



Для установки Декхауса вам потребуется ПК и облачная площадка.



`config.yml`



`resources.yml`

Cloud

Установка осуществляется согласно двум конфигурационным файлам.
Дальнейшие шаги руководства Getting Started помогут их правильно составить ...

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



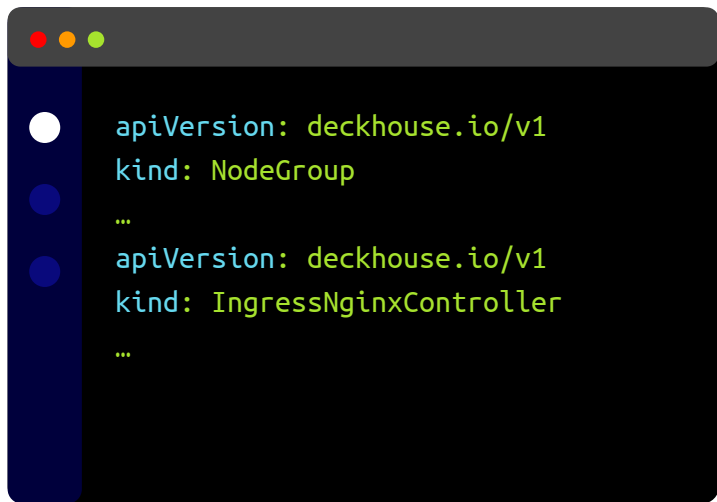
config.yml



resources.yml



... в первом файле приведены настройки доступа к облачному API, описаны параметры мастер-узлов, а также начальные настройки контроллера Deckhouse ...



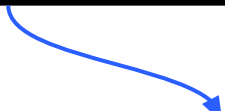
```
apiVersion: deckhouse.io/v1
kind: NodeGroup
...
apiVersion: deckhouse.io/v1
kind: IngressNginxController
...
```



config.yml



resources.yml



Cloud



... во втором приведены дополнительные Kubernetes-ресурсы, которые следует создать при инициализации. Они включают в себя спецификации для рабочих узлов, параметры Ingress-контроллера и прочие манифесты.

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



config.yml



resources.yml

Полученные конфигурационные файлы
передаём утилите dhctl, которая начнет установку.

