



Cloud

To install Deckhouse, you will need a PC and a cloud site.  
The following steps of the Getting Started guide will help you to prepare the cloud.



config.yml



resources.yml

Cloud



Installation is based on two configuration files.  
The following steps of the Getting Started guide will help you to create it correctly ...

```
apiVersion: deckhouse.io/v1alpha1
kind: ClusterConfiguration
type: Cloud
...
apiVersion: deckhouse.io/v1alpha1
kind: InitConfiguration
...
```



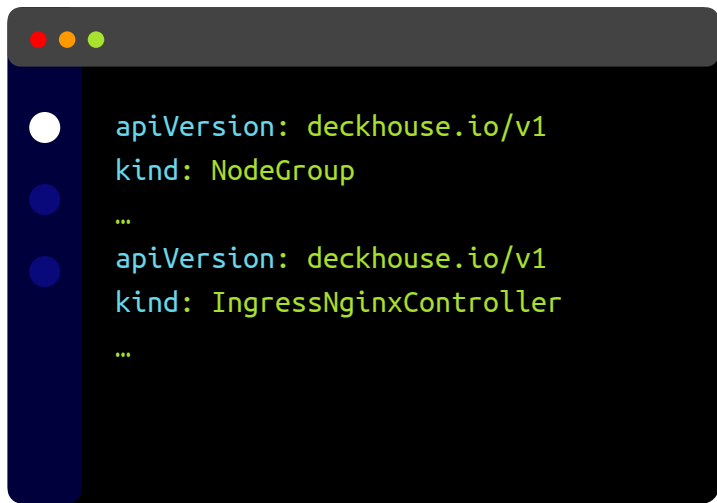
config.yml



resources.yml



... the first one describes cloud API access settings, master nodes parameters and initial settings for Deckhouse controller ...



```
apiVersion: deckhouse.io/v1
kind: NodeGroup
...
apiVersion: deckhouse.io/v1
kind: IngressNginxController
...
```




config.yml



resources.yml



... the second file describes additional Kubernetes resources that should be created during initialization process.  
These resources contain specifications for worker nodes, Ingress controller parameters  
and other manifests.



```
$ dhctl bootstrap --config config.yml --resources resources.yml
```



config.yml



resources.yml

The resulting configuration files are passed to the dhctl utility and it starts the installation.

