habitat

BY CHEF™

Mandi Walls

Technical Community Manager

@lnxchk

mandi@chef.io

https://habitat.sh

#habitatsh

http://slack.habitat.sh/

Chef and Automation

- Infrastructure Automation
- Cloud early adopters
- Digital Transformation
- Compliance Automation with InSpec
- Application Automation with Habitat

How Do We Run Applications?

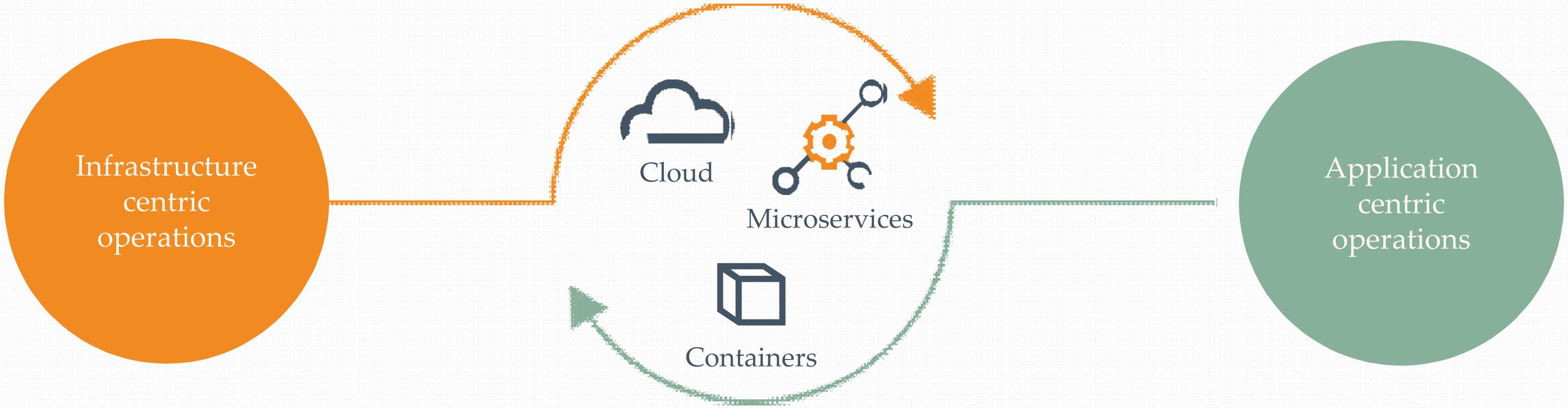
- On a computer
- With an OS
- And some libraries
- And some configuration
- And some way to start it and stop it

We’ve been moving complexity around rather than reducing it

Complexity Multiplies as Infrastructure Ages

```
def default_apache_version
  return '2.2' if node['platform_family'] == 'debian' && node['platform_version'] == '10.04'
  return '2.2' if node['platform_family'] == 'debian' && node['platform_version'] == '12.04'
  return '2.2' if node['platform_family'] == 'debian' && node['platform_version'] == '13.04'
  return '2.2' if node['platform_family'] == 'debian' && node['platform_version'] == '13.10'
  return '2.2' if node['platform_family'] == 'debian' && node['platform_version'].to_i == 6
  return '2.2' if node['platform_family'] == 'debian' && node['platform_version'].to_i == 7
  return '2.2' if node['platform_family'] == 'freebsd'
  return '2.2' if node['platform_family'] == 'omnios'
  return '2.2' if node['platform_family'] == 'rhel' && node['platform_version'].to_i == 5
  return '2.2' if node['platform_family'] == 'rhel' && node['platform_version'].to_i == 6
  return '2.2' if node['platform_family'] == 'suse'
  return '2.4' if node['platform_family'] == 'debian' && node['platform_version'] == '14.04'
  return '2.4' if node['platform_family'] == 'debian' && node['platform_version'] == '14.10'
...
end
```

Value and Work is Shifting


Infrastructure centric operations
Cloud
Microservices
Containers
Application centric operations

