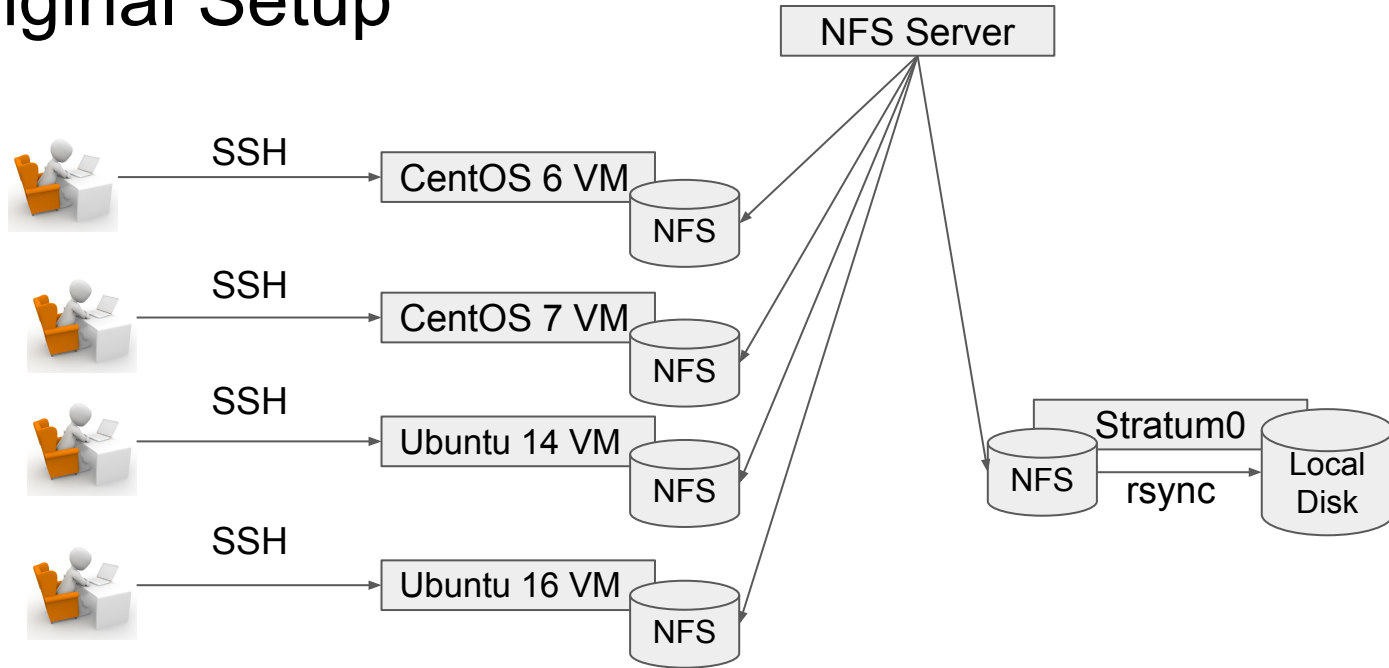


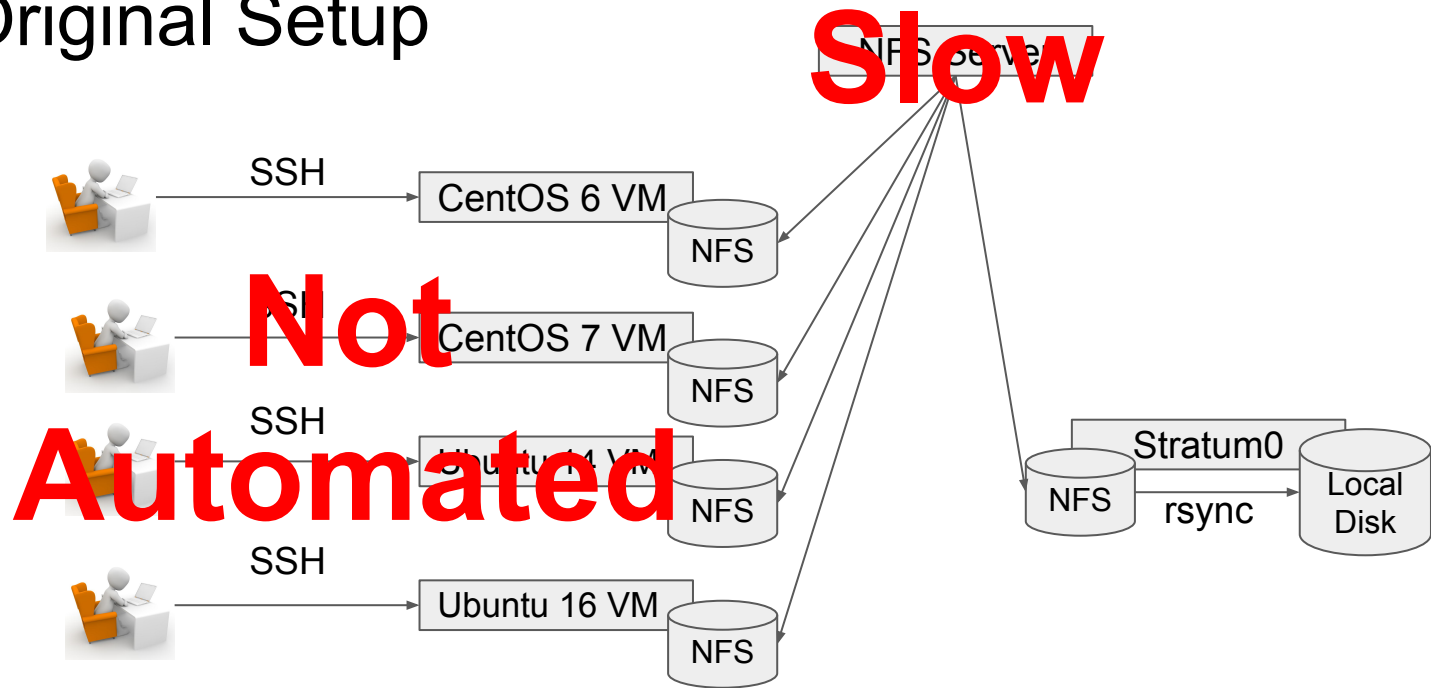
CVMFS Build and Release Pipeline Using Docker Microservices

Heath Skarlupka (heath.skarlupka@icecube.wisc.edu)
David Schultz (dschultz@icecube.wisc.edu)

Original Setup



Original Setup



1 Repository Update / Day

Could we use Devops and
Continuous Delivery principles
to build something better?

Principles

Infrastructure as code - Our infrastructure deployment model had to be repeatable and scale out as we supported new platforms

Automation - Our build and deployment process needed to be automated and needed to be started with a simple git push