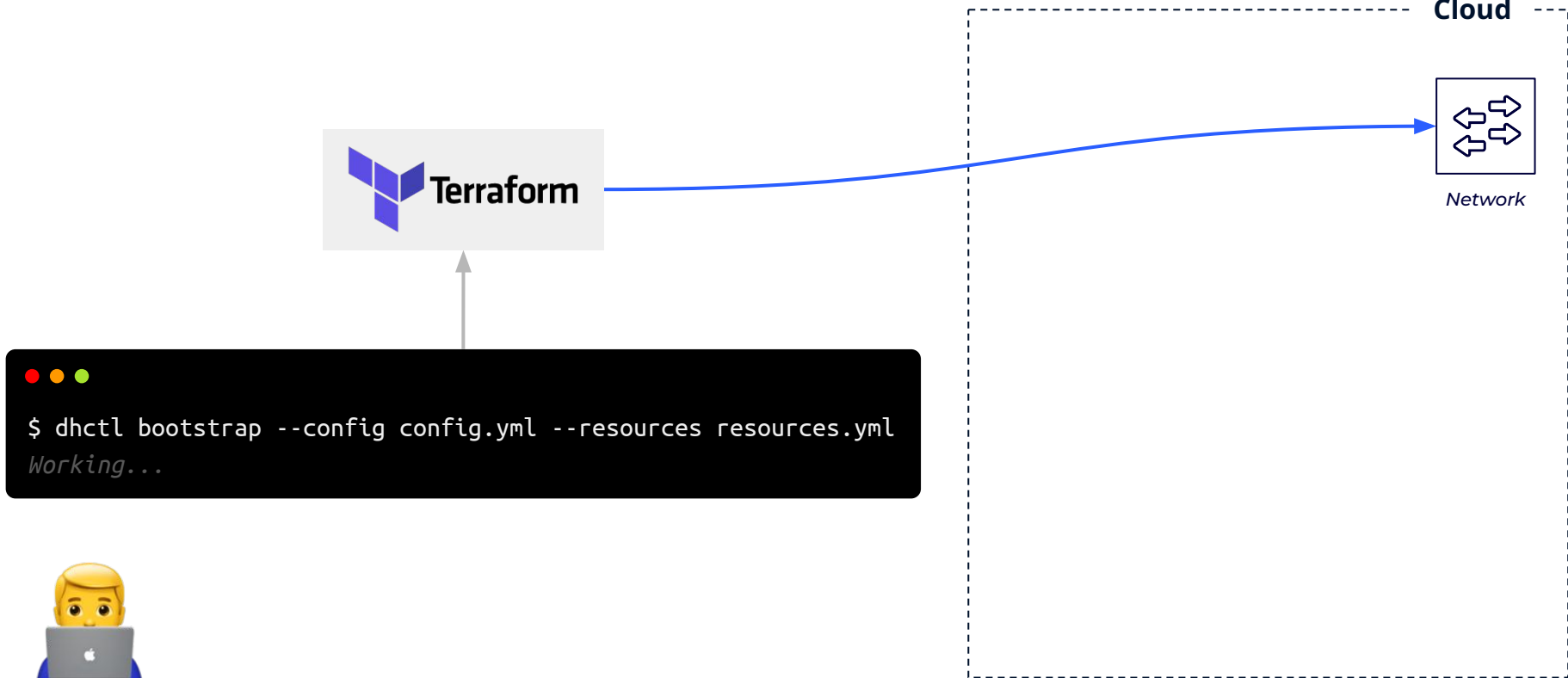


```
$ dhctl bootstrap --config config.yml --resources resources.yml  
Working...
```

