dCache and Openstack.

Deployment of dCache Services on Openstack

Christian Voß

13th International dCache workshop

May 19, 2019

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

Content

1. Overview of DESY dCache Heads and Doors
2. Openstack Infrastructure
3. Prototype Cloud dCache Setup

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

dCache installations at DESY Hamburg

- For the raw numbers on resources and pools see yesterday's talk
- Focus on heads and doors instead of pools

Head Nodes

- Each installation consists of three head nodes
- All services run in high availability

Door Nodes

- Each installation consists independent door nodes
- Doors nodes serve a purpose such as
  - grid-doors: `gsi-FTP`, `gsi-DCap`, `dCap`

dCache Head Nodes

dCache-dir	dCache-core	dCache-se

dCache HA Services

Core-Domain
PnfsManager
PoolManager
srmManager
spaceManager
gPlazma
PinManager

dCache Unique Services

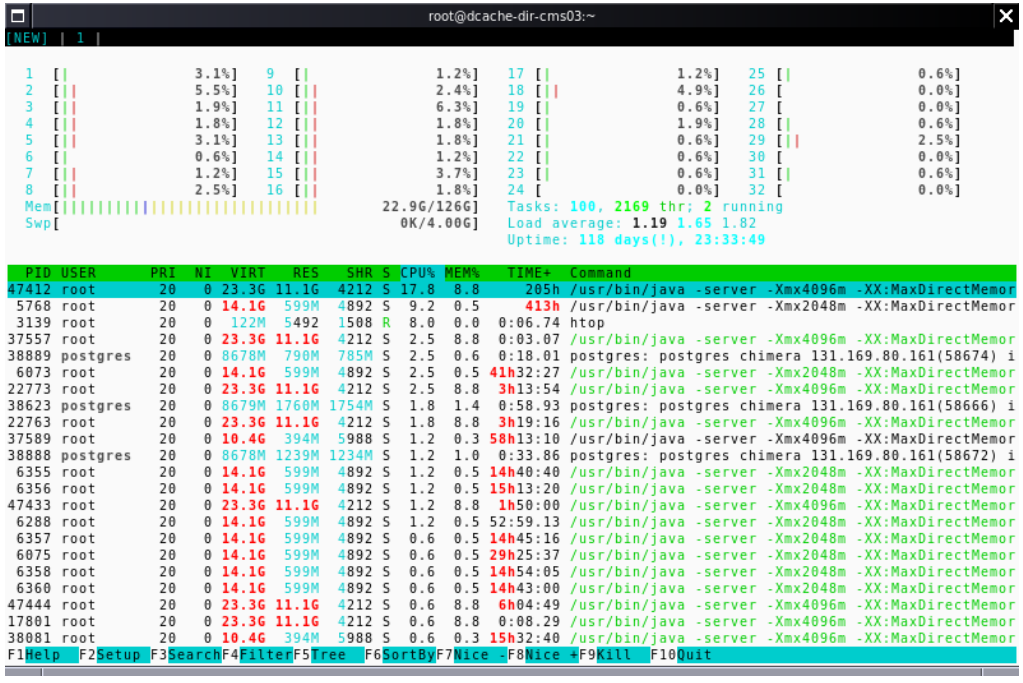
NFS Door	Admin Door	SRM
Cleaner	NFS Door	Billing
Resilient		HTTP
		Frontend