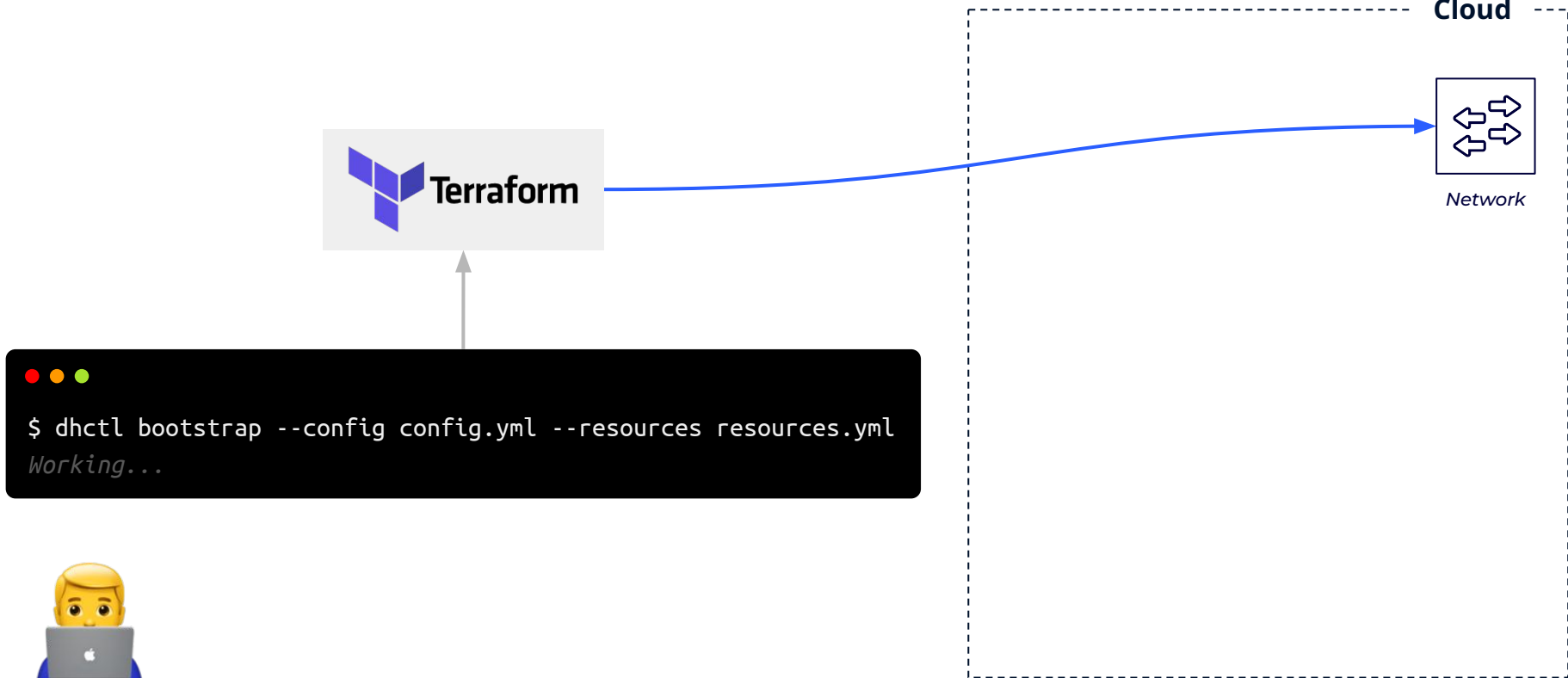


```
$ dhctl bootstrap --config config.yml --resources resources.yml  
Working...
```

