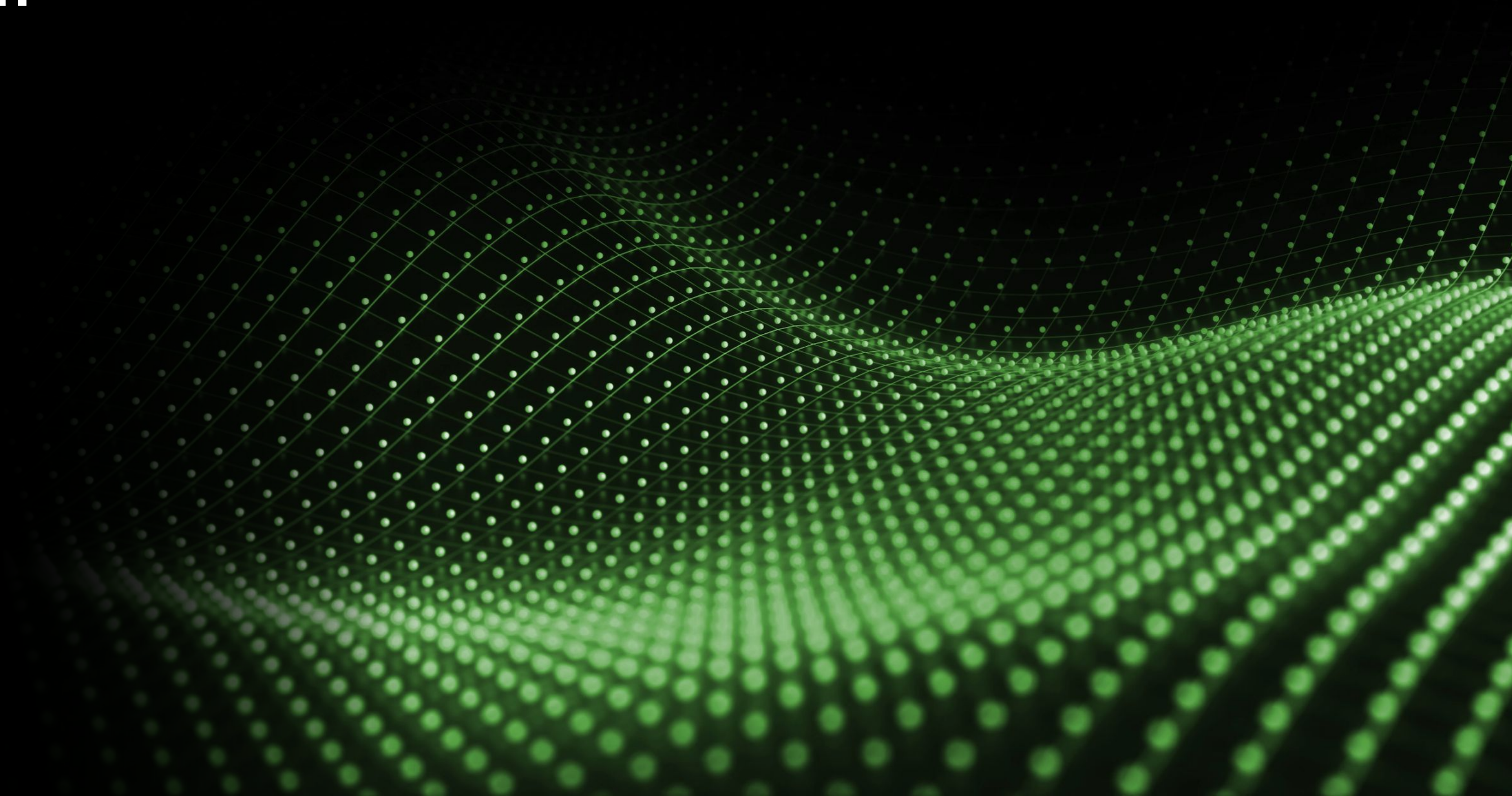
15th of January 2025

REiST
IT SOLUTIONS FOR TODAY & TOMORROW

# Unlocking OpenShift Virtualization Potential on Bare Metal: A Deep Dive with Reist Telecom AG

## Red Hat Summit:  Connect Zurich

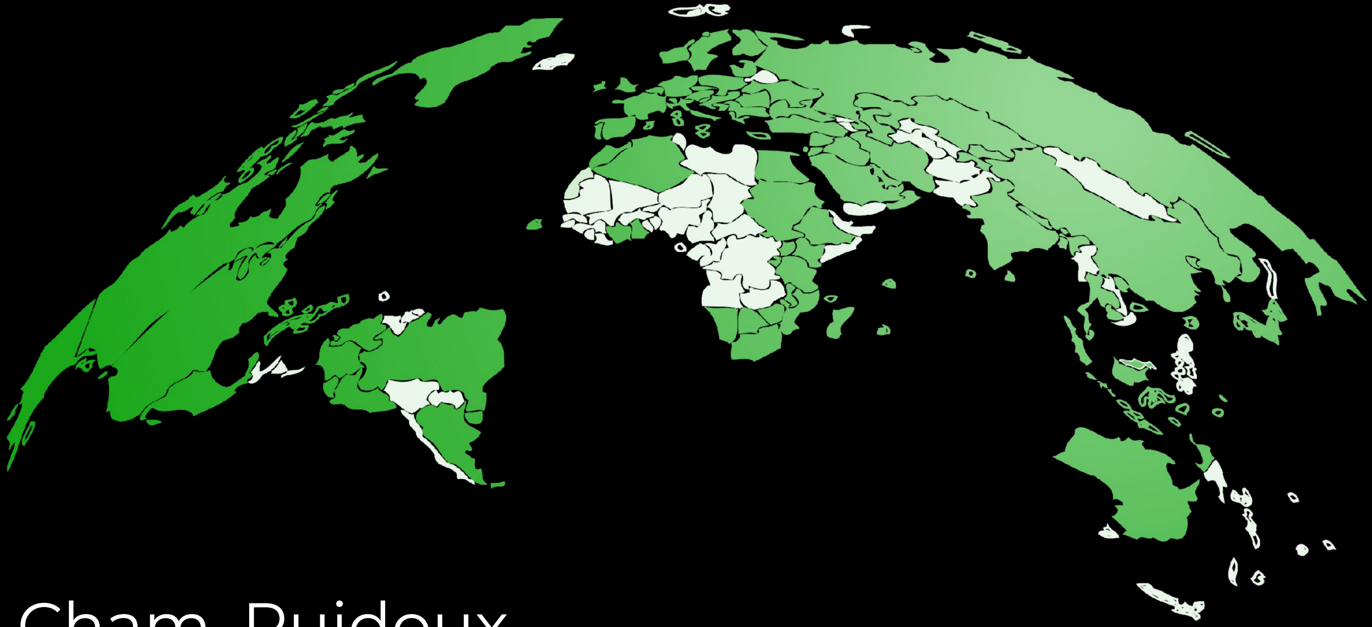


Patric Siegrist
Chief Architect

# About Reist Telecom AG

Company facts

REIST
IT SOLUTIONS FOR TODAY & TOMORROW



- Founded 2001
- 100% private owned
- 75 employees
- Located in Zurich, Basel, Cham, Puidoux
- Customers in more than 50 countries in Aviation, Manufacturing, IT, Insurance, Private Banking and other industries

Public

# About Reist Telecom AG

## Our Solution Portfolio

### Network and Security

- LAN, WAN, SD*
- ZTNA, NAC
- VPN, Remote access
- CASB, NDR

### Cloud solutions (Private, Public and Multi-cloud)

- VMs
- Hosting/housing
- K8S
- APO

### Identity and Access Management (MAYI ID)

- IAM
- PAM
