IONOS CLOUD

SUSE Rancher Prime on IONOS CLOUD: A Key Solution for Digital Independence in the DACH Region

Deployment Guide

Version 1.1, May 15th, 2026

Table of Content

Deploying Rancher Prime on the IONOS CLOUD

Deploying a High Availability (HA) SUSE Rancher Prime cluster on the IONOS CLOUD supports digital sovereignty strategies. By combining open-source software with a European cloud provider, data locations remain under control and operations stay independent. This article covers the reference architecture, manual configuration steps, and an automated Infrastructure-as-Code (IaC) approach. The Bring-Your-Own-Subscription (BYOS) model is used, with RKE2 forming the basis of Kubernetes. The focus is on deploying Rancher Manager.

The Architecture

Figure 1 shows the architecture. It combines recommendations from SUSE Rancher Prime and features of IONOS CLOUD. RKE2 is used as the underlying Kubernetes distribution, running on SUSE Linux Enterprise Server (SLES) 15 SP7.

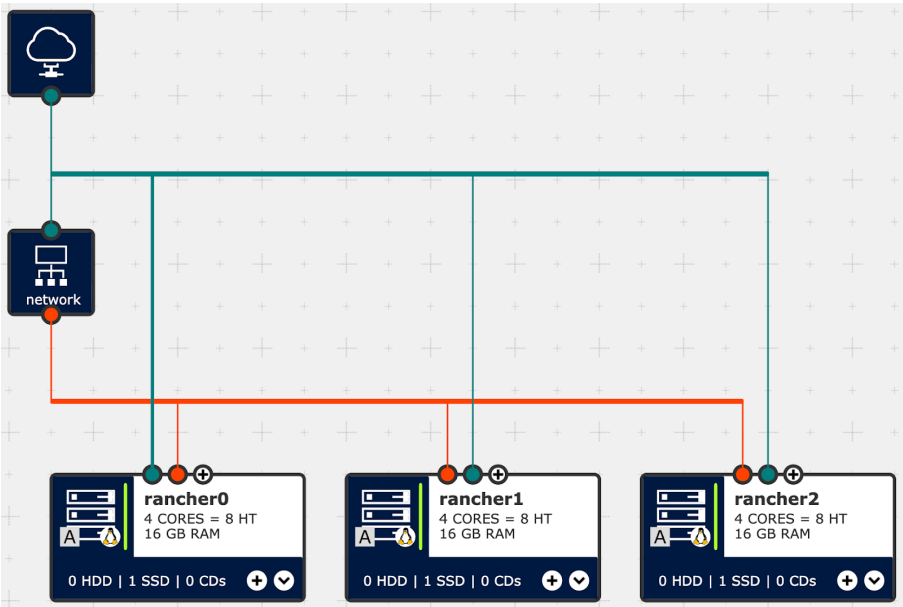

Figure 1: Rancher on IONOS CLOUD, infrastructure components architecture diagram.

Three Control Plane nodes are distributed across the infrastructure to form the Rancher Manager cluster. An IONOS CLOUD Network Load Balancer (NLB) serves as the entry point for data traffic. The network utilizes two separate LANs. A public LAN provides the nodes with internet access for updates and management, while a private LAN handles the internal

communication between the nodes. Incoming data traffic is routed from the Load Balancer to the cluster via the private network.

For computing resources, Dedicated Core servers and SSD Premium storage are used. IONOS CLOUD provides various compute options, with the complete physical core being exclusively available to the VM in Dedicated Core servers without over-provisioning. Likewise, different performance classes are available for Block Storage. The performance is already included in the service and is automatically provided without additional costs, depending on the size of the volume. The selected hardware options meet the requirements of SUSE Rancher Prime on RKE2 and offer low latency for the etcd data store.

Option 1: Manual Setup

The manual creation of the environment illustrates the interaction of the components. The IONOS CLOUD Data Center Designer (DCD) serves as the visual interface for this.

A new virtual data center is created. Figure 2 shows an example.