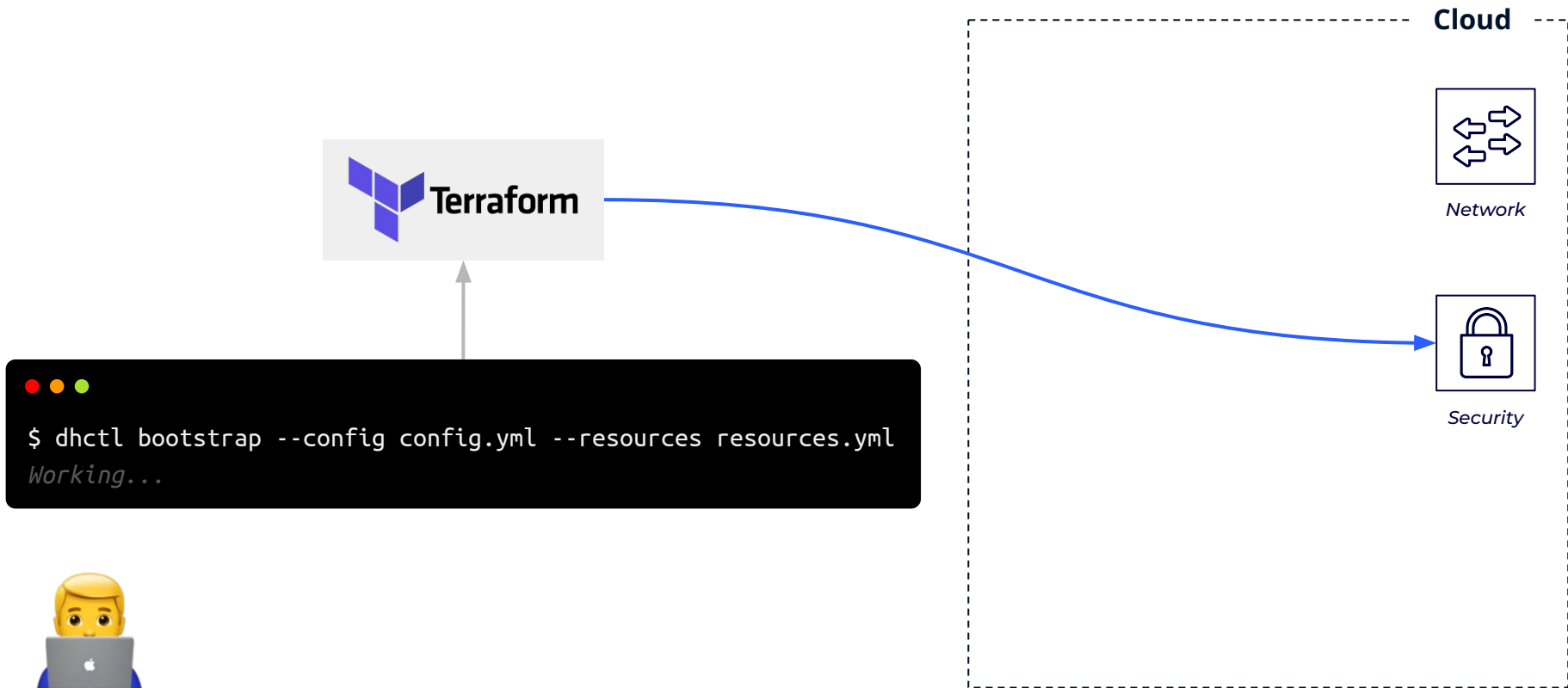


Cloud

With built-in Terraform, dhctl rolls out the basic cloud infrastructure ...



... configures the network ...