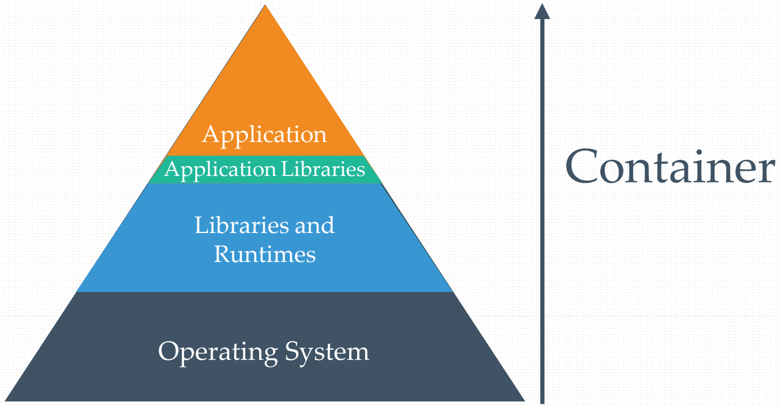
UNIT OF VALUE is server

UNIT OF WORK is configuration

UNIT OF VALUE is app

UNIT OF WORK is deployment

Artifact Inclusions



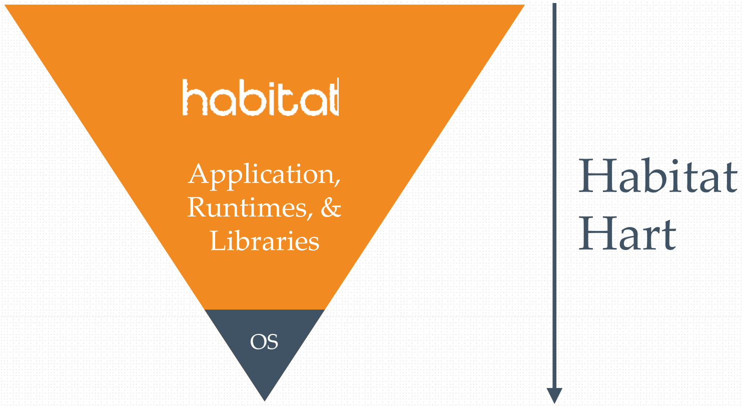
Application

Application Libraries

Libraries and Runtimes

Operating System

Container



habitat

Application, Runtimes, & Libraries

OS

Habitat Hart

Container formats recreate the traditional model of infrastructure and applications.

Poor at abstracting the Build + Run aspects of Applications

Ties dependencies to OS versions

Habitat builds from the application down

Small lightweight OS included

Embedded Supervisor for Application Management

Builds have strict dependency version control

So. Habitat.

- Ignore the underlying platform as much as possible
- Support microservices
- Manage container creep
- Make your workflow smoother

https://www.bonanza.com/listings/Premier-Food-Storage-Containers-20-Piece-Set-Grey/443972348

