



# OpenNebula

Martin Alfke

<ma@example42.com>



# Martin Alfke

- ✦ Puppet Nerd
- ✦ Freelancer
- ✦ Automation Enthusiast
- ✦ kein Berater !





# E-POST

- ✦ 2500 VM's
- ✦ mehrere Virtualisierungslösungen (OpenVZ, MCP, KVM libvirt)
- ✦ unterschiedliche Lösungen für Dev und Ops





# Agenda

- ✦ OpenNebula
- ✦ Backends
- ✦ Aufbau
- ✦ Komponenten
- ✦ Workflow
- ✦ Management (GUI/CLI/Puppet)





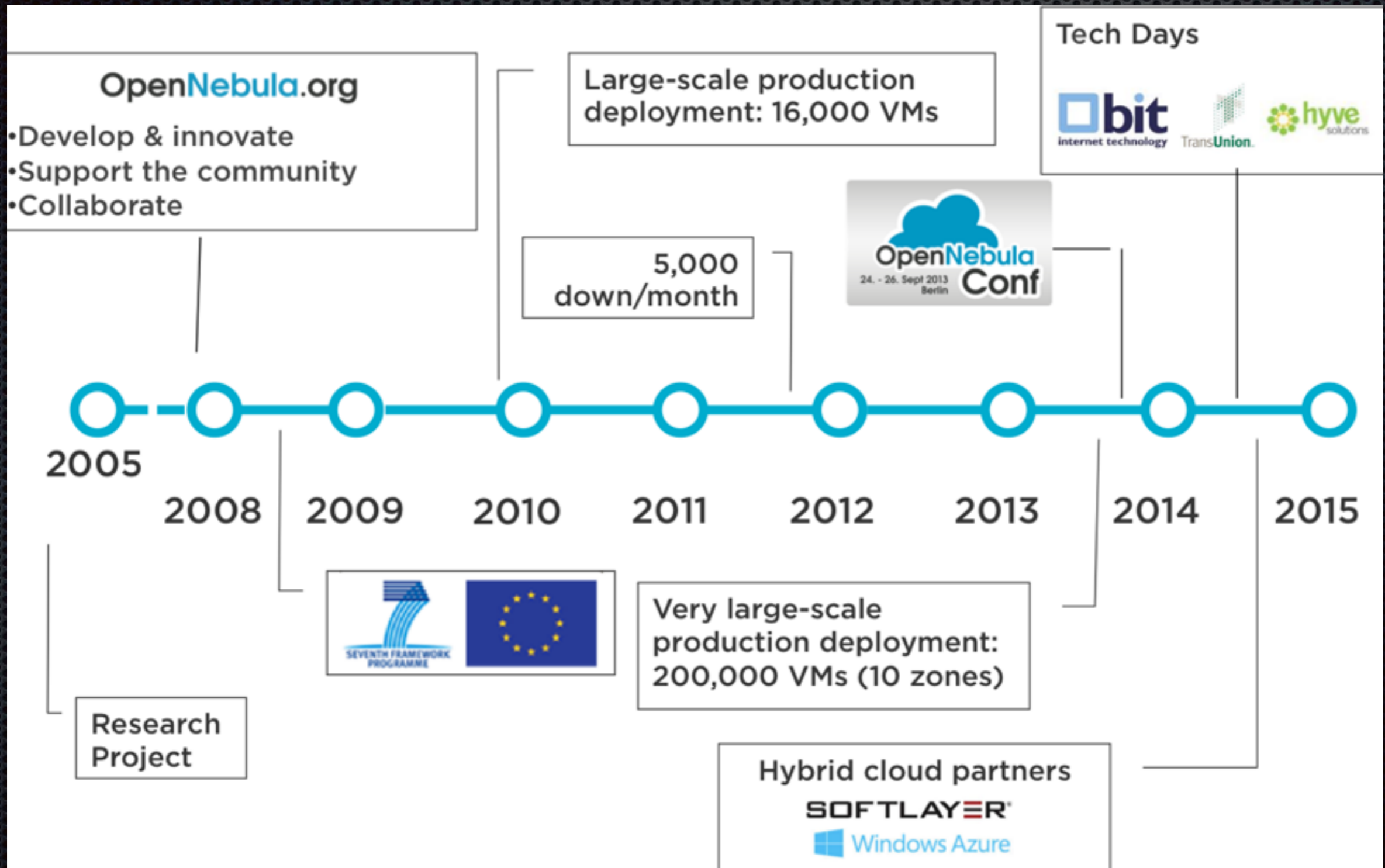
# OpenNebula

- ✦ Europäische Cloud Lösung (Forschungsprojekt Uni Madrid 2005)
- ✦ Zentraler Steuerungshost für alle Komponenten (Storage, Netzwerk, Virtualisierer, VMs)





