



DECKHOUSE

**Kubernetes  
Platform**

**L2 Load Balancer**  
**Принцип работы**



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



```
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: l2-load-balancer
spec:
  enabled: true
  settings:
    addressPools:
      - addresses:
          - 192.168.122.0/24
            name: mypool
  version: 1
```



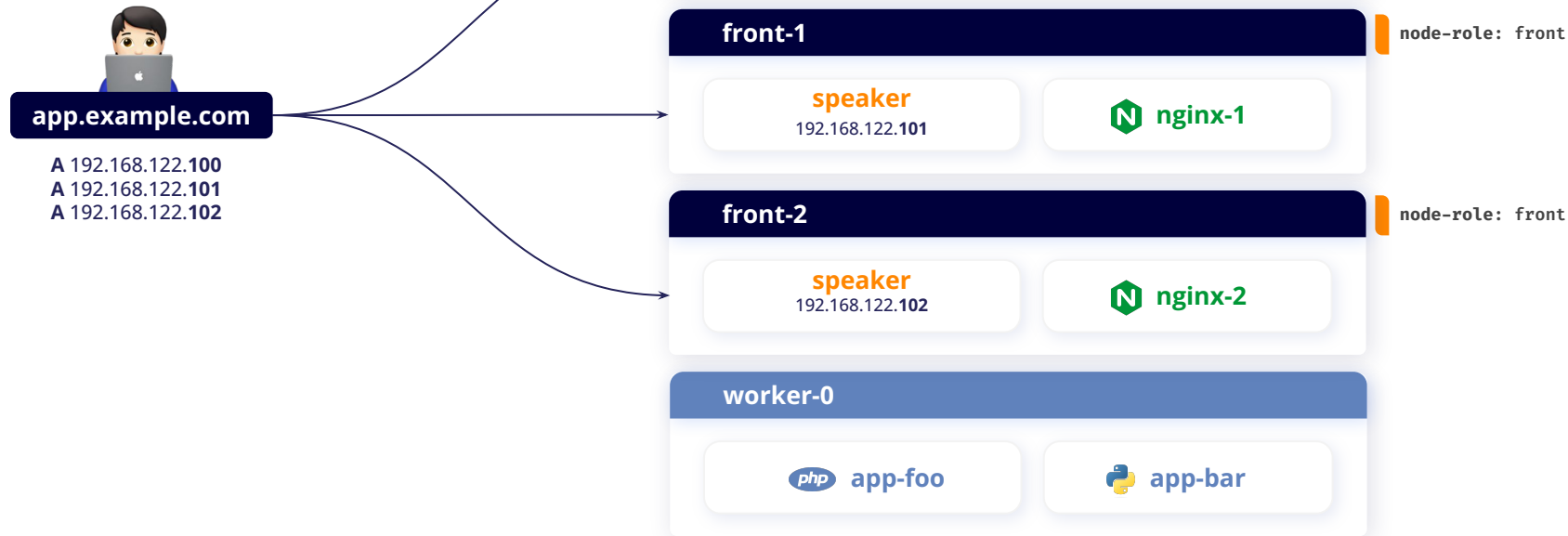
Включен модуль **l2-load-balancer** с определенным пулом адресов.



