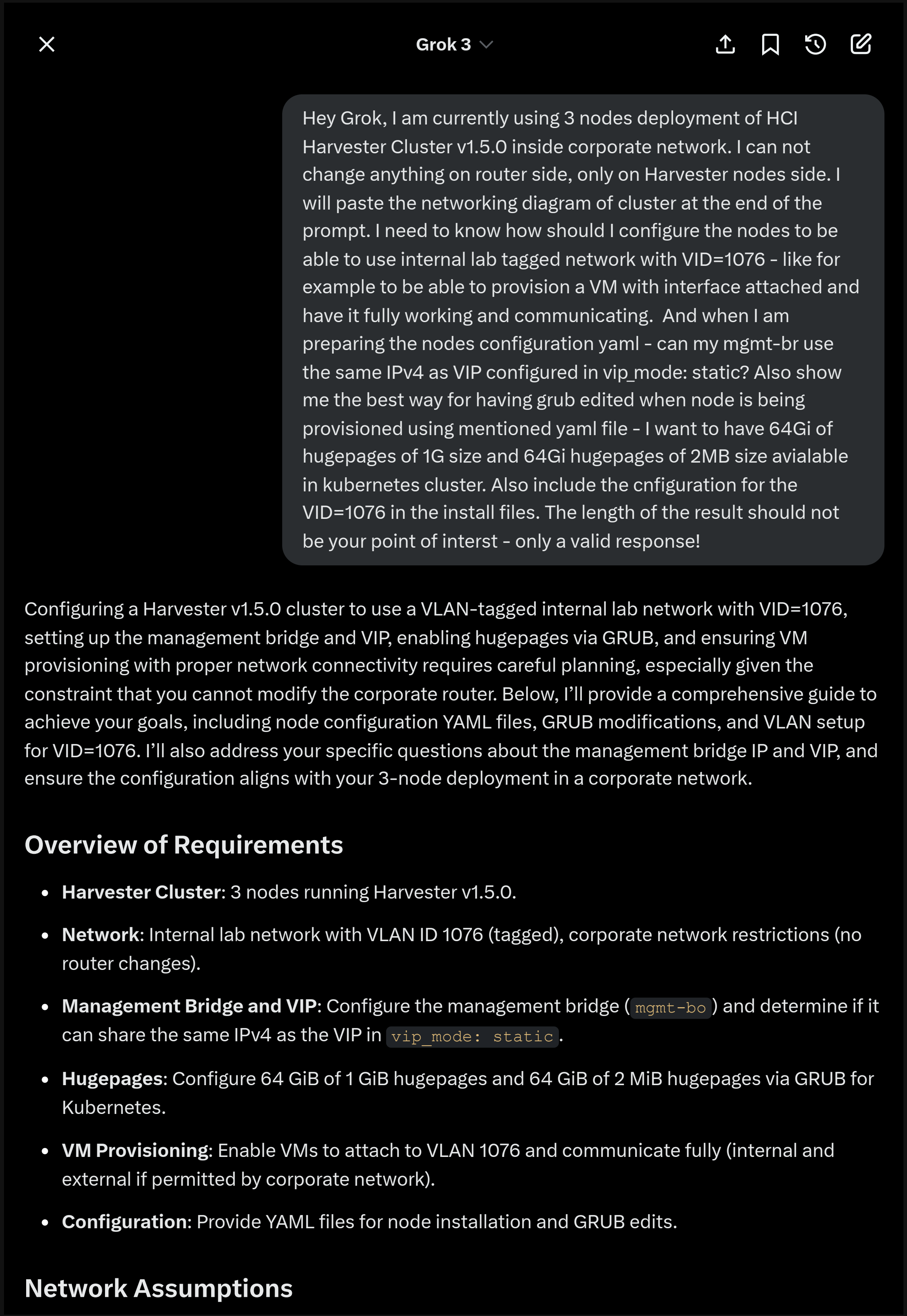


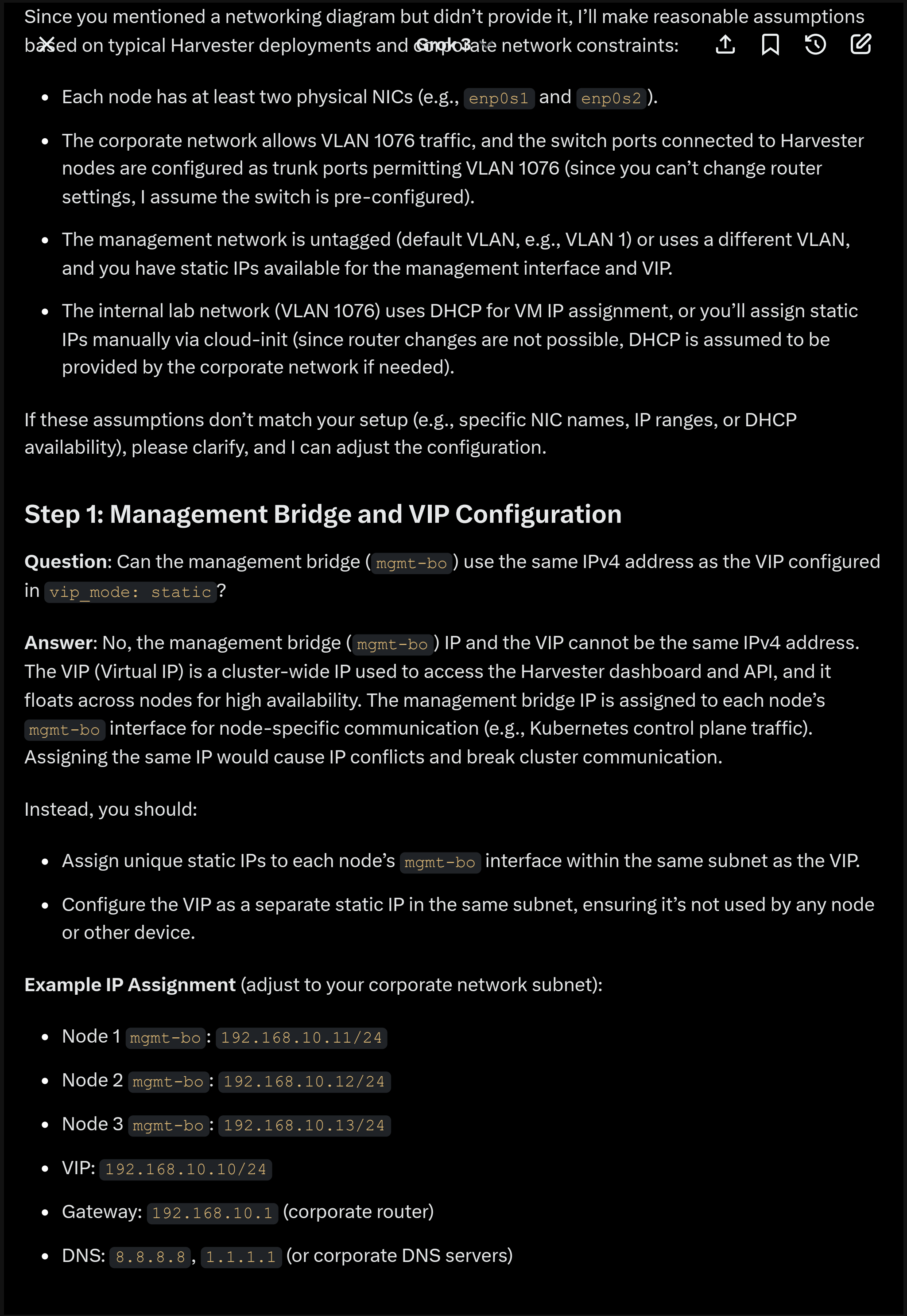
Hey Grok, I am currently using 3 nodes deployment of HCI Harvester Cluster v1.5.0 inside corporate network. I can not change anything on router side, only on Harvester nodes side. I will paste the networking diagram of cluster at the end of the prompt. I need to know how should I configure the nodes to be able to use internal lab tagged network with VID=1076 - like for example to be able to provision a VM with interface attached and have it fully working and communicating. And when I am preparing the nodes configuration yaml - can my mgmt-br use the same IPv4 as VIP configured in vip_mode: static? Also show me the best way for having grub edited when node is being provisioned using mentioned yaml file - I want to have 64Gi of hugepages of 1G size and 64Gi hugepages of 2MB size avialable in kubernetes cluster. Also include the cnfiguration for the VID=1076 in the install files. The length of the result should not be your point of interst - only a valid response!

Configuring a Harvester v1.5.0 cluster to use a VLAN-tagged internal lab network with VID=1076, setting up the management bridge and VIP, enabling hugepages via GRUB, and ensuring VM provisioning with proper network connectivity requires careful planning, especially given the constraint that you cannot modify the corporate router. Below, I'll provide a comprehensive guide to achieve your goals, including node configuration YAML files, GRUB modifications, and VLAN setup for VID=1076. I'll also address your specific questions about the management bridge IP and VIP, and ensure the configuration aligns with your 3-node deployment in a corporate network.

Overview of Requirements

- **Harvester Cluster**: 3 nodes running Harvester v1.5.0.
- **Network**: Internal lab network with VLAN ID 1076 (tagged), corporate network restrictions (no router changes).
- **Management Bridge and VIP**: Configure the management bridge (`mgmt-bo`) and determine if it can share the same IPv4 as the VIP in `vip_mode: static`.
- **Hugepages**: Configure 64 GiB of 1 GiB hugepages and 64 GiB of 2 MiB hugepages via GRUB for Kubernetes.
- **VM Provisioning**: Enable VMs to attach to VLAN 1076 and communicate fully (internal and external if permitted by corporate network).
- **Configuration**: Provide YAML files for node installation and GRUB edits.

Network Assumptions



Since you mentioned a networking diagram but didn't provide it, I'll make reasonable assumptions based on typical Harvester deployments and corporate network constraints:

- Each node has at least two physical NICs (e.g., `enp0s1` and `enp0s2`).
- The corporate network allows VLAN 1076 traffic, and the switch ports connected to Harvester nodes are configured as trunk ports permitting VLAN 1076 (since you can't change router settings, I assume the switch is pre-configured).
- The management network is untagged (default VLAN, e.g., VLAN 1) or uses a different VLAN, and you have static IPs available for the management interface and VIP.
- The internal lab network (VLAN 1076) uses DHCP for VM IP assignment, or you'll assign static IPs manually via cloud-init (since router changes are not possible, DHCP is assumed to be provided by the corporate network if needed).

If these assumptions don't match your setup (e.g., specific NIC names, IP ranges, or DHCP availability), please clarify, and I can adjust the configuration.

Step 1: Management Bridge and VIP Configuration

Question: Can the management bridge (`mgmt-bo`) use the same IPv4 address as the VIP configured in `vip_mode: static`?