# OpenNebula





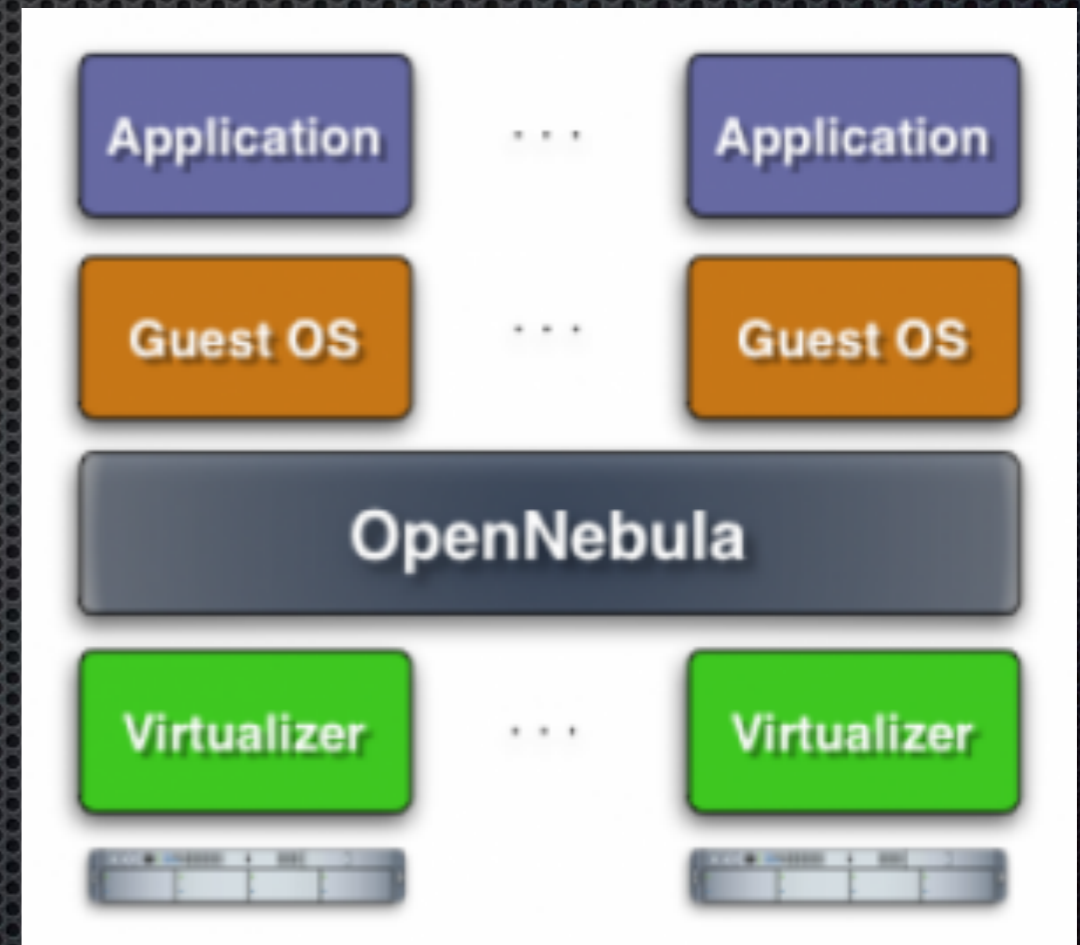
# OpenNebula

- ✦ eine Komponente (OpenNebula Management Host)
- ✦ kein Kunden/Tenant System
- ✦ User + Gruppen + Berechtigungen + Quota
- ✦ private Cloud + Cloud Bursting



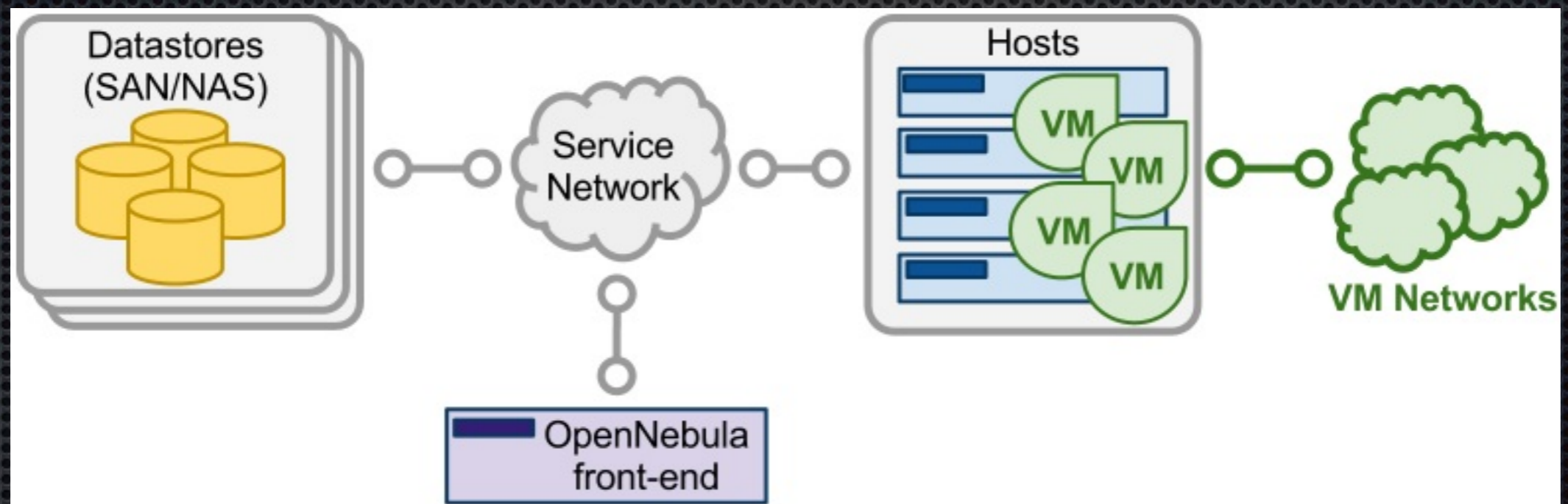
# OpenNebula

- ✦ Management von Virtualisierern
- ✦ Management von Netzwerk Interfaces
- ✦ Management von VM's





