Deckhouse
Kubernetes Platform

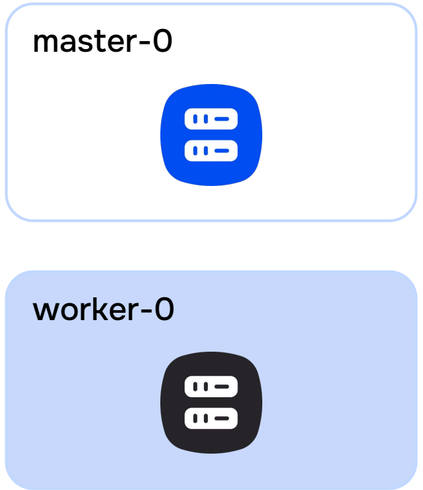
Getting started

Bare metal

To install Deckhouse Kubernetes Platform (DKP), you will need a PC and an SSH-accessible server with supported OS installed.

master-0

If you need additional nodes, prepare them in the same way.



YAML

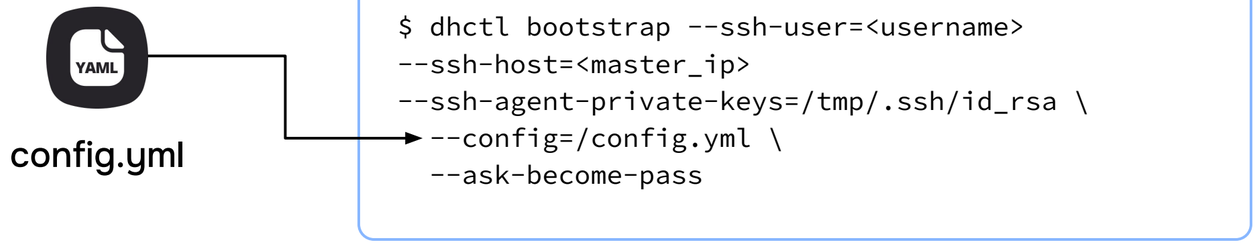
config.yml

```
apiVersion: deckhouse.io/v1alpha1
kind: ClusterConfiguration
type: Static
…
```

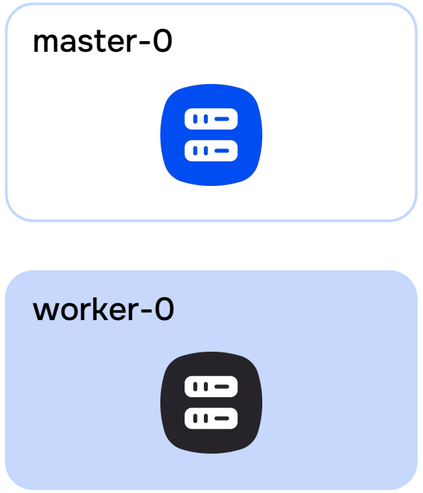
Installation is based on the configuration file.
The following steps of the Getting Started guide
will help you to create it correctly.

master-0

worker-0



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



```
$ dhctl bootstrap --ssh-user=<username>
--ssh-host=<master_ip>
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
  --config=/config.yml \
  --ask-become-pass
```

> _ SSH

master-0

worker-0

Then utility connects to master server via SSH...

```
$ dhctl bootstrap --ssh-user=<username>
--ssh-host=<master_ip>
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
  --config=/config.yml \
  --ask-become-pass
```

> _ SSH

Kubernetes

master-0

worker-0

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

```
$ dhctl bootstrap --ssh-user=<username>
--ssh-host=<master_ip>
--ssh-agent-private-keys=/tmp/.ssh/id_rsa \
  --config=/config.yml \
  --ask-become-pass
```

> _ SSH

Kubernetes

Deckhouse
ФЛАНТ

master-0

worker-0

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

The DKP controller installs the necessary modules.

Kubernetes

Deckhouse
ФЛАНТ

master-0

worker-0

At this stage the single-master cluster is ready.
If you have additional nodes, let’s join them...