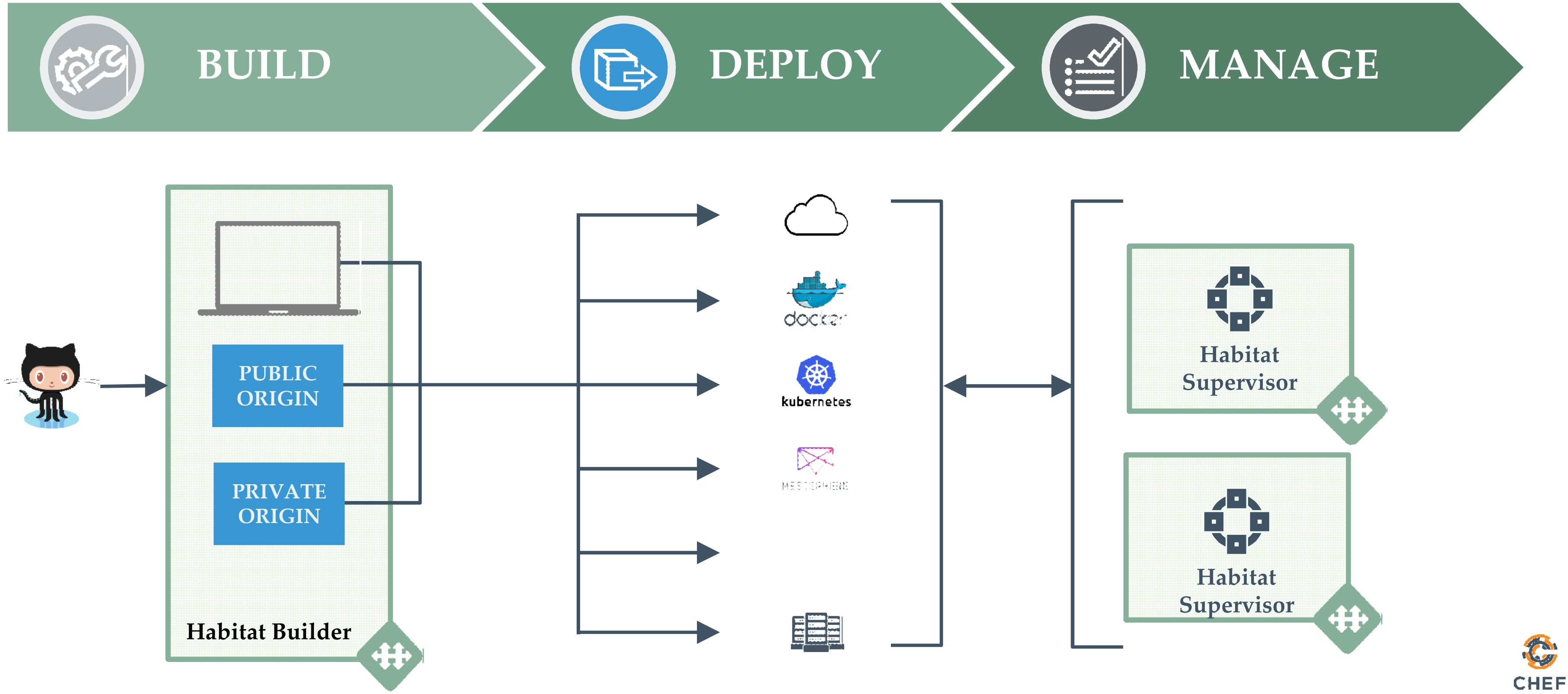
How Does Habitat Benefit Your Team

- Build once, run anywhere immutable artifacts
- Developers – Get the exact libraries needed, regardless of OS version
- Test / QA – Easier maintenance of test environments
- Operators – Run Linux habs on any distro
- Security – Query live services at runtime for potential vulnerabilities

Other Features

- Defer configuration decisions to runtime
- Build in a "clean room" : no accidental dependencies
- Service runtime and discovery
- Configuration exposed via API
- Signed packages for your organization

Cloud Native Applications with Habitat



BUILD

DEPLOY

MANAGE

PUBLIC ORIGIN

PRIVATE ORIGIN

Habitat Builder

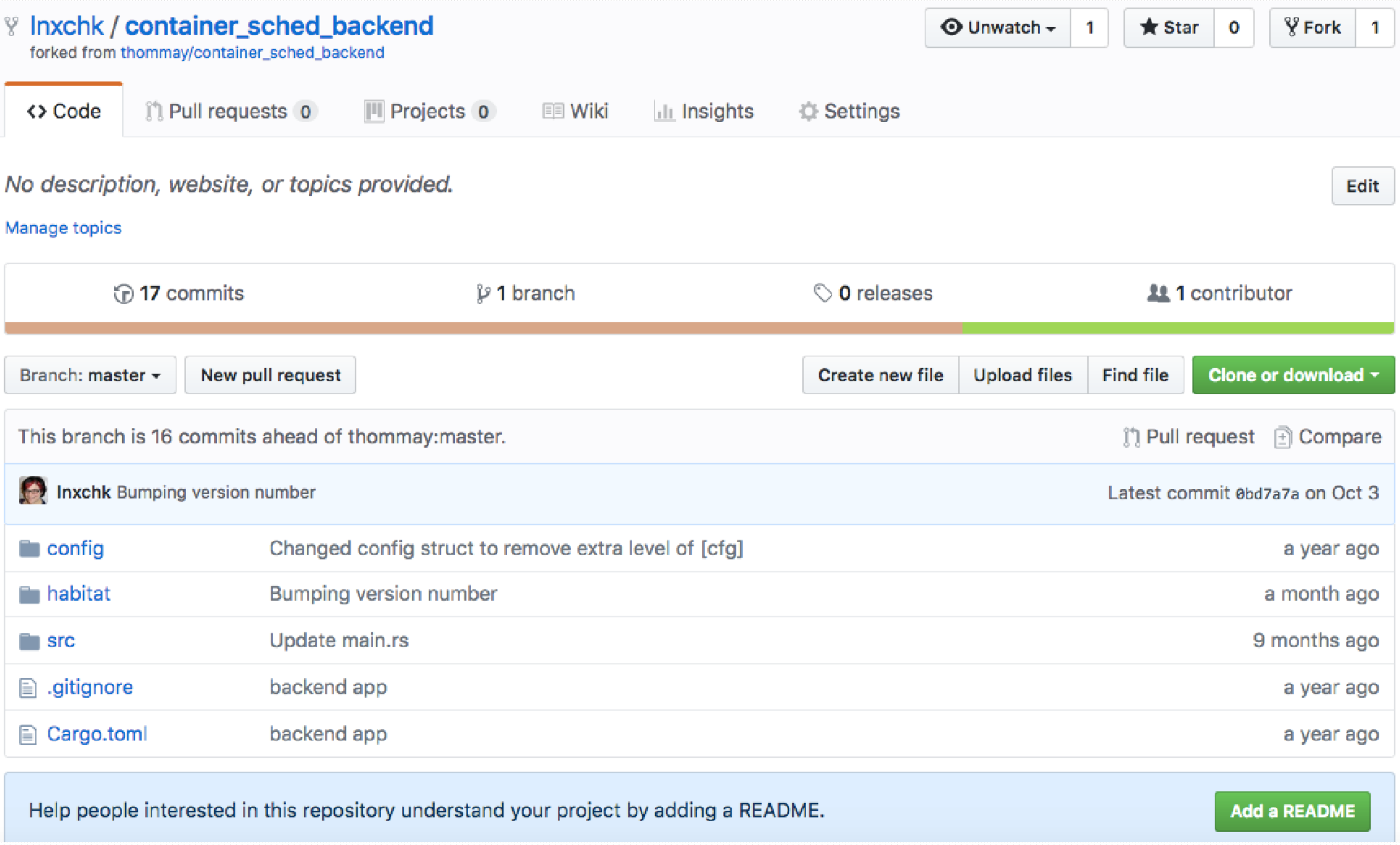
Habitat Supervisor

Habitat Supervisor

Sample Application

- container_sched_backend – small rust application
- Runs on Linux in a cluster
- The code is plugged into the Habitat Builder service for automatic builds