- CLM
- Vaulting

Cybersecurity
Monitoring
Reporting
Service Management
Operations and Support (7x24x365)

# Why OpenShift Virtualization?

REiST
IT SOLUTIONS FOR TODAY & TOMORROW

- Uncertainty of pricing model impacts with existing platform with due to Broadcom VMWare acquisition
- VM workloads with no planned containerization in near future
- Customers with the need of Swiss private cloud hosting
- One platform for Kubernetes and VM workloads

- … and why on Bare Metal?
- prerequisite to run OpenShift Virtualization
- proven and existing hardware processes in our datacenters
- Shift existing hosts and reuse on new platform

# Transformative impact

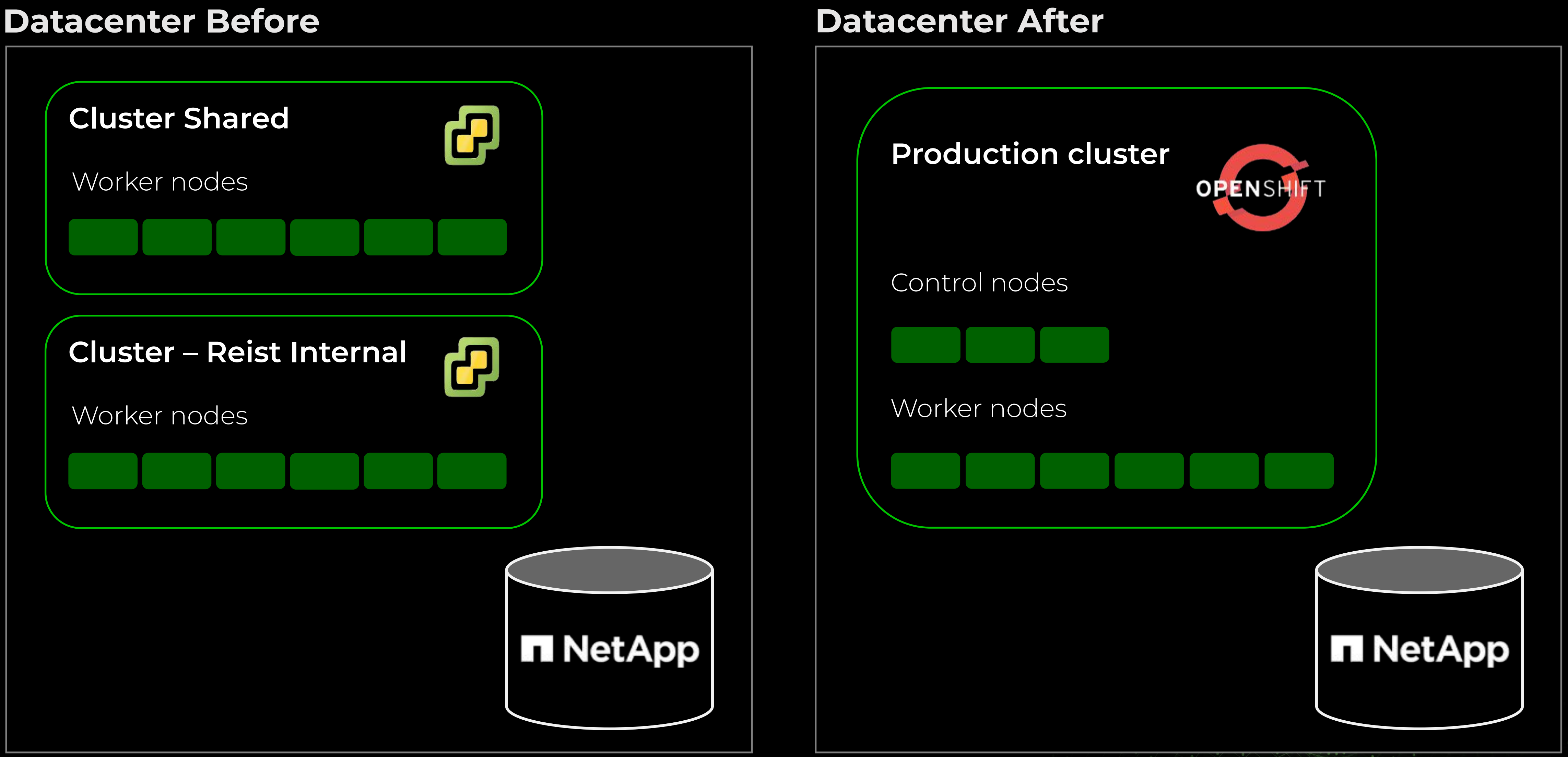
## “Legacy” VM platform

- Service provider within shared environment
- 70 bare metal hosts
- New license not flexible with compulsory minimum purchase
- Cost increase by factor 2 – 3.3 with 3y / 1y commitment

