

FULL STACK



Certified Kubernetes Administrator

FULL STACK

Kubernetes: Networking



Learning Objectives

By the end of this lesson, you will be able to:

- 🕒 Install and configure CoreDNS
- 🕒 Create a network namespace
- 🕒 Configure CNI in kubelet.service
- 🕒 Work with weave using DHCP or host-local
- 🕒 Work with name resolution in SVC in single as well as multiple namespaces
- 🕒 Create ingress with one rule having multiple paths



FULL STACK

Switching and Routing

Switching and Routing

Kube-router is a built-in solution of Kubernetes that is used to enhance the performance and simplicity of the application.

Kube-router provides a Linux-based proxy service and iptables/ipset-based network policy enforcer.

Kubernetes provides **OpenVSwicth**, a better way of building an overlay network.

FULL STACK

DNS

DNS: Basics

Domain Name System (DNS) is used to configure the kubelets and schedule pods and services on a cluster.

DNS is responsible for configuring the kubelets that let the containers use the DNS service's IP to resolve the DNS names.

Kubernetes clusters automatically configure the DNS, hence providing a mechanism for service discovery. This discovery helps the applications to find and communicate with each other.

DNS: Record Types

Here are the two types of records:

A Records

SRV Records

A records refer to the cluster
IP of the service

SRV records refer to the port
number and domain name

/etc/hosts Basics

When DNS and other options are not applicable, the /etc/hosts file is used to provide pod-level override of hostname resolution.

Host file is used to prevent docker from modifying the file once the containers are running.

Here is the command that can be used to check the hosts file content:

```
kubectl exec nginx -- cat /etc/hosts
```

Output:

```
# Kubernetes-managed hosts file.
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
fe00::0       ip6-mcastprefix
fe00::1       ip6-allnodes
fe00::2       ip6-allrouters
10.200.0.4     nginx
```

/etc/resolv.conf Basics

/etc/resolv.conf is a file that is used to store the local DNS configuration list.

The commands to be used with /etc/resolv.conf file are:

- Use **kubectl exec busybox cat /etc/resolv.conf** command to look inside the resolv.conf file
- To verify the search path and name server are set up, use:
**search default.svc.cluster.local svc.cluster.local cluster.local google.internal
c.gce_project_id.internal
nameserver 10.0.0.10
options ndots:5**

FULL STACK

CoreDNS

CoreDNS Installation and Configuration



Problem Statement: You are given a project to demonstrate the installation and configuration of CoreDNS.

ASSISTED PRACTICE

FULL STACK

Network Namespace

Creating Network Namespace



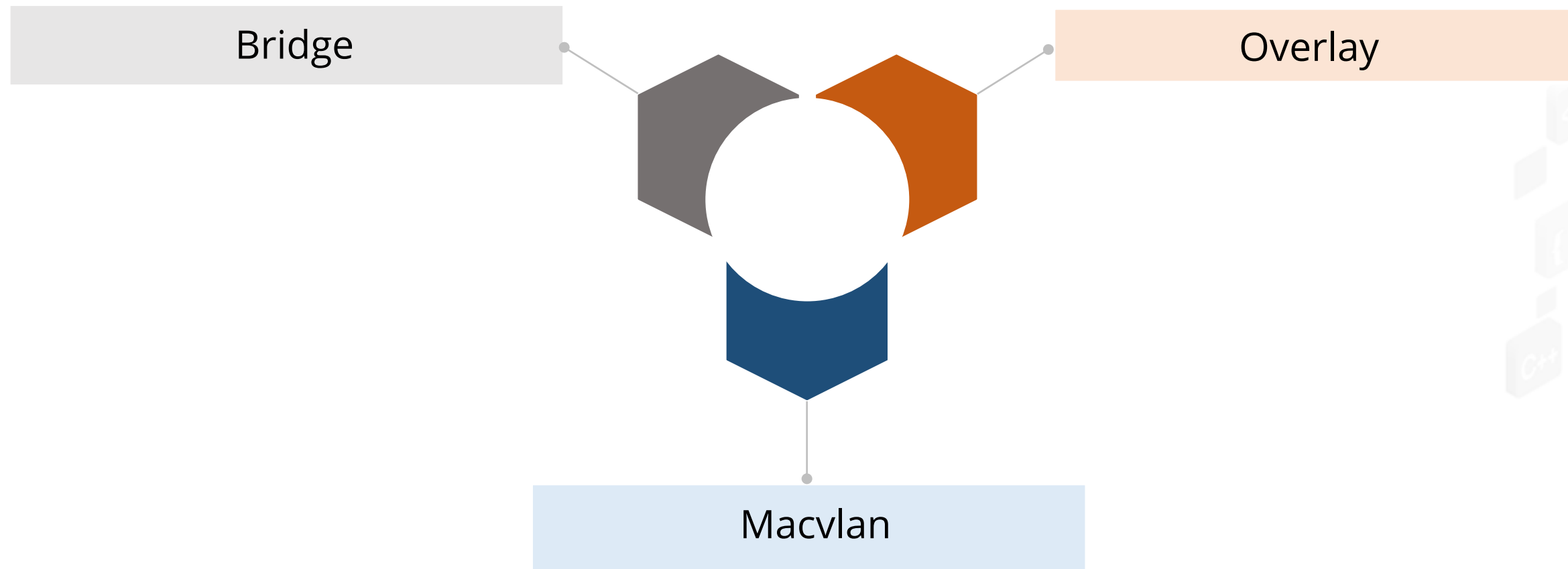
Problem Statement: You are given a task to create a network namespace.

