



IT-Infrastruktur außerhalb des Docker-/K8-/Openstack-Hypes

03.12.19

Jens Schanz



Jens Schanz

Teamleiter 2nd Level Support
„Linux- und Filialsysteme“

Linux- / Unix-Admin seit 1999

Senior System Engineer und
Infrastructure Architect

✉ jens.schanz@mueller.de

• [@jensschanz](#)

Firmenname: Müller Holding GmbH & Co. KG

Firmensitz (Verwaltung): Albstraße 92,
89081 Ulm-Jungingen

Geschäftsführer: Erwin Müller, Dr. Günther Helm

Zahl der Mitarbeiter: rund 35.000 (überwiegend
Fachkräfte)

Zahl der Auszubildenden: rund 950

Gesamtzahl Filialen: **derzeit 859, davon 561 in
Deutschland, 56 in der Schweiz, 89 in
Österreich, 13 in Spanien, 18 in Slowenien, 38
in Ungarn, 84 in Kroatien**

Filialen mit Naturshop: derzeit 292 davon 173 in
Deutschland, 15 in der Schweiz, 52 in Österreich,
10 in Spanien, 8 in Slowenien 16 in Ungarn und 18
in Kroatien

Filialgröße: 400 bis über 4.500 m² Verkaufsfläche

Gesamtlagerfläche: 246.416 m², davon 17.998 m²
Lager Ungarn, 9.251 m² Lager Schweiz, 1.080 m²
Lager Spanien

Arealgröße Konzern: 516.945 m²

Abteilungen / Fachmärkte

Drogerie (ca. 50.000 Artikel)

Multi-Media (ca. 42.000 Artikel)

Parfümerie (ca. 28.000 Artikel)

Spielwaren (ca. 20.000 Artikel)

Schreibwaren (ca. 19.000 Artikel)

Haushalt und Ambiente (ca. 11.000 Artikel)

Strümpfe (ca. 7.500 Artikel)

Naturkosmetik (ca. 4.000 Artikel)

Handarbeit (ca. 2.400 Artikel)

OTC (ca. 1.500 Artikel)

Bio Nahrung (ca. 3.000 Artikel)

Sortimentsvielfalt:

ca. 190.000 Artikel

Müller-IT in Zahlen

„Typische Mittelstands-IT“

2 Infrastruktur-Teams (~ 30 Personen) verwalten u.a.

- 3 Rechenzentren am Standort Ulm
 - + 3 Logistikzentren in Europa
- ~ 150 physikalische Server
- ~ 650 aktive Netzwerkkomponenten
- ~ 1800 virtuelle Linux-Systeme in Produktion
- ~ 500 virtuelle Windows-Systeme
- ~ 250 virtuelle Linux-Systeme für Test und Q/A
- ~ 1PB SAN- und NAS-Storage

„Typische Mittelstands-IT“

Über 200 unterschiedliche Applikationen unterstützen die Geschäftsprozesse ...

- 75% Eigenentwicklung bzw. Spezialsoftware
- 25% Standard-Software

„Typische Mittelstands-IT“

Eines der Infrastruktur-Teams (15 Personen) kümmert sich auch noch ...

- ~ 830 Filialen in Europa
- ~ 17.000 Systeme (Linux-Arbeitsplätze, Kassen)
- ~ 1.800 Router
- ~ 2.400 Switches
- ~ 2.450 Drucker

Herausforderungen

Full-Stack-Administration

- **Netzwerk**

- Tagged / Untagged VLANs, Bonding, Routing ...

- **Storage**

- Fibre-Channel, iSCSI, LUNs, WWIDs ...

- **Bare-Metall**

- UEFI, Legacy-BIOS, Multipath, Volume-Manager ...