## OpenShift Virtualization platform

- Transparent monthly usage pricing per worker node socket-pair
- Less workers needed as no separation between shared / non-shared and continuous modernization to containers / microservices
- Roughly 30% cheaper than 3y commitment with same node count

# Transformative Impact



## Datacenter Before

### Cluster Shared

Worker nodes

🟩🟩🟩🟩🟩🟩

### Cluster – Reist Internal

Worker nodes

🟩🟩🟩🟩🟩🟩

NetApp

## Datacenter After

### Production cluster

OPENSHIFT

Control nodes

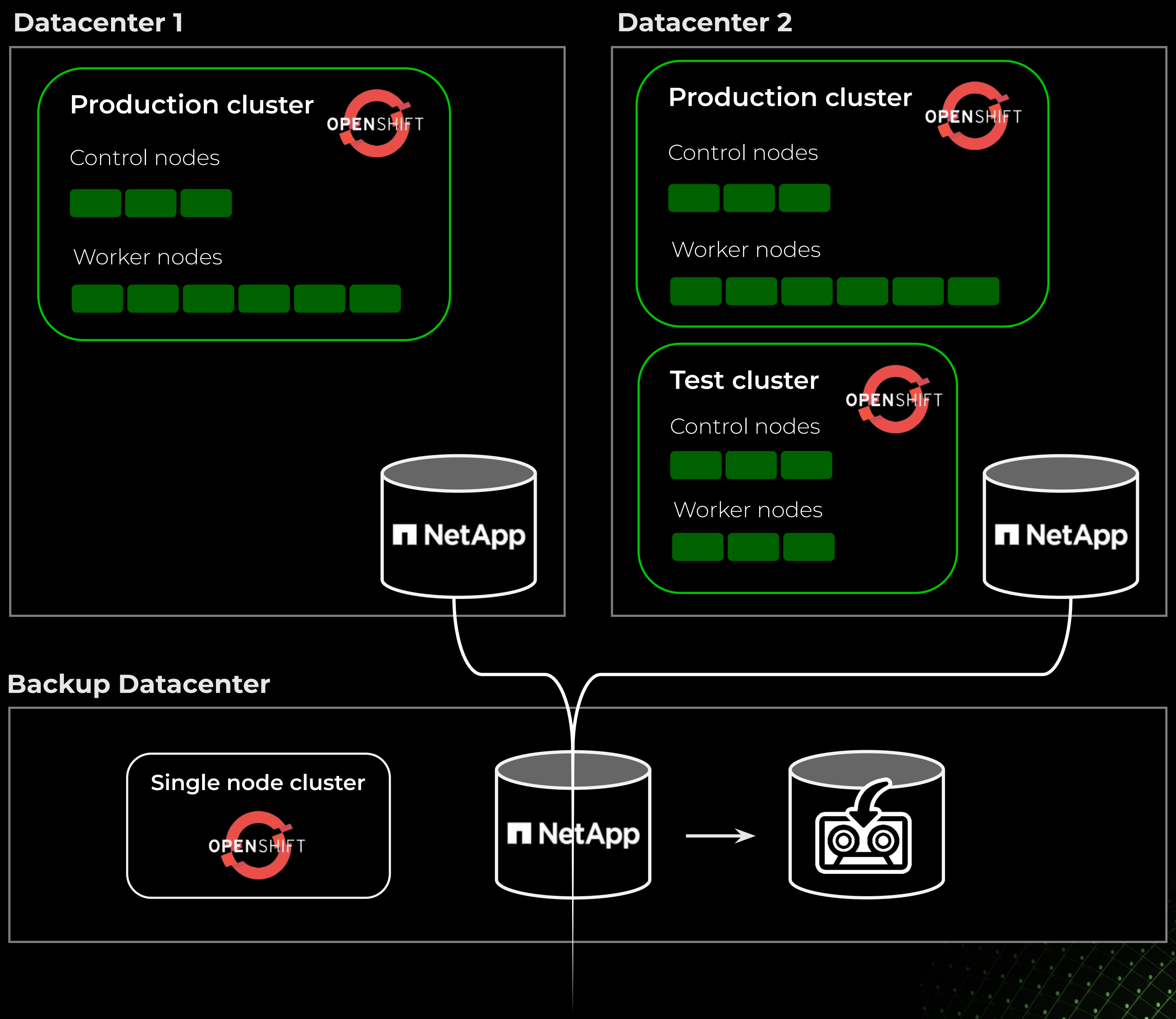
🟩🟩🟩

Worker nodes

🟩🟩🟩🟩🟩🟩

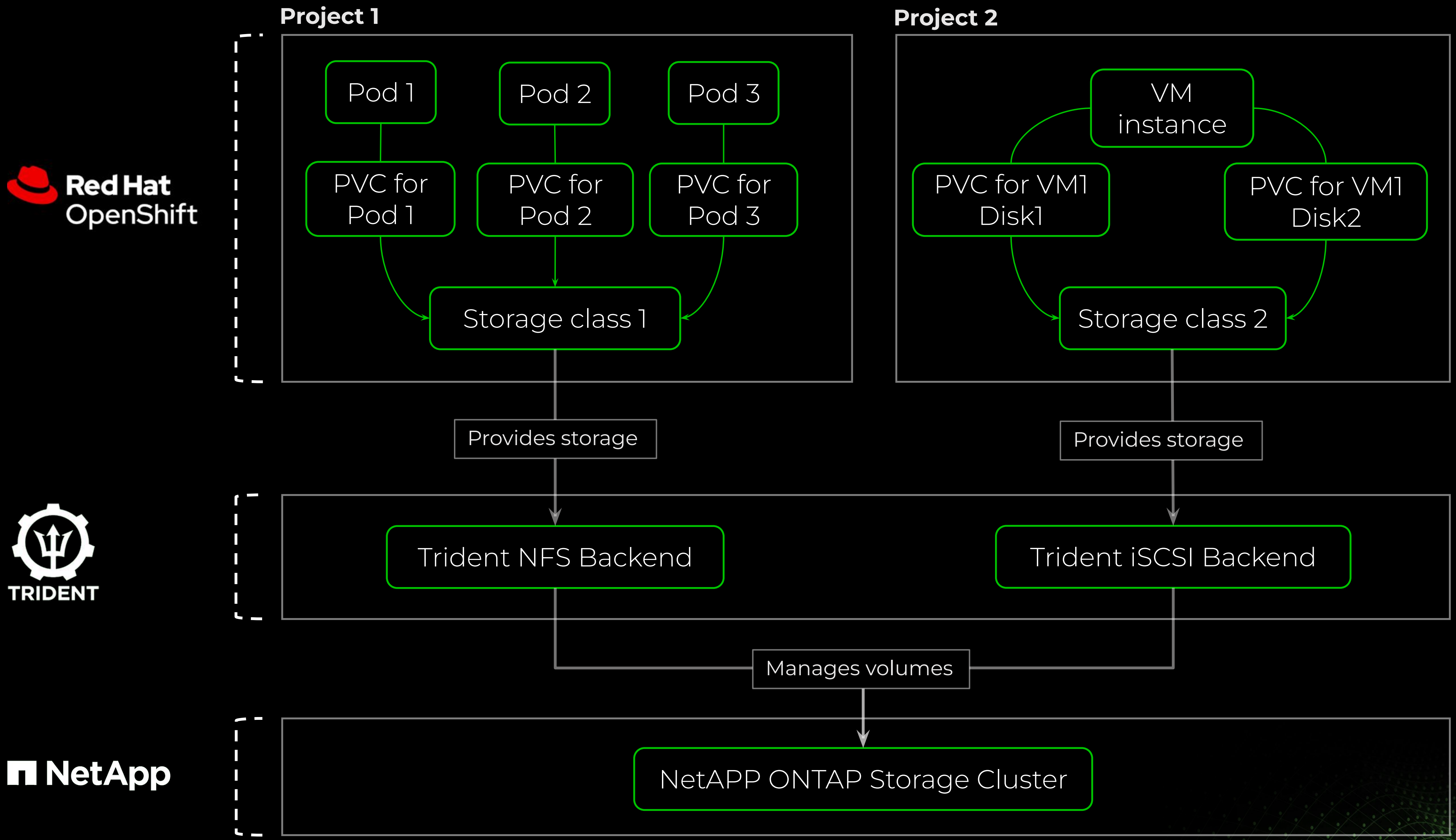
NetApp

# Transformative Impact



# Storage integration

REIST
IT SOLUTIONS FOR TODAY & TOMORROW



Public

# Networking

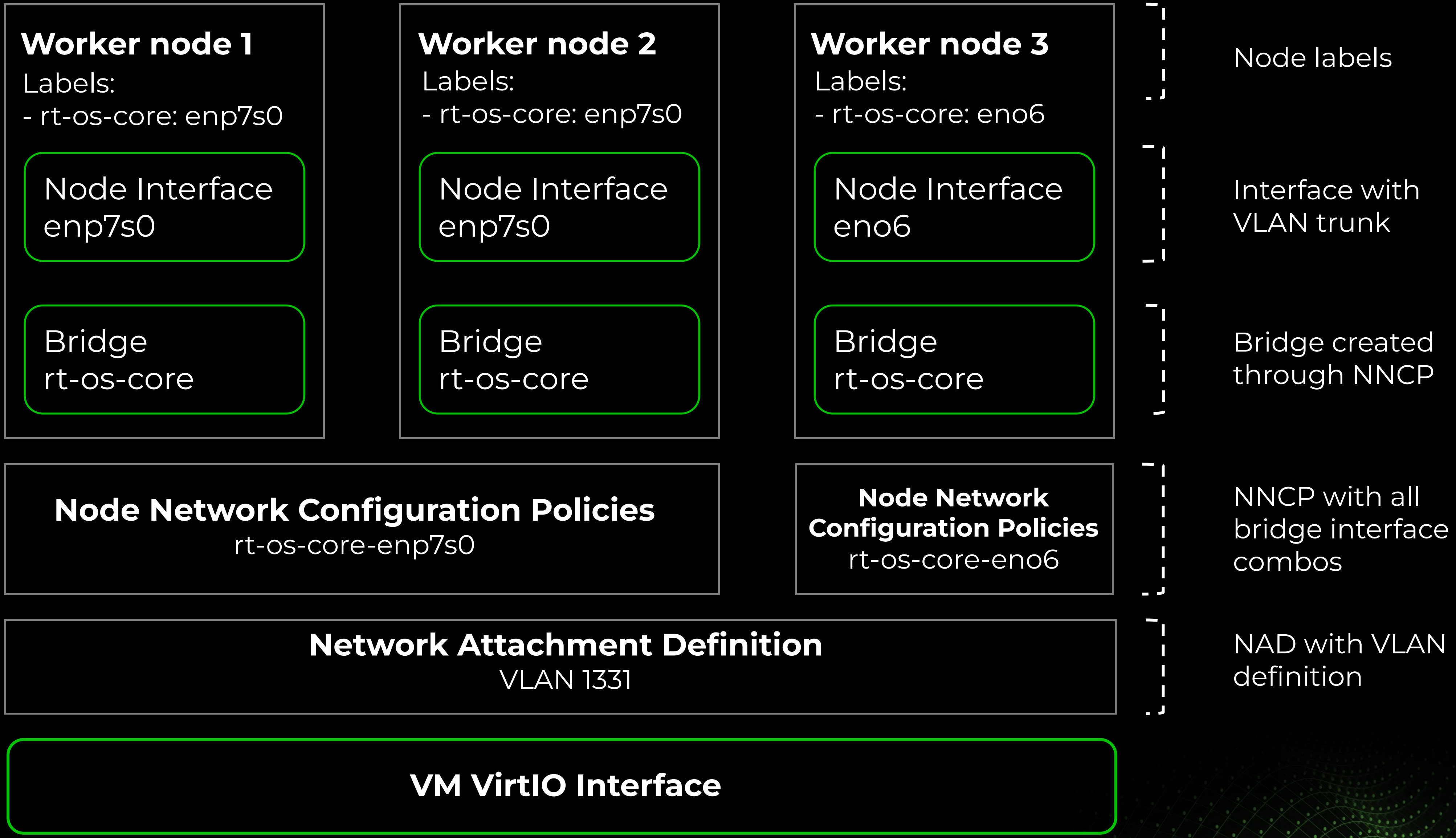
## Challenge with VMs

- Use pod default network not sufficient for legacy use-cases
- Over 100 VLANs in use for customer network segregation

