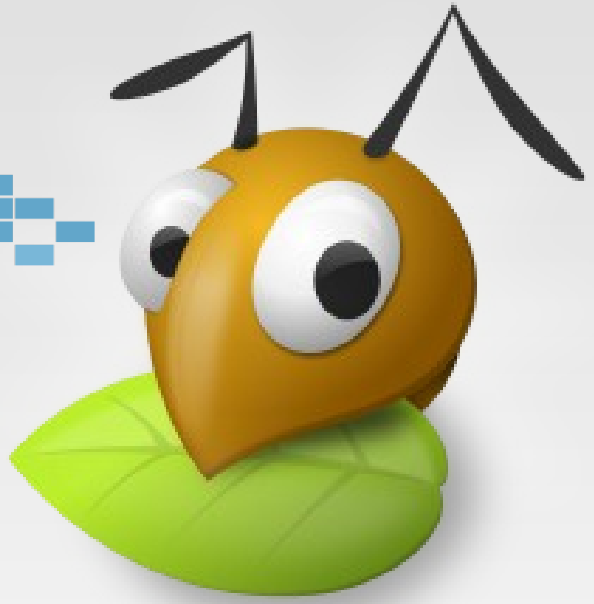
MOUNTPOINT



Approaches for duplicating Kubernetes Storage with Gluster

Niels de Vos
Gluster Developer and Maintainer
Senior Software Engineer at Red Hat
ndevos@redhat.com

Agenda

- Brief Introduction into Gluster
- Kubernetes Storage Basics
- Use-Cases for Cloning
- Initiating Cloning with Container Data Importer
- Limitations of Cloning with Gluster Volume Snapshots
- Improved Cloning based on reflinking
- Future with Container Storage Interface