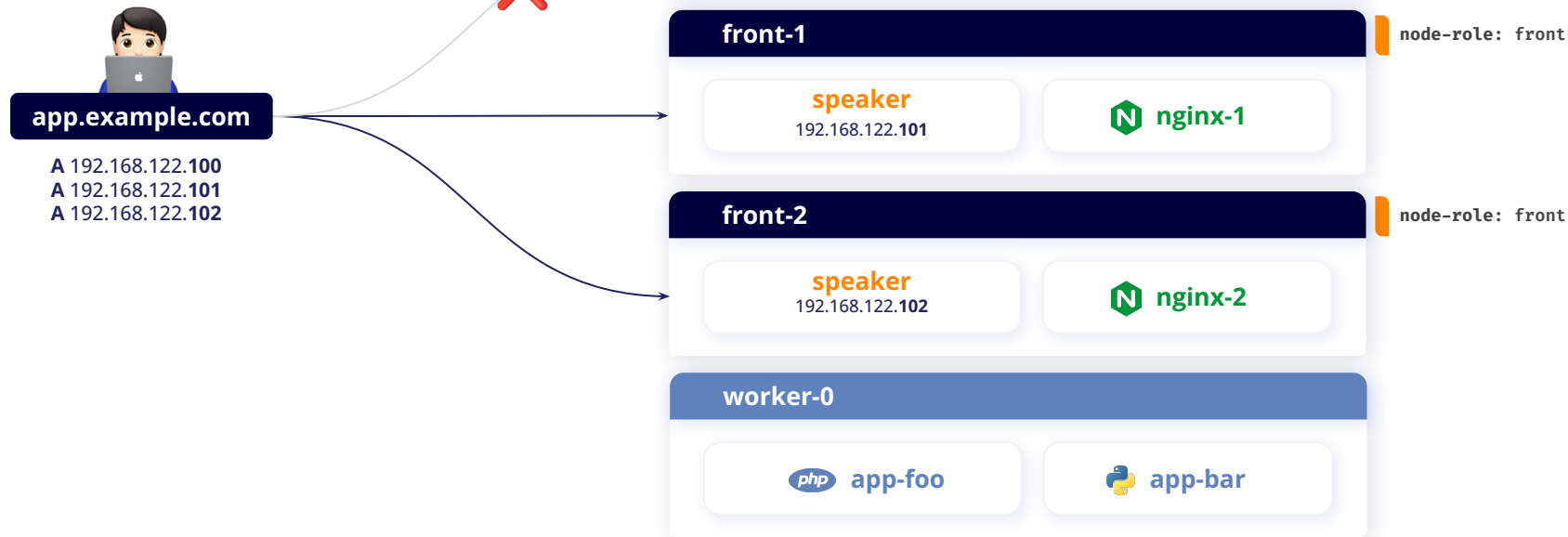
```
apiVersion: deckhouse.io/v1alpha1
kind: L2LoadBalancer
metadata:
  name: front
spec:
  addressPool: mypool
  nodeSelector:
    node-role: front
  service:
    selector:
      app: nginx
    ports:
      - name: http
        port: 80
        protocol: TCP
```



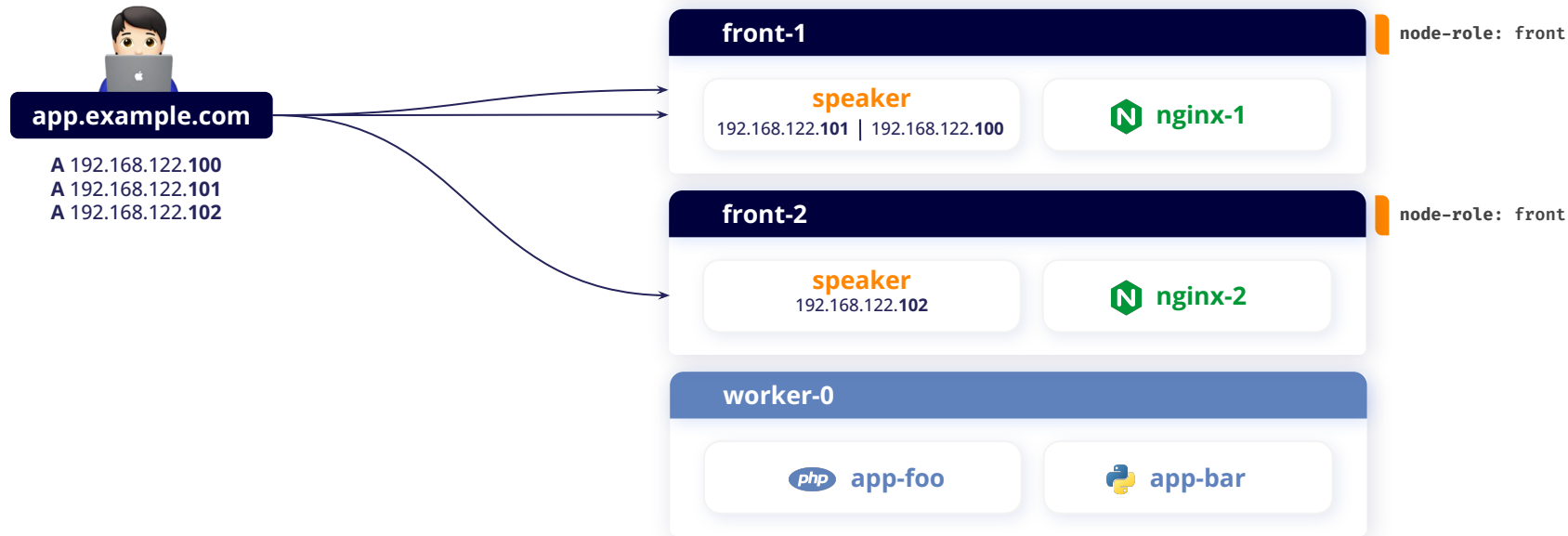
Создан ресурс **L2LoadBalancer** с указанием фронтенд-узлов, пула “публичных” IP-адресов и настройками сервиса для приложения nginx. На всех фронтенд-узлах запускаются **спикеры**, каждый из которых получает отдельный адрес из пула.



