

# L2 LoadBalancer

Принцип работы

В кластере 4 узла — 3 фронтенд-узла  
и 1 воркер-узел.

front-0



nginx-0

**node-role:**  
front

front-1



nginx-1

**node-role:**  
front

front-2



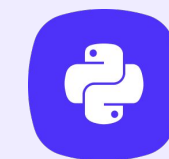
nginx-2

**node-role:**  
front

worker-0



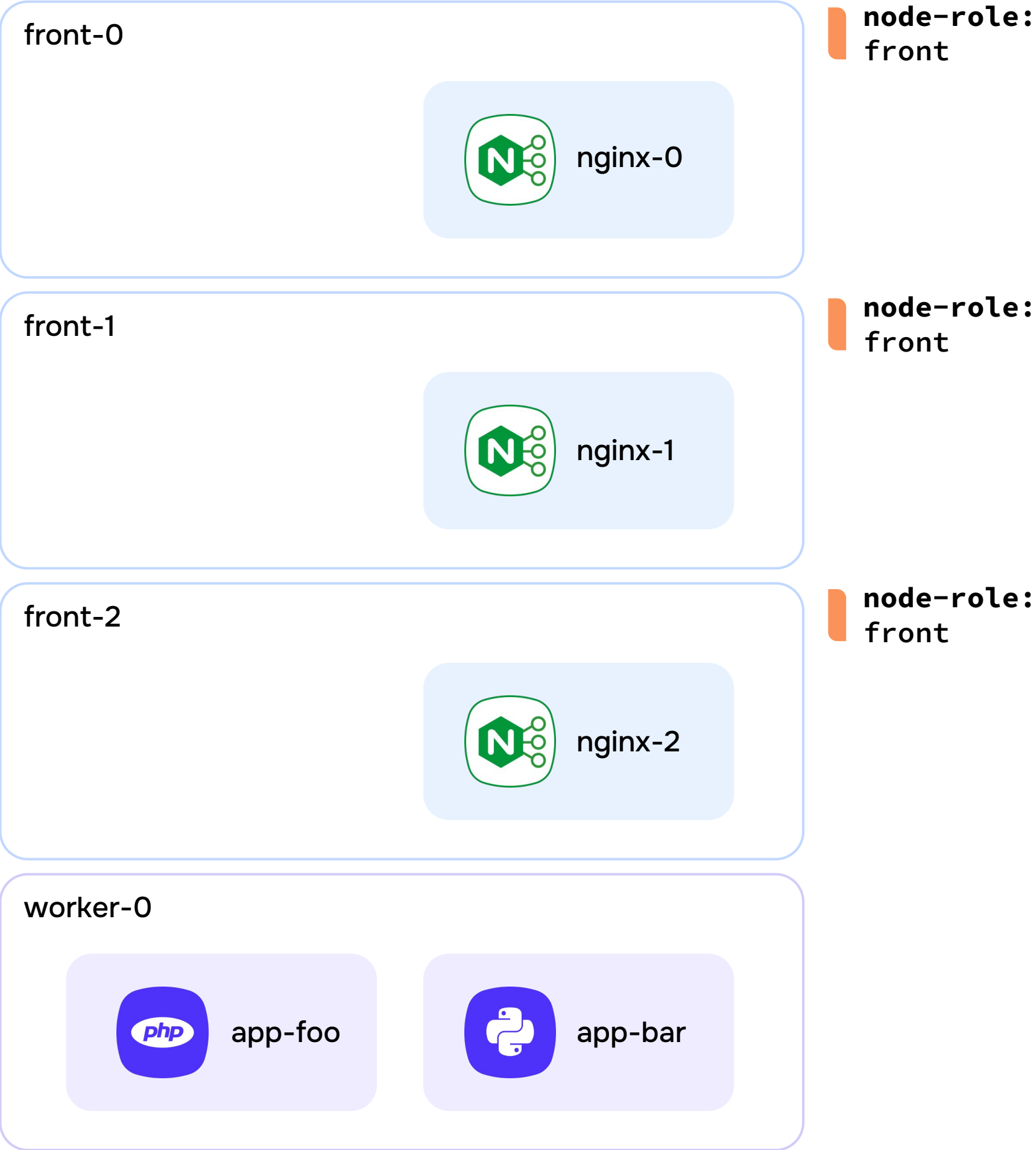
app-foo



app-bar

```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: l2-load-balancer
spec:
  enabled: true
  version: 2
```