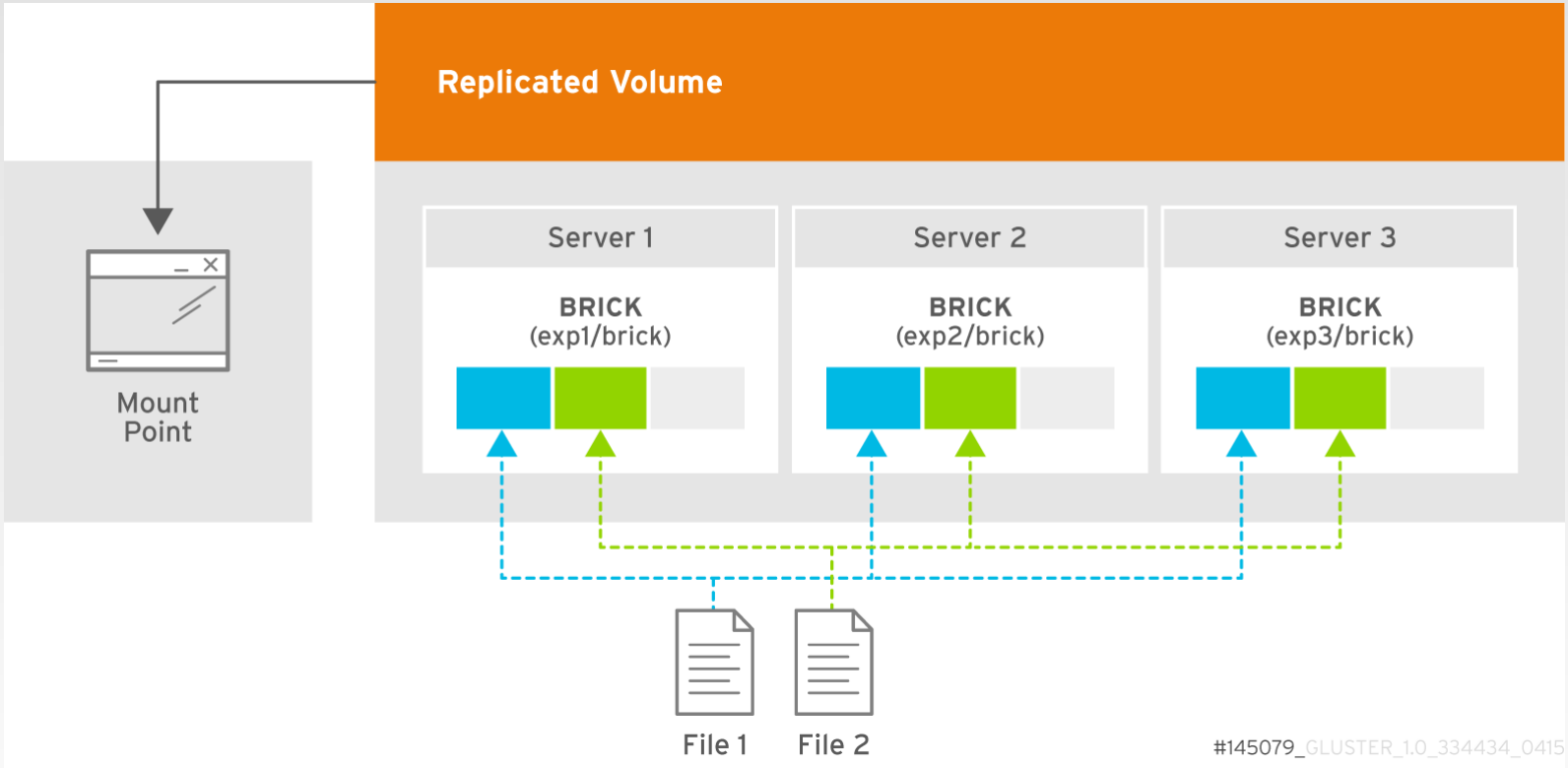
Introduction into Gluster

- Software Defined Storage
- Scale-out, distributed and high-available
- Designed as a filesystem
  - Block Storage as an add-on
  - Object Storage as an add-on
- Easy to install, configure and maintain
  - Packages available for several Linux distributions