- **Virtualisierung**

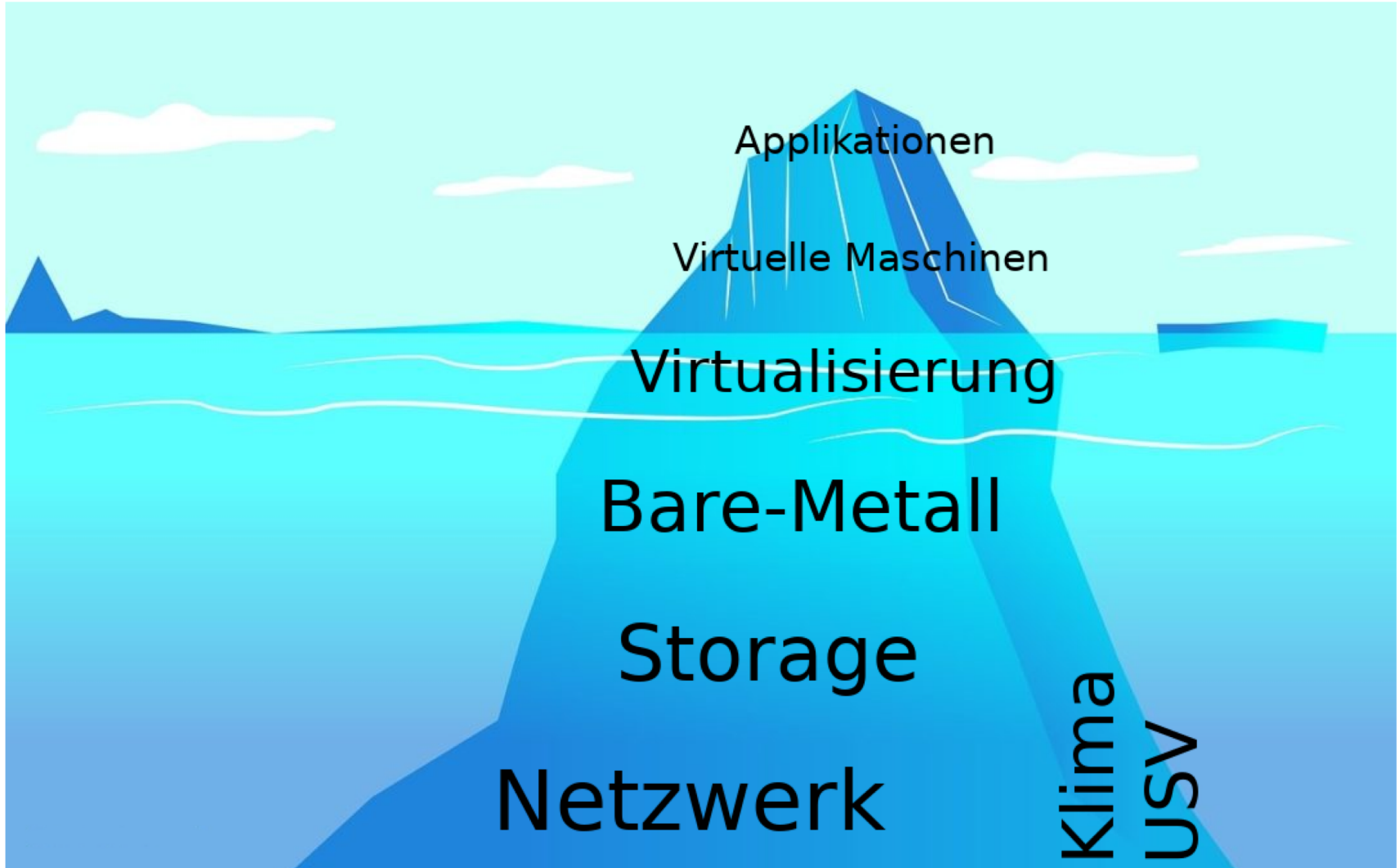
- Container vs. Para- vs. Vollvirtualisierung

- **Virtuelle Maschinen**

- Kernel namespaces, CGroups, Apparmor, SELinux ..

Full-Stack-Administration

- Applikationsdeployment
- Applikations- und Integrations-Support
- 24x7x365 Monitoring und Rufbereitschaft



Fokus

- Datacenter

Flexible Infrastruktur notwendig

- Netzwerk
- Storage
- Server
- Virtualisierung

Datacenter

Automatisierte Systemadministration

- **Reproduzierbar**

- „*Alle reden von Backup, keiner von Restore*“

- Nutzdaten auslagern

- Cattle-And-Cows-Prinzip

- **Parallelisierbar**

- Scale-Up vs. Scale-Out

- **Orchestrierbar**

- > 10.000 Systeme und mehr

Kleinsten gemeinsamen Nenner ermitteln

- Reine Container-Virtualisierung funktioniert nicht
- Unterschiedliche Workloads (Datenbanken, Java, Monitoring ...) beachten
- Provisionierung und Konfiguration ist überall notwendig

Solide und bewährte Technik auswählen

- Robuste Dateisysteme wie z.B. ext4 verwenden
- Logical-Volume-Manager funktioniert überall ähnlich
- Fibre-Channel und iSCSI als Storage-Protokolle

Wissen ist Macht

- Nichts wissen ist fatal ...
- Technik auswählen für die Wissen und Know-How zur Verfügung steht

– **„#1 If you break it, can you fix it?“**

Standards wählen

- Keine exotischen Dinge, auch wenn die Features verlockend sind
- Möglichst offene und transparente Techniken aussuchen

Vendor-Lock-In vermeiden

- Hardware
- Software

Infrastructure-As-Code Prinzip

- „Script-All-The-Things“-Strategie
- Versionierbarkeit
- Transparenz
- Baseline schaffen
- Qualität sichern und erhöhen