Tools

[Kubernetes](#) - Microservices scheduler across multiple servers

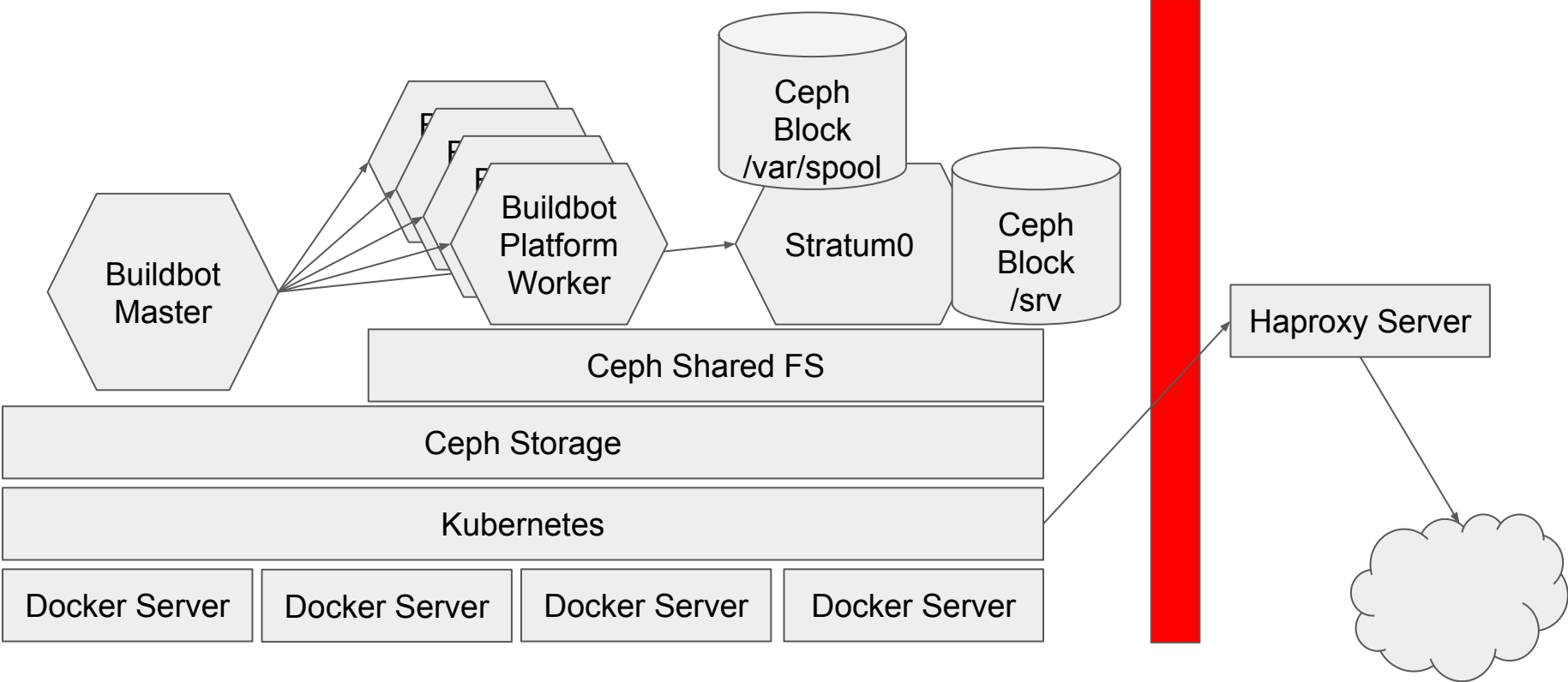
[Ceph storage](#) - Shared filesystem and block device deployment in Kubernetes

[Docker](#) - Microservices are a key component to infrastructure as code design

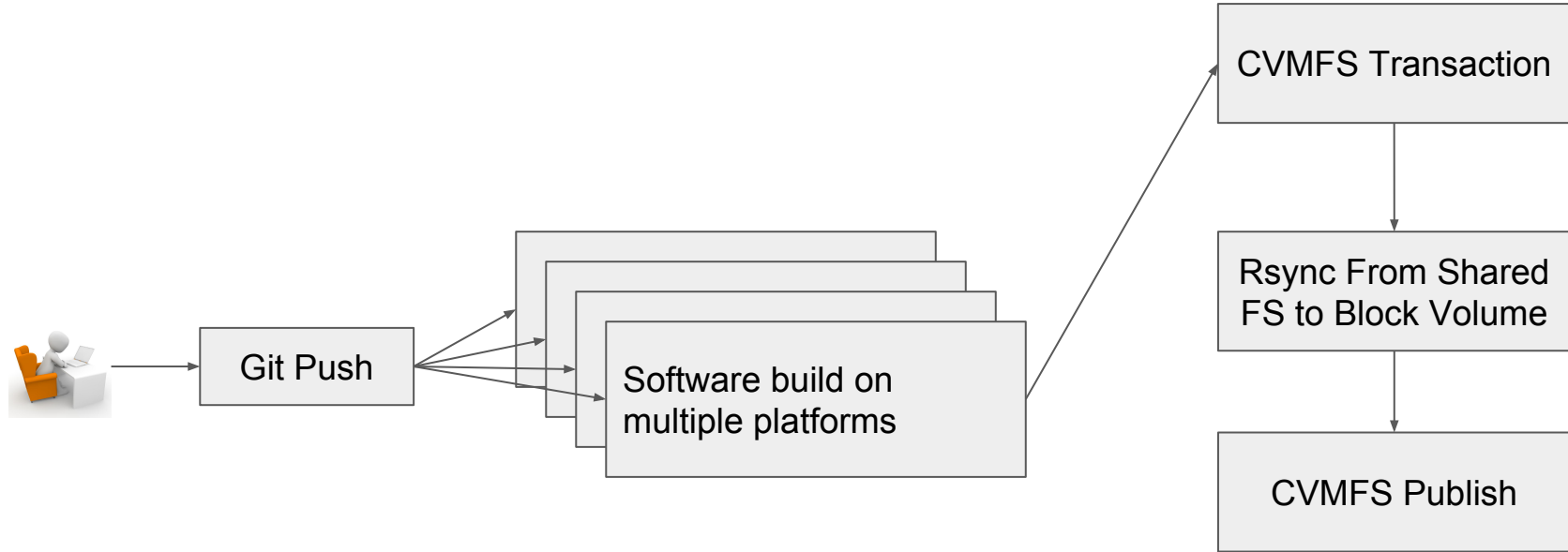
[Buildbot](#) - Build pipeline software. Pick your poison. Jenkins would work too.

