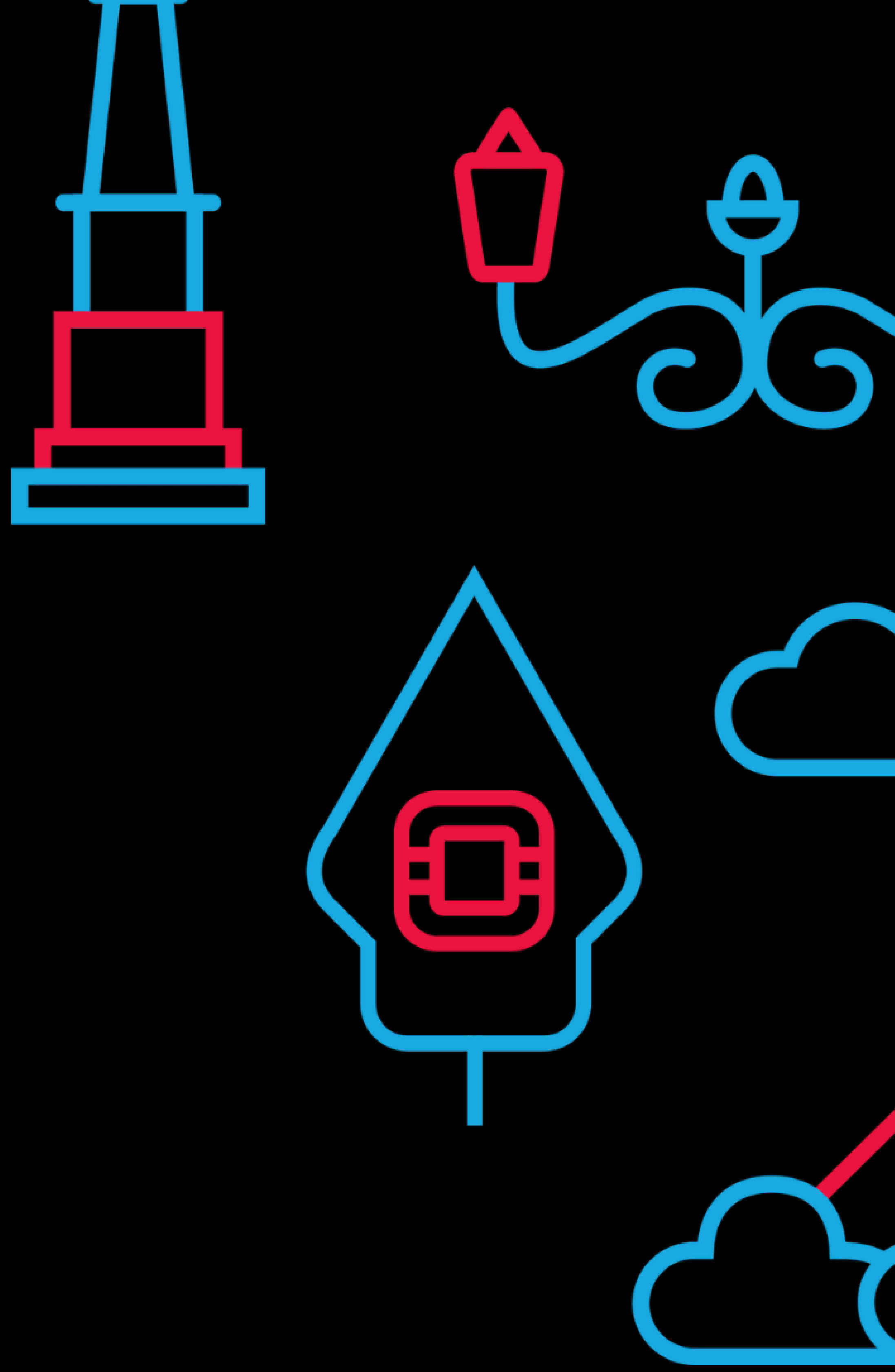
Exit Plan dari VMware
Strategi Migrasi ke OpenStack w/ vJailbreak

Muhammad Ilham Syarifuddin

VP Engineering

boer
technology



Yogyakarta, 19 July 2025

Why an Exit Plan is Crucial?

The Challenge

- Broadcom Acquisition (Late 2023) fundamentally changed VMware's licensing model
- Significant Cost Increases affecting IT budgets across organizations
- Increased vendor dependency risk and reduced flexibility
- Strategic shift forcing IT practitioners to reconsider infrastructure choices

The Opportunity

- Migration to open-source private cloud platforms
- Reduced licensing costs and vendor lock-in
- Greater flexibility and infrastructure control
- Strategic timing for seamless transition planning

openstack.

Enterprise-grade Cloud Platform

PROXMOX

Open Source Virtualization Platform

MicroStack

Lightweight OpenStack Distribution

OpenStack as Strategic Alternative

Embracing open-source infrastructure for enhanced flexibility, cost efficiency, and enterprise-grade reliability

Flexibility

Complete control over your infrastructure.

Customize each cloud service component to match your specific requirements and business needs.

FULL CONTROL

Cost-Effective

Eliminate expensive licensing fees and reduce **TCO**

with predictable, transparent pricing models that scale with your business.

NO LICENSE FEE

Enterprise Grade

Proven reliability with numerous enterprise distributions.

Battle-tested in production environments across global organizations.

PRODUCTION READY

OpenInfra FOUNDATION

Community

Backed by a vibrant global ecosystem of developers, vendors,

and enterprises continuously driving innovation and best practices.

GLOBAL ECOSYSTEM

Migration Journey

Embracing open-source infrastructure for enhanced flexibility, cost efficiency, and enterprise-grade reliability

PLAN

Assessment App & VMs
Inventory discovery & dependency mapping

Gather Platform Info
VMware/OpenStack credentials and configurations

Success Metrics
KPIs, acceptance criteria & Migration Timeline

IMPLEMENT

Integrate Platforms
Connect OpenStack, VMware, and migration tools

Understand Mappings
Resource mapping and workflow configuration

Configure Migration Types
Define conversion and migration patterns

PILOTING

Test / PoC
Execute proof of concept migrations

Finalize Test VMs
Select representative workloads for testing

Gather Feedback
Collect performance & issues data. Refine migration procedures

SCALE 1-X

Execute Migration Plans
Follow defined timeline and phases

Migrate Production
Move live applications and VMs

Optimize & Monitor
Track performance, costs, & user satisfaction. Post-migration Validation

Success Metrics
Migration success rate >95%, Performance parity or improvement, Zero data loss, Downtime within agreed SLA, Cost reduction achievement.

Migration Tools

vJailbreak

Open-source tool for seamless migration by handling disk conversion & driver injection. Known for its **GUI & Kubernetes-based** execution.

Type: **Agentless**
Interface: **GUI** & **CLI**
Execution: **Runs on Kubernetes at destination**
Dependencies**: Python 3.6+, OpenStack APIs, VMware vSphere APIs, VDDK**

MigrateKit

Open-source, CLI-focused tool for automation and scripting. Uses Docker for execution & offers optional commercial support via **Vexxhost**.

Type: **Agentless**
Interface: **CLI**
Execution: **Runs in Container at destination**
Dependencies: **Docker Container, OpenStack APIs, VMware vSphere APIs, VDDK**

ZConverter Cloud

Enterprise-grade solution supporting **any-to-any cloud migration** with advanced features like live migration for zero-downtime cutovers.

Type: **Agent-based**
Interface: **GUI**
Execution: **Requires agent on source VM**
Dependencies**: ZConverter appliance, vCenter access**

vinchin

Vinchin provides agentless backup, disaster recovery, and **seamless V2V migration** for virtual infrastructures.