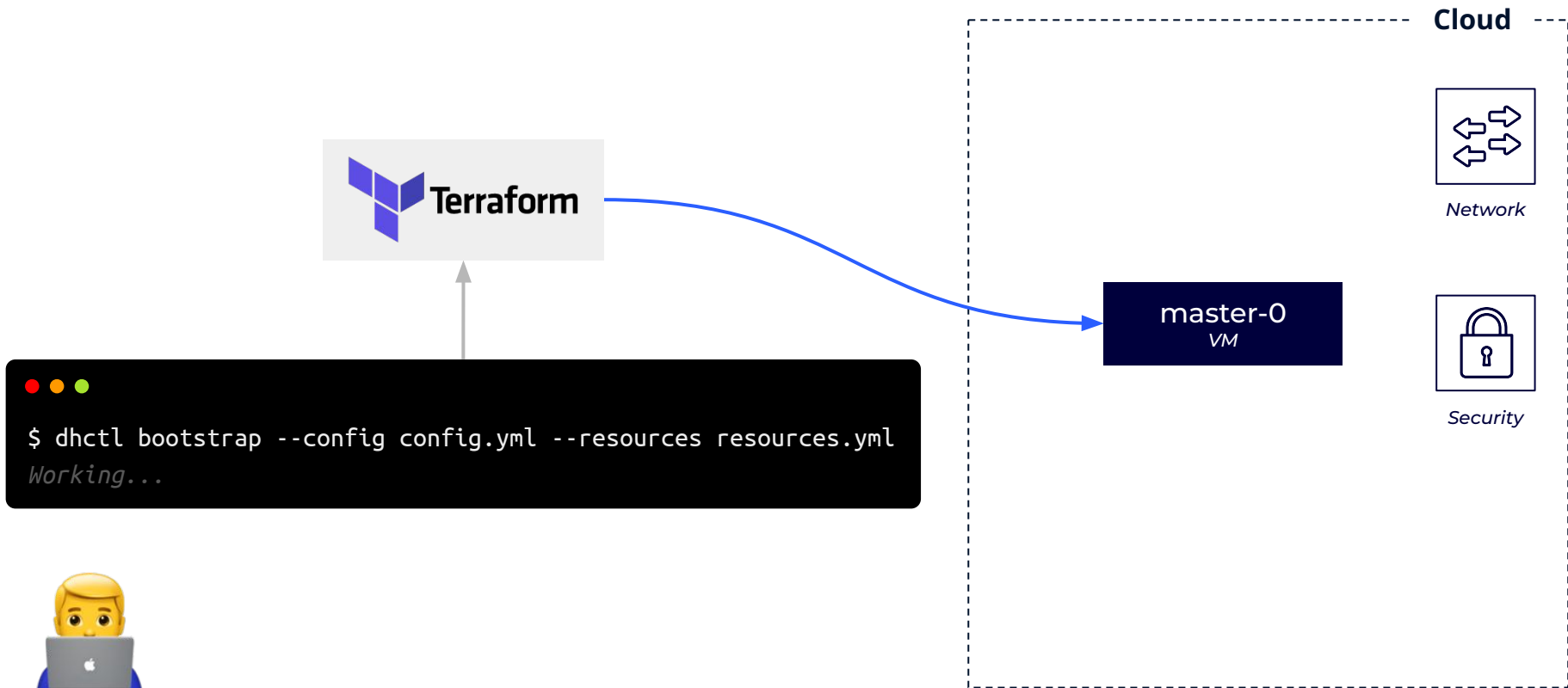


... solves basic security issues and more.

Basic infrastructure has different nature depending on the cloud type and chosen layout.

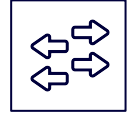
Layout types will be described in the following steps of the Getting Started guide.





With Terraform, dhctl creates a virtual machine (or several) for the future Kubernetes master node.

```
$ dhctl bootstrap --config config.yml --resources resources.yml
Working...
```



