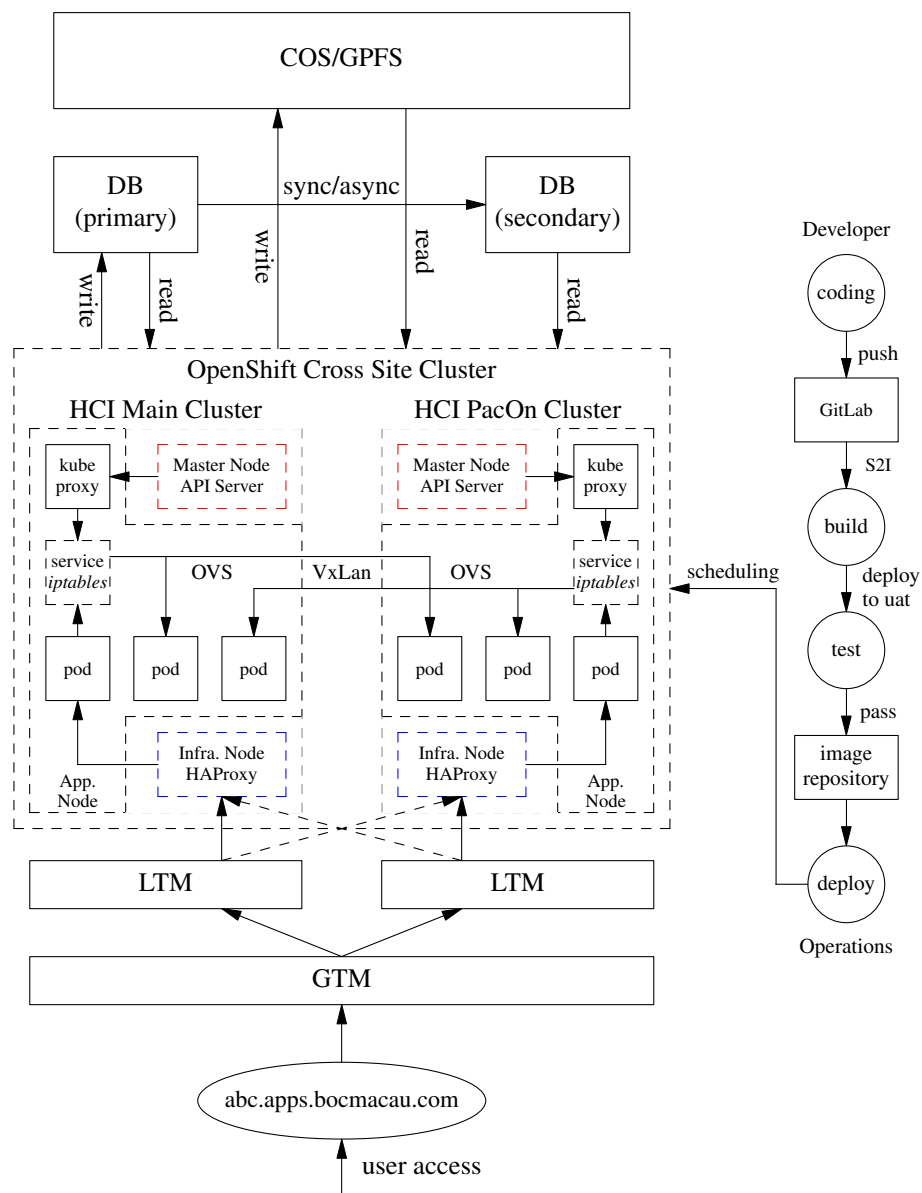# 1. Architecture

1.1. Master node

The OpenShift Container Platform master is a server that performs control functions for the whole cluster environment. It is responsible for the creation, scheduling, and management of all objects specific to OpenShift. It includes API, controller manager, and scheduler capabilities in one OpenShift binary. It is also a common practice to install an etcd key-value store on OpenShift masters to achieve a low-latency link between etcd and OpenShift masters. It is recommended that you run both OpenShift masters and etcd in highly available environments. This can be achieved by running multiple OpenShift masters in conjunction with an external active-passive load balancer and the clustering functions of etcd.

-2-

1.2. Infrastructure node

The OpenShift infrastructure node runs infrastructure-specific services such as the Docker Registry and the HAProxy router. The Docker Registry stores application images in the form of containers. The HAProxy router provides routing functions for OpenShift applications. It currently supports HTTP(S) traffic and TLS-enabled traffic via Server Name Indication (SNI). Additional applications and services can be deployed on OpenShift infrastructure nodes.