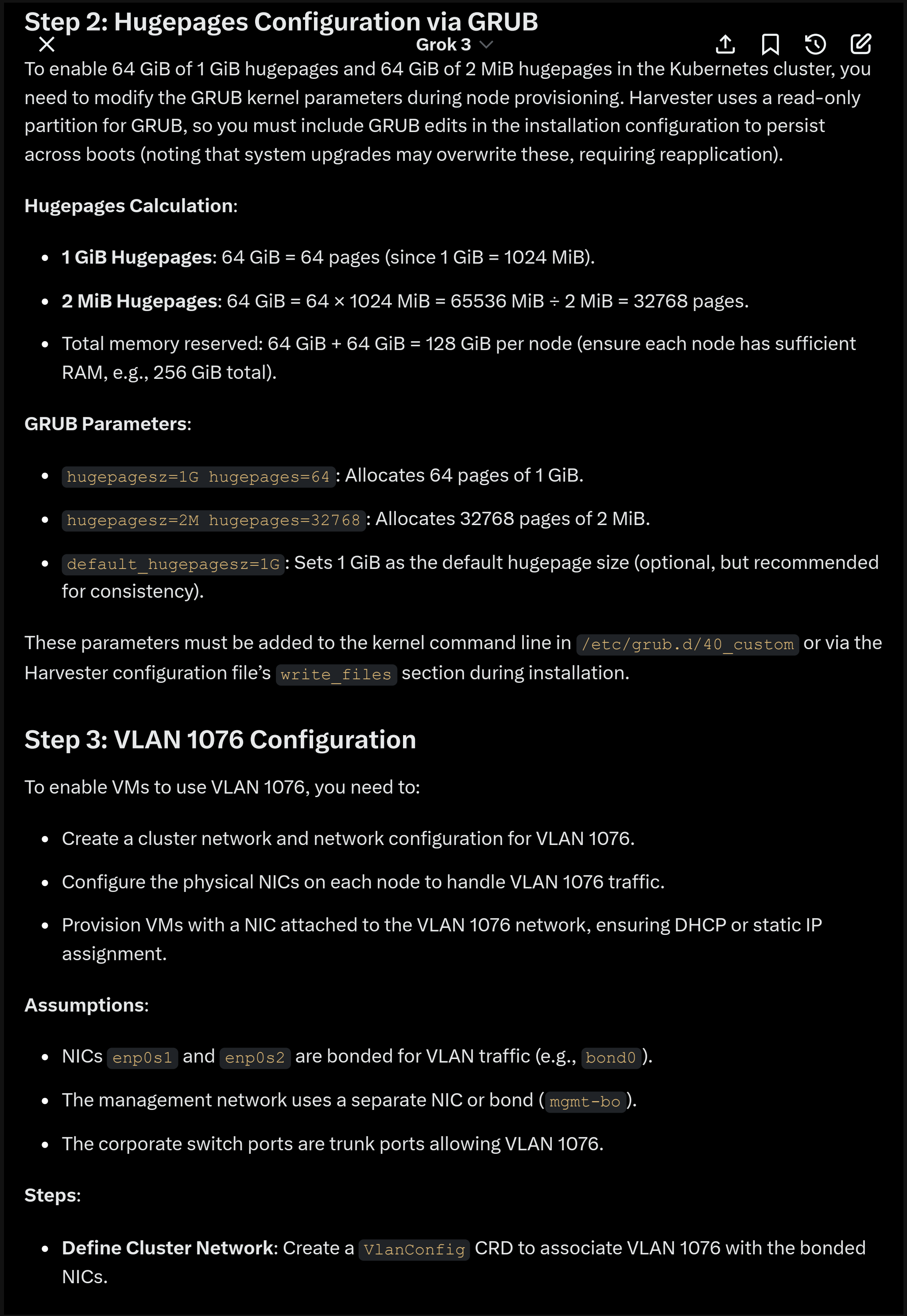
Answer: No, the management bridge (`mgmt-bo`) IP and the VIP cannot be the same IPv4 address. The VIP (Virtual IP) is a cluster-wide IP used to access the Harvester dashboard and API, and it floats across nodes for high availability. The management bridge IP is assigned to each node's `mgmt-bo` interface for node-specific communication (e.g., Kubernetes control plane traffic). Assigning the same IP would cause IP conflicts and break cluster communication.

Instead, you should:

- Assign unique static IPs to each node's `mgmt-bo` interface within the same subnet as the VIP.
- Configure the VIP as a separate static IP in the same subnet, ensuring it's not used by any node or other device.

Example IP Assignment (adjust to your corporate network subnet):

- Node 1 `mgmt-bo`: `192.168.10.11/24`
- Node 2 `mgmt-bo`: `192.168.10.12/24`
- Node 3 `mgmt-bo`: `192.168.10.13/24`
- VIP: `192.168.10.10/24`
- Gateway: `192.168.10.1` (corporate router)
- DNS: `8.8.8.8`, `1.1.1.1` (or corporate DNS servers)



Step 2: Hugepages Configuration via GRUB

To enable 64 GiB of 1 GiB hugepages and 64 GiB of 2 MiB hugepages in the Kubernetes cluster, you need to modify the GRUB kernel parameters during node provisioning. Harvester uses a read-only partition for GRUB, so you must include GRUB edits in the installation configuration to persist across boots (noting that system upgrades may overwrite these, requiring reapplication).

Hugepages Calculation:

- **1 GiB Hugepages**: 64 GiB = 64 pages (since 1 GiB = 1024 MiB).
- **2 MiB Hugepages**: 64 GiB = 64 × 1024 MiB = 65536 MiB ÷ 2 MiB = 32768 pages.
- Total memory reserved: 64 GiB + 64 GiB = 128 GiB per node (ensure each node has sufficient RAM, e.g., 256 GiB total).

GRUB Parameters:

- `hugepagesz=1G hugepages=64`: Allocates 64 pages of 1 GiB.
- `hugepagesz=2M hugepages=32768`: Allocates 32768 pages of 2 MiB.
- `default_hugepagesz=1G`: Sets 1 GiB as the default hugepage size (optional, but recommended for consistency).

These parameters must be added to the kernel command line in `/etc/grub.d/40_custom` or via the Harvester configuration file's `write_files` section during installation.

Step 3: VLAN 1076 Configuration

To enable VMs to use VLAN 1076, you need to:

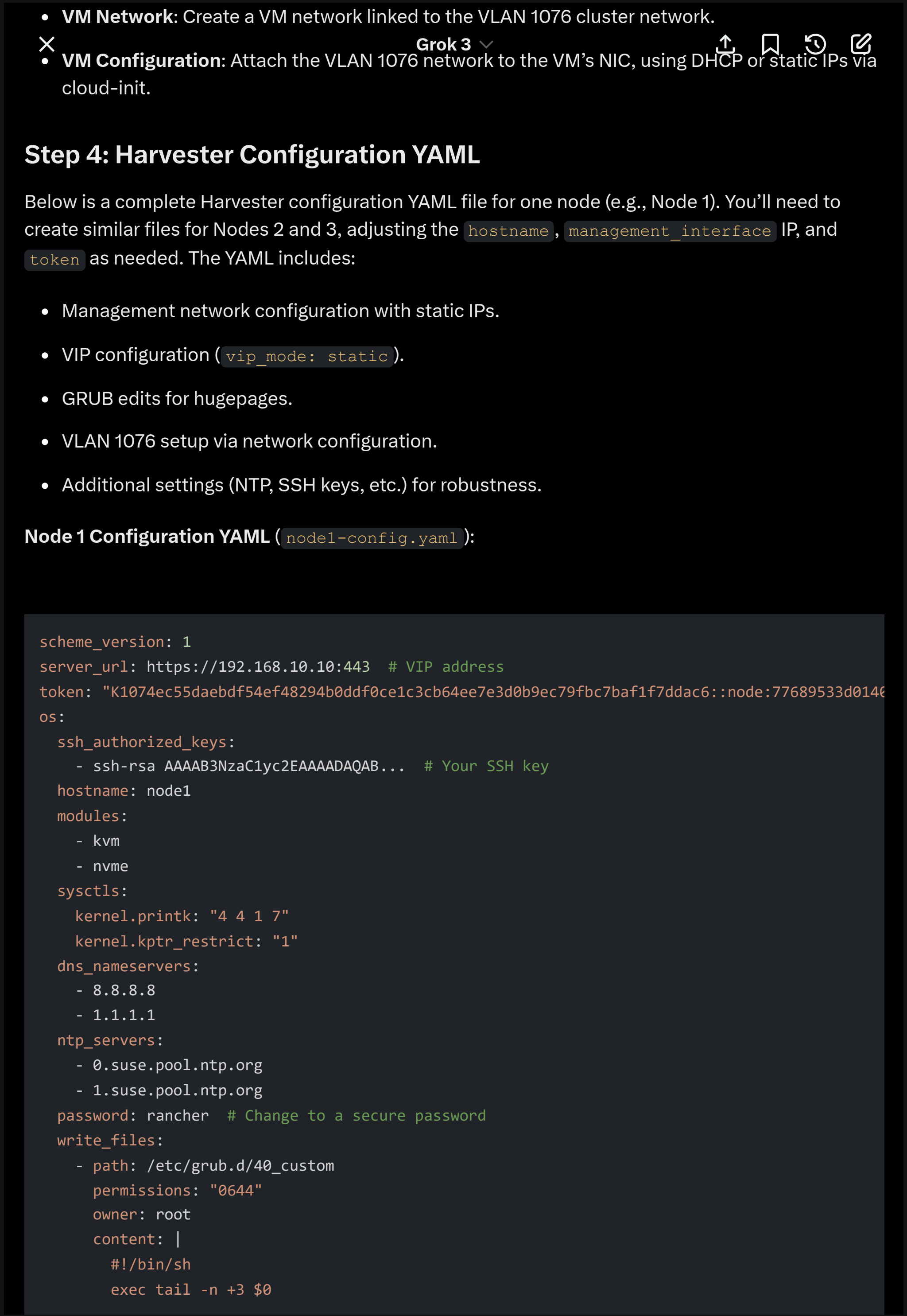
- Create a cluster network and network configuration for VLAN 1076.
- Configure the physical NICs on each node to handle VLAN 1076 traffic.
- Provision VMs with a NIC attached to the VLAN 1076 network, ensuring DHCP or static IP assignment.

Assumptions:

- NICs `enp0s1` and `enp0s2` are bonded for VLAN traffic (e.g., `bond0`).
- The management network uses a separate NIC or bond (`mgmt-bo`).
- The corporate switch ports are trunk ports allowing VLAN 1076.

Steps:

- **Define Cluster Network**: Create a `VlanConfig` CRD to associate VLAN 1076 with the bonded NICs.



- **VM Network**: Create a VM network linked to the VLAN 1076 cluster network.
- **VM Configuration**: Attach the VLAN 1076 network to the VM's NIC, using DHCP or static IPs via cloud-init.

