Deckhouse
Kubernetes Platform

Getting started

Cloud

Cloud

To install Deckhouse Kubernetes Platform (DKP),
you will need a PC and a cloud site.

Cloud

config.yml

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

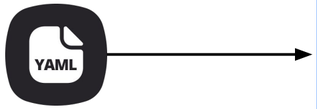
config.yml

```
apiVersion: deckhouse.io/v1alpha1
kind: Cluster Configuration
type: Cloud
…
apiVersion: deckhouse.io/v1alpha1
kind: InitConfiguration
…
apiVersion: deckhouse.io/v1
kind: NodeGroup
…
apiVersion: deckhouse.io/v1
kind: IngressNginxController
```

Cloud

In a config.yml file, you define the cloud API access settings, the master node parameters, and the initial DKP controller configuration, as well as additional Kubernetes resources created during initialization: worker node specifications, Ingress controller parameters, and other manifests...

Cloud



config.yml

```
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
```

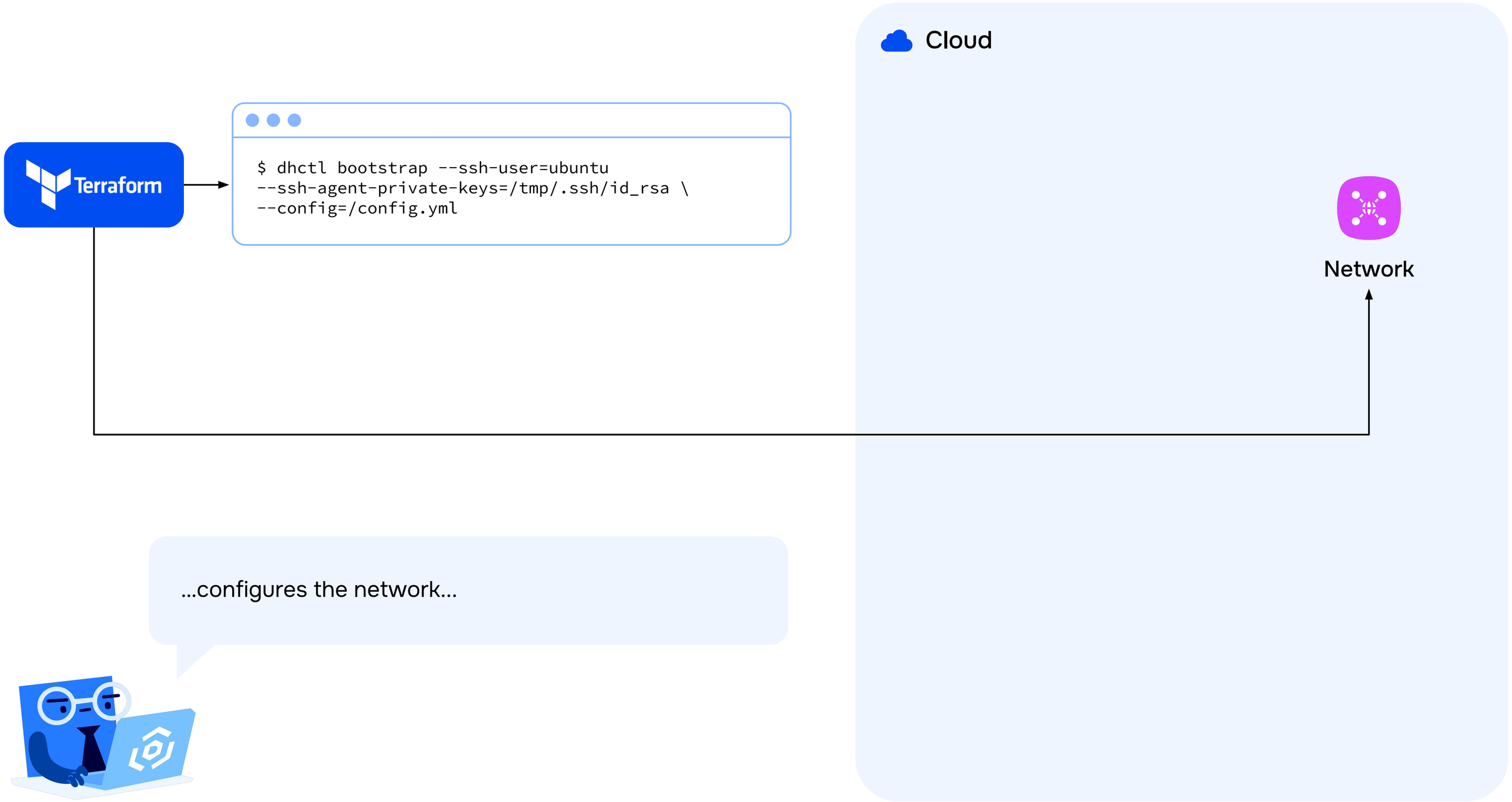
The resulting configuration file are passed to the dhctl utilityand it starts the installation.

Terraform

```
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
```

Cloud

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


Cloud
Terraform
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
Network
...configures the network...

Terraform

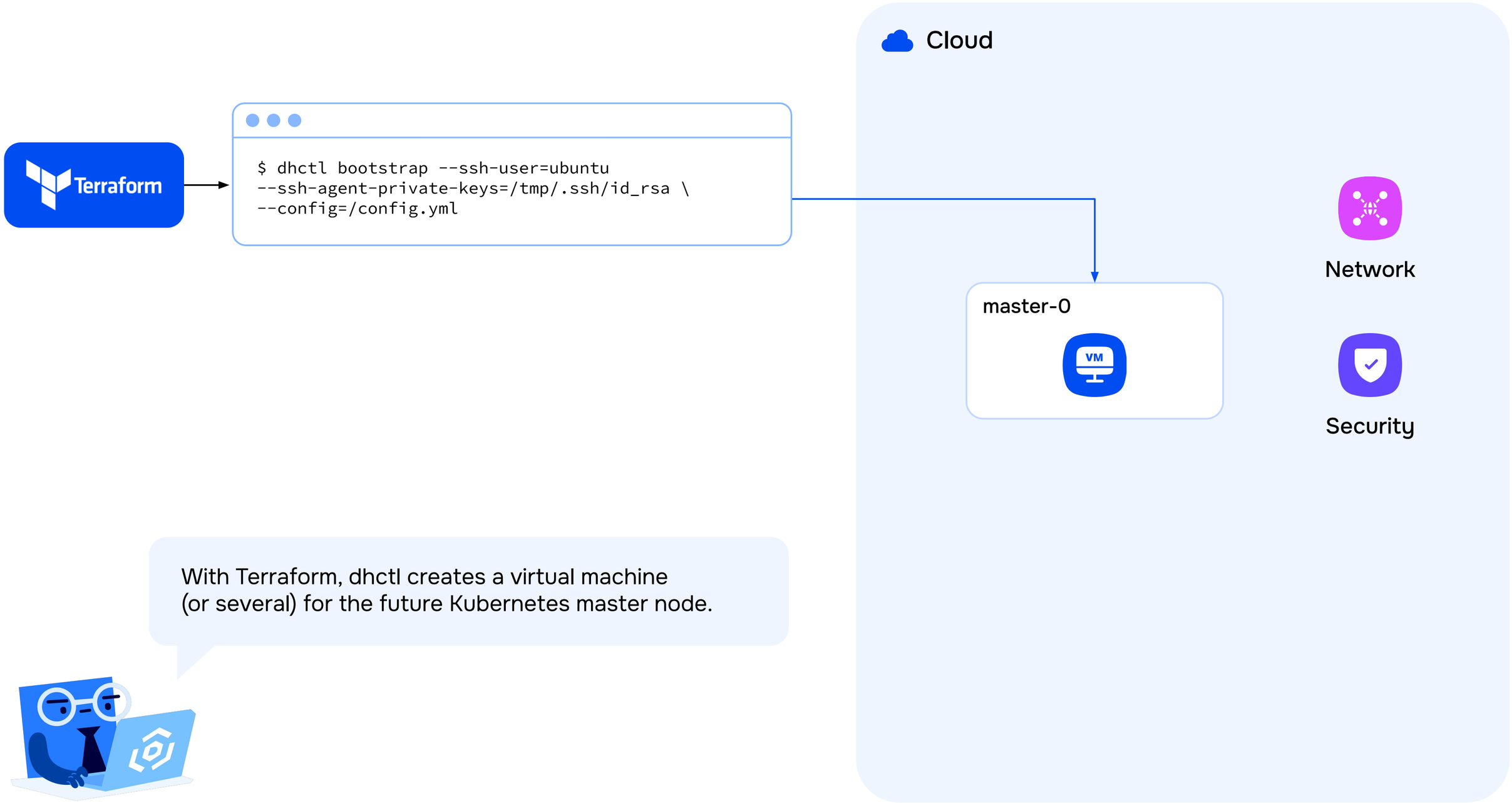
```
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
```