## Solution for transition to OpenShift Virtualization

- Additional host interfaces for VLAN trunks
  - □ Different host hardware = different interface names
- Dynamic bridges through node labels with Node Network Configuration Policies
- Consume VLANs with Network Attachment Definitions

# Networking



**Worker node 1**
Labels:
- rt-os-core: enp7s0

Node Interface enp7s0

Bridge rt-os-core

**Worker node 2**
Labels:
- rt-os-core: enp7s0

Node Interface enp7s0

Bridge rt-os-core

**Worker node 3**
Labels:
- rt-os-core: eno6

Node Interface eno6

Bridge rt-os-core

**Node Network Configuration Policies**
rt-os-core-enp7s0

**Node Network Configuration Policies**
rt-os-core-eno6

**Network Attachment Definition**
VLAN 1331

**VM VirtIO Interface**

Node labels

Interface with VLAN trunk

Bridge created through NNCP

NNCP with all bridge interface combos

NAD with VLAN definition

# VLANs with NADs

- We are now ready to consume a VLAN through Network Attachment Definitions ( NAD )

```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  annotations:
    description: RT_VLAN_1331
  name: vlan-1331                □ NAD name
  namespace: my-vm-namespaces    □ namespace
spec:
…
```

# VLANs with NADs

- We are now ready to consume a VLAN through Network Attachment Definitions ( NAD )

```
...
spec:
  config: '{
    "name":"vlan-1331",      □ NAD name
    "type":"cnv-bridge",
    "cniVersion":"0.3.1",
    "bridge":"rt-os-core",   □ Bridge name
    "vlan":1331,             □ VLAN ID
    "macspoofchk":true,
    "ipam":{},
    "preserveDefaultVlan":false
  }'
```

# Considerations and best practices

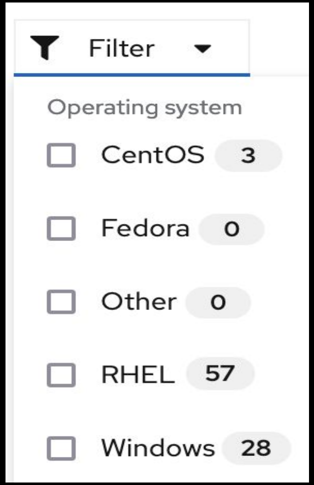
- Licensing considerations
  - □ work with node labels for correct scheduling
  - □ don’t forget about capacity planning and review

- Ensure Live Migration compatibility
  - □ Read-Write-Many ( RWX ) PVCs and Eviction Strategy
  - □ **Default CPU: smallest available within the cluster**

- Create your golden images and VM templates with cloud-init and and Ansible customization

- Think about useful annotation

```
annotations:
vm.kubevirt.io/os: rhel9 / windows2022 / ...
vm.kubevirt.io/validations: ops constraints, eg. req. memory
```



# Migration from VMWare ESXi

## Cluster preparation

- OpenShift Migration Toolkit for Virtualization ( MTV Operator )
  - Supports VMware vSphere, OpenStack, OVA, RHV and OpenShift Virtualization
- Storage Classes for **cold** / **warm** migration
  - Block storage for **warm** migration from VMware
- Allow ingress traffic from openshift-mtv namespace
- Create a VDDK init image and configure provider integration for your vCenter

## Namespace preparation

- Create target namespace
- Remove ResourceQuota and LimitRanges ( for migration )
- add required Network Attachment Definitions ( NADs )

# Migration from VMWare ESXi

## Basic VM preparation

- Decide if you go with cold or warm migration
- Rename VM to lowercase

## Additional preparation for warm migration

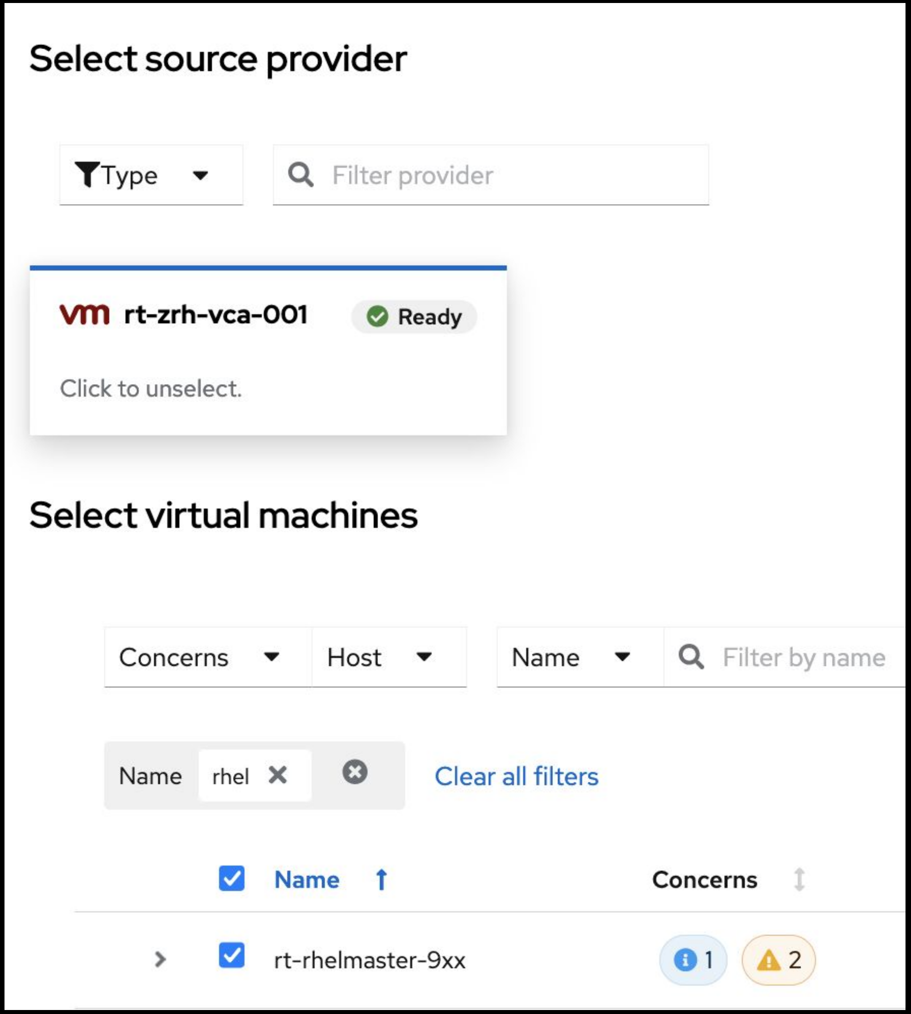
- Install VMWare Tools / open-vm-tools on Linux
- No existing or new snapshots during migration
- Set all disks to dependent - eg. swap space disks
- Enable change block tracking ( CBT )