ASSISTED PRACTICE

FULL STACK

Docker Networking

Different Network Types in Docker



FULL STACK

CNI

Container Network Interface (CNI): Requirements

- The CNI configuration file must match the CNI specifications.
- The required plugins for configuration must be present in **--cni-bin-dir**.
- If multiple configuration files are present, the first one in lexicographic order must be used.



CNM and CNI

Container Network Model (CNM) provides the interfaces that can be used to add or remove containers from a network.

Characteristics of the Container Network Model (CNM)

- Containers in the same network can communicate with each other
- Multiple endpoints per container can be used to join a container to multiple networks
- Multiple networks are supported by drivers
- An endpoint is added to a network sandbox for better connectivity

FULL STACK

Cluster Networking

Prerequisites for Node for Cluster Networking

Given below are the prerequisites for node for cluster networking:

- A node must contain information like address, condition, capability, and allocability.
- Pods on a node should be able to communicate with the pods on all the other nodes without NAT.
- Nodes containing pods in the host network should be able to communicate with the pods on all the other nodes without NAT.
- Node agents must be able to communicate with pods on the node.

FULL STACK

CNI in Kubernetes

Prerequisites for CNI

The CNI configuration file should match the given specifications:

- `cniVersion` must be **Semantic Version 2.0**
- Network name must be unique across all containers
- Additional arguments provided by the container at runtime
- An IP masquerade must be set up on the host for the networks



Configuring CNI in kubelet.service



Problem Statement: You are given a project to configure CNI in kubelet.service.

ASSISTED PRACTICE

FULL STACK

CNI in Weave

Introduction to Weaveworks



Problem Statement: You are given a project to demonstrate the workflow of weaveworks.

ASSISTED PRACTICE

FULL STACK

IP Address Management: Weave

IPAM Weave Using DHCP or Host-Local



Problem Statement: You are given a project to demonstrate the workflow of IPAM Weave using DHCP or host-local.

ASSISTED PRACTICE

FULL STACK

Service Networking

Cluster IP



Problem Statement: You are given a project to demonstrate the workflow of cluster IP.

ASSISTED PRACTICE

NodePort



Problem Statement: You are given a project to demonstrate the workflow of NodePort.

ASSISTED PRACTICE

FULL STACK

DNS in Kubernetes

Name Resolution in SVC in a Single Namespace



Problem Statement: You are given a project to demonstrate the workflow of resolving names in SVC in a single namespace.

ASSISTED PRACTICE

Name Resolution in SVC in Multiple Namespaces



Problem Statement: You are given a project to demonstrate the workflow of resolving names in SVC in multiple namespaces.

