113Kubernetes 系列（一零六）通过 Ansible 在树莓派上构建 K3S 集群

前提条件

- 在树莓派上安装 Raspbian 操作系统（64 位），将它们连接到本地网络，为其指定主机名，并启用其 SSH 服务。w3Schools（https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp） 有一个关于如何执行此操作的出色指南。（我更喜欢使用官方的 Raspberry Pi Imager（https://www.raspberrypi.com/software/） 来刷新 SD 卡，而不是 w3schools 使用的 Etcher，但您可以使用其中任何一个。）
- 在本地计算机上安装 Ansible。（我使用 Windows 中的 WSL 2 子系统来运行 Ansible。）
- 安装 kubectl。
- 可选：安装 Helm。

Ansible 设置和组件

为了使用 ansible 配置我们的集群，我们需要一个清单文件、playbook 文件和一些初始 SSH 设置，因为 ansible 使用 SSH 连接到节点。

inventory file

在 Ansible 中，清单文件是一个包含 Ansible 可以管理的目标主机或节点列表的文件。在本例中，它将包含我们所有的树莓派。

```
workers:
  hosts:
    nerminworker1:
      ansible_host: 192.168.0.66
      ansible_user: nermin

masters:
  hosts:

# This hostgroup is designed to only contains the initial bootstrap master node
bootstrapMaster:
  hosts:
    nerminmaster:
      ansible_host: 192.168.0.67
      ansible_user: nermin

pies:
  children:
    bootstrapMaster:
    masters:
    workers:
  vars:
    k3s_version: v1.24.10+k3s1
```

我们的设置中有四个组：

- **bootstrapMaster**：这是第一个获得 k3s 服务的节点，并将创建其他节点用于向集群注册的令牌。
- **masters**：这些是将通过 bootstrapMaster 注册自身的剩余 k3s 服务节点。
- **workers**：这些是将在集群中运行工作负载的 k3s 工作/代理节点。他们还通过 bootstrapMaster 和生成的 k3s 令牌自行注册。
- **pies**：该组包含所有master和worker的组合。

我的设置中有两个树莓派，尽管图片显示了三个。第三个树莓派还有另一个与 k3s 集群无关的用途。两个节点都有用户 nermin 。 nermin 用于在每个主机上执行命令。单个变量 k3s_version 用于确定我们要安装哪个版本的 k3s。

SSH 连接

将公共 SSH 密钥添加到每个树莓派，以避免在创建会话时提示输入密码。

```
$ ssh-copy-id <user>@<host>
```

在这个具体案例中：

```
$ ssh-copy-id nermin@nerminmaster
$ ssh-copy-id nermin@nerminworker1
```

您还可以使用 IP 地址作为主机而不是主机名。

> 注意：确保您已在 WSL2 系统中生成 SSH 密钥对。您可以通过运行来检查： `$ ls ~/.ssh/id_*.pub` .如果您没有看到列出的任何文件，则需要生成 SSH 密钥对。您可以通过运行以下命令来执行此操作： `$ ssh-keygen` 按照提示生成新的密钥对。请确保您的私钥安全，不要与他人共享。

安装 playbook

`install-k3s-playbook.yaml` 文件包含用于在树莓派上安装 k3s master 和workers 的 play。这包括按照文档（https://docs.k3s.io/advanced#raspberry-pi）的要求启用内存 cgroup。 Playbook 还从 bootstrap 节点检索 kubeconfig 文件，并将其放置在名为 k3sconfig 的当前目录中。

```
- name: Enable cgroups
  become: true
  hosts: pies
  tasks:
  - name: Ping hosts
    ansible.builtin.ping:
  - name: Check if cgroups are enabled
    command: cat /boot/cmdline.txt
    register: cmdlineContent
```

```
  - name: Enable cgroups
    command: sed -i -e 's/$/ cgroup_memory=1 cgroup_enable=memory/' /boot/cmdline.txt
    when: "'cgroup_memory=1 cgroup_enable=memory' not in cmdlineContent.stdout"
    notify:
    - Restart pi
  handlers:
  - name: Restart pi
    ansible.builtin.reboot:

- name: Install k3s bootstrap server
  become: true
  hosts: bootstrapMaster
  tasks:
  - name: Ping host
    ansible.builtin.ping:
  - name: Install k3s bootstrap server
    shell: curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION={{ k3s_version }} K3S_NODE_NAME={{ inventory_hostname }} K3S_KUBE
  - name: Extract K3S_TOKEN from server output
    command: cat /var/lib/rancher/k3s/server/node-token
    register: k3s_token
    failed_when: k3s_token is failed or k3s_token.stdout is undefined
  - name: Set K3S_TOKEN as a fact
    set_fact:
      k3s_token: "{{ k3s_token.stdout }}"

- name: Install k3s servers
  become: true
  hosts: masters
  tasks:
  - name: Ping hosts
    ansible.builtin.ping:
  - name: Install k3s servers
    shell: curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION={{ k3s_version }} K3S_URL=https://{{ hostvars['nerminmaster']['ar

- name: Install k3s workers
  become: true
  hosts: workers
  tasks:
  - name: Ping hosts
    ansible.builtin.ping:
  - name: Install k3s workers
    shell: curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION={{ k3s_version }} K3S_URL=https://{{ hostvars['nerminmaster']['ar

- name: Fetch k3s kubeconfig
  become: true
  hosts: bootstrapMaster
  tasks:
  - name: Fetch kubeconfig
    fetch:
      src: /etc/rancher/k3s/k3s.yaml
      dest: k3sconfig
      flat: true
```