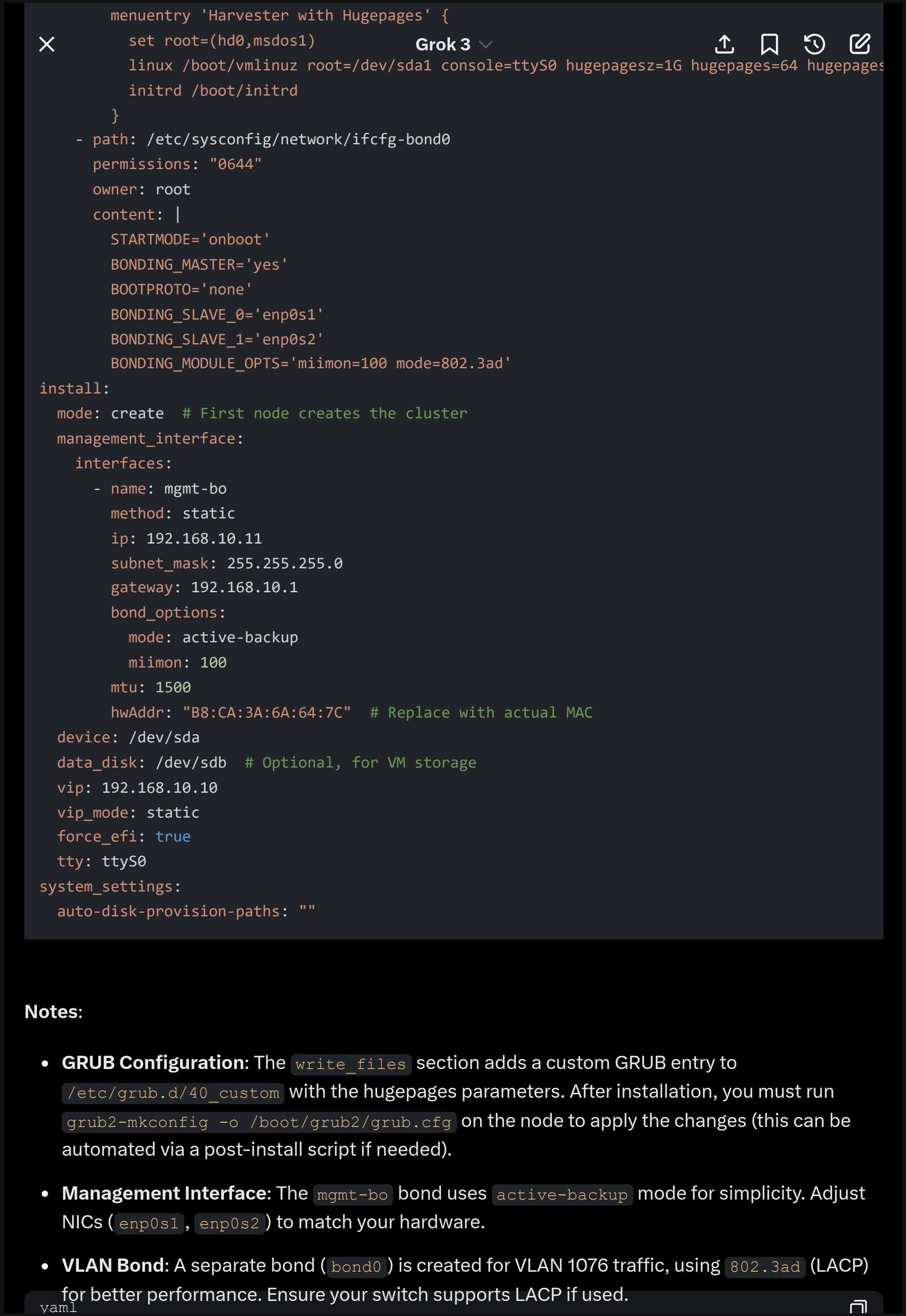
Step 4: Harvester Configuration YAML

Below is a complete Harvester configuration YAML file for one node (e.g., Node 1). You'll need to create similar files for Nodes 2 and 3, adjusting the `hostname`, `management_interface` IP, and `token` as needed. The YAML includes:

- Management network configuration with static IPs.
- VIP configuration (`vip_mode: static`).
- GRUB edits for hugepages.
- VLAN 1076 setup via network configuration.
- Additional settings (NTP, SSH keys, etc.) for robustness.

Node 1 Configuration YAML (`node1-config.yaml`):

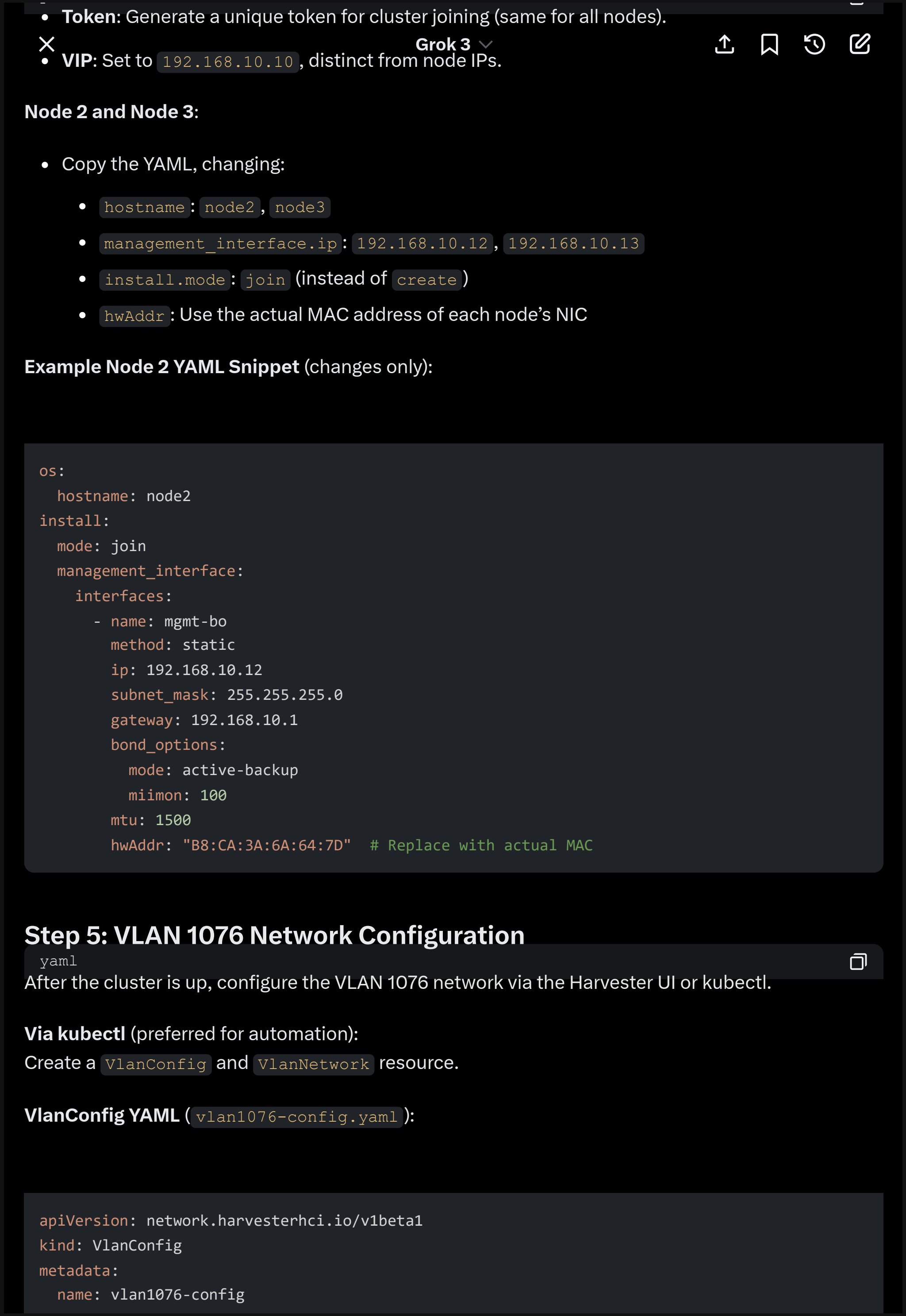
```
scheme_version: 1
server_url: https://192.168.10.10:443  # VIP address
token: "K1074ec55daebdf54ef48294b0ddf0ce1c3cb64ee7e3d0b9ec79fbc7baf1f7ddac6::node:77689533d014
os:
  ssh_authorized_keys:
    - ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB...  # Your SSH key
  hostname: node1
  modules:
    - kvm
    - nvme
  sysctls:
    kernel.printk: "4 4 1 7"
    kernel.kptr_restrict: "1"
  dns_nameservers:
    - 8.8.8.8
    - 1.1.1.1
  ntp_servers:
    - 0.suse.pool.ntp.org
    - 1.suse.pool.ntp.org
  password: rancher  # Change to a secure password
  write_files:
    - path: /etc/grub.d/40_custom
      permissions: "0644"
      owner: root
      content: |
        #!/bin/sh
        exec tail -n +3 $0
```



```
        menuentry 'Harvester with Hugepages' {
          set root=(hd0,msdos1)
          linux /boot/vmlinuz root=/dev/sda1 console=ttyS0 hugepagesz=1G hugepages=64 hugepages
          initrd /boot/initrd
        }
    - path: /etc/sysconfig/network/ifcfg-bond0
      permissions: "0644"
      owner: root
      content: |
        STARTMODE='onboot'
        BONDING_MASTER='yes'
        BOOTPROTO='none'
        BONDING_SLAVE_0='enp0s1'
        BONDING_SLAVE_1='enp0s2'
        BONDING_MODULE_OPTS='miimon=100 mode=802.3ad'
install:
  mode: create  # First node creates the cluster
  management_interface:
    interfaces:
      - name: mgmt-bo
        method: static
        ip: 192.168.10.11
        subnet_mask: 255.255.255.0
        gateway: 192.168.10.1
        bond_options:
          mode: active-backup
          miimon: 100
        mtu: 1500
        hwAddr: "B8:CA:3A:6A:64:7C"  # Replace with actual MAC
  device: /dev/sda
  data_disk: /dev/sdb  # Optional, for VM storage
  vip: 192.168.10.10
  vip_mode: static
  force_efi: true
  tty: ttyS0
system_settings:
  auto-disk-provision-paths: ""
```

Notes:

- **GRUB Configuration**: The `write_files` section adds a custom GRUB entry to `/etc/grub.d/40_custom` with the hugepages parameters. After installation, you must run `grub2-mkconfig -o /boot/grub2/grub.cfg` on the node to apply the changes (this can be automated via a post-install script if needed).
- **Management Interface**: The `mgmt-bo` bond uses `active-backup` mode for simplicity. Adjust NICs (`enp0s1`, `enp0s2`) to match your hardware.
- **VLAN Bond**: A separate bond (`bond0`) is created for VLAN 1076 traffic, using `802.3ad` (LACP) for better performance. Ensure your switch supports LACP if used.



- **Token**: Generate a unique token for cluster joining (same for all nodes).
- **VIP**: Set to `192.168.10.10`, distinct from node IPs.

Node 2 and Node 3:

- Copy the YAML, changing:
  - `hostname`: `node2`, `node3`
  - `management_interface.ip`: `192.168.10.12`, `192.168.10.13`
  - `install.mode`: `join` (instead of `create`)
  - `hwAddr`: Use the actual MAC address of each node's NIC

Example Node 2 YAML Snippet (changes only):

```
os:
  hostname: node2
install:
  mode: join
  management_interface:
    interfaces:
      - name: mgmt-bo
        method: static
        ip: 192.168.10.12
        subnet_mask: 255.255.255.0
        gateway: 192.168.10.1
        bond_options:
          mode: active-backup
          miimon: 100
        mtu: 1500
        hwAddr: "B8:CA:3A:6A:64:7D"  # Replace with actual MAC
```