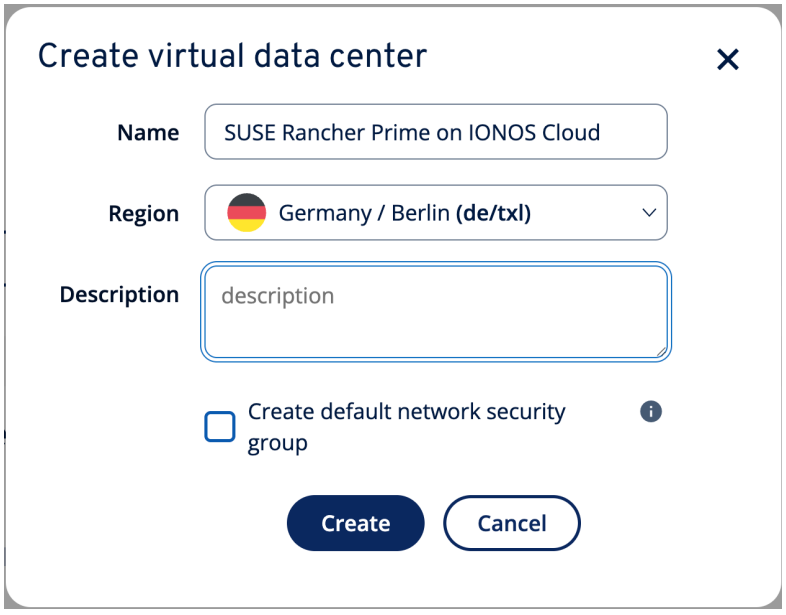

Figure 2: Dialog for a new virtual data center "SUSE Rancher Prime on IONOS CLOUD" in the region "Germany / Berlin (de/txl)"

For the cluster, three virtual machines are configured, each with 4 dedicated cores and 16 GB RAM. (Figure 3).

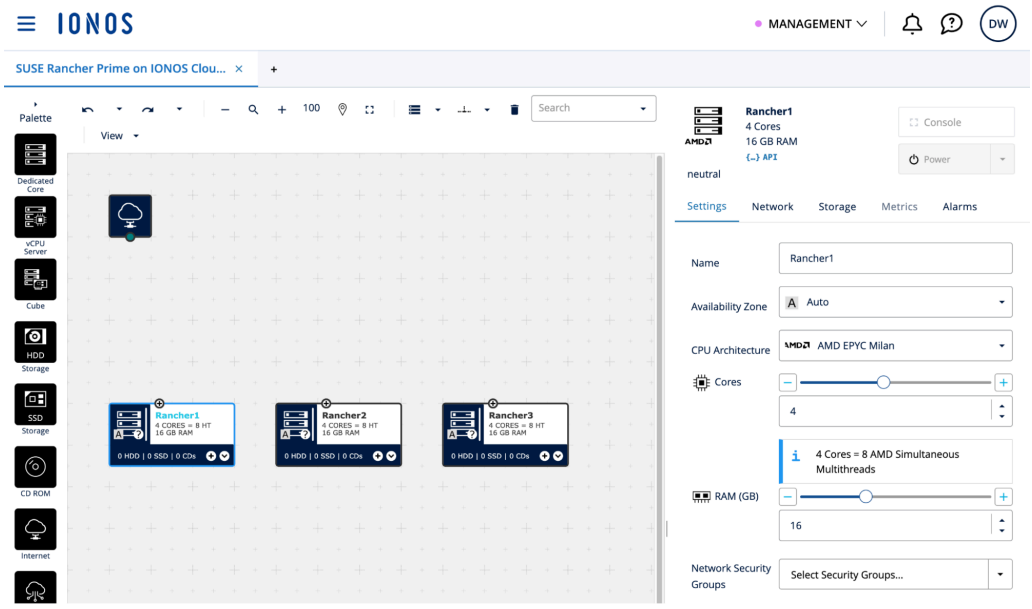

Figure 3: Three virtual machines in DCD with dedicated cores for the Rancher Manager cluster.

Each VM receives a 600 GB SSD Premium storage volume named "root" based on the SLES 15 SP7 BYOS image. IONOS CLOUD scales the I/O performance based on the volume size to provide the required IOPS for etcd (Figure 5). As previously described, the storage performance is already included in the service price. A Public SSH Key is added to the image to allow later access to the VMs via a terminal.

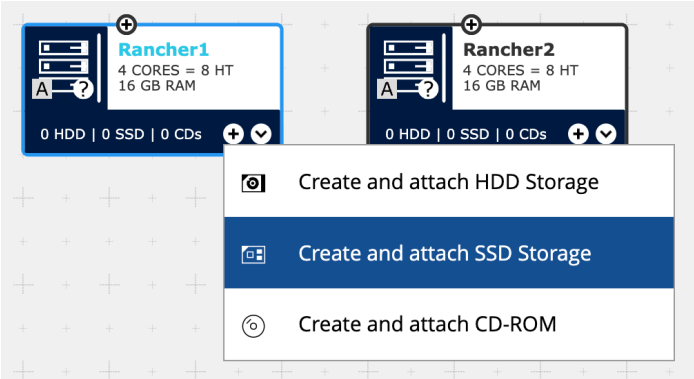

Figure 4: Creating and attaching a new SSD storage volume to the virtual machine “Rancher1”.