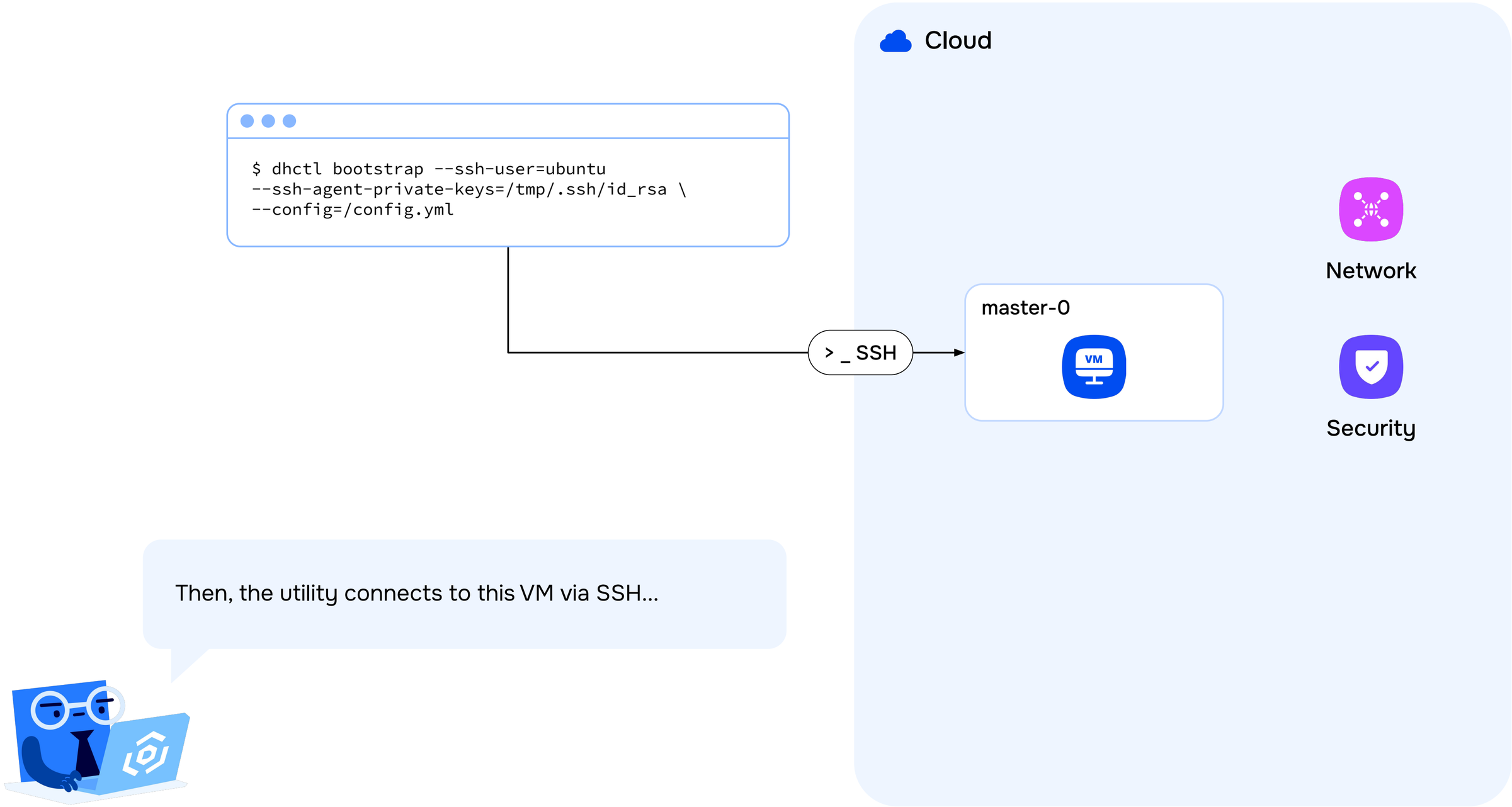
Cloud

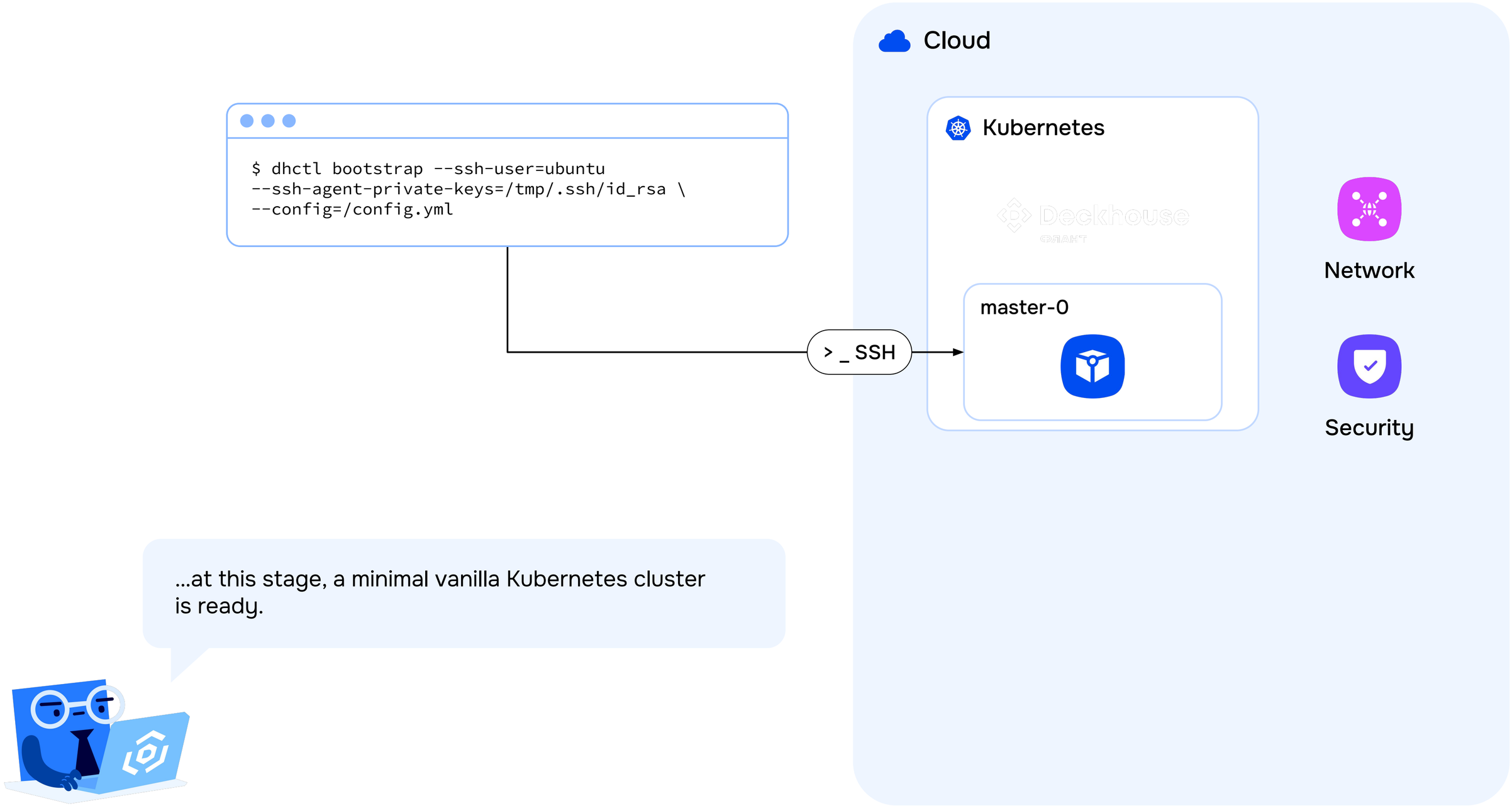
Network