  ctkEnabled TRUE
  scsi0:0.ctkEnabled TRUE □ for every disk

# Migration from VMWare ESXi

R E i S T
IT SOLUTIONS FOR TODAY & TOMORROW

## Migration plan ( 1 / 3 )



Select source provider

rt-zrh-vca-001 Ready

Click to unselect.

Select virtual machines

Name rhel

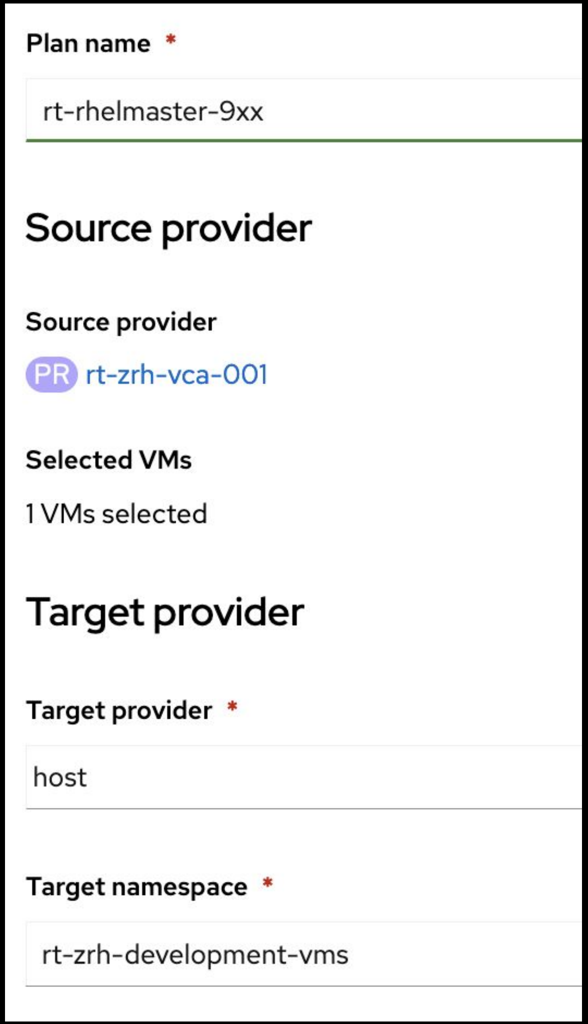
| | Name | Concerns |
|---|---|---|
| ☑ | rt-rhelmaster-9xx | 1 2 |

- □ Source Provider
- □ Browse and select VM

# Migration from VMWare ESXi

REiST
IT SOLUTIONS FOR TODAY & TOMORROW

## Migration plan ( 2 / 3 )



Plan name *

rt-rhelmaster-9xx

### Source provider

Source provider

PR rt-zrh-vca-001

Selected VMs

1 VMs selected

### Target provider

Target provider *

host

Target namespace *

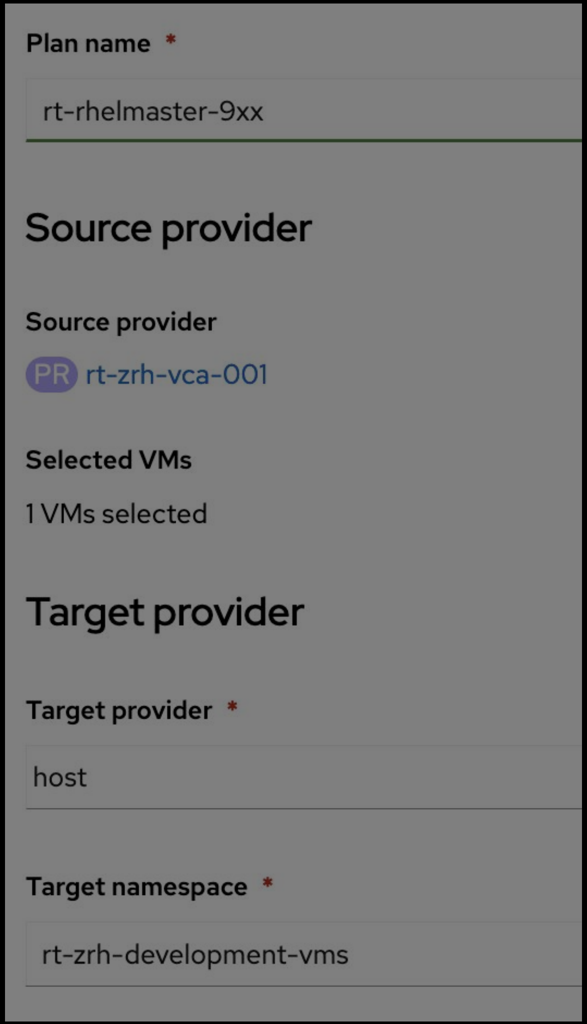
rt-zrh-development-vms

- Plan Name ( will reflect in PVC name )