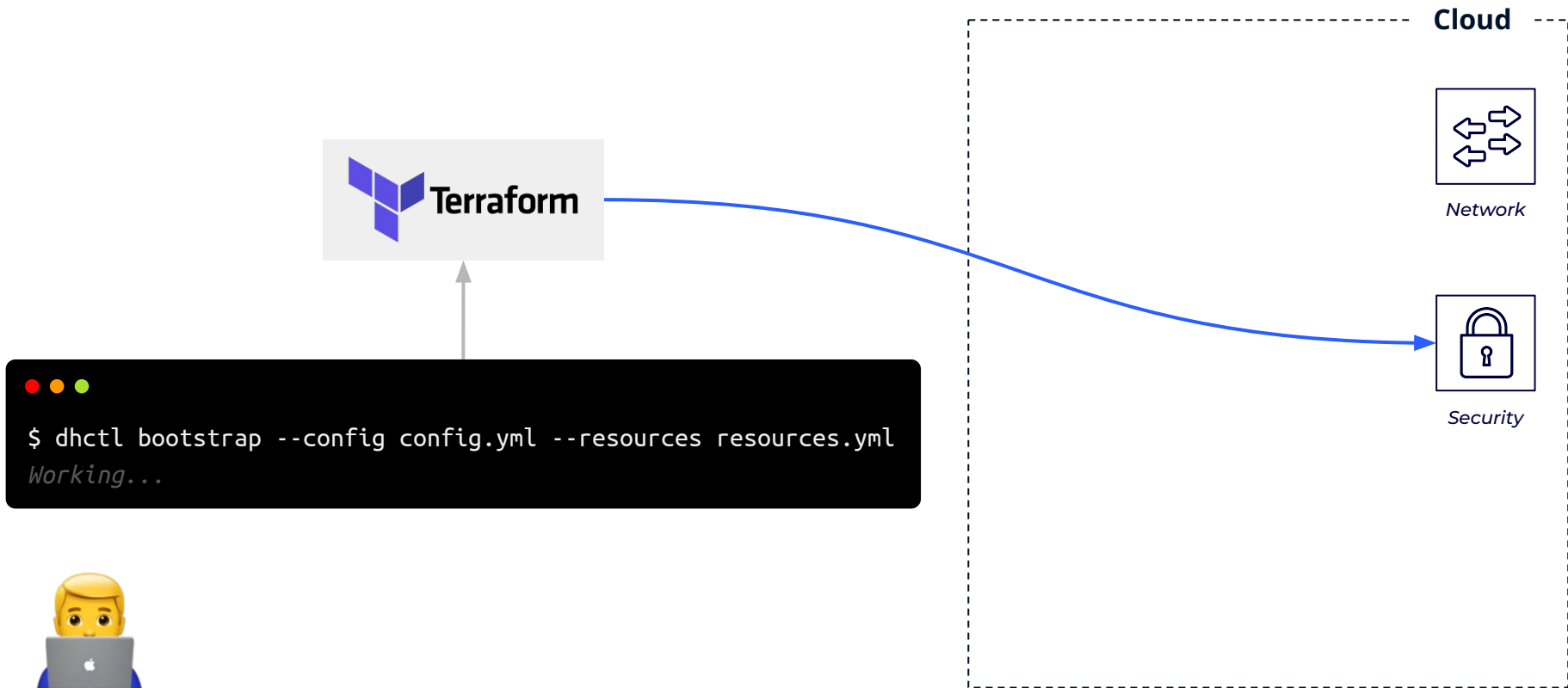


Cloud

С помощью встроенного Terraform dhctl разворачивает