Type: **Agentless**
Interface: **GUI**
Execution: **Requires agent on source VM**
Dependencies: **Vinchin Backup Server appliance, OpenStack APIs, VMware vSphere APIs**

Tool Selection Matrix

Tool	Strategy Fit	Interface	Agent Model	Key Dependencies	Best Applications
vJailbreak	**Lift & Shift**	GUI	Agentless	VDDK, Kubernetes, OpenStack APIs	Web servers, databases, business apps
MigrateKit	**Lift & Shift**	CLI	Agentless	VDDK, Docker, OpenStack APIs	Enterprise apps, automated workflows
ZConverter	**Hybrid, Backup-Driven**	GUI/CLI	Agent-based	ZConverter appliance, VM agents	Multi-hypervisor, complex networking
Vinchin	**Hybrid, Backup-Driven**	GUI/CLI	Agent-based	Vinchin agents, backup repository	Mission-critical, data-sensitive

Quick Selection Guide

Need GUI + Easy Setup
vJailbreak (runs on Kubernetes)

CLI + Enterprise Support
MigrateKit (with Vexxhost support)

Multi-source Migration
ZConverter (supports multiple hypervisors)

High-risk Systems
Vinchin, ZConverter (backup-driven approach)

Cloud-native Goals
Heat templates, Terraform (rebuild)

Decision Matrix

Simple + Fast
Jailbreak for web apps, file servers

Enterprise + Automated
MigrateKit for large-scale deployments

Complex + Multi-source
ZConverter for mixed environments

Critical + Protected
Vinchin for mission-critical systems

Modern + Optimized
Custom development for cloud-native

Migration Strategies & Workload Analysis

Matching the Approach to the Application

LIFT & SHIFT (REHOST)

Migrating as a VM as-is, without redesigning its architecture. This is the fastest approach with minimal changes

Example Workloads :

- Web Servers (Apache HTTP Server, Nginx, IIS)
- Application Servers (Tomcat, JBoss, WebLogic)
- File Servers (Windows File Server, Samba, NFS)
- Development/Test (CI/CD environments, staging servers)
- Business/Internal (CMS, legacy applications, etc)

Recommended Tools

vJailbreak, MigrateKit, ZConverter, Vinchin

Implementation

VM discovery → Network mapping → Storage conversion → Metadata preservation

RE-PLATFORM (REBUILD)

Completely rewriting applications for OpenStack native features maximizes platform benefits but requires the most effort.

Example Workloads :

- Cloud-Native Apps (Microservices, containerized apps)
- Data Analytics (Hadoop, Spark, Elasticsearch clusters)
- High-Performance (HPC apps, GPU workloads)
- Scalable Services (LB web services, API gateways)
- Modern Databases (MongoDB, Cassandra, Redis clusters)

Recommended Tools

CI/CD (Jenkins, GitLab), Docker, Kubernetes (Magnum), Ansible

Implementation

App modernization → Microservices adoption → Auto-scaling → Security hardening

HYBRID APPROACH

Combining strategies. For example, 'lift & shift' a legacy database, while rebuilding its front-end as a modern, containerized application.

Example Workloads :

- ERP Systems (SAP/Oracle with a legacy DB but a new web portal)
- Financial Apps (Core banking backends (lift & shift) with new mobile front-ends (rebuild))
- Large Websites (CMS backend (lift & shift) with a decoupled, scalable front-end)

Recommended Tools

A combination of all tools, tailored to each component.

Implementation

Workload categorization → Selective migration → Progressive optimization

Advanced Migration Implementation Details

Matching the Approach to the Application

LIFT & SHIFT (REHOST)

Detailed Execution Process

Pre-Migration
VM inventory, dependency mapping, performance baseline establishment

Conversion Phase
VMDK to QCOW2 conversion, driver injection, metadata extraction

Network Setup
Security group creation, floating IP allocation, router configuration

Validation
Boot testing, application functionality, performance verification

Cutover
DNS updates, load balancer changes, traffic redirection

RE-PLATFORM (REBUILD)

Cloud-Native Transformation

Architecture Review
Microservices decomposition, container strategy, API design

Infrastructure as Code
Heat templates, Ansible playbooks, Terraform modules

Auto-scaling Setup
Heat auto-scaling groups, monitoring integration, scaling policies

Security Enhancement
KeyManager integration, network policies, RBAC implementation

Cutover
DNS updates, load balancer changes, traffic redirection

HYBRID APPROACH

Best of Both

Workload Classification
Stateless for lift-and-shift, stateful for optimization

Phased Migration
Non-critical first, mission-critical with re-platform/rebuild

Data Synchronization
Real-time replication, ETL processes, backup/restore

Security Enhancement
KeyManager integration, network policies, RBAC implementation

Cutover
Blue-green deployment, canary releases, feature flags

vJailbreak Deep Dive

PLATFORM9

Released in early 2024, vJailbreak is an **open-source migration tool** developed by **Platform9** to facilitate the seamless migration of VMs from VMware vSphere to OpenStack. It simplifies the transition by providing a **user-friendly graphical interface** and automating the otherwise complex processes of VM conversion and workload migration.

Key Features & Benefits:

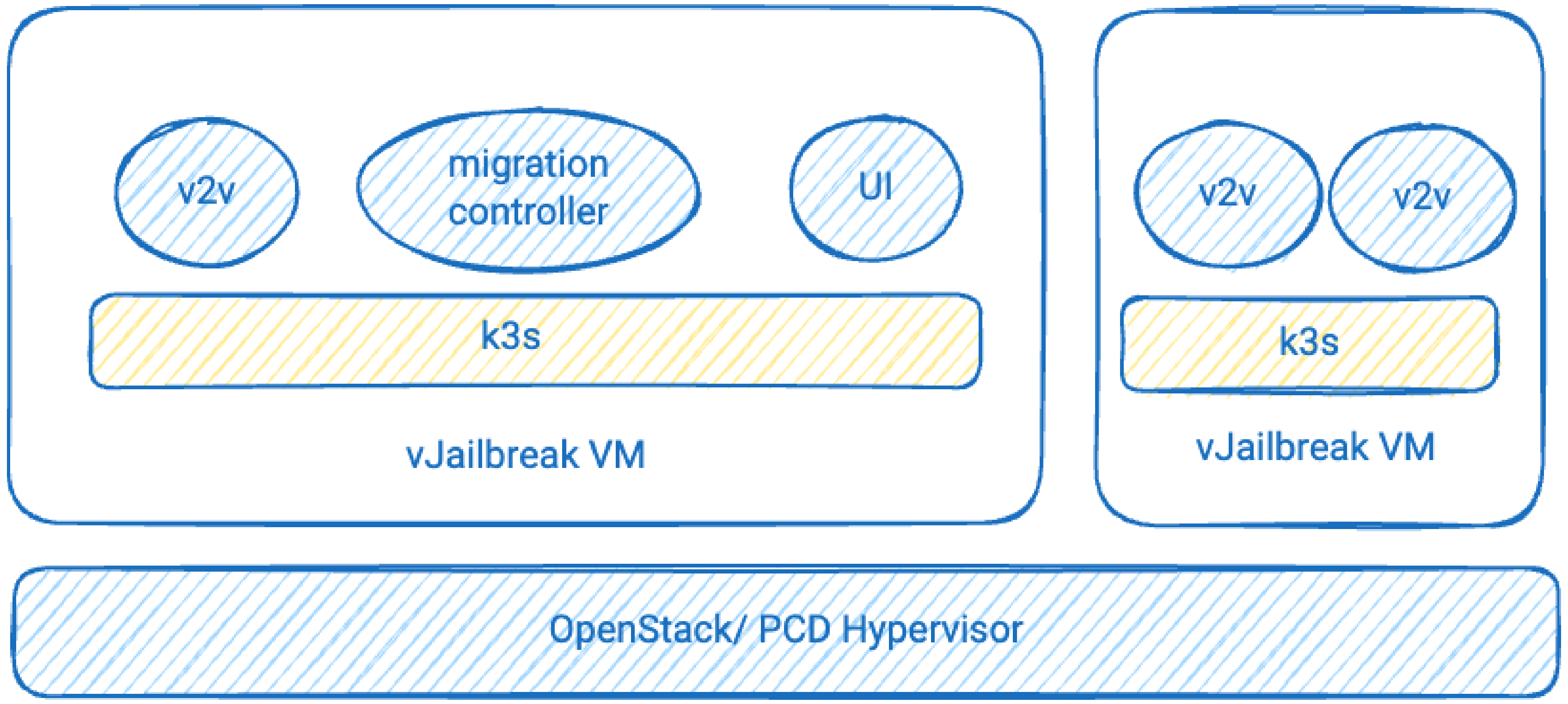
- Agentless migration (no software installation on source VMs)
- Kubernetes-based deployment for scalability
- Web-based GUI for easy operation
- VDDK integration for direct VMware access
- Minimal downtime during migration
- Automated format conversion (VMDK to QCOW2/RAW)
- Enterprise Scale (Proven with 40,000+ VM migrations)

Technical Specifications:

Python 3.6+ **Kubernetes 1.18+** **OpenStack Ussuri+** **vSphere 6.7+** **VDDK 7.0+** **Docker Runtime**

Supported Formats: **VMDK to QCOW2/RAW conversion**
Network Requirements: **API connectivity between VMware and OpenStack**
Resource Requirements: **8GB RAM, 4 vCPUs, 100GB+ Disk minimum for migration service**
OS Base: **Ubuntu 22.04 LTS with custom image (k3s)**

vJailbreak Architecture



Core Component :