ASSISTED PRACTICE

FULL STACK

CoreDNS in Kubernetes

CoreDNS Setup



Problem Statement: You are given a project to set up the CoreDNS.

ASSISTED PRACTICE

FULL STACK

Ingress

Ingress Requirements

The basic requirement is an ingress controller that:

- Satisfies an ingress
- Fits the reference specification



Introduction to Ingress Controller

It is necessary to have an ingress controller running for the ingress resources to work.

Ingress controllers do not start automatically with a cluster, unlike other controllers.

Below are a few of additional controllers:

- Ambassador API Gateway
- AppsCode Inc
- Contour
- Citrix
- F5 Networks
- Gloo
- Istio
- NGINX
- Skipper
- Traefik



Introduction to Ingress Resources

An Ingress needs **apiVersion**, **kind**, and **metadata** fields with all other Kubernetes resources.

Below is an example of minimal ingress resource:

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: test-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /testpath
        backend:
          serviceName: test
          servicePort: 80
```


Ingress Creation with One Rule and Multiple Paths



Problem Statement: You are given a project to create ingress with one rule and multiple paths.

ASSISTED PRACTICE

Key Takeaways

You are now able to:

- Install and configure CoreDNS
- Create a network namespace
- Configuring CNI in kubelet.service
- Work with weave using DHCP or host-local
- Work with name resolution in SVC in single as well as multiple namespaces
- Create ingress with one rule having multiple paths



FULL STACK



Knowledge Check

Knowledge Check

1

Which of the following is NOT a network type in docker?

- a. Bridge
- b. Connector
- c. Overlay
- d. Macvlan



Knowledge Check

1

Which of the following is NOT a network type in docker?

- a. Bridge
- b. Connector
- c. Overlay
- d. Macvlan



The correct answer is **b**

Connector is not a network type in docker.

Knowledge Check

2

Which of the following records refer to the port number and domain name?

- a. A records
- b. SRV records
- c. ETC records
- d. CNM records



Knowledge
Check

2

Which of the following records refer to the port number and domain name?

- a. A records
- b. SRV records
- c. ETC records
- d. CNM records



The correct answer is **b**

SRV records refer to the port number and the domain name.

**Knowledge
Check**

3

Which of the following is NOT a controller?

- a. Ambassador API Gateway
- b. Citrix
- c. NGINX
- d. F8 networks



Knowledge
Check

3

Which of the following is NOT a controller?

- a. Ambassador API Gateway
- b. Citrix
- c. NGINX
- d. F8 networks



The correct answer is **d**

F8 networks is not a controller.

Knowledge Check

4

_____ is used to provide pod-level override of hostname resolution.

- a. /etc/resolve.conf
- b. /etc/hosts
- c. --cni-bin-dir
- d. /etc/resolve/config



Knowledge
Check

4

_____ is used to provide pod-level override of hostname resolution.

- a. /etc/resolve.conf
- b. /etc/hosts
- c. --cni-bin-dir
- d. /etc/resolve/config



The correct answer is **b**

/etc/hosts is used to provide pod-level override of hostname resolution.

**Knowledge
Check**

5

Which of the following is a prerequisites of CNI?

- a. The CNI configuration file must match the CNI specifications
- b. The required plugins for configuration must be present in `--cni-bin-dir`
- c. If multiple configuration files are present, the first one in lexicographic order must be used
- d. All of the above



Knowledge
Check

5

Which of the following is a prerequisites of CNI?

- a. The CNI configuration file must match the CNI specifications
- b. The required plugins for configuration must be present in --cni-bin-dir
- c. If multiple configuration files are present, the first one in lexicographic order must be used
- d. All of the above



The correct answer is **d**

The CNI configuration file must match the CNI specifications, the required plugins for configuration must be present in --cni-bin-dir, and if multiple configuration files are present, the first one in lexicographic order must be used.



Problem Statement: One of the biggest challenges while developing a highly scalable application is improving the concurrent user capacity and the overall reliability of the application. The use of Kubernetes helps improve these by distributing the workload across multiple servers/DNS, decreasing the overall burden placed on each server.

Objective: Achieve load balancing using ingress in GKE.