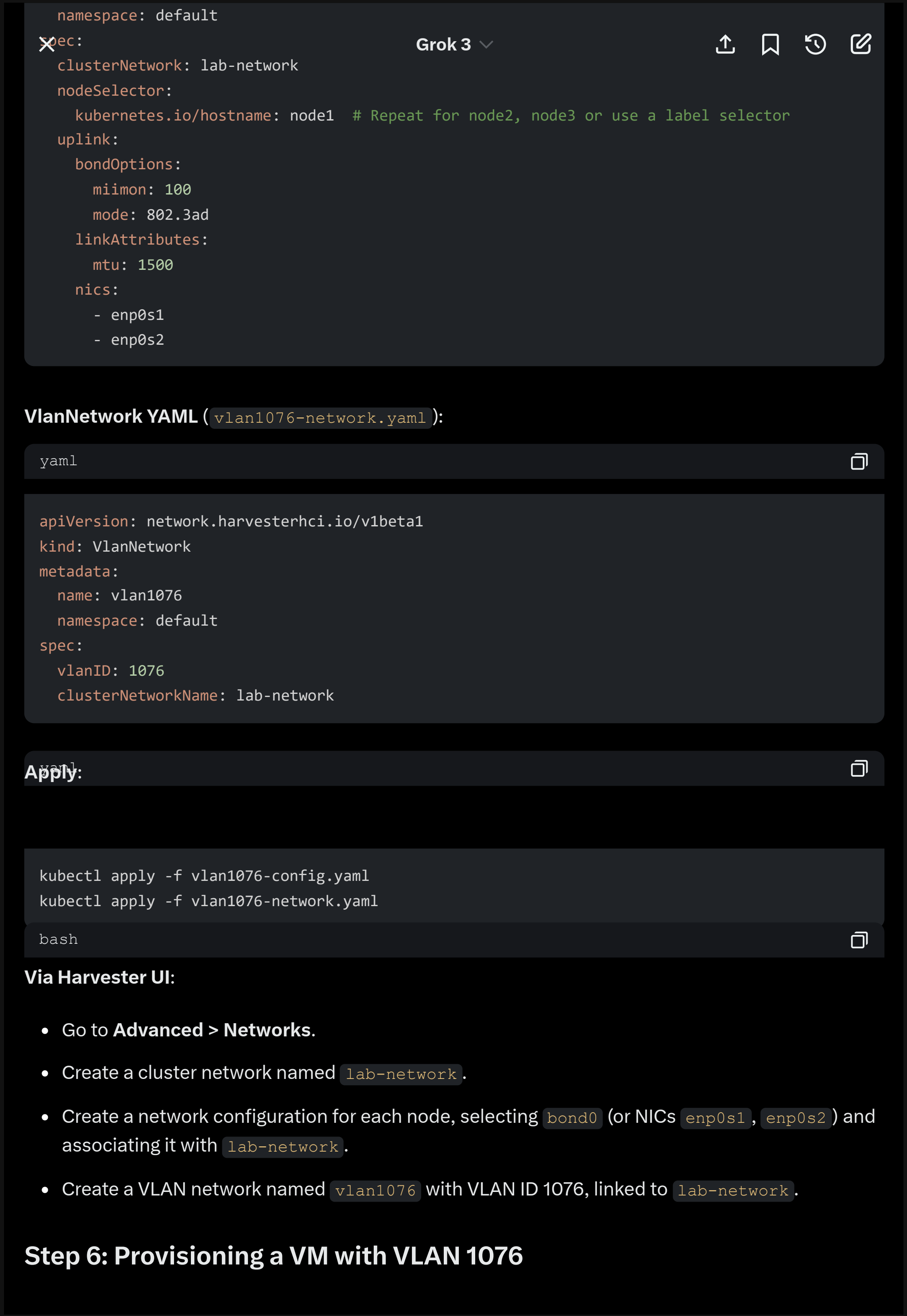
Step 5: VLAN 1076 Network Configuration

After the cluster is up, configure the VLAN 1076 network via the Harvester UI or kubectl.

Via kubectl (preferred for automation):
Create a `VlanConfig` and `VlanNetwork` resource.

VlanConfig YAML (`vlan1076-config.yaml`):

```
apiVersion: network.harvesterhci.io/v1beta1
kind: VlanConfig
metadata:
  name: vlan1076-config
```



```
  namespace: default
spec:
  clusterNetwork: lab-network
  nodeSelector:
    kubernetes.io/hostname: node1  # Repeat for node2, node3 or use a label selector
  uplink:
    bondOptions:
      miimon: 100
      mode: 802.3ad
    linkAttributes:
      mtu: 1500
    nics:
      - enp0s1
      - enp0s2
```

VlanNetwork YAML (`vlan1076-network.yaml`):

```yaml
apiVersion: network.harvesterhci.io/v1beta1
kind: VlanNetwork
metadata:
  name: vlan1076
  namespace: default
spec:
  vlanID: 1076
  clusterNetworkName: lab-network
```

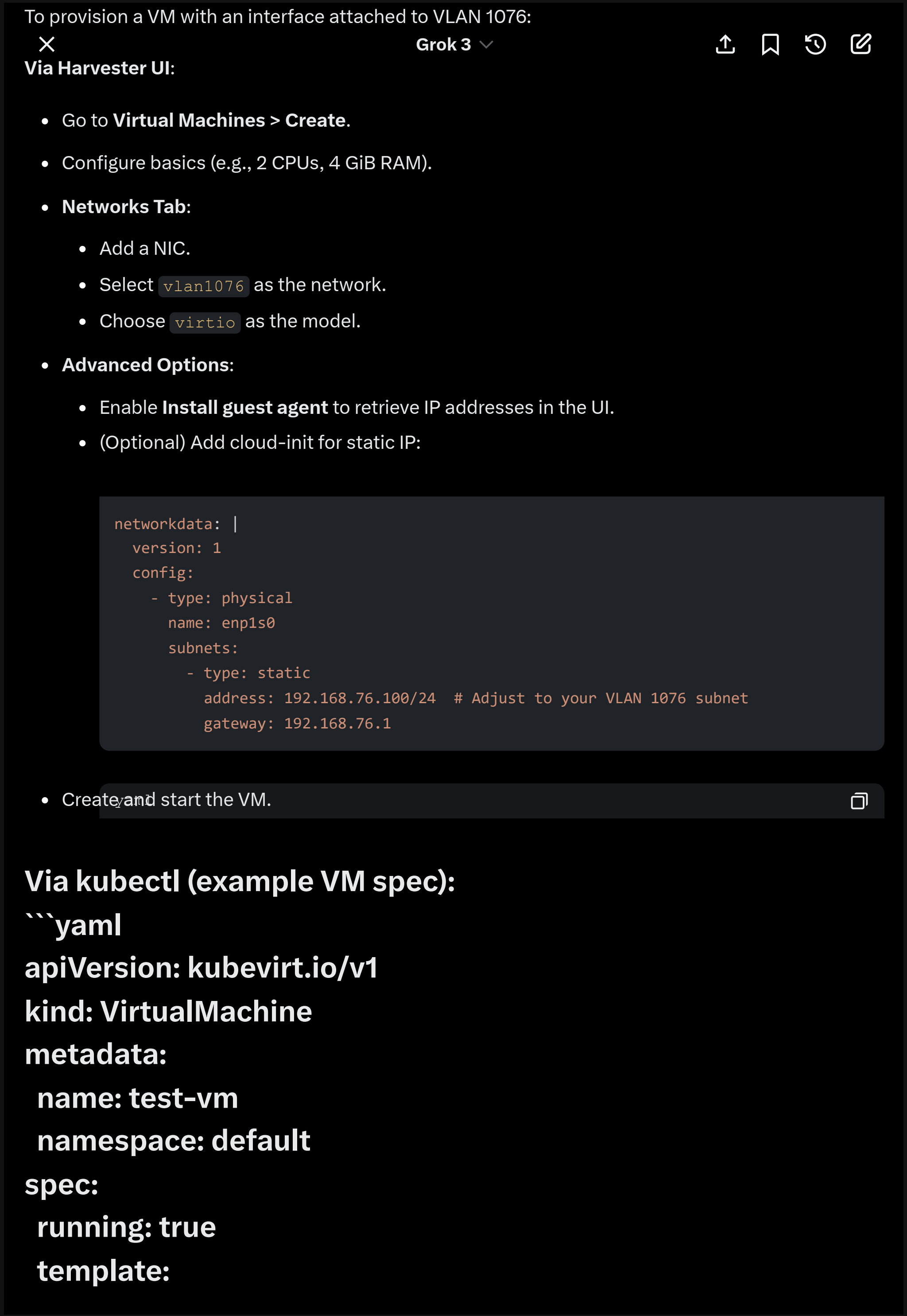
Apply:

```bash
kubectl apply -f vlan1076-config.yaml
kubectl apply -f vlan1076-network.yaml
```

Via Harvester UI:

- Go to **Advanced > Networks**.
- Create a cluster network named `lab-network`.
- Create a network configuration for each node, selecting `bond0` (or NICs `enp0s1`, `enp0s2`) and associating it with `lab-network`.
- Create a VLAN network named `vlan1076` with VLAN ID 1076, linked to `lab-network`.

Step 6: Provisioning a VM with VLAN 1076



To provision a VM with an interface attached to VLAN 1076:

Via Harvester UI:

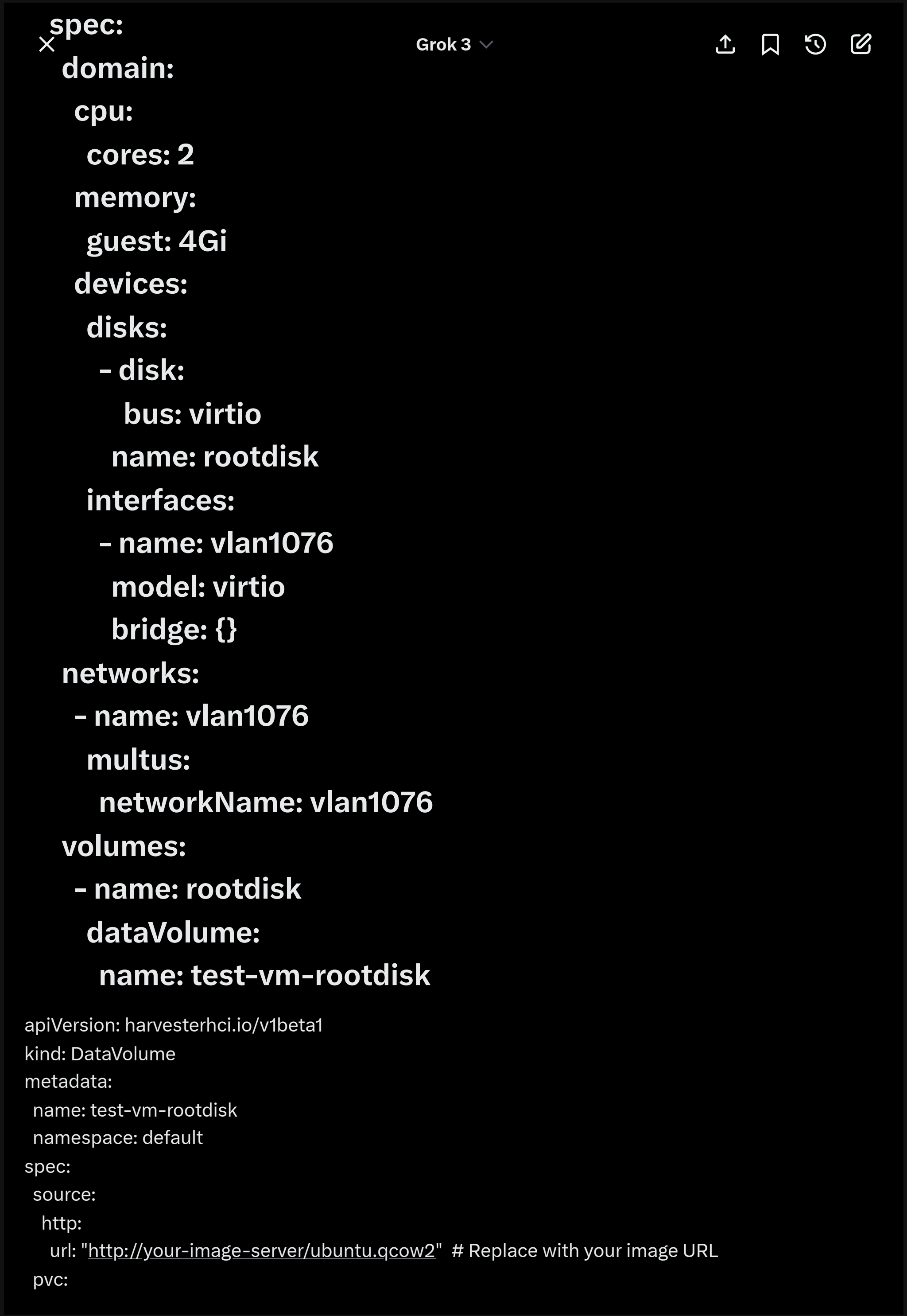
- Go to **Virtual Machines > Create**.
- Configure basics (e.g., 2 CPUs, 4 GiB RAM).
- **Networks Tab**:
  - Add a NIC.
  - Select `vlan1076` as the network.
  - Choose `virtio` as the model.
- **Advanced Options**:
  - Enable **Install guest agent** to retrieve IP addresses in the UI.
  - (Optional) Add cloud-init for static IP:

```yaml
networkdata: |
  version: 1
  config:
    - type: physical
      name: enp1s0
      subnets:
        - type: static
          address: 192.168.76.100/24  # Adjust to your VLAN 1076 subnet
          gateway: 192.168.76.1
```

- Create and start the VM.