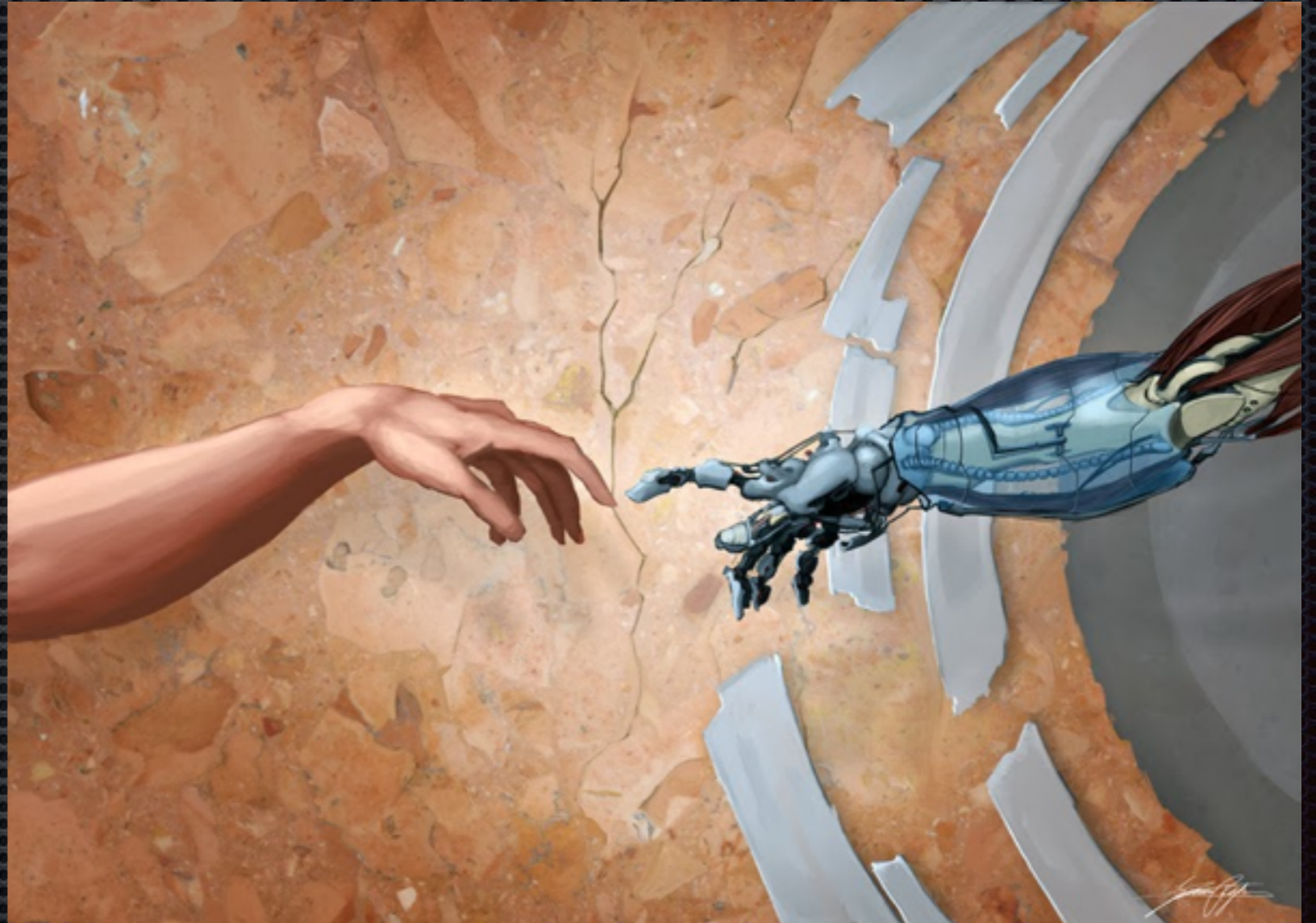
# OpenNebula





# Backends

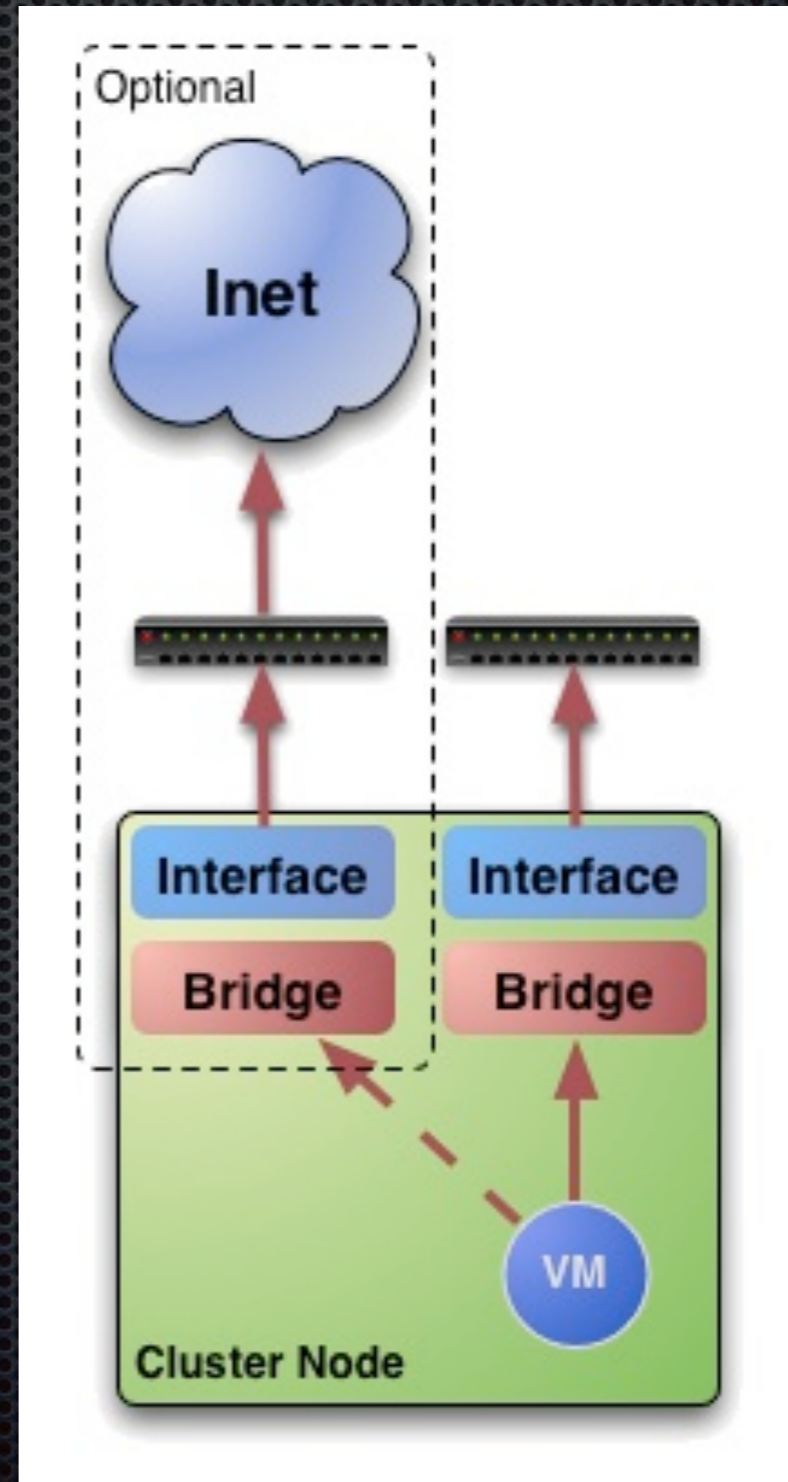
- ✦ Netzwerk
- ✦ Storage
- ✦ Virtualisierung
- ✦ Image





# Netzwerk Backends

- ✦ 802.1q
- ✦ ebtables / iptables
- ✦ ovswitch
- ✦ dummy (NAT)





