Configuration Example

Harvester configuration file can be provided during manual or automatic installation to configure various settings. The following is a configuration example:

```yaml
scheme_version: 1
server_url: https://cluster-VIP:443
token: TOKEN_VALUE
os:
  ssh_authorized_keys:
    - ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB...
    - github:username
  write_files:
  - encoding: ""
    content: test content
    owner: root
    path: /etc/test.txt
    permissions: '0755'
  hostname: myhost
  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
  environment:
    http_proxy: http://myserver
    https_proxy: http://myserver
  labels:
    topology.kubernetes.io/zone: zone1
    foo: bar
    mylabel: myvalue
install:
  mode: create
  management_interface:
    interfaces:
    - name: ens5
      hwAddr: "B8:CA:3A:6A:64:7C"
    method: dhcp
  force_efi: true
  device: /dev/sda
  data_disk: /dev/sdb
  silent: true
  iso_url: http://myserver/test.iso
  poweroff: true
  no_format: true
  debug: true
  tty: ttyS0
  vip: 10.10.0.19
  vip_hw_addr: 52:54:00:ec:0e:0b
  vip_mode: dhcp
  cluster_pod_cidr: 10.52.0.0/16
  cluster_service_cidr: 10.53.0.0/16
  cluster_dns: 10.53.0.10
  force_mbr: false
  addons:
    harvester_vm_import_controller:
      enabled: false
      values_content: ""
    harvester_pcidevices_controller:
      enabled: false
      values_content: ""
    rancher_monitoring:
      enabled: true
      values_content: ""
    rancher_logging:
      enabled: false
      values_content: ""
    harvester_seeder:
      enabled: false
      values_content: ""
system_settings:
  auto-disk-provision-paths: ""
```

Configuration Reference

Below is a reference of all configuration keys.

:::caution **Security Risks**: The configuration file contains credentials which should be kept secret. Please do not make the configuration file publicly accessible. :::

:::note **Configuration Priority**: When you provide a remote Harvester Configuration file during the install of Harvester, the Harvester Configuration file will not overwrite the values for the inputs you had previously filled out and selected. Priority is given to the values that you input during the guided install. For instance, if you have in your Harvester Configuration file specified `os.hostname` and during install you fill in the field of `hostname` when prompted, the value that you filled in will take priority over your Harvester Configuration's `os.hostname` . :::

`scheme_version`

Definition

The version of scheme reserved for future configuration migration.

This configuration is mandatory for migrating the configuration to a new scheme version. It tells Harvester the previous version and the need to migrate.

:::note This field didn't take any effect in the current Harvester version. :::

:::caution Make sure that your custom configuration always has the correct scheme version. :::

`server_url`

Definition

`server_url` is the URL of the Harvester cluster, which is used for the new `node` to join the cluster.

This configuration is mandatory when the installation is in `JOIN` mode. The default format of `server_url` is `https://cluster-VIP:443`.

:::note

To ensure a high availability (HA) Harvester cluster, please use either the Harvester cluster VIP or a domain name in `server_url`.

:::

Example

```yaml
server_url: https://cluster-VIP:443
install:
  mode: join
```

`token`

Definition

The cluster secret or node token. If the value matches the format of a node token it will automatically be assumed to be a node token. Otherwise it is treated as a cluster secret.

In order for a new node to join the Harvester cluster, the token should match what the server has.

Example

```yaml
token: myclustersecret
```

Or a node token

```yaml
token:
"K1074ec55daebdf54ef48294b0ddf0ce1c3cb64ee7e3d0b9ec79fbc7baf1f7ddac6::node:77689533d014
```

`os.ssh_authorized_keys`

Definition

A list of SSH authorized keys that should be added to the default user, `rancher`. SSH keys can be obtained from GitHub user accounts by using the format `github:${USERNAME}`. This is done by downloading the keys from `https://github.com/${USERNAME}.keys`.

Example

```
os:
  ssh_authorized_keys:
    - "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC2TBZGjE+J8ag11dzkFT58J3XPONrDVmalCNrKxsfADfyy0eqdZrG8hc
    - "github:ibuildthecloud"
```

`os.write_files`

A list of files to write to disk on boot. The `encoding` field specifies the content's encoding. Valid `encoding` values are:

- `""` : content data are written in plain text. In this case, the `encoding` field can be also omitted.
- `b64` , `base64` : content data are base64-encoded.
- `gz` , `gzip` : content data are gzip-compressed.
- `gz+base64` , `gzip+base64` , `gz+b64` , `gzip+b64` : content data are gzip-compressed first and then base64-encoded.

