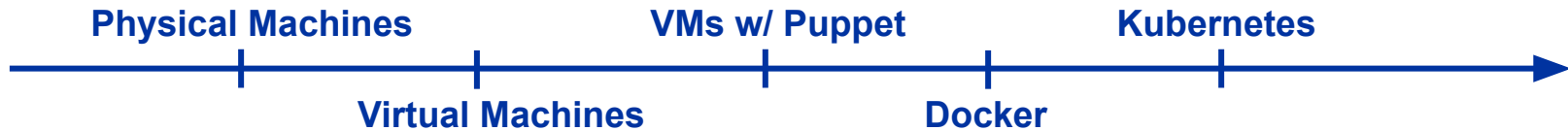
- <https://lcgdocs.web.cern.ch/lcgdocs/report/>
- <http://cdash.cern.ch/index.php?project=LCGSoft>

## Documentation:

- <http://lcginfo.cern.ch/>
- <http://lcgdocs.web.cern.ch/lcgdocs/>

# A BRIEF HISTORY OF BUILD INFRASTRUCTURE

Moving from “pets to cattle”



[1] <http://www.allpetsvetcenter.com>

[2] <http://blog.lenovo.com/en/blog/pets-cattle-and-it-infrastructure>

# CURRENT PRODUCTION STATE



- Most Nightlies run in a Docker container on a CentOS 7 host
- Ubuntu builds run in a Docker container on a Ubuntu 18.04 host (new Kernel)
- Some Nightlies (devgeantv), experimental and releases run on VMs directly
- Mixture of different environments
  - VMs and Docker
  - AFS, EOS, CVMFS

# IMPROVING OUR DOCKER INFRASTRUCTURE



## Update and refactor existing Docker images

- Follow new style guidelines for Dockerfiles
- Same behaviour across all Docker images
  - *user* sftnight, *group* sf with correct IDs
  - *same workspace*
  - *bash as default command*
- Updated HEP\_OSlibs
  - (!) Currently only used in SLC6, Ubuntu 16.04 and Ubuntu 18.10
  - Increases Docker image size significantly
- Added Ubuntu 19.04 and Fedora 30

# IMPROVING OUR DOCKER INFRASTRUCTURE



## Automated build and deployment of new Docker images:

- lcg\_build\_docker\_images in Jenkins
- [https://epsft-jenkins.cern.ch/job/lcg\\_build\\_docker\\_images/](https://epsft-jenkins.cern.ch/job/lcg_build_docker_images/)

## Our Docker Registry: [https://gitlab.cern.ch/sft/docker/container\\_registry](https://gitlab.cern.ch/sft/docker/container_registry)

- Adding backups with timestamps
- New naming scheme: Use names instead of tags

## Documentation:

- <http://lcgdocs.web.cern.ch/lcgdocs/tasks/newpackages.html>

# INTRODUCING UTILITY DOCKER IMAGE



RedHat: Principles of container-based application design [1]

- One Docker image per task
- Split build and other tasks into different images
- `gitlab-registry.cern.ch/sft/docker/utility:latest`
  - based on CERN-supported CentOS 7 image
  - Kerberos, XRootD, S3 support
  - lightweight

[1] <https://www.redhat.com/cms/managed-files/cl-cloud-native-container-design-whitepaper-f8808kc-201710-v3-en.pdf>

# INTRODUCING AEDIFEX



<https://gitlab.cern.ch/jheinz/aedifex>



- Consolidate multiple scripts into a few general-purpose files
- Document code and add README files
- Improve logical structure
- Move from Python 2 to Python 3 (using *distro*, *subprocess*)
- Drop support for AFS, Windows, Intel Compiler, old versions of GCC and Clang
- Rename some ENV variables for better understanding → Needs changes in Jenkins
- Target: Docker builds on CentOS 7 hosts, not yet Kubernetes-ready



# KUBERNETES



**Kubernetes** (short **k8s**) is an open source system for managing containerized applications across multiple hosts; providing basic mechanisms for deployment, maintenance, and scaling of applications. -- <https://github.com/kubernetes/kubernetes/>

## Our test cluster on OpenStack » Orchestration

- 1cgapp-kubernetes -- <https://openstack.cern.ch/project/stacks/>
- Runs on our OpenStack quota → Limits auto-scaling
- Kubernetes 1.13
- 1 master node, 3 worker slaves
- CVMFS support via CSI driver -- <https://github.com/cernops/cvmfs-csi>

# KUBERNETES



## Support at CERN:

- CloudDocs <https://clouddocs.web.cern.ch/clouddocs/>
- <https://mattermost.web.cern.ch/it-dep/channels/cloud-containers>
- Ricardo Brito Da Rocha [Ricardo.Rocha@cern.ch](mailto:Ricardo.Rocha@cern.ch)

## See talk at CernVM Workshop 2019:

- <https://indico.cern.ch/event/757415/contributions/3421574/>

# KUBERNETES: USAGE



## Two clients

1. kubectl command line program on lxplus-cloud.cern.ch
2. REST API [1] with API KEY from Jenkins (accessible only inside CERN)

## Definition of resources:

- YAML markup language

[1] <https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.13/#create-job-v1-batch>

# KUBERNETES: USAGE



**Live Demo?**

# CHALLENGES AND OUTLOOK



- Kubernetes jobs provide no immediate feedback on Jenkins: Fire and forget
- No central volume yet for caching, logs, build artefacts
- No clear way to inject Kerberos ticket for EOS access
  - Proposal: Add encrypted ticket to utility image and pass decryption key via ENV
- Add additional section to LCG Docs about Kubernetes usage