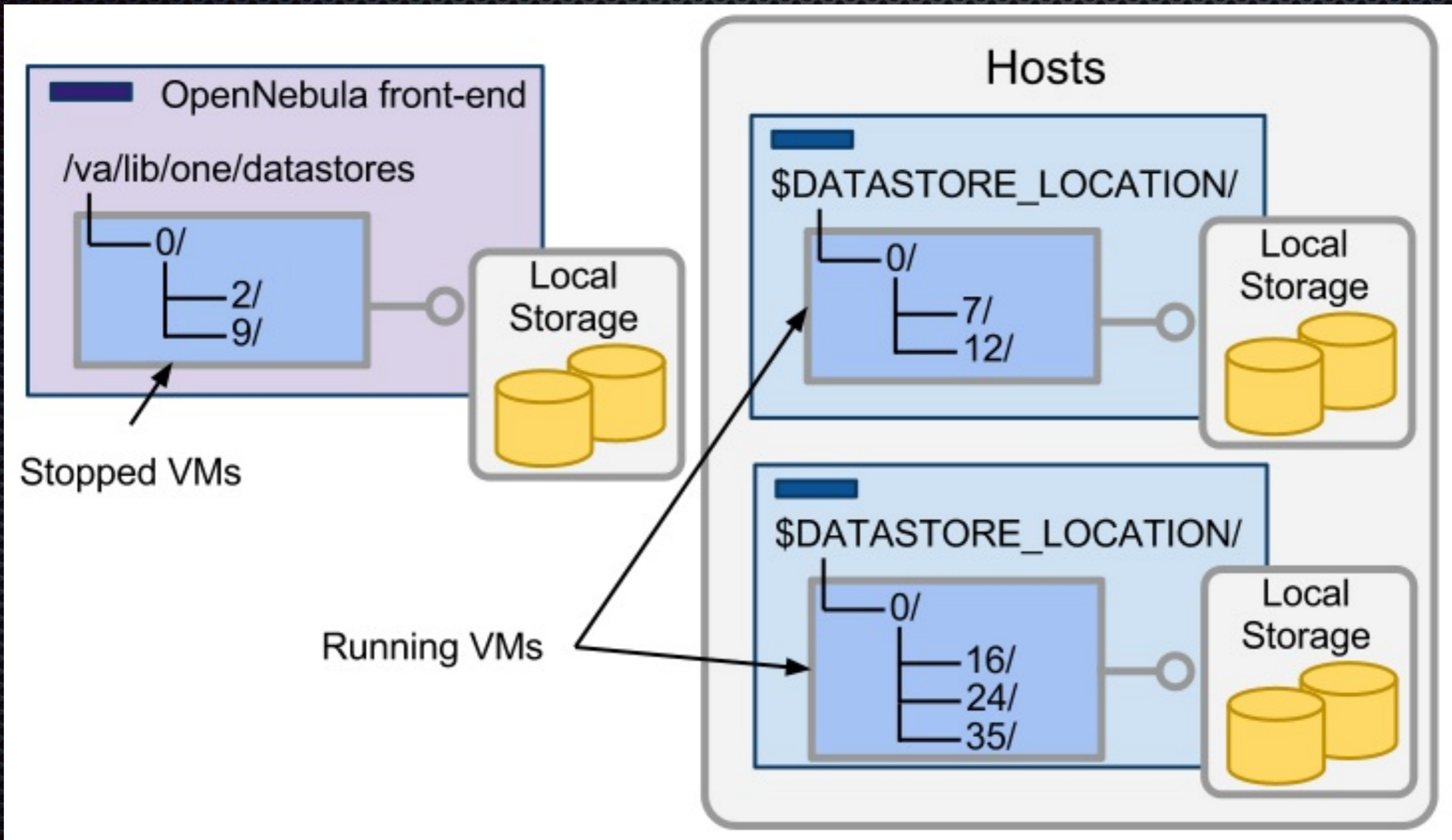
# Storage Backends

- ✦ Lokaler Storage
- ✦ NFS
- ✦ Ceph
- ✦ Gluster
- ✦ iSCSI (Community Plugin)