Einheitliche Build- und Deployment-Prozesse

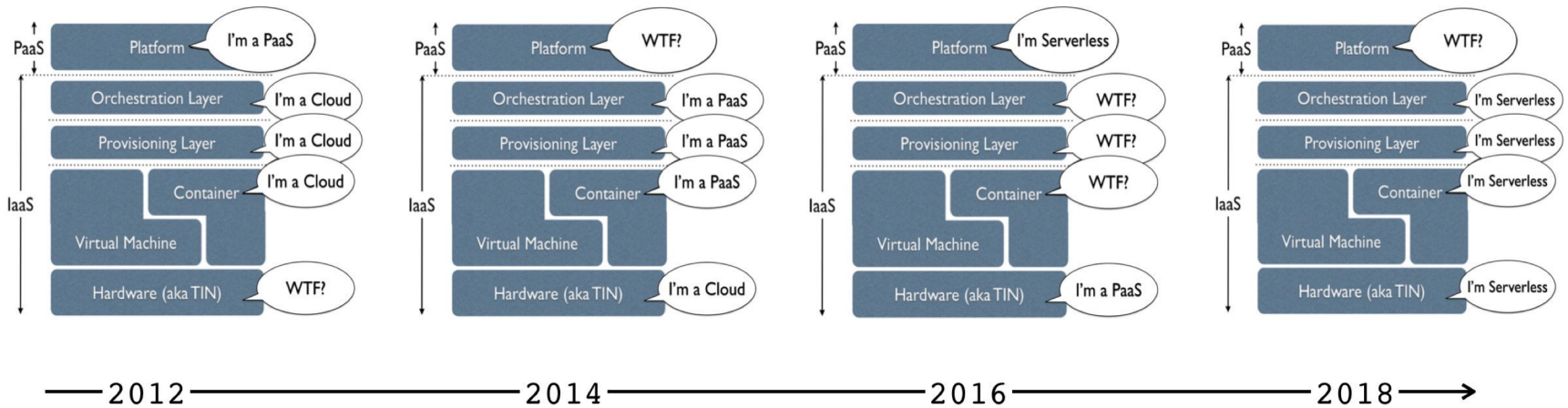
- Continuous Deployment

- „*Deploy fast, deploy often*“

- Continuous Integration

- „*Deploy everything that's stable and tested*“

The "Evolution" of Technology Thought Leadership



Virtualisierung - Proxmox VE -

Wikipedia

*“ Proxmox VE (Proxmox Virtual Environment; kurz PVE) ist eine auf Debian basierende Open-Source-Virtualisierungsplattform zum Betrieb von virtuellen Maschinen mit einer Web-Oberfläche zur Einrichtung und Steuerung von x86-Virtualisierungen. Die Umgebung basiert auf QEMU mit der Kernel-based Virtual Machine (**KVM**). PVE bietet neben dem Betrieb von klassischen virtuellen Maschinen (Gastsystemen), die auch den Einsatz von Virtual Appliances erlauben, auch LinuX Containers (**LXC**) an. “*

Proxmox

*“ Proxmox VE basiert auf Debian GNU/Linux und nutzt einen **modifizierten** Linux Kernel. Der Quellcode von Proxmox VE ist unter der Open Source-Lizenz GNU Affero General Public License, Version 3 (GNU AGPL, v3) veröffentlicht. Die AGPL, v3 erlaubt allen Nutzern den Zugriff auf den Source Code oder auch eigenen Code zum Projekt **beizutragen**. “*

Releases

•Proxmox VE 4.x

First Release: 2015-10, Debian Version 8 (Jessie), Debian EOL: 2018-06,
Proxmox VE EOL: 2018-06

•Proxmox VE 5.x

First Release: 2017-07, Debian Version 9 (Stretch), Debian 2020-07: Proxmox
VE EOL: 2020-07

•Proxmox VE 6.x

First Release: 2019-07, Debian Version 10 (Buster), Debian EOL: tba, Proxmox
VE EOL: tba

•<https://pve.proxmox.com/wiki/Roadmap>
<https://forum.proxmox.com/threads/proxmox-ve-support-lifecycle.35755/>

Community-Edition

```
deb http://download.proxmox.com/debian/pve stretch pve-no-subscription
```

Enterprise-Edition

```
deb https://enterprise.proxmox.com/debian/pve stretch pve-enterprise
```

Eigenbau

https://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Buster

Virtualisierung-Technologien

- Kernel-based Virtual Machine (KVM)
 - Linux
 - Windows
- Linux Containers (LXC)
 - Linux
 - Linux-Applikationen (Sandboxing)

Live-Migration

- KVM → Shared-Storage (SAN, NFS) notwendig
- LXC → Local Storage notwendig

Zentrales Management

- Mult-Master-Funktionalität
- Selbsterklärende Web-Oberfläche
- CLI für Proxmox VE, KVM und LCX
- RESTful API für Anbindung externer Tools

Proxmox Cluster File System (pmxcfs)

- Synchronisierung Konfiguration über alle Nodes

Umfangreiches Berechtigungssystem

- Gruppen
- Pools

Integriertes Backup und Restore

- Scheduled Backups über CRON
- „Live Backups“ über LVM-Snapshot möglich
- „Cold-Backups“ über Shutdown/Start möglich
- Kompression (lzo / gzip) möglich
- Umfangreiche Konfiguration möglich
 - Snapshot / Suspend
 - Limitierung Disk-I/O, Netzwerk-I/O

HA Cluster

- Automatischer Fail-Over / Fail-Back von VMs
- Watchdog-basiertes Fencing
- Prioritäten-Steuerung
- Proxmox Cluster Filesystem (pmxcfs)

Firewall

- Integrierte Firewall-Funktionalität zur Isolierung von VMs und Containern
- Verteilte (distributed) Firewall „pmxcfs“
- Unterstützung IPv4 und IPv6

Software-Defined-Netzwerk

- Bridged-Netzwerk
- VLANs (IEEE 802.1Q)
 - Tagged und Untagged VLANs
- Netzwerk-Bonding/Link-Aggregation

„Alle VMs können eine Bridge teilen, so als ob virtuelle Netzwerk-Kabel von jedem Gast in den gleichen Switch gingen. Damit die VMs nach außen kommunizieren können, werden Bridges zu physischen Netzwerkkarten mit einer TCP / IP-Konfiguration angehängt.“

Flexible Speichersubsysteme

•Lokal

- LVM / LVM-thin
- Lokales Verzeichnis
- ZFS

•Netzwerkspeicher

- LVM / LVM-thin über Fibre Channel oder iSCSI
- NFS
- CIFS
- Ceph RBD / CephFS
- GlusterFS