Network

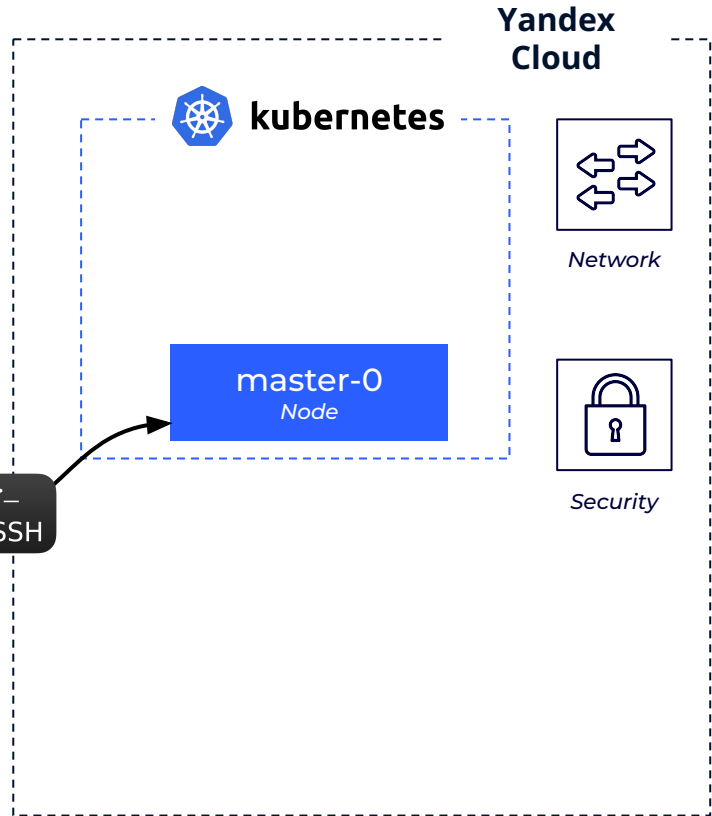


Security

Cloud

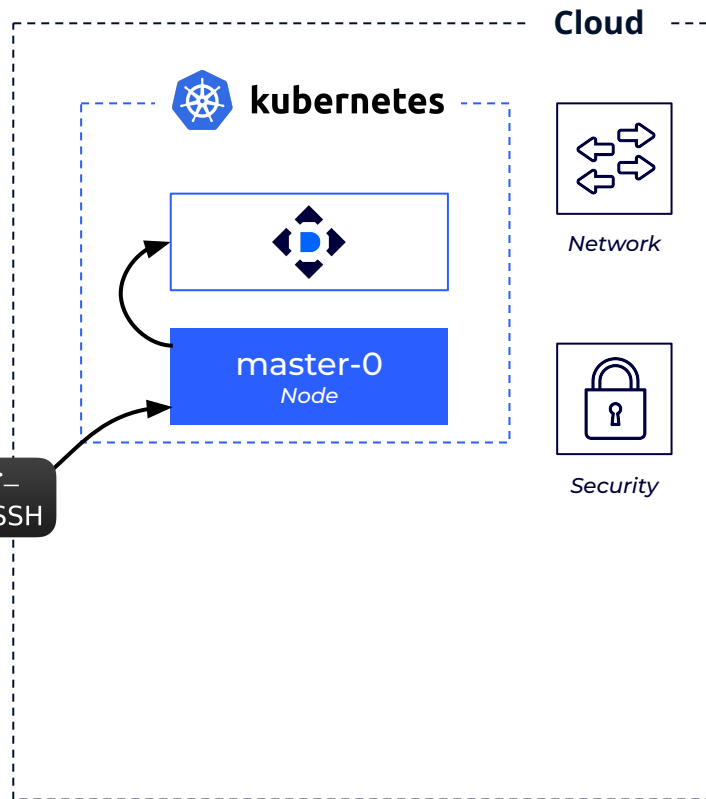
Then, the utility connects to this VM via SSH ...

```
$ dhctl bootstrap --config config.yml --resources resources.yml
Working...
```



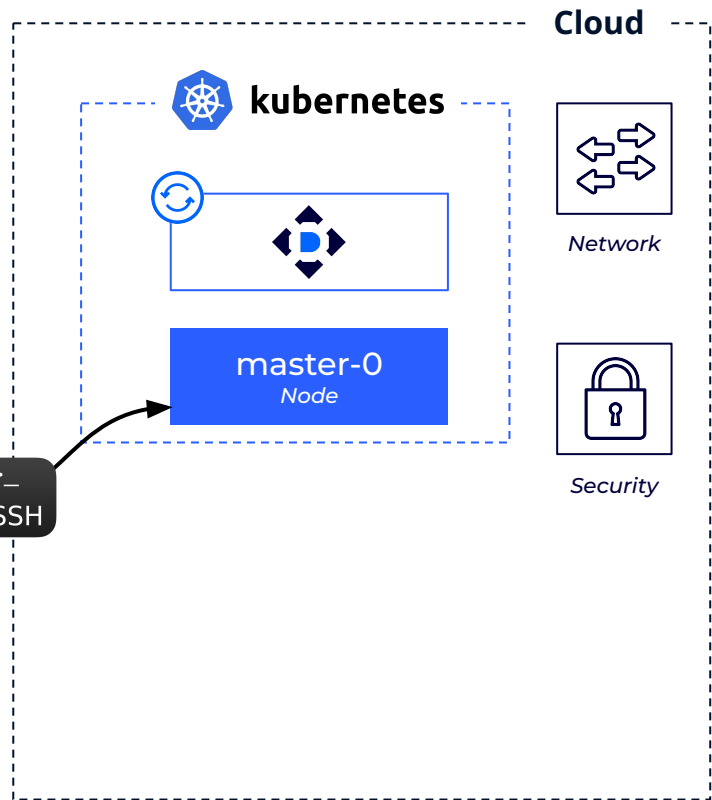
... and initializes the Kubernetes cluster.  
At this stage, a minimal Kubernetes vanilla cluster is ready.

```
$ dhctl bootstrap --config config.yml --resources resources.yml
Working...
```