- **v2v-helper** :executing migration process as a pod
- **v2v-cli**: Optional command-line interface tool (not required)
- **UI**: User-friendly graphical interface for migration management
- **migration-controller**: Kubernetes controller scheduling migration tasks
- **k3s**: Lightweight Kubernetes distribution for container orchestration

VM Deployment Architecture

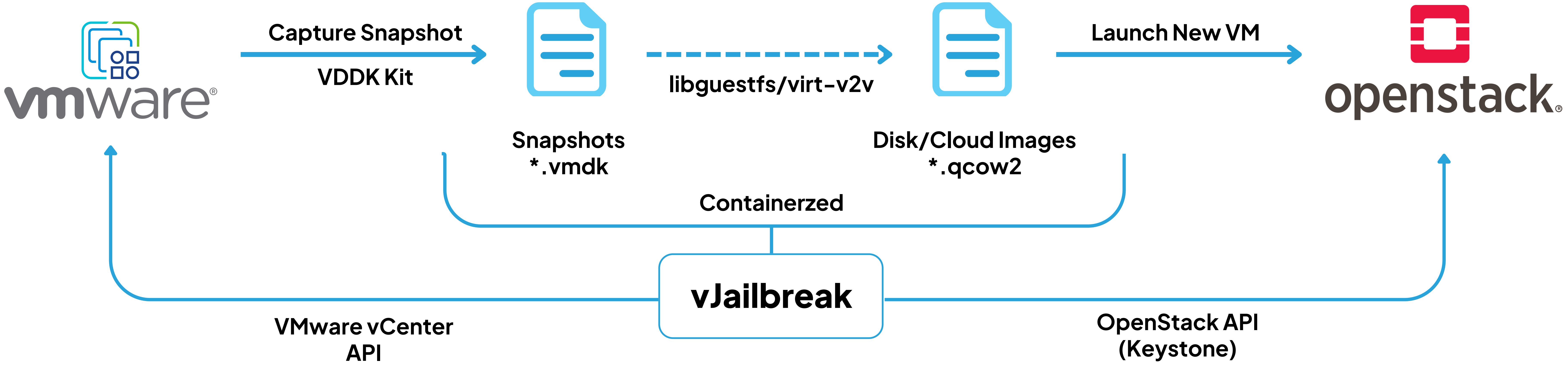
vJailbreak is deployed as a virtual machine in the target OpenStack environment. It uses a lightweight k3s Kubernetes cluster to manage its components and can be scaled by deploying additional VMs to handle concurrent migrations.

For high-throughput scenarios, the vJailbreak VM can be scaled out by deploying additional vJailbreak VMs in the target OpenStack environment to handle multiple migration tasks concurrently.

```
root@vjailbreak:~# kubectl get pods -n migration-system -o wide
NAME                                        READY STATUS    IP            NODE
migration-controller-manager-6d457f8477-bfcg7 1/1   Running   172.16.0.92   vjailbreak
migration-vpwned-sdk-5c9bc4655d-5zj54         1/1   Running   10.42.0.76    vjailbreak
vjailbreak-ui-64f494bb87-vd6wz                1/1   Running   10.42.0.75    vjailbreak
```

```
root@vjailbreak:~# crictl images | grep vjailbreak
IMAGE                                    TAG      IMAGE ID       SIZE
quay.io/platform9/vjailbreak-controller  v0.1.15  cb4b40f5971f8  37.1MB
quay.io/platform9/vjailbreak-ui          v0.1.15  72b2cfd9d9ad8  6.88MB
quay.io/platform9/vjailbreak-v2v-helper  v0.1.15  dc928c40769d7  1.16GB
quay.io/platform9/vjailbreak-vpwned      v0.1.15  870e61f8ae1d8  26.3MB
quay.io/platform9/vjailbreak             alpine   034d7219f2c78  5.6MB
```

vJailbreak Migration Workflow



VM Discovery
Connect to vCenter and scan available VMs with metadata

Volume Creation
Create OpenStack volume and attach to vJailbreak VM

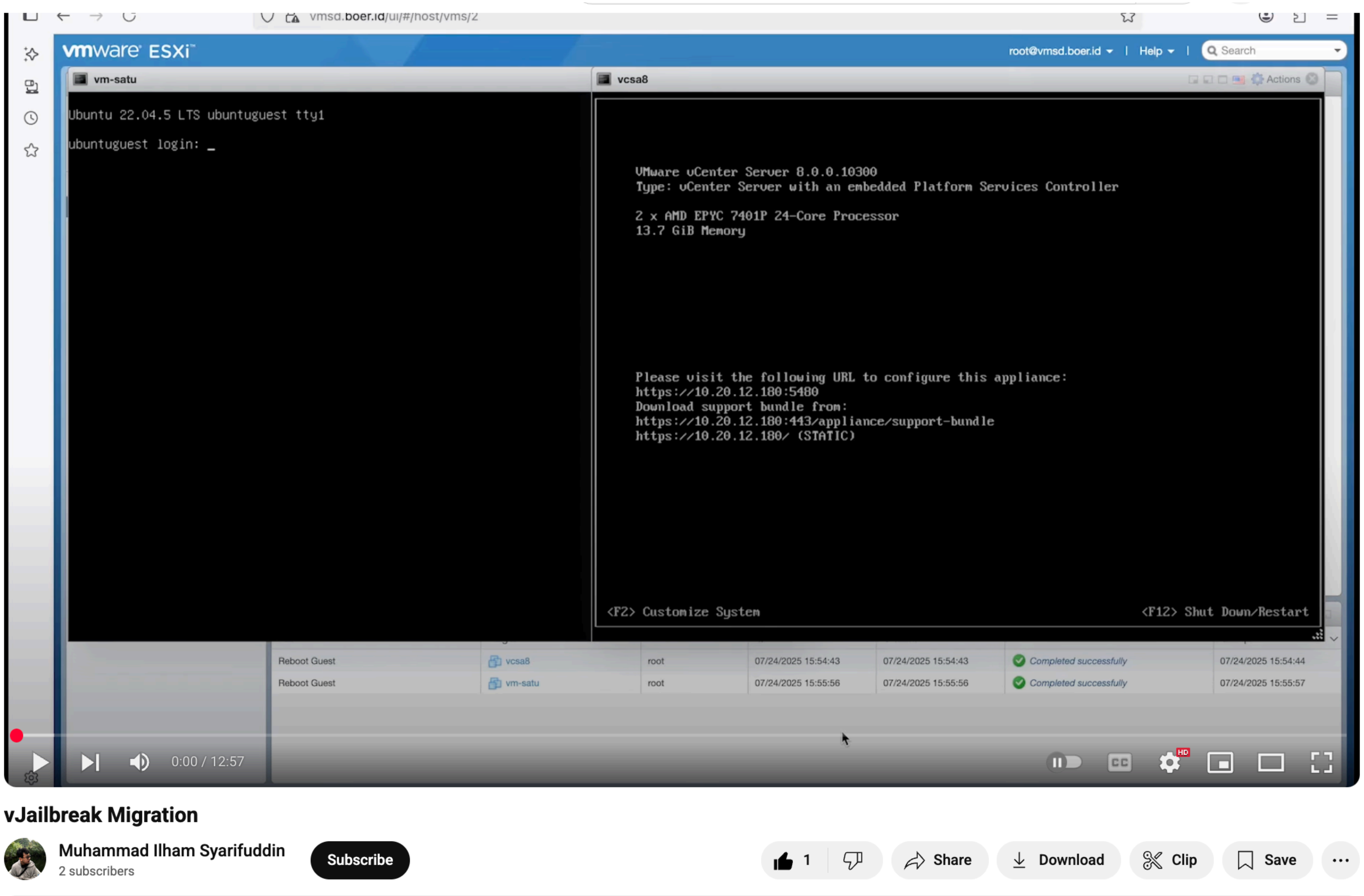
Data Migration
Use CBT to copy VM data while source remains running

Format Conversion
Use CBT to copy VM data while source remains running

VM Launch
Create new instance and perform final cutover

This architecture enables live migration with minimal downtime by leveraging Changed Block Tracking during the data copy phase, while the containerized conversion process ensures consistent and reliable format transformation from VMware VMDK to OpenStack-compatible QCOW2 format.