1.3. Application node

The OpenShift application nodes run containerized applications created and deployed by developers. An OpenShift application node contains the OpenShift node components combined into a single binary, which can be used by OpenShift masters to schedule and control containers.
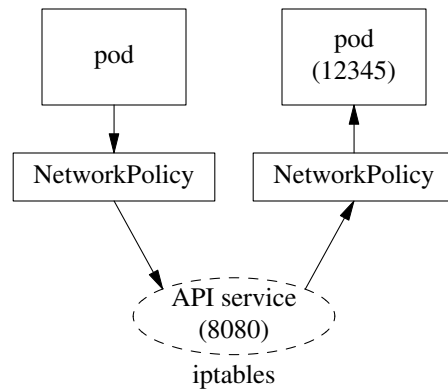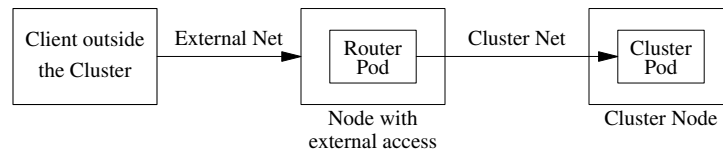
2. OpenShift network
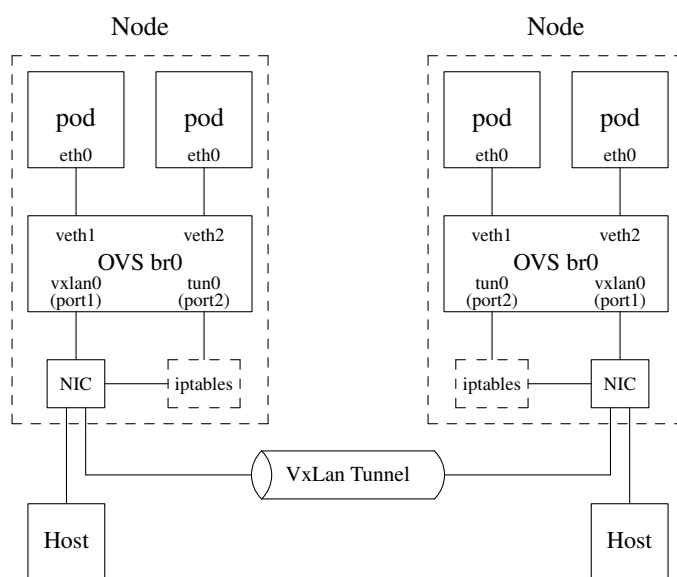
In the default configuration, the cluster network is the **10.128.0.0/14** network, and nodes allocated **/23** subnets (i.e., 10.128.0.0/23, 10.128.2.0/23, 10.128.4.0/23, and so on). This means that the cluster network has 512 subnets available to assign to nodes, and a given node is allocated 510 addresses that it can assign to the containers running on it.

- Pod to pod in the same node.
  eth0 (pod) → vethA → br0 → vethB → eth0 (pod)

- Pod to pod in the different nodes.
  eth (pod) → vethA → br0 → vxlan0 → network → vxlan0 → br0 → vethB → eth0 (pod)

- Pod to external host.

  $$eth0\ (pod) \rightarrow vethA \rightarrow br0 \rightarrow tun0 \rightarrow \underset{\texttt{(MASQUERADE)}}{\texttt{SNAT}} \rightarrow eth0\ (NIC) \rightarrow Host$$

- Pod to services (with NetworkPolicy).

```
┌──────────┐              ┌──────────┐
│   pod    │              │   pod    │
│          │              │ (12345)  │
└────┬─────┘              └────▲─────┘
     │                         │
     ▼                         │
┌──────────────┐       ┌──────────────┐
│NetworkPolicy │       │NetworkPolicy │
└──────┬───────┘       └──────▲───────┘
        ╲                    ╱
         ╲                  ╱
          ▼                ╱
      ⟨ API service ⟩
      ⟨   (8080)   ⟩
         iptables
```

- User to frontend pod.

```
┌────────────┐ External Net ┌────────────┐ Cluster Net ┌───────────┐
│Client outside│────────────▶│ ┌────────┐ │────────────▶│ ┌───────┐ │
│ the Cluster │              │ │ Router │ │             │ │Cluster│ │
│            │              │ │  Pod   │ │             │ │  Pod  │ │
└────────────┘              │ └────────┘ │             │ └───────┘ │
                            └────────────┘             └───────────┘
                              Node with                 Cluster Node
                            external access
```

3. VxLan
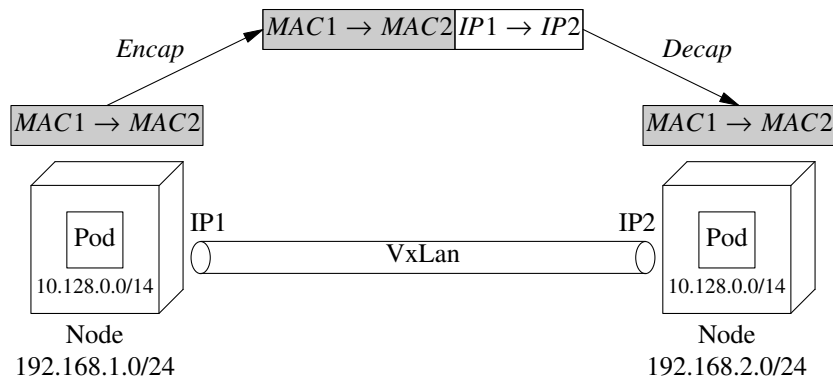
Default cluster network is **10.128.0.0/14**.