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



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




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

[REDACTED]


[REDACTED]



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




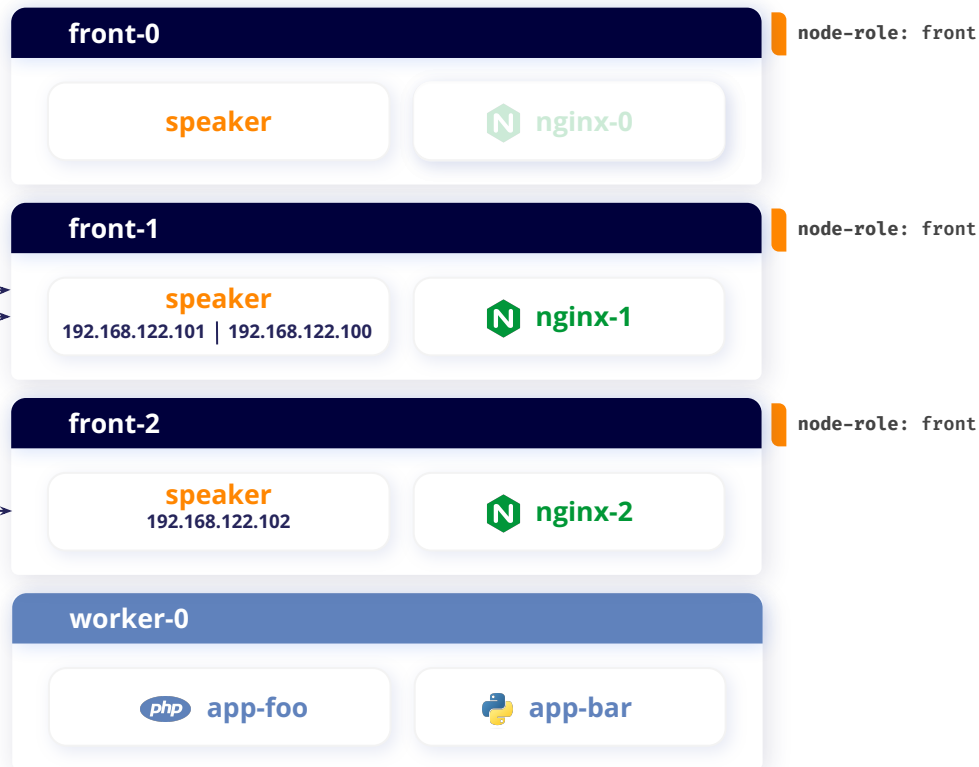
Каждый фронтенд-узел участвует в обработке прикладных запросов. Для этого в публичном DNS-имени приложения прописаны три A-записи.

