# 云原生云主机服务 (KubeVirt) 使用手册 Lite

# 准备工作 ∂

# 安装 virtctl 🔗

- 1 wget <a href="https://github.com/kubevirt/kubevirt/releases/download/v0.47.1/virtctl-v0.47.1-linux-amd64">https://github.com/kubevirt/kubevirt/releases/download/v0.47.1/virtctl-v0.47.1-linux-amd64</a>
- 2 chmod a+x virtctl-v0.47.1-linux-amd64
- 3 cp virtctl-v0.47.1-linux-amd64 /usr/local/bin/virtctl

## 下载镜像 🔗

CentOS:

CentOS Linux and Stream cloud images download

#### 公司自制镜像:

注意:自制的CentOS镜像kubevirt不可用

X86-标准镜像维护列表

#### Microsoft官方Windows ISO镜像:

Microsoft Evaluation Center

# 安装服务 ♂

云产品: 云原生云主机安装

过程略

# 使用 ⊘

# 镜像管理 ⊘

#### 上传镜像 ♂

云原生云主机服务通过CDI模块实现上传镜像的能力

执行以下命令上传镜像:

1 virtctl image-upload pvc <PVC名称> --size=10Gi --image-path=<本地镜像文件路径> --storage-class=<存储类> --wait-secs=240 --uploadproxy-url=<https://cdi-uploa

CDI会自动创建一个指定PVC名称的PVC来保存镜像。

### 删除镜像 ♂

### 删除对应PVC即可:

1 kubectl delete pvc < PVC名称>

# 预设管理 (将被弃用) ⊘

预设仅支持定义domain属性

### 创建预设 ⊘

创建 5.vmipreset-small.yaml 如下:

```
1 apiVersion: kubevirt.io/v1alpha3
2 kind: VirtualMachineInstancePreset
3 metadata:
4 name: c2-g4
5 spec:
6 selector:
7
    matchLabels:
8 kubevirt.io/size: 2C-4G
9 domain:
10
    cpu:
11
     cores: 2
12
    resources:
13
     requests:
14
     memory: 4G
15
     devices: {}
```

### 查看预设 ⊘

1 kubectl get virtualmachineinstancepresets

### 修改预设 ⊘

1 kubectl edit virtualmachineinstancepresets c2-g4

# 删除预设 ♂

1 kubectl delete virtualmachineinstancepresets c2-g4

# 云主机管理 ⊘

# 创建云主机 ⊘

云原生云主机的属性均通过CR VirtualMachine定义。

### 配置项 🔗

# 设置CPU:

```
1 spec:
2 template:
3 spec:
4 domain:
5 cpu:
6 cores: 2
```

### 设置内存:

```
spec:
template:
spec:
domain:
```

```
5 resources:
6 requests:
7 memory: 4G
```

# 设置存储:

- 卷设置:
  - o PVC

```
spec:
template:
spec:
volumes:
- name: sys-disk
persistentVolumeClaim:
claimName: centos-7.9
```

- 盘设置:
  - o disk

```
1 spec:
     template:
3
      spec:
4
       domain:
5
        devices:
6
          disks:
          # This makes it a disk
8
          - disk:
9
            # This makes it exposed as /dev/vda, being the only and thus first
10
            # disk attached to the VM
            bus: virtio
11
12
           name: sys-disk
```

o cdrom

```
1 spec:
2
   template:
3
     spec:
4
      domain:
5
        devices:
6
         disks:
 7
         - name: cdromiso
8
          # This makes it a cdrom
9
          cdrom:
10
           # This makes the cdrom writeable
11
           readOnly: false
12
           # This makes the cdrom be exposed as SATA device
13
           bus: sata
```

# 设置启动顺序:

```
      1
      spec:

      2
      template:

      3
      spec:

      4
      domain:

      5
      devices:

      6
      disks:

      7
      - bootOrder: 1
```

```
8 cdrom:
9 bus: sata
10 name: cdromiso
11 -bootOrder: 2
12 disk:
13 bus: virtio
14 name: harddrive
```

### 设置网络:

- 接口设置:
  - slirp

略

o macvtap

略

• 网络设置:

0

- 固定IP
  - o pod默认网络:

当前pod默认网络使用为flannel,不支持固定IP

- 。 集群第二网络:
  - kube-ovn

kube-ovn的新特性已经支持云主机的生命周期中自动分配并固定IP与Mac地址(详见:EAS-103081)。若需要手动指定IP与Mac地址,可通过以下方式:

```
1 spec:
2 template:
3 metadata:
4 annotations:
5 ovn.kubernetes.io/ip_address: <ipt地址>
6 ovn.kubernetes.io/mac_address: <mac地址>
```

sriov-cni/ib-sriov-cni

不支持固定IP

### 使用预设:

```
1 spec:
2 template:
3 metadata:
4 labels:
5 kubevirt.io/size: 2C-4G
```

#### 秘钥注入:

首先创建SSH公钥Secret:

```
apiVersion: v1
data:
key1: <ssh_public_key>
kind: Secret
metadata:
name: public-key
type: Opaque
```

通过access credentials接口使用configDrive传递方式注入:

```
1 spec:
2 template:
3
      spec:
       domain:
 4
5
        devices:
 6
          disks:
 7
         - disk:
8
           bus: virtio
9
           name: cloudinitdisk
       accessCredentials:
10
11
       - sshPublicKey:
          propagationMethod:
12
13
          configDrive: {}
14
          source:
15
           secret:
16
            secretName: public-key
17
       volumes:
18
       - cloud In it Config Drive:\\
19
          userData: |
20
          #cloud-config
           disable_root: false
21
22
        name: cloudinitdisk
```

### 密码注入:

用户,密码,以及其他初始化命令可通过 cloudInitNoCloud 卷的方式注入:

该方式也可注入秘钥,但上文秘钥注入方式相较于此的好处是与cloud-init用户数据(user data)隔离

```
1 spec:
    2
                    template:
    3
                             spec:
     4
                                    domain:
    5
                                        devices:
     6
                                                disks:
     7
                                              - disk:
   8
                                                         bus: virtio
    9
                                                     name: cloudinitdisk
10
                                    volumes:
                                    - cloudInitNoCloud:
11
12
                                                userData: |
13
                                                  #cloud-config
14
                                                  users:
15
                                                      - name: escore
16
                                                               gecos: ES Core User
17
                                                               sudo: ALL=(ALL) NOPASSWD:ALL
                                                               passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6
18
19
                                                               shell: /bin/bash
20
                                                               home: /home/escore
21
                                                               lock_passwd: false
22
                                                               ssh_pwauth: true
23
                                           name: cloudinitdisk
```

#### 创建 Linux 云主机 ⊘

\*\*示例一:\*\*创建从PVC启动,使用默认Pod网络的云主机

#### 1.vm-centos-boot-pvc.yaml :

```
1 apiVersion: kubevirt.io/v1alpha3
2 kind: VirtualMachine
3 metadata:
4 name: centos-boot-pvc
5 spec:
6 running: true
 7
     template:
8
      metadata:
9
       labels:
10
        kubevirt.io/domain: centos-boot-pvc
11
12
       domain:
13
14
        cores: 2
15
        devices:
         disks:
16
         - disk:
17
18
           bus: virtio
19
          name: boot-disk
20
         interfaces:
21
         - masquerade: {}
22
          name: default
23
        machine:
24
         type: q35
25
        resources:
26
         requests:
27
          memory: 4G
28
       networks:
29
       - name: default
30
        pod: {}
31
       volumes:
32
       - name: boot-disk
33
        persistent Volume Claim:\\
34
         claimName: centos-7.9
```