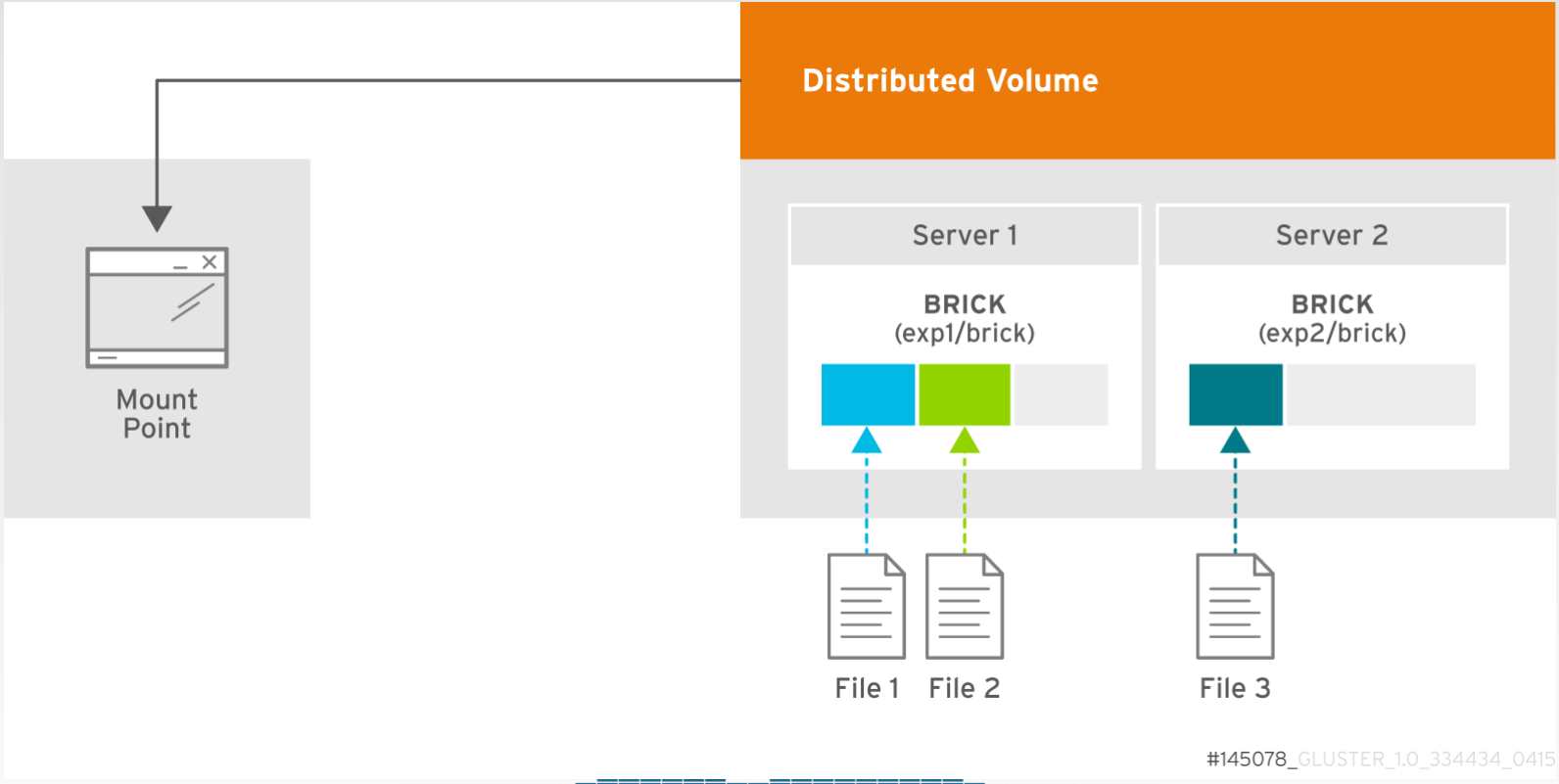
Introduction into Gluster

- High Availability
  - 3-way Replication
  - 2-way Replication + Arbiter
  - Dispersed Volumes
- Scalability
  - Distributed Volumes

3-way Replication



Distributed Volumes



Commonly Used Features

- Meta-data caching
- Geo-replication
- Volume snapshots
- Policy based split-brain resolution
- Brick multiplexing
- Sharding

Related Open Source Projects

- oVirt
- Samba
- NFS-Ganesha
- QEMU
- Kubernetes
- glusterfs-coreutils
- Gluster-colonizer
- ...

Kubernetes Storage Basics

- StorageClass
- PersistentVolumes
- PersistentVolumeClaim

- Containerized Data Importer

StorageClass

```
---
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: glusterfile
provisioner: gluster.org/glusterfile
parameters:
  resturl: "http://heketi.default.svc.cluster.local:8080"
  restuser: admin
```


PersistentVolumeClaim

```
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: store-the-data-here
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: "glusterfile"
  resources:
    requests:
      storage: 1Gi
```