Example

```
os:
  write_files:
  - encoding: b64
    content: CiMgVGhpcyBmaWxlIGNvbnRyb2xzIHRoZSBzdGF0ZSBvZiBTRUxpbnV4...
    owner: root:root
    path: /etc/connman/main.conf
    permissions: '0644'
  - content: |
      # My new /etc/sysconfig/samba file

      SMDBOPTIONS="-D"
    path: /etc/sysconfig/samba
  - content: !!binary |
      f0VMRgIBAQAAAAAAAAAAAAIAPgABAAAAwARAAAAAAABAAAAAAAAAAJAVAAAAAA
      AEAAHgAdAAYAAAAFAAAAQAAAAAAAAABAAEAAAAAAAEAAQAAAAAAAwAEAAAAAAA
      AAAAAAAAAwAAAAQAAAAAAgAAAAAAAAACQAAAAAAAAAJAAAAAAAAcAAAAAAAAAB
      ...
    path: /bin/arch
    permissions: '0555'
  - content: |
      15 * * * * root ship_logs
    path: /etc/crontab
```

`os.persistent_state_paths`

Definition

The `os.persistent_state_paths` option allows you to configure custom paths where modifications made to files will persist across reboots. Any changes to files in these paths will not be lost after a reboot.

Example

Refer to the following example config for installing `rook-ceph` in Harvester:

```yaml
os:
  persistent_state_paths:
    - /var/lib/rook
    - /var/lib/ceph
  modules:
    - rbd
    - nbd
```

`os.after_install_chroot_commands`

Definition

You can add additional software packages with `after_install_chroot_commands`. The `after-install-chroot` stage, provided by [elemental-toolkit](), allows you to execute commands not restricted by file system write issues, ensuring the persistence of user-defined commands even after a system reboot.

Example

Refer to the following example config for installing an RPM package in Harvester:

```yaml
os:
  after_install_chroot_commands:
    - rpm -ivh <the url of rpm package>
```

DNS resolution is unavailable in the `after-install-chroot stage`, and the `nameserver` might not be available. If you need to access a domain name to install a package using an URL, create a temporary `/etc/resolv.conf` file first. For example:

```yaml
os:
  after_install_chroot_commands:
    - "rm -f /etc/resolv.conf && echo 'nameserver 8.8.8.8' | sudo tee /etc/resolv.conf"
    - "mkdir /usr/local/bin"
    - "curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 && chmod 700 get_helm.sh && ./get_helm.sh"
    - "rm -f /etc/resolv.conf && ln -s /var/run/netconfig/resolv.conf /etc/resolv.conf"
```

:::note

Upgrading Harvester causes the changes to the OS in the `after-install-chroot` stage to be lost. You must also configure the `after-upgrade-chroot` to make your changes persistent across an upgrade. Refer to [Runtime persistent changes]() before upgrading Harvester.

:::

`os.hostname`

Definition

Set the system hostname. The installer will generate a random hostname if the user doesn't provide a value.

Example

```yaml
os:
  hostname: myhostname
```

`os.modules`

Definition

A list of kernel modules to be loaded on start.

Example

```yaml
os:
  modules:
    - kvm
    - nvme
```

`os.sysctls`

Definition

Kernel sysctl to set up on start. These are the typical configurations found in `/etc/sysctl.conf`. Values must be specified as strings.

Example

```yaml
os:
  sysctls:
    kernel.printk: 4 4 1 7 # the YAML parser will read as a string
    kernel.kptr_restrict: "1" # force the YAML parser to read as a string
```

`os.dns_nameservers`

Definition

Fallback DNS name servers to use if DNS is not configured by DHCP or in the OS.

Example

```yaml
os:
  dns_nameservers:
    - 8.8.8.8
    - 1.1.1.1
```

`os.ntp_servers`

Definition

Fallback ntp servers to use if NTP is not configured elsewhere in the OS. Highly recommend to configure `os.ntp_servers` to avoid time synchronization issue between machines.

Example

```yaml
os:
  ntp_servers:
    - 0.suse.pool.ntp.org
    - 1.suse.pool.ntp.org
```

`os.password`

Definition

The password for the default user, `rancher`. By default, there is no password for the `rancher` user. If you set a password at runtime it will be reset on the next boot. The value of the password can be clear text or an encrypted form. The easiest way to get this encrypted form is to change your password on a Linux system and copy the value of the second field from `/etc/shadow`. You can also encrypt a password using OpenSSL. For the encryption algorithms supported by Harvester, please refer to the table below.