базовую облачную инфраструктуру ...



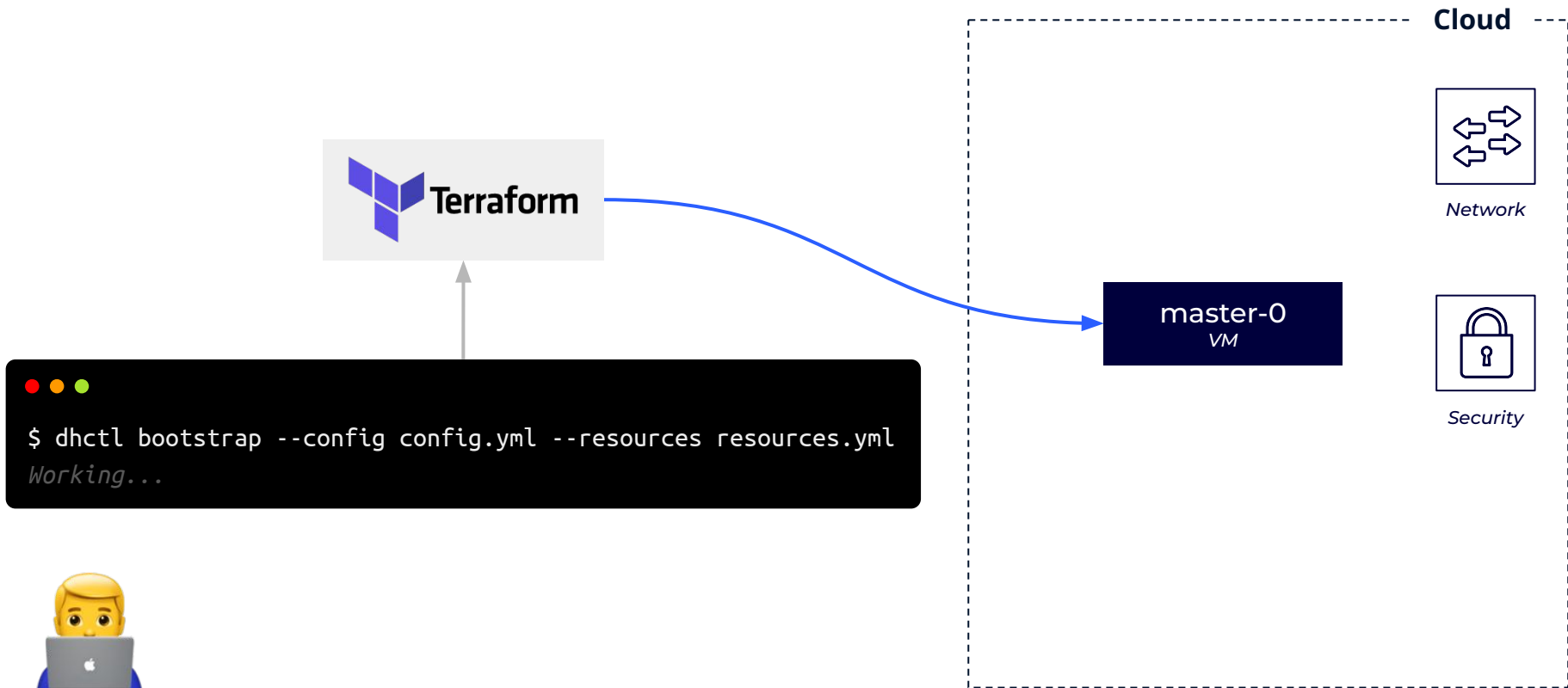
... настраивает сеть ...