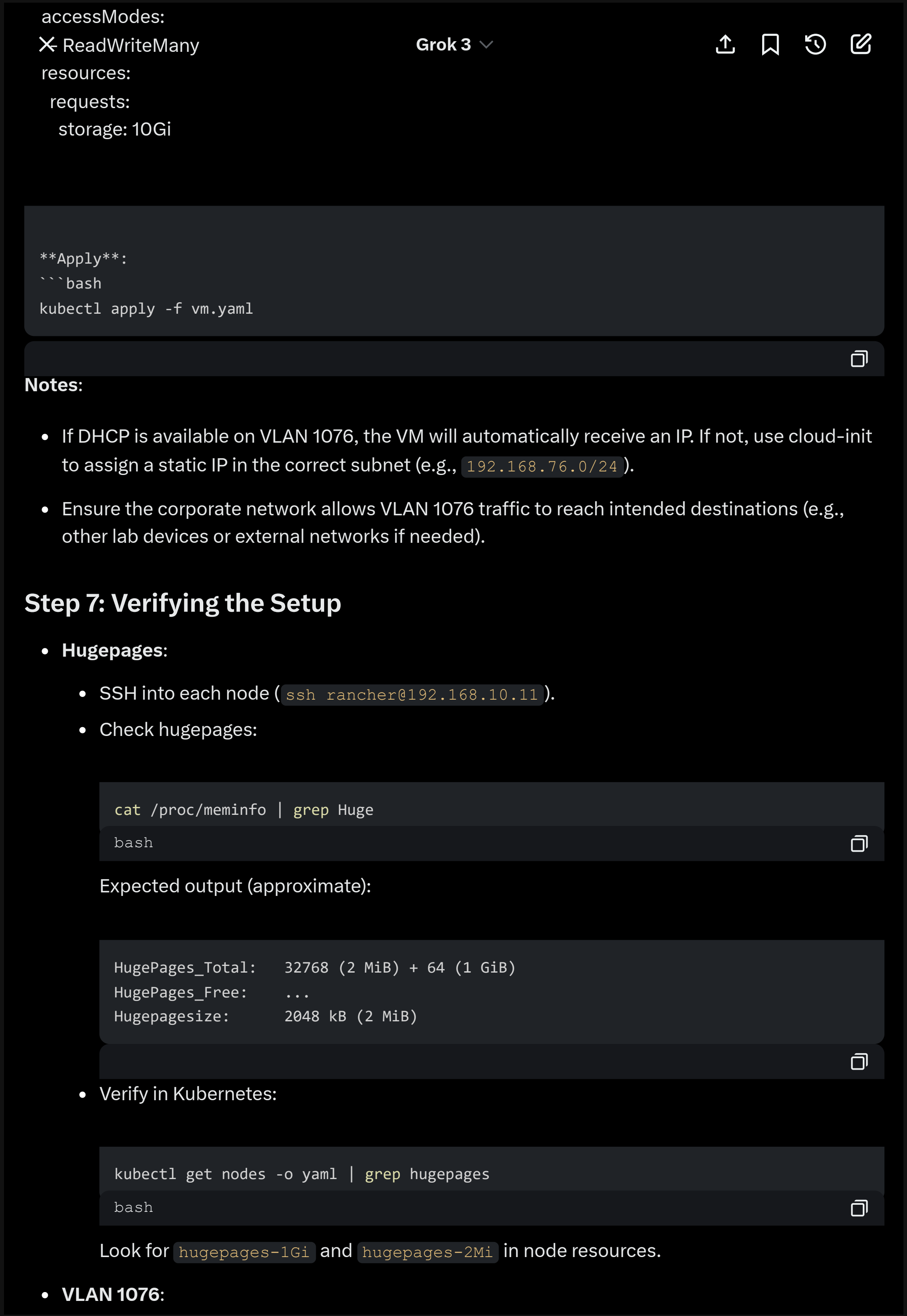
Via kubectl (example VM spec):

**```yaml**
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
name: test-vm
namespace: default
spec:
running: true
template:



```yaml
spec:
  domain:
    cpu:
      cores: 2
    memory:
      guest: 4Gi
    devices:
      disks:
        - disk:
            bus: virtio
          name: rootdisk
      interfaces:
        - name: vlan1076
          model: virtio
          bridge: {}
  networks:
    - name: vlan1076
      multus:
        networkName: vlan1076
  volumes:
    - name: rootdisk
      dataVolume:
        name: test-vm-rootdisk
```

```yaml
apiVersion: harvesterhci.io/v1beta1
kind: DataVolume
metadata:
  name: test-vm-rootdisk
  namespace: default
spec:
  source:
    http:
      url: "http://your-image-server/ubuntu.qcow2"  # Replace with your image URL
  pvc:
```



```yaml
    accessModes:
      ReadWriteMany
    resources:
      requests:
        storage: 10Gi
```

~~~
**Apply**:
```bash
kubectl apply -f vm.yaml
~~~

Notes:

- If DHCP is available on VLAN 1076, the VM will automatically receive an IP. If not, use cloud-init to assign a static IP in the correct subnet (e.g., `192.168.76.0/24`).
- Ensure the corporate network allows VLAN 1076 traffic to reach intended destinations (e.g., other lab devices or external networks if needed).

Step 7: Verifying the Setup

- **Hugepages**:
  - SSH into each node (`ssh rancher@192.168.10.11`).
  - Check hugepages:

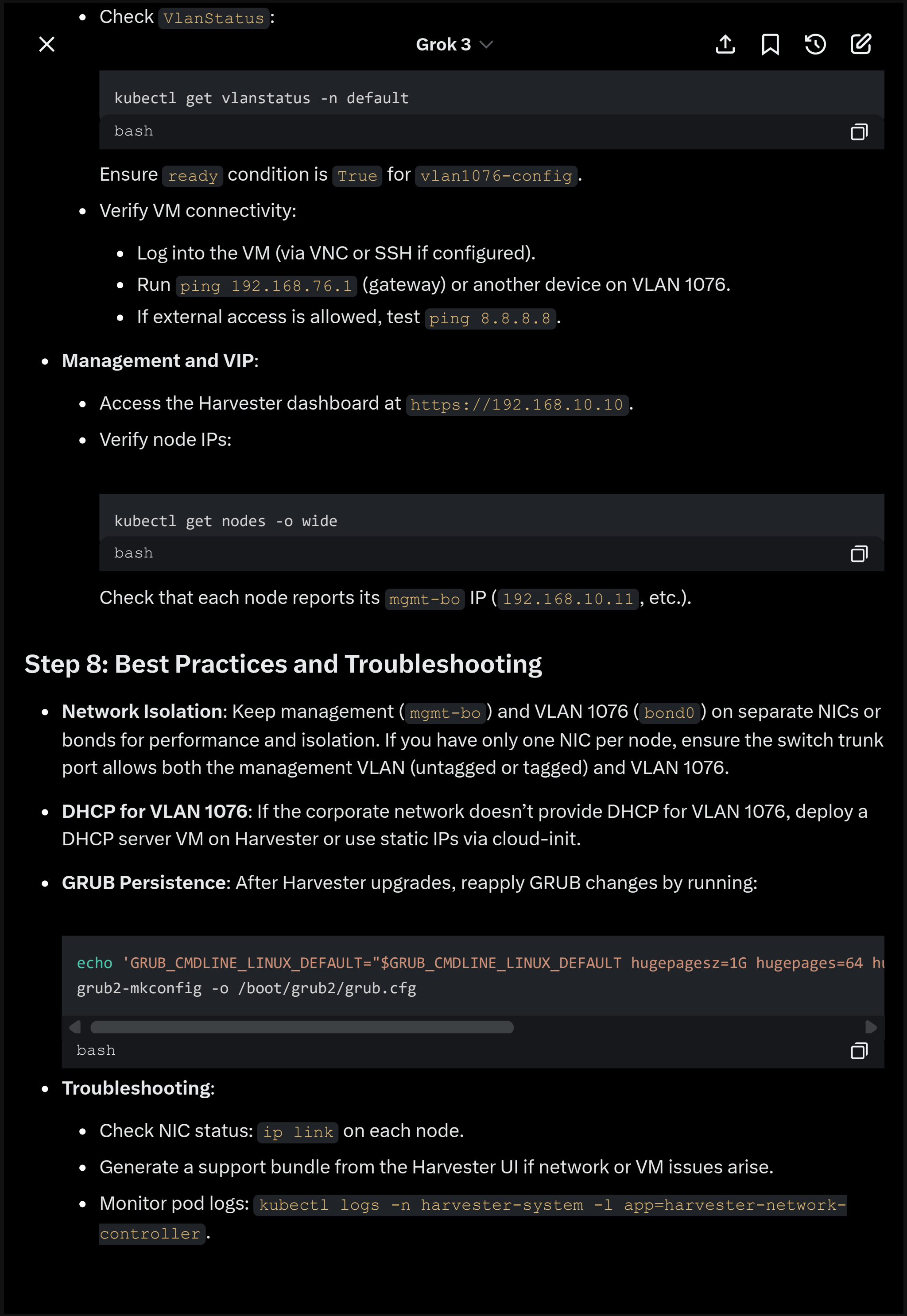
    ```bash
    cat /proc/meminfo | grep Huge
    ```

    Expected output (approximate):

    ```
    HugePages_Total:   32768 (2 MiB) + 64 (1 GiB)
    HugePages_Free:    ...
    Hugepagesize:      2048 kB (2 MiB)
    ```
  - Verify in Kubernetes:

    ```bash
    kubectl get nodes -o yaml | grep hugepages
    ```

    Look for `hugepages-1Gi` and `hugepages-2Mi` in node resources.
- **VLAN 1076**:



- Check `VlanStatus`:

```
kubectl get vlanstatus -n default
```

  Ensure `ready` condition is `True` for `vlan1076-config`.

- Verify VM connectivity:
  - Log into the VM (via VNC or SSH if configured).
  - Run `ping 192.168.76.1` (gateway) or another device on VLAN 1076.
  - If external access is allowed, test `ping 8.8.8.8`.

- **Management and VIP**:
  - Access the Harvester dashboard at `https://192.168.10.10`.
  - Verify node IPs:

```
kubectl get nodes -o wide
```

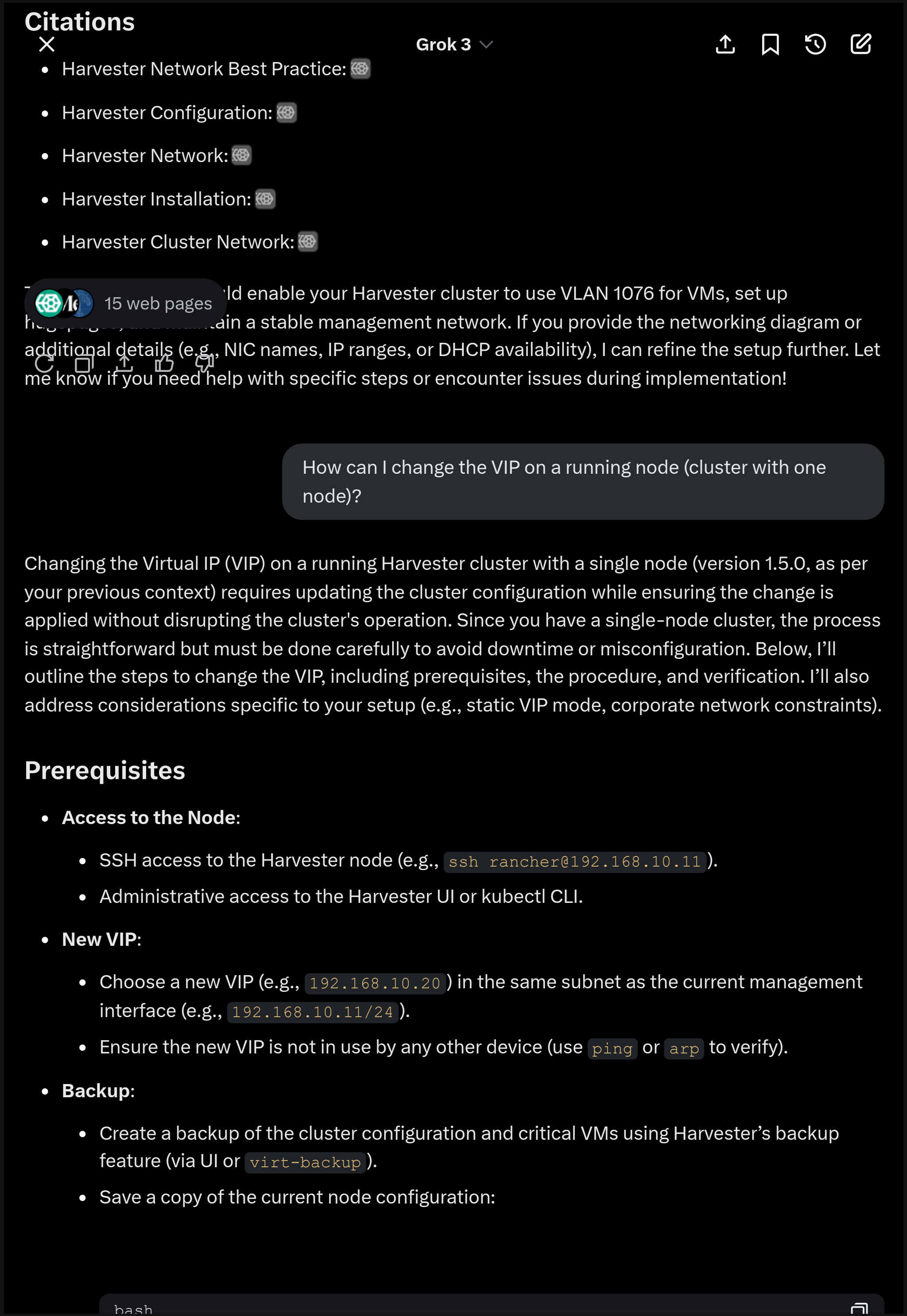
  Check that each node reports its `mgmt-bo` IP (`192.168.10.11`, etc.).

Step 8: Best Practices and Troubleshooting

- **Network Isolation**: Keep management (`mgmt-bo`) and VLAN 1076 (`bond0`) on separate NICs or bonds for performance and isolation. If you have only one NIC per node, ensure the switch trunk port allows both the management VLAN (untagged or tagged) and VLAN 1076.
- **DHCP for VLAN 1076**: If the corporate network doesn't provide DHCP for VLAN 1076, deploy a DHCP server VM on Harvester or use static IPs via cloud-init.
- **GRUB Persistence**: After Harvester upgrades, reapply GRUB changes by running:

```
echo 'GRUB_CMDLINE_LINUX_DEFAULT="$GRUB_CMDLINE_LINUX_DEFAULT hugepagesz=1G hugepages=64 hu
grub2-mkconfig -o /boot/grub2/grub.cfg
```

- **Troubleshooting**:
  - Check NIC status: `ip link` on each node.
  - Generate a support bundle from the Harvester UI if network or VM issues arise.
  - Monitor pod logs: `kubectl logs -n harvester-system -l app=harvester-network-controller`.



Citations

- Harvester Network Best Practice:
- Harvester Configuration:
- Harvester Network:
- Harvester Installation:
- Harvester Cluster Network:

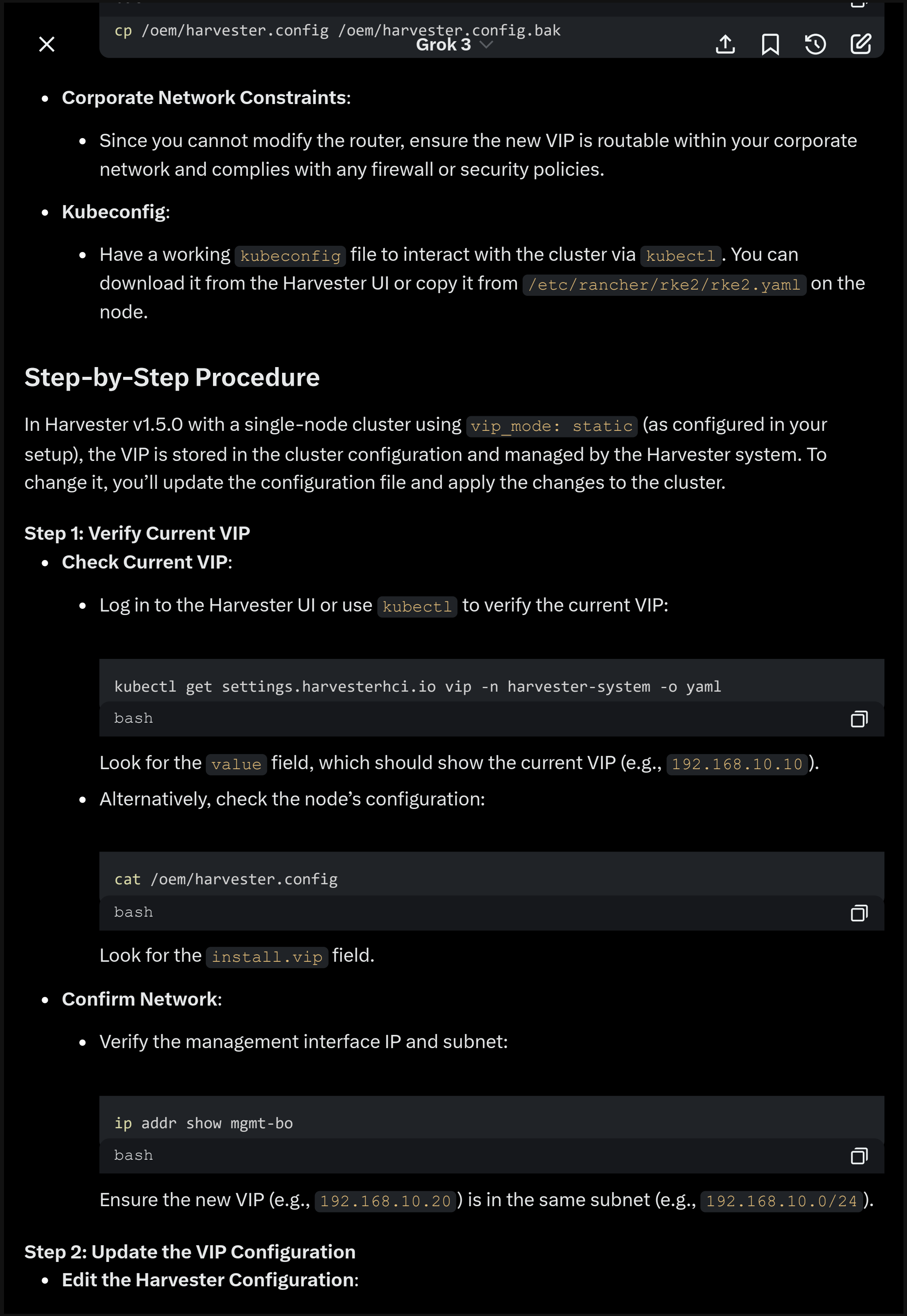
T[illegible]ld enable your Harvester cluster to use VLAN 1076 for VMs, set up h[illegible]ain a stable management network. If you provide the networking diagram or additional details (e.g., NIC names, IP ranges, or DHCP availability), I can refine the setup further. Let me know if you need help with specific steps or encounter issues during implementation!

How can I change the VIP on a running node (cluster with one node)?

Changing the Virtual IP (VIP) on a running Harvester cluster with a single node (version 1.5.0, as per your previous context) requires updating the cluster configuration while ensuring the change is applied without disrupting the cluster's operation. Since you have a single-node cluster, the process is straightforward but must be done carefully to avoid downtime or misconfiguration. Below, I'll outline the steps to change the VIP, including prerequisites, the procedure, and verification. I'll also address considerations specific to your setup (e.g., static VIP mode, corporate network constraints).

Prerequisites

- **Access to the Node**:
  - SSH access to the Harvester node (e.g., `ssh rancher@192.168.10.11`).
  - Administrative access to the Harvester UI or kubectl CLI.
- **New VIP**:
  - Choose a new VIP (e.g., `192.168.10.20`) in the same subnet as the current management interface (e.g., `192.168.10.11/24`).
  - Ensure the new VIP is not in use by any other device (use `ping` or `arp` to verify).
- **Backup**:
  - Create a backup of the cluster configuration and critical VMs using Harvester's backup feature (via UI or `virt-backup`).
  - Save a copy of the current node configuration:



```
cp /oem/harvester.config /oem/harvester.config.bak
```

- **Corporate Network Constraints**:
  - Since you cannot modify the router, ensure the new VIP is routable within your corporate network and complies with any firewall or security policies.
- **Kubeconfig**:
  - Have a working `kubeconfig` file to interact with the cluster via `kubectl`. You can download it from the Harvester UI or copy it from `/etc/rancher/rke2/rke2.yaml` on the node.

Step-by-Step Procedure

In Harvester v1.5.0 with a single-node cluster using `vip_mode: static` (as configured in your setup), the VIP is stored in the cluster configuration and managed by the Harvester system. To change it, you'll update the configuration file and apply the changes to the cluster.

Step 1: Verify Current VIP

- **Check Current VIP**:
  - Log in to the Harvester UI or use `kubectl` to verify the current VIP:

    ```bash
    kubectl get settings.harvesterhci.io vip -n harvester-system -o yaml
    ```

    Look for the `value` field, which should show the current VIP (e.g., `192.168.10.10`).
  - Alternatively, check the node's configuration:

    ```bash
    cat /oem/harvester.config
    ```

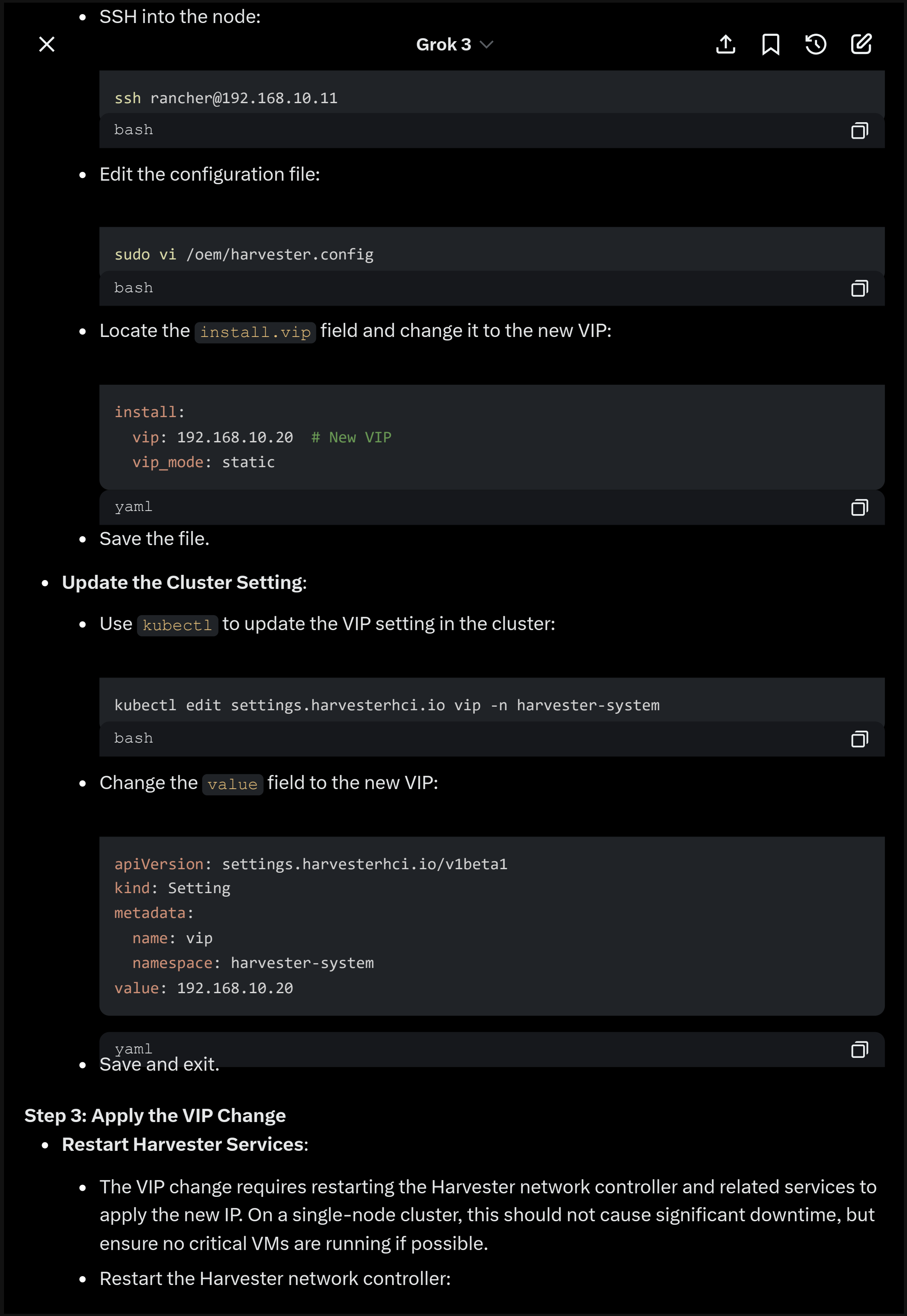
    Look for the `install.vip` field.
- **Confirm Network**:
  - Verify the management interface IP and subnet:

    ```bash
    ip addr show mgmt-bo
    ```

    Ensure the new VIP (e.g., `192.168.10.20`) is in the same subnet (e.g., `192.168.10.0/24`).

Step 2: Update the VIP Configuration

- **Edit the Harvester Configuration**:



  - SSH into the node:

    ```bash
    ssh rancher@192.168.10.11
    ```

  - Edit the configuration file:

    ```bash
    sudo vi /oem/harvester.config
    ```

  - Locate the `install.vip` field and change it to the new VIP:

    ```yaml
    install:
      vip: 192.168.10.20  # New VIP
      vip_mode: static
    ```

  - Save the file.
- **Update the Cluster Setting**:
  - Use `kubectl` to update the VIP setting in the cluster:

    ```bash
    kubectl edit settings.harvesterhci.io vip -n harvester-system
    ```

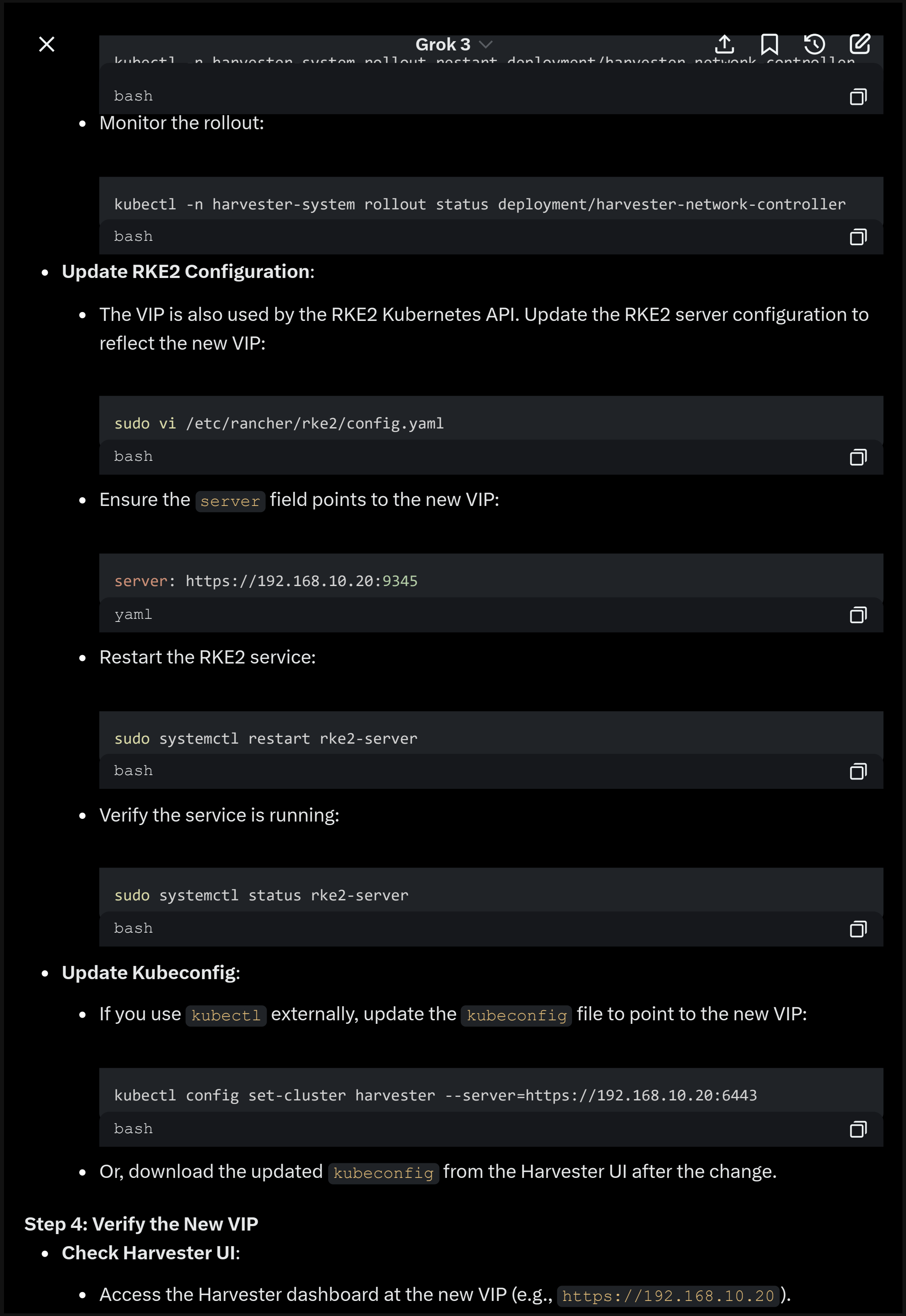
  - Change the `value` field to the new VIP:

    ```yaml
    apiVersion: settings.harvesterhci.io/v1beta1
    kind: Setting
    metadata:
      name: vip
      namespace: harvester-system
    value: 192.168.10.20
    ```

  - Save and exit.

Step 3: Apply the VIP Change

- **Restart Harvester Services**:
  - The VIP change requires restarting the Harvester network controller and related services to apply the new IP. On a single-node cluster, this should not cause significant downtime, but ensure no critical VMs are running if possible.
  - Restart the Harvester network controller:



```
kubectl -n harvester-system rollout restart deployment/harvester-network-controller
```

  - Monitor the rollout:

```
kubectl -n harvester-system rollout status deployment/harvester-network-controller
```

- **Update RKE2 Configuration**:
  - The VIP is also used by the RKE2 Kubernetes API. Update the RKE2 server configuration to reflect the new VIP:

```
sudo vi /etc/rancher/rke2/config.yaml
```

  - Ensure the `server` field points to the new VIP:

```
server: https://192.168.10.20:9345
```

  - Restart the RKE2 service:

```
sudo systemctl restart rke2-server
```

  - Verify the service is running:

```
sudo systemctl status rke2-server
```

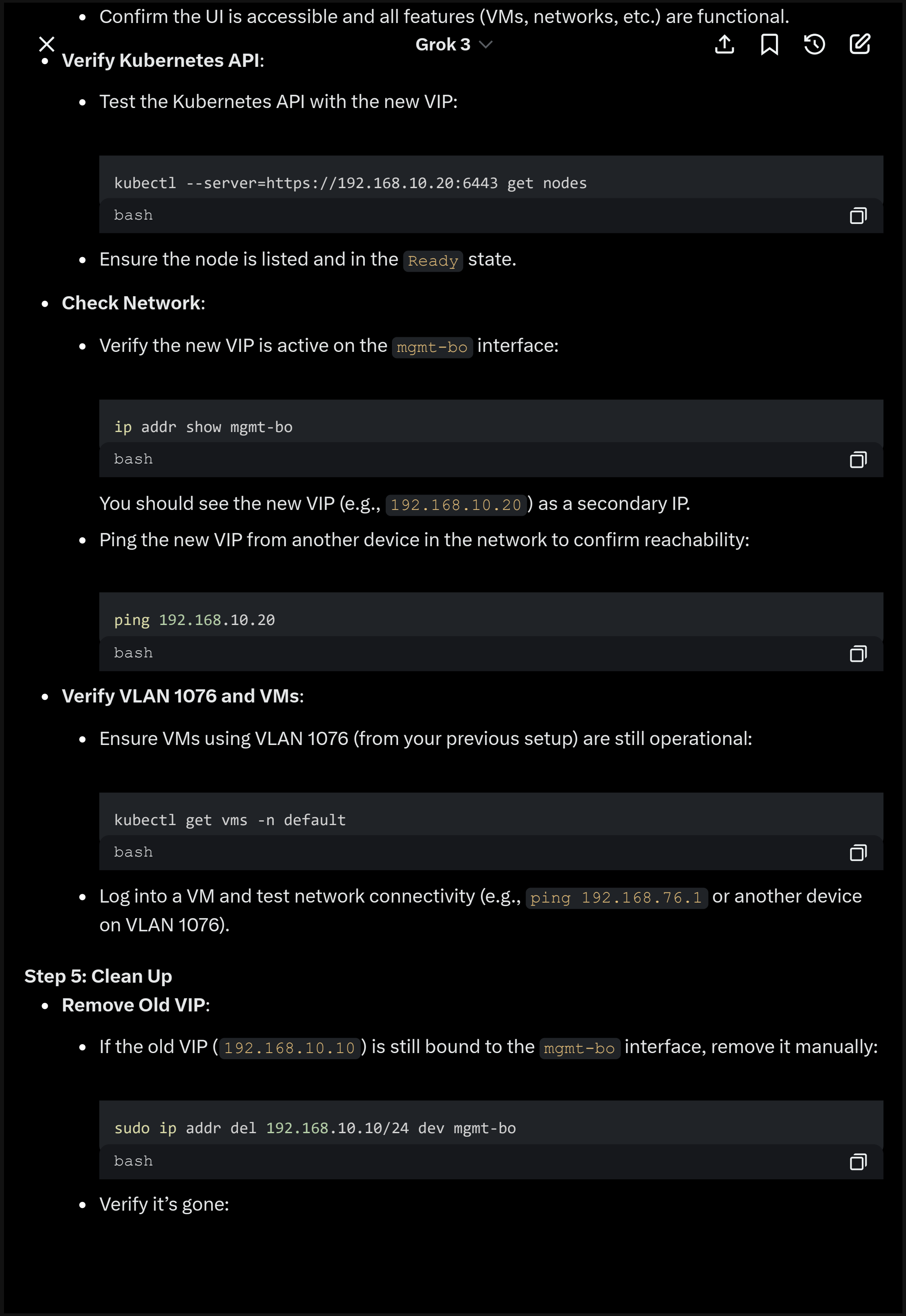
- **Update Kubeconfig**:
  - If you use `kubectl` externally, update the `kubeconfig` file to point to the new VIP:

```
kubectl config set-cluster harvester --server=https://192.168.10.20:6443
```

  - Or, download the updated `kubeconfig` from the Harvester UI after the change.

Step 4: Verify the New VIP

- **Check Harvester UI**:
  - Access the Harvester dashboard at the new VIP (e.g., `https://192.168.10.20`).



  - Confirm the UI is accessible and all features (VMs, networks, etc.) are functional.
- **Verify Kubernetes API**:
  - Test the Kubernetes API with the new VIP:

```
kubectl --server=https://192.168.10.20:6443 get nodes
```

  - Ensure the node is listed and in the `Ready` state.
- **Check Network**:
  - Verify the new VIP is active on the `mgmt-bo` interface:

```
ip addr show mgmt-bo
```

  You should see the new VIP (e.g., `192.168.10.20`) as a secondary IP.
  - Ping the new VIP from another device in the network to confirm reachability:

```
ping 192.168.10.20
```

- **Verify VLAN 1076 and VMs**:
  - Ensure VMs using VLAN 1076 (from your previous setup) are still operational:

```
kubectl get vms -n default
```

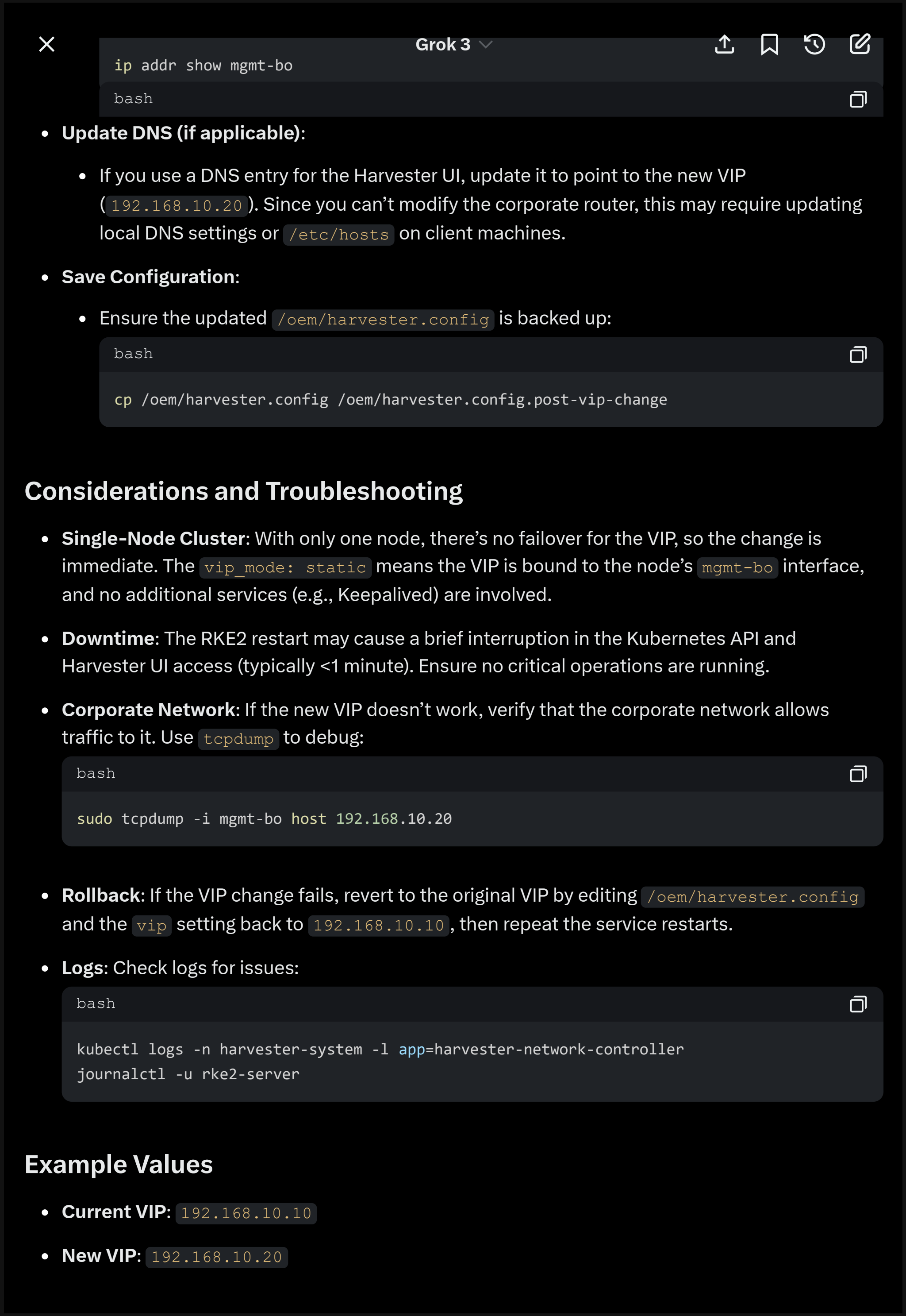
  - Log into a VM and test network connectivity (e.g., `ping 192.168.76.1` or another device on VLAN 1076).

Step 5: Clean Up

- **Remove Old VIP**:
  - If the old VIP (`192.168.10.10`) is still bound to the `mgmt-bo` interface, remove it manually:

```
sudo ip addr del 192.168.10.10/24 dev mgmt-bo
```

  - Verify it's gone:



```bash
ip addr show mgmt-bo
```

- **Update DNS (if applicable)**:
  - If you use a DNS entry for the Harvester UI, update it to point to the new VIP (`192.168.10.20`). Since you can’t modify the corporate router, this may require updating local DNS settings or `/etc/hosts` on client machines.
- **Save Configuration**:
  - Ensure the updated `/oem/harvester.config` is backed up:

```bash
cp /oem/harvester.config /oem/harvester.config.post-vip-change
```

Considerations and Troubleshooting

- **Single-Node Cluster**: With only one node, there’s no failover for the VIP, so the change is immediate. The `vip_mode: static` means the VIP is bound to the node’s `mgmt-bo` interface, and no additional services (e.g., Keepalived) are involved.
- **Downtime**: The RKE2 restart may cause a brief interruption in the Kubernetes API and Harvester UI access (typically <1 minute). Ensure no critical operations are running.
- **Corporate Network**: If the new VIP doesn’t work, verify that the corporate network allows traffic to it. Use `tcpdump` to debug:

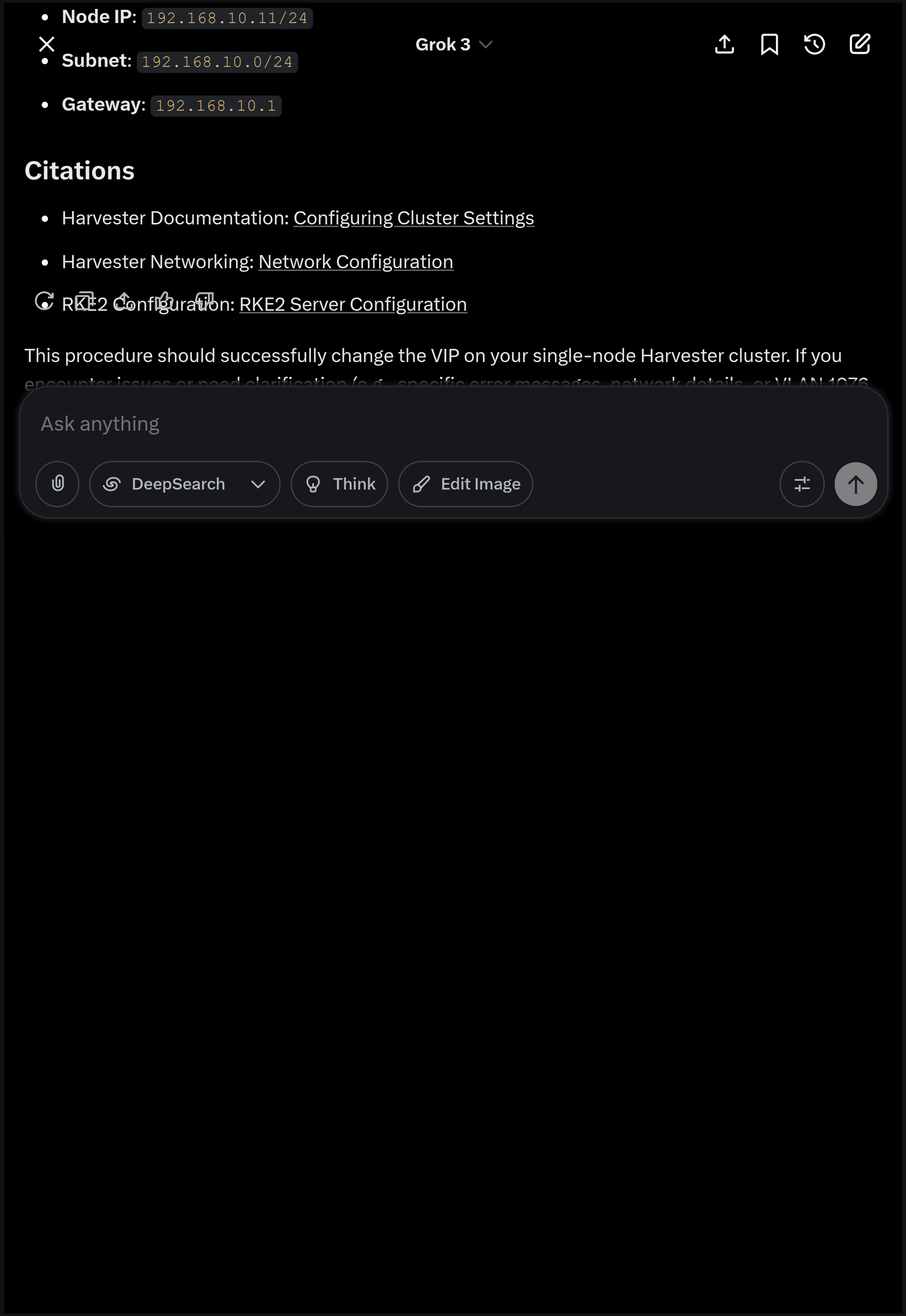
```bash
sudo tcpdump -i mgmt-bo host 192.168.10.20
```

- **Rollback**: If the VIP change fails, revert to the original VIP by editing `/oem/harvester.config` and the `vip` setting back to `192.168.10.10`, then repeat the service restarts.
- **Logs**: Check logs for issues:

```bash
kubectl logs -n harvester-system -l app=harvester-network-controller
journalctl -u rke2-server
```

Example Values

- **Current VIP**: `192.168.10.10`
- **New VIP**: `192.168.10.20`



- **Node IP**: `192.168.10.11/24`
- **Subnet**: `192.168.10.0/24`
- **Gateway**: `192.168.10.1`

Citations

- Harvester Documentation: Configuring Cluster Settings
- Harvester Networking: Network Configuration
- RKE2 Configuration: RKE2 Server Configuration

This procedure should successfully change the VIP on your single-node Harvester cluster. If you