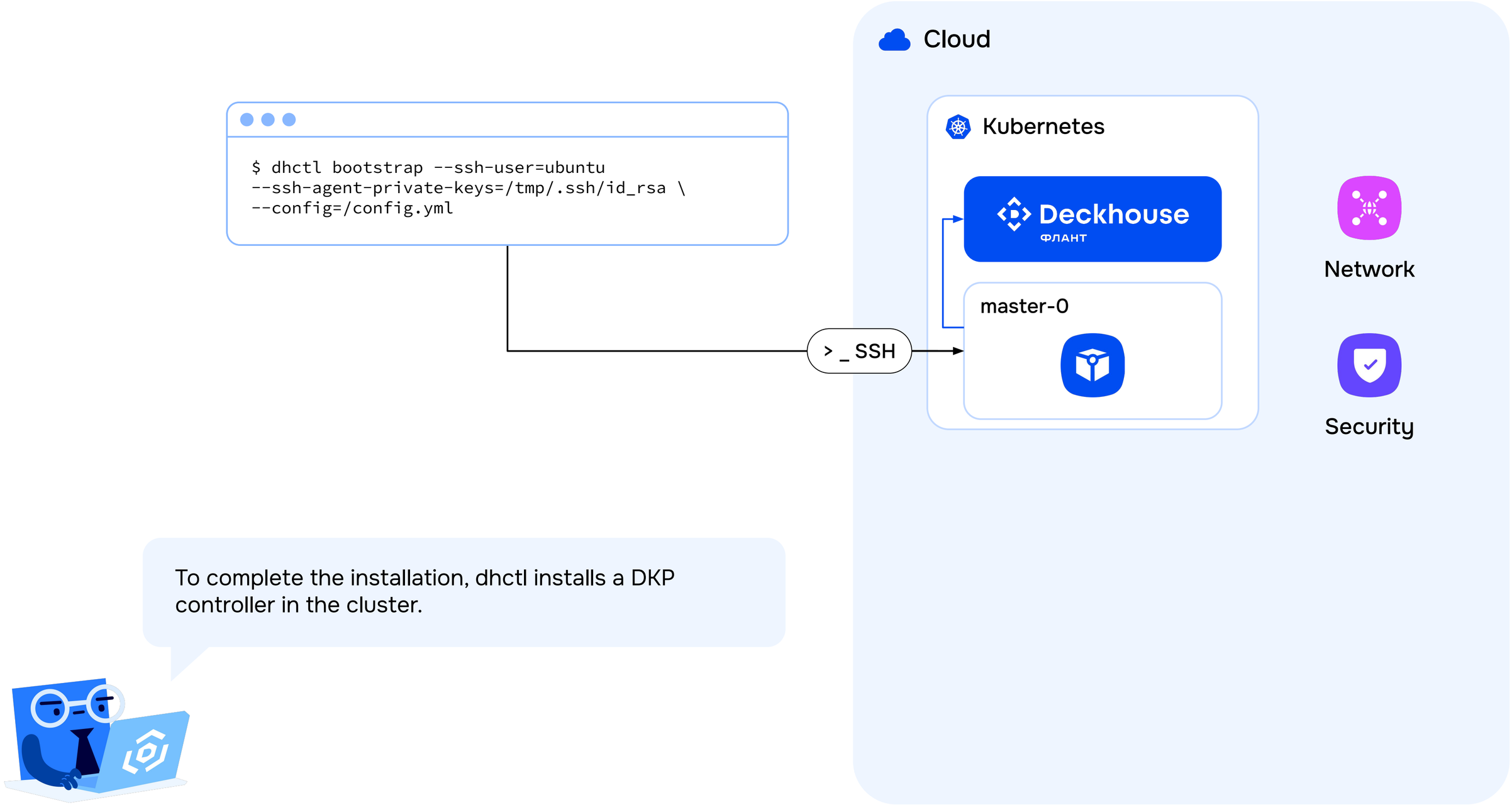
Security

...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.