# \*\*示例二:\*\*创建从DV启动,使用集群第二网络的云主机

## 2.vm-centos-net-ovn-boot-dv.yaml :

```
1 apiVersion: kubevirt.io/v1alpha3
2 kind: VirtualMachine
3 metadata:
4 name: centos-net-ovn-boot-dv
5 spec:
6 running: true
 7
     template:
      metadata:
9
       labels:
10
        kubevirt.io/domain: centos-net-ovn-boot-dv
11
      spec:
12
       domain:
13
        cpu:
14
        cores: 2
15
        devices:
16
         disks:
```

```
17
                             - disk:
18
                                    bus: virtio
19
                                name: boot-disk
20
                             - disk:
21
                                    bus: virtio
22
                                name: cloudinitdisk
23
                             interfaces:
24
                             - bridge: {}
                                name: default
25
26
                          machine:
27
                             type: q35
28
                          resources:
29
                             requests:
30
                                memory: 4G
31
                       networks:
32
                       - name: default
33
                          multus:
                             default: true
34
35
                            networkName: eks-managed/kube-ovn
36
                       volumes:
37
                       - name: boot-disk
38
                          dataVolume:
39
                             name: centos-dv
40
                       - cloudInitNoCloud:
41
                             userData: |
42
                               #cloud-config
43
                                users:
44
                                  - name: escore
45
                                       gecos: ES Core User
46
                                       sudo: ALL=(ALL) NOPASSWD:ALL
47
                                       passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPSICnV8P6rHe1kcA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3Ch
48
                                       shell: /bin/bash
49
                                       home: /home/escore
50
                                       lock_passwd: false
51
                                       ssh_pwauth: true
52
                          name: cloudinitdisk
53
                data Volume Templates:\\
54
                - metadata:
55
                       name: centos-dv
56
                   spec:
57
                      pvc:
58
                          access Modes:\\
59
                          - ReadWriteOnce
60
                          resources:
61
                            requests:
                               storage: 30Gi
62
63
                          storageClassName: capacity
64
                       source:
65
66
                             namespace: default
67
                             name: centos-7.9
```

该云主机配置为集群第二网络:kube-ovn,通过DV模板定义的DV启动,此外还配置了 cloudInitNoCloud 卷用于注入自定义用户与密码。

\*\*示例三: \*\*创建使用预设的云主机

5.vm-centos-boot-dv-with-preset.yaml :

1 apiVersion: kubevirt.io/v1alpha3

```
2 kind: VirtualMachine
   3
           metadata:
             name: centos-boot-dv2
  5 spec:
  6
               running: true
   7
               template:
  8
                   metadata:
 9
                     labels:
                        kubevirt.io/domain: centos-boot-dv2
10
11
                         kubevirt.io/size: 2C-4G
12
                   spec:
13
                     domain:
14
                        devices:
15
                           disks:
                           - disk:
16
17
                                  bus: virtio
18
                               name: boot-disk
19
                           - disk:
20
                                  bus: virtio
21
                               name: cloudinitdisk
22
                           interfaces:
23
                           - masquerade: {}
24
                               name: default
25
                         machine:
26
                           type: q35
27
                      networks:
28
                     - name: default
29
                        pod: {}
30
                      volumes:
31
                      - name: boot-disk
                         dataVolume:
32
33
                           name: centos-dv2
34
                      - cloudInitNoCloud:
35
                           userData: |
36
                              #cloud-config
37
                               users:
38
                                - name: escore
39
                                     gecos: ES Core User
40
                                     sudo: ALL=(ALL) NOPASSWD:ALL
41
                                     passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgbIbJh3CDXfs2GkmH038g4EW2KPsICnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6rHe1kcA/PS1CnV8P6
42
                                     shell: /bin/bash
43
                                     home: /home/escore
44
                                     lock_passwd: false
45
                                     ssh_pwauth: true
46
                         name: cloudinitdisk
47
               dataVolumeTemplates:
48
               - metadata:
49
                     name: centos-dv2
50
                   spec:
51
                     pvc:
52
                        accessModes:
53
                        - ReadWriteOnce
54
                         resources:
55
                           requests:
56
                               storage: 30Gi
57
                        storageClassName: capacity
58
                      source:
59
                         pvc:
```

```
namespace: default
name: centos-7.9
```

该云主机通过 labels 指定预设配置,注入了CPU与内存的规格。

\*\*示例四: \*\*创建使用sriov网络的云主机 (手动)