Other Services

Zookeeper	Zookeeper	Zookeeper
Chimera Master	Chimera Slave	SRM/Pin Master

dCache Head Nodes Loads

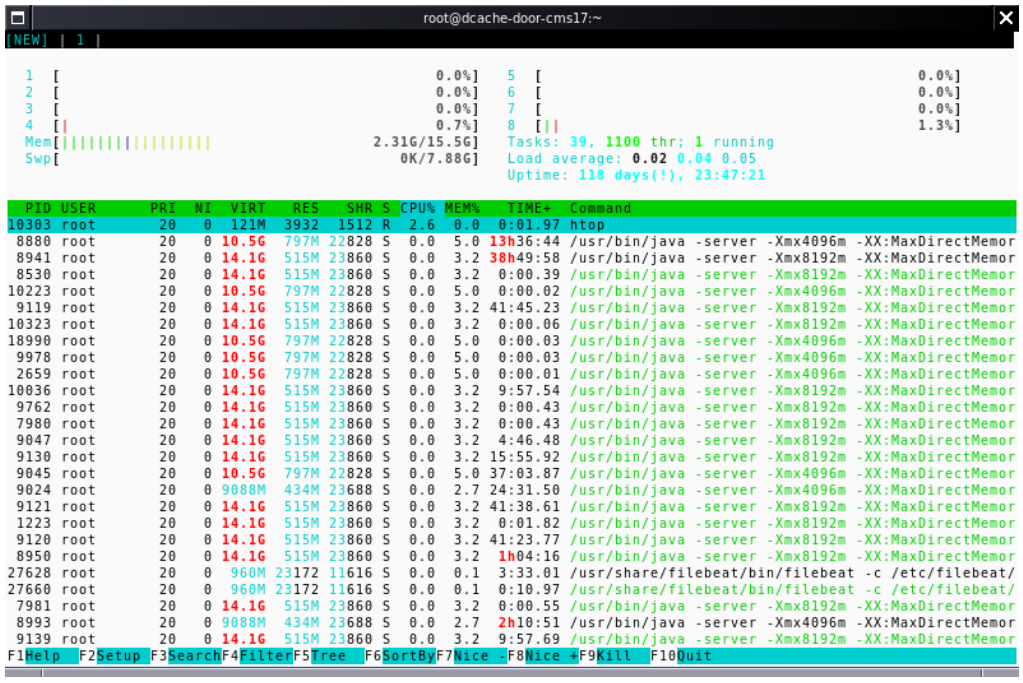
- General purpose machines with large amounts of RAM for `postgres`
- Machines have minimal load in normal operation
- Chimera database is stored in memory



- Updates/Reboots in sync with dCache updates

dCache Doors and Door Loads

- Small general purpose machines
- Even less load and more idle CPUs



- Updates/Reboots in sync with dCache updates

Openstack and dCache Deployment

Why are we talking about Openstack?

- Head nodes run out of their five-year-warranty
- Door nodes up to eight years old
- Buy new nodes with a lot of idle computing resources?
- Hardware studies last summer without convincing results
- DESY-IT spent quite some money on extending Openstack

Why not move over to Openstack?

Technical Implications

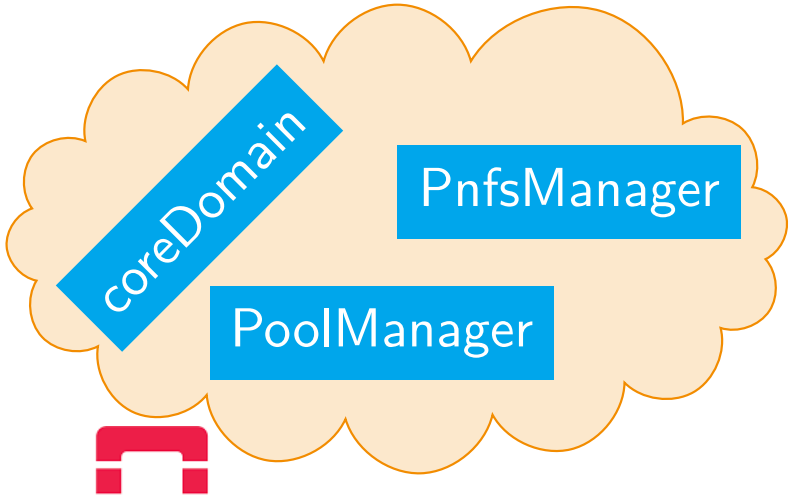
Pros

- Microservice architecture → one service in one domain
- One service in one domain → one domain on one vm
- Limit resources per Domain (JVM) → similar to Openstack flavours
- Native redundancy thanks to Openstack infrastructure
- More flexible life cycles