DEMO RECORD



```
Ubuntu 22.04.5 LTS ubuntuguest tty1

ubuntuguest login: _
```

```
VMware vCenter Server 8.0.0.10300
Type: vCenter Server with an embedded Platform Services Controller

2 x AMD EPYC 7401P 24-Core Processor
13.7 GiB Memory




Please visit the following URL to configure this appliance:
https://10.20.12.180:5480
Download support bundle from:
https://10.20.12.180:443/appliance/support-bundle
https://10.20.12.180/ (STATIC)



<F2> Customize System                                  <F12> Shut Down/Restart
```

Reboot Guest	vcsa8	root	07/24/2025 15:54:43	07/24/2025 15:54:43	Completed successfully	07/24/2025 15:54:44
Reboot Guest	vm-satu	root	07/24/2025 15:55:56	07/24/2025 15:55:56	Completed successfully	07/24/2025 15:55:57

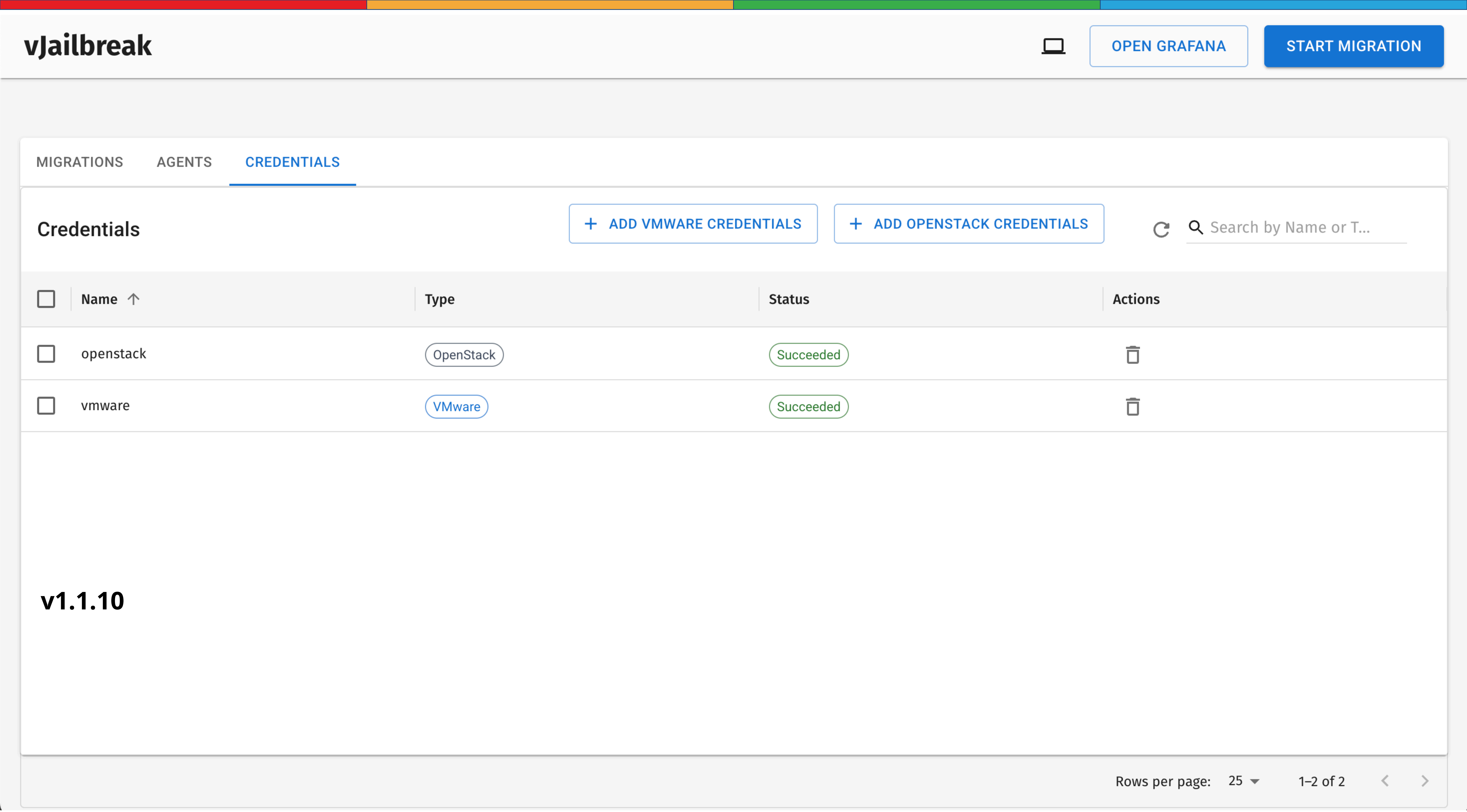
vJailbreak Migration

Muhammad Ilham Syarifuddin
2 subscribers

https://youtu.be/j23Ldz-hBB8?si=DwZZSa4cjcSYzcZ7

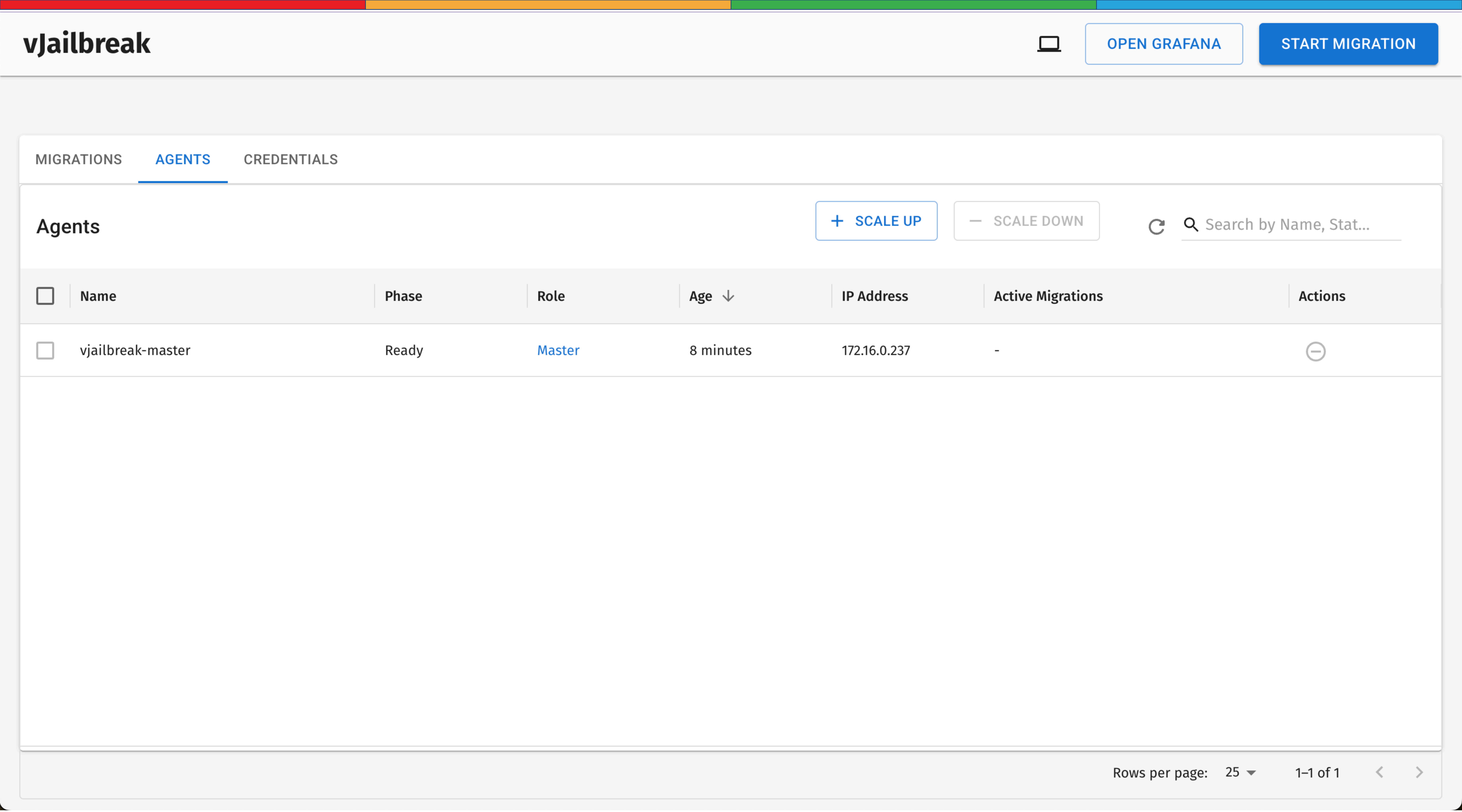
THANK YOU

Yogyakarta, 19 July 2025



□	Name ↑	Type	Status	Actions
□	openstack	OpenStack	Succeeded	🗑
□	vmware	VMware	Succeeded	🗑

