An Introduction to Crowbar

Johannes Graßler
Cloud Developer
johannes.grassler@suse.com

Introduction

Crowbar Overview

- Open Source installation and configuration management framework
- Originally started by Dell, continued by SUSE
- Mainly used for SUSE OpenStack Cloud, Ceph, Cloud Foundry
- Configured through Ruby on Rails Web UI and REST API with command line clients

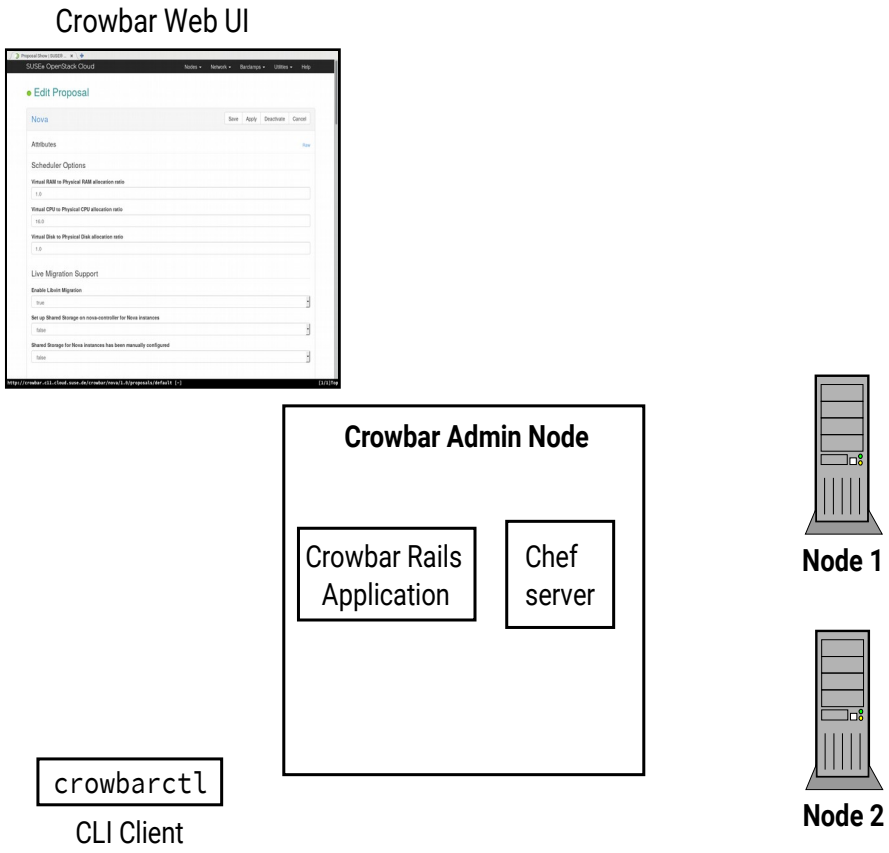
Features

- PXE boot discovery and installation for bare metal nodes
- Easily extensible through *bar clamps*, one per OpenStack component
- Pick and chose from the barclamps you actually need