Kubernetes

Deckhouse
ФЛАНТ

master-0

worker-0

YAML

ng.yml

```yaml
apiVersion: deckhouse.io/v1
kind: NodeGroup
metadata:
  name: worker
...
```

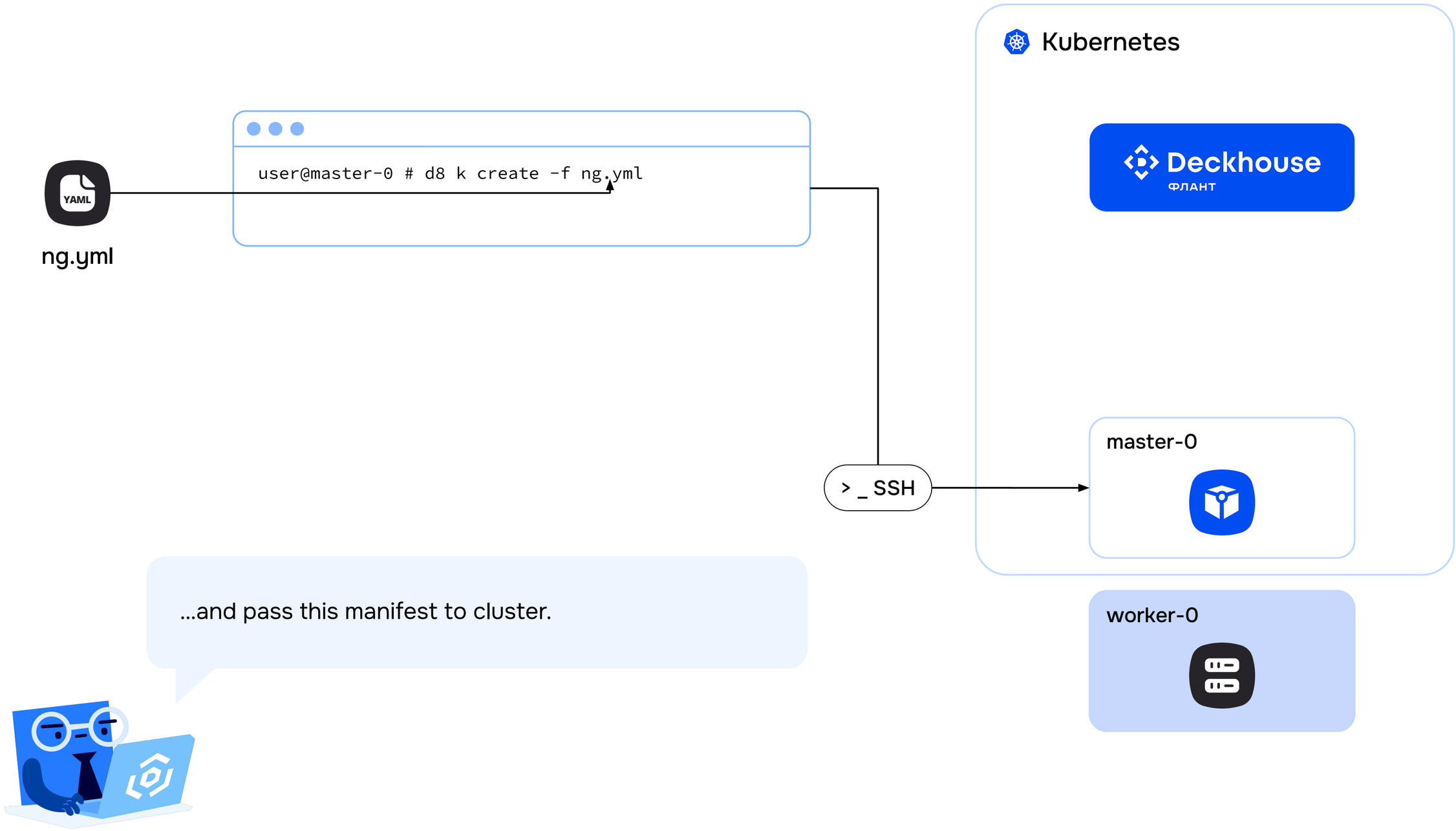
...to do this you will need to prepare a NodeGroup configuration...
The «Node management» section of documentation will help you create the manifest.