Algorithm	Command	Support
SHA-512	`openssl passwd -6`	Yes
SHA-256	`openssl passwd -5`	Yes
MD5	`openssl passwd -1`	Yes
MD5, Apache variant	`openssl passwd -apr1`	Yes
AIX-MD5	`openssl passwd -aixmd5`	No

Example

Encrypted:

```yaml
os:
  password:
"$6$kZYUnRaTxNdg4W8H$WSEJydGWsNpaRbbbRdTDLJ2hDLbkizxSFGW2RtexlqG6njEATaGQG9ssztjaKDCsaN
```

Or clear text:

```yaml
os:
  password: supersecure
```

`os.environment`

Definition

Environment variables to be set on K3s and other processes like the boot process. Primary use of this field is to set the HTTP proxy.

Example

```yaml
os:
  environment:
    http_proxy: http://myserver
    https_proxy: http://myserver
```

:::note

This example sets the HTTP(S) proxy for **foundational OS components**. To set up an HTTP(S) proxy for Harvester components such as fetching external images and backup to S3 services, see Settings/http-proxy.

:::

`os.labels`

Definition

Labels to be added to this Node.

Example

```yaml
os:
  labels:
    topology.kubernetes.io/zone: zone1
    foo: bar
    mylabel: myvalue
```

`os.sshd.sftp`

Definition

Subsystem used to configure the OpenSSH Daemon (sshd). Harvester currently only supports `sftp` .

Example

```yaml
os:
  sshd:
    sftp: true  #  The SFTP subsystem is enabled.
```

`install.addons`

Versions: v1.2.0 and later

Definition: Setting that defines the default addon status. Harvester addons are disabled by default.

Supported values:

- `vm-import-controller` (chartName: harvester-vm-import-controller)
- `pcidevices-controller` (chartName: harvester-pcidevices-controller)
- `rancher-monitoring`
- `rancher-logging`
- `harvester-seeder` (experimental)

Example:

```
install:
  addons:
    rancher_monitoring:
      enabled: true
    rancher_logging:
      enabled: false
```

`install.automatic`

Definition: Setting that forces the installer to skip the interactive steps in the installation process.

When enabled, the configuration is either retrieved from the value of `harvester.install.config_url` or defined individually using kernel parameters.

`install.data_disk`

Versions: v1.0.1 and later

Definition: Default device for storing VM data.

When installing via PXE, use `/dev/disk/by-id/$id` or `/dev/disk/by-path/$path` to specify the storage device if the server contains multiple physical volumes.

Default value: Storage device configured in the setting `install.device`

Example:

```
install:
  data_disk: /dev/sdb
```

`install.debug`

Definition: Setting that enables additional logging and debugging during installation.

`install.device`

Definition: Device on which the Harvester OS is installed.

When installing via PXE, use `/dev/disk/by-id/$id` or `/dev/disk/by-path/$path` to specify the storage device if the server contains multiple physical volumes.

`install.force_efi`

Definition: Setting that forces EFI installation even when EFI is not detected.

Default value: `false`

`install.force_mbr`

Definition: Setting that forces usage of MBR partitioning on BIOS systems.

Harvester uses GPT partitioning on UEFI and BIOS systems by default. Compatibility issues may require you to use MBR partitioning instead.

If you specify the same storage device for both `install.device` and `install.data_disk`, Harvester creates an additional partition for storing VM data. This additional partition is not created when you force usage of MBR partitioning. Instead, VM data is stored in a partition that stores OS data.

Example:

```yaml
install:
  force_mbr: true
```

`install.harvester.longhorn.default_settings.guaranteedInstanceManagerCPU`

Versions: v1.4.0 and later

Definition: Percentage of the total allocatable CPU on each node to be reserved for each Longhorn Instance Manager pod.

Using the default value is recommended for high system availability. When deploying single-node Harvester clusters, you can specify a value less than 12.

For more information about how to set the correct value, see Guaranteed Instance Manager CPU in the Longhorn documentation.

Default value: 12

Supported values: 0 to 12. All other values are considered 12.

Example:

```yaml
harvester:
  longhorn:
    default_settings:
      guaranteedInstanceManagerCPU: 6
```

`install.harvester.storage_class.replica_count`

Versions: v1.1.2 and later

Definition: Replica count of the default Harvester StorageClass `harvester-longhorn`.