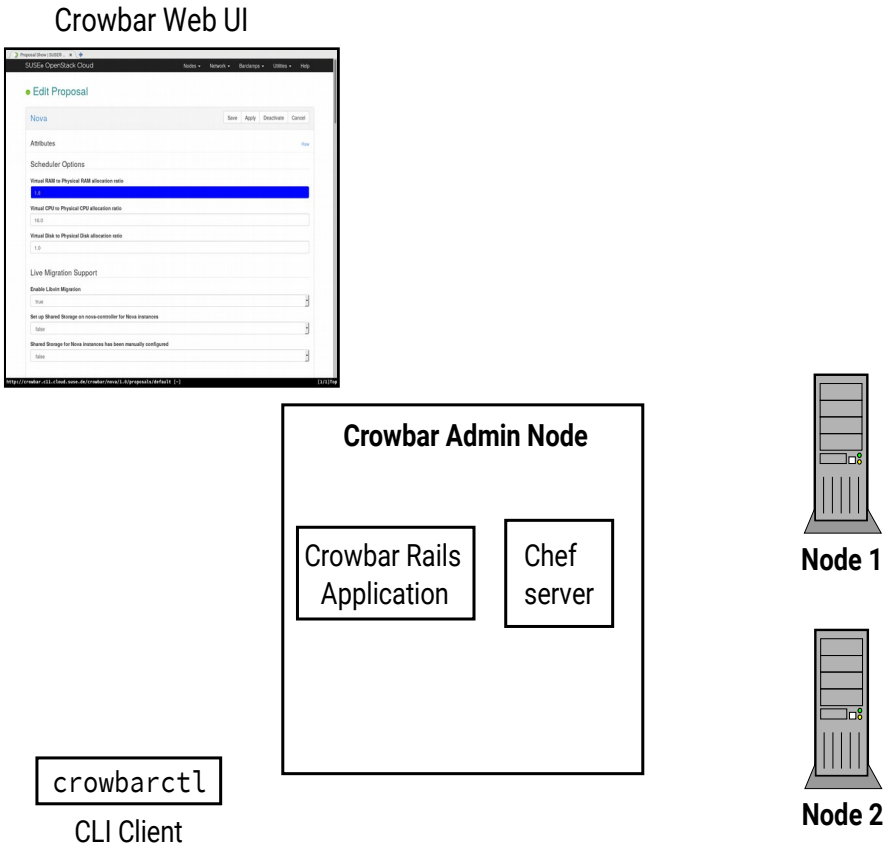
Barclamps: Crowbar's Configuration Modules

- Chef cookbooks for configuration management
- Default parameters for chef recipes from *data bags*: JSON data with schema
  - Crowbar runtime configuration (aka *Proposal*) is initialized from data bag
- Barclamp view in Crowbar Web UI for customizing parameters
- Validation code and UI elements in Rails application

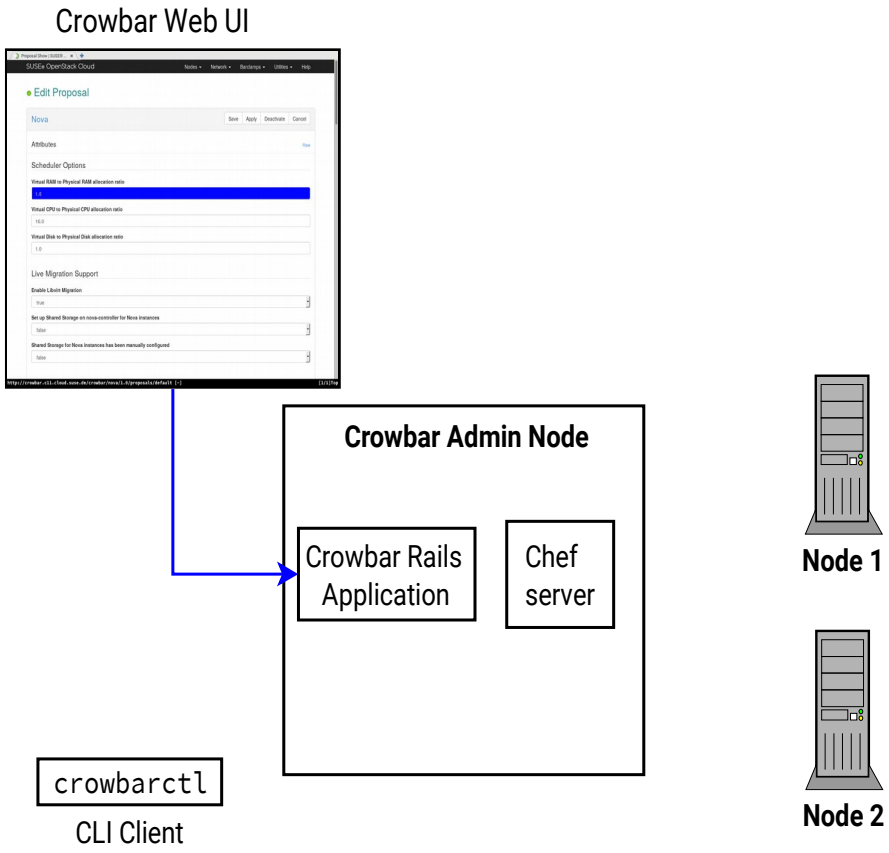
Web UI workflow (1)



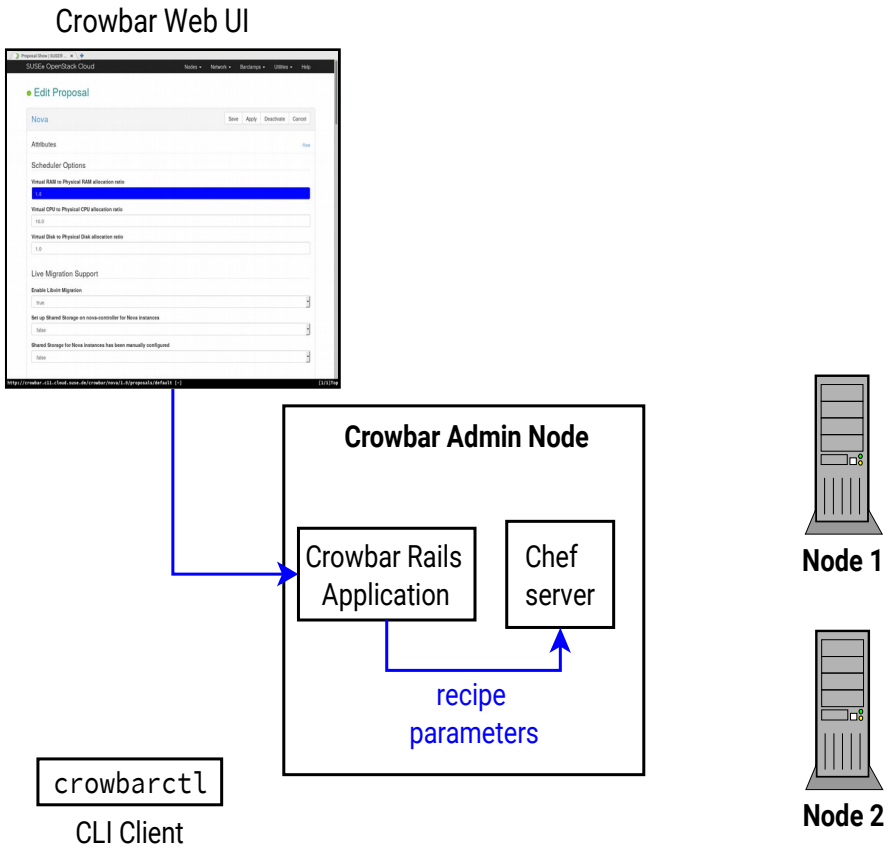
Web UI workflow (2)



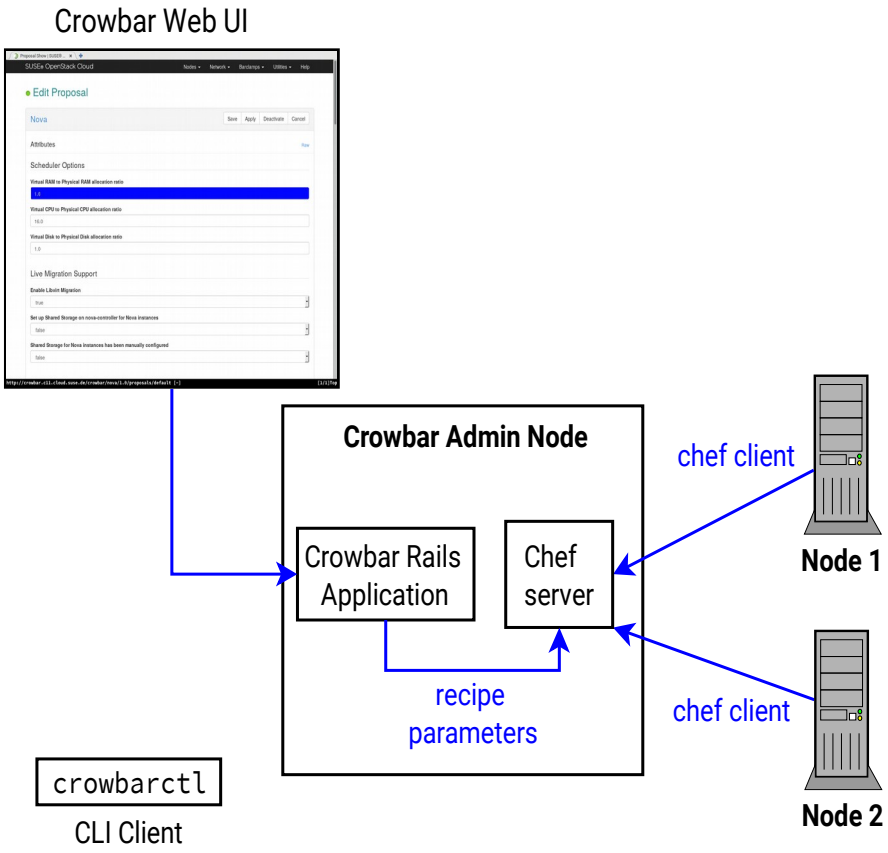
Web UI workflow (3)



