KUBESPHERE 尚硅谷

云原生实战

KubeVirt 核心概念解析

刘远清--KubeSphere Virtualization 后端开发

Agenda

1. 虚拟机
2. 虚拟机镜像
3. 磁盘和卷
4. 网络

虚拟机

virtualmachines(VM): 为群集内的 VirtualM
拟机，确保虚拟机实例的启动状态。

virtualmachineinstances（VMI）：类似于
VirtualMachineInstance 对象即表示一台正
置

VirtualMachineInstanceReplicaSet: 类似K
VirtualMachineInstance，并且保证指定数

VirtualMachineInstanceMigrations：提供

```
spec:
  domain:
    cpu:
      cores: 1
    devices:
      disks:
      - bootOrder: 1
        disk:
          bus: virtio
        name: vol-0wsve9ix
      - disk:
          bus: virtio
        name: cloudinitdisk
      interfaces:
      - macAddress: ce:44:ac:10:00:06
        macvtap: {}
        name: eth0
    machine:
      type: q35
    resources:
      limits:
        cpu: "4"
        memory: 4Gi
      requests:
        cpu: 1333m
        memory: 4Gi
  hostname: wtest
  networks:
  - multus:
      networkName: kubesphere-virtualization-system/vxnet-lgxxk7dx
    name: eth0
  volumes:
  - cloudInitNoCloud:
      networkDataBase64:
      userDataBase64:
    name: cloudinitdisk
  - name: vol-0wsve9ix
    persistentVolumeClaim:
      claimName: tpl-vol-0wsve9ix
```

虚拟机镜像

Containerized-Data-Importer (CDI) is a persistent storage management add-on for Kubernetes. It's primary goal is to provide a declarative way to build Virtual Machine Disks on PVCs for Kubevirt VMs

datavolumes.cdi.kubevirt.io -- dataVolume

```go
// DataVolumeSource represents the source for our Data Volume, thi
type DataVolumeSource struct {
    HTTP     *DataVolumeSourceHTTP     `json:"http,omitempty"`
    S3       *DataVolumeSourceS3       `json:"s3,omitempty"`
    Registry *DataVolumeSourceRegistry `json:"registry,omitempty"`
    PVC      *DataVolumeSourcePVC      `json:"pvc,omitempty"`
    Upload   *DataVolumeSourceUpload   `json:"upload,omitempty"`
    Blank    *DataVolumeBlankImage     `json:"blank,omitempty"`
    Imageio  *DataVolumeSourceImageIO  `json:"imageio,omitempty"`
    VDDK     *DataVolumeSourceVDDK     `json:"vddk,omitempty"`
}
```

```yaml
spec:
  pvc:
    accessModes:
    - ReadWriteMany
    resources:
      requests:
        storage: 20Gi
    volumeMode: Block
  source:
    http:
      url: http://172.16.0.2/Win10_20H2_Chinese_Simplified_x64.iso
```