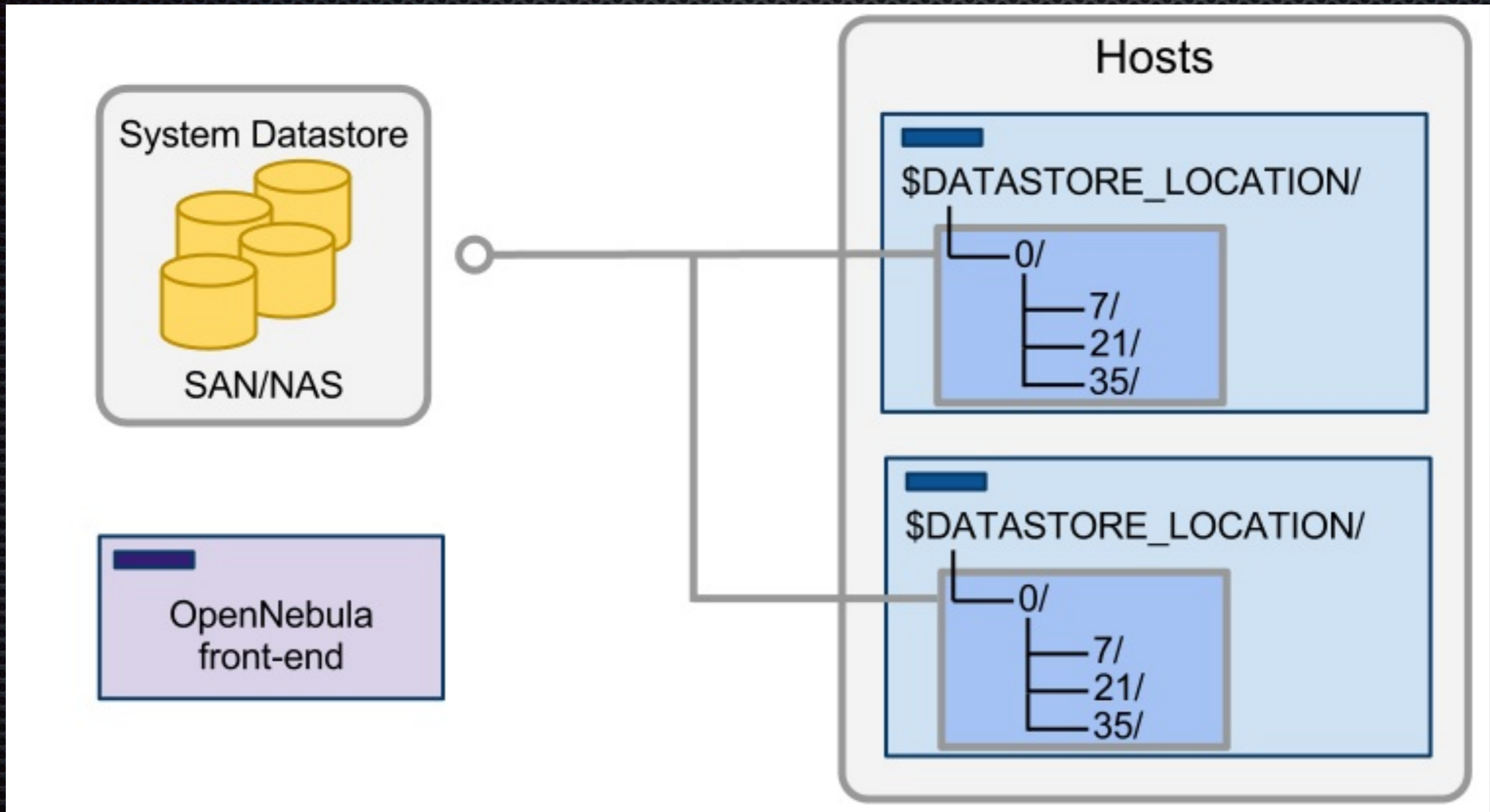


# Storage Backends - Lokal



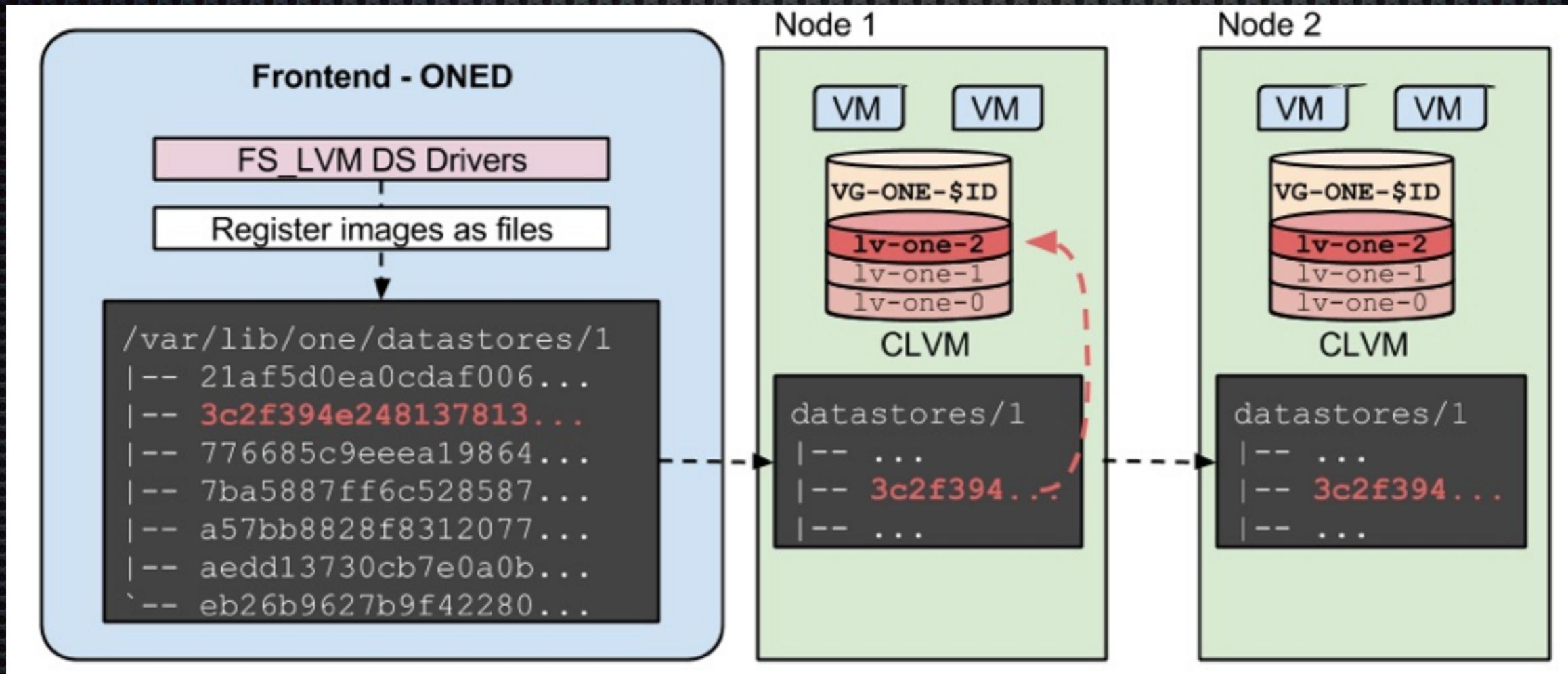


# Storage Backends - Shared





# Storage Backends - LVM





# Virt Backends

- ✦ KVM
- ✦ VMware
- ✦ Xen
- ✦ Azure
- ✦ Public Clouds (AWS/Rackspace/...)





# Image Backends

- ✦ LVM
- ✦ qcow2/raw
- ✦ Ceph





# Aufbau

- ✦ Planung
  - ✦ Cluster
  - ✦ Storage
  - ✦ Over Commitment





# Aufbau - Cluster

- ✦ Trennung und Gruppierung von Virtualisierungen
- ✦ Verteilung nach Kriterien:
  - ✦ Anzahl von VM's
  - ✦ Last
  - ✦ Individuelle User Berechtigungen