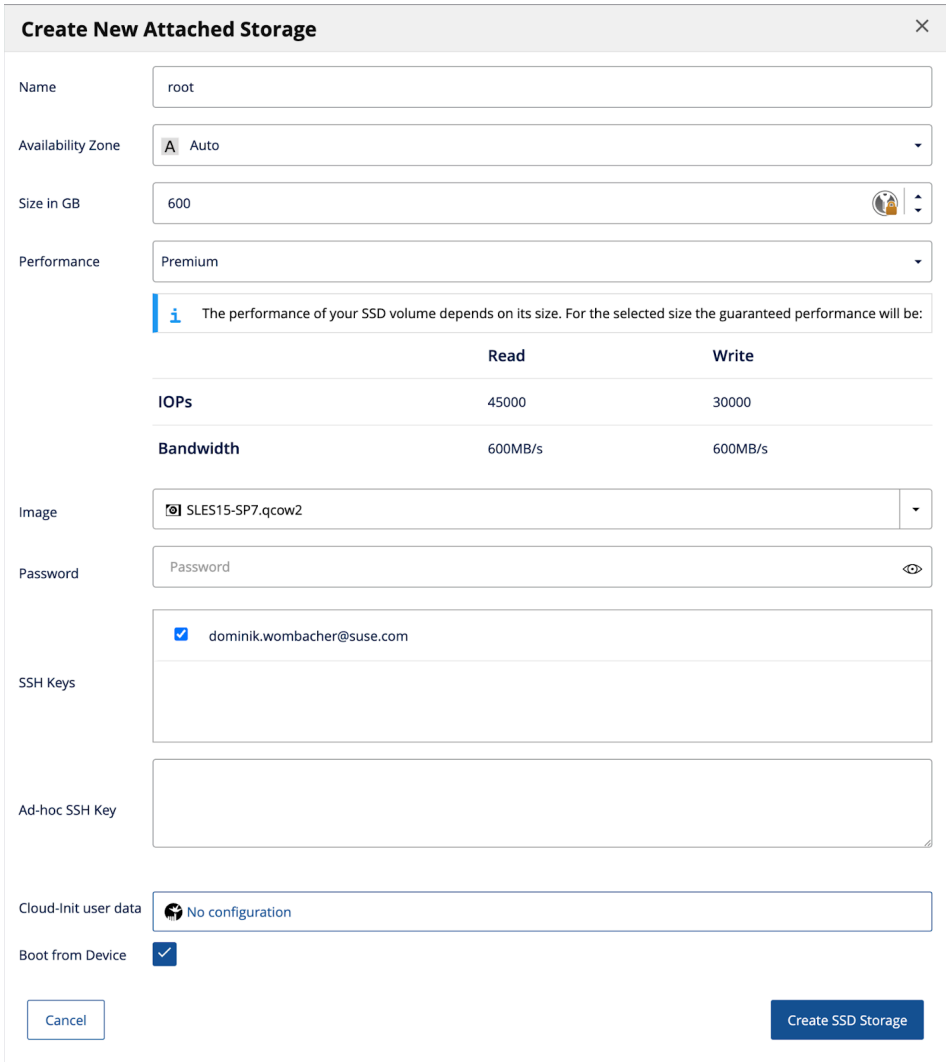

Figure 5: The new storage volume is called "root", has a size of "600 GB", the performance type "Premium", and uses the SLES15-SP7 BYOS image.

Figure 6 shows the three nodes with attached root volumes.

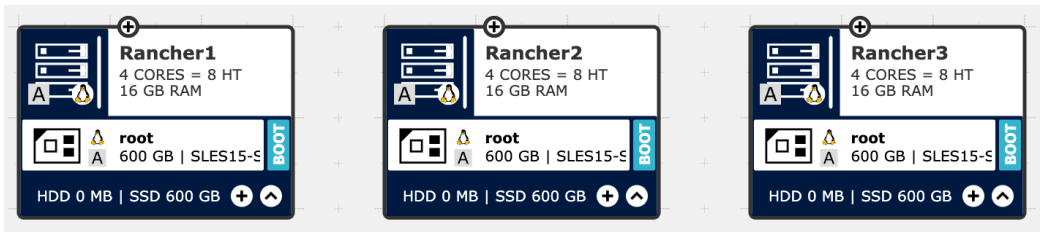

Figure 6: Virtual machines “Rancher1”, “Rancher2”, and “Rancher3” with the newly created root volume.

Each VM requires two network interfaces. One connects to the public LAN for the internet connection (Figure 7). The second connects to the private LAN for internal cluster data traffic (Figure 8). Further information: Networks in virtual data centers.