该方式需要手动规划网络

### 通过cloud-init NoCloud方式注入网络配置:

```
1 apiVersion: kubevirt.io/v1alpha3
2 kind: VirtualMachine
 3 metadata:
    name: centos-sriov
5 spec:
6
     data Volume Templates:\\
 7
      - metadata:
8
        creationTimestamp: null
9
         name: centos-sriov-dv
10
        spec:
11
        pvc:
12
          accessModes:
13
          - ReadWriteMany
14
          resources:
15
           requests:
16
            storage: 30Gi
17
          storageClassName: nfs-client
18
        source:
19
          pvc:
20
           name: centos-7.9
21
           namespace: nsa
22
     running: true
23
     template:
24
      metadata:
25
        labels:
26
        kubevirt.io/domain: centos-sriov
27
       spec:
28
        domain:\\
29
        cpu:
30
         cores: 2
31
         devices:
32
          disks:
33
           - disk:
34
             bus: virtio
35
            name: boot-disk
36
           - disk:
37
             bus: virtio
38
            name: cloudinitdisk
          interfaces:
39
           - bridge: {}
40
41
            name: ovn-net
42
           - sriov: \{\}
43
            macAddress: de:ad:00:00:be:aa
44
            name: sriov-net
45
         machine:
46
          type: q35
47
         resources:
48
          requests:
```

```
49
                                     memory: 4G
50
                          networks:
51
                             - multus:
52
                                     default: true
53
                                    networkName: eks-managed/kube-ovn
54
                                name: ovn-net
55
                             - multus:
56
                                     networkName: eks-managed/sriov-network
57
58
                          volumes:
59
                             - dataVolume:
60
                                     name: centos-sriov-dv
61
                                name: boot-disk
62
                             - cloudInitNoCloud:
63
                                     userData:
64
                                        #cloud-config
65
                                         disable_root: false
66
                                        ssh_pwauth: true
67
                                        users:
68
                                           - name: escore
69
                                               gecos: ES Core User
70
                                               sudo: ALL=(ALL) NOPASSWD:ALL
71
                                               passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6rHe1kcA/PslCnV8P6
72
                                               shell: /bin/bash
73
                                               home: /home/escore
74
                                               lock_passwd: false
75
                                               ssh_pwauth: true
76
                                        runcmd:
77
                                            - dhclient eth1
78
                                     networkData: |
79
                                        version: 2
80
                                        ethernets:
                                            sriov-net:
81
82
                                               set-name: eth0
83
                                               match:
84
                                                   macaddress: 'de:ad:00:00:be:aa'
85
                                               addresses:
86
                                                  - 10.37.1.64/24
87
                                 name: cloudinitdisk
```

NoCloud网络配置格式规范详见: Networking Config Version 1, [Networking Config Version 2] (https://cloudinit.readthedocs.io/en/latest/topics/network-config-format-v2.html)

### 通过cloud-init ConfigDrive方式注入网络配置:

```
1 apiVersion: kubevirt.io/v1alpha3
 2 kind: VirtualMachine
3 metadata:
    name: centos-sriov
5 spec:
 6
     data Volume Templates:\\
      - metadata:
        creationTimestamp: null
8
9
        name: centos-sriov-dv
10
       spec:
11
        pvc:
12
         accessModes:
13
          - ReadWriteMany
14
         resources:
```

```
15
                               requests:
16
                                  storage: 30Gi
17
                           storageClassName: nfs-client
18
19
                           pvc:
                               name: centos-7.9
20
21
                               namespace: nsa
22
               running: true
23
               template:
24
                   metadata:
25
                     labels:
26
                        kubevirt.io/domain: centos-sriov
27
                   spec:
28
                      domain:
29
                        cpu:
30
                           cores: 2
31
                         devices:
                           disks:
32
33
                               - disk:
34
                                     bus: virtio
35
                                 name: boot-disk
36
                               - disk:
37
                                     bus: virtio
38
                                  name: cloudinitdisk
39
                           interfaces:
40
                              - bridge: {}
41
                                 name: ovn-net
42
                               - sriov: {}
43
                                  macAddress: de:ad:00:00:be:aa
44
                                   name: sriov-net
45
                         machine:
46
                           type: q35
                         resources:
47
48
                           requests:
49
                              memory: 4G
50
                      networks:
51
                        - multus:
52
                               default: true
53
                               networkName: eks-managed/kube-ovn
54
                           name: ovn-net
55
                         - multus:
56
                               networkName: eks-managed/sriov-network
57
                           name: sriov-net
58
                      volumes:
59
                        - dataVolume:
60
                               name: centos-sriov-dv
61
                           name: boot-disk
62
                         \hbox{-} {\bf cloud Init Config Drive:} \\
63
                               userData: |
64
                                  #cloud-config
                                 disable_root: false
65
66
                                  ssh_pwauth: true
67
                                  users:
68
                                     - name: escore
69
                                        gecos: ES Core User
70
                                        sudo: ALL=(ALL) NOPASSWD:ALL
71
                                        passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3ChA/P8rgbIbJh3Ch
72
                                        shell: /bin/bash
```

```
73
                                                                                                                  home: /home/escore
                                                                                                                  lock_passwd: false
74
75
                                                                                                                  ssh_pwauth: true
76
                                                                                                 runcmd:
77
                                                                                                         - dhclient eth1
78
                                                                                          networkData: |
79
                                                                                                    \begin{tabular}{ll} \label{tab:condition} \begin{tabular}{ll} \
80
                                                                                 name: cloudinitdisk
```