Git - Good way to track versions, bugs, and history

New Setup



Buildbot Workflow



Kubernetes Examples

https://github.com/WIPACrepo/cvmfs_buildbot/tree/master/kubernetes

Shared Storage

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

 name: cvmfs-buildbot-worker-pv-claim

spec:

 accessModes:

 - ReadWriteMany

 persistentVolumeReclaimPolicy: Retain

 #StorageClass: cephfs

 resources:

 requests:

 storage: 1Ti

Stratum0 Worker Deployment

https://raw.githubusercontent.com/WIPACrepo/cvmfs_buildbot/master/kubernetes/worker-cvmfs-centos7-stratum0.json

Buildbot Examples

```
factory = util.BuildFactory()
    factory.addStep(steps.Git(
        repourl='git://github.com/WIPACrepo/cvmfs.git',
        mode='full',
        method='clobber',
        workdir='build',
    ))
    factory.addStep(steps.ShellCommand(
        name='build cvmfs',
        command=[
            'python', 'builders/build.py',
            '--src', 'icecube.opensciencegrid.org',
            '--dest', '/cvmfs/icecube.opensciencegrid.org',
            '--variant', util.Property('variant'),
        ],
        env={
            'CPUS': util.Property('CPUS', default='1'),
            'MEMORY': util.Property('MEMORY', default='1'),
        },
        workdir='build',
        haltOnFailure=True,
        locks=[
            cfg.locks['cvmfs_shared'].access('exclusive')
        ],
    ))
```

```
factory = util.BuildFactory()
    factory.addStep(steps.ShellCommand(
        name='open transaction',
        command=['cvmfs_server', 'transaction', 'icecube.opensciencegrid.org'],
        haltOnFailure=True,
    ))
    factory.addStep(steps.ShellCommand(
        name='rsync',
        command=[
            'cvmfs_rsync',
        ],
    ))
    util.Interpolate('/cvmfs-source/icecube.opensciencegrid.org/${prop:variant}s'),
    util.Interpolate('/cvmfs/icecube.opensciencegrid.org/${prop:variant}s'),
    ],
    haltOnFailure=True,
    doStepIf=BuildPassed,
    ))
    factory.addStep(steps.ShellCommand(
        name='publish transaction',
        command=['cvmfs_server', 'publish', 'icecube.opensciencegrid.org'],
        haltOnFailure=True,
        doStepIf=BuildPassed,
        hideStepIf=lambda results, s: results==SKIPPED,
    ))
    factory.addStep(steps.ShellCommand(
        name='abort transaction',
        command=['cvmfs_server', 'abort', '-f', 'icecube.opensciencegrid.org'],
        haltOnFailure=True,
        doStepIf=BuildFailed,
        hideStepIf=lambda results, s: results==SKIPPED,
    ))
```

Improvements