# Aufbau - Cluster

Web Development



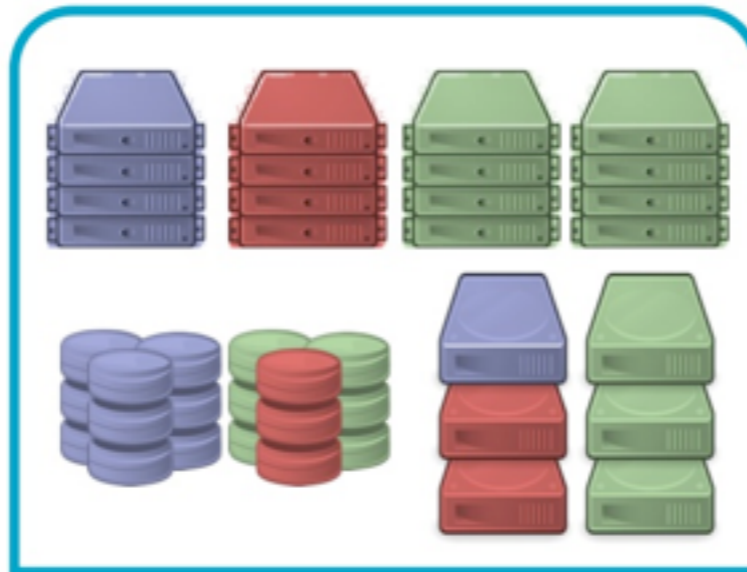
Human Resources



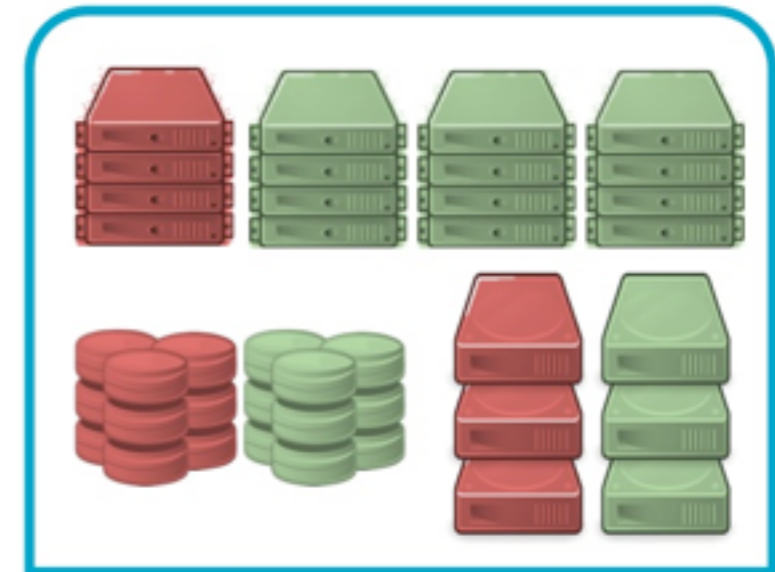
Big Data Analysis



Public Clouds



DC West Coast



DC Europe



# Aufbau - Storage

- ✦ Lokale Storage - VM Images müssen via SSH kopiert werden
- ✦ Shared Image Storage - VM Images werden lokal kopiert (cp)
- ✦ Shared OpenNebula Installation - VM Images werden gelinkt (ln -s)



# Aufbau - Overcommitment

- ✦ RAM Over Commitment
  - ✦ benötigt KSM
  - ✦ benötigt eine Code Änderung in OpenNebula (reserved RAM)
- ✦ CPU Over Commitment
  - ✦ vCPU im Template



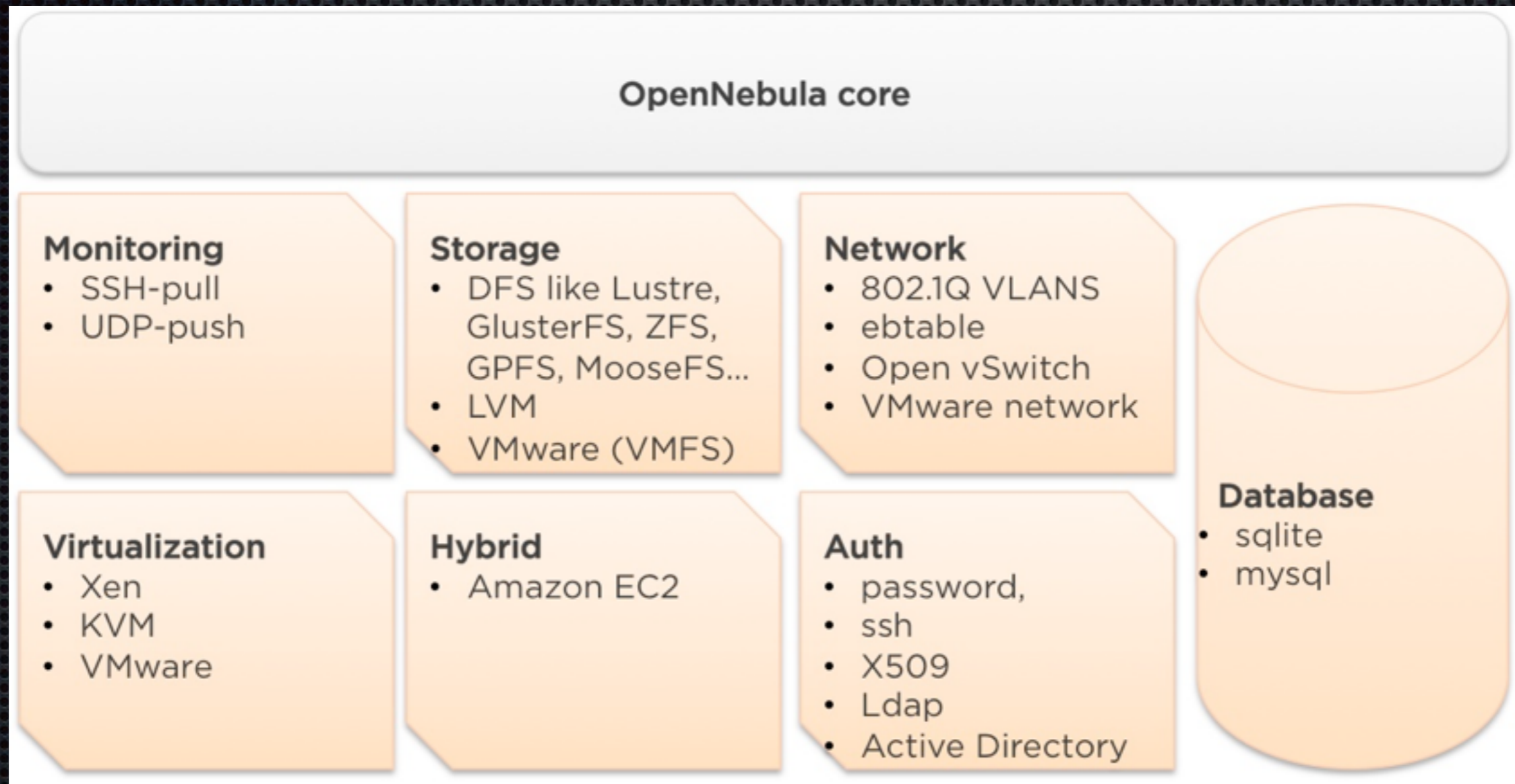
# Komponenten

- ✦ Management Node
- ✦ Virtualisierer





# Komponenten - Übersicht





# Komponenten - Mgmtnode

- ✦ Opennebula (notwendig)
- ✦ Apache + Passenger (sunstone web gui - optional)
- ✦ VNC Proxy (optional)
- ✦ MySQL DB (notwendig - sqlite für Tests)
- ✦ User: oneadmin + ssh key (notwendig)



# Komponenten - Virtualisierer

- ✦ libvirt
- ✦ User: oneadmin + authorized key file
- ✦ sudoers (bridgetcl, ...)
- ✦ policykit (RH) / User Berechtigungen für /dev/qemu (Debian)



# Workflow

- ✦ Netzwerk
- ✦ Image
- ✦ Kontextualisierung
- ✦ Template
- ✦ VM
- ✦ Hooks





# Workflow - Netzwerk

- ✦ 802.1q
- ✦ ebtables
- ✦ open vswitch
- ✦ dummy (nat)





