Rancher & Terraform

Christian Frank
December 14th
HashiTalks: DACH

Cover artwork © Pascal Campion
Experience

- Debian
- Kubernetes
- Docker
Cluster Creation
Options

There are three options to create a Kubernetes cluster with Rancher on Amazon Web Services:

• Managed Kubernetes (EKS)
• Rancher Node-Driver
• Custom nodes
Rancher Node-Driver

• Within a Terraform plan, define the Rancher provider
• Optional: In the Rancher cluster definition, define the AWS cloud provider within the RKE cluster options
• Optional: Create the necessary IAM policies
• Create appropriate cloud credentials
• Create node templates
• Create the cluster (using rancher-machine)

➤ Note: The creation of RKE2 / k3s Clusters will be using the Cluster API and Fleet (currently in tech preview)
Terraform Provider

Rancher2 Provider

The Rancher2 provider is used to interact with the resources supported by Rancher v2.

The provider can be configured in 2 modes:

- Admin: this is the default mode, intended to manage rancher2 resources. It should be configured with the `api_url` of the Rancher server and API credentials, `token_key` or `access_key` and `secret_key`.

- Bootstrap: this mode is intended to bootstrap a rancher2 system. It is enabled if `bootstrap = true`. In this mode, `token_key` or `access_key` and `secret_key` cannot be provided. More info at `rancher2_bootstrap` resource.

Example Usage

https://registry.terraform.io/providers/rancher/rancher2/latest/docs
Rancher Provider

# Rancher
provider "rancher2" {
  api_url = var.rancher-url
  token_key = var.rancher-token
}

# Rancher cloud credentials

```hcl
resource "rancher2_cloud_credential" "credential_ec2" {
  name = "EC2 Credentials"
  amazonec2_credential_config {
    access_key = var.ec2-access-key
    secret_key = var.ec2-secret-key
  }
}
```
## EC2 Credentials

### Cloud Credentials

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>API Key</th>
<th>Description</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EC2 Credentials</td>
<td>AKIAI562FX745QI06MXE</td>
<td></td>
<td>59 mins</td>
</tr>
</tbody>
</table>

**Amazon**
Node Templates

```hcl
# Rancher node template
resource "rancher2_node_template" "template_ec2" {
  name = "EC2 Node Template"
  cloud_credential_id = rancher2_cloud_credential.credential_ec2.id
  engine_install_url = var.dockerurl

  amazonec2_config {
    ami = var.image
    region = var.ec2-region
    security_group = [var.ec2-secgroup]
    subnet_id = var.ec2-subnet
    vpc_id = var.ec2-vpc
    zone = var.ec2-zone
    root_size = var.disksize
    instance_type = var.type
  }
}
```
## Node Templates

<table>
<thead>
<tr>
<th>State</th>
<th>Name</th>
<th>Owner</th>
<th>Provider</th>
<th>Location</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EC2 Node Template</td>
<td>Default Admin</td>
<td>Amazon EC2</td>
<td>eu-central-1a</td>
<td>m3.xlarge</td>
</tr>
</tbody>
</table>

Owner: Default Admin
Node Pool

```terraform
# Rancher node pool
resource "rancher2_node_pool" "nodepool_ec2" {
  cluster_id = rancher2_cluster.cluster_ec2.id
  name = "nodepool"
  hostname_prefix = "rke-${random_id.instance_id.hex}-"
  node_template_id = rancher2_node_template.template_ec2.id
  quantity = var.numnodes
  control_plane = true
  etcd = true
  worker = true
}
```
Node Pool

<table>
<thead>
<tr>
<th>Name</th>
<th>Instance ID</th>
<th>Instance state</th>
<th>Instance type</th>
<th>Status check</th>
<th>Alarm status</th>
<th>Availability Zone</th>
<th>Public IPv4 DNS</th>
<th>Public IPv4 DNS</th>
<th>Private IPv4 DNS</th>
<th>Elastic IP addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>rke-ecbb4c-3</td>
<td>i-0c2551923c3f0501a</td>
<td>Running</td>
<td>m3.xlarge</td>
<td>2/2 checks passed</td>
<td>No alarms</td>
<td>eu-central-1a</td>
<td>ec2-54-95-73-173.eu-central-1.amazonaws.com</td>
<td>54.119.216.37</td>
<td>172.31.19.238</td>
<td>ip-172-31-19-238.eu-central-1.compute.amazonaws.com</td>
</tr>
</tbody>
</table>
Kubernetes Cluster

```hcl
# Rancher cluster
code = "rancher2_cluster" "cluster_ec2" {
    name = "ec2-${random_id.instance_id.hex}"
    description = "Terraform"

    rke_config {
        kubernetes_version = var.k8version
        ignore_docker_version = false
        network {
            plugin = "flannel"
        }
    }
}
```
# Kubernetes Cluster

