



DECKHOUSE

**Kubernetes
Platform**

L2 LoadBalancer
Принцип работы

L2 LoadBalancer



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

L2 LoadBalancer



```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: metallb
spec:
  enabled: true
  version: 2
```



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

L2 LoadBalancer



```
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**.

L2 LoadBalancer

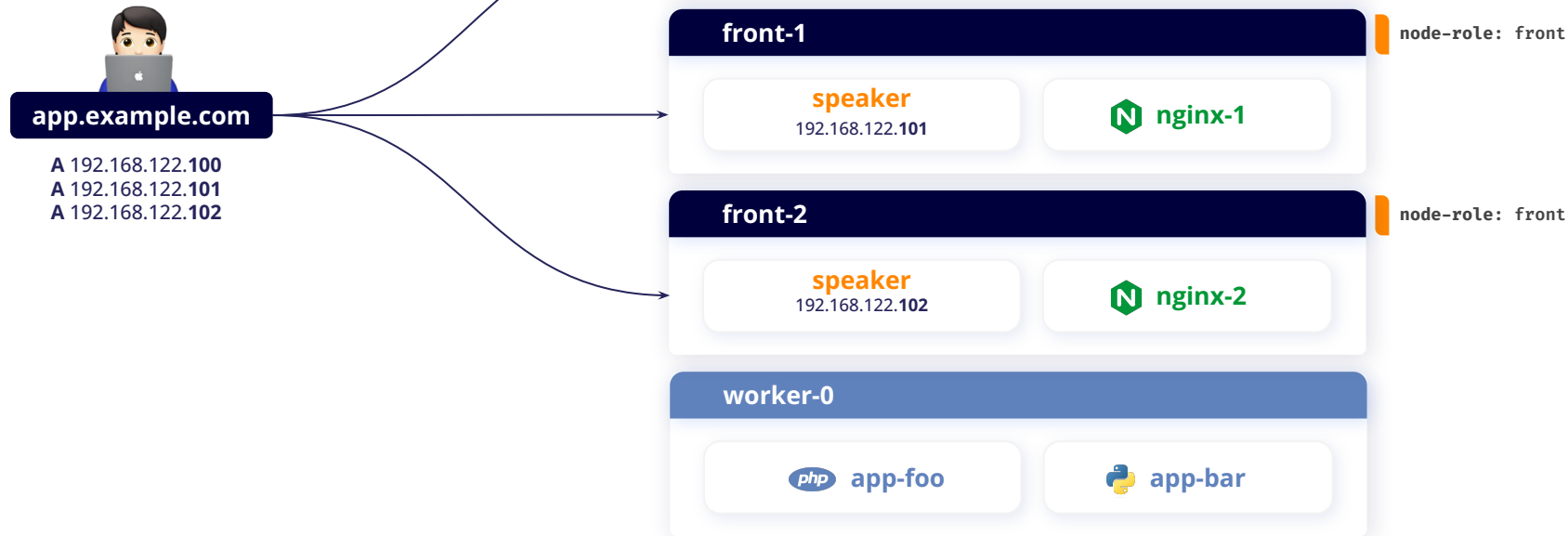