ConfigDrive网络配置格式规范详见: OpenStack Metadata Service Network: Sample API

#### 创建 Windows 云主机 ⊘

**示例一:** iso 启动

首先需要创建一个卷作为云主机的数据盘,PVC示例 1.pvc-winhd.yaml 如下:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: winhd
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 150Gi
storageClassName: capacity
```

### 云主机示例 1.vm-win2k12-boot-iso.yaml 如下:

```
1 apiVersion: kubevirt.io/v1alpha3
2 kind: VirtualMachine
3 metadata:
4 name: win2k12-boot-iso
5 spec:
6 running: true
7 template:
8
     metadata:
9
       labels:
10
        kubevirt.io/domain: win2k12-boot-iso
11
      spec:
12
       domain:
13
       cpu:
14
        cores: 4
15
        devices:
16
         disks:
17
         - bootOrder: 1
18
          cdrom:
19
           bus: sata
20
          name: cdromiso
21
         - disk:
22
           bus: virtio
23
          name: harddrive
24
         - cdrom:
25
           bus: sata
26
          name: virtiocontainerdisk
27
         interfaces:
28
         - masquerade: {}
29
          name: default
```

```
30
         machine:
31
          type: q35
32
         resources:
33
          requests:
34
           memory: 8G
35
        networks:
36
        - name: default
37
         pod: {}
38
        volumes:
39
        - name: cdromiso
40
         persistentVolumeClaim:
41
          {\color{red} \textbf{claimName:}} \ iso-win-2k12-zh
42
        - name: harddrive
43
         persistentVolumeClaim:
44
          claimName: winhd
45
        - containerDisk:
46
          image: hub.easystack.io/production/virtio-container-disk
47
         name: virtiocontainerdisk
```

容器盘 hub.easystack.io/production/virtio-container-disk 包含了 Windows 云主机所需驱动。

安装过程参考: KubeVirt: installing Microsoft Windows from an ISO

**示例二:** raw 启动

云主机示例 2.vm-win2k19-boot-dv.yaml 如下:

```
1 apiVersion: kubevirt.io/v1alpha3
2 kind: VirtualMachine
3 metadata:
4 name: win2k19-boot-dv
5 spec:
6
     running: true
 7
     template:
8
      metadata:
9
       labels:
10
        kubevirt.io/domain: win2k19-boot-dv
11
      spec:
12
       domain:
13
        cpu:
14
         cores: 4
15
        devices:
         disks:
16
17
         - disk:
18
            bus: virtio
19
          name: boot-disk
20
         interfaces:
21
         - masquerade: {}
22
          name: default
23
        machine:
24
         type: q35
25
        resources:
26
         requests:
27
          memory: 8G
28
       networks:
29
       - name: default
30
        pod: {}
31
       volumes:
32
       - name: boot-disk
33
        dataVolume:
```

```
34
         name: win2k19-dv
35
     dataVolumeTemplates:
36
     - metadata:
37
       name: win2k19-dv
38
      spec:
39
      pvc:
40
       accessModes:
41
       - ReadWriteOnce
42
        resources:
43
       requests:
44
         storage: 50Gi
45
        storageClassName: capacity
46
       source:
47
        pvc:
48
        namespace: default
49
        name: win-2k19-en
```

# 批量创建云主机 ∂

VirtualMachineInstanceReplicaSet 允许你在 KubeVirt 中批量创建和管理虚拟机实例。它具有以下重要字段和功能:

spec.selector:用于选择要管理的虚拟机实例的标签选择器。只有与该标签选择器匹配的虚拟机实例才会被ReplicaSet管理。

spec.template:定义了创建虚拟机实例的模板,包括虚拟机的规格、存储、网络等配置信息。

spec.replicas:指定要创建的虚拟机实例的数量。

status.replicas:表示当前实际运行的虚拟机实例的数量。

当你创建一个 VirtualMachineInstanceReplicaSet 时,KubeVirt 会自动根据 ReplicaSet 的定义,创建指定数量的虚拟机实例。如果有虚拟机实例的数量不符合定义,KubeVirt 会自动进行扩缩容操作,以确保虚拟机实例的数量与定义的数量一致。

你可以使用 kubectl 或其他 Kubernetes 客户端工具,以 YAML 或 JSON 格式定义 VirtualMachineInstanceReplicaSet 对象,并将其应用 到 Kubernetes 集群中。

#### 创建 VirtualMachineInstanceReplicaSet