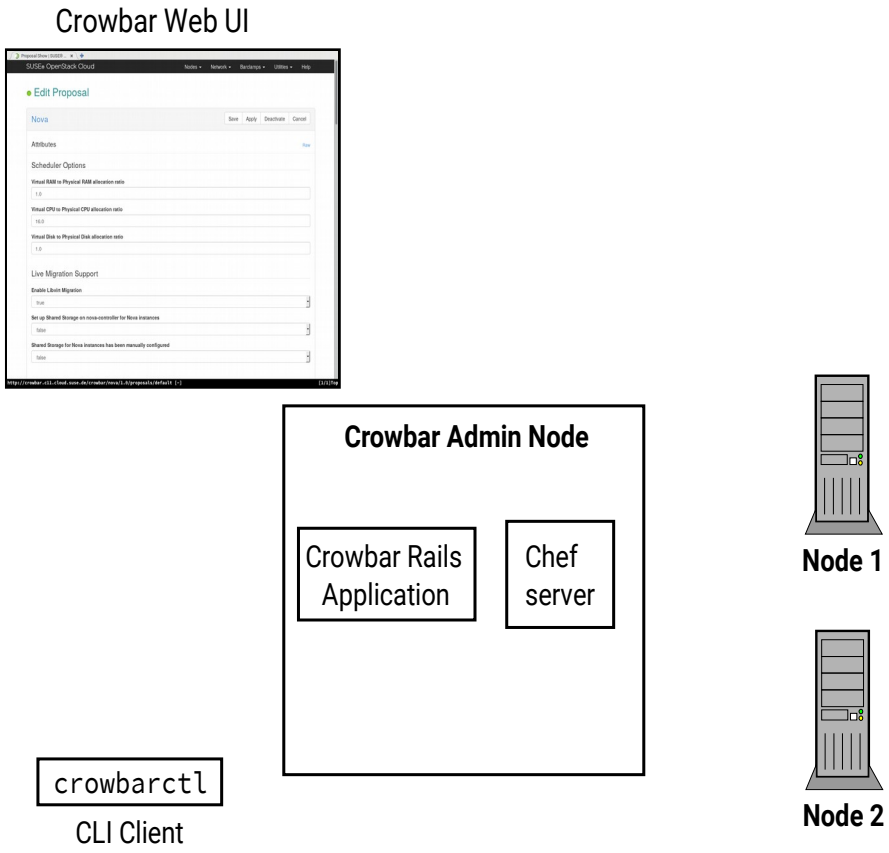
Web UI workflow (4)