PROXMOX Virtual Environment 6.0-4 [Documentation](#) [Create VM](#) [Create CT](#) [root@pam](#)





Server View **Datacenter (democluster)**

- pve6-demo1
 - ceph (pve6-demo1)
 - cephfs (pve6-demo1)
 - local (pve6-demo1)
 - local-lvm (pve6-demo1)
- pve6-demo2
 - 100 (Buster)
 - ceph (pve6-demo2)
 - cephfs (pve6-demo2)
 - local (pve6-demo2)
 - local-lvm (pve6-demo2)
- pve6-demo3
 - 101 (debianct)
 - ceph (pve6-demo3)
 - cephfs (pve6-demo3)
 - local (pve6-demo3)
 - local-lvm (pve6-demo3)
- production
- testlab

Datacenter

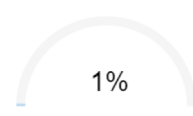
- Search
- Summary
- Cluster
- Ceph
- Options
- Storage
- Backup
- Replication
- Permissions
- Users
- Groups
- Pools
- Roles
- Authentication
- HA
- Firewall
- Support

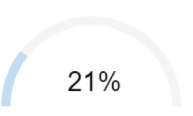
Health

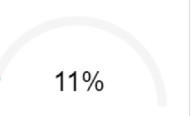
Status  **Nodes**  Online 3  Offline 0 **Ceph**  **HEALTH_OK**

Cluster: democluster, Quorate: Yes


Resources

CPU  1% of 18 CPU(s)



Memory  21% 4.92 GiB of 23.36 GiB



Storage  11% 17.66 GiB of 153.91 GiB

Subscriptions







Community  Your subscription status is valid.

Guests

Virtual Machines  Running 1  Stopped 0

LXC Container  Running 1  Stopped 0

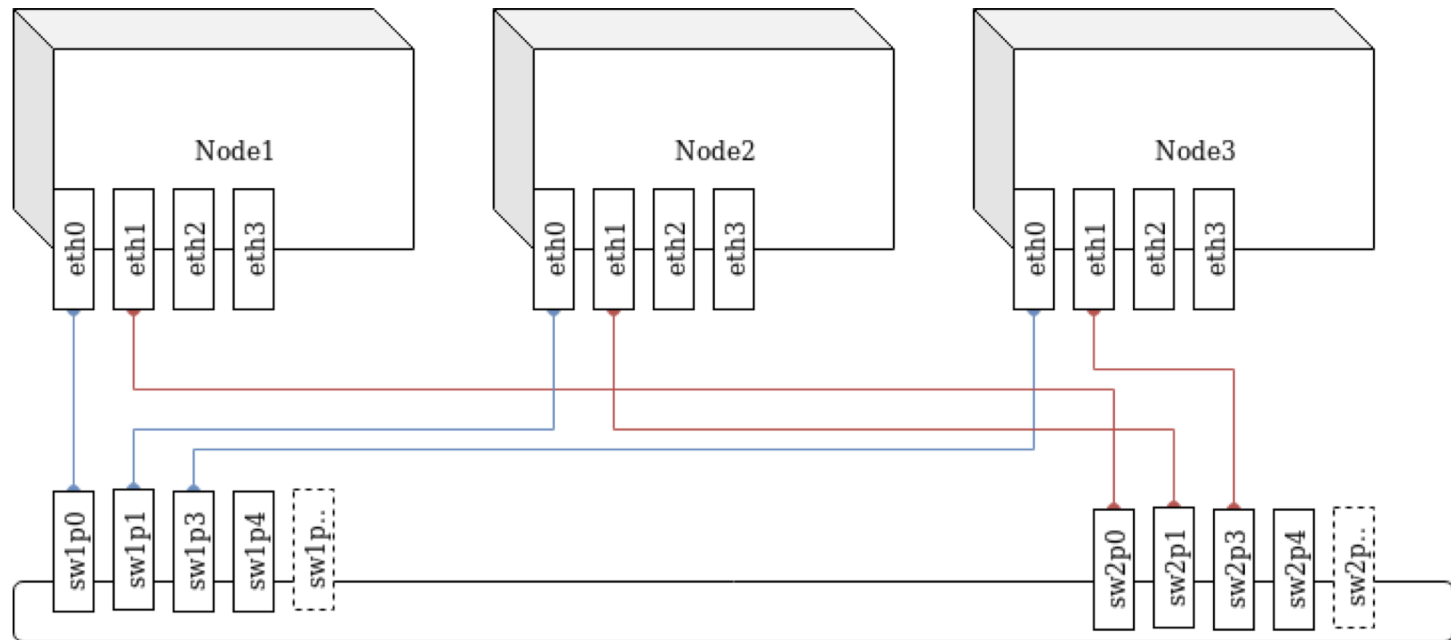
Nodes

Name	ID	Online	Support	Server Address	CPU usage	Memory usage	Uptime
pve6-demo1	1		Community	192.168.6.80	1%	 18%	00:06:36
pve6-demo2	2		Community	192.168.6.81	1%	 26%	00:06:35
pve6-demo3	3		Community	192.168.6.82	1%	 19%	00:06:30

