

Smooth Sailing with Helm v3

Brandon B. Jozsa / Solution Architect
@v1k0d3n

Follow us on:



Agenda ➤

- History of Helm
- Overview of Helm v2 (Tiller)
- Changes in Helm v3
 - Security
 - Operations
 - Developer
- What's Available in v3 beta
- Demonstration/Walkthrough

Changes in Helm v3

Operations:

- Tiller Removed
- Security Enhancements
- Release Model has Changed (Deployment State)
- Helm Controller

Development:

- CRD Handling
- Library Chart
- Lua Introduced

Target Audience

General Information:


- Looking for general knowledge on changes from Helm v2 to Helm v3
- Looking for information on the increased security model of Helm v3

Intermediate Users:

- How to improve the security model for existing Helm v2 implementations
- Information on how to build a Helm Chart in Helm v3

Advanced Workflows:

- How to leverage Helm v3 libraries for improved scale and security



**Shown
in
Demo!**

Most of this Meetup discussion will be interactive and driven via demo/walkthroughs.

What is Helm?

Is it a package manager?

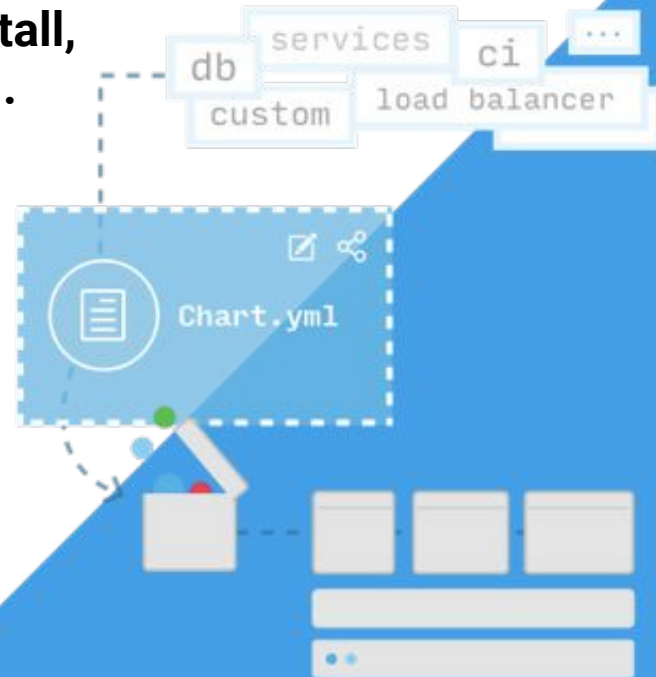
**Well, it does handle install,
upgrade and delete...**



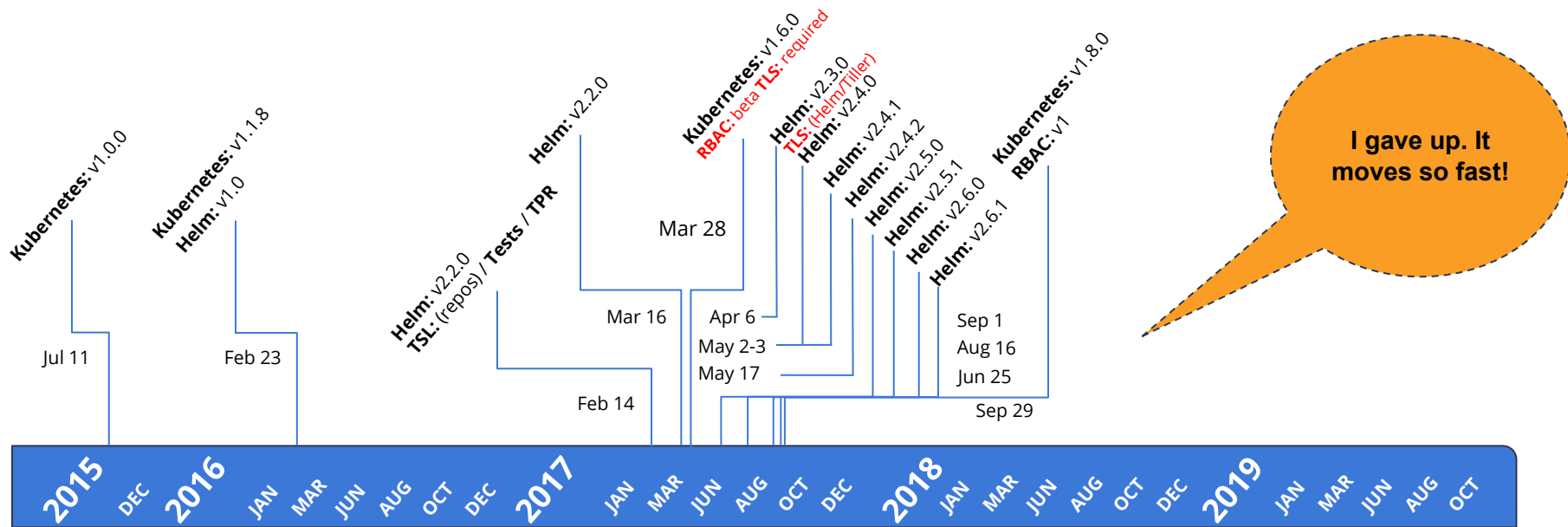
**Mustache...really?
yup.**

**Golang?
Sort of?**

**Is it a template rendering engine?
More than anything, yes. This.**

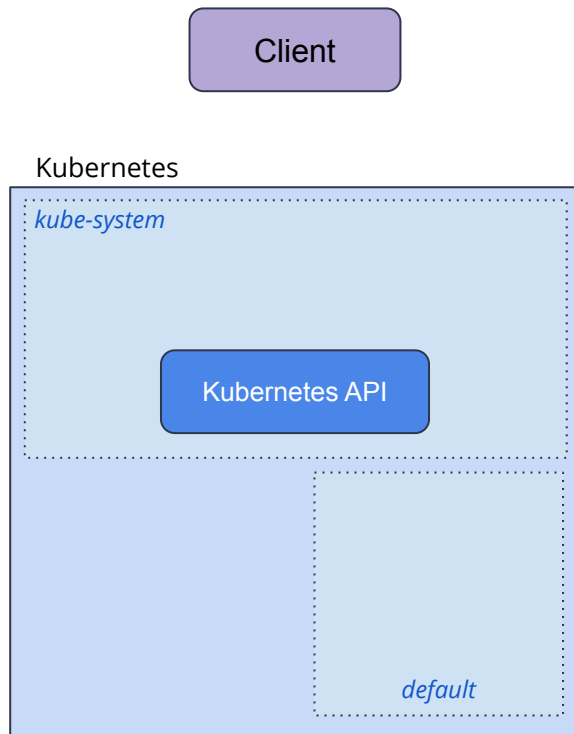


Timeline of Helm (and Kubernetes)