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

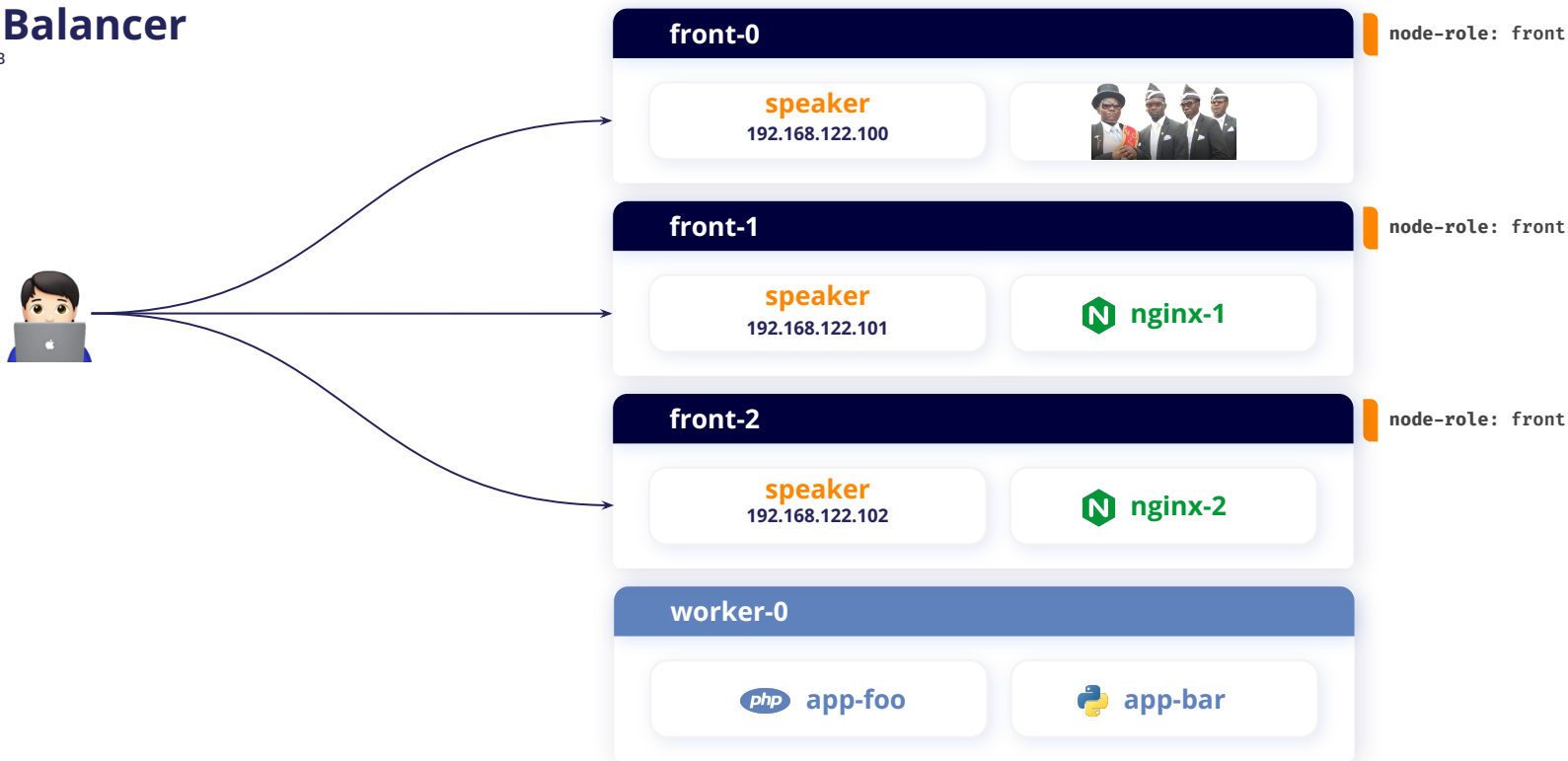


В случае выхода из строя приложения nginx на одном из фронтенд-узлов или самого узла, треть запросов обрывается...

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



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



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



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

# L2LoadBalancer

powered by MetalLB

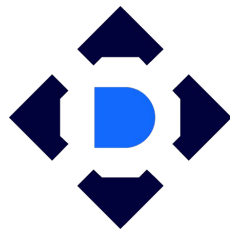


```
apiVersion: deckhouse.io/v1alpha1
kind: L2LoadBalancer
metadata:
  name: front
spec:
  addressPool: mypool
  nodeSelector:
    node-role: front
  service:
    selector:
      app: nginx
    ports:
      - name: http
        port: 80
        protocol: TCP
```



Создан ресурс L2LoadBalancer с указанием фронтенд-узлов, пула “публичных” IP-адресов и настройками сервиса для приложения nginx. На всех фронтенд-узлах запускаются speaker’ы, каждый из которых получает отдельный адрес из пула.

L2 Load Balancer.  
Принцип работы.