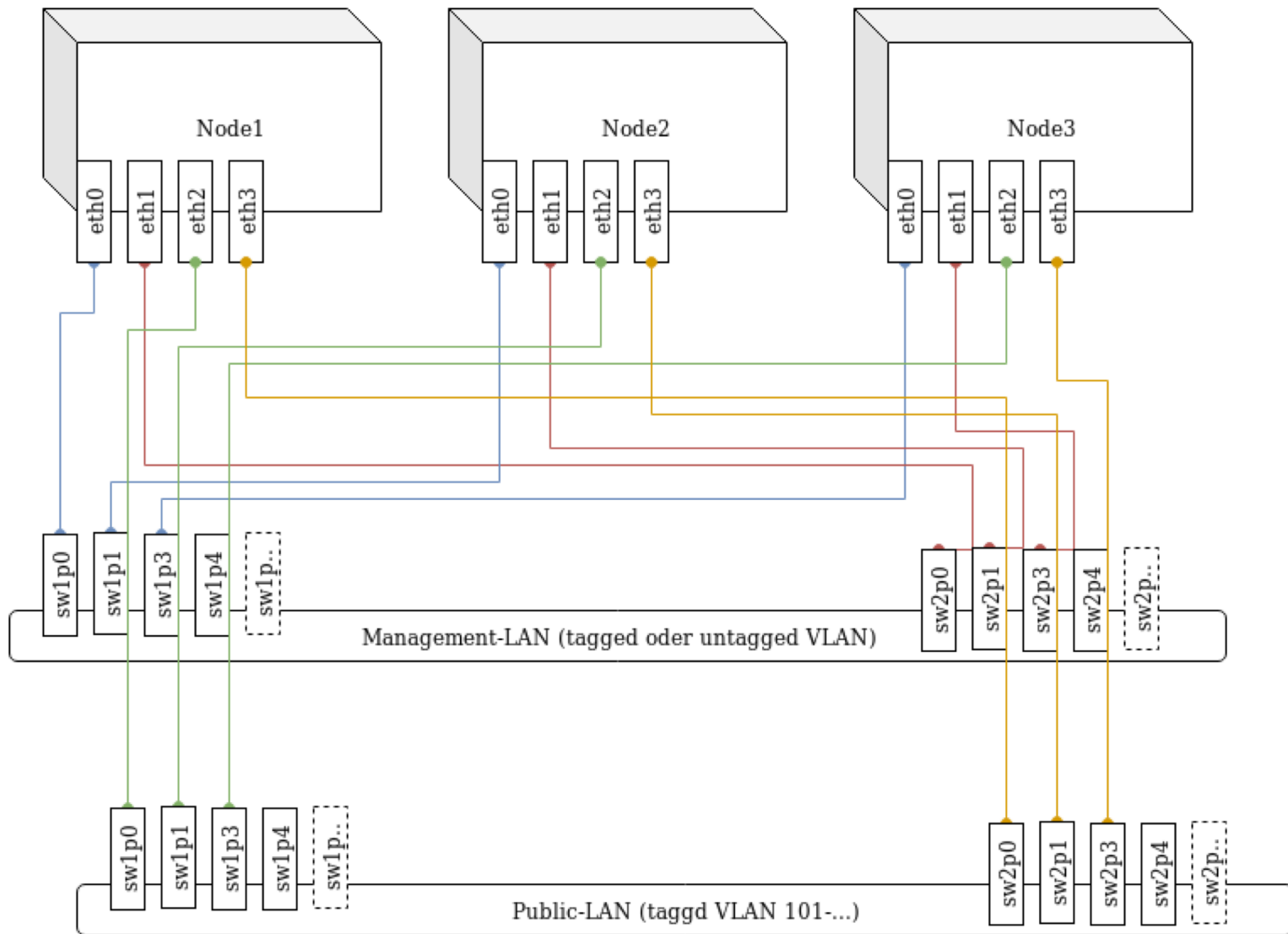
Tasks **Cluster log**

Start Time ↓	End Time	Node	User name	Description	Status
Jul 16 11:31:59	Jul 16 11:32:02	pve6-demo1	root@pam	Shell	OK
Jul 16 11:31:18	Jul 16 11:31:19	pve6-demo2	root@pam	VM 100 - Start	OK
Jul 16 11:29:15	Jul 16 11:30:00	pve6-demo3	root@pam	CT 101 - Start	OK
Jul 16 11:27:21	Jul 16 11:27:23	pve6-demo3	root@pam	Start all VMs and Containers	OK
Jul 16 11:27:15	Jul 16 11:27:17	pve6-demo2	root@pam	Start all VMs and Containers	OK
Jul 16 11:27:13	Jul 16 11:27:15	pve6-demo1	root@pam	Start all VMs and Containers	OK