- Application has a configuration setting for "favorite color"

Github Repository



lnxchk / container_sched_backend

forked from thommay/container_sched_backend

No description, website, or topics provided.

Manage topics

17 commits · 1 branch · 0 releases · 1 contributor

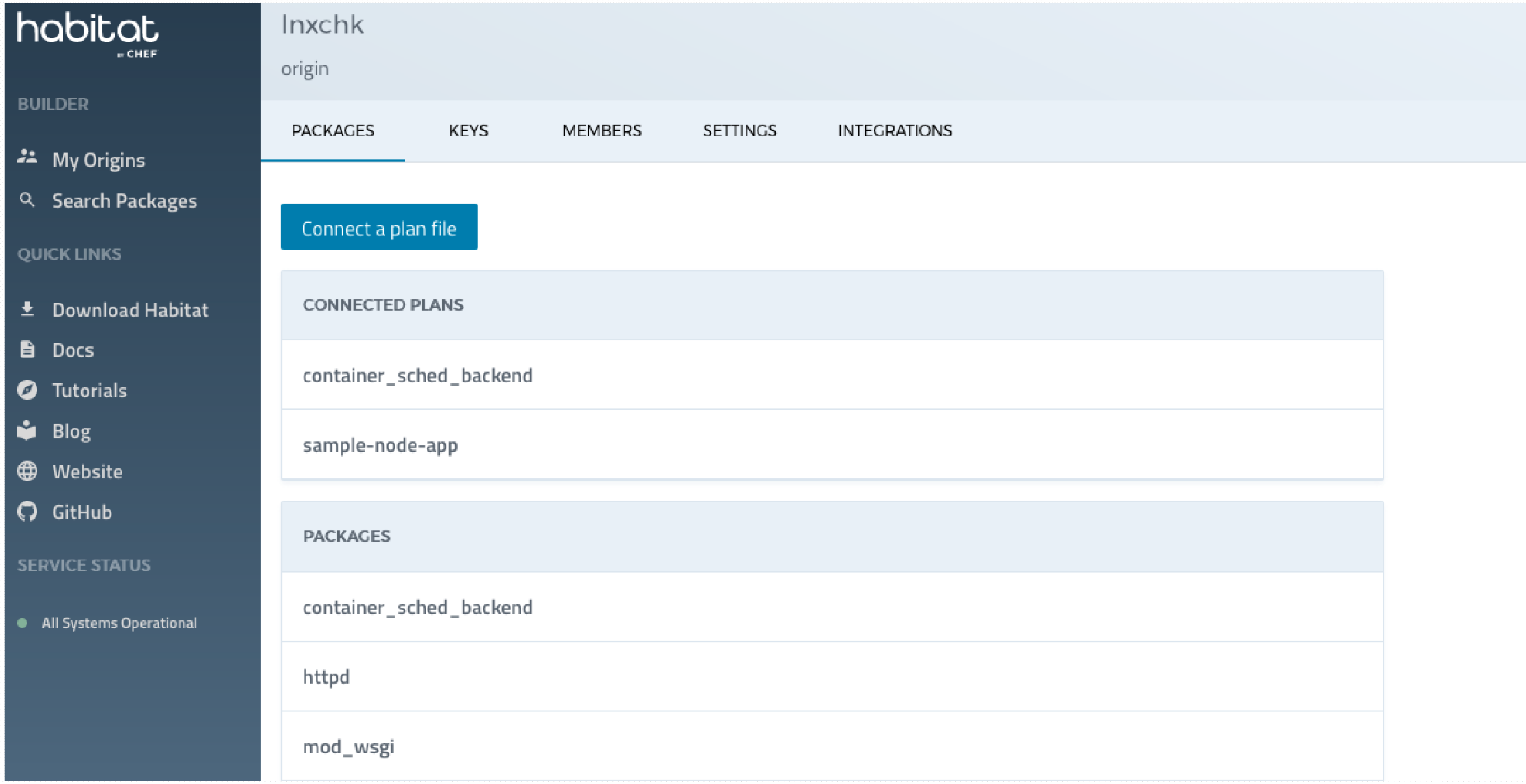
This branch is 16 commits ahead of thommay:master.