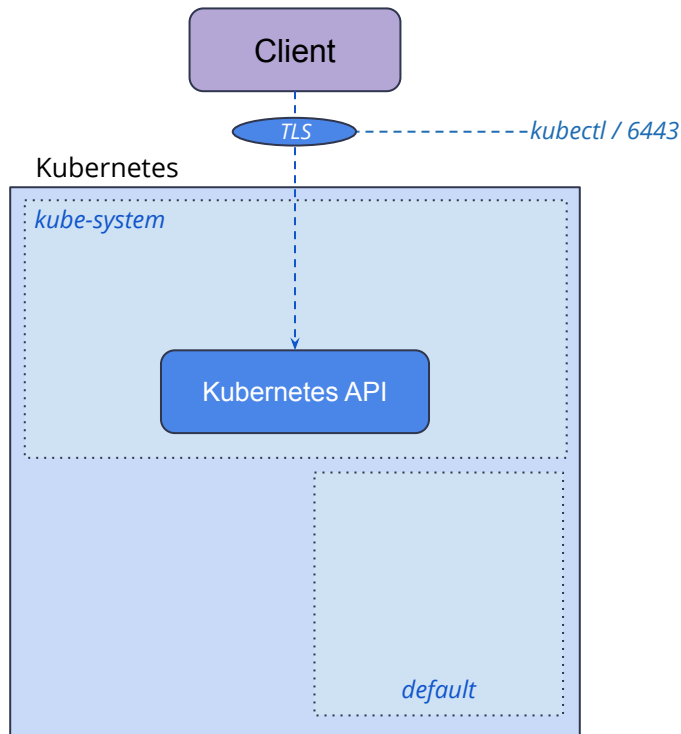


Let's talk about: **Tiller**

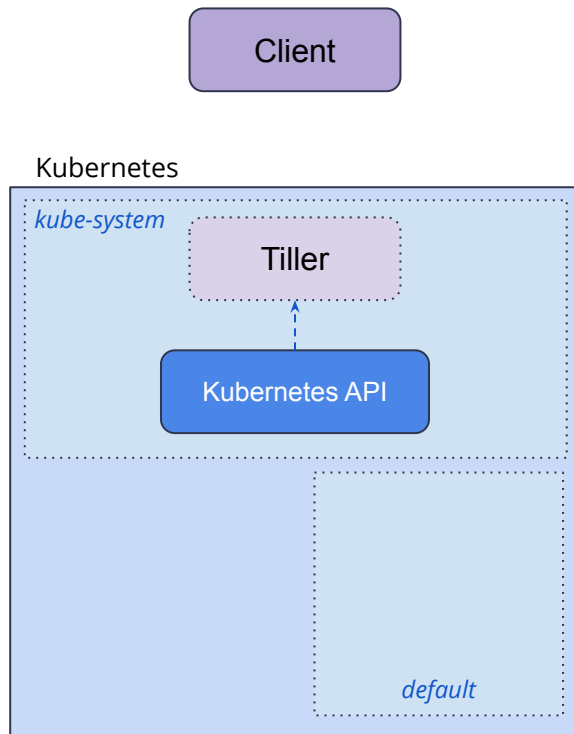
Helm v2 Architecture: Talkin' Tiller



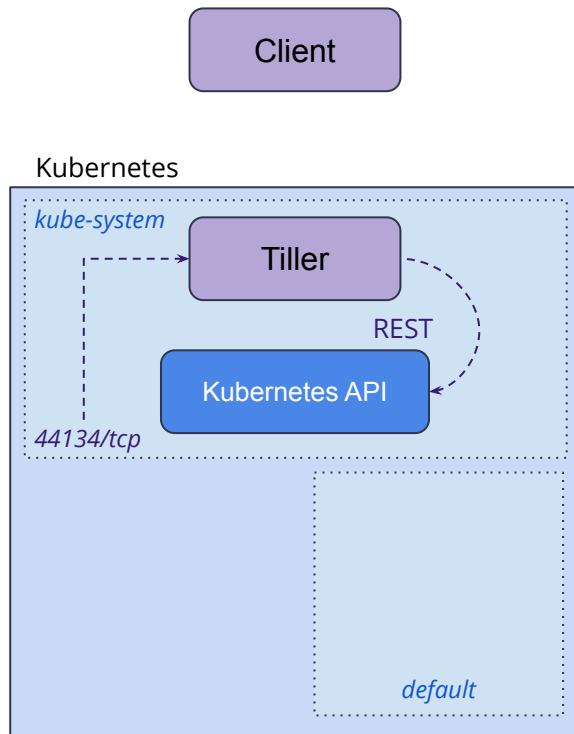
Helm v2 Architecture: Talkin' Tiller



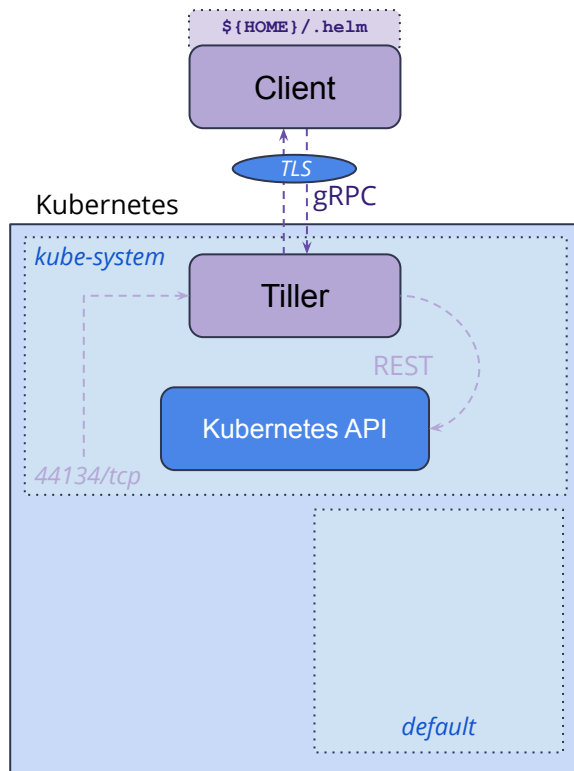
Helm v2 Architecture: Talkin' Tiller



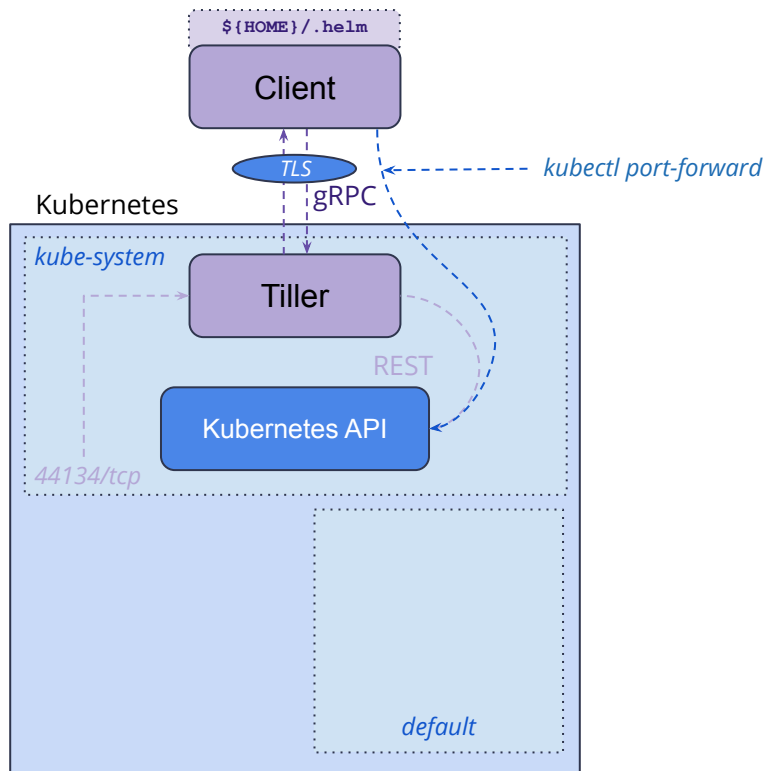
Helm v2 Architecture: Talkin' Tiller



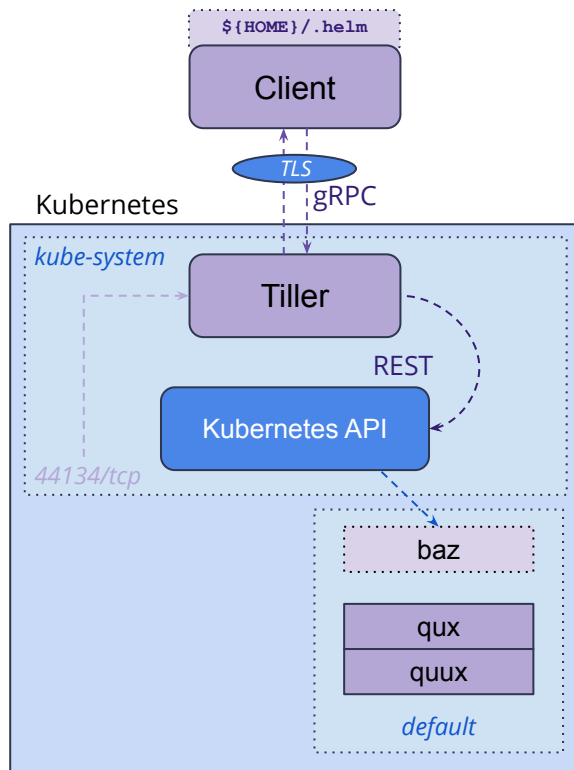
Helm v2 Architecture: Talkin' Tiller



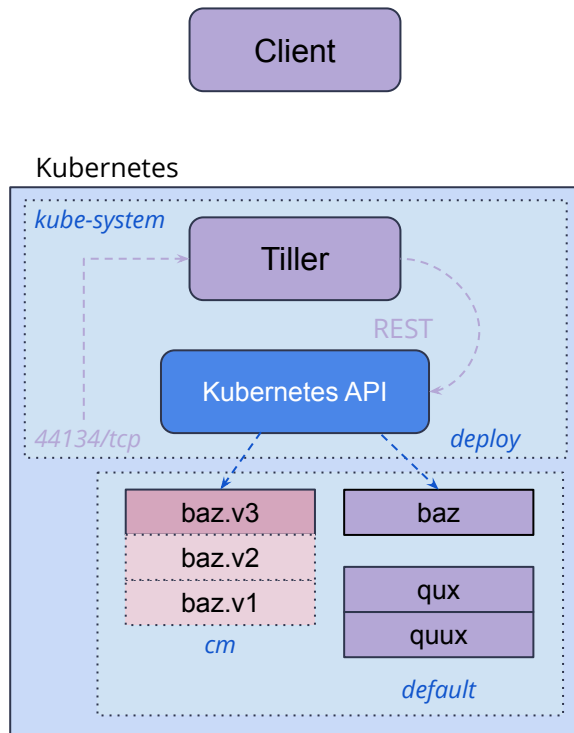
Helm v2 Architecture: Talkin' Tiller



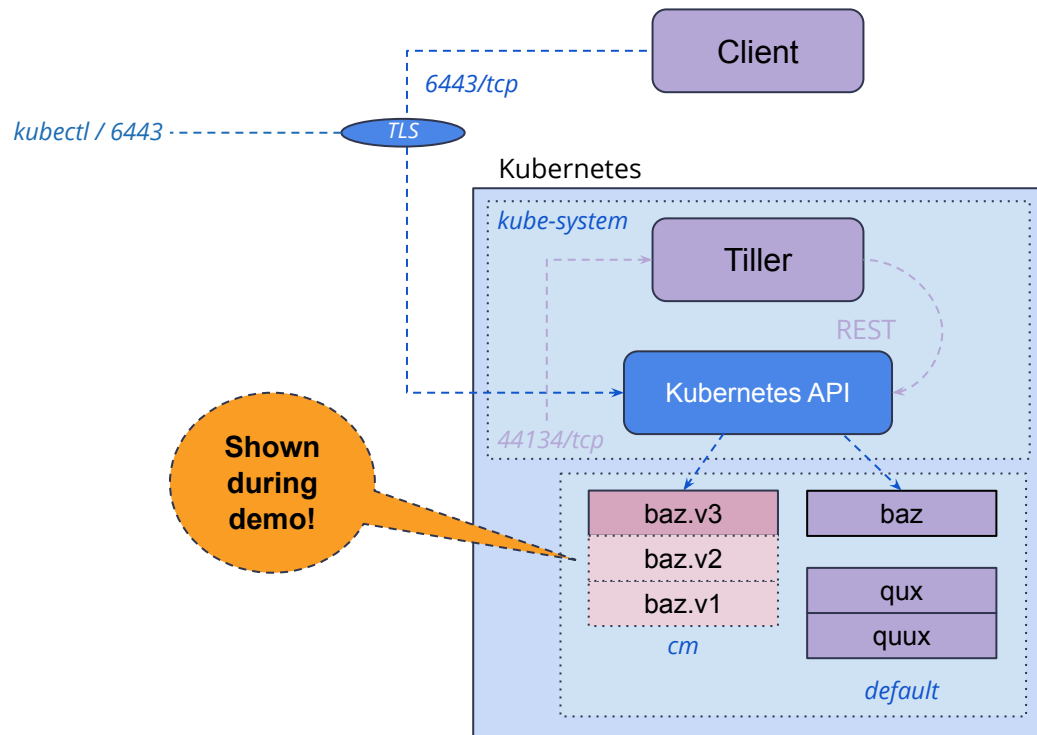