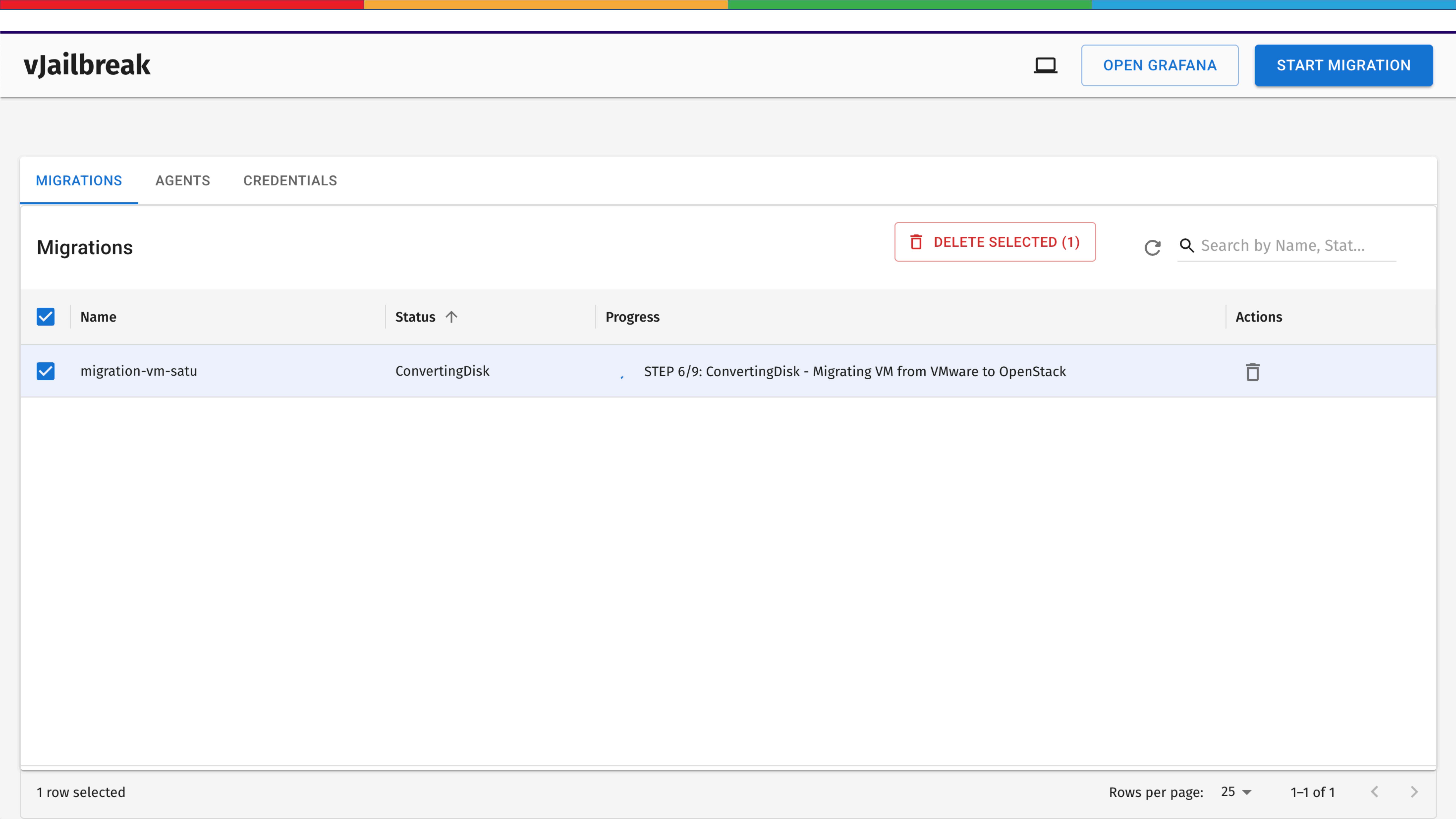
v1.1.10



Agents

☐	Name	Phase	Role	Age ↓	IP Address	Active Migrations	Actions
☐	vjailbreak-master	Ready	Master	8 minutes	172.16.0.237	-	⊖

Rows per page: 25 ▾ 1–1 of 1 ‹ ›

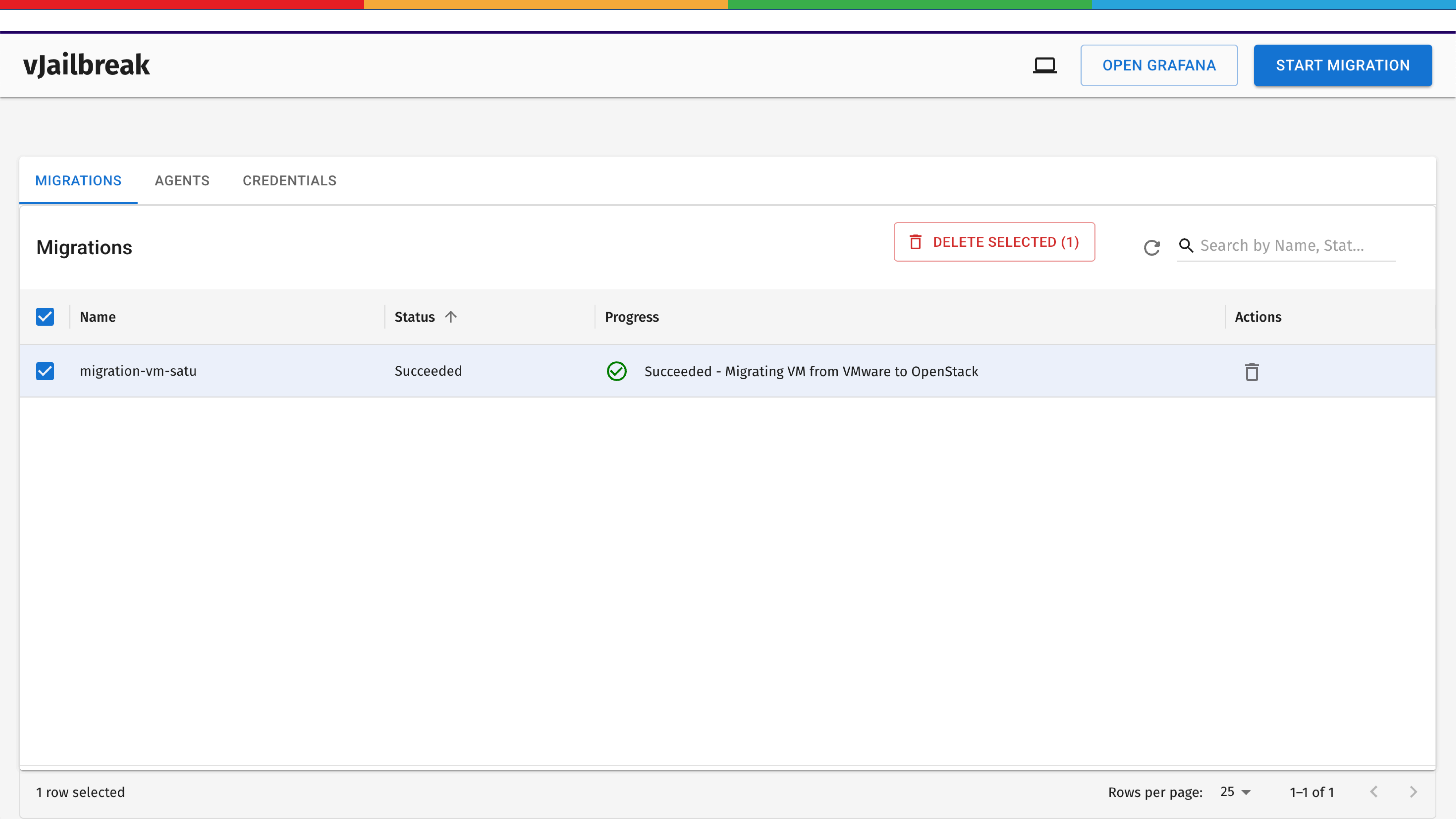


Migrations

☑	Name	Status ↑	Progress	Actions
☑	migration-vm-satu	ConvertingDisk	STEP 6/9: ConvertingDisk - Migrating VM from VMware to OpenStack	🗑