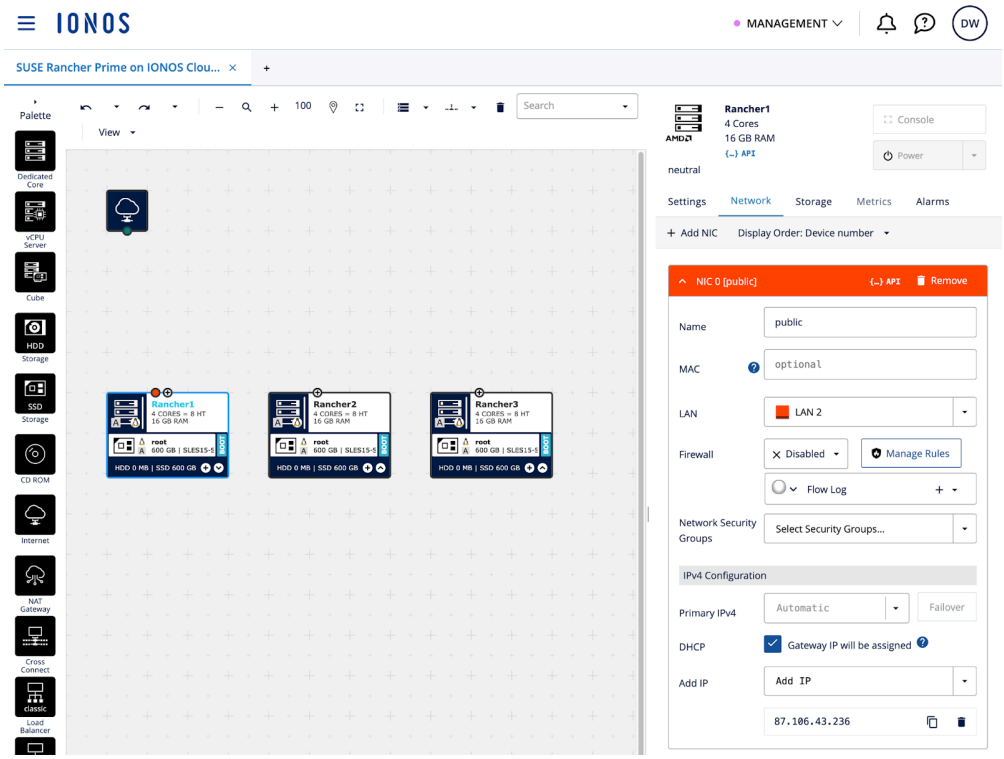

Figure 7: Public network interface on “Rancher1” with a static public IP address.

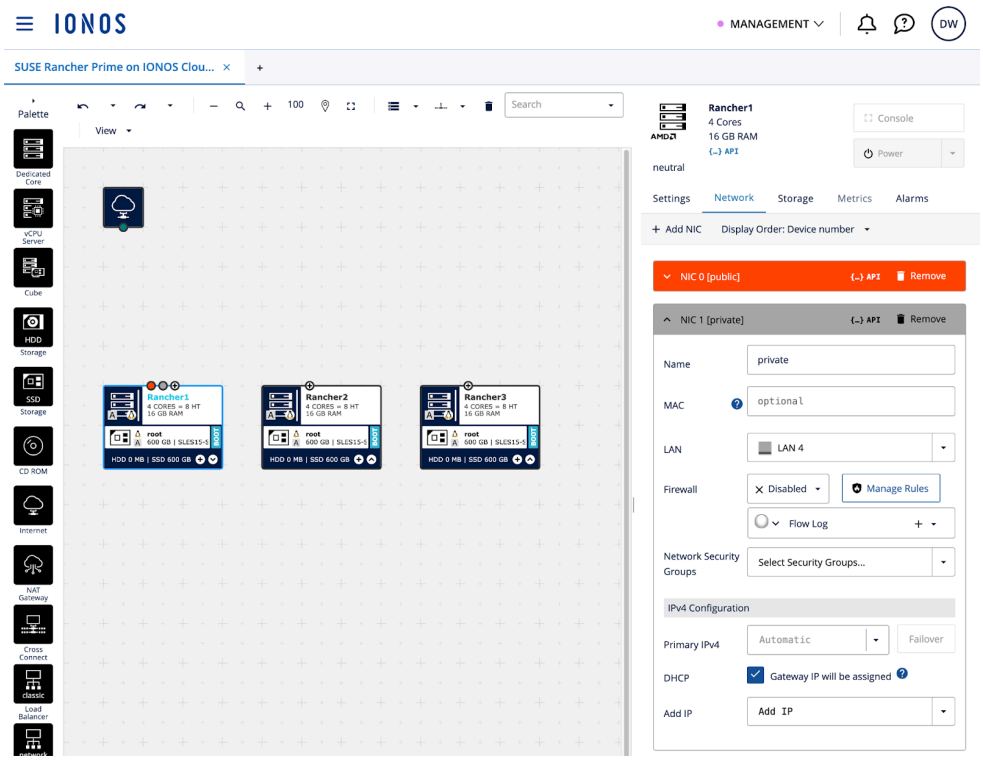

Figure 8: Private network interface on “Rancher1” with a dynamic IP address.

These steps are repeated on all three virtual machines and connected to the same LAN (NIC 0 = public, NIC 1 = private). The result is visible in Figure 9.

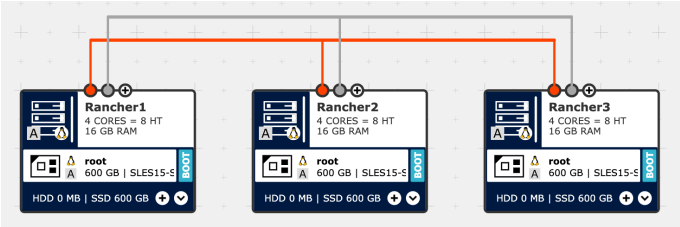

Figure 9: The three virtual machines “Rancher1”, “Rancher2”, and “Rancher3” are assigned public and private network interfaces and are connected to each other.