lnxchk Bumping version number — Latest commit 0bd7a7a on Oct 3

config	Changed config struct to remove extra level of [cfg]	a year ago
habitat	Bumping version number	a month ago
src	Update main.rs	9 months ago
.gitignore	backend app	a year ago
Cargo.toml	backend app	a year ago

Help people interested in this repository understand your project by adding a README.

Builder Package Registry



lnxchk

origin

CONNECTED PLANS

container_sched_backend

sample-node-app

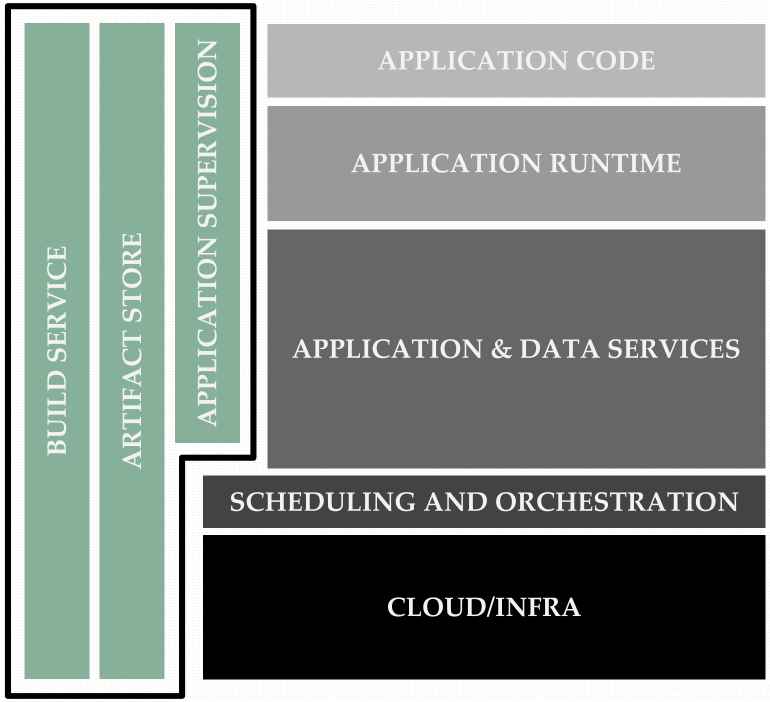
PACKAGES

container_sched_backend

httpd

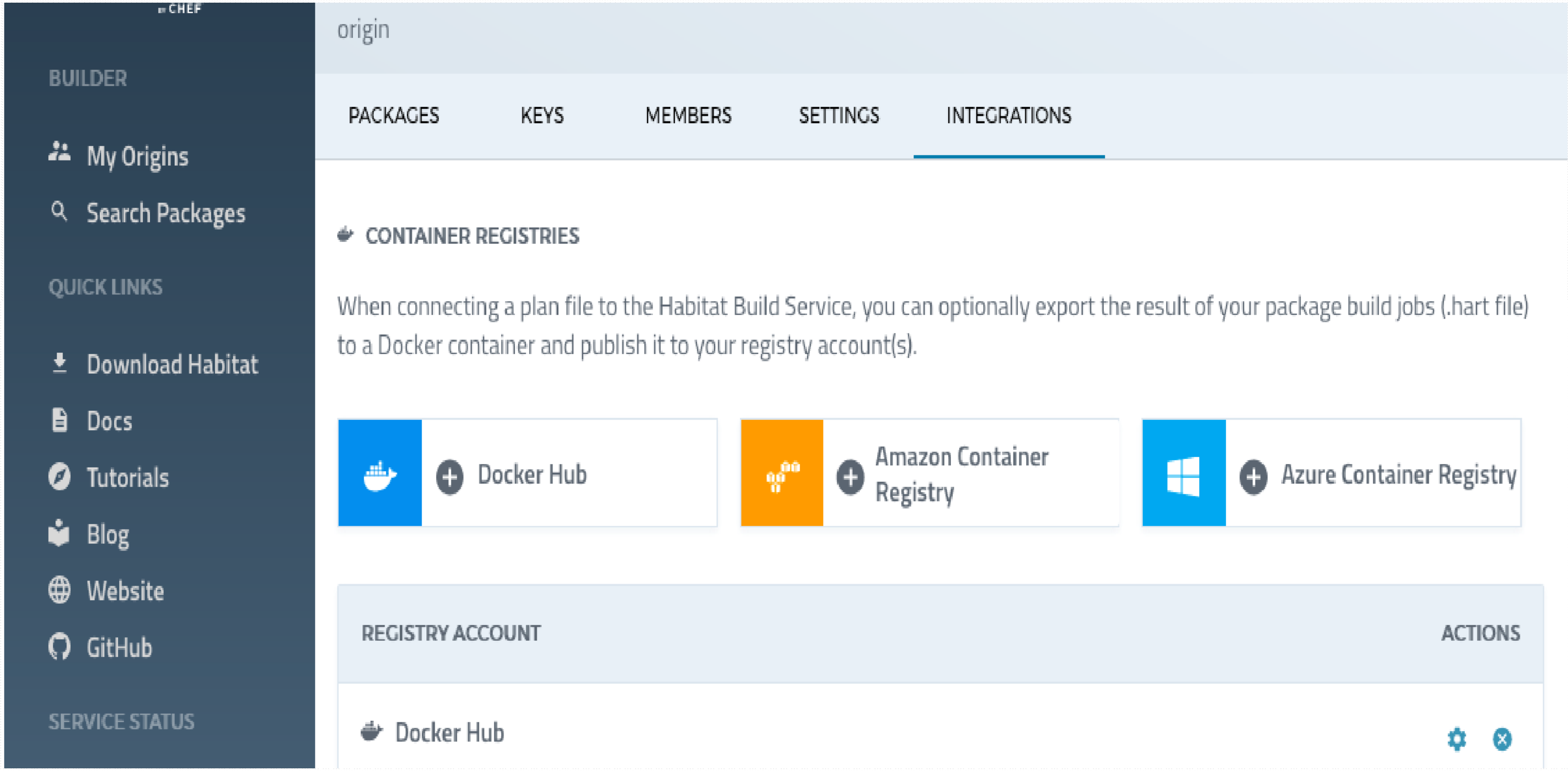
mod_wsgi

Leave it to Builder



- Build service
- And an artifact store
- Private repos
- Check code in to GitHub, let Builder create your artifact
- On-Prem version is depot only right now