Kubernetes

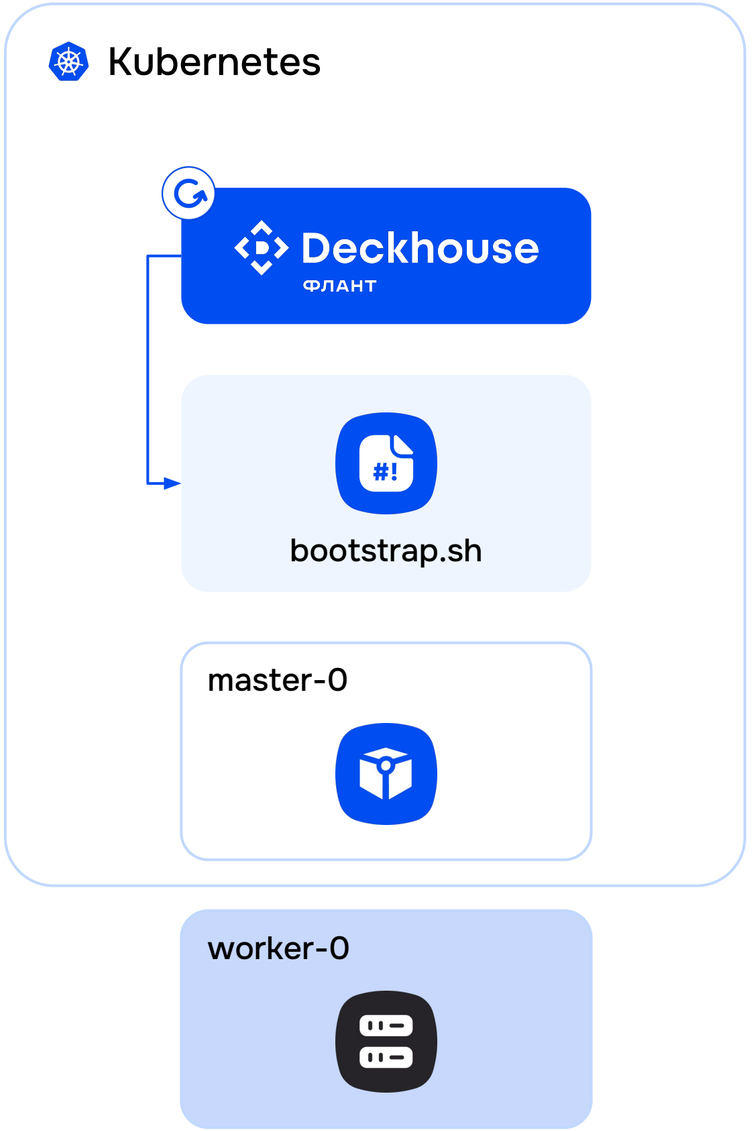
Deckhouse
ФЛАНТ

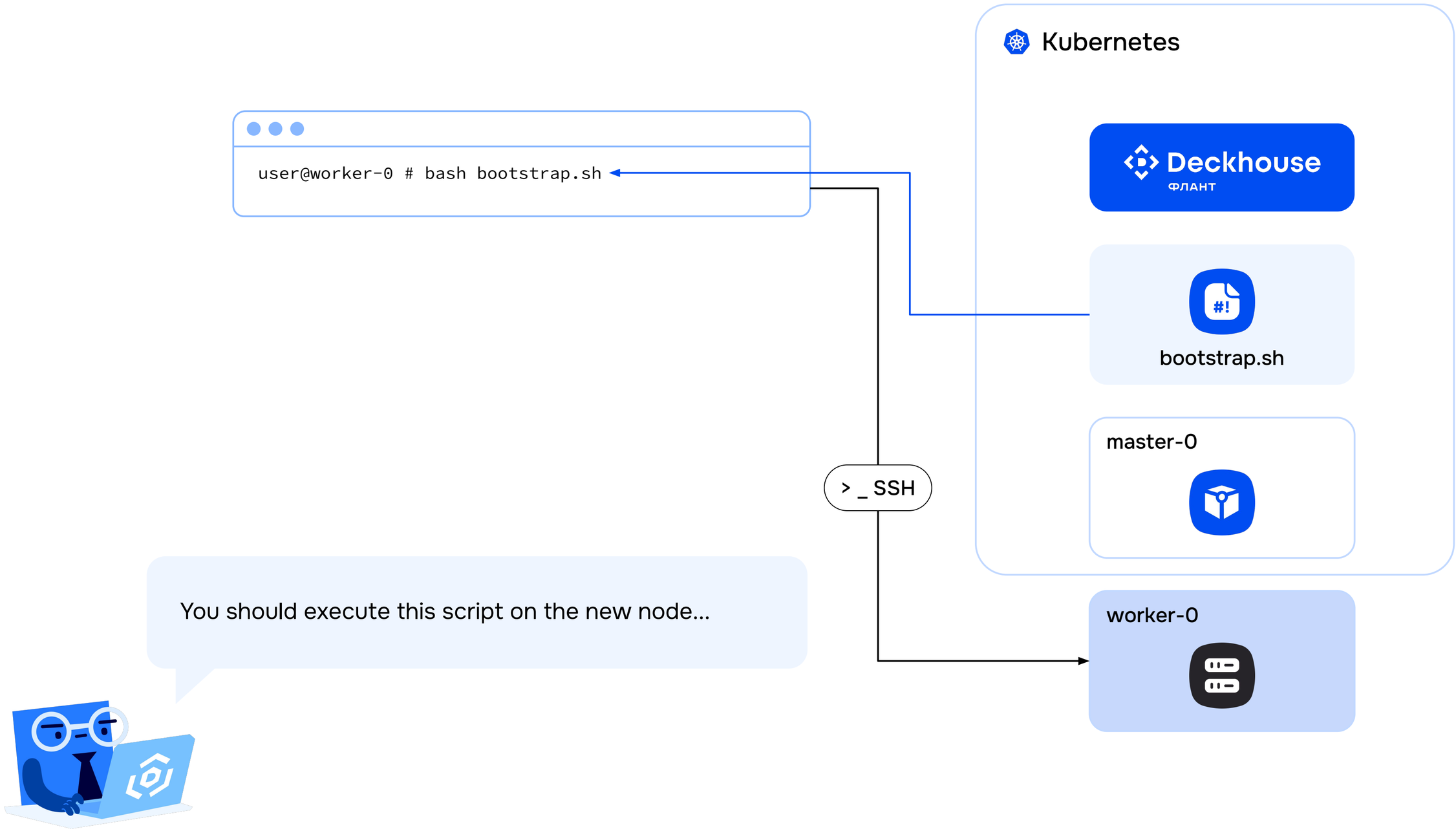
master-0

worker-0


Kubernetes
Deckhouse
ФЛАНТ
YAML
ng.yml
user@master-0 # d8 k create -f ng.yml
> _ SSH
master-0
worker-0
...and pass this manifest to cluster.

The DKP controller reacts to the creation of the NodeGroupand compiles the shell script for manual joining nodes to the cluster.




Kubernetes
user@worker-0 # bash bootstrap.sh
Deckhouse
ФЛАНТ
#!
bootstrap.sh
master-0
> _ SSH
worker-0
You should execute this script on the new node...

...and node will become the part of the cluster.