Cons

- Virtualisation of dCache pools pointless
- Virtualisation of data base servers pointless
- Configuration management

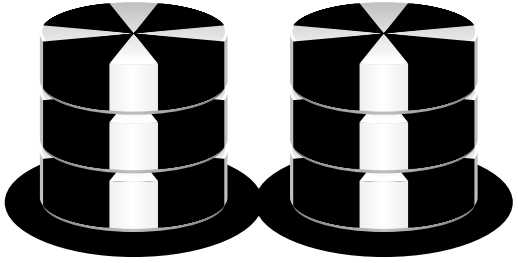
dCache Cloud Deployment Schema



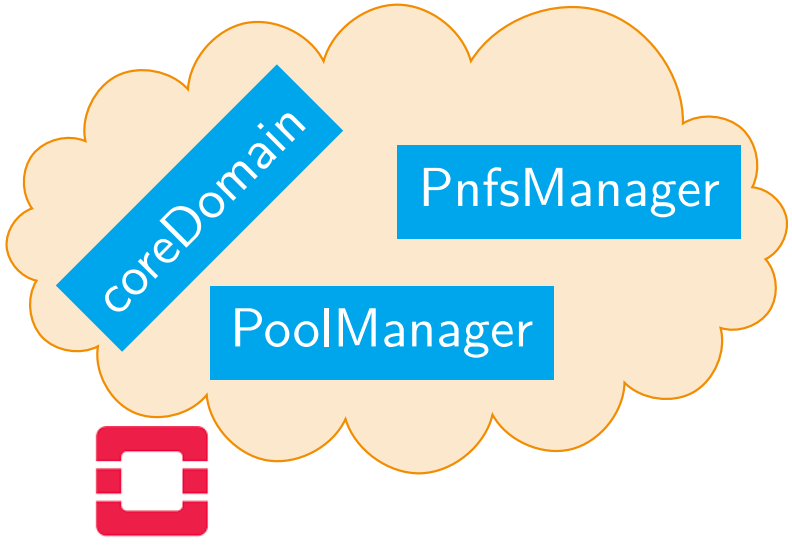
openstack.

Dedicated Database Server

dCache Pools



dCache Cloud Deployment Schema



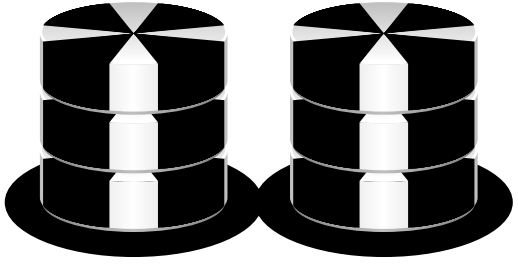
openstack.

Dedicated Database Server

Issues to be tackled

- dCache/Zookeeper Deployment
- Configuration/integration into DESY-IT
- Networking and security

dCache Pools



Content

1. Overview of DESY dCache Heads and Doors
2. Openstack Infrastructure
3. Prototype Cloud dCache Setup

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

Openstack at DESY

> DESY currently runs two independent Openstack installations

openstack.desy.de

- Prototype installation
- Limited, out of warranty hardware resources
- Non optimised CEPH storage
- Not project based
- Part of the DESY intranet
- Ports controlled by Openstack

cc.desy.de

- Pre-production installation
- Usage as demonstrator
- Sizeable hardware resources
- Dedicated CEPH infrastructure
- Fully project based
- Not part of the intranet
- Ports controlled via central security office based on IPs

Openstack at DESY

- DESY currently runs two independent Openstack installations

openstack.desy.de

- Prototype installation
- Limited, out of warranty hardware resources
- Non optimised CEPH storage
- Not project based
- Part of the DESY intranet
- Ports controlled by Openstack

cc.desy.de

- Pre-production installation
- Usage as demonstrator
- Sizeable hardware resources
- Dedicated CEPH infrastructure
- Fully project based
- Not part of the intranet
- Ports controlled via central security office based on IPs

Tackling Issues – Deployment/Configuration

Classical World

This is, where we are

- Thinking node/role based
- Take care of full stack
- Automate full stack

Cloud World

This is, where others are

- Thinking service based
- Take care of their application
- Automate their stack
- Rely on automation of others

Combining Both Worlds

> Can we bring both worlds together?