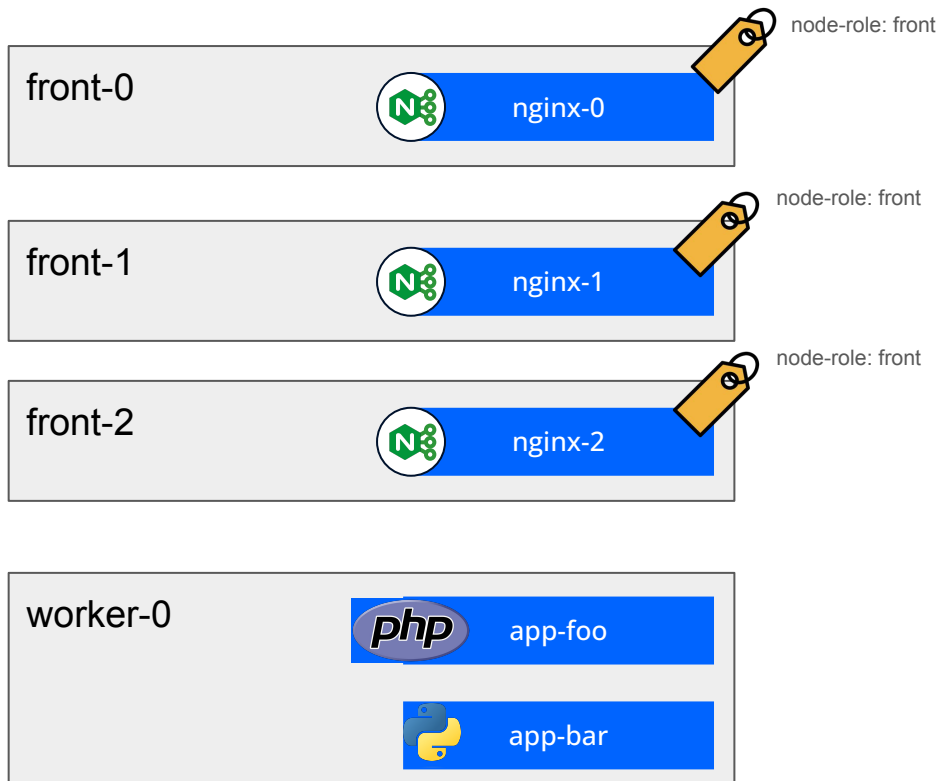
FLANT

**Deckhouse**  
**Kubernetes Platform**



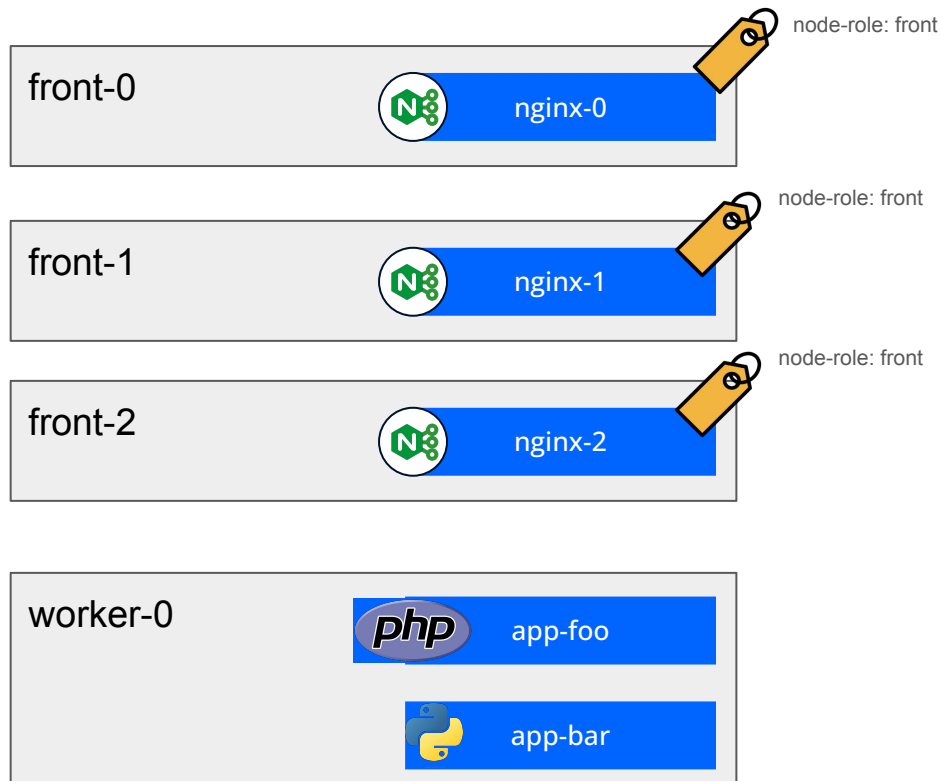
# L2LoadBalancer

powered by MetalLB



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





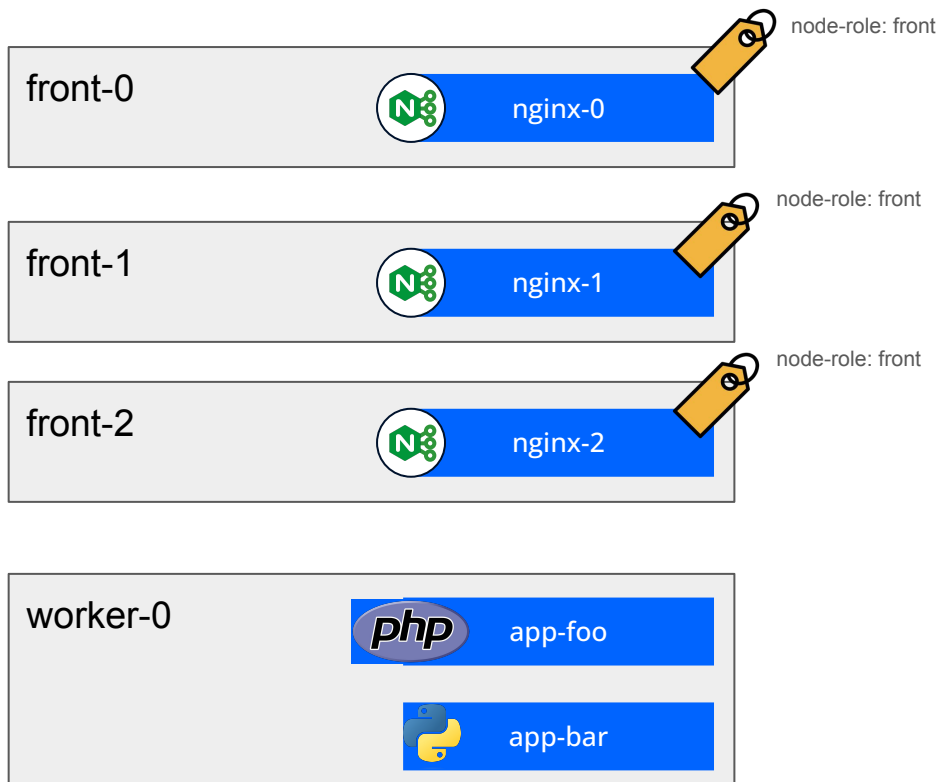
Включаем модуль L2 Load Balancer и определяем в его настройках пул адресов



# L2LoadBalancer

powered by MetalLB

```
apiVersion: deckhouse.io/v1alpha1
kind: L2LoadBalancer
metadata:
  name: front
spec:
  addressPool: mypool
  nodeSelector:
    node-role: front
  service:
    selector:
      app: nginx
  ports:
    - name: http
      port: 80
      protocol: TCP
```



Создан ресурс L2LoadBalancer с указанием фронтенд-узлов, пула “публичных” IP-адресов и настройками сервиса для приложения nginx.

