Deploying into RKE

Unit 2.2.1

Deploying into RKE

Choose a Hostname for the Rancher Server

Select a hostname for the Rancher server environment. The IP address you attach to this hostname later must be reachable by downstream clusters. Although it's theoretically possible to change this hostname after installation, doing so creates instability in the environment. It's far better to choose a name like `rancher.mydomain.com` and keep that hostname for the life of the environment.

Provision an RKE Cluster

We've already shown you how to deploy an RKE cluster in, and in that section we discussed how RKE's flexibility means that you can deploy a single-node cluster and add more nodes later to make it HA.

This installation process has more steps than the Docker container method, but if you're considering using Rancher in production and want to have the most options available for using it in the future, you can follow all of the steps here but with a single node RKE cluster. It won't be highly available, but you can convert it later.

Deploy a Load Balancer in Front of the Cluster

Even with a multi-node RKE cluster, you'll be accessing the Rancher server via a single URL. The ingress controller will listen on all nodes in the cluster, so deploy a load balancer to receive traffic for your chosen URL and route it to the hosts in the cluster.

This can be a hardware load balancer, a cloud load balancer like an ELB or NLB, or a software load balancer like Nginx or HAProxy.

Whatever you choose, operate the load balancer at Layer 4 only, passing TCP directly through on 80 and 443. Do not configure the load balance at Layer 7 for HTTP and HTTPS.

Although it's possible to configure TLS termination on the load balancer and communicate between the load balancer and the Rancher server over an unencrypted connection on port 80, this is not a recommended method of install. More information on how to do this is available in our documentation.

This load balancer will be dedicated to the Rancher server cluster, which will appear inside of Rancher as the “local” cluster. Workloads other than Rancher should never run on the “local” cluster, so each downstream Kubernetes cluster that Rancher manages will need its own load balancer.

After the load balancer has been deployed, configure DNS to point your chosen hostname to the addresses or hostnames of the load balancer, according to the instructions for your chosen architecture.

Alternate Options For Cloud Native Environments

If your cluster is located on premises or in an environment where you can dynamically attach multiple IP addresses to a node, you can look at MetalLB as an option. MetalLB enables a service of type LoadBalancer on the cluster and assigns a dedicated IP to the service. This service would sit in front of the Ingress Controller and pass traffic to it the same way an external load balancer would.

Install Rancher With Helm

Rancher is installed with Helm, the package manager for Kubernetes. These instructions cover Helm 3, the latest version of Helm at the time of writing. If needed, instructions for Helm 2 are available in the Rancher documentation.

All of these steps take place on the system where you ran rke up to install RKE, and where you have the configuration file for kubectl. Install Helm and verify that you're pointed at the correct cluster by running kubectl get nodes before continuing.

Add the Helm Chart Repository

Rancher provides three repositories for Helm charts:

- Latest: Recommended for trying out the newest features. Not recommended for production.
- Stable: Recommended for production environments
- Alpha: Experimental previews of upcoming releases. Definitely not recommended for production.

If you use an Alpha release, there is no support for upgrading to, from, or between releases. Alpha releases are designed to be viewed and then deleted.

When you've decided the repository to use, add it to your helm repository list.

Create a Namespace for Rancher

Rancher will be installed into the cattle-system namespace, which must exist before we install the chart.

Choose Your SSL Configuration

Rancher will always be protected by TLS, and you have three options for how to provision this component:

1. Rancher-generated self-signed certificates
2. Real certificates from Let's Encrypt
3. Certificates that you provide (real or self-signed)

The first two options require a Kubernetes package called cert-manager that handles certificate generation and renewal from external sources. If you're using your own certificates, you can skip the next step.

*Install **cert-manager***

Cert Manager is under active deployment by a company called JetStack, so the best instructions for installing it are available from their site.

Install Rancher

Option 1: Rancher-Generated Self-Signed Certificates

This is the default option when installing Rancher and requires no additional configuration. You only need to specify the namespace and the hostname when installing.

This installation option requires two parameters:

```
--set hostname=rancher.mydomain.com
--namespace cattle-system
```

Option 2: Real Certificates From Let's Encrypt

To request a certificate from Let's Encrypt you must have the load balancer and hostname properly configured. Let's Encrypt will issue an http-01 challenge that cert-manager will process. Certificates issued by Let's Encrypt will be automatically renewed before their expiration date.

In addition to the parameters listed in Option 1, this installation option requires two additional parameters:

```
--set ingress.tls.source=letsEncrypt
--set letsEncrypt.email=you@domain.com
```

Let's Encrypt uses the email to communicate with you about any issues with the certificate, such as its upcoming expiration. Please use a real email address for this parameter.

Option 3: Certificates That You Provide

If you have your own certificates, either from a public or private CA, you will load these into a Kubernetes Secret and tell Rancher and the Ingress Controller to use that secret to provision TLS.

In addition to the parameters listed in Option 1, this option requires the following additional parameter:

```
--set ingress.tls.source=secret
```

If your certificates are signed by a private CA (or self-signed), you will also need to provide:

```
--set privateCA=true
```

After initiating the install, you'll need to create the secrets for the TLS certificates before the install will complete.

1. Create a file called tls.crt with the certificate
2. Create a file called tls.key with the private key
3. Create a secret called tls-rancher-ingress from those files.
4. This secret is of type tls

If using a private CA, create a file called cacerts.pem with the private CA information. Then create a secret called tls-ca from this file. This secret is of type **generic.**

Verify the Rancher Installation

Verify the installation by watching the rollout status of the rancher deployment in the cattle-system namespace.

References

Installing Rancher into Kubernetes - https://rancher.com/docs/rancher/v2.x/en/installation/k8s-install/

TLS Termination on Load Balancer - https://rancher.com/docs/rancher/v2.x/en/installation/options/single-node-install-external-lb/

Install Rancher with Helm 2 - https://rancher.com/docs/rancher/v2.x/en/installation/options/helm2/

MetalLB - https://metallb.org

Cert Manager - http://docs.cert-manager.io/

JetStack - https://www.jetstack.io/