... решает базовые вопросы безопасности и прочее.

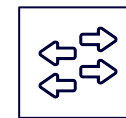
Базовая инфраструктура имеет различную природу в зависимости от типа облака и выбранной схемы размещения.

Схемы размещения будут описаны в следующих шагах руководства Getting Started.



С помощью Terraform dhctl создаёт виртуальную машину (или несколько)
под будущий мастер-узел Kubernetes.

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



Network

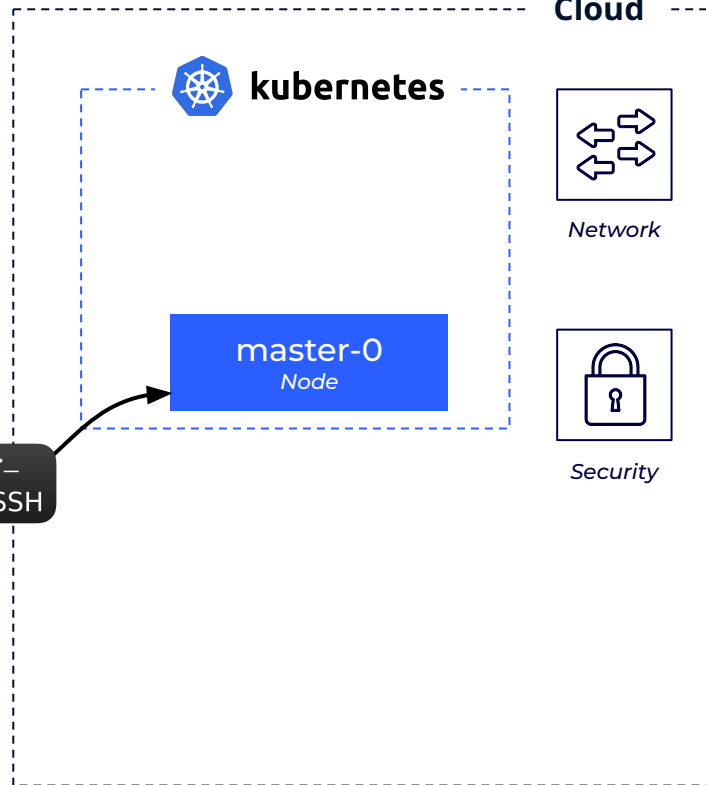


Security

Cloud

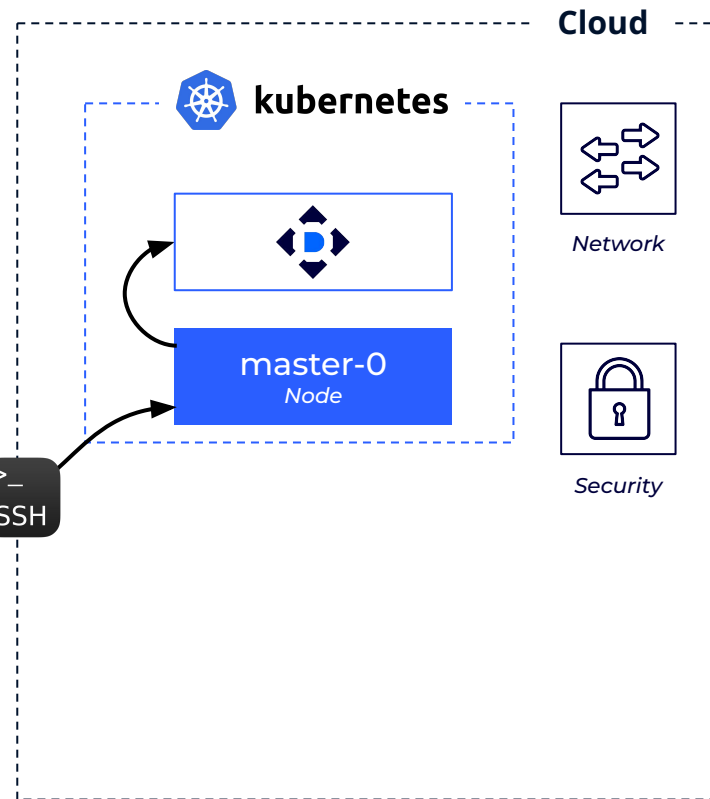
Далее подключается к данной VM через SSH ...

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



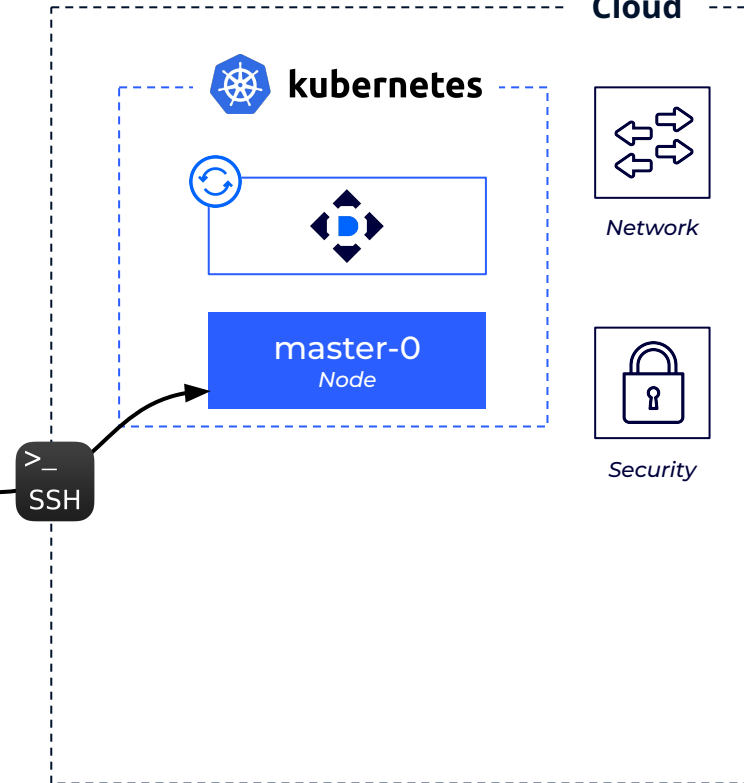
... и инициализирует кластер Kubernetes.
На данном этапе готов минимальный vanilla-кластер Kubernetes.