Deployment

> For this prototype deploy static dCache
> Use VMs and not containers
> Use Openstack::HEAT template for deployment

Openstack::HEAT Overview

HEAT is the internal `Openstack` orchestration module

- No external dependencies
- Syntax is `yaml` based
- HEAT templates describe the complete architecture of a *stack*
- Allows to use resources from other `Openstack` modules
- Allows to define
  - Machines, block devices and device maps
  - Networks and routing
  - Define security groups (network control lists)
  - Configuration via `cloud-init` scripts (installed on VMs)
- Whole stack created at once as one entity
- Whole stack removed at once as one entity

Content

1. Overview of DESY dCache Heads and Doors
2. Openstack Infrastructure
3. Prototype Cloud dCache Setup

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

Openstack::HEAT for dCache

Openstack::HEAT Template

- Describes a compute stack in term of resources

```
message_Domain01:
  type: OS::Nova::Server
  properties:
    name : dot-message01
    image: CentOS-7-x86_64-GenericCloud-1809
    flavor: m1.xlarge
    key_name: dot-administrator
    security_groups:
      - default
      - dCache
    networks:
      - network : DESY-VLAN-240
    user_data_format: SOFTWARE_CONFIG
    user_data: {get_resource: server_init}

java_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      yum install -y java-1.8.0-openjdk.x86_64

server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

Openstack::HEAT for dCache

Openstack::HEAT Template

- Describes a compute stack in term of resources
- Different resource types

```
message_Domain01:
  type: OS::Nova::Server
  properties:
    name : dot-message01
    image: CentOS-7-x86_64-GenericCloud-1809
    flavor: m1.xlarge
    key_name: dot-administrator
    security_groups:
      - default
      - dCache
    networks:
      - network : DESY-VLAN-240
    user_data_format: SOFTWARE_CONFIG
    user_data: {get_resource: server_init}

java_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      yum install -y java-1.8.0-openjdk.x86_64

server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

Openstack::HEAT for dCache

Openstack::HEAT Template

- Describes a compute stack in term of resources
- Different resource types
- Allows configuration of
  - Machine characteristics

```
message_Domain01:
  type: OS::Nova::Server
  properties:
    name : dot-message01
    image: CentOS-7-x86_64-GenericCloud-1809
    flavor: m1.xlarge
    key_name: dot-administrator
    security_groups:
      - default
      - dCache
    networks:
      - network : DESY-VLAN-240
    user_data_format: SOFTWARE_CONFIG
    user_data: {get_resource: server_init}

java_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      yum install -y java-1.8.0-openjdk.x86_64

server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

Openstack::HEAT for dCache

Openstack::HEAT Template

- Describes a compute stack in term of resources
- Different resource types
- Allows configuration of
  - Machine characteristics
  - Network

```
message_Domain01:
  type: OS::Nova::Server
  properties:
    name : dot-message01
    image: CentOS-7-x86_64-GenericCloud-1809
    flavor: m1.xlarge
    key_name: dot-administrator
    security_groups:
      - default
      - dCache
    networks:
      - network : DESY-VLAN-240
    user_data_format: SOFTWARE_CONFIG
    user_data: {get_resource: server_init}

java_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      yum install -y java-1.8.0-openjdk.x86_64

server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

Openstack::HEAT for dCache

Openstack::HEAT Template

- Describes a compute stack in term of resources
- Different resource types
- Allows configuration of
  - Machine characteristics
  - Network
  - Network control lists

```
message_Domain01:
  type: OS::Nova::Server
  properties:
    name : dot-message01
    image: CentOS-7-x86_64-GenericCloud-1809
    flavor: m1.xlarge
    key_name: dot-administrator
    security_groups:
      - default
      - dCache
    networks:
      - network : DESY-VLAN-240
    user_data_format: SOFTWARE_CONFIG
    user_data: {get_resource: server_init}

java_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      yum install -y java-1.8.0-openjdk.x86_64

server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

Openstack::HEAT for dCache

Openstack::HEAT Template

- Describes a compute stack in term of resources
- Different resource types
- Allows configuration of
  - Machine characteristics
  - Network
  - Network control lists
  - Access control lists

