

85Kubernetes 系列（七十八）使用 Kubevirt 在 Kubernetes 中部署虚拟机

介绍

Kubevirt 为将微服务与 Kubernetes 集群中的虚拟机集成提供了无缝解决方案。首先，使用 Kubevirt Operator 将 Kubevirt 安装到您的集群上，然后使用 Kubevirt API 定义和创建虚拟机。使用 Kubevirt，您可以像管理任何其他 Kubernetes 资源一样管理虚拟机，使用 `kubectrl` 等工具来监控和修改它们。

除了 RBAC 和网络策略等 Kubernetes 功能外，Kubevirt 还提供了一个用户友好的基于 Web 的界面来管理虚拟机。这使您可以轻松地创建和配置 VM，以及监控它们的性能和资源使用情况。通过使用 Kubevirt 在 Kubernetes 中部署您的虚拟机，您可以享受容器化微服务的可扩展性和可靠性，同时还可以为您的应用程序基础架构利用虚拟机的多功能性和强大功能。

目的

此场景介绍了在 Kubernetes 上创建虚拟机实例 (VMI) 以及如何使用 `virtctl` 与其控制台进行交互。本教程基于 Kubevirt 文档。（<https://kubevirt.io/labs/kubernetes/lab1.html>）

实操

先决条件

- Kubernetes 集群
- Kubevirt 已经安装在 Kubernetes 集群上

安装

创建虚拟机

```

nano vm.yaml

...

apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
  name: testvm
spec:
  running: false
  template:
    metadata:
      labels:
        kubevirt.io/size: small
        kubevirt.io/domain: ubuntu-bionic
    spec:
      domain:
        cpu:
          cores: 1
        devices:
          disks:
            - name: containervolume
              disk:
                bus: virtio
            - name: cloudinitvolume
              disk:
                bus: virtio
          interfaces:
            - name: default
              masquerade: {}
          resources:
            requests:
              memory: 2048M
        networks:
          - name: default
            pod: {}
      volumes:
        - name: containervolume
          containerDisk:
            image: tedezed/ubuntu-container-disk:20.0
        - name: cloudinitvolume
          cloudInitNoCloud:
            userData: |-
              #cloud-config
              chpasswd:
                list: |
                  ubuntu:ubuntu
                  root:toor
                expire: False

```

将清单应用到 Kubernetes

```
kubectl create ns vm
kubectl apply -f vm.yaml -n vm
...output...
virtualmachine.kubevirt.io "testvm" created
virtualmachineinstancepreset.kubevirt.io "small" created
```

管理虚拟机

获取现有虚拟机的列表

```
kubectl get vms -n vm
...output...
NAME      AGE      STATUS    READY
testvm    117s     Stopped   False
.....
kubectl get vms -o yaml testvm -n vm
```

要启动虚拟机，您可以使用：

```
virtctl start testvm -n vm
...output...
VM testvm was scheduled to start
.....

kubectl get vms -n vm
...output...
NAME      AGE      STATUS    READY
testvm    4m27s    Running   True
```

启动后我们可以检查状态VM

```
kubectl get vmis -n vm
NAME      AGE      PHASE      IP              NODENAME          READY
testvm    2m6s     Running    10.10.101.140   kubevirt-worker1  True
```

控制虚拟机状态

关闭它：

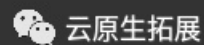
```
virtctl stop testvm -n vm
```

```
root@kubevirt-bastion:~# virtctl stop testvm -n vm
VM testvm was scheduled to stop
root@kubevirt-bastion:~# kubectl get vmis -n vm
NAME      AGE      PHASE      IP              NODENAME          READY
testvm    23m     Running    10.10.101.140   kubevirt-worker1  False
```

要删除虚拟机：

```
kubectl delete vm testvm -n vm
```

```
root@kubevirt-bastion:~# kubectl delete vm testvm -n vm
virtualmachine.kubevirt.io "testvm" deleted
root@kubevirt-bastion:~# kubectl get vmis -n vm
No resources found in vm namespace.
```



总结

总之，Kubevirt 是在 Kubernetes 集群中部署虚拟机的强大工具。通过以这种方式集成微服务和虚拟机，您可以利用容器化应用程序的可扩展性和灵活性，同时还可以利用虚拟化的强大功能和多功能性。借助 Kubevirt，您可以像管理任何其他 Kubernetes 资源一样使用熟悉的工具和界面（例如 kubectl 和基于 Web 的仪表板）管理虚拟机。

此外，Kubevirt 还提供了一系列功能来帮助您充分利用虚拟机，包括对各种操作系统、虚拟化硬件设备和高性能网络的支持。无论您运行的是复杂的企业应用程序还是简单的开发环境，Kubevirt 都可以为您的所有虚拟化需求提供无缝集成的解决方案。因此，如果您希望在 Kubernetes 中部署虚拟机，请务必查看 Kubevirt 并了解它如何帮助您简化基础架构并提高应用程序性能。