Kubernetes

Deckhouse
ФЛАНТ

bootstrap.sh

master-0

worker-0

YAML

Ingress-nginx-controller.yml

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

Kubernetes

Deckhouse
ФЛАНТ

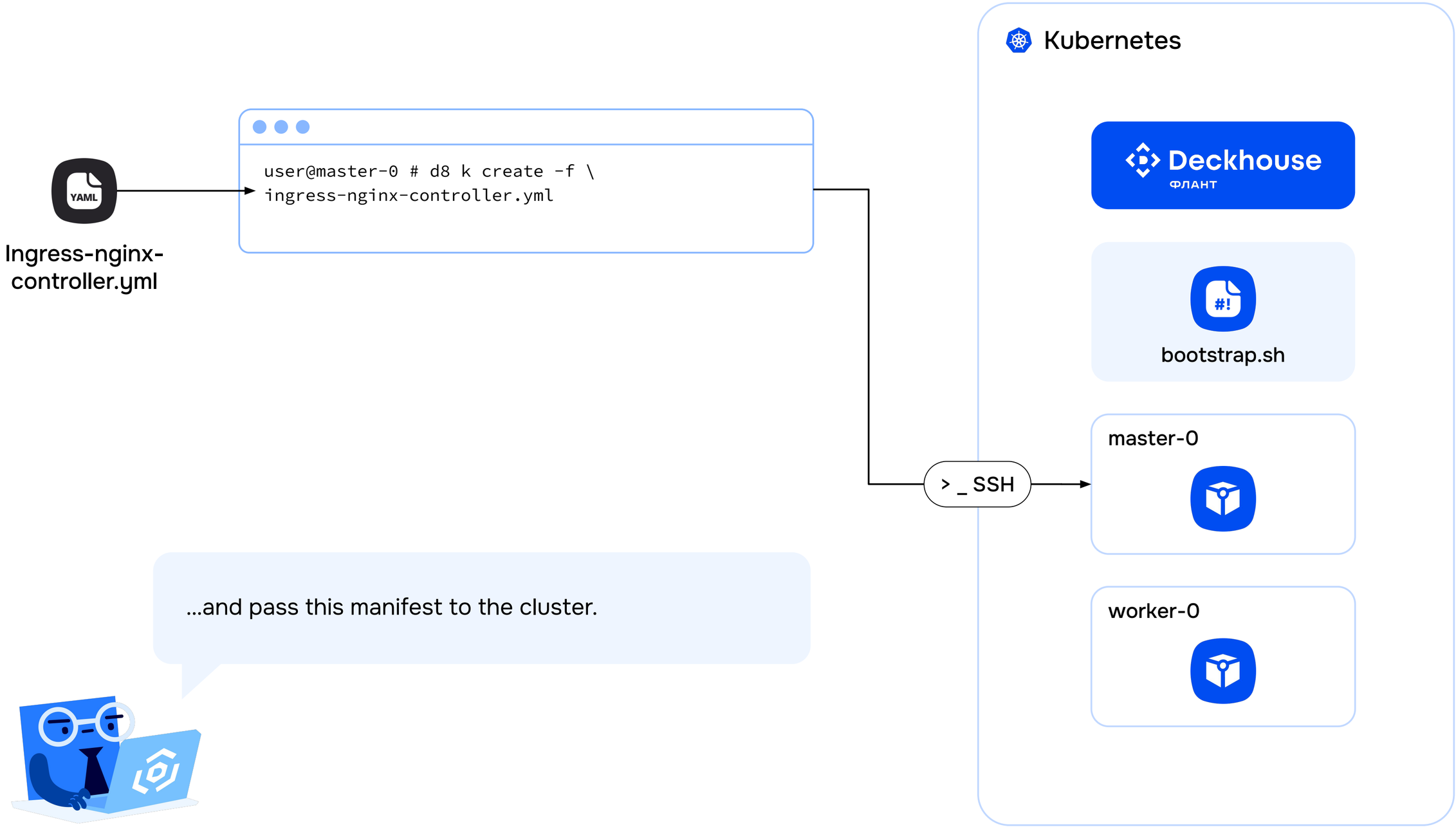
#!