Container Integrations

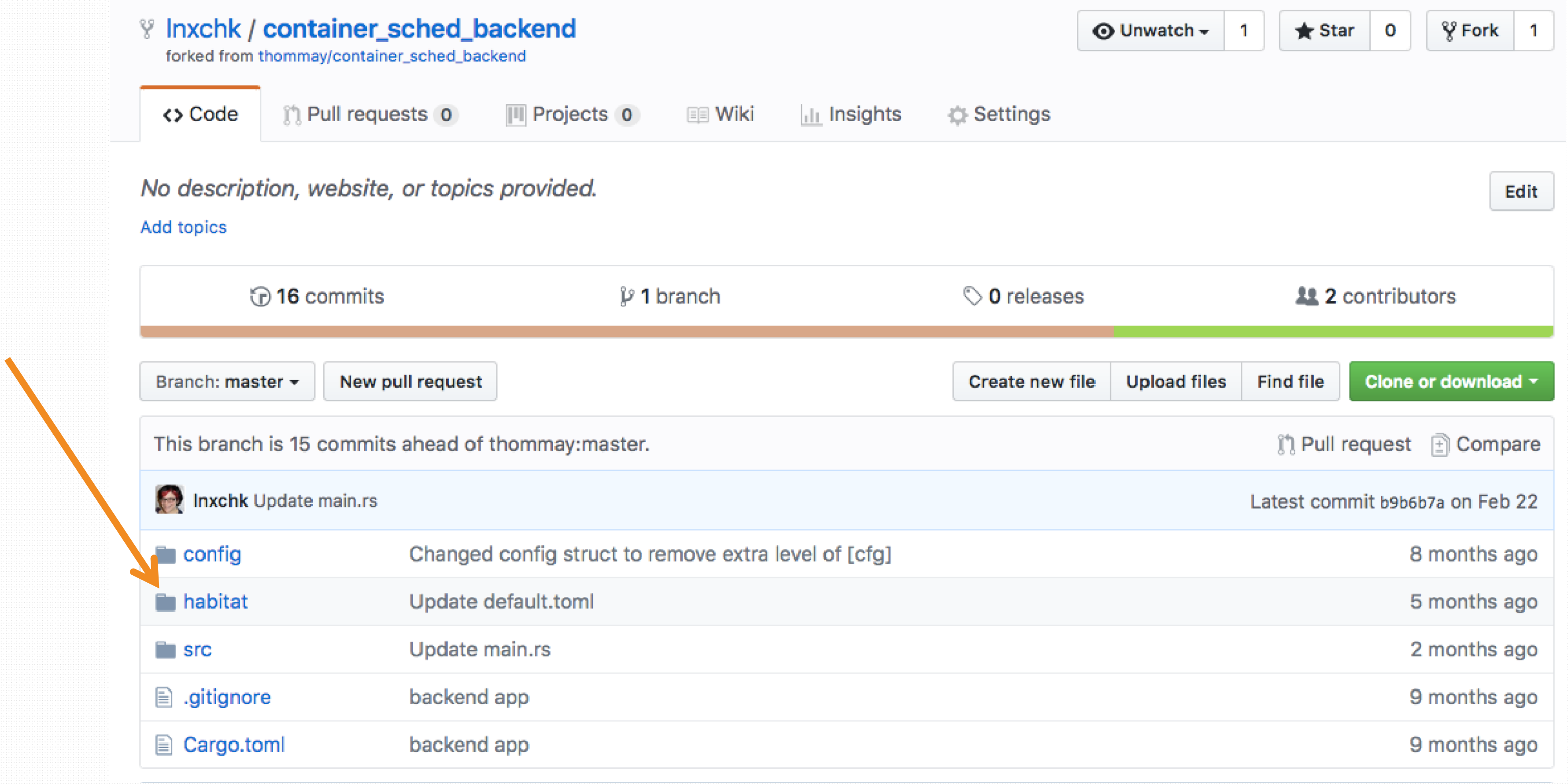


CONTAINER REGISTRIES

When connecting a plan file to the Habitat Build Service, you can optionally export the result of your package build jobs (.hart file) to a Docker container and publish it to your registry account(s).

REGISTRY ACCOUNT	ACTIONS
Docker Hub	

Habitat Files Travel with the App Code



lnxchk / container_sched_backend

forked from thommay/container_sched_backend

No description, website, or topics provided.