```
1 apiVersion: kubevirt.io/v1
2 kind: VirtualMachineInstanceReplicaSet
3 metadata:
4 name: ubuntureplicaset
5 spec:
6 replicas: 3 #代表创建3副本虚拟机
     selector:
8
      matchLabels:
9
      kubevirt.io/domain: ubuntu-ovn
10 template:
11
      metadata:
12
       annotations:
13
        ovn.kubernetes.io/allow_live_migration: 'true'
14
15
        kubevirt.io/domain: ubuntu-ovn
16
      spec:
17
       accessCredentials:
       - sshPublicKey:
18
19
          propagationMethod:
20
           configDrive: {}
21
          source:
22
           secret:
23
            secretName: env35
24
       domain:
25
        cpu:
```

```
26
                                  cores: 2
27
                              devices:
28
                                 disks:
29
                                    - disk:
30
                                            bus: virtio
31
                                       name: boot-disk
32
                                     - disk:
33
                                            bus: virtio
34
                                         name: cloudinitdisk
35
                                 interfaces:
36
                                    - name: default
37
                                        bridge: {}
38
                              machine:
39
                                 type: q35
40
                              resources:
41
                                requests:
42
                                     memory: 4G
43
                           networks:
44
                             - name: default
45
                                 multus:
46
                                     default: true
47
                                     networkName: eks-managed/kube-ovn
48
                          volumes:
49
                             - name: boot-disk
50
                                 containerDisk:
51
                                    image: kubevirt/fedora-cloud-container-disk-demo:latest
52
                                     imagePullPolicy: IfNotPresent
                              - cloud In it Config Drive:\\
53
54
                                      userData: |
55
                                         #cloud-config
56
                                         disable_root: false
57
                                        ssh_pwauth: true
58
                                         users:
59
                                           - name: escore
60
                                                gecos: ES Core User
61
                                                sudo: ALL=(ALL) NOPASSWD:ALL
62
                                                passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHFad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyTyvldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHfad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyVldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHfad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyVldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8P6rHe1kcA/passwd: \$6\$xovHfad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyVldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8p6rHe1kcA/passwd: \$6\$xovHfad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyVldqK7hiExVP8rgblbJh3CDXfs2GkmH038g4EW2KPslCnV8p6rHe1kcA/passwd: \$6\$xovHfad3iYjLc801\$jCog26PDU3r4BKBxFS3bejiglTyVldqK7hiExVP8rgblbJh3CDXfs2GkmH03g4EW2KPslCnV8p6rHe1kcA/passwd: \$6\$xovHfad3iYjLc8018jChA/passwd: \$6$xovHfad3iYjLc8018jChA/passwd: \$6$xovHfad3iYjLc8018jChA/passwd: \$6$xovHfad3iYjLc8018jChA/passwd: \$6$xovHfad3iYjLc8018jChA/passwd: \$6$xovHfad3iYjLc8018jChA/passwd: \$6$xovHfad3iYjLc8018jChA/passwd: \$6$xovHfad3iYjLc801
63
                                                shell: /bin/bash
64
                                                home: /home/escore
65
                                                lock_passwd: false
66
                                                ssh_pwauth: true
67
                                  name: cloudinitdisk
```

# 通过容器镜像的方式创建云主机 ⊘

用户可以将VirtualMachineInstance磁盘注入到容器映像中,这种方式可以被KubeVirt运行时使用。磁盘必须放置在容器内的/disk目录下。支持raw和qcow2格式。为了减小容器映像的大小,建议使用qcow2。containerdisks可以并且应该基于scratch。

制作容器镜像 🖉

将本地VirtualMachineInstance磁盘注入到容器映像中,制作成启动镜像。

Dockerfile示例如下

```
FROM scratch
ADD --chown=107:107 ubuntu.qcow2 /disk/
```

# 生成镜像:

1 docker build -t vmidisks/ubuntu:latest .

#### 空硬盘批量创建 🖉

emptyDisk在批量创建云主机时挂载空硬盘

### 例:创建并挂在一个2Gib的云硬盘