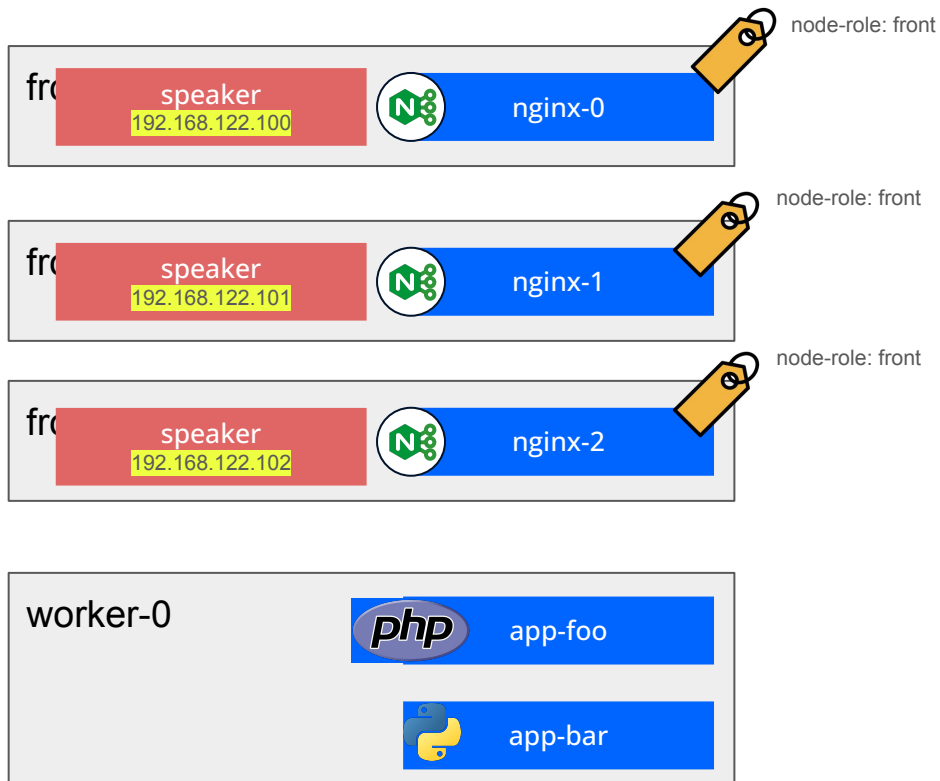


# L2LoadBalancer

powered by MetalLB



```
apiVersion: deckhouse.io/v1alpha1
kind: L2LoadBalancer
metadata:
  name: front
spec:
  addressPool: mypool
  nodeSelector:
    node-role: front
  service:
    selector:
      app: nginx
  ports:
    - name: http
      port: 80
      protocol: TCP
```

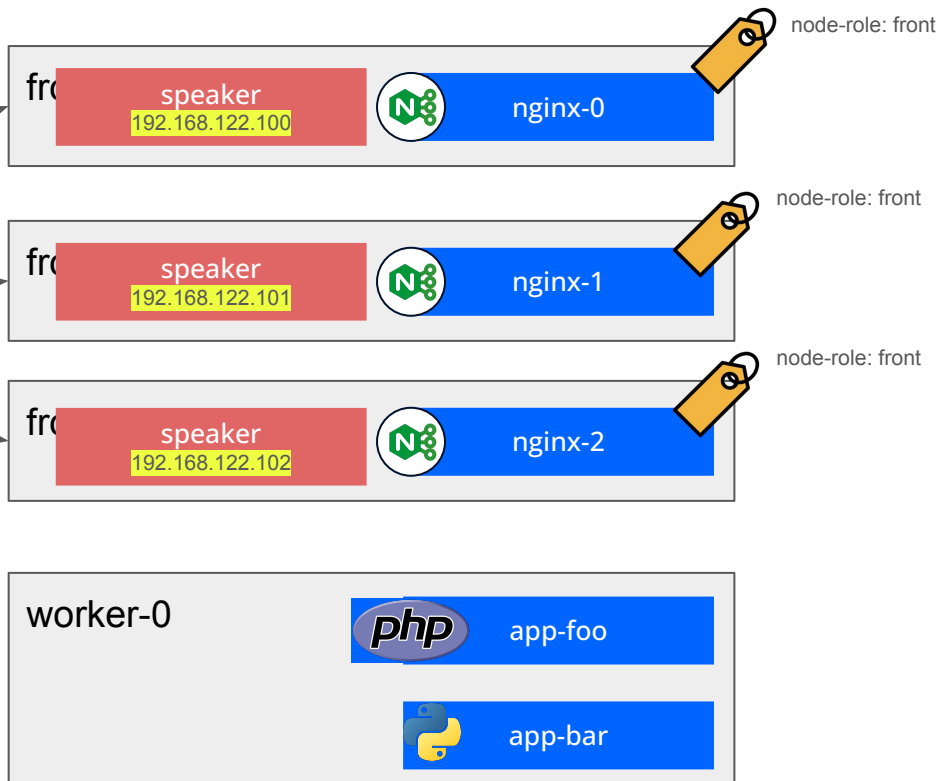


На всех фронтенд-узлах запускаются спикеры, каждый из которых получает отдельный адрес из пула.