```
message_Domain01:
  type: OS::Nova::Server
  properties:
    name : dot-message01
    image: CentOS-7-x86_64-GenericCloud-1809
    flavor: m1.xlarge
    key_name: dot-administrator
    security_groups:
      - default
      - dCache
    networks:
      - network : DESY-VLAN-240
    user_data_format: SOFTWARE_CONFIG
    user_data: {get_resource: server_init}

java_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      yum install -y java-1.8.0-openjdk.x86_64

server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

Openstack::HEAT for dCache

Openstack::HEAT Template

- Describes a compute stack in term of resources
- Different resource types
- Allows configuration of
  - Machine characteristics
  - Network
  - Network control lists
  - Access control lists
  - Service configuration

```
message_Domain01:
  type: OS::Nova::Server
  properties:
    name : dot-message01
    image: CentOS-7-x86_64-GenericCloud-1809
    flavor: m1.xlarge
    key_name: dot-administrator
    security_groups:
      - default
      - dCache
    networks:
      - network : DESY-VLAN-240
    user_data_format: SOFTWARE_CONFIG
    user_data: {get_resource: server_init}

java_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      yum install -y java-1.8.0-openjdk.x86_64

server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

Openstack::HEAT dCache Installations

From blank machines to a running dCache

> Deeper look at `server_init` resource

```
server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

> Install dCache and its dependencies

```
dcache_install:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ungrouped
    config: |
      #!/usr/bin/bash
      wget https://www.dcache.org/downloads/1.9/repo/5.0/dcache-5.0.8-1.noarch.rpm -O /tmp/dc
      yum install -y /tmp/dcache-latest.rpm
```

Openstack::HEAT dCache Configuration

From blank machines to a running dCache

> Deeper look at `server_init` resource

```
server_init:
  type: OS::Heat::MultipartMime
  properties:
    parts:
    - config: {get_resource: system_init}
    - config: {get_resource: java_install}
    - config: {get_resource: dcache_install}
    - config: {get_resource: dcache_http_conf}
```

> Configure dCache services

- Prepare `/etc/dcache/dcache.conf` to get layout via web server

```
[root@dot-poolmanager01 ~]# cat /etc/dcache/dcache.conf
dcache.role=dot-poolmanager
dcache.layout.uri=http://dcache-dot1:2288/layouts/${dcache.role}.conf
[root@dot-poolmanager01 ~]# dcache status
DOMAIN                                 STATUS                PID   USER    LOG
dot-poolmanager01_poolmanagerDomain running (for 3 days) 5762 dcache /var/log/dcache/dot-pool
```

- Other config files cannot be streamed

Non-Streamable Configuration Files

- Concerned files (missing out on some):
  - `/etc/dcache/gplazma.conf`
  - `/etc/dcache/gss.conf`
  - `/etc/dcache/info-provider.xml`
  - `/etc/dcache/storage.xml` (CMS)
  - `/etc/grid-security/grid-vorolemap`
  - `/etc/grid-security/storage-authzdb`
  - `/etc/dcache/admin/users/acls/*.*.*`
  - `/etc/dcache/admin/authorized_keys2`
  - Host certificates
- Need to be rolled out directly onto the domain nodes
- Separate file for each installation
- Generally already managed by puppet

Configuration File Distribution

> All files should be managed centrally ideally via puppet

`dcache-os-dot` prototype

> Files stored on dedicated web server
> Download depending on role via bash magic

Magical Christmas Land

> `dCache` supports `urls` for all concerned files

Proper Orchestration or Configuration Management

> Use Existing `Puppet` infrastructure to deploy configs centrally
> Distribution and setup by *e.g.* `Kubernetes` using the central configs

Dependant Services for dCache

dCache relies on a couple of external services

`Zookeeper`, `PostgreSQL`, `Kafka`, Grid Security package

Right now distributed among head and door nodes

Dependant Services for dCache

dCache relies on a couple of external services

`Zookeeper`, `PostgreSQL`, `Kafka`, Grid Security package

Right now	distributed among head and door nodes
`PostgreSQL`	Deployment on dedicated database hosts per installation
`Kafka`	A dedicated `Kafka` cluster for all instances
`Zookeeper`	Switch to a single cluster run on the `Kafka` cluster
Host Cert	Use Wild card certificates from configuration server
Grid Security	Ensure packages installation in `system_init` `HEAT` step

Dependant Services for dCache

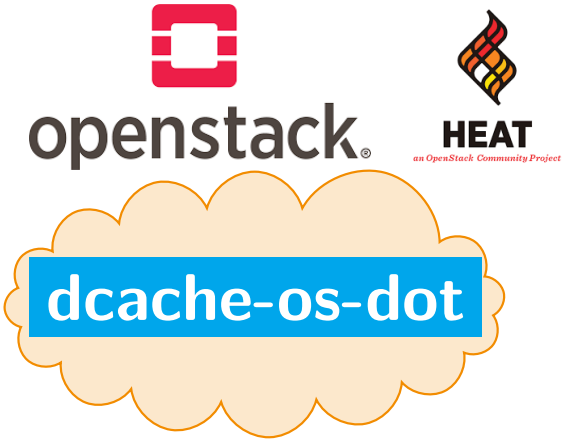
dCache relies on a couple of external services

`Zookeeper`, `PostgreSQL`, `Kafka`, Grid Security package

Right now	distributed among head and door nodes
`PostgreSQL`	Deployment on dedicated database hosts per installation
`Kafka`	A dedicated `Kafka` cluster for all instances
`Zookeeper`	Switch to a single cluster run on the `Kafka` cluster
Host Cert	Use Wild card certificates from configuration server
Grid Security	Ensure packages installation in `system_init` `HEAT` step

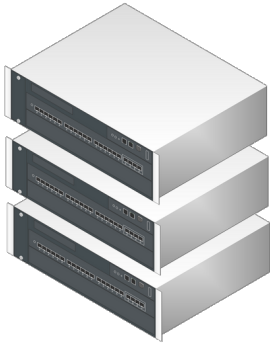
> All steps applied to the DOT `dCache`

Modified DOT dCache deployment



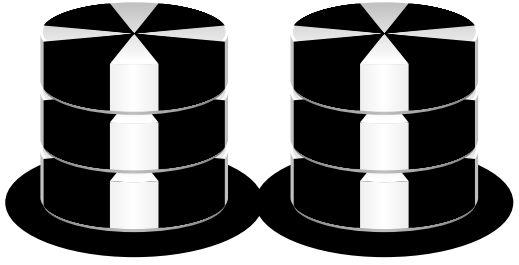
Dedicated Database Server

dot-support-cluster



- `Kafka` cluster (HA with 3 nodes)
- `Zookeeper`
- Webserver for configs

dCache Pools



Modified DOT dCache deployment

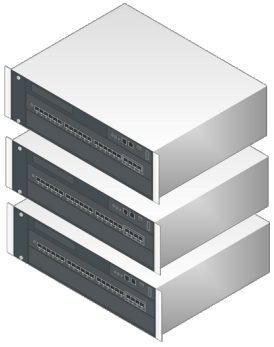
openstack

HEAT
an OpenStack Community Project

dcache-os-dot

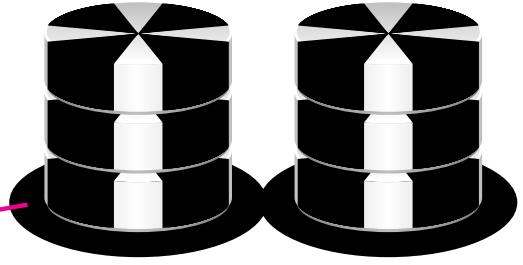
Dedicated Database Server

dot-support-cluster



- `Kafka` cluster (HA with 3 nodes)
- `Zookeeper`
- Webserver for configs

dCache Pools



DESY internal network

Modified DOT dCache deployment

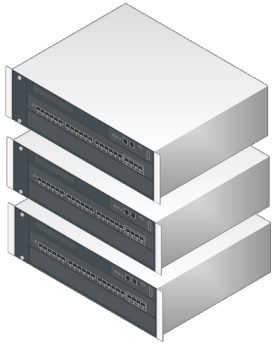
openstack

HEAT
an OpenStack Community Project

dcache-os-dot

sec_conf

dot-support-cluster

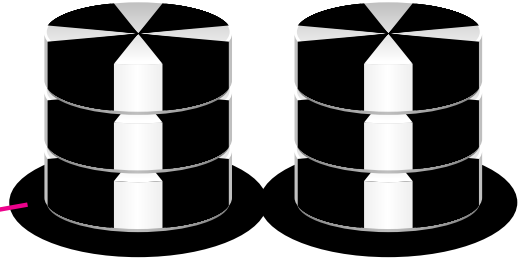


- Kafka cluster (HA with 3 nodes)
- Zookeeper
- Webserver for configs

Dedicated Database Server

PostgreSQL

dCache Pools



DESY internal network

DESY.

Modified DOT dCache deployment

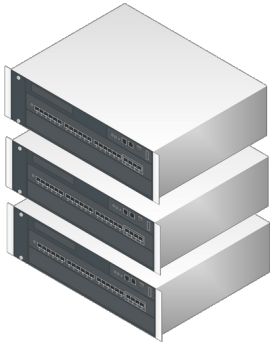
dcache-os-dot

Dedicated Database Server

sec_pgsql

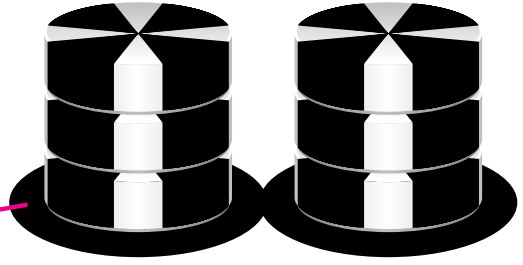
sec_conf

dot-support-cluster



- Kafka cluster (HA with 3 nodes)
- Zookeeper
- Webserver for configs

dCache Pools



DESY internal network

Modified DOT dCache deployment

Dedicated Database Server

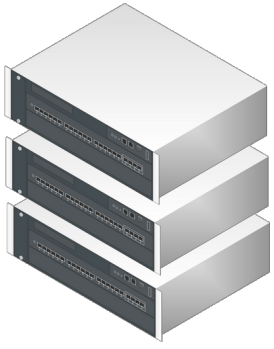
dcache-os-dot

sec_pgsql

sec_conf

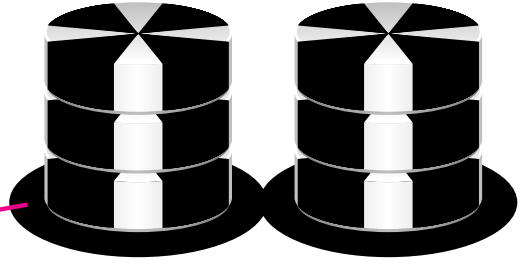
sec_pool

dot-support-cluster



- Kafka cluster (HA with 3 nodes)
- Zookeeper
- Webserver for configs

dCache Pools



DESY internal network

First Experiences and Conclusions

First Impressions

> Working prototype with ongoing performance studies

Future Development

Future Development

Requirements from `Openstack` infrastructure

- Full production status of `Openstack`
- Designated project based configuration
- Strategy on dealing with network control lists

Requirements from dcache-operations

- Automated configuration management
- Automated machine handling
- Smart integration into existing infrastructure