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



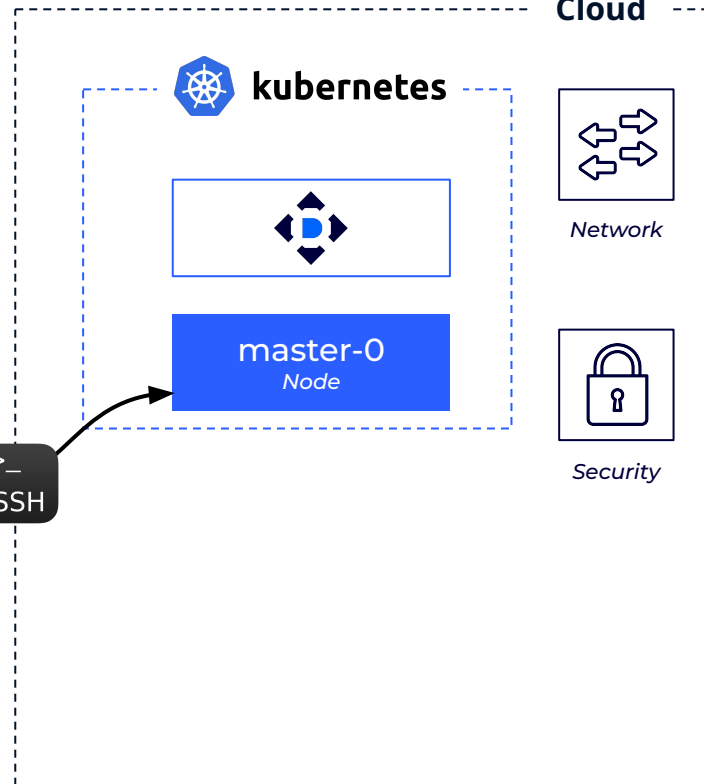
Для завершения инсталляции утилита устанавливает в кластер контроллер Deckhouse.

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



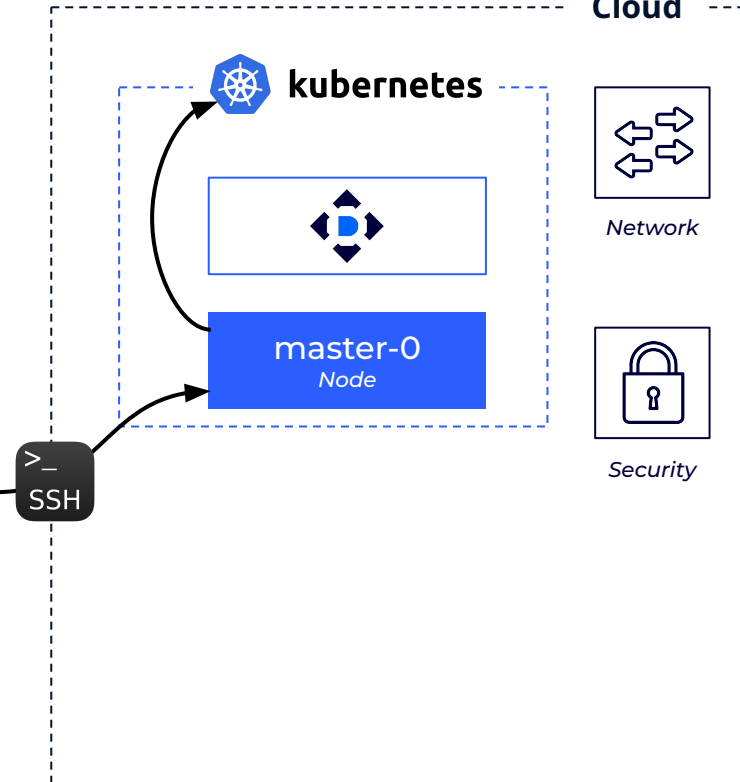
Контроллер Deckhouse при первом запуске устанавливает необходимые модули и устанавливает связь с API облака.