Helm v2 Architecture: Talkin' Tiller



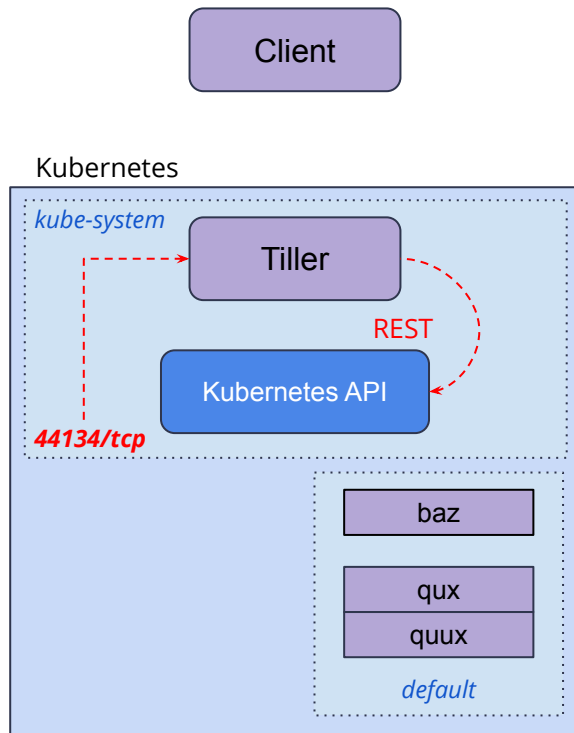
Helm v2 Architecture: Talkin' Tiller



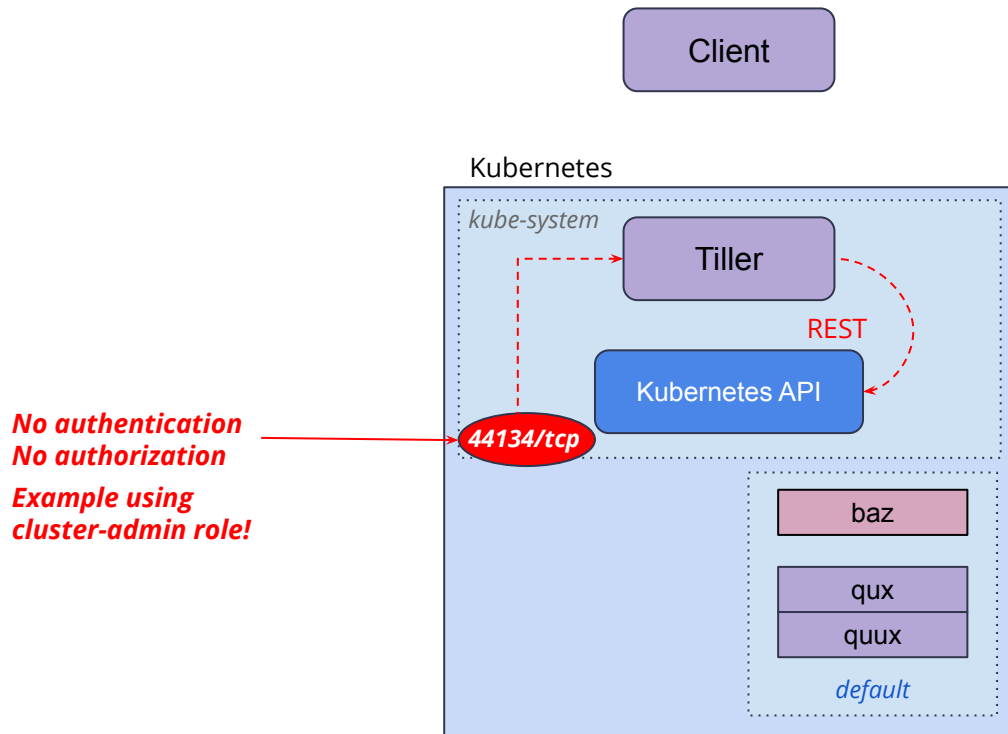
Helm v2 Architecture: Talkin' Tiller



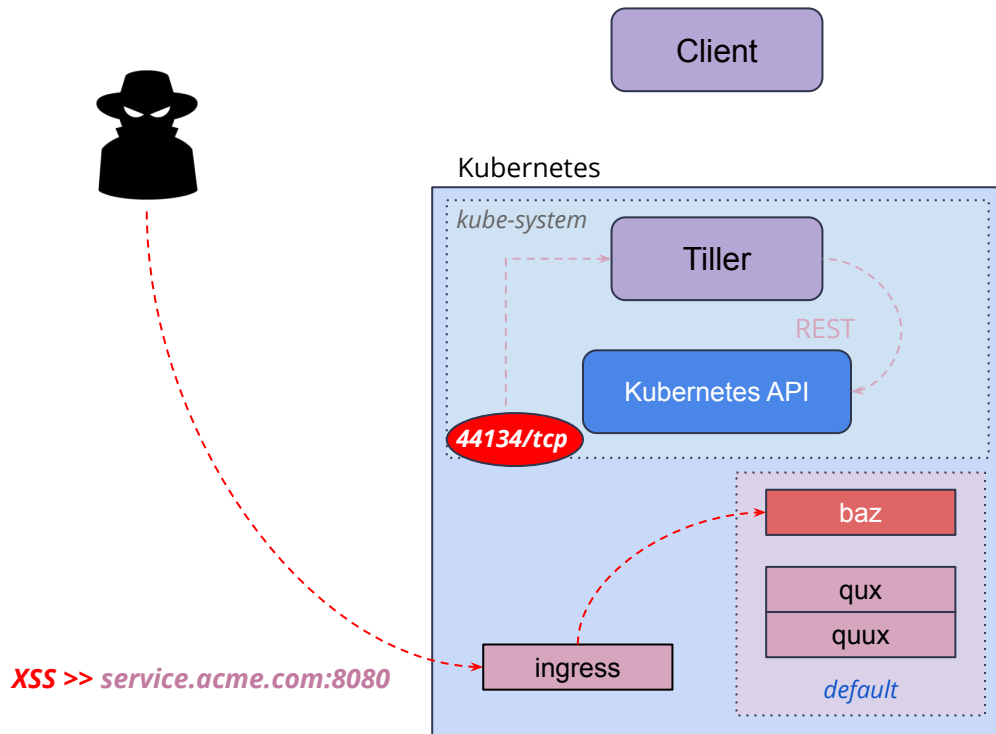
Helm v2 Architecture: Talkin' Tiller



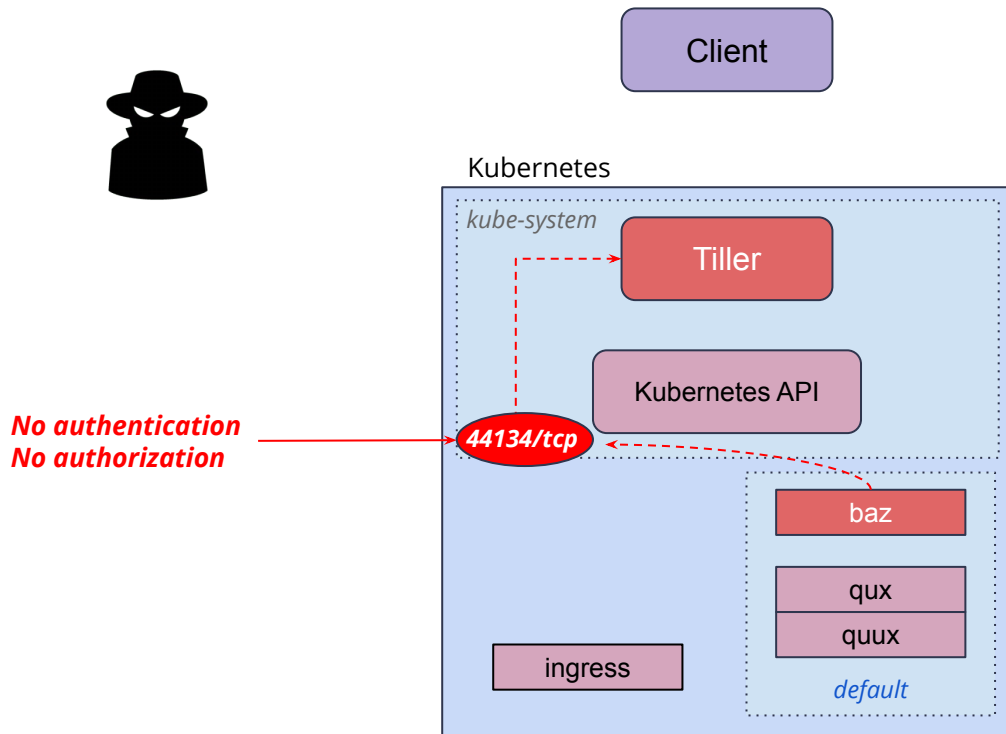
Helm v2 Architecture: Talkin' Tiller



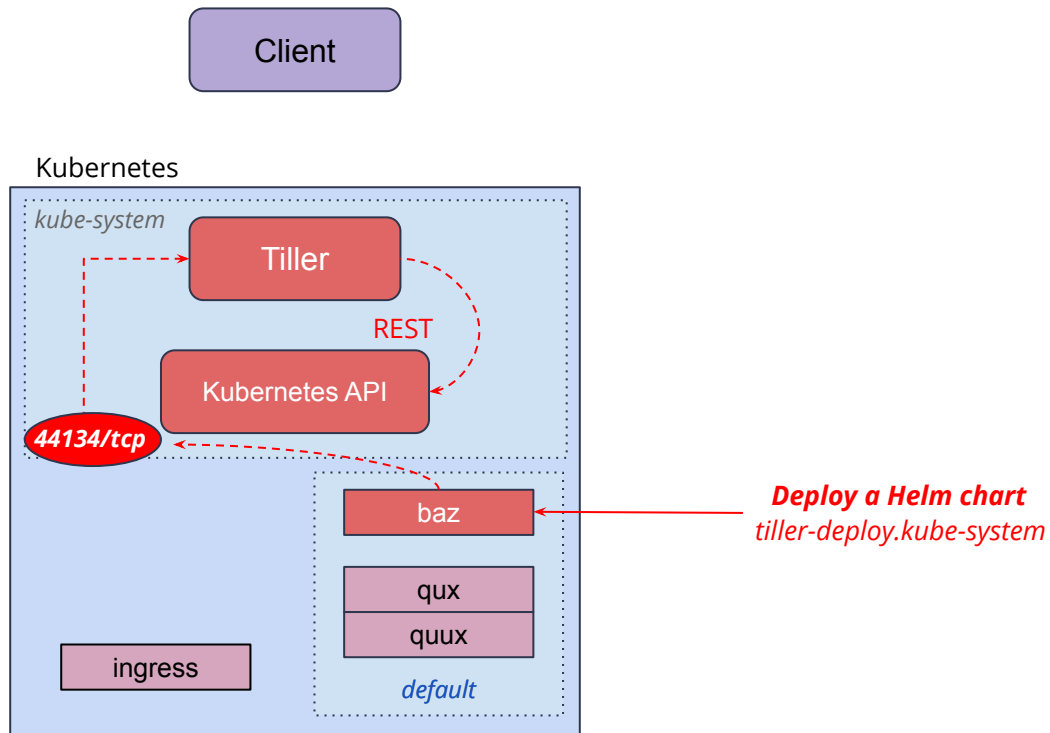
Helm v2 Architecture: Talkin' Tiller



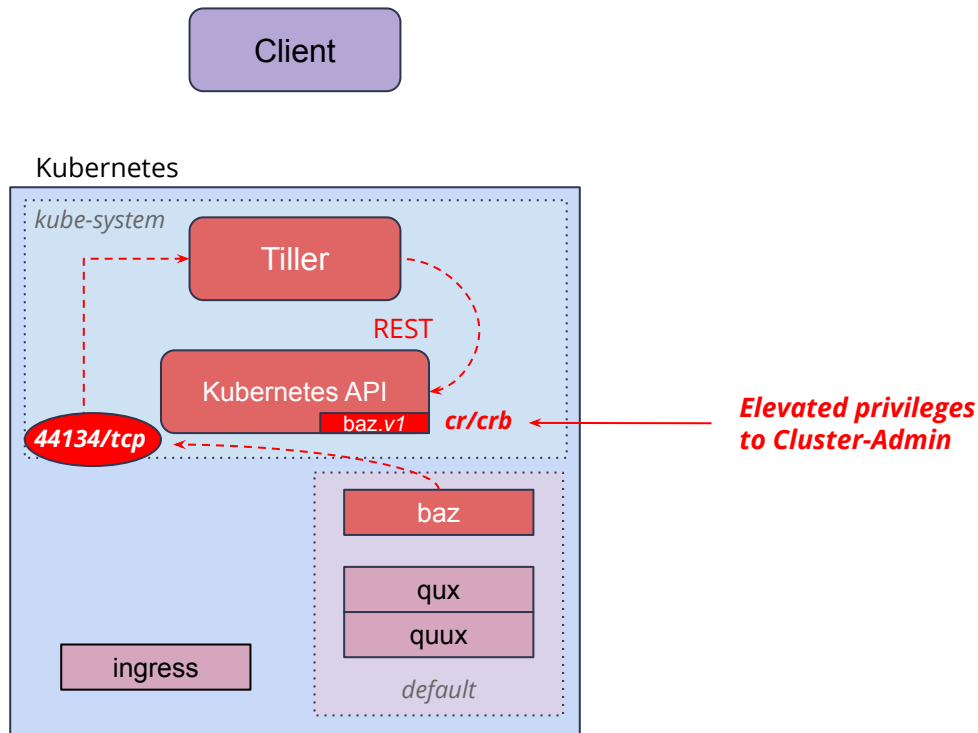
Helm v2 Architecture: Talkin' Tiller



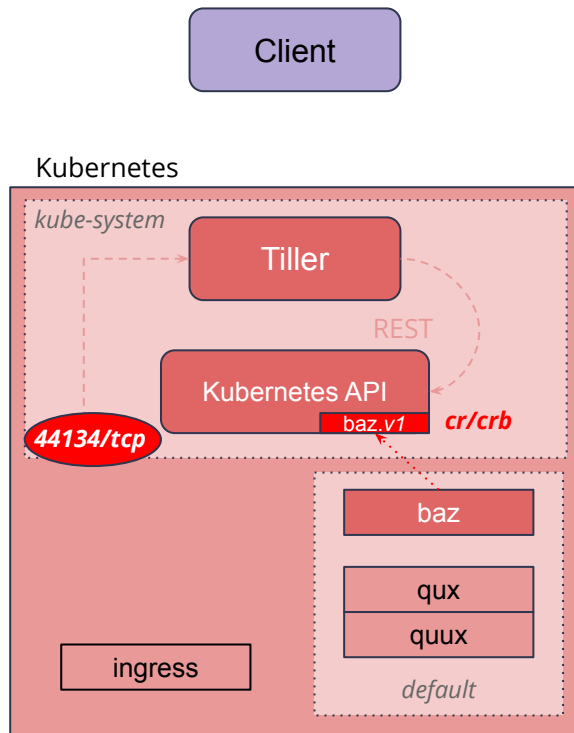