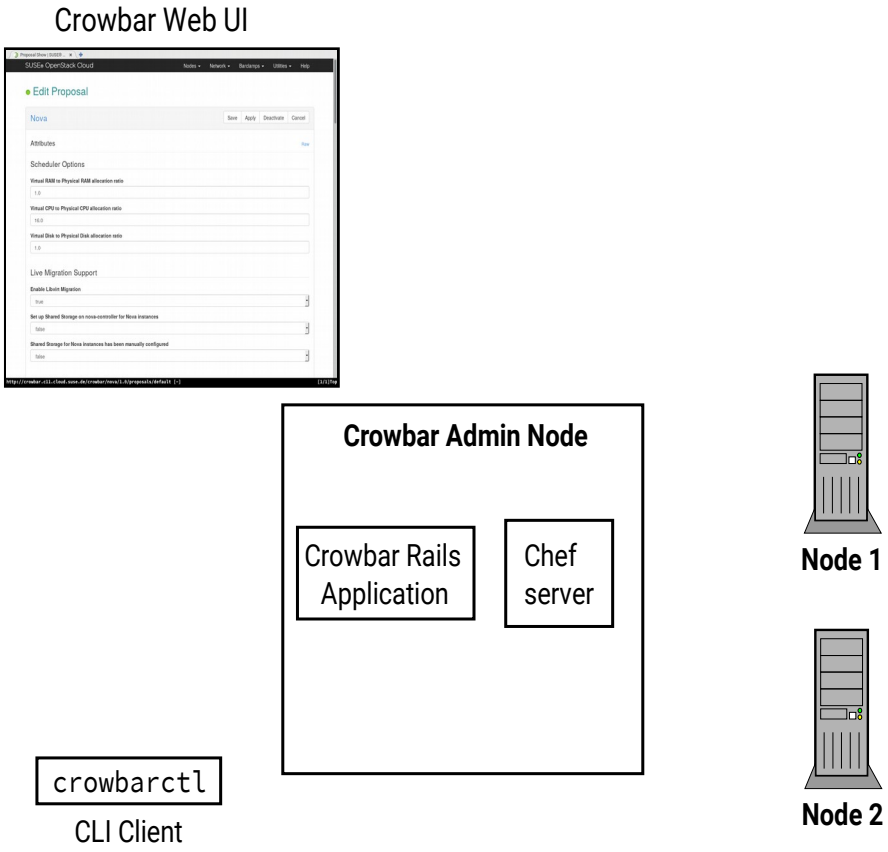
Web UI workflow (5)



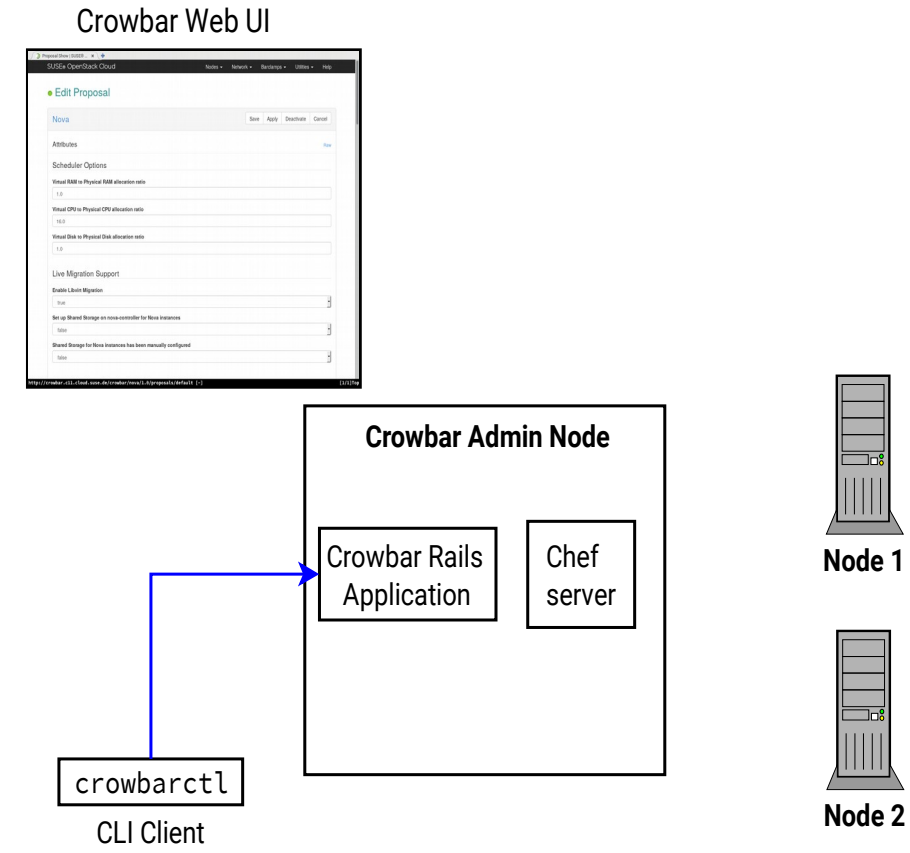
CLI/REST API workflow (1)



CLI/REST API workflow (2)

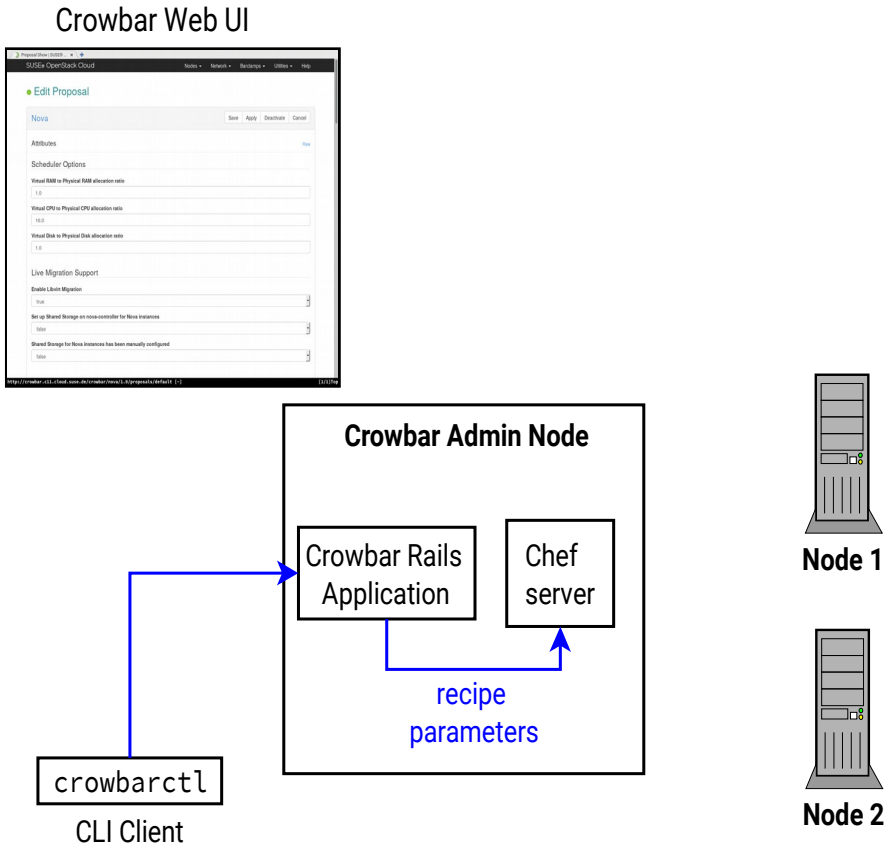


CLI/REST API workflow (3)



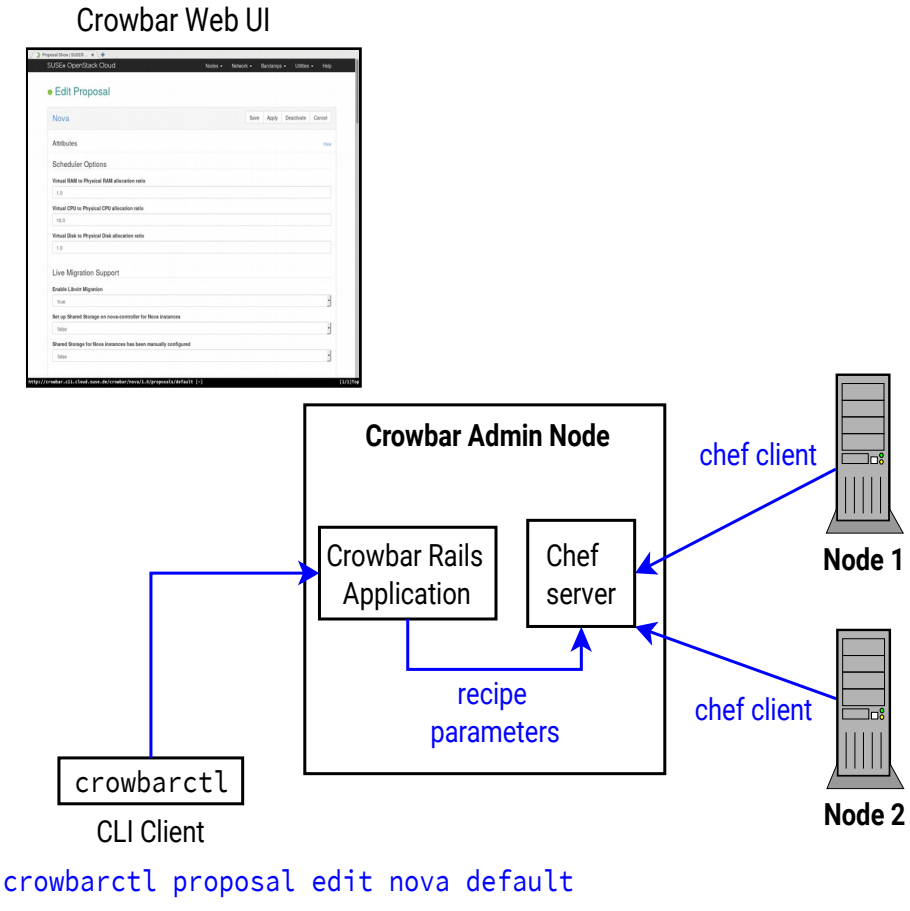

```
crowbarctl proposal edit nova default
```