- Target Namespace

# Migration from VMWare ESXi

## Migration plan ( 3 / 3 )



Plan name *

rt-rhelmaster-9xx

### Source provider

Source provider

PR rt-zrh-vca-001

Selected VMs

1 VMs selected

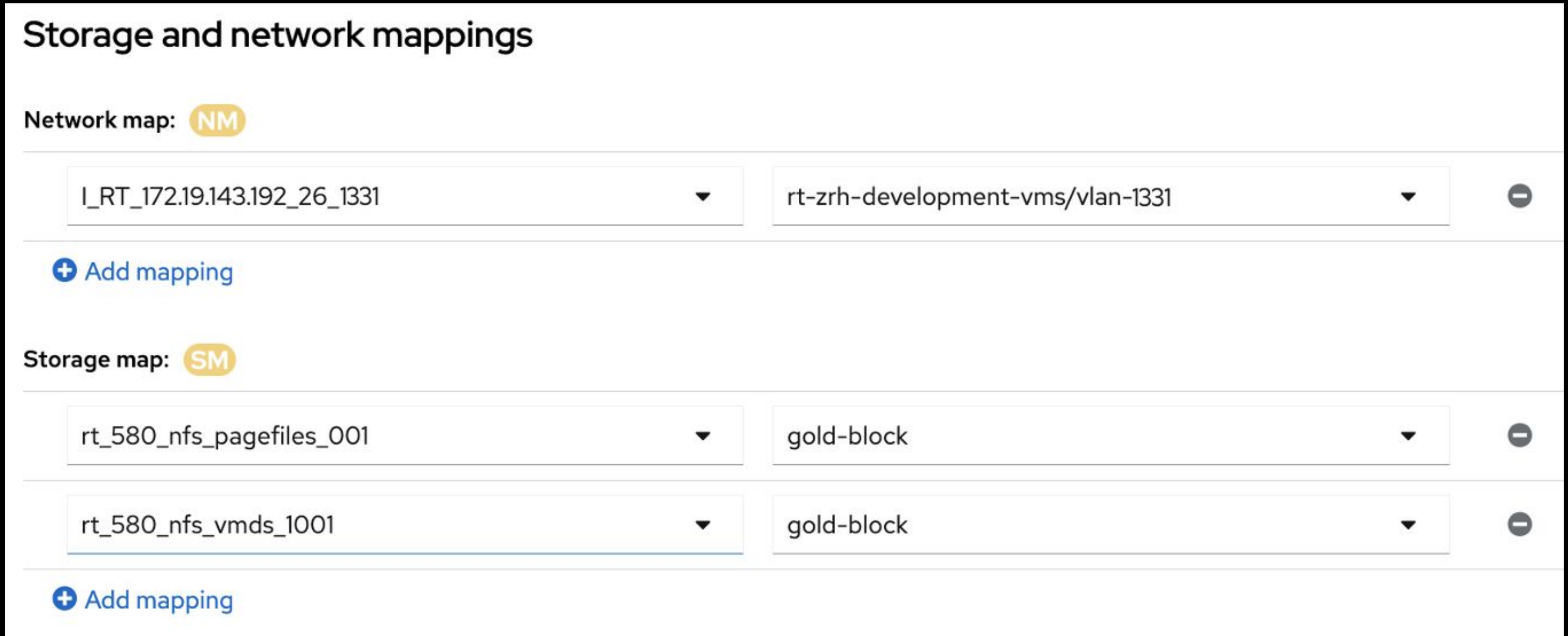
### Target provider

Target provider *

host

Target namespace *

rt-zrh-development-vms



### Storage and network mappings

Network map: NM

| | |
|---|---|
| I_RT_172.19.143.192_26_1331 | rt-zrh-development-vms/vlan-1331 |

Add mapping

Storage map: SM

| | |
|---|---|
| rt_580_nfs_pagefiles_001 | gold-block |
| rt_580_nfs_vmds_1001 | gold-block |

Add mapping

# Migration from VMWare ESXi

## COLD migration steps

- Shutdown VM on Vmware manually
- Start migration plan
- Disks synced and refractored by VDDK init image
- Check VM settings
- Start VM on OpenShift Virtualization
- VMware Tools replaced by Qemu Guest Agent automatically