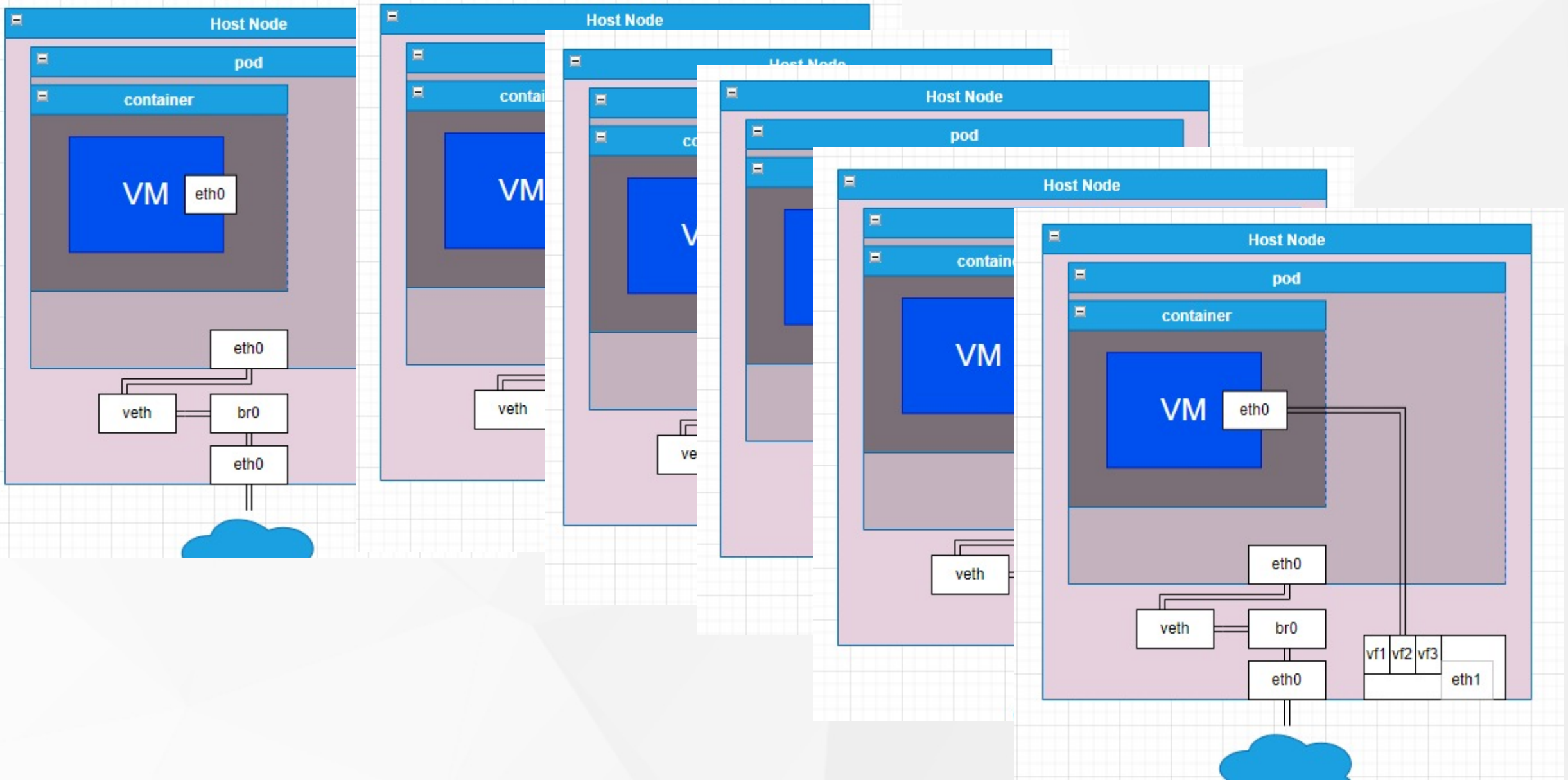
磁盘和卷

在 spec.volumes 下可以指定多种类型的卷：

cloudInitNoCloud：Cloud-init相关的配置，用于修改或者初始化虚拟机中的配置信息
containerDisk：指定一个包含 qcow2 或 raw 格式的 docker 镜像，重启 vm 数据会丢失
dataVolume：动态创建一个 PVC，并用指定的磁盘映像填充该 PVC，重启 vm 数据不会丢失
emptyDisk：从宿主机上分配固定容量的空间，映射到vm中的一块磁盘，emptyDisk 的生命周期与 vm 等同，重启 mv 数据会丢失
ephemeral: 在虚机启动时创建一个临时卷，虚机关闭后自动销毁，临时卷在不需要磁盘持久性的任何情况下都很有用。
hostDisk：在宿主机上创建一个 img 镜像文件，挂给虚拟机使用。重启 vm 数据不会丢失。
persistentVolumeClaim: 指定一个 PVC 创建一个块设备。重启 vm 数据不会丢失。

configMap
serviceAccount
secret ： 可以把信息configMap，serviceAccount，secret写入到 iso 磁盘中，挂给虚拟机。