bootstrap.sh

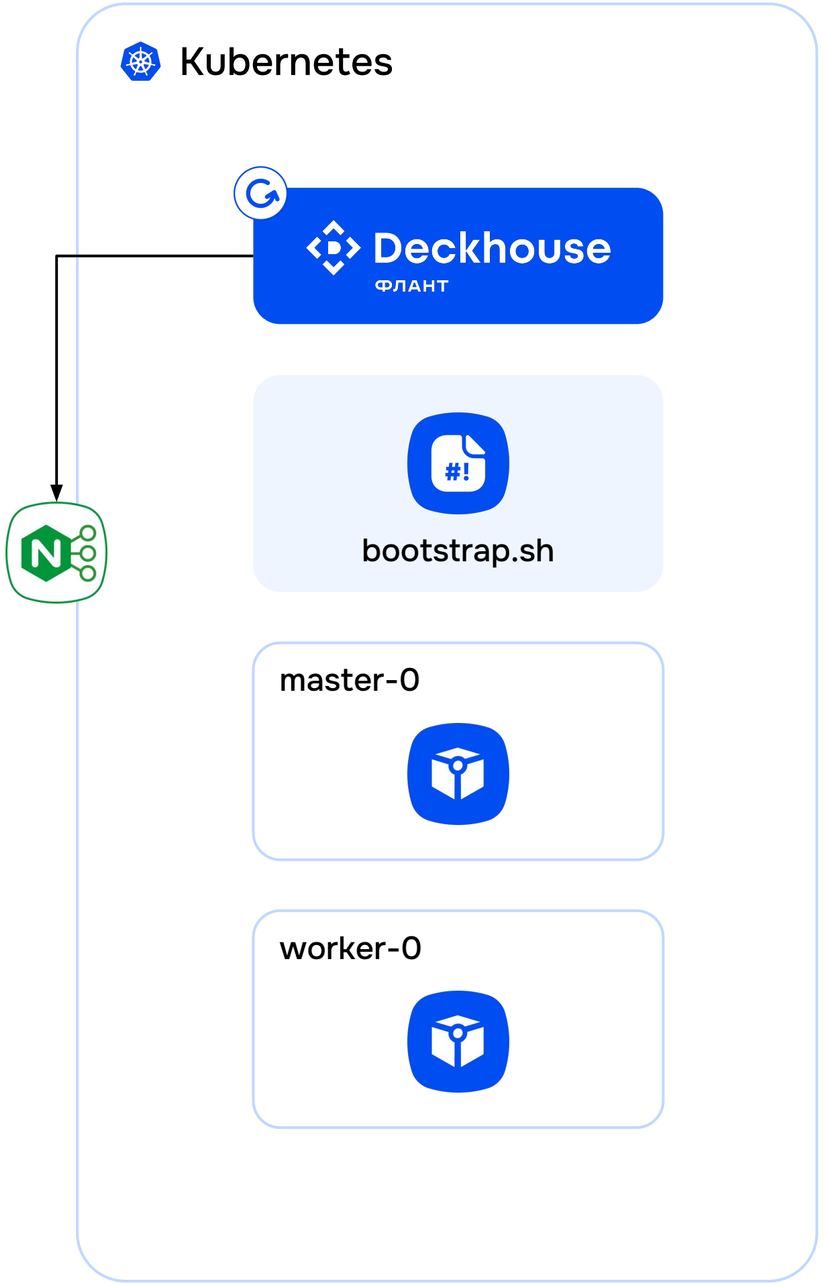
master-0

worker-0

Now we should organize the Ingress controller.
To do this, we make IngressNginxController manifest...


Kubernetes
Deckhouse
ФЛАНТ
bootstrap.sh
master-0
worker-0
YAML
Ingress-nginx-controller.yml
user@master-0 # d8 k create -f \
ingress-nginx-controller.yml
> _ SSH
...and pass this manifest to the cluster.

The DKP controller reacts to the creation of the resource and configures the Ingress controller.



Kubernetes

Deckhouse
ФЛАНТ

bootstrap.sh

master-0

worker-0

The cluster is ready to work!