Helm v2 Architecture: Talkin' Tiller



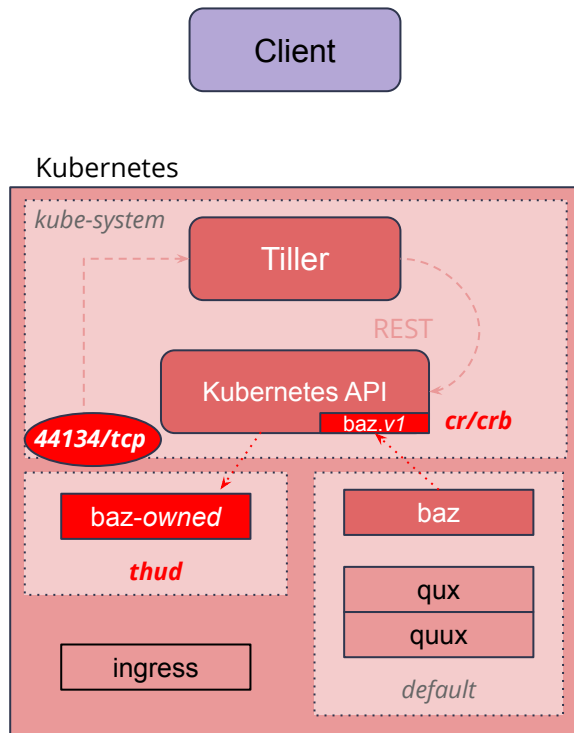
Helm v2 Architecture: Talkin' Tiller



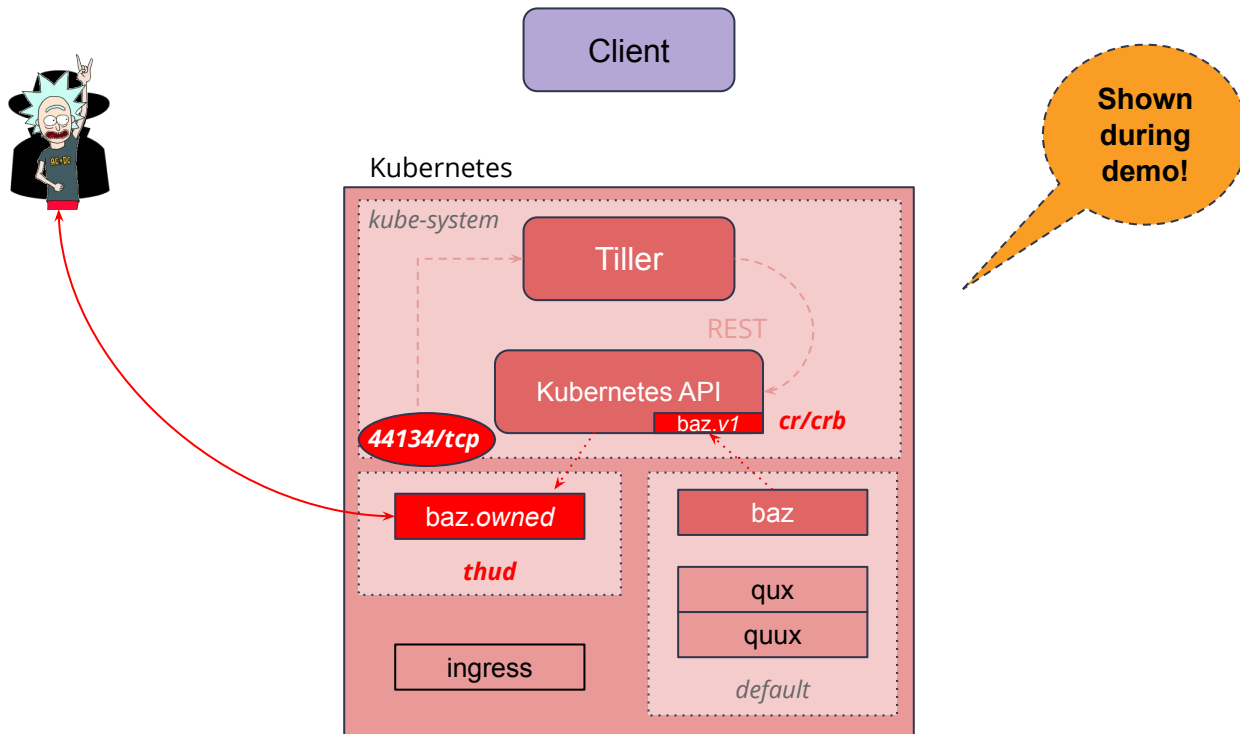
Helm v2 Architecture: Talkin' Tiller



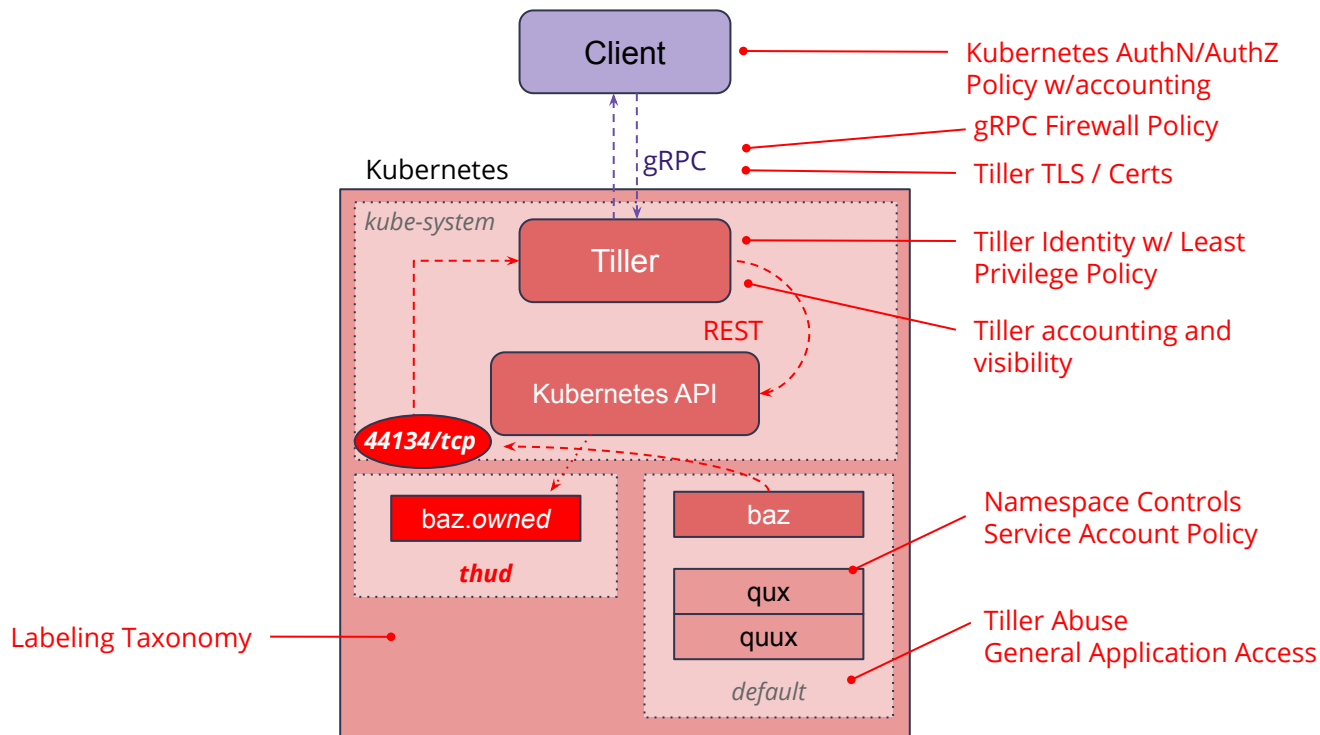
Helm v2 Architecture: Talkin' Tiller



Helm v2 Architecture: Talkin' Tiller



Helm v2 Architecture: Talkin' Tiller





Kubernetes already has a good
framework for least privileges

...so use it...

Changes in Helm v3: Security Enhancements

Security Enhancements: *Available Now*

- Tiller removed
- Leverages the Kubernetes security model
 - Same identity as in a user's `kubeconfig`
 - Use standard organization authN/authZ policy
- Communication using same methods as Kubernetes (TLS)
- Audits are now recorded at the cluster or federated level



Changes in Helm v3: Operational

New Repository/Registry Features: *Experimental*

- Push/Pull to/from a registry (OCI compliance coming soon)
- `helm login` to act similar to `docker login`
- OAuth2 authorization flow bearer/token auth
- Plugins can leverage customized spec/protocols (ex. `s3://`)
- Administrators can limit who has access to upload/install charts



Notes:

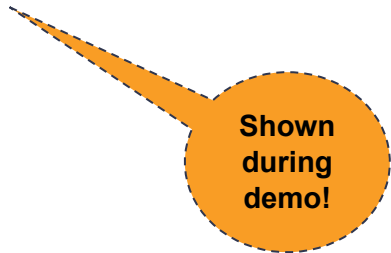
- *You can try `oras` for push/pull to/from a registry as well*
- *Enable experimental features with the following:*

```
export HELM_EXPERIMENTAL_OCI=1
```

Changes in Helm v3: Operational

Releases: *Available Now (Partially)*

- Release object in the form of a CRD and matching secret
- Secret represents a version of application instance (state)
- Namespaced artifacts
- Versions will have a ULID rather than old Helm v2 integers



Shown
during
demo!

Changes in Helm v3: Operational

New Helm Controller (CI/CD): *Planned*

- Not included as part of the Helm core project