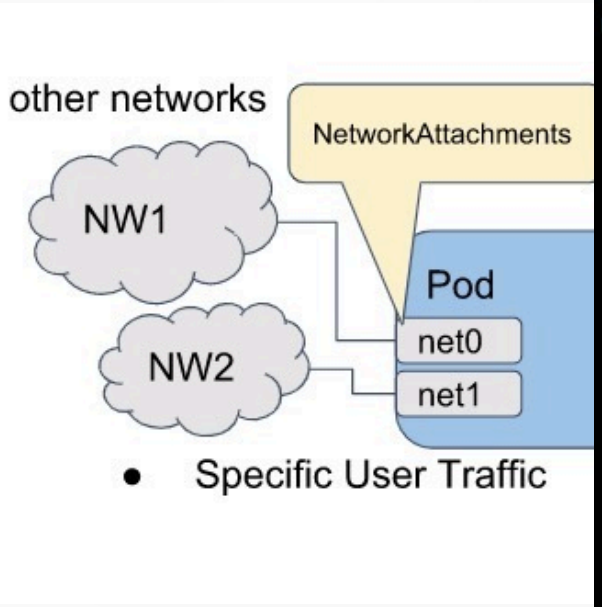
网络


Host Node
pod
container
VM
eth0
veth
br0
eth0
vf1
vf2
vf3
eth1

网络

Multus

- Multus CNI enables attaching multiple network interfaces to pods in Kubernetes.
- CR - network-attachment-definitions.k8s.cni.cncf.io




```
apiVersion: k8s.cni.cncf.io/v1
kind: NetworkAttachmentDefinition
metadata:
  annotations:
    k8s.v1.cni.cncf.io/resourceName: macvtap.network.kubevirt.io/eth0.40
  creationTimestamp: "2021-10-26T08:31:15Z"
  generation: 1
  managedFields:
  - apiVersion: k8s.cni.cncf.io/v1
    fieldsType: FieldsV1
    fieldsV1:
      f:metadata:
        f:annotations:
          .: {}
          f:k8s.v1.cni.cncf.io/resourceName: {}
      f:spec:
        .: {}
        f:config: {}
    manager: express
    operation: Update
    time: "2021-10-26T08:31:15Z"
  name: vxnet-fnhcrvo0
  namespace: kubesphere-virtualization-system
  resourceVersion: "3289055"
  uid: e06000d1-4ffc-4f55-a51f-a6f91ec473c8
spec:
  config: '{"cniVersion": "0.3.1","type": "macvtap","mtu": 1300}'
```

网络

Connecting a virtual machine to a network co
- Frontend -- *spec.domain.devices.interfaces*
- Backend -- *spec.networks*

Each network should declare its type by defin
- pod -- Default Kubernetes network
- multus -- Secondary network provided usi

Binding method for *spec.domain.devices.inte*
- bridge -- Connect using a linux bridge
- sriov -- Pass through a SR-IOV PCI device v
- masquerade -- Connect using Iptables rule
- macvtap -- Connect using macvtap

```
spec:
  domain:
    cpu:
      cores: 1
    devices:
      disks:
      - bootOrder: 1
        disk:
          bus: virtio
        name: vol-0wsve9ix
      - disk:
          bus: virtio
        name: cloudinitdisk
      interfaces:
      - macAddress: ce:44:ac:10:00:06
        macvtap: {}
        name: eth0
    machine:
      type: q35
    resources:
      limits:
        cpu: "4"
        memory: 4Gi
      requests:
        cpu: 1333m
        memory: 4Gi
  hostname: wtest
  networks:
  - multus:
      networkName: kubesphere-virtualization-system/vxnet-lgxxk7dx
    name: eth0
  volumes:
  - cloudInitNoCloud:
      networkDataBase64:
      userDataBase64:
    name: cloudinitdisk
  - name: vol-0wsve9ix
    persistentVolumeClaim:
      claimName: tpl-vol-0wsve9ix
```

Reference

https://github.com/kubevirt/kubevirt
https://github.com/kubevirt/containerized-data-importer/
https://github.com/k8snetworkplumbingwg/multus-cni

KUBESPHERE 尚硅谷

谢 谢 观 看