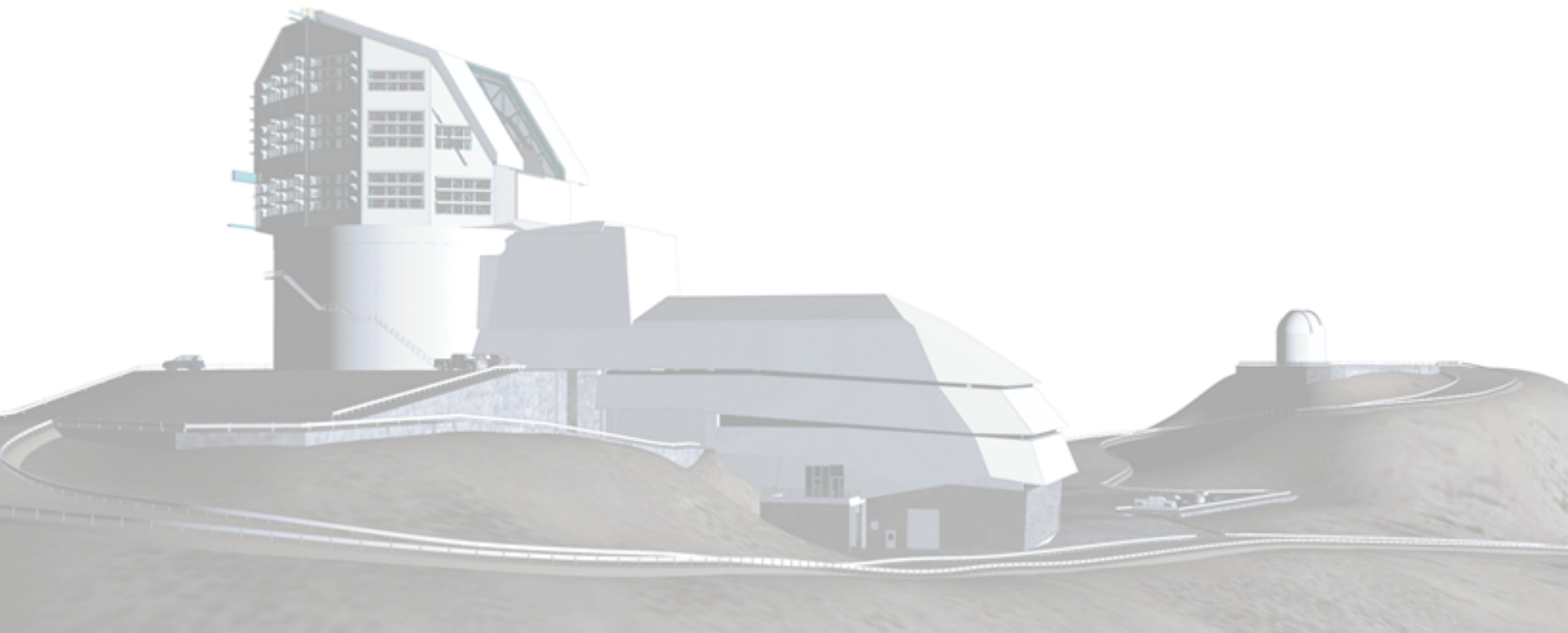
Vera C. Rubin Observatory
Rubin Observatory Project Office

# rke Update Playbook

**Carlos Barría**

**ITTN-071**

**Latest Revision: 2023-05-10**



## Abstract

following document details the procedure to update the current version of ‘rke‘ into a newer version with the objective of moving forward with the Kubernetes version installed in the designated cluster.

# Change Record

| Version | Date | Description | Owner name |
|---|---|---|---|
| 1 | YYYY-MM-DD | Unreleased. | Carlos Barría |

*Document source location:* `https://github.com/lsst-it/ittn-071`

# Contents

# rke Update Playbook

## 1 Introduction

In the following document we will explain step by step the current procedure to upgrade the version of Kubernetes through rke (Rancher Kubernetes Engine), which is the current tool we use to assemble our Kubernetes Clusters.

It is very important to try to keep up with the current version of this software as it can:

- Apply security patches to the software.
- Apply bug fixes to solve issues from the past versions
- Improve performance in some cases.
- Add new features that could improve utilization of the product.

## 2 Playbook

1. First, we need to open a Jira Ticket to attach our work and set the state to In Progress.
2. Clone or update your current version of both repositories in lsst github account lsst-control and lsst-puppet-hiera-private.
3. Create a branch on both repositories with the IT ticket number you created. *(ex: IT-3956_rke_updates)*
4. Update the rke version to match the desired Kubernetes version.
   (a) Check the rke github repository for the release information rke releases *Note: rke and k8s versions are different, you need to check that the version of rke supports the version of k8s you need to upgrade to.*
   (b) In the rke releases page, search for the kubernetes version you want to upgrade and notice which rke version supports it. *ex: Kubernetes version v1.25.9-rancher2-1 needs rke version v1.4.6-rc2 use:* `rke config -l -a` *to see supported versions on the current installed rke*
   (c) Update the rke version in the hierarchy on the lsst-control repository and push those changes. (ex: rke.yaml)
   (d) If this is a new version of rke that's never been used, we need to add the checksum of such binaries to the *profile::core::rke* module. (ex: rke.pp)
5. Once the changes have been pushed, change all the hosts that belong to the cluster into the environment of your branch.
   (a) Search with `params.role = rke` select the cluster hosts.
   (b) Force puppet to run in those nodes.
6. Connect to the head node of the cluster via ssh.
7. Switch into the user rke with `sudo -iu rke`
8. Pre-updates sanity check:
   (a) Verify the nodes of the cluster are online and ready. `kubectl get nodes`
   (b) Check new version of the rke client. `rke --version`
   (c) If the cluster has *rook-ceph* deployed check on the health status.

9. Switch to the *k8s-cookbook* repository and update the *cluster.yaml* file in the *rke* folder of the cluster.
   (a) Update your local repository or clone the *k8s-cookbook*.
   `ex: git clone https://github.com/lsst-it/k8s-cookbook`
   (b) Create a branch on this repository with the same name used in the control repository. `git checkout -b IT-3956_rke_updates`
   (c) Edit the cluster.yaml file inside the rke folder on the selected k8s cluster and change it to the desired version. `ex: kubernetes_version: "v1.25.9-rancher2-1"`
   (d) Stage the file, commit your changes and push them into the remote branch.
   `ex: git add cluster.yaml; git commit -m "(ruka) bump k8s to v1.59.9-rancher2-1"; git push`
   (e) Log into the cluster and switch to the rke user. `ex: sudo -iu rke`
   (f) Update the current repository and switch to your branch.
   `ex: git fetch --all --prune; git checkout IT-3956_rke_updates`
   (g) Ensure the repository is clean with no local changes and the commit is correct.
   `ex: git status; git log -p`
10. Announce the update on the relevant Slack channel before proceeding further.
11. Set yourself on the rke folder on the selected cluster and run:
    (a) Save current config with `make tarball`.
    (b) Create a snapshot of the current etcd `make snapshot`.
    (c) Update the version. `rku up`
    (d) Watch the nodes get updated. `kubectl get nodes -w`
    (e) When the process is done is good to run `rke up` again to check the process was idempotent.
12. Validate and check status on the following:
    (a) Pods rollouts and broken states with `kubectl get pods -A -w` *Note: canal changes may take a while to rollout*
    (b) Deployments on the kube-system namespace with `kubectl -n kube-system get deployments`
    (c) Deamonset status on the kube-system ns with `kubectl -n kube-system get deamonset`
    (d) Check the status of the rook-ceph deployment if the cluster has one. *Note: Through the dashboard or inside the rook-ceph-tool pod with* `ceph status`

13. Grab the status of the cluster and report back into the JIRA Ticket.
14. Announce on the correct Slack channel that the update is complete.
15. Open both branches pull request to merge this new changes into the production branch.
    *Note: PRs on k8s-cookbook and lsst-control repositories*
16. Merged both changes into production and delete the branch created in the `lsst-puppet-hiera-private`
17. If the cluster was in the summit then roll forward the `cp_production` branch.
18. Go back into foreman and change the corresponding hosts back to the production environment.
19. Finally, close the Jira Ticket.

# A References

# B Acronyms

| Acronym | Description |
|---|---|
| DM | Data Management |