# Workflow - Image

- ✦ Muss vorab angefertigt (packer) oder vom Marketplace geladen werden
- ✦ Braucht one-context Package oder cloud-init (v0.7.3 oder neuer)
- ✦ Hinweis: Image erst auf Management Node kopieren



# Workflow - Kontextualisierung

- ✦ Anpassungen des Images (hostname, Netzwerk Konfiguration, ssh pub key, ...)
- ✦ Kann durch eigene Skripte erweitert werden
- ✦ VM erhält eine CD-ROM mit den Kontext Informationen

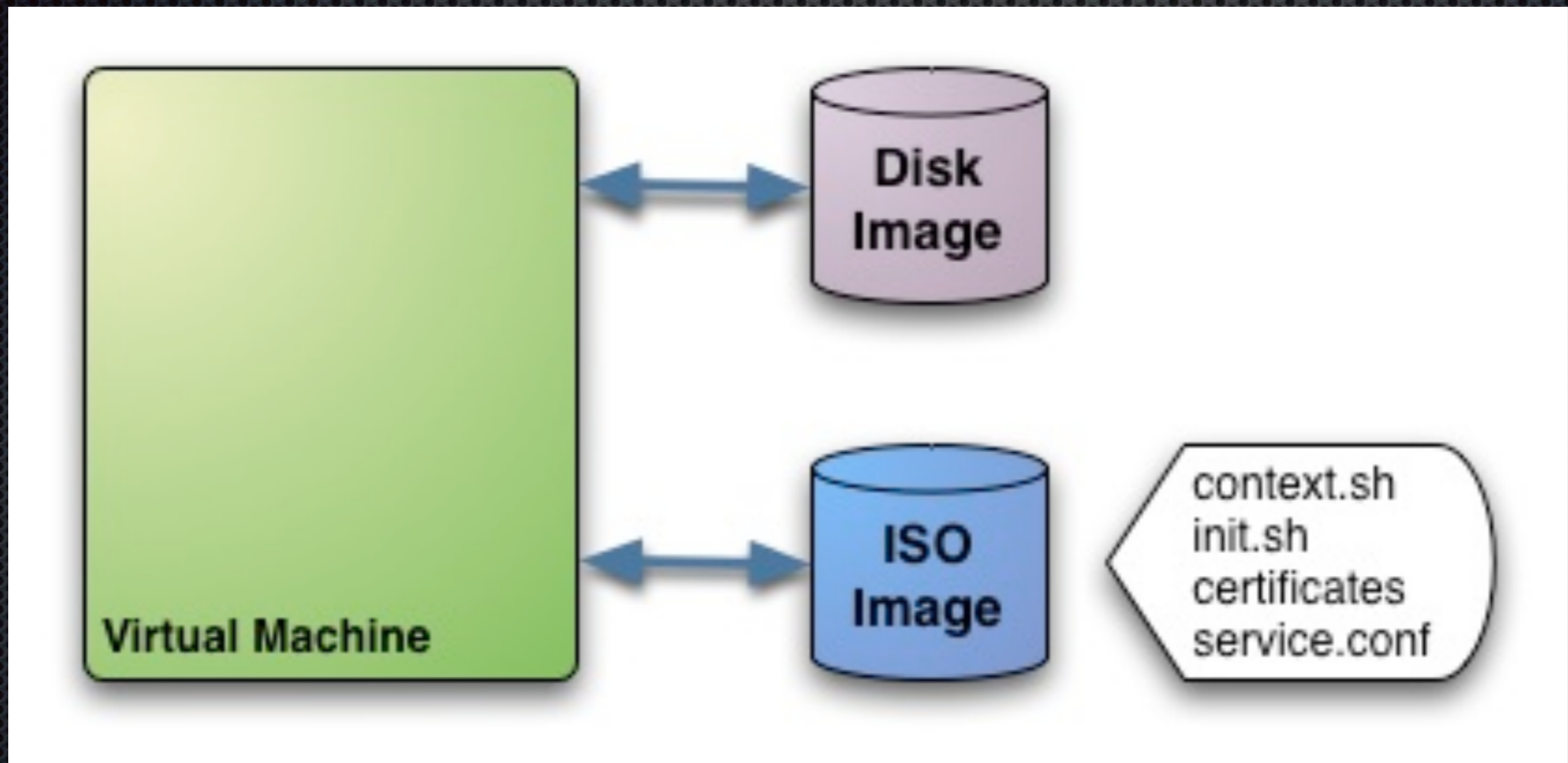


# Workflow - Kontextualisierung

```
# Context variables generated by OpenNebula
DISK_ID='1'
ETH0_DNS='10.175.4.11'
ETH0_GATEWAY='10.175.16.1'
ETH0_IP='10.175.24.1'
ETH0_MASK='255.255.240.0'
ETH0_NETWORK='10.175.16.0'
FILES_DS='/var/lib/one/datastores/2/
ef48a35be2d8723ef7788175be6087eb:\''init.sh'\'' '
HOSTNAME='martin.xxxxxxxxxx'
NETWORK='YES'
TARGET='hdb'
```



# Workflow - Kontextualisierung





# Workflow - Template

- ✦ Zusammenfassung von Netzwerk, CPU, RAM und Image
- ✦ Hardware Spezifika (z.B. ACPI, Architektur)
- ✦ Dient als Vorlage für VM



# Workflow - VM

- VM wird auf Basis eines Templates initialisiert



# Workflow - Hooks

- ✦ Hooks werden bei bestimmten VM Stati ausgeführt (Prolog, Boot, Running, ...)
- ✦ DNS (nsupdate), Puppet Zertifikatsmanagement, PuppetDB remove
- ✦ Anmelden einer VM an einem Dritt-System



# Workflow - Hooks

```
#!/bin/bash
```

```
# häßliches Bash Script
```

```
set -e
```

```
NAME=`onevm show $1 | grep HOSTNAME | cut -d '"' -f2`
```

```
DOMAIN=`echo $NAME | cut -d '.' -f2-`
```

```
curl -k -X DELETE -H "Accept: pson" https://puppet.  
$DOMAIN:8140/production/certificate_status/$NAME
```



# GUI / CLI

- ✦ Sunstone Webinterface
- ✦ CLI tools
- ✦ XMLRPC Interface
- ✦ Puppet





# GUI / CLI

- ✦ GUI
  - ✦ Sunstone Web Interface
  - ✦ Anbindung an LDAP möglich
  - ✦ Verwendet ebenfalls eine Token Datei



# GUI / CLI

The screenshot displays the OpenNebula Sunstone GUI dashboard. On the left is a navigation sidebar with the following menu items: Dashboard, System, Virtual Resources, Virtual Machines, Templates, Images, Files & Kernels, Infrastructure, and OneFlow. The main dashboard area is titled "Dashboard" and shows the user "oneadmin".