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

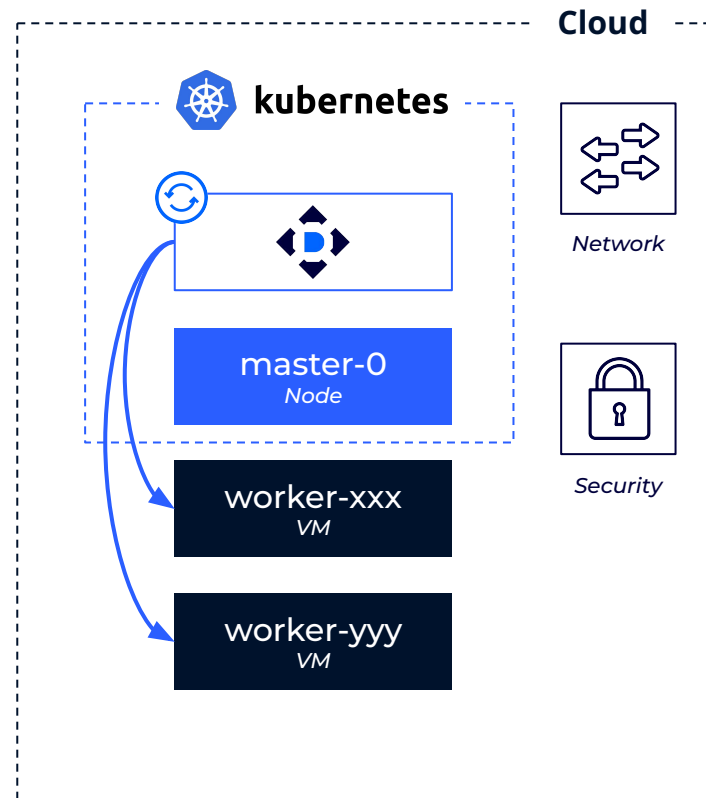


Базовый кластер готов.

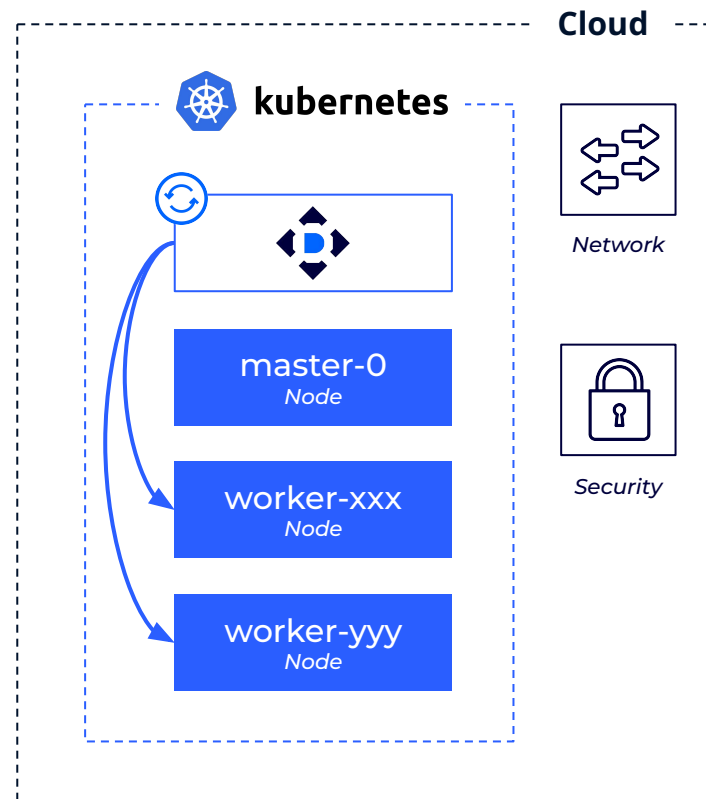
```
$ dhctl bootstrap --config config.yml --resources resources.yml
Working...
```



