

NETWORKING (ANGOL NYELVŰ)

4. forduló

NOKIA

A kategória támogatója: NOKIA

Ismertető a feladathoz

A 4. forduló után elérhetőek lesznek a helyezések %-os formában: azaz kiderül, hogy a kategóriában a versenyzők TOP 20% - 40% - 60% -ához tartozol-e!

Szeretnénk rá felhívni figyelmedet, hogy a játék nem Forma-1-es verseny! Ha a gyorsaságod miatt kilököd a rendesen haladó versenyzőket, kizárást vonhat maga után!

Round 4 – K8S is responsible to implement the networking between services

Felhasznált idő: 00:00/10:00

Elért pontszám: 0/6

1. feladat 0/1 pont

What are the valid characteristics about Kubernetes networking?

Válaszok

- ☐ Kubernetes cannot run on single node cluster, it is impossible because of multi node networking topology.
- ☒ Kubernetes pod networking can be validated with http probes
Ez a válasz helyes, de nem jelölted meg.
- ☐ Kubernetes control plane elements are kube-apiserver, etc, kube-scheduler and container runtime
- ☐ Kubernetes node networking build up from multiple cluster
- ☐ Kubernetes using only virtual ethernet cables to implement its pod communication
- ☐ Kubernetes Pods are the smallest deployable units so its networking is opened externally by default
- ☒ Kubernetes using the kernel network namespace feature to isolate the network stack of a process
Ez a válasz helyes, de nem jelölted meg.
- ☐ Kubernetes pods are using ring network topology to communicate with each other

Magyarázat

Kubernetes using namespaces to isolate the resources like networking.

<https://www.digitalocean.com/community/tutorials/how-to-inspect-kubernetes-networking>

Probes can contain checkings with different purpose like http/tcp probes.

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/#http-probes>

2. feladat 0/2 pont

What is CNI – Container network interface? Please select the right definition.

Válasz

- ☒ CNI is a plugin interface which inserting a network interface into the container network namespace. It is also making any necessary changes on the host.
Ez a válasz helyes, de nem jelölted meg.
- ☐ CNI is an abstract implementation of Multus and Calico CNI plugi
- ☐ CNI is responsible to provide ADD, DEL, CHECK linux kernel level access for DPDK
- ☐ CNI is the direct link between kubernetes and Network card (NIC) to improve packet performance.
- ☐ CNI is handles input from Calico and transmit to Kubernetes Network attachment definition with ADD, DEL, MOD low level DPDK packet handling commands.
- ☐ CNI is responsible for network traffic encryption to provide secure network operations for kubernetes.
- ☐ CNI is a 3rd party implementation and not part of kubernetes. It is an extension developed by other companies.
- ☐ CNI is packet compression method for Multus CNI.

Magyarázat

Kubernetes network plugins details are explained here:

<https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/network-plugins/>

3. feladat 0/3 pont

Let's assume we have a Kubernetes cluster where gui-service SVC needs to be exposed with secure http connection. The Ingress controller name is equal to its ingress class which is test-system. Matching is based on a URL path prefix splitted by "/"

Please **select the best matching** ingress definition based on the given conditions.

Válasz

☒ Option 1:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
  namespace: test
  annotations:
    kubernetes.io/ingress.class: test-system
    nginx.ingress.kubernetes.io/backend-protocol: HTTPS
    nginx.ingress.kubernetes.io/listen-ports: '["HTTPS":443]'
spec:
  rules:
  - http:
      paths:
      - path: /swms
```

```
pathType: Prefix
backend:
  service:
    name: gui-service
    port:
      number: 443
```

Ez a válasz helyes, de nem jelölted meg.

☐ Option 2:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
  namespace: test
  annotations:
    kubernetes.io/ingress.class: system-test
    nginx.ingress.kubernetes.io/backend-protocol: HTTPS
    nginx.ingress.kubernetes.io/listen-ports: '["HTTPS":443]'
spec:
  rules:
  - http:
      paths:
      - path: /swms
        pathType: Prefix
        backend:
          service:
            name: gui-service
            port:
              number: 443
```

☐ Option 3:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
  namespace: test
  annotations:
    kubernetes.io/ingress.class: system-test
    nginx.ingress.kubernetes.io/backend-protocol: HTTP
    nginx.ingress.kubernetes.io/listen-ports: '["HTTP": 80], {"HTTPS":443}'
spec:
  rules:
  - http:
      paths:
      - path: /swms
        pathType: Prefix
        backend:
          service:
            name: gui-service
            port:
              number: 80
```

☐ Option 4:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
  namespace: test
```

```

annotations:
  kubernetes.io/ingress.class: test-system
  nginx.ingress.kubernetes.io/backend-protocol: HTTPS
  nginx.ingress.kubernetes.io/listen-ports: '["HTTPS":443]'
spec:
  rules:
  - http:
      paths:
      - path: /swms
        pathType: Prefix
        backend:
          service:
            name: oam-gui-backend
            port:
              number: 443

```

☐ Option 5:

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
  namespace: test
  annotations:
    kubernetes.io/ingress.class: test-system
    nginx.ingress.kubernetes.io/backend-protocol: HTTPS
    nginx.ingress.kubernetes.io/listen-ports: '["HTTPS":443]'
spec:
  rules:
  - http:
      paths:
      - path: /swms
        pathType: Exact
        backend:
          service:
            name: gui-service
            port:
              number: 443

```

☐ Option 6:

```

apiVersion: networking.k8s.io/v1
kind: nodeport
metadata:
  name: test-ingress
  namespace: test
  annotations:
    kubernetes.io/ingress.class: test-system
    nginx.ingress.kubernetes.io/backend-protocol: HTTPS
    nginx.ingress.kubernetes.io/listen-ports: '["HTTPS":443]'
spec:
  rules:
  - http:
      paths:
      - path: /swms
        pathType: Prefix
        backend:
          service:
            name: gui-service

```

```
port:
  number: 443
```

☐ Option 7:

```
apiVersion: networking.k8s.io/v1
kind: nodeport
metadata:
  name: test-ingress
  namespace: test
  annotations:
    nginx.ingress.kubernetes.io/backend-protocol: HTTPS
    nginx.ingress.kubernetes.io/listen-ports: '["HTTPS":443]'
spec:
  rules:
  - http:
      paths:
      - path: /swms
        pathType: Prefix
        backend:
          service:
            name: gui-service
            port:
              number: 443
```

☐ Option 8: None of the ingress definitions are valid based on the description.

Magyarázat

Option 1: right choice.

Option 2: ingress class is wrong

Option 3: http used instead of https

Option 4: service.name is wrong

Option 5: pathType is exact

Option 6: kind: nodeport is wrong.

Option 7: ingress class is missing.

Option 8: see Option 1.



[Legfontosabb tudnivalók](#) [Kapcsolat](#) [Versenyszabályzat](#) [Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos