

# Introduction

This document describes Kubernetes Dashboard release procedure and version maintenance.

## Release procedure

So you want to release a new version of Kubernetes Dashboard? Great, you just need to follow the steps below.

1. Send a pull request that increases Kubernetes Dashboard version number in `build/conf.js`. The property name is `deploy.version.release`. Follow versioning guidelines. Keep `package.json` and `src/deploy/kubernetes-dashboard.yaml` versions in sync.
2. Get the pull request reviewed and merged.
3. Create a git release tag for the merged pull request. Release description should include changelog.
4. Build and push production images to container registry. Use `gulp push-to-gcr:release`.
5. Update addons on the Kubernetes core repository. Dashboard addon directory is here. If the update is minor, all that needs to be done is to change image version number in the main controller config file (`dashboard-controller.yaml`), and other configs, as described in the header of the config. If the release is major, this needs coordination with Kubernetes core team and possibly alignment with the schedule of the core.
6. Also update addon config in the minikube repo

## Versioning guidelines

Kubernetes Dashboard versioning follows semver in spirit. This means that it uses `vMAJOR.MINOR.PATCH` version numbers, but uses UX and consumer-centric approach for incrementing version numbers.

1. Increment MAJOR when there are breaking changes that affect user's workflows or the UX gets major redesign.
2. Increment MINOR when new functionality is added or there are minor UX changes.
3. Increment PATCH in case of bug fixes and minor new features.

Versions `0.X.Y` are reserved for initial development and may not strictly follow the guidelines.