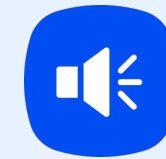
Включен модуль **MetalLB**: ModuleConfig версия 2.



```
apiVersion: network.deckhouse.io/v1alpha1
kind: MetalLoadBalancerClass
metadata:
  name: front
spec:
  addressPool:
    - 192.168.122.100-192.168.122.150
  isDefault: false
  nodeSelector:
    node-role: front
  type: L2
```

Создан ресурс **MetalLoadBalancerClass** с указанием фронтенд-узлов и пула «публичных» IP-адресов. Таким образом легко создавать «зоны», привязывая определенные пулы адресов к группе узлов. На всех фронтенд-узлах запускаются спикеры.

front-0



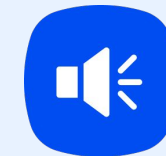
speaker



nginx-0

**node-role:**  
front

front-1



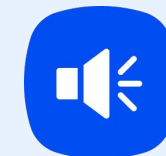
speaker



nginx-1

**node-role:**  
front

front-2



speaker



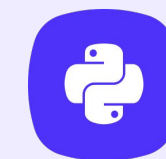
nginx-2

**node-role:**  
front

worker-0



app-foo

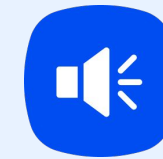


app-bar

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-deployment
  annotations:
    network.deckhouse.io/l2-load-balancer-name:
ingress
    network.deckhouse.io/l2-load-balancer-external-
ips-count: "3"
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
  LoadBalancerClass: front
```

Создан ресурс **Service** с типом **LoadBalancer**.  