The IONOS CLOUD Network Load Balancer handles the incoming cluster traffic. It performs TCP health checks to find responsive targets. Subsequently, it forwards the traffic to active nodes. A new instance is created in the DCD, a public and a private IP address are added, and the interfaces are connected to the existing networks. The result is shown in Figure 10.

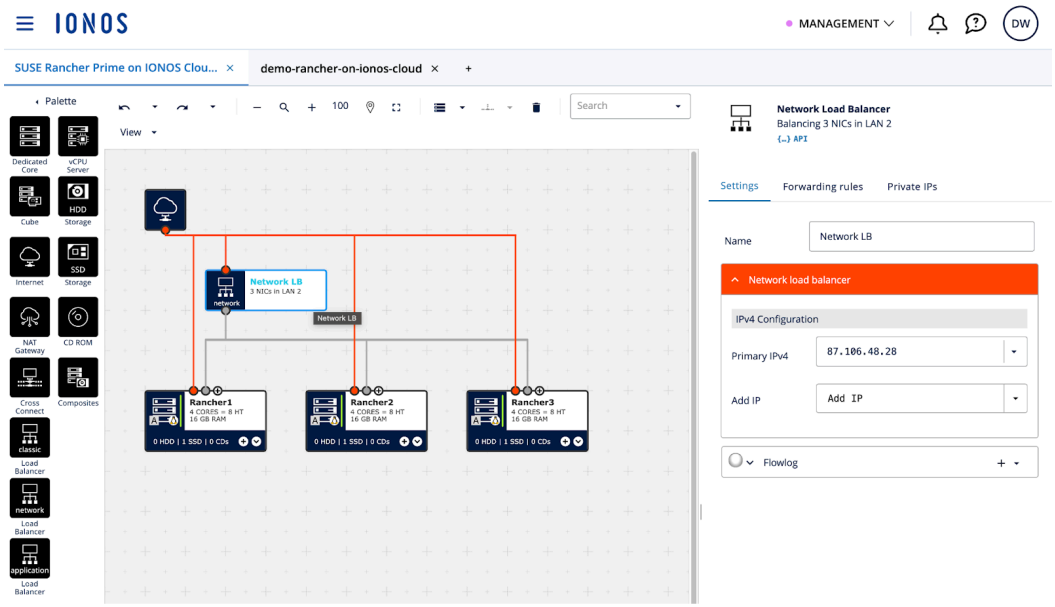

Figure 10: A new Network Load Balancer with a public and a private IP address, connected to the existing LANs.

Forwarding rules for TCP ports 80 and 443 handle the Rancher UI traffic. Additional rules for TCP port 9345 allow new downstream nodes to register with the RKE2 cluster, and port 6443 serves the Kubernetes API. Further reading: [RKE2: Configuring the fixed registration address]

Figure 11 shows an example of such a forwarding rule based on HTTP.