1 row selected

Rows per page: 25 1–1 of 1



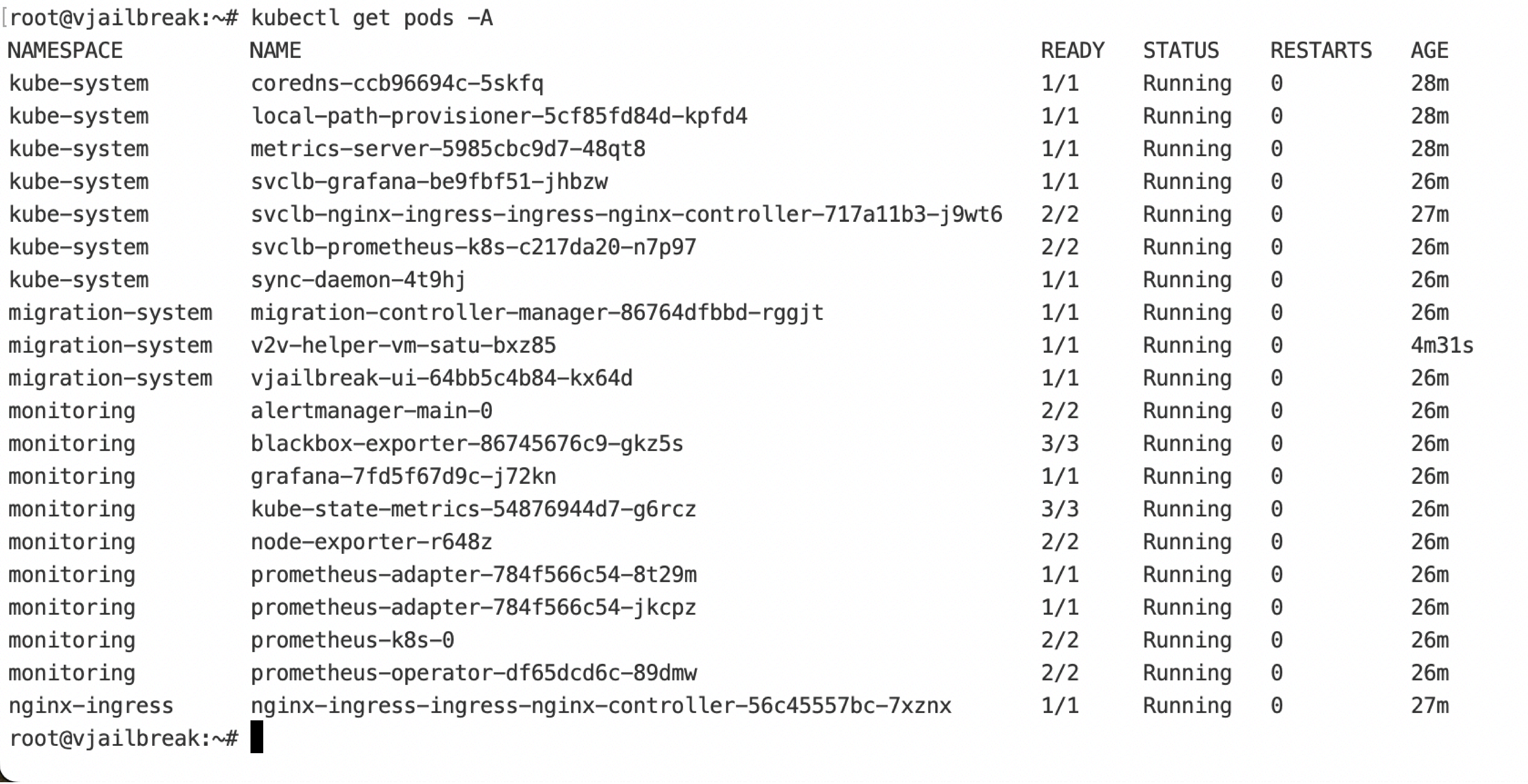
Migrations

☑	Name	Status ↑	Progress	Actions
☑	migration-vm-satu	Succeeded	✓ Succeeded - Migrating VM from VMware to OpenStack	🗑