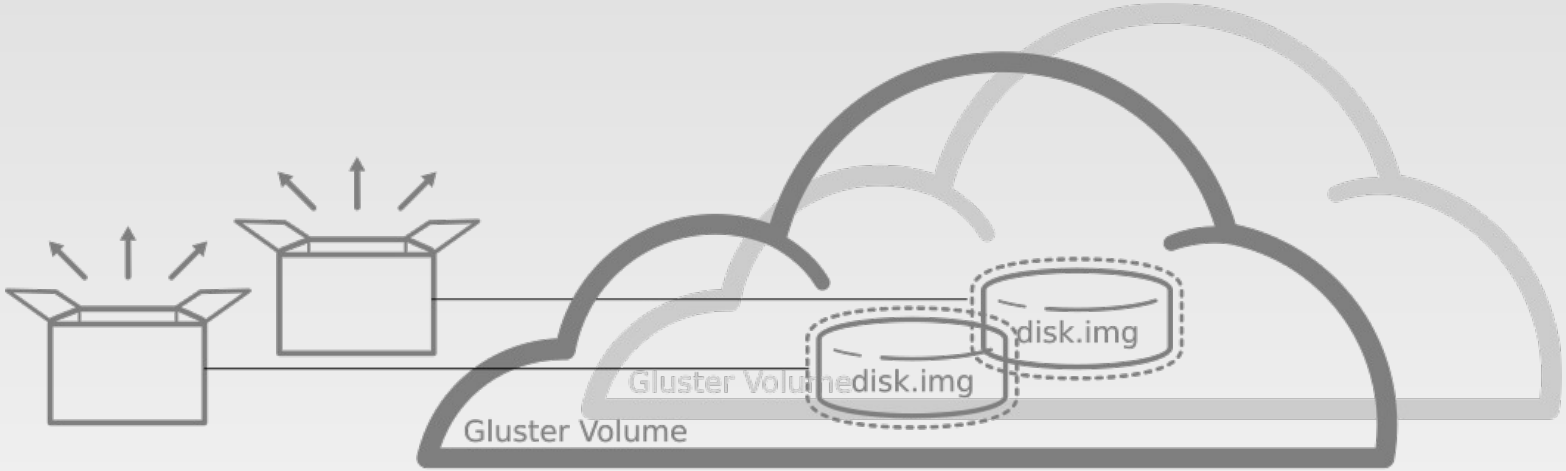

Use-Cases for Cloning

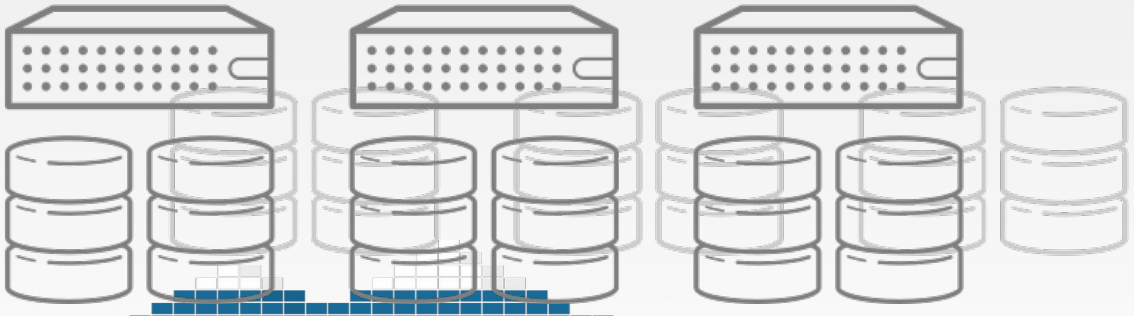
- Testing maintenance operations
  - Upgrading applications, data conversion
  - Migration from storage provider
- Scaling access to (readonly) data
- Base population for consumption and modification
  - ‘golden’ data for similar applications or instances
- Backup and archiving

Initiating Cloning with CDI

- SmartCloning
  - Exposed as a feature in the StorageClass
  - Based on annotations for a PVC
  - Creates a new PVC, with `CloneOf` annotation

SmartCloning





Initiating Cloning with CDI

- Host-Assisted Cloning
  - ‘dumb’ recursive copy between PVCs
  - Fallback when missing ‘CloneOf’ annotation

Host-Assisted Cloning

Limitations with Gluster Volume Snapshots

- Whole volumes only
- Gluster management operation
- LVM inherited limits
  - Thin-pool configuration and monitoring
- No (re)distribution over storage servers

Improved Cloning based on Reflink

- File cloning on modern filesystems
- Copy-on-write semantics
- Standard `copy_file_range()` syscall
- On Gluster likely also through `setxattr()`
- Local filesystems only, restricted to bricks

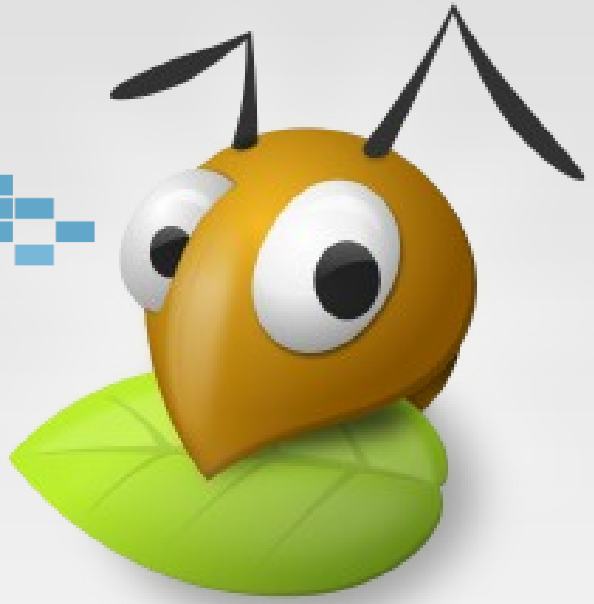
Improved Cloning based on Reflink

- PVCs based on sub-directories
- Clone of a PVC becomes a recursive copy
- Need to be careful with rebalance

Next: Container Storage Interface

- CSI-Snapshot Controller for Kubernetes
  - Uses DataSource object in PVC description

- Proposed feature for CSI and Kubernetes
  - https://github.com/container-storage-interface/spec/pull/244
  - https://github.com/kubernetes/community/pull/2533



Thanks!

Niels de Vos – ndevos@redhat.com

https://people.redhat.com/ndevos/talks/2018-08-mountpoint/