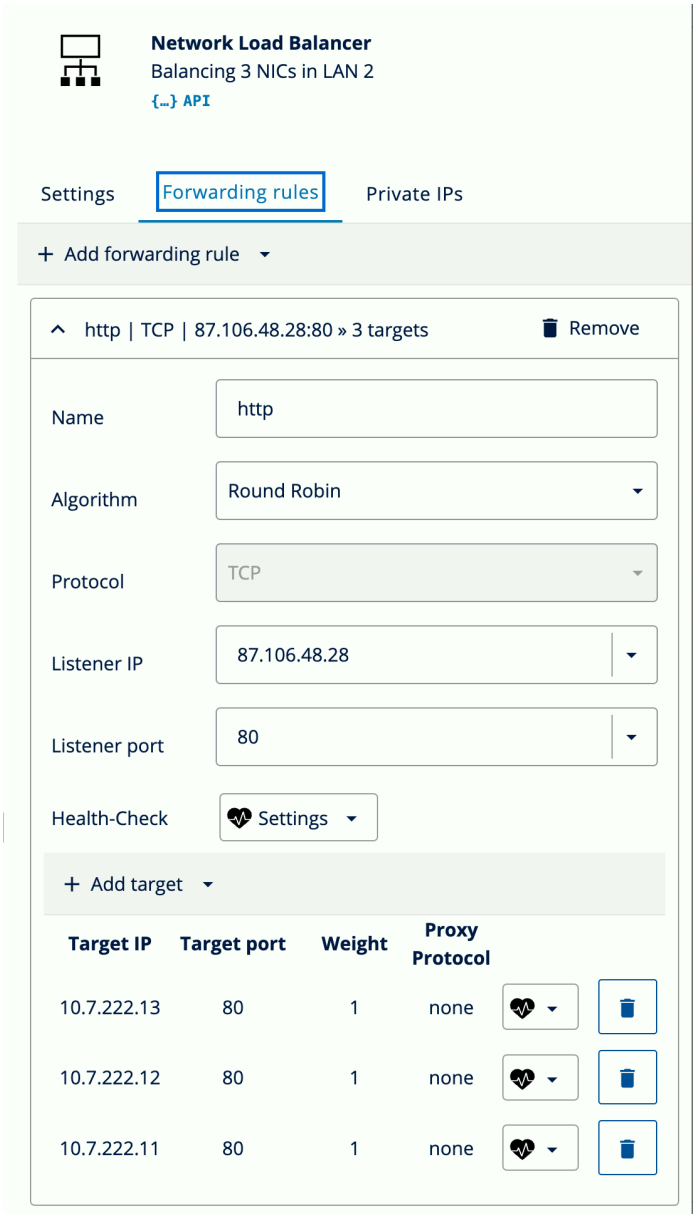

Figure 11: Forwarding rule of the Network Load Balancer for HTTP traffic to the three virtual machines of the Rancher Manager

Four rules for the different ports are added, with all rules referencing the three virtual machines for the Rancher Manager (Figure 12).

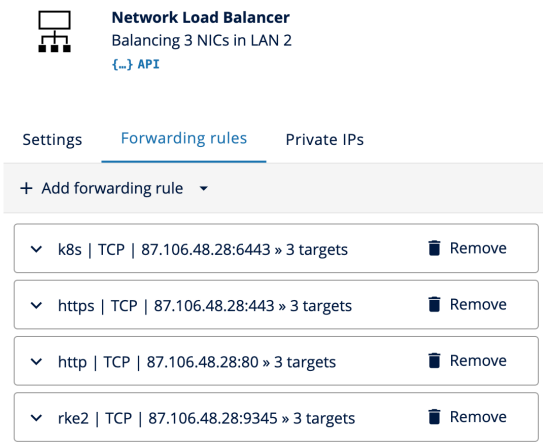

Figure 12: Network Load Balancer configuration with four forwarding rules for HTTP (80), HTTPS (443), RKE2 (9345), and K8s (6443) completed.

In the DCD, "Provision Changes" is selected and the completion of the provisioning is awaited (Figure 13).

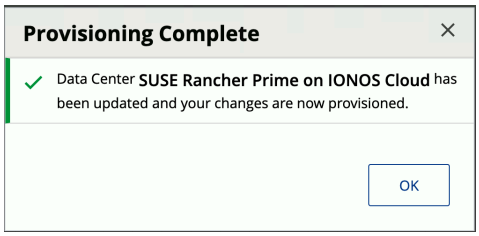

Figure 13: Confirmation that the infrastructure deployment in the data center "SUSE Rancher Prime on IONOS CLOUD" was successfully completed.

Once the infrastructure is ready, access to the nodes is via SSH to configure the software. The operating system must be registered with the SUSE Customer Center via SUSEConnect.

The following commands are executed to register the virtual machine with SCC, install iptables, and apply updates:

```
SUSEConnect -r <REGISTRATIONCODE> -e <EMAIL>

zypper ref && zypper --non-interactive in iptables

zypper --non-interactive dup
```

The file /etc/rancher/rke2/config.yaml is created on all virtual machines as a privileged user. The value for "token" must be identical on every node, as additional nodes use this value to join the cluster. The "server" parameter is only set on the second and third nodes.

```
token: <SECRETCLUSTERTOKEN>

server: https://<LoadBalancerPublicIP>:9345

node-external-ip:  <VMPublicIP>

node-ip:          <VMPrivateIP>

advertise-address: <VMPrivateIP>


tls-san:

 - <LoadBalancerPublicIP>

 - <VMPublicIP>
```

Note: If DNS names are configured that point to the Load Balancer and/or virtual machines, these will also be added to the "tls-san" list. Further information: RKE2: Installation – High Availability.

The installation command is executed on the first node to start the RKE2 cluster. The version "v1.34.2+rke2r1" can be adjusted according to the requirements and current needs.

```
curl -sfL https://get.rke2.io --output install.sh

chmod +x install.sh

INSTALL_RKE2_VERSION="v1.34.2+rke2r1" ./install.sh
```

Important: If no custom "token" was specified in the RKE2 configuration, the node token must be retrieved from the original server and configured on the additional nodes before executing the RKE2 installation script. This token is located at "/var/lib/rancher/rke2/server/node-token".

The "join" command, including the cluster token, is executed on the remaining two nodes to complete the HA setup.

```
# Ensure parameters in “/etc/rancher/rke2/config.yaml” are set correctly

# Especially that "server" is configured before installation and cluster join

curl -sfL https://get.rke2.io --output install.sh

chmod +x install.sh

INSTALL_RKE2_VERSION="v1.34.2+rke2r1" ./install.sh
```

The command syntax and details for execution can be found in the official documentation. Further information: RKE2: Launching Additional Server Nodes.