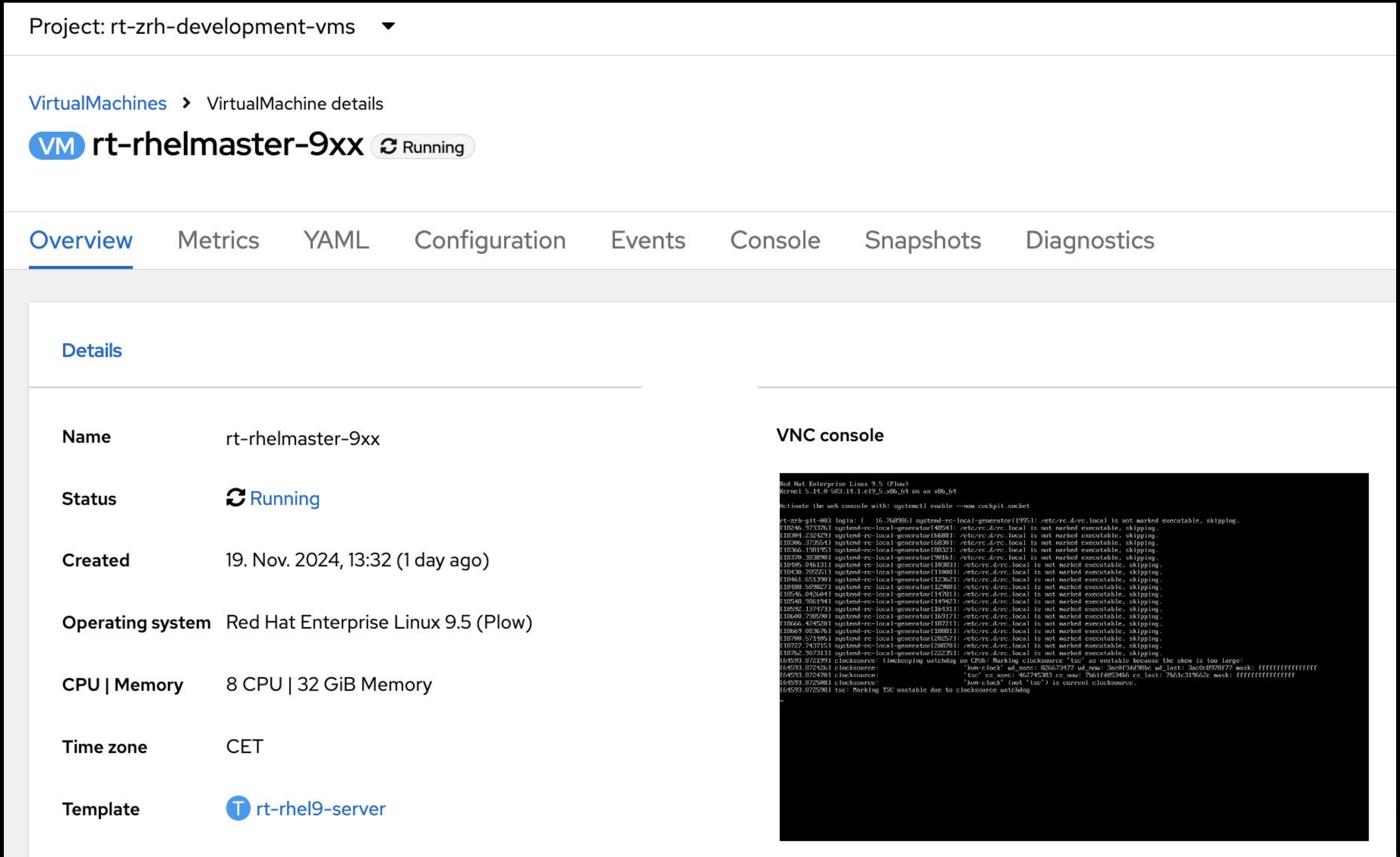
# Migration from VMWare ESXi

## WARM migration steps

- Start migration plan
  - □ Automatic snapshot on VMWare and incremental sync every hour

- Cutover when ready - instant or scheduled
- VM shuts down on VMWare
- Last incremental sync and refractoring by VDDK init image
- VM starts on OpenShift Virtualization
- VMWare Tools replaced by Qemu Guest Agent automatically

# Virtual Machine GUI



Project: rt-zrh-development-vms ▾

VirtualMachines › VirtualMachine details

VM rt-rhelmaster-9xx Running

Overview | Metrics | YAML | Configuration | Events | Console | Snapshots | Diagnostics

**Details**

| | |
|---|---|
| **Name** | rt-rhelmaster-9xx |
| **Status** | Running |
| **Created** | 19. Nov. 2024, 13:32 (1 day ago) |
| **Operating system** | Red Hat Enterprise Linux 9.5 (Plow) |
| **CPU \| Memory** | 8 CPU \| 32 GiB Memory |
| **Time zone** | CET |
| **Template** | T rt-rhel9-server |

**VNC console**

```
Red Hat Enterprise Linux 9.5 (Plow)
Kernel 5.14.0-503.14.1.el9_5.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

rt-zrh-git-003 login: [   16.768986] systemd-rc-local-generator[1995]: /etc/rc.d/rc.local is not marked executable, skipping.
[18246.973376] systemd-rc-local-generator[4054]: /etc/rc.d/rc.local is not marked executable, skipping.
[18304.232429] systemd-rc-local-generator[6688]: /etc/rc.d/rc.local is not marked executable, skipping.
[18306.373554] systemd-rc-local-generator[6830]: /etc/rc.d/rc.local is not marked executable, skipping.
[18366.190195] systemd-rc-local-generator[8832]: /etc/rc.d/rc.local is not marked executable, skipping.
[18370.383890] systemd-rc-local-generator[9016]: /etc/rc.d/rc.local is not marked executable, skipping.
[18405.046131] systemd-rc-local-generator[10383]: /etc/rc.d/rc.local is not marked executable, skipping.
[18430.785551] systemd-rc-local-generator[11000]: /etc/rc.d/rc.local is not marked executable, skipping.
[18461.651390] systemd-rc-local-generator[12362]: /etc/rc.d/rc.local is not marked executable, skipping.
[18488.509827] systemd-rc-local-generator[12980]: /etc/rc.d/rc.local is not marked executable, skipping.
[18546.042604] systemd-rc-local-generator[14781]: /etc/rc.d/rc.local is not marked executable, skipping.
[18548.906194] systemd-rc-local-generator[14942]: /etc/rc.d/rc.local is not marked executable, skipping.
[18592.137473] systemd-rc-local-generator[16431]: /etc/rc.d/rc.local is not marked executable, skipping.
[18608.798590] systemd-rc-local-generator[16917]: /etc/rc.d/rc.local is not marked executable, skipping.
[18666.424520] systemd-rc-local-generator[18721]: /etc/rc.d/rc.local is not marked executable, skipping.
[18669.083676] systemd-rc-local-generator[18881]: /etc/rc.d/rc.local is not marked executable, skipping.
[18700.571405] systemd-rc-local-generator[20257]: /etc/rc.d/rc.local is not marked executable, skipping.
[18727.743715] systemd-rc-local-generator[20870]: /etc/rc.d/rc.local is not marked executable, skipping.
[18762.907313] systemd-rc-local-generator[22235]: /etc/rc.d/rc.local is not marked executable, skipping.
[64593.872199] clocksource: timekeeping watchdog on CPU6: Marking clocksource 'tsc' as unstable because the skew is too large:
[64593.872426] clocksource:                       'kvm-clock' wd_nsec: 826673477 wd_now: 3ac0f9dd98bc wd_last: 3ac0c8978f77 mask: ffffffffffffffff
[64593.872470] clocksource:                       'tsc' cs_nsec: 462745383 cs_now: 7b61fd0534b6 cs_last: 7b61c319662c mask: ffffffffffffffff
[64593.872508] clocksource:                       'kvm-clock' (not 'tsc') is current clocksource.
[64593.872598] tsc: Marking TSC unstable due to clocksource watchdog
_
```

Public

# Failover scenarios

Planned failover…

- Works out of the box on node maintenance
- Live Migration takes place for VMs with RWX storage / set evition strategy
- Other VMs are powered off and started on another schedulable node

… and unplanned failover

- Achieved with Node Health Check Operator
- Self Node Remediation or Fence Agent Remediation
- Customizable remediation strategy / minimum of healthy nodes

# Backup

## Challenge

- Regular OADP / Velero Backups to S3 bucket
  - □ Takes too long for TBs / PB of data every night

## Our solution

- Use CSI volume snapshots and only copy Metadata to S3 buckets
- Move snapshots to backup datacenter with NetApp SnapMirror
- Configure NetApp and Tape backup through REST API with Ansible to “understand” OpenShift labels

REiST

IT SOLUTIONS FOR TODAY & TOMORROW

# Q & A