## Cluster Dashboard

### Terraform
- Provider: RKE1
- Kubernetes Version: v1.21.5
- Created: 25 mins ago

### Total Resources
- Total: 58

### Nodes
- Count: 3

### Deployments
- Count: 11

### Capacity

<table>
<thead>
<tr>
<th>Component</th>
<th>Used</th>
<th>Reserved</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pods</td>
<td>3/31</td>
<td>0/12</td>
<td>14.76%</td>
</tr>
<tr>
<td>Cores</td>
<td>0.7/12</td>
<td>0.6/12</td>
<td>25.58%</td>
</tr>
<tr>
<td>Memory</td>
<td>3.6/41 GB</td>
<td>0.6/4 GB</td>
<td>4.68%</td>
</tr>
</tbody>
</table>

### Alerts
- Name: Watchdog
- Message: This is an alert meant to ensure that the entire alerting pipeline is functional. This alert is always firing, therefore it should always be firing in Alertmanager and always fire against a receiver. There are integrations with various notification mechanisms that send a notification when this alert is not firing. For example, the "DeadManSwitch" integration in PaperDuty.
Kubernetes Cluster

### Resources

<table>
<thead>
<tr>
<th>NAME</th>
<th>PROVIDER</th>
<th>TYPE</th>
<th>MODULE</th>
<th>UPDATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>before</td>
<td>hashicorp/nul</td>
<td>null_resourc...</td>
<td>root</td>
<td>Oct 21 2021</td>
</tr>
<tr>
<td>cluster_ec2</td>
<td>rancher/rancher...</td>
<td>rancher2_clis...</td>
<td>root</td>
<td>Oct 21 2021</td>
</tr>
<tr>
<td>credential_ec2</td>
<td>rancher/rancher...</td>
<td>rancher2_clis...</td>
<td>root</td>
<td>Oct 21 2021</td>
</tr>
<tr>
<td>delay</td>
<td>hashicorp/nul</td>
<td>null_resourc...</td>
<td>root</td>
<td>Oct 21 2021</td>
</tr>
<tr>
<td>instance_id</td>
<td>hashicorp/rand...</td>
<td>random_id...</td>
<td>root</td>
<td>Oct 21 2021</td>
</tr>
</tbody>
</table>

### Metrics

Metrics will appear once your next run is applied.

### Tags (0)

Tags have not been added to this workspace.

### Run Triggers

(0)
Sample Terraform plans

https://github.com/chfrank-cgn/Rancher

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Last Updated</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>ala-cluster</td>
<td>Azure update to Rancher 2.5.5</td>
<td>11 months ago</td>
<td></td>
</tr>
<tr>
<td>ac-cluster-1</td>
<td>Alert Manager</td>
<td>17 days ago</td>
<td></td>
</tr>
<tr>
<td>ac-cluster-2</td>
<td>Alert Manager</td>
<td>17 days ago</td>
<td></td>
</tr>
<tr>
<td>ac1-cluster-1</td>
<td>EKS Cluster</td>
<td>10 minutes ago</td>
<td></td>
</tr>
<tr>
<td>ac2-cluster-1</td>
<td>EKS Cluster</td>
<td>10 minutes ago</td>
<td></td>
</tr>
<tr>
<td>gcp-cluster</td>
<td>Rancher 2.6.0, Terrafom 1.0.7</td>
<td>last month</td>
<td></td>
</tr>
<tr>
<td>gcp-nfs-helm2</td>
<td>NFS and Helm</td>
<td>2 years ago</td>
<td></td>
</tr>
<tr>
<td>gcp-nfs-helm3</td>
<td>Terrafom 1.0</td>
<td>3 months ago</td>
<td></td>
</tr>
<tr>
<td>lo-cluster</td>
<td>OpenStack on leafcloud</td>
<td>10 months ago</td>
<td></td>
</tr>
<tr>
<td>openstack</td>
<td>OpenStack config file</td>
<td>2 years ago</td>
<td></td>
</tr>
<tr>
<td>ovf-Cluster</td>
<td>OVF Cluster</td>
<td>2 years ago</td>
<td></td>
</tr>
<tr>
<td>vm-cluster</td>
<td>VMware cluster</td>
<td>8 months ago</td>
<td></td>
</tr>
<tr>
<td>qattributes</td>
<td>Initialized</td>
<td>2 years ago</td>
<td></td>
</tr>
<tr>
<td>CODE_OF_CONDUCT.md</td>
<td>Initialized</td>
<td>2 years ago</td>
<td></td>
</tr>
<tr>
<td>LICENSE</td>
<td>Initiated</td>
<td>2 years ago</td>
<td></td>
</tr>
<tr>
<td>README.md</td>
<td>OpenStack on leafcloud</td>
<td>10 months ago</td>
<td></td>
</tr>
<tr>
<td>README.md.backup</td>
<td>AKS Cluster</td>
<td>2 years ago</td>
<td></td>
</tr>
</tbody>
</table>

https://github.com/chfrank-cgn/Rancher
Thank you
cfrank@chfrank.net