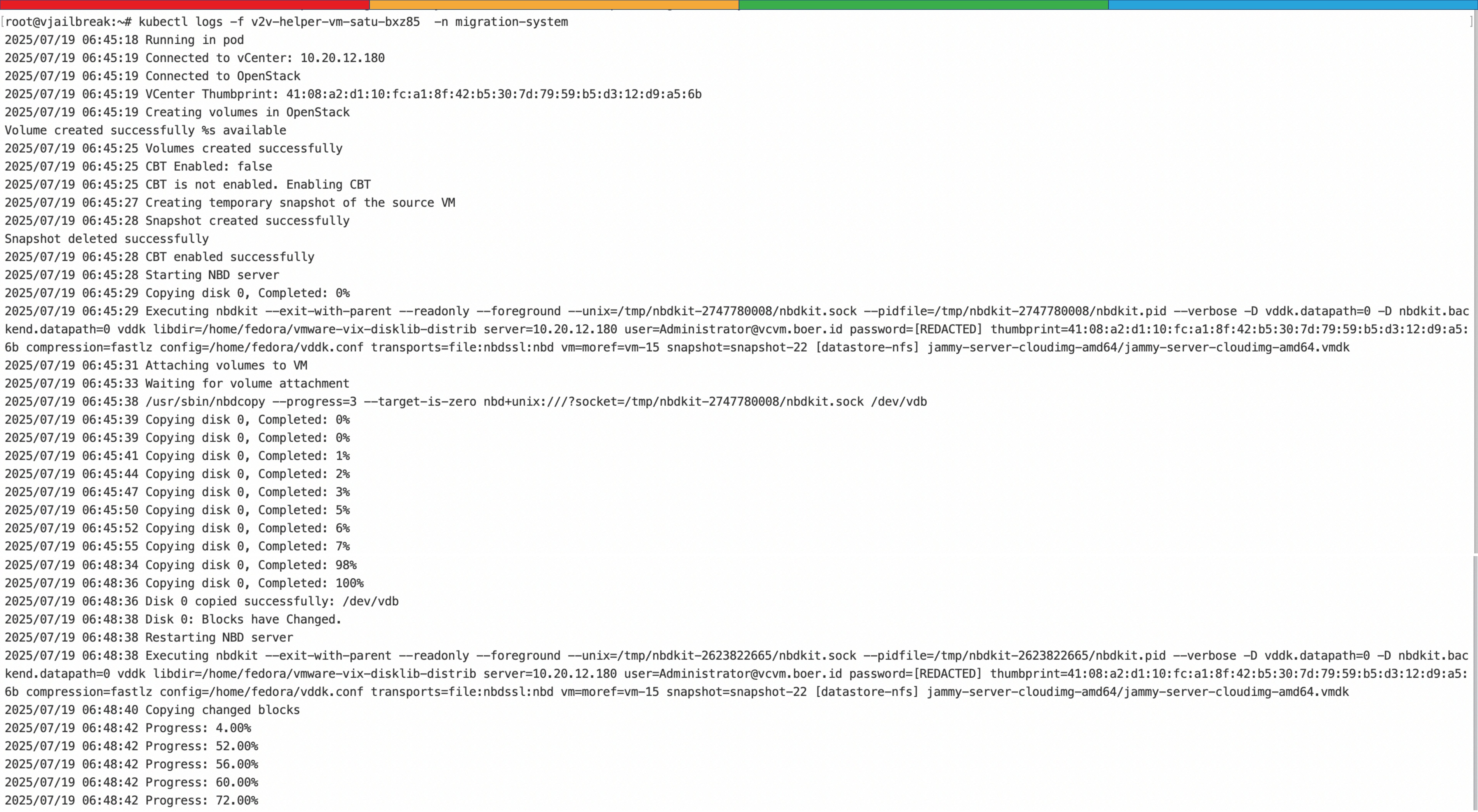
1 row selected

Rows per page: 25

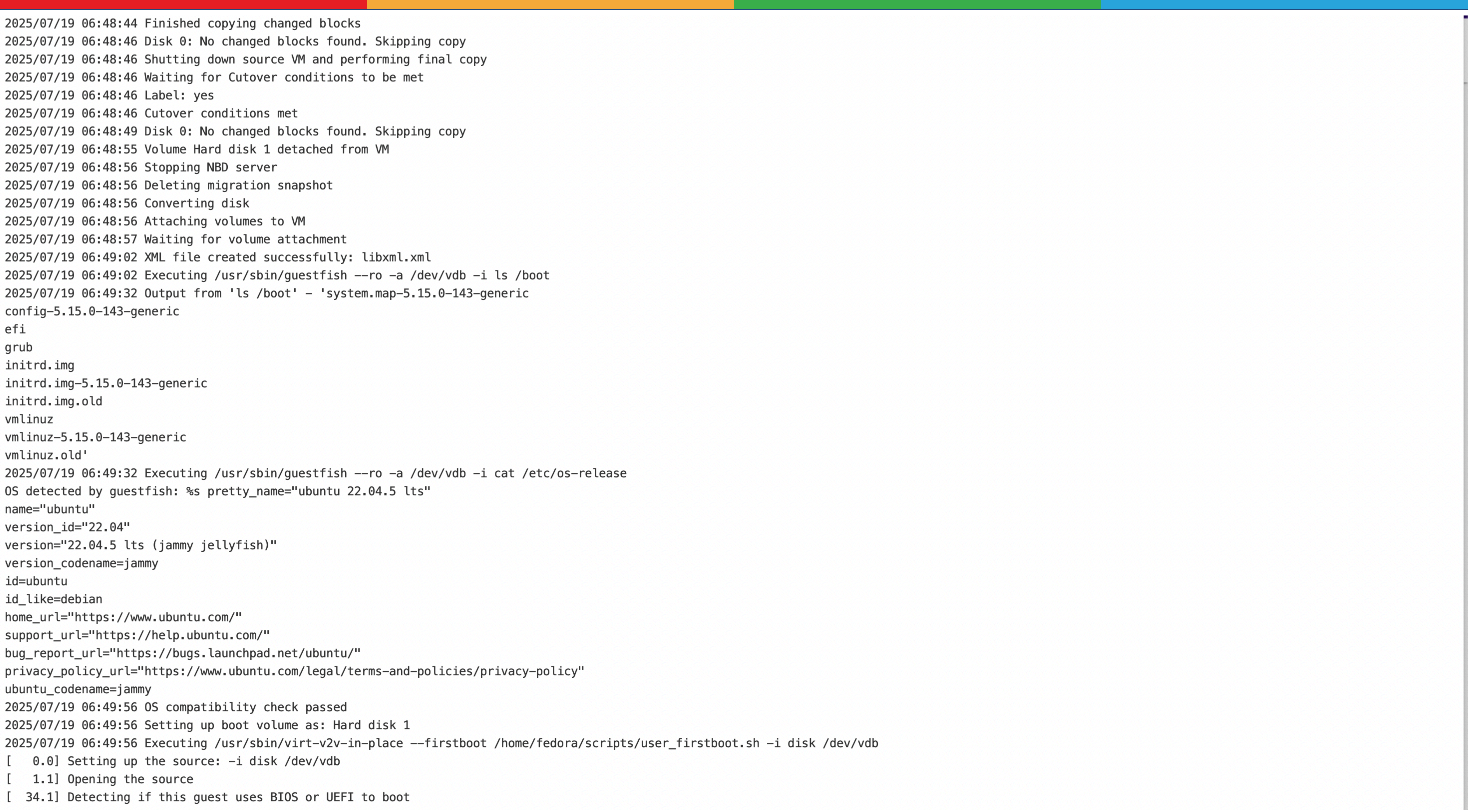
1–1 of 1



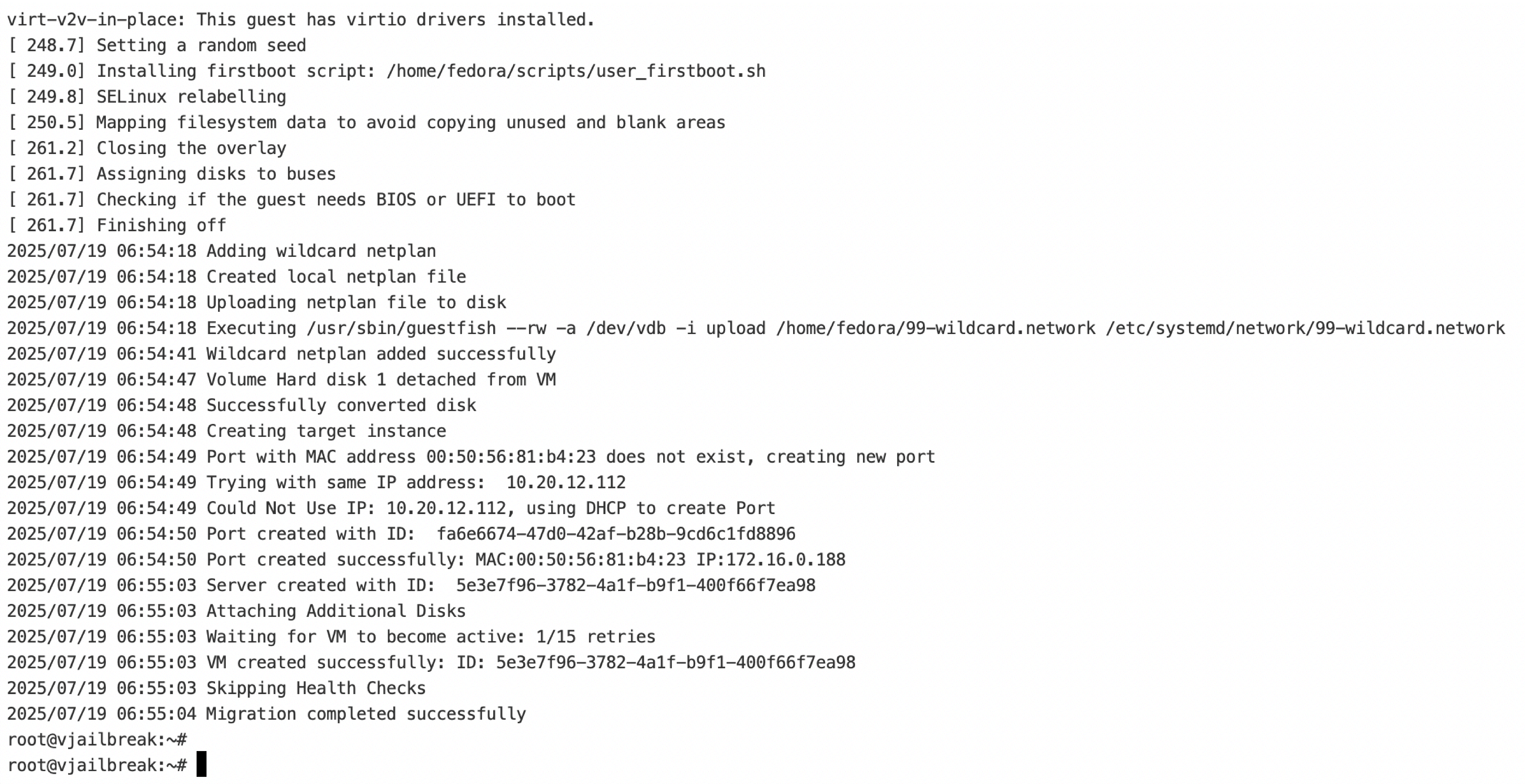
```
root@vjailbreak:~# kubectl get pods -A
NAMESPACE          NAME                                                         READY   STATUS    RESTARTS   AGE
kube-system        coredns-ccb96694c-5skfq                                      1/1     Running   0          28m
kube-system        local-path-provisioner-5cf85fd84d-kpfd4                      1/1     Running   0          28m
kube-system        metrics-server-5985cbc9d7-48qt8                              1/1     Running   0          28m
kube-system        svclb-grafana-be9fbf51-jhbzw                                 1/1     Running   0          26m
kube-system        svclb-nginx-ingress-ingress-nginx-controller-717a11b3-j9wt6  2/2     Running   0          27m
kube-system        svclb-prometheus-k8s-c217da20-n7p97                          2/2     Running   0          26m
kube-system        sync-daemon-4t9hj                                            1/1     Running   0          26m
migration-system   migration-controller-manager-86764dfbbd-rggjt                1/1     Running   0          26m
migration-system   v2v-helper-vm-satu-bxz85                                     1/1     Running   0          4m31s
migration-system   vjailbreak-ui-64bb5c4b84-kx64d                               1/1     Running   0          26m
monitoring         alertmanager-main-0                                          2/2     Running   0          26m
monitoring         blackbox-exporter-86745676c9-gkz5s                           3/3     Running   0          26m
monitoring         grafana-7fd5f67d9c-j72kn                                     1/1     Running   0          26m
monitoring         kube-state-metrics-54876944d7-g6rcz                          3/3     Running   0          26m
monitoring         node-exporter-r648z                                          2/2     Running   0          26m
monitoring         prometheus-adapter-784f566c54-8t29m                          1/1     Running   0          26m
monitoring         prometheus-adapter-784f566c54-jkcpz                          1/1     Running   0          26m
monitoring         prometheus-k8s-0                                             2/2     Running   0          26m
monitoring         prometheus-operator-df65dcd6c-89dmw                          2/2     Running   0          26m
nginx-ingress      nginx-ingress-ingress-nginx-controller-56c45557bc-7xznx      1/1     Running   0          27m
root@vjailbreak:~#
```



```
[root@vjailbreak:~# kubectl logs -f v2v-helper-vm-satu-bxz85  -n migration-system
2025/07/19 06:45:18 Running in pod
2025/07/19 06:45:19 Connected to vCenter: 10.20.12.180
2025/07/19 06:45:19 Connected to OpenStack
2025/07/19 06:45:19 VCenter Thumbprint: 41:08:a2:d1:10:fc:a1:8f:42:b5:30:7d:79:59:b5:d3:12:d9:a5:6b
2025/07/19 06:45:19 Creating volumes in OpenStack
Volume created successfully %s available
2025/07/19 06:45:25 Volumes created successfully
2025/07/19 06:45:25 CBT Enabled: false
2025/07/19 06:45:25 CBT is not enabled. Enabling CBT
2025/07/19 06:45:27 Creating temporary snapshot of the source VM
2025/07/19 06:45:28 Snapshot created successfully
Snapshot deleted successfully
2025/07/19 06:45:28 CBT enabled successfully
2025/07/19 06:45:28 Starting NBD server
2025/07/19 06:45:29 Copying disk 0, Completed: 0%
2025/07/19 06:45:29 Executing nbdkit --exit-with-parent --readonly --foreground --unix=/tmp/nbdkit-2747780008/nbdkit.sock --pidfile=/tmp/nbdkit-2747780008/nbdkit.pid --verbose -D vddk.datapath=0 -D nbdkit.bac
kend.datapath=0 vddk libdir=/home/fedora/vmware-vix-disklib-distrib server=10.20.12.180 user=Administrator@vcvm.boer.id password=[REDACTED] thumbprint=41:08:a2:d1:10:fc:a1:8f:42:b5:30:7d:79:59:b5:d3:12:d9:a5:
6b compression=fastlz config=/home/fedora/vddk.conf transports=file:nbdssl:nbd vm=moref=vm-15 snapshot=snapshot-22 [datastore-nfs] jammy-server-cloudimg-amd64/jammy-server-cloudimg-amd64.vmdk
2025/07/19 06:45:31 Attaching volumes to VM
2025/07/19 06:45:33 Waiting for volume attachment
2025/07/19 06:45:38 /usr/sbin/nbdcopy --progress=3 --target-is-zero nbd+unix:///?socket=/tmp/nbdkit-2747780008/nbdkit.sock /dev/vdb
2025/07/19 06:45:39 Copying disk 0, Completed: 0%
2025/07/19 06:45:39 Copying disk 0, Completed: 0%
2025/07/19 06:45:41 Copying disk 0, Completed: 1%
2025/07/19 06:45:44 Copying disk 0, Completed: 2%
2025/07/19 06:45:47 Copying disk 0, Completed: 3%
2025/07/19 06:45:50 Copying disk 0, Completed: 5%
2025/07/19 06:45:52 Copying disk 0, Completed: 6%
2025/07/19 06:45:55 Copying disk 0, Completed: 7%