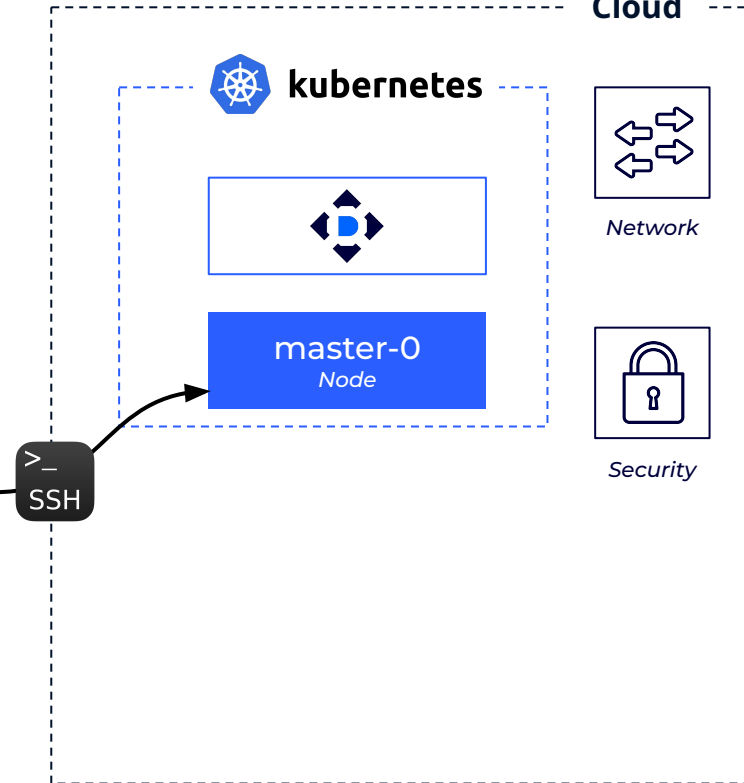
To complete the installation, dhctl installs a Deckhouse controller in the cluster.

```
$ dhctl bootstrap --config config.yml --resources resources.yml
Working...
```



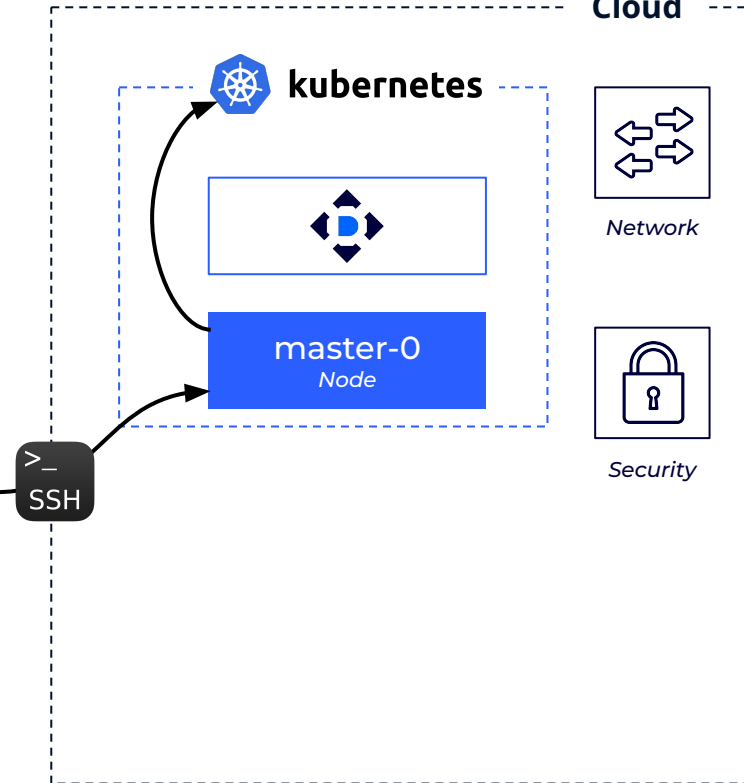
The Deckhouse controller installs the necessary modules and establishes a connection to the cloud API.

```
$ dhctl bootstrap --config config.yml --resources resources.yml
Working...
```