The final step is the installation of Rancher Manager. This is done from a cluster server or a local machine. Prerequisites for this are kubectl, helm, and a valid kubeconfig.

The Helm CLI is used to deploy Cert-Manager and then the Rancher Chart.

For SUSE Rancher Prime, the special Prime Chart Repository should be used, which grants access to the supported version. Community users can utilize the standard repositories.

Let’s Encrypt, self-signed certificates, or a private Certificate Authority (CA) can be used. Further information: RKE2: Installing/Upgrading Rancher on a Kubernetes Cluster.

```
# SUSE Rancher Prime

# See:
https://documentation.suse.com/cloudnative/rancher-manager/v2.12/en/installation-and-upgrade/install-rancher.html#_1_add_the_helm_chart_repository

helm repo add rancher-prime <AddHelmChartRepoURL>

# Rancher community

helm repo add rancher-stable https://releases.rancher.com/server-charts/stable

# Create namespace for Rancher

kubectl create namespace cattle-system

# Add the Jetstack Helm repository
```

```
helm repo add jetstack https://charts.jetstack.io

# Update local Helm chart repository cache

helm repo update

# Install the cert-manager Helm chart

helm install cert-manager jetstack/cert-manager \

 --namespace cert-manager \

 --create-namespace \

 --set crds.enabled=true

# Once cert-manager is installed, verify it is deployed correctly by checking the cert-manager namespace for running pods

kubectl get pods --namespace cert-manager

# Rancher Generated Certificates (Default)

helm install rancher rancher-<CHART_REPO>/rancher \

 --namespace cattle-system \

 --set hostname=<LoadBalancerPublicIPOrDNSName> \

 --set bootstrapPassword=<BootstrapPassword>

# Wait for Rancher to be rolled out

kubectl -n cattle-system rollout status deploy/rancher
```

Rancher Manager is now operational and accessible via HTTPS using the Network Load Balancer's public IP address or DNS name.

The custom bootstrap password is entered or retrieved using the *kubectl* command in Figure 14.

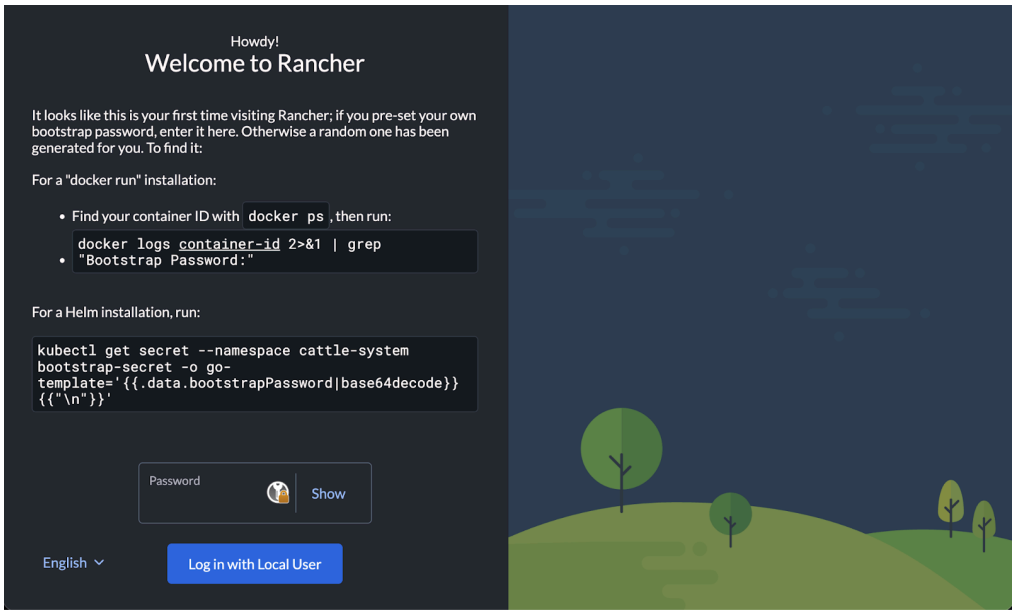

Figure 14: Initial setup screen for Rancher Manager with instructions on how to retrieve and deploy the bootstrap password.

A password for the admin user is set, the Rancher Manager server URL is confirmed or adjusted, and the EULA is accepted (Figure 15).