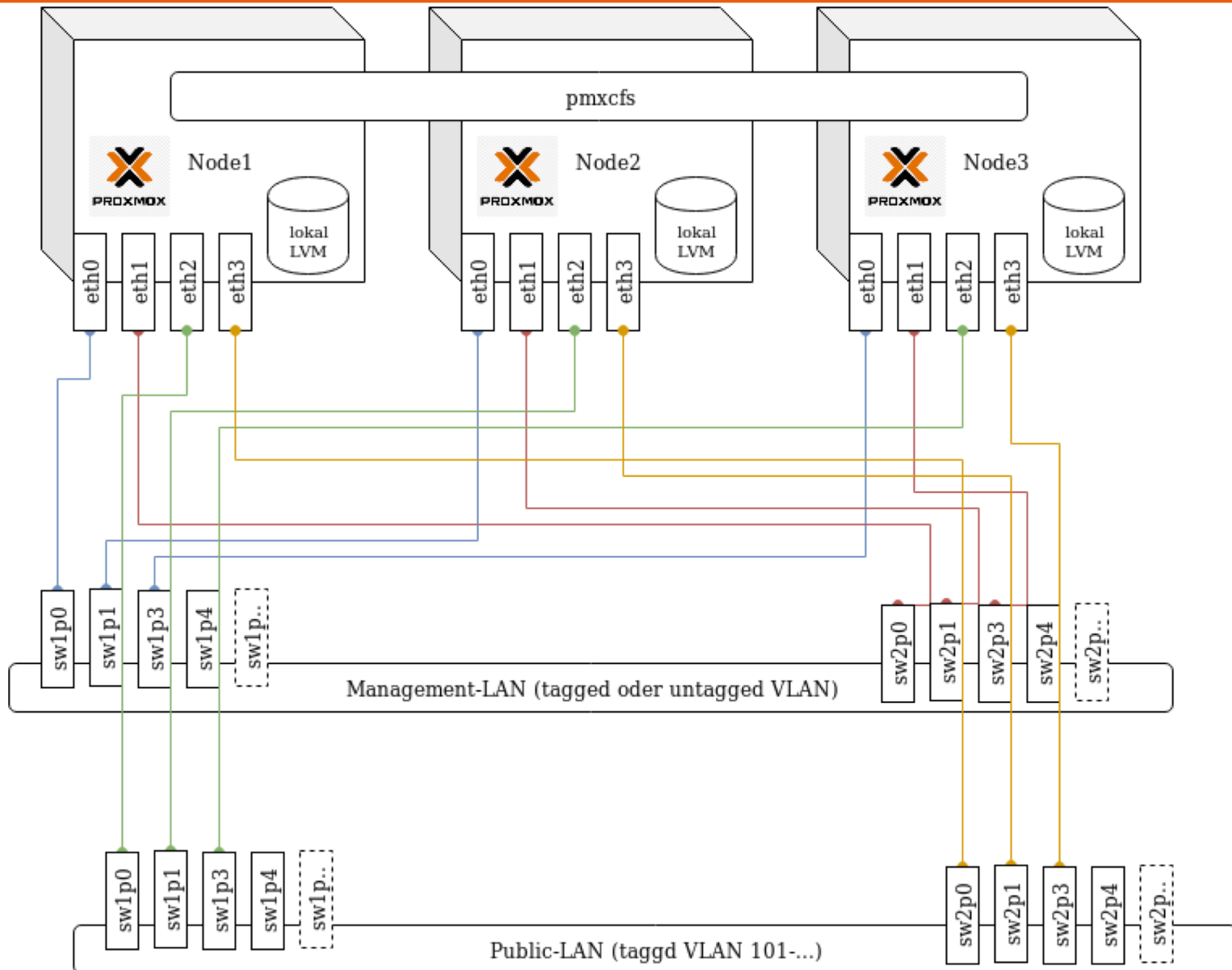
Aufbau



Datacenter (3-Node-Setup) - Vorbereitung

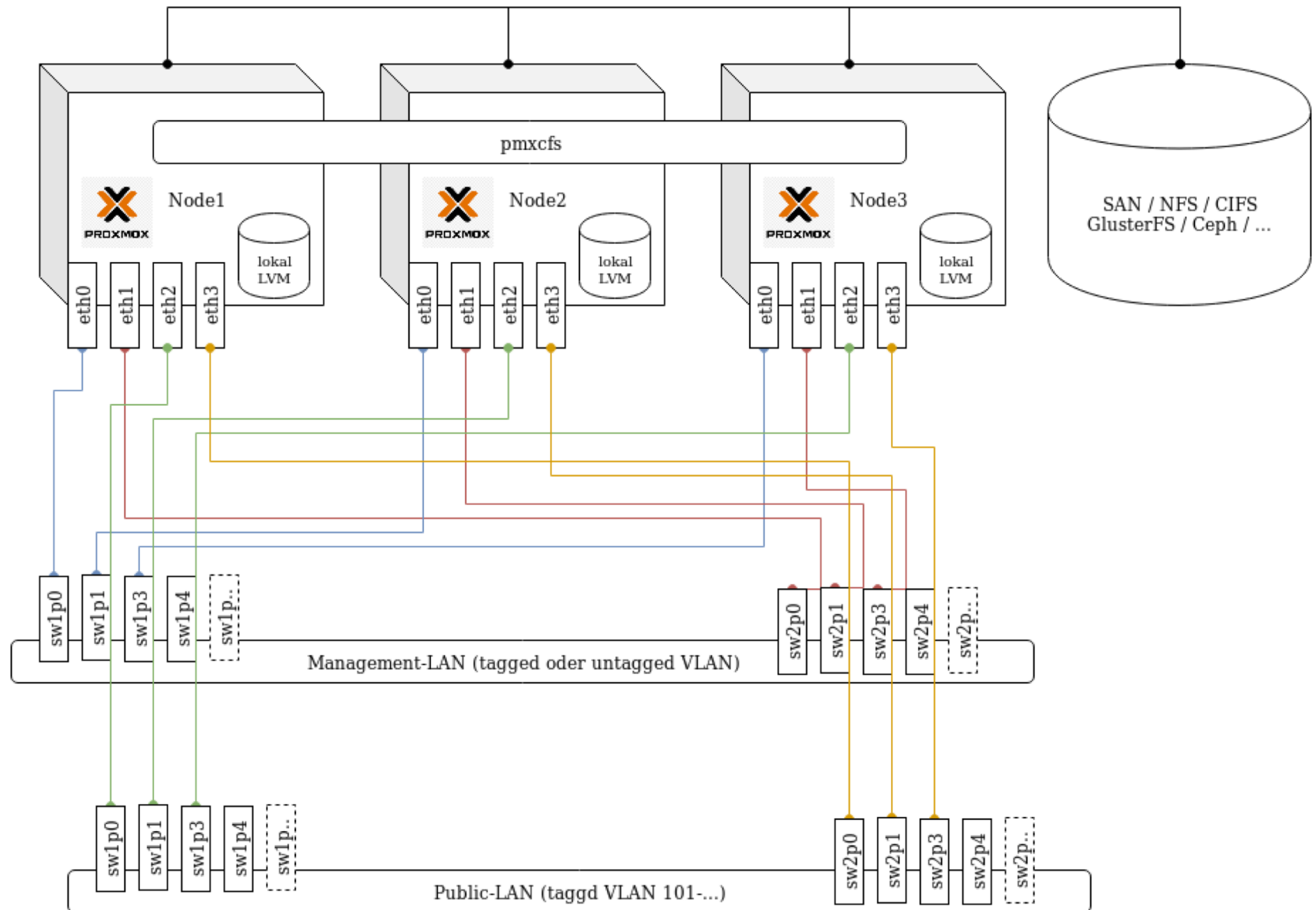


Datacenter (3-Node-Setup) - Proxmox-Cluster



- Betrieb von KVM und LXC-Containern
- Backup und Restore
- Firewall (kann aktiviert werden)
- Live-Migration für LXC-Container (sofern kein LVM)

Datacenter (3-Node-Setup) - Proxmox-HA-Cluster



- Betrieb von KVM und LXC-Containern
- Backup und Restore
- Firewall (kann aktiviert werden)
- Live-Migration für LXC-Container
- HA mit Fail-Over, Fail-Back und Fencing
- Live-Migration von KVM-VMs

Provisionierung Management

Templates

Templates sind fertig installierte KVMs oder LXC-Container, welche anschließend noch individualisiert werden müssen.

Aus Templates können Vms folgender Art erstellt werden:

- Clone**

1:1 Kopie

- Linked Clone**

Aufs Ursprungs-Image verlinkte Kopie

Automatische Installation

- Debian Preseed
- Autoyast
- FAI
- ...

über PXE-TFTP-Boot möglich

Tool-Unterstützung nutzen

- Ansible

https://docs.ansible.com/ansible/latest/modules/proxmox_module.html

- Foreman

https://github.com/theforeman/foreman_fog_proxmox

- Terraform

<https://github.com/Telmate/terraform-provider-proxmox>

- Orcharhino

<https://orcharhino.com/orcharhino-4-1-0/>

Deployment

VM / Container erstellen / klonen

- Web-GUI → manuell