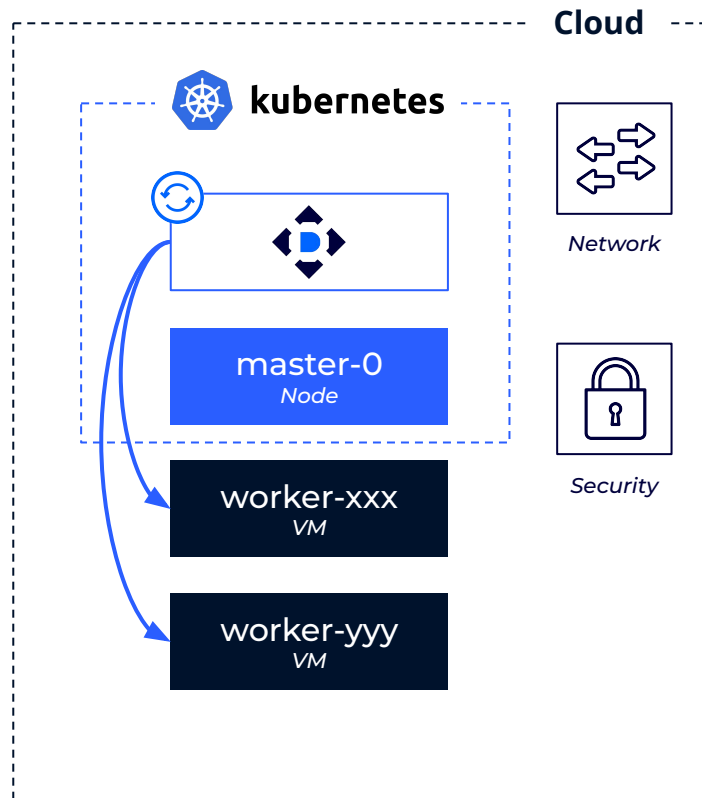


The base cluster is ready.

```
$ dhctl bootstrap --config config.yml --resources resources.yml
Working...
```

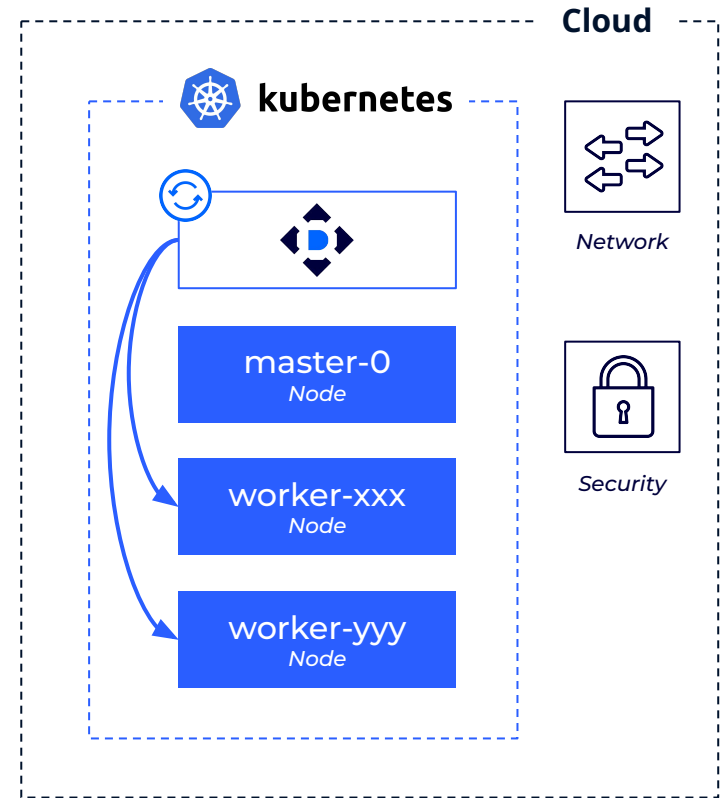


It remains to apply additional Kubernetes resources from the resources.yml file.