2025/07/19 06:48:34 Copying disk 0, Completed: 98%
2025/07/19 06:48:36 Copying disk 0, Completed: 100%
2025/07/19 06:48:36 Disk 0 copied successfully: /dev/vdb
2025/07/19 06:48:38 Disk 0: Blocks have Changed.
2025/07/19 06:48:38 Restarting NBD server
2025/07/19 06:48:38 Executing nbdkit --exit-with-parent --readonly --foreground --unix=/tmp/nbdkit-2623822665/nbdkit.sock --pidfile=/tmp/nbdkit-2623822665/nbdkit.pid --verbose -D vddk.datapath=0 -D nbdkit.bac
kend.datapath=0 vddk libdir=/home/fedora/vmware-vix-disklib-distrib server=10.20.12.180 user=Administrator@vcvm.boer.id password=[REDACTED] thumbprint=41:08:a2:d1:10:fc:a1:8f:42:b5:30:7d:79:59:b5:d3:12:d9:a5:
6b compression=fastlz config=/home/fedora/vddk.conf transports=file:nbdssl:nbd vm=moref=vm-15 snapshot=snapshot-22 [datastore-nfs] jammy-server-cloudimg-amd64/jammy-server-cloudimg-amd64.vmdk
2025/07/19 06:48:40 Copying changed blocks
2025/07/19 06:48:42 Progress: 4.00%
2025/07/19 06:48:42 Progress: 52.00%
2025/07/19 06:48:42 Progress: 56.00%
2025/07/19 06:48:42 Progress: 60.00%
2025/07/19 06:48:42 Progress: 72.00%
```



```
2025/07/19 06:48:44 Finished copying changed blocks
2025/07/19 06:48:46 Disk 0: No changed blocks found. Skipping copy
2025/07/19 06:48:46 Shutting down source VM and performing final copy
2025/07/19 06:48:46 Waiting for Cutover conditions to be met
2025/07/19 06:48:46 Label: yes
2025/07/19 06:48:46 Cutover conditions met
2025/07/19 06:48:49 Disk 0: No changed blocks found. Skipping copy
2025/07/19 06:48:55 Volume Hard disk 1 detached from VM
2025/07/19 06:48:56 Stopping NBD server
2025/07/19 06:48:56 Deleting migration snapshot
2025/07/19 06:48:56 Converting disk
2025/07/19 06:48:56 Attaching volumes to VM
2025/07/19 06:48:57 Waiting for volume attachment
2025/07/19 06:49:02 XML file created successfully: libxml.xml
2025/07/19 06:49:02 Executing /usr/sbin/guestfish --ro -a /dev/vdb -i ls /boot
2025/07/19 06:49:32 Output from 'ls /boot' - 'system.map-5.15.0-143-generic
config-5.15.0-143-generic
efi
grub
initrd.img
initrd.img-5.15.0-143-generic
initrd.img.old
vmlinuz
vmlinuz-5.15.0-143-generic
vmlinuz.old'
2025/07/19 06:49:32 Executing /usr/sbin/guestfish --ro -a /dev/vdb -i cat /etc/os-release
OS detected by guestfish: %s pretty_name="ubuntu 22.04.5 lts"
name="ubuntu"
version_id="22.04"
version="22.04.5 lts (jammy jellyfish)"
version_codename=jammy
id=ubuntu
id_like=debian
home_url="https://www.ubuntu.com/"
support_url="https://help.ubuntu.com/"
bug_report_url="https://bugs.launchpad.net/ubuntu/"
privacy_policy_url="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
ubuntu_codename=jammy
2025/07/19 06:49:56 OS compatibility check passed
2025/07/19 06:49:56 Setting up boot volume as: Hard disk 1
2025/07/19 06:49:56 Executing /usr/sbin/virt-v2v-in-place --firstboot /home/fedora/scripts/user_firstboot.sh -i disk /dev/vdb
[   0.0] Setting up the source: -i disk /dev/vdb
[   1.1] Opening the source
[  34.1] Detecting if this guest uses BIOS or UEFI to boot
```



```
virt-v2v-in-place: This guest has virtio drivers installed.
[ 248.7] Setting a random seed
[ 249.0] Installing firstboot script: /home/fedora/scripts/user_firstboot.sh
[ 249.8] SELinux relabelling
[ 250.5] Mapping filesystem data to avoid copying unused and blank areas
[ 261.2] Closing the overlay
[ 261.7] Assigning disks to buses
[ 261.7] Checking if the guest needs BIOS or UEFI to boot
[ 261.7] Finishing off
2025/07/19 06:54:18 Adding wildcard netplan
2025/07/19 06:54:18 Created local netplan file
2025/07/19 06:54:18 Uploading netplan file to disk
2025/07/19 06:54:18 Executing /usr/sbin/guestfish --rw -a /dev/vdb -i upload /home/fedora/99-wildcard.network /etc/systemd/network/99-wildcard.network
2025/07/19 06:54:41 Wildcard netplan added successfully
2025/07/19 06:54:47 Volume Hard disk 1 detached from VM
2025/07/19 06:54:48 Successfully converted disk
2025/07/19 06:54:48 Creating target instance
2025/07/19 06:54:49 Port with MAC address 00:50:56:81:b4:23 does not exist, creating new port
2025/07/19 06:54:49 Trying with same IP address:  10.20.12.112
2025/07/19 06:54:49 Could Not Use IP: 10.20.12.112, using DHCP to create Port
2025/07/19 06:54:50 Port created with ID:  fa6e6674-47d0-42af-b28b-9cd6c1fd8896
2025/07/19 06:54:50 Port created successfully: MAC:00:50:56:81:b4:23 IP:172.16.0.188
2025/07/19 06:55:03 Server created with ID:  5e3e7f96-3782-4a1f-b9f1-400f66f7ea98
2025/07/19 06:55:03 Attaching Additional Disks
2025/07/19 06:55:03 Waiting for VM to become active: 1/15 retries
2025/07/19 06:55:03 VM created successfully: ID: 5e3e7f96-3782-4a1f-b9f1-400f66f7ea98
2025/07/19 06:55:03 Skipping Health Checks
2025/07/19 06:55:04 Migration completed successfully
root@vjailbreak:~#
root@vjailbreak:~# █
```