卸载 playbook

`uninstall-k3s-playbook.yaml` 卸载每个树莓派上的 k3s 服务和脚本。这使得清理变得容易。

```
- name: Uninstall k3s on workers
  become: true
  hosts: workers
  tasks:
  - name: Ping hosts
    ansible.builtin.ping:
  - name: Uninstall k3s agent
    command: /usr/local/bin/k3s-agent-uninstall.sh

- name: Uninstall k3s on servers
  become: true
  hosts: masters
  tasks:
  - name: Ping hosts
    ansible.builtin.ping:
  - name: Uninstall k3s server
    command: /usr/local/bin/k3s-uninstall.sh

- name: Uninstall k3s on bootstrap servers
  become: true
  hosts: bootstrapMaster
  tasks:
  - name: Ping hosts
    ansible.builtin.ping:
  - name: Uninstall k3s server
    command: /usr/local/bin/k3s-uninstall.sh
```

创建 K3S 集群

现在是有趣的部分。通过执行 install playbook 创建 k3s 集群：

```
$ ansible-playbook -i inventory.yaml install-k3s-playbook.yaml
```

就是这样！输出应如下所示（我的树莓派 启用了 cgroup，因此跳过该任务）：

```
PLAY [Enable cgroups] *********************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
ok: [nerminworker1]
ok: [nerminmaster]

TASK [Ping hosts] *************************************************************************************************************
ok: [nerminmaster]
ok: [nerminworker1]

TASK [Check if cgroups are enabled] *******************************************************************************************
changed: [nerminworker1]
changed: [nerminmaster]

TASK [Enable cgroups] *********************************************************************************************************
skipping: [nerminmaster]
skipping: [nerminworker1]

PLAY [Install k3s bootstrap server] *******************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
ok: [nerminmaster]

TASK [Ping host] **************************************************************************************************************
ok: [nerminmaster]

TASK [Install k3s bootstrap server] *******************************************************************************************
changed: [nerminmaster]

TASK [Extract K3S_TOKEN from server output] ***********************************************************************************
changed: [nerminmaster]

TASK [Set K3S_TOKEN as a fact] ************************************************************************************************
ok: [nerminmaster]

PLAY [Install k3s servers] ****************************************************************************************************
skipping: no hosts matched

PLAY [Install k3s workers] ****************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
ok: [nerminworker1]

TASK [Ping hosts] *************************************************************************************************************
ok: [nerminworker1]

TASK [Install k3s workers] ****************************************************************************************************
changed: [nerminworker1]

PLAY [Fetch k3s kubeconfig] ***************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
ok: [nerminmaster]

TASK [Fetch kubeconfig] *******************************************************************************************************
changed: [nerminmaster]

PLAY RECAP ********************************************************************************************************************
nerminmaster               : ok=10   changed=4    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
nerminworker1              : ok=6    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Playbook 将 kubeconfig 文件从 bootstrap master 下载到本地计算机上的当前目录，名称为 k3sconfig。该文件如下所示：

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: ...
    server: https://127.0.0.1:6443
  name: default
contexts:
- context:
    cluster: default
    user: default
  name: default
current-context: default
kind: Config
preferences: {}
users:
- name: default
  user:
    client-certificate-data: ...
    client-key-data: ...
```