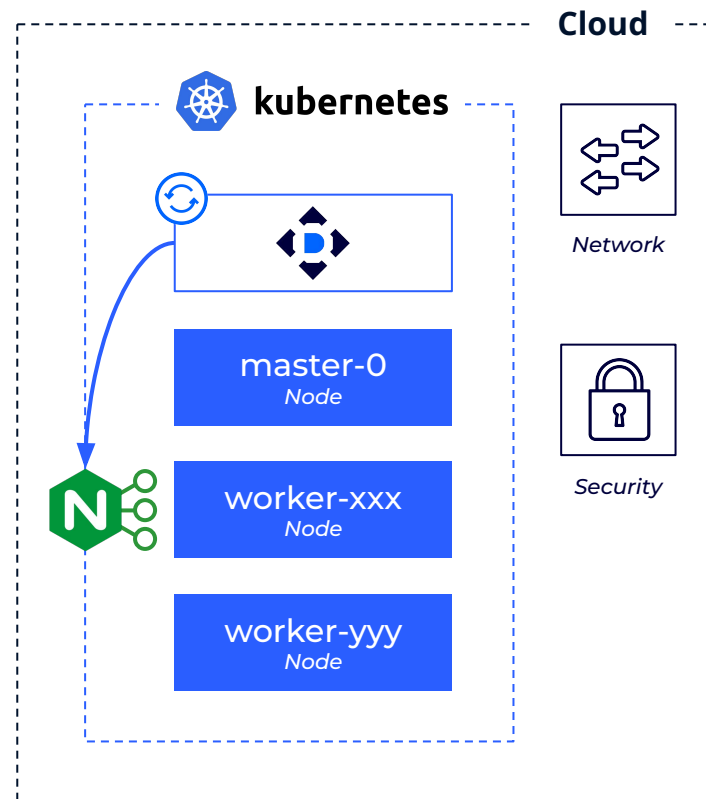


The Deckhouse controller reacts to the creation of the resources  
and creates the required set of nodes ...

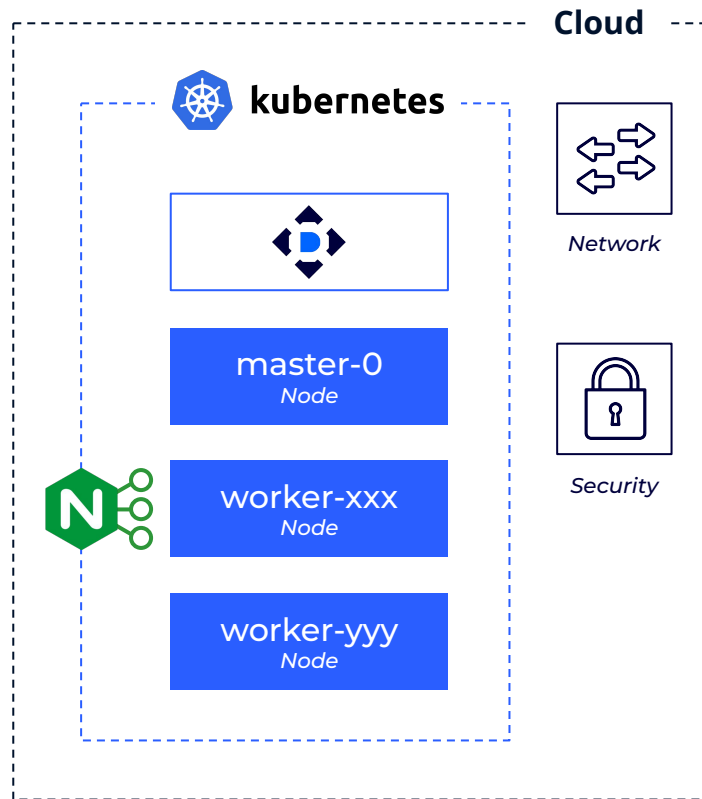




... and then joins them to the cluster.



In addition, the Deckhouse controller sets up the Ingress controller.



The cluster is ready to work!