- Rest-API → Ansible
- CLI → Ansible

Template(-Engine) aus CMDB

```
TEMPLATE_VERSION: 2
ROOT_SERVER: proxmox-101.domain.tld
INFRASTRUCTURE: production
INFRASTRUCTURE_COUNTRY: de
SERVER_NAME: sto-nfs-01.domain.tld
SERVER_NAME_SHORT: "{{ SERVER_NAME | regex_replace('^(?P<short>[^\.\.]*')\\\.\\\.\\.',
'\g<short>') }}"
NEWID: "{{ IPADDR | regex_replace('\\.\\.(?P<last>[0-9]{2})$', '.0\\g<last>') |
regex_replace('\\.\\.(?P<third>[0-9]{1,3})\\.\\.(?P<fourth>[0-9]{3})$',
'\g<third>\\g<fourth>') }}"
IPADDR: 192.168.2.102
GATEWAY_IP: "{{ IPADDR | regex_replace('[0-9]*$', '1') }}"
TAG: "{{ IPADDR | regex_replace('\\.\\.(?P<third>[0-9]{2,3})\\.\\.(?P<fourth>[0-9]{2,3})$',
'\g<third>') | regex_replace('^(?P<two>[0-9]{2})$', '8\\g<two>') }}"
REBOOT_VAR: True
CLONE_SOURCE: Debian9
PUPPET_HOSTGROUP: production
SOFTWARE_INSTALL: True
CORES: 4
MEMORY: 4096
DISKSIZE: 20
STORAGE: local-lvm
DESCRIPTION: "My New Server"
```

VM-Deployment



- .VM klonen / erstellen
- .VM booten
- .VM auf Zielsystem verschieben
- .VM konfigurieren
- .Software installieren
- .Reboot
- .Ready → Produktion

CD-Service → Continuous Delivery Service

- Paketverwaltung
 - RPM, DEB
 - Eigenes Paket-Format (tar.gz, gzip)
- Installiert nach dem booten automatisch Software anhand einer CMDB
- Triggert Puppet um die Konfiguration zu vervollständigen