将 server 值中的 localhost IP 替换为引导主节点的 IP 或主机名。

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: ...
    server: https://nerminmaster:6443  # <----- This value here.
  name: default
contexts:
- context:
    cluster: default
    user: default
  name: default
current-context: default
kind: Config
preferences: {}
users:
- name: default
  user:
    client-certificate-data: ...
    client-key-data: ...
```

验证集群是否已启动并正在运行：

```
$ kubectl get nodes --kubeconfig k3sconfig
NAME           STATUS   ROLES                       AGE     VERSION
nerminmaster   Ready    control-plane,etcd,master   8m42s   v1.24.10+k3s1
nerminworker1  Ready    <none>                      8m7s    v1.24.10+k3s1
```

成功运行！

可选：将 rancher 服务器部署到 k3s

如果我们有一个可以用来查看和管理集群的 UI，那就太好了。 Rancher 服务器带有这个 UI！

要部署 Rancher 服务，请执行以下命令：

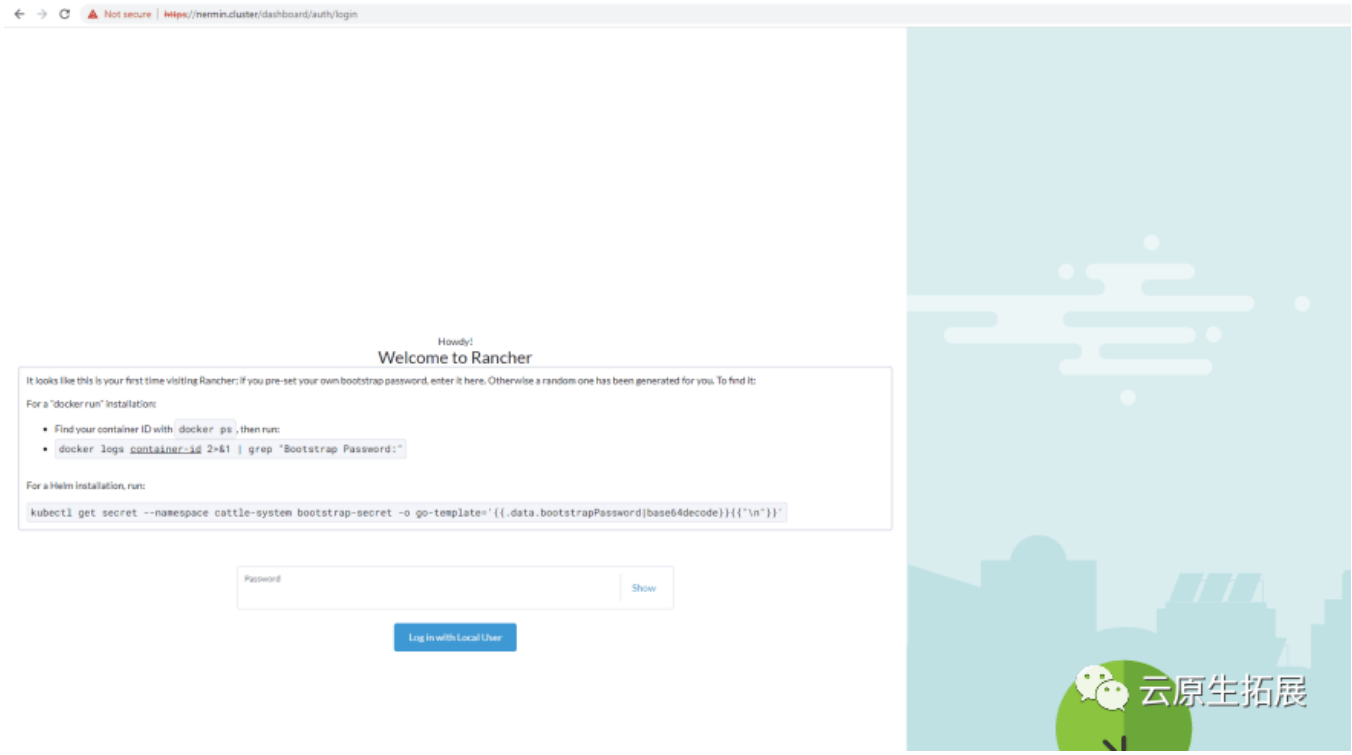
```
## CERT MANAGER