Key metrics are displayed in summary cards:

- Storage:** 44 IMAGES, 104.9GB USED
- Users:** 16 USERS, 2 GROUPS
- Network:** 7 VNETS, 115 USED IPs
- Hosts:** 5 TOTAL, 5 ON, 0 OFF, 0 ERROR
- Virtual Machines:** 97 TOTAL, 96 ACTIVE, 1 PENDING, 0 FAILED

Two line graphs are shown for Hosts:

- CPU:** A line graph showing CPU usage over time from 12:23 to 14:03. The y-axis ranges from 0 to 15000. Three lines are plotted: Allocated (red), Real (blue), and Total (grey).
- MEMORY:** A line graph showing memory usage over the same time period. The y-axis ranges from 0KB to 572.2GB. Three lines are plotted: Allocated (red), Real (blue), and Total (grey).

Two more line graphs are shown for Virtual Machines:

- NET DOWNLOAD SPEED:** A line graph showing network download speed from 12:23 to 13:21. The y-axis ranges from 0B/s to 9.5MB/s.
- NET UPLOAD SPEED:** A line graph showing network upload speed from 12:23 to 13:21. The y-axis ranges from 0B/s to 14.3MB/s.

Legend for the Hosts graphs: Allocated (red), Real (blue), Total (grey).

OpenNebula 4.2.0 by C12G Labs.



# GUI / CLI

The screenshot displays the OpenNebula Sunstone interface. On the left is a navigation sidebar with categories like Dashboard, System, Virtual Resources, Infrastructure, and OneFlow. The main content area is titled 'Hosts' and shows a summary of 5 total hosts, all of which are 'ON'. Below this is a table listing individual hosts with their IDs, names, clusters, RVMs, CPU, and memory usage. Host ID 14 is selected. Below the table are two line graphs: 'CPU' and 'MEMORY', both showing 'Allocated', 'Real', and 'Total' usage over time. The CPU graph shows a total usage of approximately 2400 units, while the memory graph shows a total usage of 95.4GB. The interface also includes a search bar, a 'Create' button, and a 'Delete' button.

**OpenNebula Sunstone**

Hosts 5 TOTAL 5 ON 0 OFF 0 ERROR oneadmin

Create Delete More Search

ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
0	srv-211.█	BISS-Cluster	14	1400 / 2400 (58%)	76GB / 70.8GB (107%)	ON
2	srv-212.█	BISS-Cluster	26	700 / 2400 (29%)	60GB / 70.8GB (85%)	ON
14	srv-213.█	BISS-Cluster	13	325 / 2400 (14%)	42GB / 70.8GB (59%)	ON
17	srv-220.█	BISS-Cluster	25	700 / 2400 (29%)	82.5GB / 126GB (65%)	UPDATE
21	srv-219.█	BISS-Cluster	13	600 / 2400 (25%)	46GB / 126GB (37%)	ON

Showing 1 to 5 of 5 entries

Information Graphs

**CPU**

3000  
2000  
1000  
0

13:33 16:20 19:06 21:53 00:40 03:26 06:13 09:00

Allocated Real Total

**MEMORY**

95.4GB  
71.5GB  
47.7GB  
23.8GB  
0KB

13:33 16:20 19:06 21:53 00:40 03:26 06:13 09:00

Allocated Real Total

OpenNebula 4.2.0 by C12G Labs.



# GUI / CLI

The screenshot displays the OpenNebula Sunstone web interface. The top header shows 'Virtual Machines' with a summary: 94 TOTAL, 91 ACTIVE, 0 OFF, 1 PENDING, 0 FAILED. A user profile 'oneadmin' is visible in the top right. A sidebar on the left contains navigation menus for Dashboard, System, Virtual Resources, Infrastructure, and OneFlow. The main content area features a table of Virtual Machines and a detailed view for a selected VM.

ID	Owner	Group	Name	Status	Host	IPs	VNC
24	malfke	users	martin.██████████	RUNNING	srv-219.██████████	10.175.24.8	
47	oneadmin	oneadmin	jenkins-build-01.██████████	RUNNING	srv-211.██████████	10.175.24.21	
48	oneadmin	oneadmin	jenkins-build-02.██████████	RUNNING	srv-211.██████████	10.175.24.22	
49	oneadmin	oneadmin	jenkins-build-03.██████████	RUNNING	srv-211.██████████	10.175.24.23	
50	oneadmin	oneadmin	dashboard-01.██████████	RUNNING	srv-220.██████████	10.175.24.24	

Virtual Machine - martin.██████████	
ID	24
Name	martin.██████████
State	ACTIVE
LCM State	RUNNING
Host	srv-219.██████████
Start time	16:30:02 08/16/2013
Deploy ID	one-24
Reschedule	no

Permissions:	Use	Manage	Admin
Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Other	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ownership	
Owner	malfke
Group	users

Tags	
FOOBAR	test
ERROR	Tue Oct 1 23:10:25 2013 : Error deploying

OpenNebula 4.2.0 by C12G Labs.



# GUI / CLI

- ✦ CLI
  - ✦ User braucht eine Token Datei ~/.one/one\_auth
  - ✦ Verschiedene Kommandos (one\*)



# GUI / CLI

onehost list

onehost show <id>

onehost create <fqdn> -i <im> -v <vm> -n <net>

onehost create one-virt -i kvm -v qemu -n dummy



# GUI / CLI - XMLRPC

```
#!/usr/bin/python
# based on script of Chris Usher June 2009
# import xmlrpclib re hashlib

server = 'http://<opennebula>:2633/RPC2'
user = "oneadmin"
password = "<password>"
one_auth = '{0}:{1}'.format(user, password)

def getVMInfo(id):
    response = xmlrpclib.ServerProxy(server).one.vm.info(one_auth,id)
    return response[1]

print getVMInfo(24)
```



# GUI / CLI - Puppet

```
onehost { '<fqdn>':  
  ensure    => present,  
  im_mad    => 'kvm',  
  vm_mad    => 'kvm',  
  vn_mad    => 'vlan', # ← 802.1q  
}
```

```
onevm { '<vm_name>':  
  ensure    => present,  
  tempate   => '<template_name>',  
}
```



# Demo







# Fragen

Martin Alfke

<ma@example42.com>