Using the default value is recommended for high storage availability. When deploying single-node Harvester clusters, you must set the value to 1.

For more information, see Default Replica Count in the Longhorn documentation.

Default value: 3

Supported values: 1 to 3. All other values are considered 3.

Example:

```yaml
install:
  harvester:
    storage_class:
      replica_count: 1
```

`install.iso_url`

Definition: URL of ISO image to be downloaded and used to install Harvester when booting from the kernel or vmlinuz.

`install.management_interface`

Definition: Network interfaces for the host machine.

Harvester uses the systemd net naming scheme. Ensure that the interface name is present on the target machine before installation.

Fields:

- `method` : Method used to assign an IP to the network. Supported values:
  - `dhcp` : Harvester requests an IP from the DHCP server.
  - `static` : IP and gateway addresses are manually assigned.
- `ip` : Static IP assigned to the network. This field is required when the value of `method` is `static` .
- `subnet_mask` : Subnet mask of the network. This field is required when the value of `method` is `static` .
- `gateway` : Gateway address assigned to the network. This field is required when the value of `method` is `static` .
- `interfaces` : Array of network interfaces. The installer combines the specified interfaces (slaves) into a single logical bonded interface.
  - `interfaces.name` : Name of a slave interface.
  - `interfaces.hwAddr` : Hardware MAC address of a slave interface. This field is optional.
- `bond_options` : Options for bonded interfaces. When unspecified, the following options are used:
  - `mode` : balance-tlb
  - `miimon` : 100
- `mtu` : Maximum transmission unit (MTU) for the interface.
- `vlan_id` : VLAN ID for the interface.

Example:

```yaml
install:
  mode: create
  management_interface:
    interfaces:
    - name: ens5
      hwAddr: "B8:CA:3A:6A:64:7D"  # Optional
    method: dhcp
    bond_options:
      mode: balance-tlb
      miimon: 100
    mtu: 1492
    vlan_id: 101
```

`install.mode`

Definition: Mode of installing Harvester.

Supported values:

- `create` : Create a new Harvester installation.
- `join` : Join an existing Harvester installation. You must specify the `server_url` .

Example:

```
install:
  mode: create
```

`install.no_format`

Definition: Setting that prevents partitioning and formatting of the installation disk.

`install.persistent_partition_size`

Definition: Size of the partition COS_PERSISTENT in Gi or Mi.

This partition stores data such as system packages and container images. The minimum value is 150 Gi.

Default value: 150 Gi

Example:

```
install:
  persistent_partition_size: 150Gi
```

`install.poweroff`

Definition: Setting that shuts down (instead of rebooting) the server after installation.

`install.rawdiskimagepath`

Definition: Setting that forces the installer to only install the Harvester hypervisor (without any configuration). You must enable `harvester.install.automatic` to use this setting.

`install.role`

Definition: Role assigned to a node at the time of installation. When unspecified, Harvester assigns the `default` role.

- `default` : Allows a node to function as a management node or a worker node.
- `management` : Allows a node to be prioritized when Harvester promotes nodes to management nodes.
- `worker` : Restricts a node to being a worker node (never promoted to management node) in a specific cluster.

- `witness` : Restricts a node to being a witness node (only functions as an etcd node) in a specific cluster.

`install.silent`

> *Definition: Reserved*

`install.skipchecks`

Definition: Setting that allows installation to proceed even if minimum requirements for production use are not met

The installer automatically checks if the hardware meets the minimum requirements for production use. When performing automated installation via PXE Boot, if any of the checks fail, installation is stopped, and warnings are printed to the system console and saved to `/var/log/console.log` in the installation environment.

To override this behavior, set `install.skipchecks=true` . When set to `true` , warning messages are still saved to `/var/log/console.log` , but the installation proceeds even if hardware requirements for production use are not met.

Default value: `false`

Example:

```
install:
  skipchecks: true
```

`install.tty`

Definition: TTY device used for the console.

Example:

```
install:
  tty: ttyS0,115200n8
```

`install.vip`

Definition: VIP of the Harvester management endpoint.

After installation, you can access the Harvester UI at `https://<VIP>` .

`install.vip_mode`

Definition: Mode of assigning the VIP.

Supported values:

- `dhcp` : Harvester sends DHCP requests to get the VIP. You must specify the hardware address using the `install.vip_hw_addr` field.
- `static` : Harvester uses a static VIP.