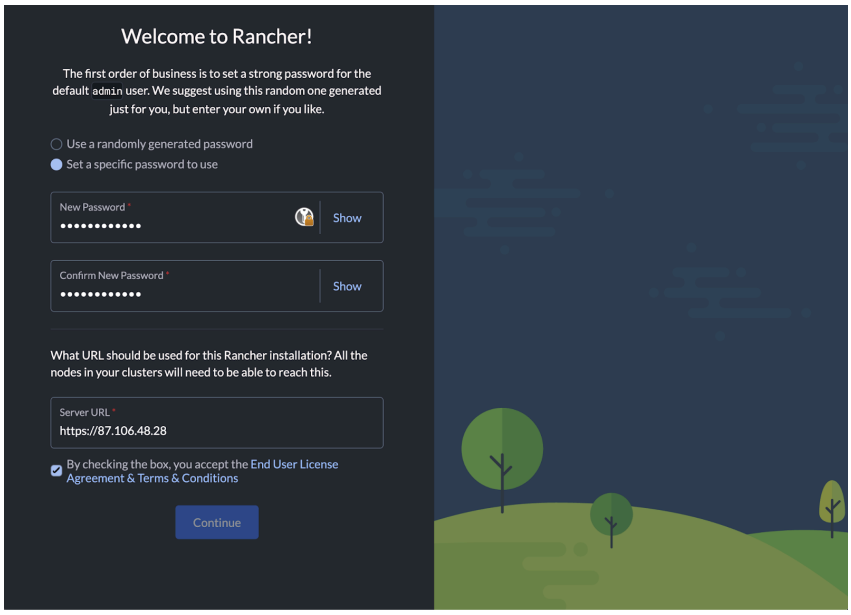

Figure 15: Rancher Manager welcome screen for entering a new password for the admin user and confirming or adjusting the server URL.

After completion, the Welcome to Rancher screen and the cluster overview will be displayed (Figure 16).

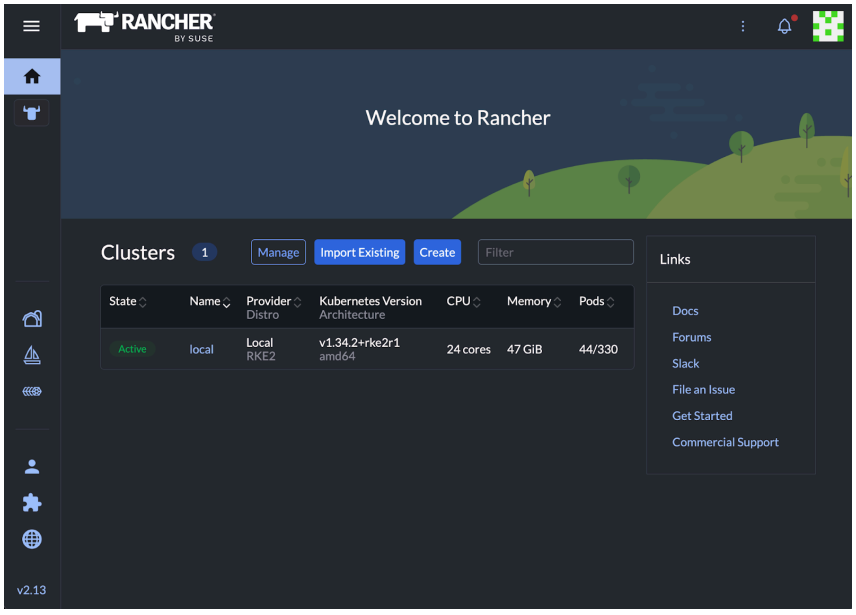

Figure 16: "Welcome to Rancher" screen and cluster overview after successful completion of initial setup

Option 2: Automated Setup

To automate the infrastructure, the provided code is used as a reference Demo: Rancher on IONOS CLOUD repository.

This code automates the setup. It defines the virtual data center and the LANs. It provisions network load balancing and virtual machines. The dual network interfaces and storage settings are also configured.

The configuration is modular. It uses the official IONOS CLOUD provider and the tf-rancher-up modules. The RKE2 module uses Cloud-init to automate the software bootstrapping. It registers IONOS CLOUD servers with SCC, downloads RKE2, and joins the cluster upon startup. The Rancher module handles the Helm chart installation. It waits until the Kubernetes API is healthy, and then deploys Cert-Manager and Rancher.

This code is used as a reference and blueprint for your own IaC (Infrastructure as Code) automation. Deployment variables are adjusted in a terraform.tfvars file. By running `terraform apply` or `tofu apply`, the infrastructure components are spun up, the forwarding rules for the Load Balancer are configured, and the software installation is started. The output provides the URL of the Rancher Manager once the process is complete.

The project's README file contains all the details on configuration and usage.

Conclusion

A highly available Rancher Manager has been deployed in the IONOS CLOUD. This article showed the manual setup to explain the architecture. Likewise, the automated Infrastructure-as-Code approach was covered.

This environment provides a solid foundation for Kubernetes management. It combines open-source flexibility with a European provider to support data sovereignty goals. This provides a platform that prioritizes local data control.

The management cluster is running. Now workloads can be integrated by registering existing Kubernetes clusters or starting new Kubernetes clusters on existing custom nodes.

More information on using the IONOS CLOUD Cluster API (CAPI) Provider will be published shortly. Technical sessions about Rancher on IONOS CLOUD are planned for the future.