В нем указаны имя LoadBalancerClass, специальная аннотации с необходимым количеством IP-адресов. Каждый фронтенд-узел получает один и более адрес из пула.

front-0



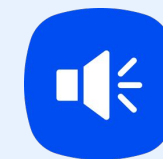
speaker  
192.168.122.100



nginx-0

node-role:  
front

front-1



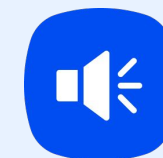
speaker  
192.168.122.101



nginx-1

node-role:  
front

front-2



speaker  
192.168.122.102



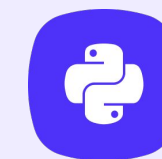
nginx-2

node-role:  
front


worker-0



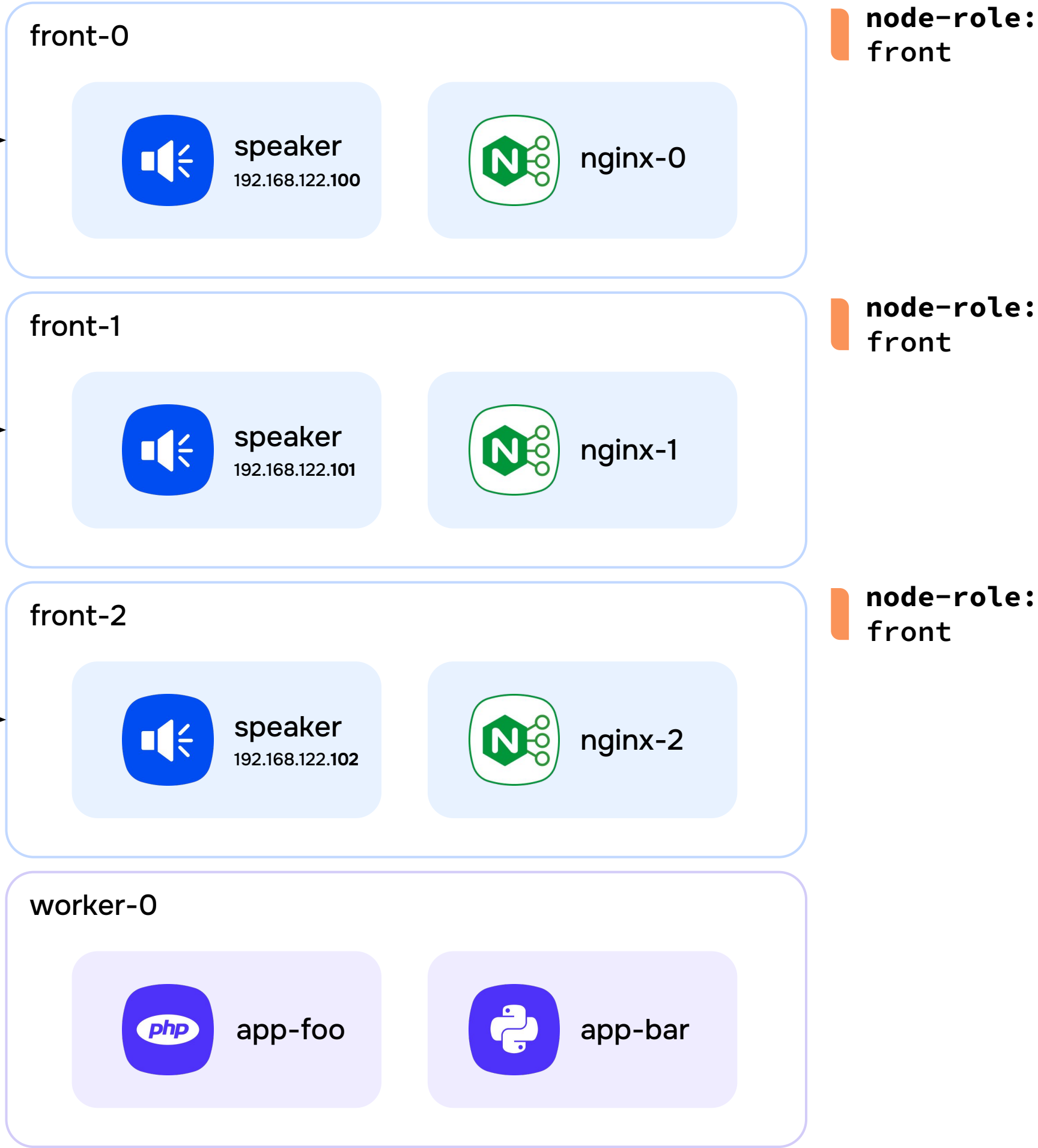
app-foo



app-bar




**app.example.com**  
A 192.168.122.100  
A 192.168.122.101  
A 192.168.122.102

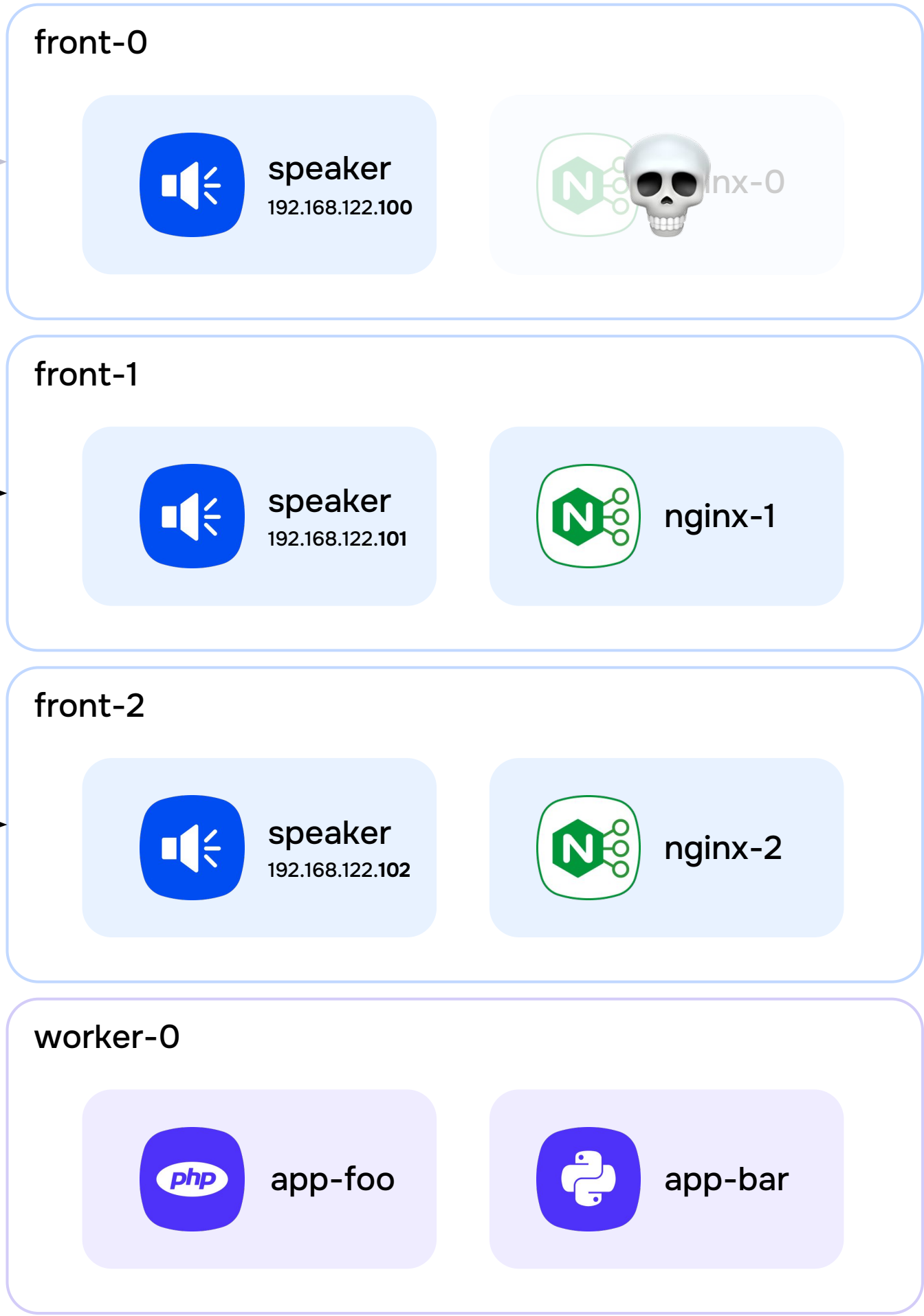