Terraform
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
Cloud
master-0
VM
Network
Security
With Terraform, dhctl creates a virtual machine
(or several) for the future Kubernetes master node.


Cloud
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
> _ SSH
master-0
VM
Network
Security
Then, the utility connects to this VM via SSH...


Cloud
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
> _ SSH
Kubernetes
Deckhouse
ФЛАНТ
master-0
Network
Security
...at this stage, a minimal vanilla Kubernetes cluster is ready.



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

```
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
```

> _ SSH

Cloud

Kubernetes

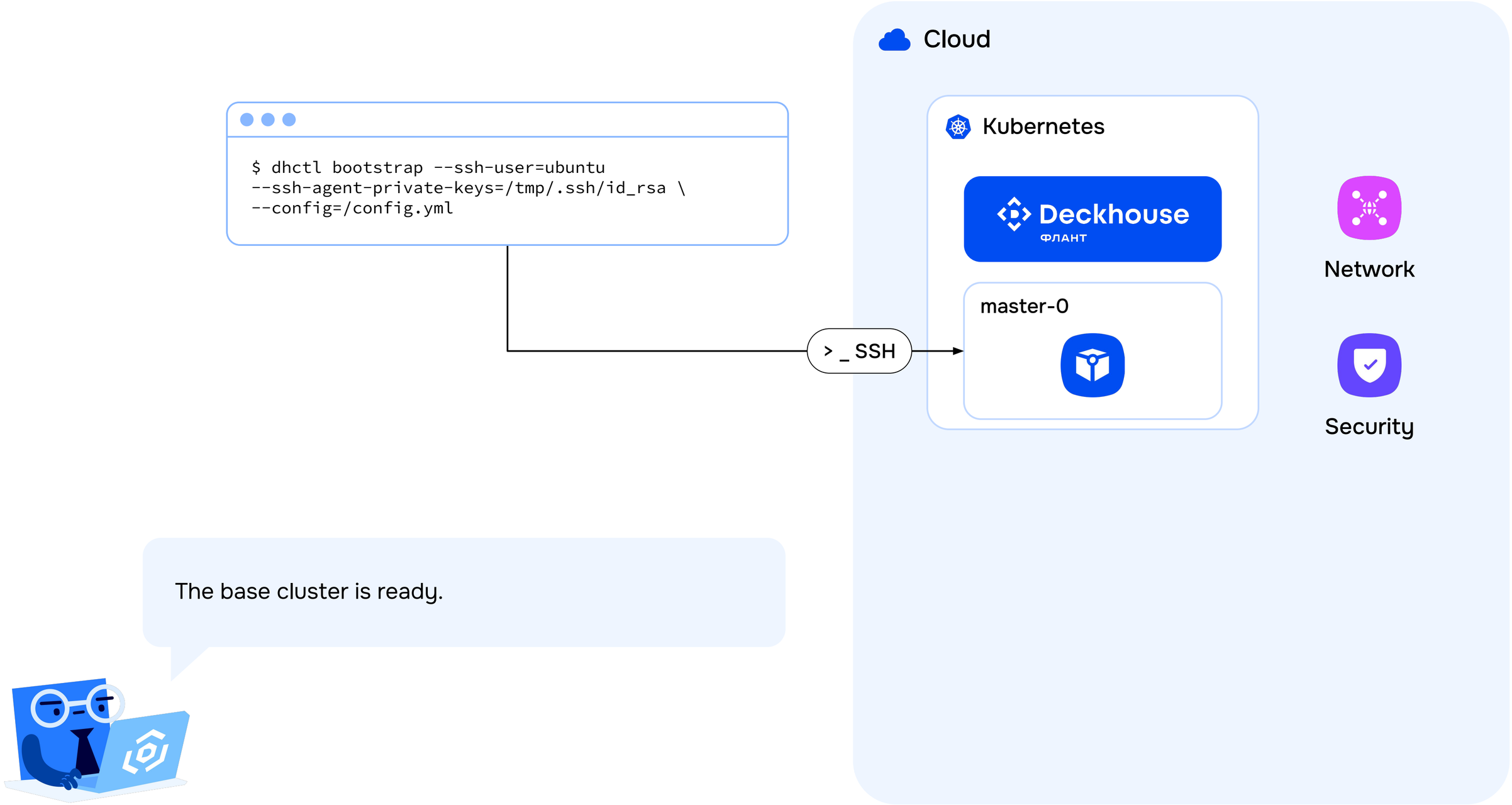
Deckhouse
ФЛАНТ

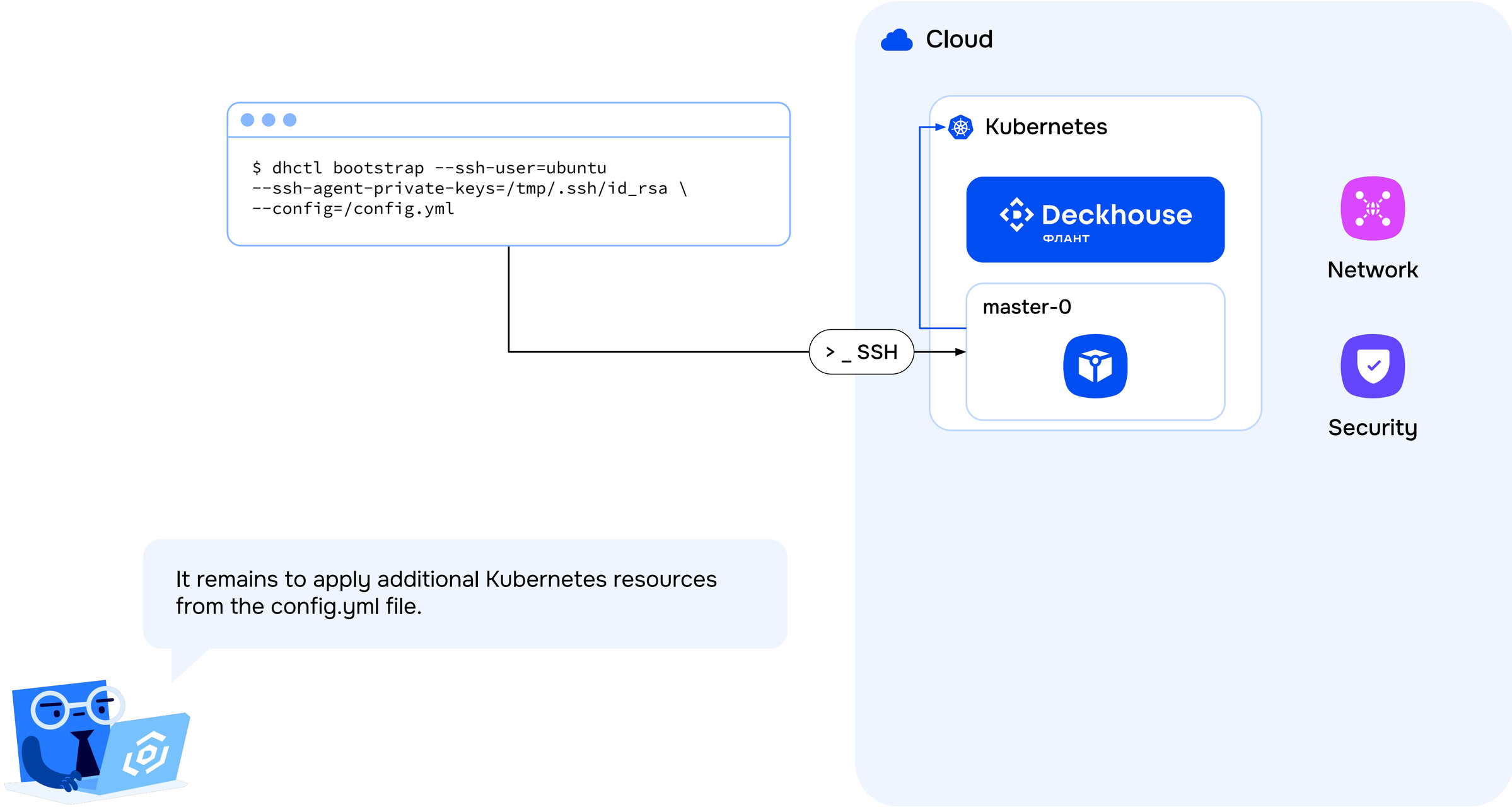
master-0

Network

Security

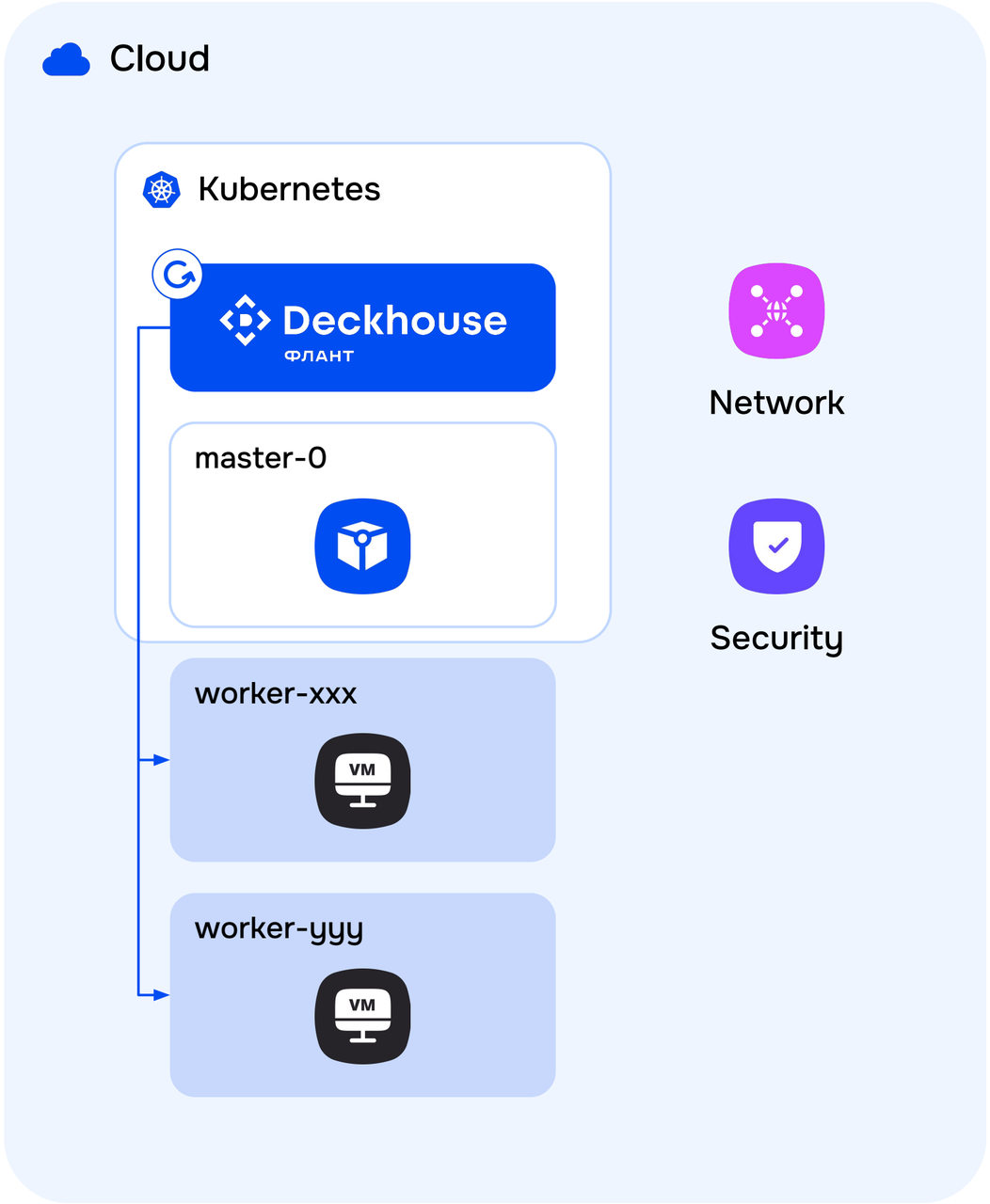
The DKP controller installs the necessary modules and establishes a connection to the cloud API.


Cloud
$ dhctl bootstrap --ssh-user=ubuntu
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
--config=/config.yml
> _ SSH
Kubernetes
Deckhouse
ФЛАНТ
master-0
Network
Security
The base cluster is ready.