- A CRD that will accept value overrides as part of a `HelmRequest`
- Subset of client functionality
- More granular RBAC restrictions

Changes in Helm v3: Development

Library Chart: *Available Now*

- No artifacts
 - Chart of declarations: functions, utilities, templates, etc.
- Support for Lua scripts in an the *new* `ext/` folder
- Defined in `Chart.yaml`, no longer in `requirements.yaml`



Demo!

Changes in Helm v3: Development

Introduction of Lua: *Planned*

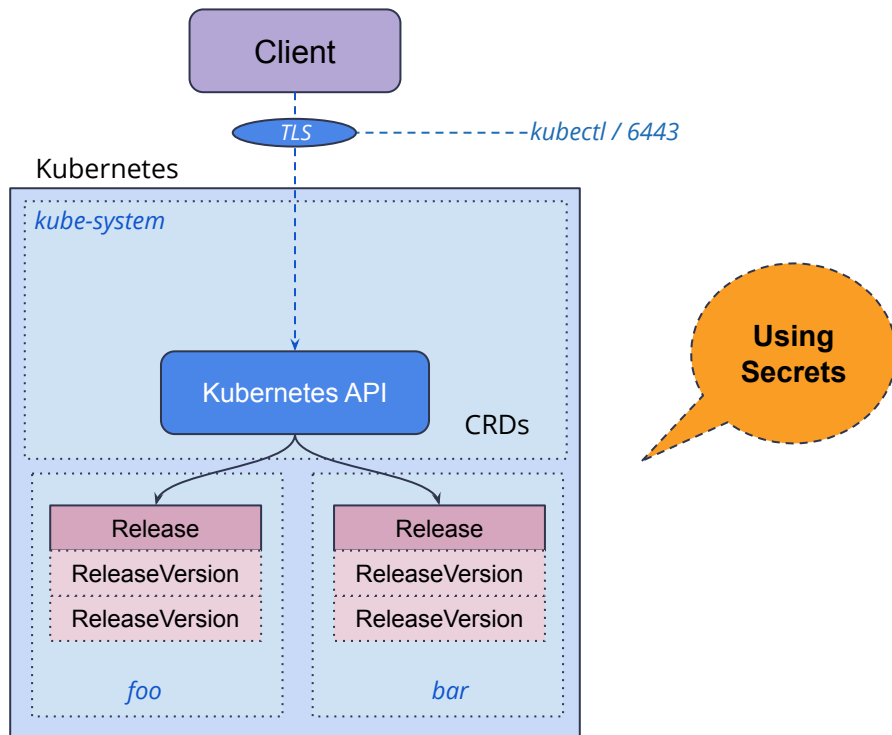
- New `ext/` folder
- Greater chart/sub-chart controls
- Lua is widely used in other popular projects
- Easy to learn
- Powerful and fast scripting language
- Considered to win out when compared with other languages:
 - Python, Ruby, Typescript, JavaScript

Changes in Helm v3: Other Operational Improvements

Other: *Planned*

- Managed Hooks
- `index.yaml` moving to `index.json` with version partitions
- `values.yaml` schemas via `values.schema.yaml`
- Bye bye `helm reset` and `helm serve`
- Consider running the Chart Museum Plugin:
<https://github.com/jdolitsky/helm-servecm>

Helm v3 Architecture: *Proposed Aug 8th 2018*



Demo/Walkthroughs



Thank You!