Example:

```yaml
install:
  vip: 192.168.0.100
  vip_mode: static
```

`install.vip_hw_addr`

Definition: Hardware address corresponding to the VIP.

You must configure an on-premises DHCP server to offer the configured VIP. This field is required when the value of `install.vip_mode` is `dhcp`. For more information, see [Management Address]().

Example:

```yaml
install:
  vip: 10.10.0.19
  vip_mode: dhcp
  vip_hw_addr: 52:54:00:ec:0e:0b
```

`install.cluster_pod_cidr`

Definition: CIDR of the Harvester pods.

Use this field to override the default pod CIDR of 10.52.0.0/16.

Example:

```yaml
install:
  cluster_pod_cidr: 172.16.0.0/16
```

`install.cluster_service_cidr`

Definition: CIDR of the Harvester services.

Use this field to override the default service CIDR of 10.53.0.0/16.

:::info important

If you change this CIDR, you must ensure that the cluster DNS IP (`install.cluster_dns`) is within this range.

:::

Example:

```yaml
install:
  cluster_service_cidr: 172.22.0.0/16
```

`install.cluster_dns`

Definition: IP of the Harvester DNS service.

Use this field to override the default DNS service IP of 10.53.0.10.

:::info important

This IP must be within the range defined by the `cluster_service_cidr` field.

:::

Example:

```yaml
install:
  cluster_dns: 172.16.0.10
```

`install.webhooks`

Definition: Webhooks that allow you to receive notifications for certain installer-related events.

The installer sends HTTP requests to the specified URL. Multiple requests can be sent for a single event but if one request fails, the remaining requests are not sent.

Fields:

- `event` : Event type that triggers an HTTP action on the webhook.
  - `STARTED` : The installation has started.
  - `SUCCEEDED` : The installation was completed without errors.
  - `FAILED` : The installation was unsuccessful.
- `method` : HTTP method
- `url` : URL to which HTTP requests are sent
- `insecure` : When set to `true` , Harvester does not verify the server's certificate. The default value is `false` .
- `basicAuth` : When set to `true` , Harvester uses the "Basic" HTTP authentication scheme.
- `headers` : When set to `true` , custom headers are included in the HTTP requests. Headers such as `Content-Length` are automatically included.
- `payload` *: When set to `true` , payload data is sent with the HTTP requests. You may need to set the correct Content-Type header in the `headers` field to ensure that the server accepts the request.

Example:

```yaml
install:
  webhooks:
    - event: SUCCEEDED
      method: GET
      url: http://10.100.0.100/cblr/svc/op/nopxe/system/{{.Hostname}}
    - event: STARTED
      method: GET
      url: https://10.100.0.100/started/{{.Hostname}}
      insecure: true
      basicAuth:
        user: admin
        password: p@assword
    - event: FAILED
      method: POST
      url: http://10.100.0.100/record
      headers:
        Content-Type:
           - 'application/json; charset=utf-8'
      payload: |
        {
          "host": "{{.Hostname}}",
          "device": "hd"
        }
```

`install.wipedisks`

Definition: Setting that clears all disk partitions on the host using the `sgdisk` command.

`install.with-net-images`

Definition: Setting that determines if images are pulled from the internet after installation.

The value of this field is typically derived from the kernel parameter `harvester.install.with_net_images`. When the value is `true`, Harvester does not preload images packaged in the installation medium, and instead pulls images from the internet when necessary.

`system_settings`

Definition

You can overwrite the default Harvester system settings by configuring `system_settings`. See the Settings page for additional information and the list of all the options.

:::note

Overwriting system settings only works when Harvester is installed in "create" mode. If you install Harvester in "join" mode, this setting is ignored. Installing in "join" mode will adopt the system settings from the existing Harvester system.

:::

Example

The example below overwrites `containerd-registry`, `http-proxy` and `ui-source` settings. The values must be a `string`.

```
system_settings:
  containerd-registry: '{"Mirrors": {"docker.io": {"Endpoints": ["https://myregistry.local:5000"]}}, "Configs": {"myregistry.local:5000": {"Auth": {"Username": "testuser", "Password": "testpassword"}, "TLS": {"InsecureSkipVerify": false}}}}'
  http-proxy: '{"httpProxy": "http://my.proxy", "httpsProxy": "https://my.proxy", "noProxy": "some.internal.svc"}'
  ui-source: auto
```