CD-Service → Continuous Delivery Service

```
> root@cfg-jenkins-02:/opt/cdservice# bash install.sh -d

using prod.conf
SHOW_ONLY is set, only listing software

software: jdk, version: 1.8.0-161.6, repoType: tar.gz, repoUrl:
http://repo.domain.tld/debian9/

software: apache2, version: current, repoType: system, repoUrl:

software: jenkins, version: current, repoType: system, repoUrl:

software: git, version: current, repoType: system, repoUrl:

> root@cfg-jenkins-02:/opt/cdservice#
```

CD-Service → Continuous Delivery Service

```
> root@app-web-01:/opt/cdservice# bash install.sh -d
```

```
software: jdk, version: 1.5.0-22.12, repoType: tar.gz, repoUrl:  
http://repo.domain.tld/debian9/
```

```
software: apache2, version: current, repoType: system, repoUrl:
```

```
software: apache_tomcat, version: 5.5.26-35, repoType: tar.gz, repoUrl:  
http://repo.domain.tld/debian9/
```

parameter:

```
JAVA_ENV=-DgraphiteHost=db-influxdb-01.domain.tld -DgraphitePort=2003 -  
DgraphiteFlushSize=15
```

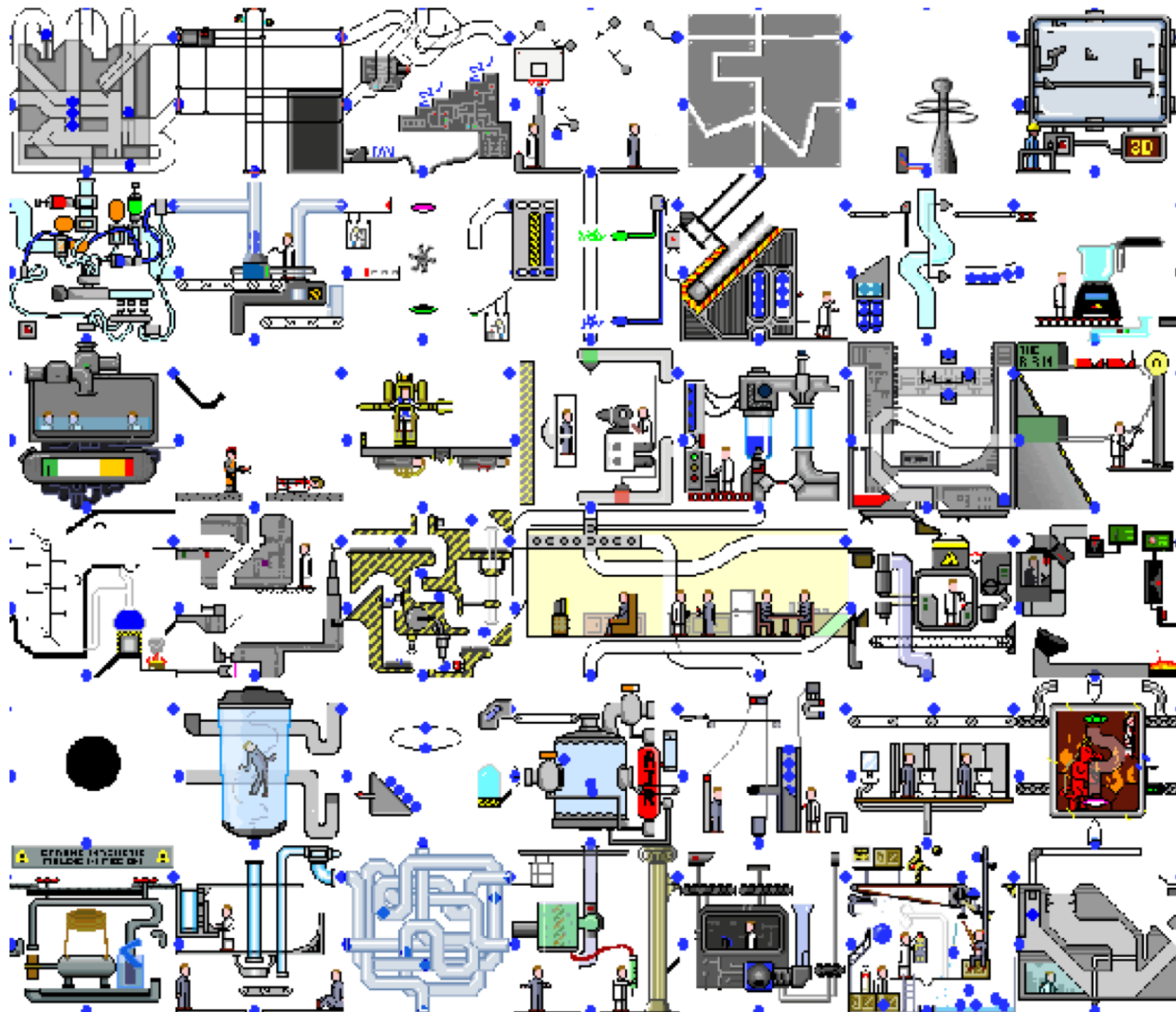
```
JAVA_MEMORY=-Xms512m -Xmx2048m -XX:PermSize=128m -XX:MaxPermSize=512m
```

```
JAVA_OPTS=
```

```
JAVA_SOFTWARE=
```

```
RESTART=true
```

```
> root@cfg-jenkins-02:/opt/cdservice#
```



- Danke für die Aufmerksamkeit -

... Fragen?