# L2LoadBalancer

powered by MetalLB

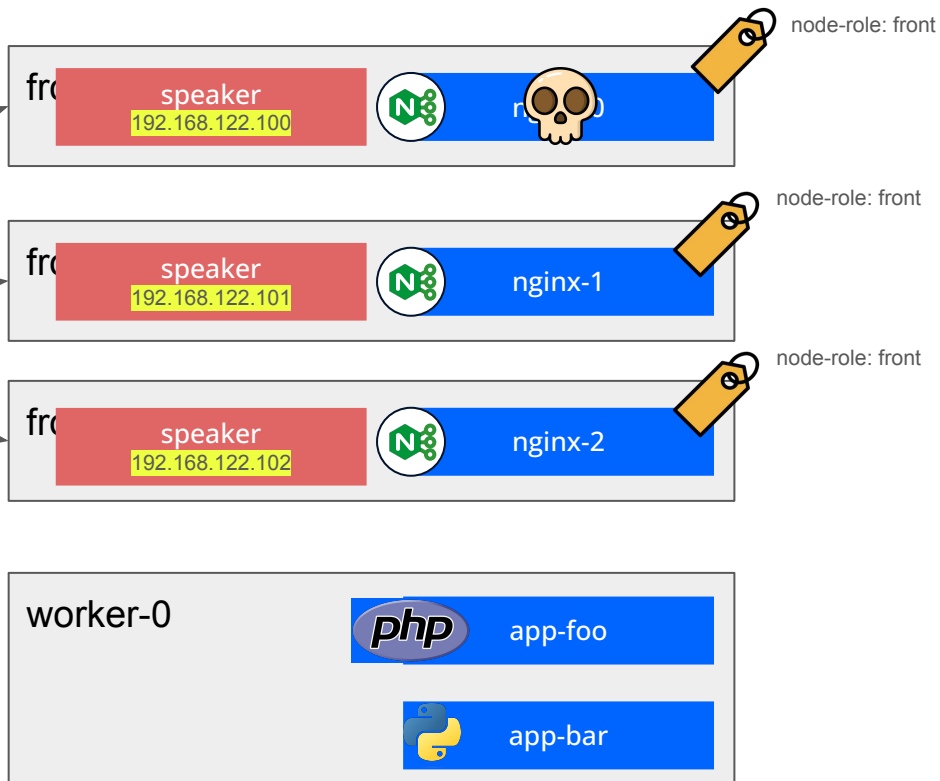


Каждый фронтенд-узел участвует в обработке прикладных запросов.  
Для этого в публичном DNS-имени приложения прописаны три A-записи.



# L2LoadBalancer

powered by MetalLB

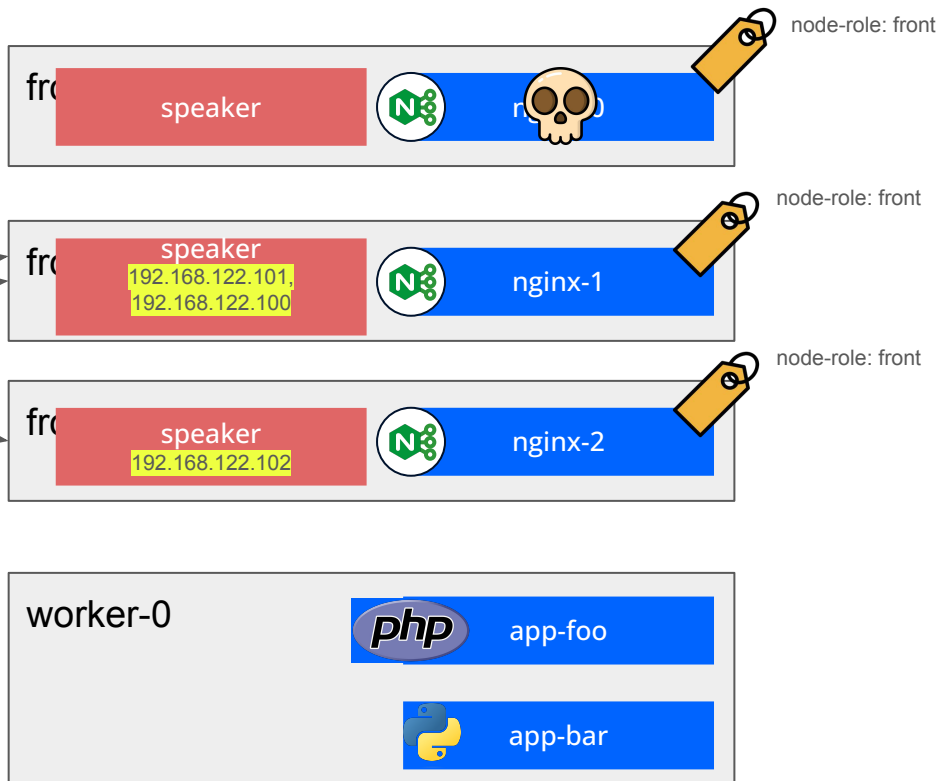


В случае выхода из строя приложения nginx на одном из фронтенд-узлов или самого узла, треть запросов обрывается...



# L2LoadBalancer

powered by MetalLB



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