```
1 spec:
 2 terminationGracePeriodSeconds: 5
3 domain:
 4
      resources:
5
       requests:
6
        memory: 64M
 7
      devices:
8
       disks:
9
       - name: containerdisk
10
        disk:
        bus: virtio
11
12
       - name: emptydisk
13
        disk:
14
         bus: virtio
15
     volumes:
      - name: containerdisk
16
17
       containerDisk:
18
        image: kubevirt/cirros-registry-disk-demo:latest
19
      - name: emptydisk
20
       emptyDisk:
21
        capacity: 2Gi
```

#### 挂载configmap 🔗

KubeVirt 将会将 ConfigMap 中的内容挂载到虚拟机实例的指定路径中。您可以根据自己的需求,调整文件的路径和名称。

```
1 spec:
 2 domain:
 3
      devices:
 4
        disks:
 5
       - disk:
6
        bus: virtio
        name: containerdisk
       - disk:
8
        bus: virtio
9
10
        name: cloudinitdisk
11
       - disk:
12
        name: app-config-disk
        # set serial
13
14
        serial: CVLY623300HK240D
15
      machine:
16
       type: ""
17
      resources:
18
       requests:
19
        memory: 1024M
20
     termination Grace Period Seconds: 0\\
21
     volumes:
22
     - name: containerdisk
23
       containerDisk:
24
        image: {\it kubevirt/fedora-cloud-container-disk-demo:} latest
25
     - cloud In it No Cloud:\\
26
        userData: |-
```

```
27
         #cloud-config
28
         password: fedora
29
         chpasswd: { expire: False }
30
         bootcmd:
31
        # mount the ConfigMap
32
         - "sudo mkdir /mnt/app-config"
33
          - "sudo mount /dev/$(lsblk --nodeps -no name, serial | grep CVLY623300HK240D | cut -f1 -d' ') /mnt/app-config"
34
       name: cloudinitdisk
      - configMap:
35
36
        name: app-config
37
       name: app-config-disk
```

### 查看云主机实例 ∂

1 kubectl get virtualmachineinstances

### 云主机操作 ⊘

电源管理 ⊘

#### 云主机启动

1 virtctl start centos-net-ovn-boot-dv

#### 云主机关机

1 virtctl stop centos-net-ovn-boot-dv

## 云主机暂停

1 virtctl pause vm centos-net-ovn-boot-dv

### 云主机恢复

1 virtctl unpause vm centos-net-ovn-boot-dv

# 云主机重启

1 virtctl restart centos-net-ovn-boot-dv

### 配置管理 🔗

### 调整规格

- 1 kubectl edit virtualmachines centos-net-ovn-boot-dv
- 2 #编辑 cpu, memory 字段
- 3 virtctl restart centos-net-ovn-boot-dv

## 通过预设调整规格

新建预设: 5.vmipreset-medium.yaml:

```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachineInstancePreset
metadata:
name: c4-g8
spec:
selector:
matchLabels:
kubevirt.io/size: 4C-8G
domain:
```

```
      10
      cpu:

      11
      cores: 4

      12
      resources:

      13
      requests:

      14
      memory: 8G

      15
      devices: {}
```

编辑 spec.template.metadata.labels 字段:

1 kubectl edit virtualmachines centos-boot-dv2

```
template:
metadata:
creationTimestamp: null
labels:
kubevirt.io/domain: centos-boot-dv2
# kubevirt.io/size: 2C-4G
kubevirt.io/size: 4C-8G
```

#### 重启云主机

1 virtctl restart centos-boot-dv2

# 调整启动顺序

1 kubectl edit virtualmachines centos-boot-iso
2 #编辑 bootOrder 字段
3 virtctl restart centos-boot-iso

### 存储管理 🖉

### 热插拔

先创建一个空DV: 3.dv-hotplug.yaml

```
1 apiVersion: cdi.kubevirt.io/v1beta1
2 kind: DataVolume
3 metadata:
4 name: disk-hotplug
5 spec:
6 source:
7 blank: {}
8 pvc:
9
     accessModes:
10
     - ReadWriteOnce
11
    resources:
12
      requests:
13
      storage: 5Gi
     storageClassName: capacity
```

## • 卸载云硬盘

1 virtctl removevolume centos-boot-dv --volume-name=disk-hotplug

### 维护 ⊘

#### 热迁移

1 virtctl migrate centos-boot-dv

#### 冷迁移

云主机关机后,编译VirtualMachine添加以下字段指定目的节点:

```
spec:
template:
spec:
nodeSelector:
kubernetes.io/hostname: <hostname>
```

#### 远程连接 🔗

注意:集群中需要打通相关端口以供外部访问

#### SSH

集群内可通过pod IP直接进行SSH连接(若云主机对接kube-ovn,需要在创建了ovn0设备的节点上访问)

集群外可通过公网IP(EIP),ingress-nginx暴露四层等方式映射22端口以供集群外部访问

#### VNC

通过 virtvnc 服务访问

#### 远程桌面 (Windows)

可通过NodePort, ingress-nginx暴露四层等方式映射3389端口以供集群外部访问

#### 删除云主机 ♂

1 kubectl delete virtualmachines centos-net-ovn-boot-dv

# 监控服务 ≥

kubevirt-prometheus-metrics 是一个用于将 KubeVirt 指标暴露给 Prometheus 的Service,通过在 KubeVirt 中嵌入 Prometheus 客户端库,将 KubeVirt 的指标数据暴露给 Prometheus 服务器。

kubevirt-prometheus-metrics 通过在 KubeVirt 组件中注入 Prometheus 客户端,自动从 KubeVirt 的内部组件(如 virt-api、virt-controller、virt-handler等)收集指标数据,并将其暴露给 Prometheus 进行数据采集。

监控使用prometheus-operator,创建ServiceMonitor将kubevirt暴露给prometheus进行监控,创建 kubevirt-servicemonitor.yml 文件, 内容如下:

```
1 apiVersion: monitoring.coreos.com/v1
2 kind: ServiceMonitor
3 metadata:
4 name: prometheus-kubevirt-metrics
5 namespace: kubevirt
6
    labels:
7
      application: prometheus-servciemonitor
8 spec:
9
     endpoints:
10 - bearerTokenSecret:
11
      key: ""
12
      port: metrics
13
      scheme: https
14
      tlsConfig:
15
       ca: {}
16
       cert: {}
17
       insecureSkipVerify: true
```

- 18
  namespaceSelector:

  19
  matchNames:

  20
   kubevirt

  21
  selector:

  22
  matchLabels:

  23
  app.kubernetes.io/component: kubevirt
- 1 # kubectl create -f kubevirt-servicemonitor.yml

注:metadata中的labels需要与prometheus-operator中定义的LabelSelector相匹配。spec.selector.matchLabels 中的label与endpoint中的label 相匹配。

创建过后在prometheus界面能够看到被发现的kubevirt相关的target,说明监控数据已被采集。

# 卸载服务 🖉

注意:卸载服务之前,需要手动移除所有VirtualMachine资源,并确认删除完成,否则会阻塞卸载

云产品:云原生云主机解除安装

过程略

该部分仅做内部参考

# 补充说明 ♂

# 以下功能有条件支持 ⊘

NA SE
说明
依赖 VolumeSnapshot 能力,该能力并非k8s核心 API,而是 CRDs。需要 CSI drivers 支持并提供这些 能力。
需要 CSI 的 external provisioner 支持 ReadWriteMany 能力
新版本kubevirt提供了 VirtualMachineFlavor,VirtualMachineClusterFlavor 资源定义规格,但该规格仅支持配置cpu相关字段,可定义"通用计算型","计算优化型"两种规格类型,但无法达到ESCloud同等细化程度。当前官方文档还未对该特性进行说明,使用方法仅能从源码中推测。 版本已存在 VirtualMachineInstancePreset 资源可覆盖该部分能力,两者若同时使用推测会产生冲突。 // VirtualMachineInstancePreset 的设计思路(behaving)与Kubernetes原生的PodPresets一致,但PodPresets 机制已在Kubernetes的新版本中被移除(详见:remove pod presets #94090),故有理由怀疑 VirtualMachineInstancePreset 可能在未来版本被弃用然后由 VirtualMachineFlavor,VirtualMachineClusterFlavor 资源顶替。
<b>有一条一个型工的片条。 有限作为</b>

	<pre> (8.12更新) 根据社区最新动态 , VirtualMachineInstancePreset 确认将被弃用 , 当</pre>
编辑镜像配置	需要 CSI的 external provisioner 支持 resize 能力

# 以下功能不支持 🔗

功能点	说明
编辑云主机名称	该功能在v0.41.0版本后移除,详见:Removal of VM rename #5564