Add topics

16 commits | 1 branch | 0 releases | 2 contributors

This branch is 15 commits ahead of thommay:master.

lnxchk Update main.rs — Latest commit b9b6b7a on Feb 22

config	Changed config struct to remove extra level of [cfg]	8 months ago
habitat	Update default.toml	5 months ago
src	Update main.rs	2 months ago
.gitignore	backend app	9 months ago
Cargo.toml	backend app	9 months ago

Have a Plan

```
pkg_name=container_sched_backend
pkg_origin=lnxchk pkg_version="0.1.0" pkg_build_deps=(core/rust)
pkg_deps=(core/glibc core/gcc core/gcc-libs)
pkg_bin_dirs=(bin)
bin="container_sched_backend"
pkg_exports=([out]=cfg.out)
do_build() {
  cargo build
}
do_install() {
  install -v -D "$PLAN_CONTEXT/../target/debug/$bin" \ "$pkg_prefix/bin/$bin”
}
pkg_svc_run="$bin"
```

Metadata

Build

Install

Run

Build Output

- By default, it’s a hart – a compressed tarball with some metadata and a signature
- You can export to other formats, like Docker containers
- The hart itself it runnable

Run a Hart

- Set up a first host: leader/follower topology and rolling updates

```
sudo hab sup run lnxchk/container_sched_backend  --topology leader --strategy rolling
```

- Connect additional nodes to the ring, giving them a peer to attach to

```
sudo hab sup run lnxchk/container_sched_backend --strategy rolling --peer 172.31.37.193
```

Updating Configuration at Runtime

- Update all or part of the configuration while the apps are running
- Send the update to a member of the mesh and they will all update

```
sudo hab config apply \ \
 container_sched_backend.default 2 new.toml
```

Channels and Service Groups

- Builder can have multiple channels where harts are published
  - Default is "unstable", can promote via CLI or web
  - A cluster can subscribe to any channel, will use "stable" by default
- Service groups in the cluster have their name appended to the artifact name
  - container_sched_backend.default

Querying the API

- Rest api on port 9631
- /services endpoint for information about the running services
- Query it with some simple tools

  https://github.com/lnxchk/hab_tools

Show the dependencies

```
$ ./hab_deps.rb
lnxchk/container_sched_backend/0.1.3/20181003183354
|-  core/binutils/2.30/20180608050633
|-  core/gcc-libs/7.3.0/20180608091701
|-  core/gcc/7.3.0/20180608051919
|-  core/glibc/2.27/20180608041157
|-  core/gmp/6.1.2/20180608051426
|-  core/libmpc/1.1.0/20180608051824
|-  core/linux-headers/4.15.9/20180608041107
|-  core/mpfr/4.0.1/20180608051629
|-  core/zlib/1.2.11/20180608050617
```

Search for a package with a vulnerability

```
$ ./hab_deps_search.rb -d core/mpfr/4.0.1/20180608051629
searching host localhost for dep core/mpfr/4.0.1/20180608051629
lnxchk/container_sched_backend/0.1.3/20181003183354 matches
```

Check the whole group is on the same version

```
$ ./hab_service_versions.rb -h localhost -g container_sched_backend.default
searching host localhost for dep container_sched_backend.default
172.31.39.99 has version lnxchk/container_sched_backend/0.1.3/20181003183354
172.31.37.152 has version lnxchk/container_sched_backend/0.1.3/20181003183354
172.31.46.78 has version lnxchk/container_sched_backend/0.1.3/20181003183354
172.31.35.43 has version lnxchk/container_sched_backend/0.1.3/20181003183354
```

Automatic Updates

- Started the supervisors with "--strategy rolling" to allow for rolling updates
  - Also none – no auto updates; and at-once for all at once updates
- This app is hooked through builder, so update from github can push updates to the running instances
- Packages published to channels
  - Promote to stable for subscriber updates or subscribe to other channels at runtime

Shortcut for common platforms: Scaffolding

- Default core-built dependencies for common runtimes
- Ruby and Node so far

```
pkg_name=MY_APP
pkg_origin=MY_ORIGIN
pkg_version=MY_VERSION
pkg_scaffolding=core/scaffolding-ruby
```

https://www.habitat.sh/docs/concepts-scaffolding/

Summary

- Single build pathway from Github through to deployment
- Run Linux-based instances on any distribution
- Update configuration at runtime
- Verify the live environment via REST API
- Targeted platforms: Java, node.js, Ruby, Windows .NET

Join Us!

- On Slack
  http://slack.habitat.sh
- Online! With Tutorials
  https://www.habitat.sh/
- On Github
  https://github.com/habitat-sh