# Add jetstack helm repo
helm repo add jetstack https://charts.jetstack.io
helm repo update

# Create cert-manager namespace
kubectl create namespace cert-manager --kubeconfig k3sconfig

# Install cert-manager
helm upgrade --install cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --version v1.8.2 \
  --set installCRDs=true \
  --wait \
  --kubeconfig k3sconfig



## RANCHER SERVER

# Create cattle-system namespace
kubectl create ns cattle-system --kubeconfig k3sconfig

# Add rancher-latest helm repo
helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
helm repo update

# Install rancher
helm upgrade --install rancher rancher-latest/rancher \
  --version 2.7.0 \
  --namespace cattle-system \
  --set hostname=nermin.cluster \
  --set replicas=1 \
  --set rancherImageTag=v2.7.2-rc5-linux-arm64 \
  --wait \
  --kubeconfig k3sconfig
```

在主机文件中创建一条指向工作节点 IP 的记录。在 Windows 上，此文件位于 C:/Windows/System32/drivers/etc/hosts 。在文件中添加记录：

```
192.168.0.66   nermin.cluster
```

请注意，该域名与 Rancher Chart 中的 hostname 值相同。

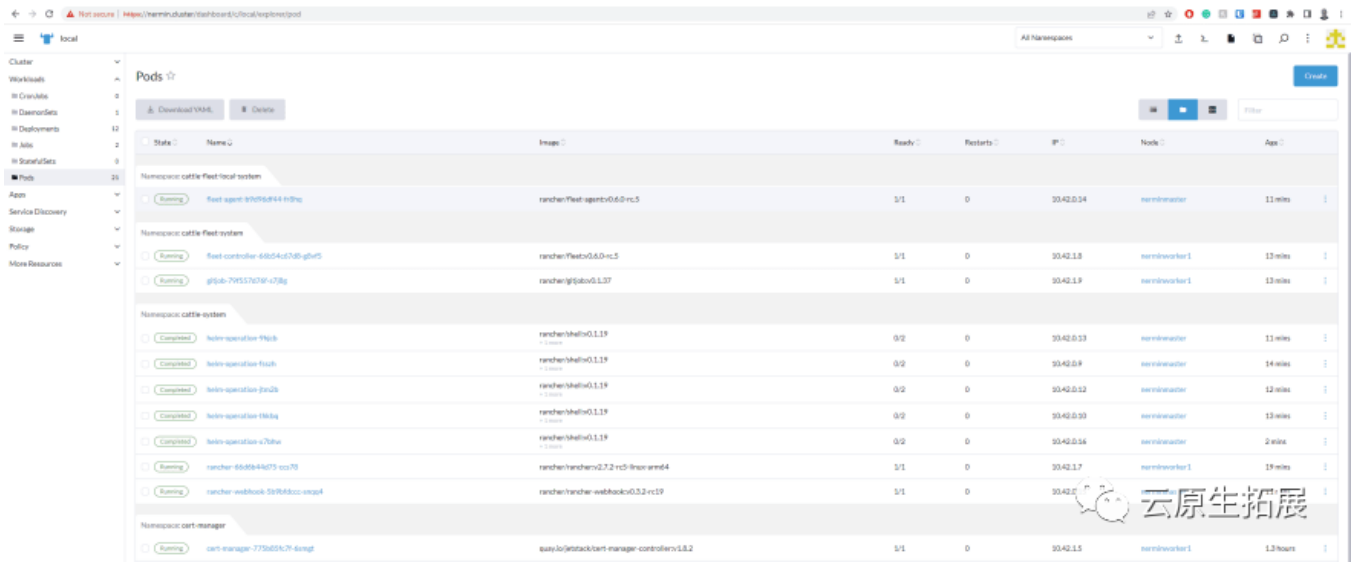
转到域名 nermin.cluster 。没有受信任的证书，因此浏览器会警告我们不要访问不受信任的站点。按“高级”，然后继续到 nermin.cluster（不安全）。现在，您将进入登录页面。



要获取密码，请运行：

```
$ kubectl get secret \
  --namespace cattle-system bootstrap-secret \
  -o go-template='{{.data.bootstrapPassword|base64decode}}{{"\n"}}' \
  --kubeconfig k3sconfig

qstl5v7n4j6tmmr89rnhhb6qcf69krfljnwxm96sds2xpgbm4zp4zf
```

将密码复制并粘贴到输入字段中，您将被允许进入 Rancher 服务 UI。



清理

运行 uninstall-k3s-playbook 以从所有节点中删除 k3s：

```
$ ansible-playbook -i inventory.yaml uninstall-k3s-playbook.yaml
```