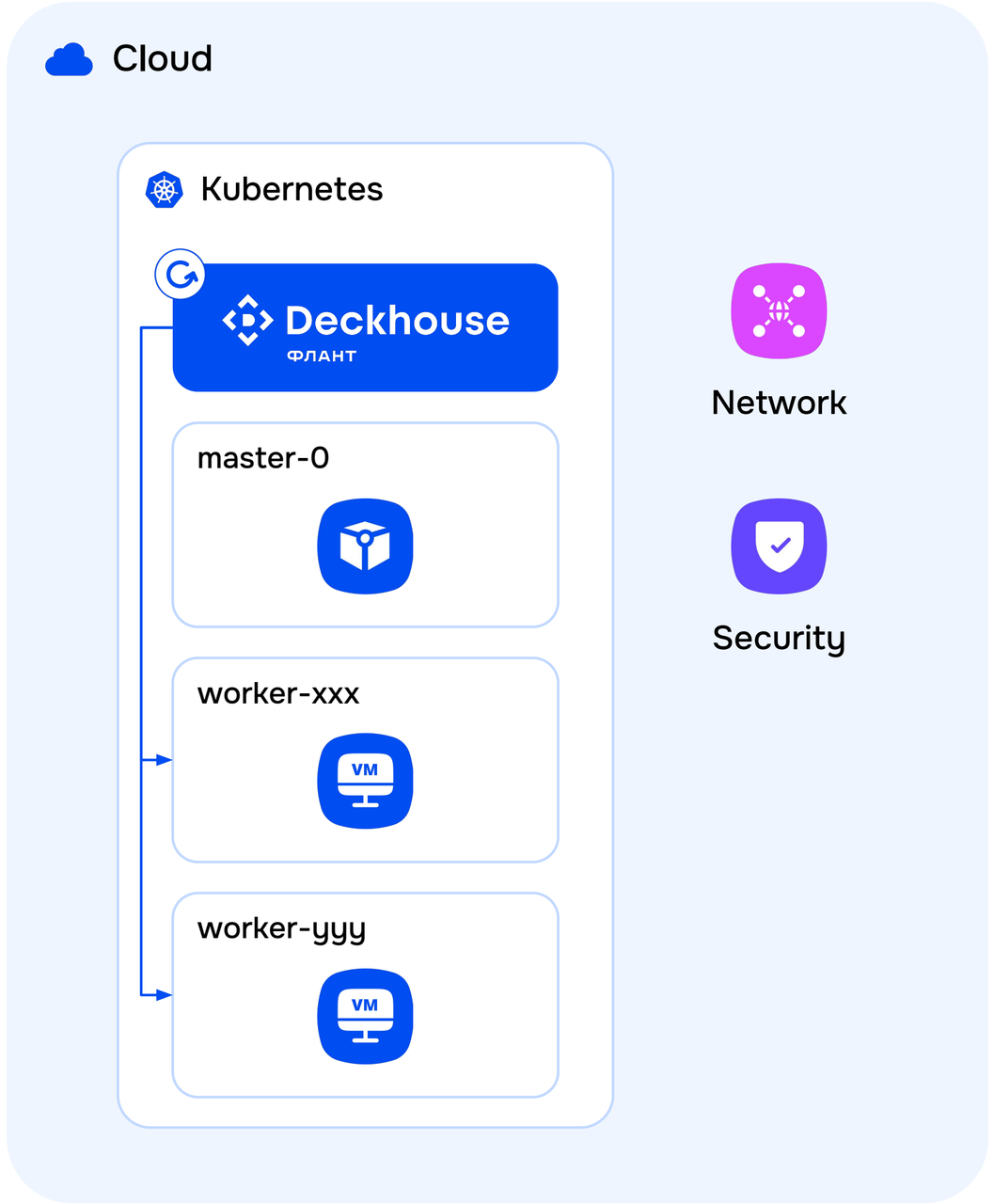


It remains to apply additional Kubernetes resources from the config.yml file.

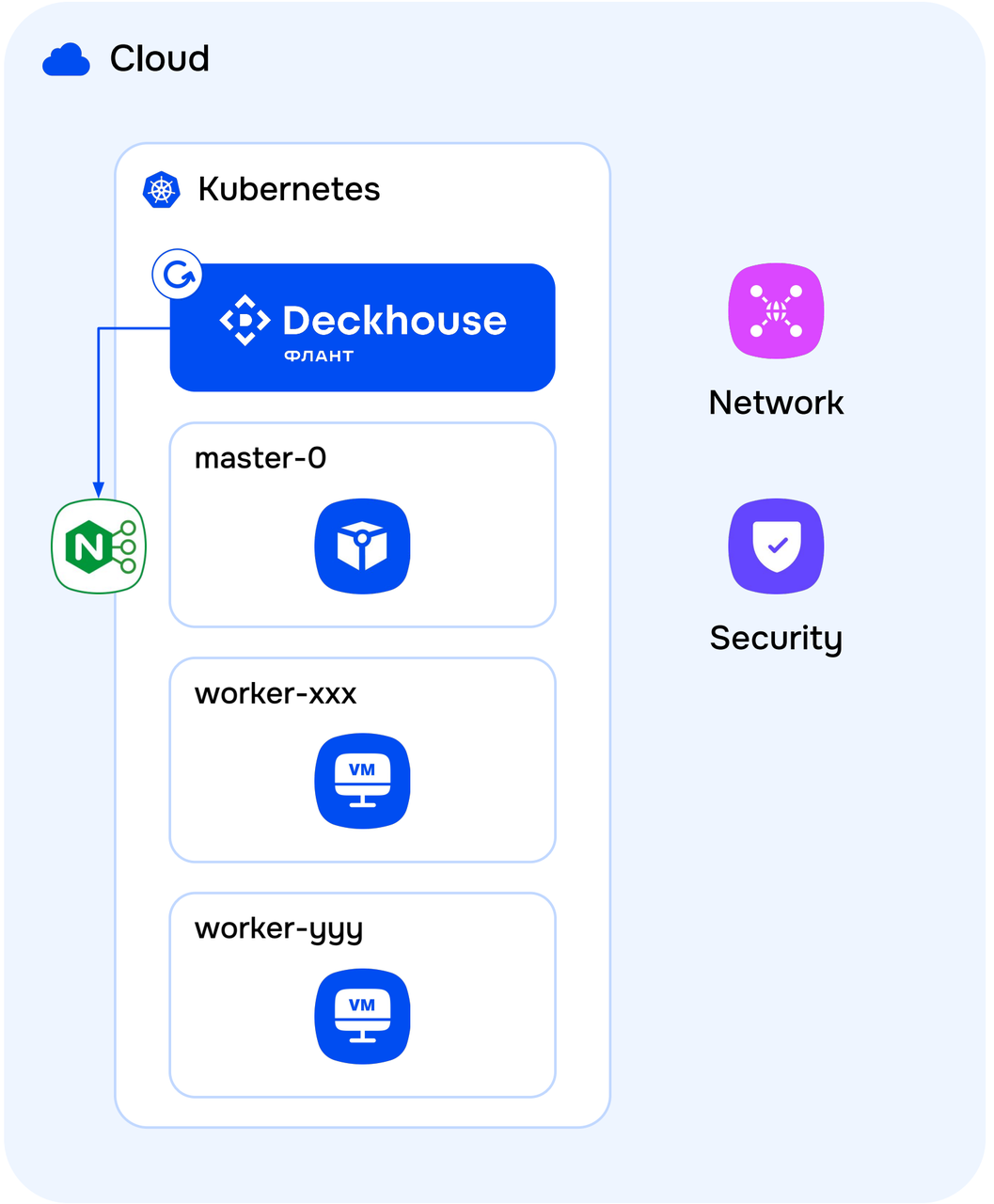
The DKP controller reacts to the creation of the resources and creates the required set of nodes...



...and then joins them to the cluster.



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



The cluster is ready to work!