Каждый фронтенд-узел участвует в обработке прикладных запросов. Для этого в публичном DNS-имени приложения прописаны три A-записи.





**app.example.com**  
A 192.168.122.100  
A 192.168.122.101  
A 192.168.122.102




node-role:  
front

node-role:  
front

node-role:  
front

В случае выхода из строя приложения nginx на одном из фронтенд-узлов или самого узла, треть запросов обрывается...

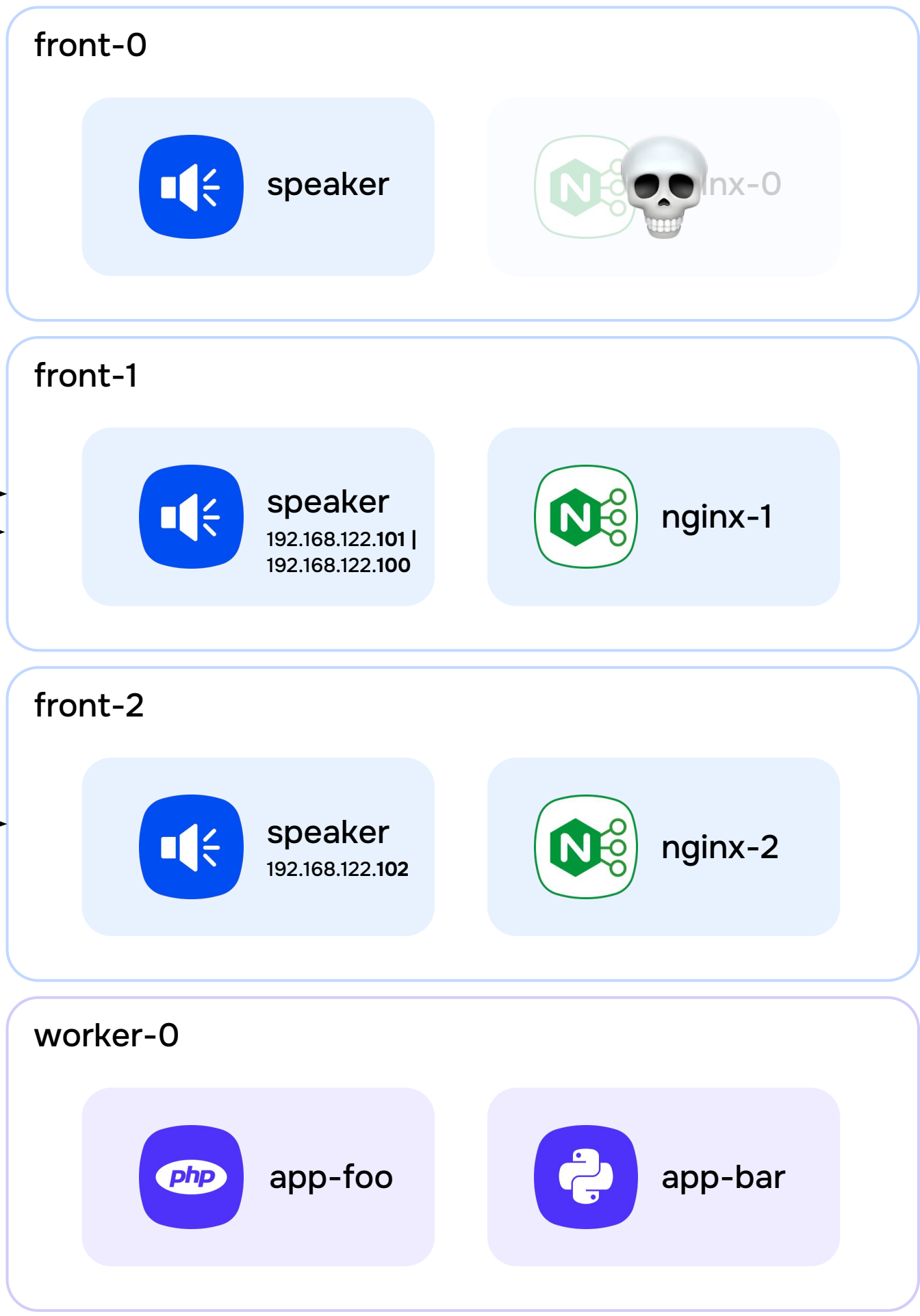


app.example.com

A 192.168.122.100

A 192.168.122.101

A 192.168.122.102



**node-role:**  
front

**node-role:**  
front

**node-role:**  
front

...и один из оставшихся фронтенд-узлов принимает на себя «проблемный» IP-адрес и обрабатывает соответствующие прикладные запросы.