DECKHOUSE

Kubernetes Platform

L2 Load Balancer Basics

L2LoadBalancer

There are three frontends nodes and one worker in the cluster.

L2LoadBalancer

```yaml
apiVersion: deckhouse.io/v1alpha1
kind: ModuleConfig
metadata:
  name: l2-load-balancer
spec:
  enabled: true
  version: 1
```

We turn on **l2-load-balancer** module.

L2LoadBalancer

```yaml
apiVersion: deckhouse.io/v1alpha1
kind: L2LoadBalancer
metadata:
  name: front
spec:
  addressPool:
  - 192.168.122.100-192.168.122.150
  nodeSelector:
    node-role: front
```

An L2LoadBalancer resource has been created specifying front-end nodes and a pool of "public" IP addresses. This allows for easily creating "zones" by associating specific address pools with a group of nodes. Speakers are run on all front-end nodes.

L2LoadBalancer

```yaml
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
```

An **Service** resource with type **LoadBalancer** has been created. It contains special annotations with the name of the L2LoadBalancer and the required number of IP addresses. Speakers are launched on all frontend nodes, each obtaining one or more addresses from the pool.

L2LoadBalancer

app.example.com

A 192.168.122.**100**
A 192.168.122.**101**
A 192.168.122.**102**

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

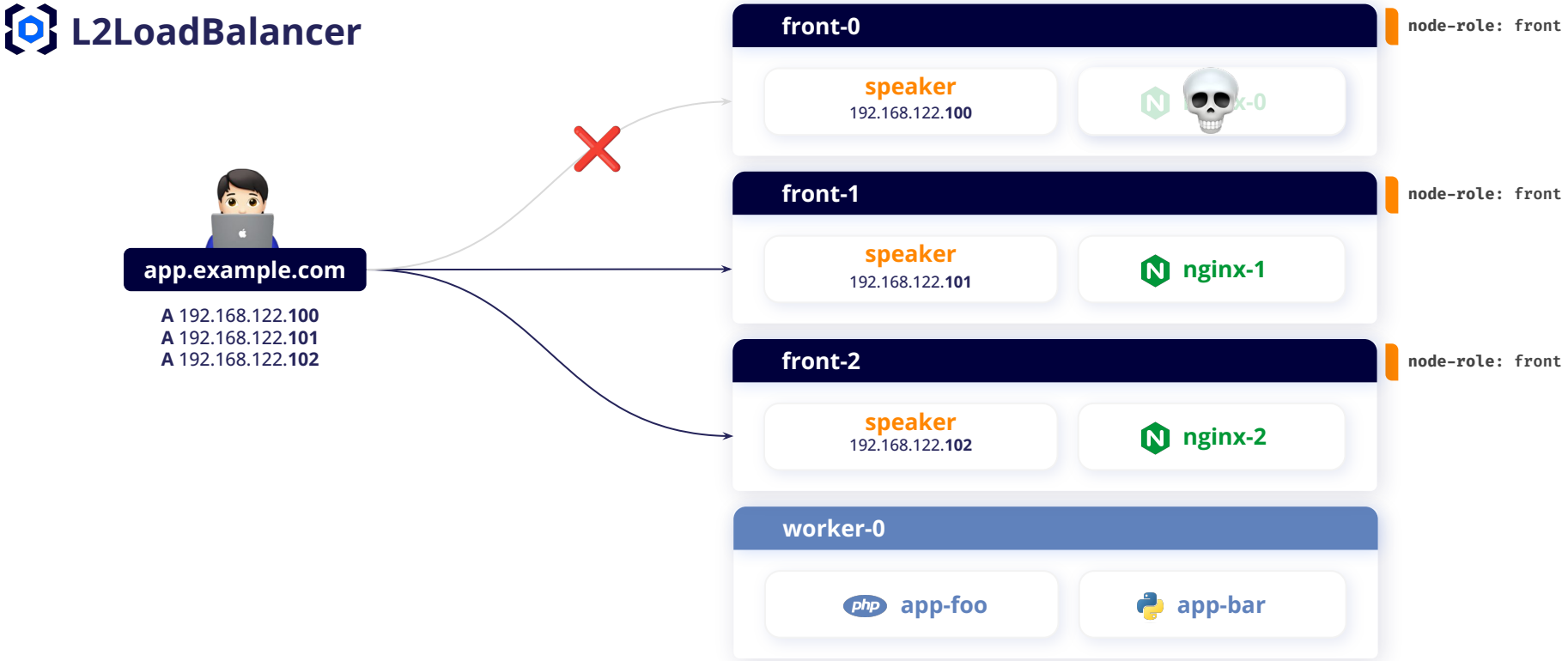
node-role: front

worker-0

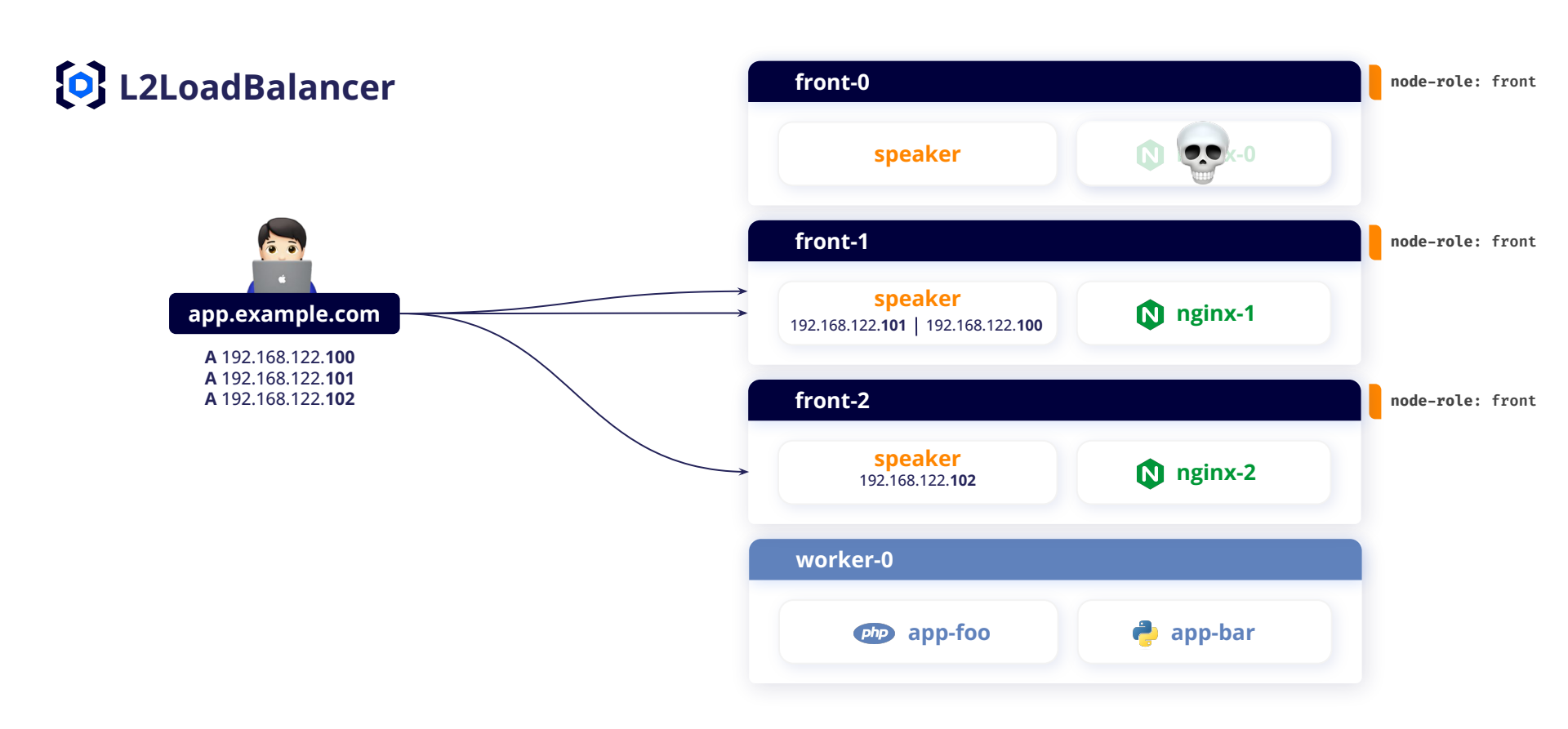
app-foo

app-bar

Each front-end node participates in handling application requests.
For this, three A records are specified in the public DNS name of the application.



In the event of a failure of the nginx application on one of the front-end nodes or the node itself, a third of the requests will fail,...



...and one of the remaining front-end nodes will take over the "problematic" IP address and handle the incoming application requests.