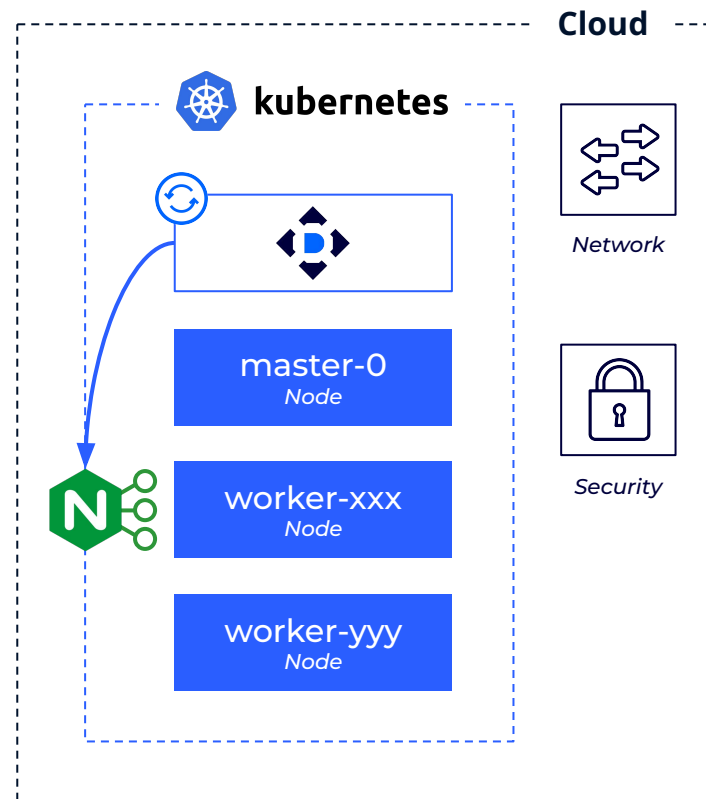
Осталось применить дополнительные Kubernetes-ресурсы из resources.yml.



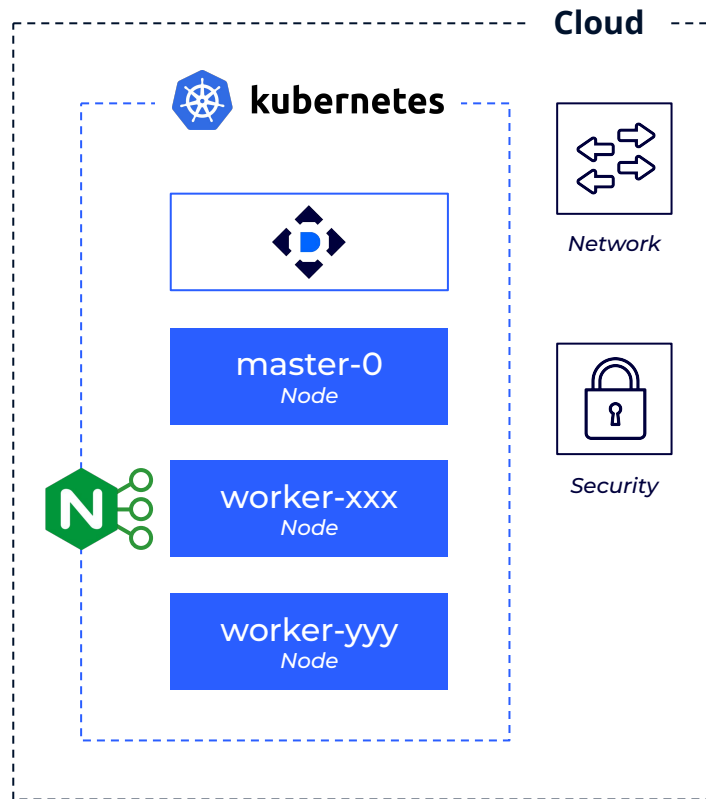
Контроллер Deckhouse реагирует на появление ресурсов
и создаёт заданный набор узлов ...



... а затем добавляет их в кластер.



Помимо узлов, согласно настройкам из `resources.yml` контроллер Deckhouse настраивает Ingress-контроллер.



Кластер готов к работе!