```
apiVersion: v1
kind: Service
metadata:
  name: nginx-deployment
  annotations:
    network.deckhouse.io/l2-load-balancer-external-ips-count: "3"
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
  LoadBalancerClass: front
```



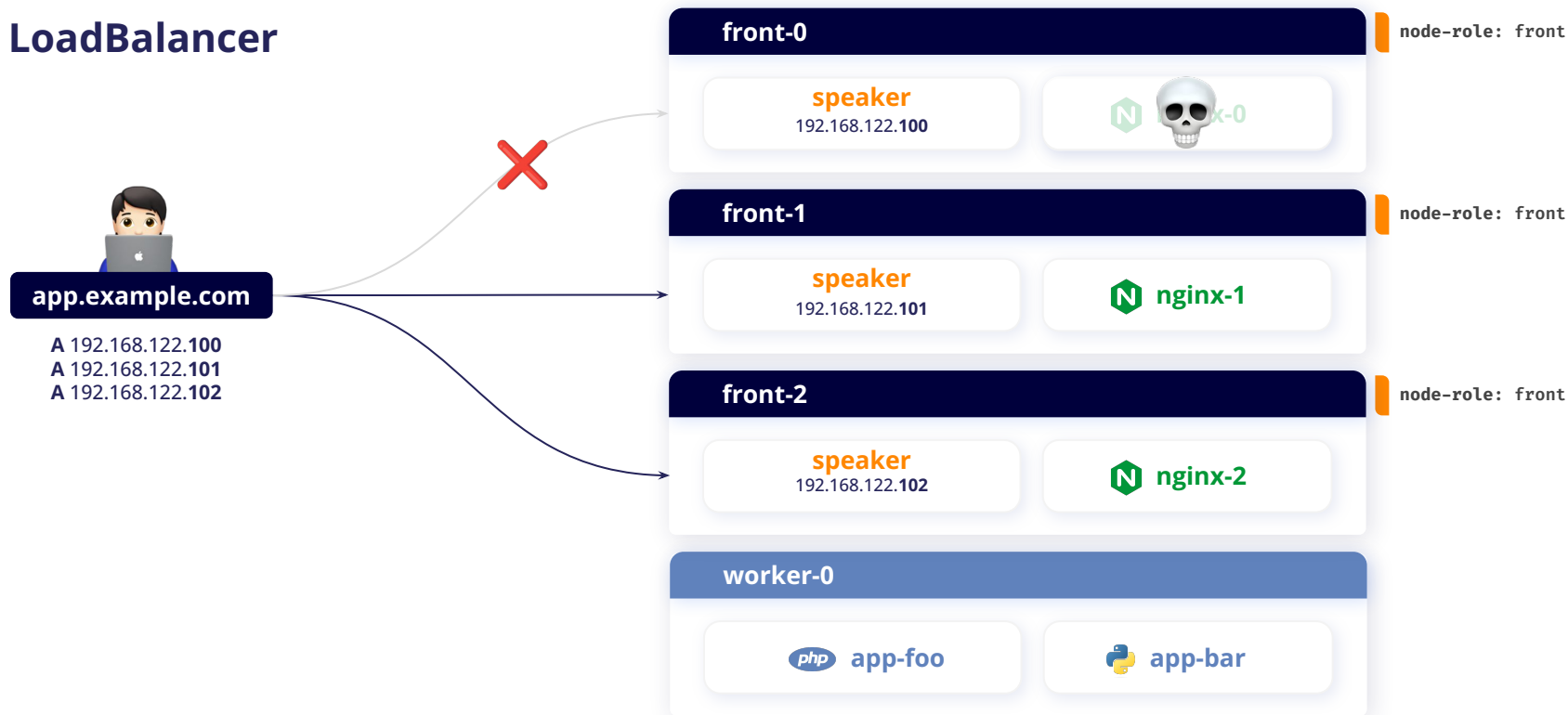
Создан ресурс **Service** с типом **LoadBalancer**. В нем указаны имя **LoadBalancerClass**, специальная аннотации с необходимым количеством IP-адресов.
Каждый фронтенд-узел получает один и более адрес из пула.

L2 LoadBalancer



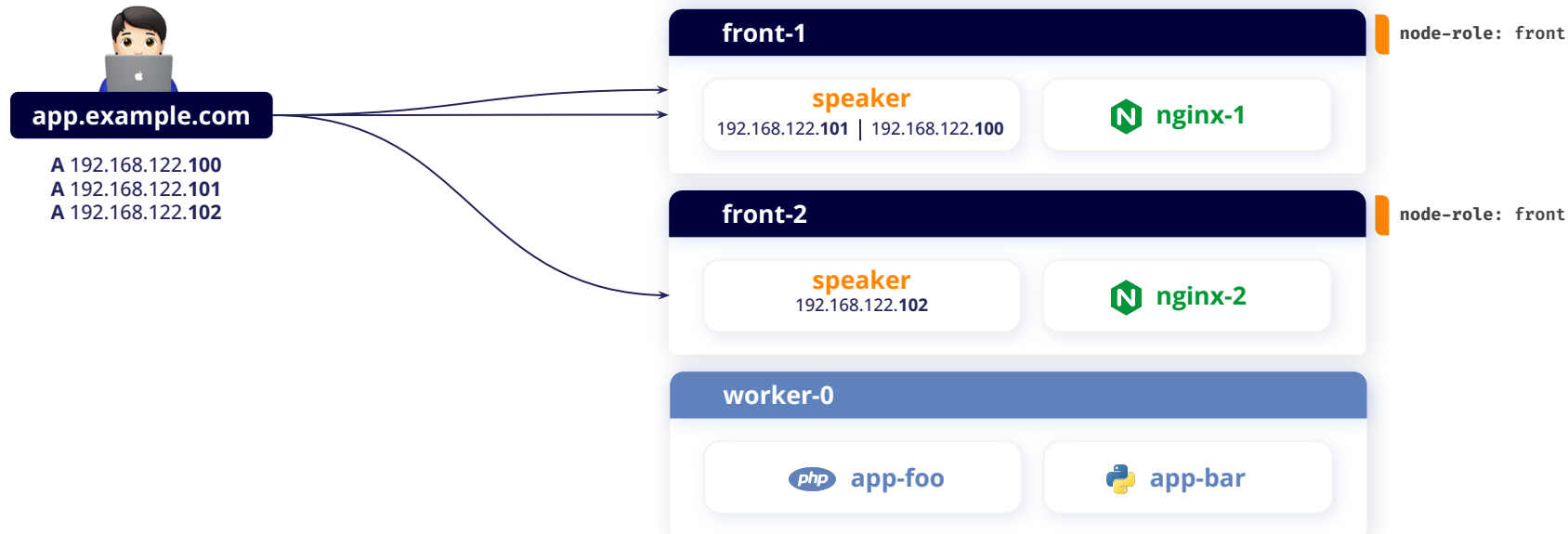
Каждый фронтенд-узел участвует в обработке прикладных запросов. Для этого в публичном DNS-имени приложения прописаны три A-записи.

L2 LoadBalancer



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

L2 LoadBalancer



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