KUBERNETES ARCHITECTURE

Kubernetes Architecture

CNI Network Configuration File

To provide container networking, Kubernetes is standardizing on the Container Network Interface (CNI) specification. Since v1.6.0, the goal of kubeadm (the Kubernetes cluster bootstrapping tool) has been to use CNI, but you may need to recompile to do so.

CNI is an emerging specification with associated libraries to write plugins that configure container networking and remove allocated resources when the container is deleted. Its aim is to provide a common interface between the various networking solutions and container runtimes. As the CNI specification is language-agnostic, there are many plugins from Amazon ECS, to SR-IOV, to Cloud Foundry, and more.

With CNI, you can write a network configuration file:

```
{
    "cniVersion": "0.2.0",
    "name": "mynet",
    "type": "bridge",
    "bridge": "cni0",
    "isGateway": true,
    "ipMasq": true,
    "ipam": {
        "type": "host-local",
        "subnet": "10.22.0.0/16",
        "routes": [
            { "dst": "0.0.0.0/0" }
             ]
     }
}
```

This configuration defines a standard Linux bridge named **cni0**, which will give out IP addresses in the subnet 10.22.0.0./16. The bridge plugin will configure the network interfaces in the correct namespaces to define the container network properly.

The main **README** of the CNI GitHub repository has more information.