CLI/REST API workflow (4)




```
crowbarctl proposal edit nova default
```

CLI/REST API workflow (5)



Contributing: Where to find Crowbar

- Main application: https://github.com/crowbar
- Core barclamps: https://github.com/crowbar-core
- OpenStack barclamps: https://github.com/crowbar/crowbar-openstack
- Mailing list: `crowbar@googlegroups.com`
- IRC: `#crowbar` on FreeNode

Development: Contribution workflow

- Fork the repository you intend to contribute to on Github and create a topic branch
- Once you are done with your modifications submit a pull request
- Acceptance Criteria:
  - Two positive reviews
  - All CI tests pass
- Patches get merged into `master` first and may need to be backported

Case Study: Creating a New OpenStack Barclamp

Example Barclamp: Barbican

- Example for case study: the Barbican barclamp (recently created)
- See commit cc1fea37169a4 for details
- Preparation: Fork https://github.com/crowbar/crowbar-openstack and create topic branch

Step 1: Data Bag to Define Parameters

Note: all paths on this slide are relative to `chef/data_bags/`

- Create `template-barbican.schema`: JSON formatted schema. Describes parameter types.
- Create `template-barbican.json`: JSON formatted data structure with default paramaters

Step 2: Creating a Chef Cookbok

Note: all paths on this slide are relative to `chef/cookbooks/barbican`

- Create Chef recipes in `recipes/`
  - Create 1 or more role recipes to aggregate recipes
- Declare dependencies in metadata.rb

Step 3: Roles

- Available role recipes need to be known to Crowbar App
- Add a role definition for each role recipe in `chef/roles/`.

Step 4: Register Barclamp in the Crowbar App

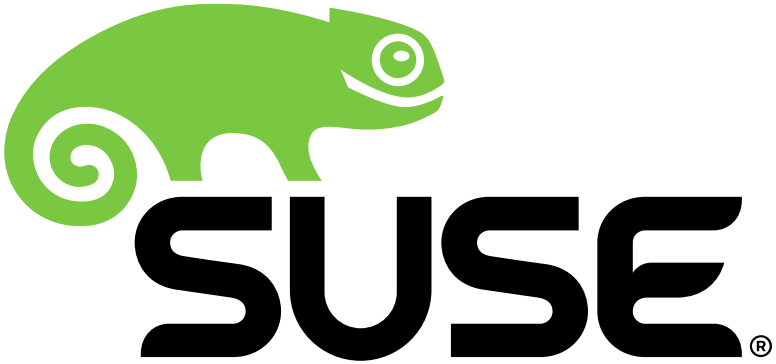
Note: all paths on this slide are relative to `crowbar_framework/` (unless stated otherwise)

Minimal changes:

- Create `barbican.yml` and `bin/crowbar_barbican` (relative to repository root)
- Create UI controller: `app/controllers/barbican_controller.rb`
- Create UI model: `app/models/barbican_service.rb`
- Create UI view: `app/views/barclamp/barbican/_edit_attributes.html.haml`
- Create UI labels: `config/locales/barbican/en.yml`

Slides and Transcript:
http://github.com/jgrassler/talks

Thank you.


SUSE®
We adapt. You succeed.

Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org