OpenCompose Spec Proposal

Story

As a *developer*, I want to easily *describe* and *run* a number of services that are *part of* my application. I want to be able to store the environment config *alongside my code*.

I **do** know: the stack it is based on, what ports my service exposes, what volumes it needs to mount, what environment variables to configure.

I **don't** know: the details of the topology and resources available in stage/prod

I **may** want to indicate: the resources required to run the services, which services should be colocated

Goals

- A format usable by both humans and software
- Can be checked in to SCM
- Executable without additional metadata
    - Allows overlaying additional metadata (e.g. implementation specific file supplied at deployment)
- Can be extended by implementors to allow innovation
- Allows external linking (doesn’t necessarily describe a complete application)
- Tooling friendly
- Native solution - not a file-conversion story

Value Proposition for Developers

Describe what you know about your services in a simple, executable format that you can run locally.

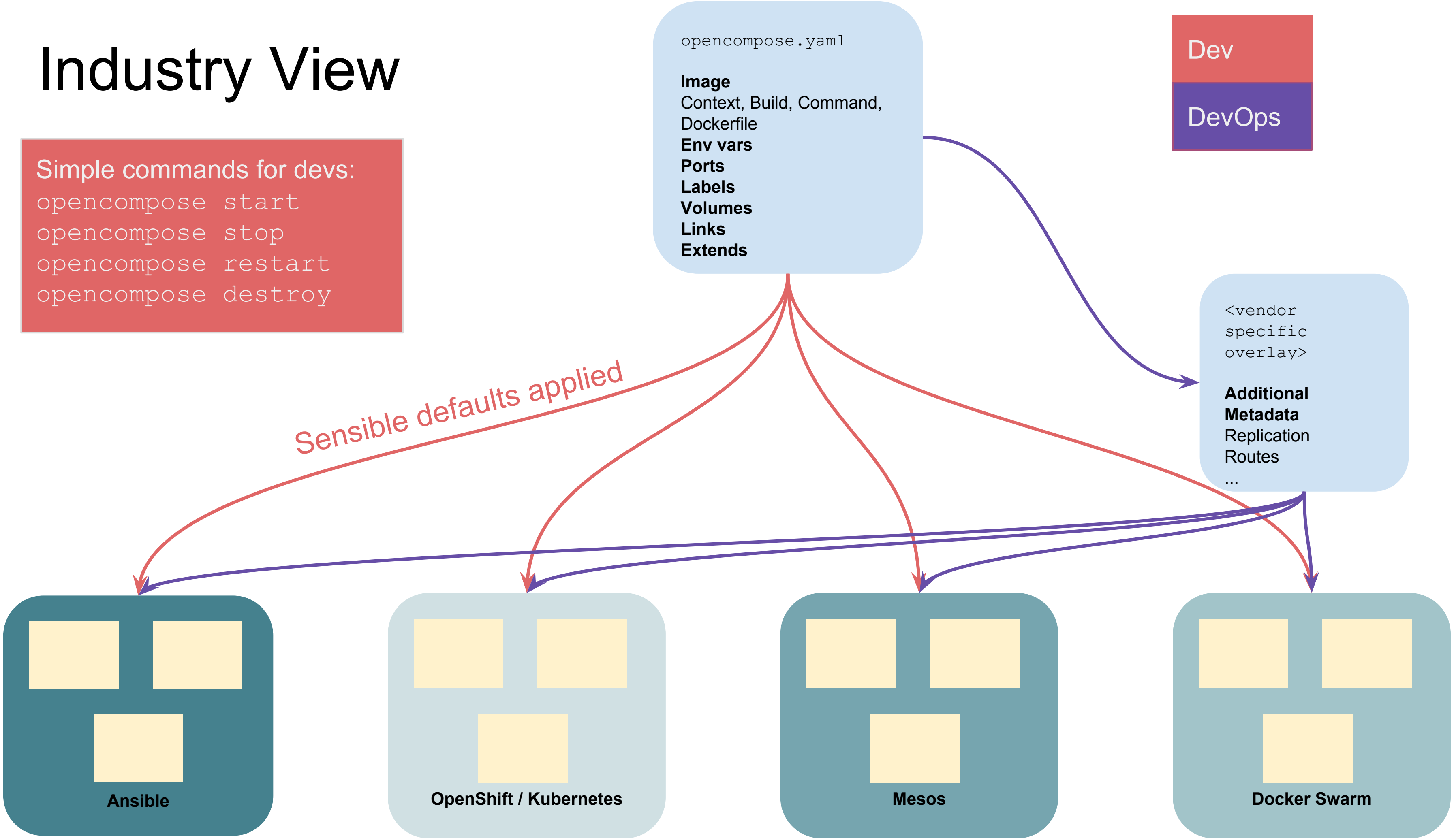
Take your locally developed app, and collaborate with your team and others to describe how your app runs in other environments such as stage and prod.

Describe the dependencies between your services and those produced by others.

Industry View

Simple commands for devs:

```
opencompose start
opencompose stop
opencompose restart
opencompose destroy
```



`opencompose.yaml`

Image
Context, Build, Command,
Dockerfile
Env vars
Ports
Labels
Volumes
Links
Extends

`<vendor specific overlay>`

Additional Metadata
Replication
Routes
...

opencompose.yaml

- File format based Docker compose
- Focused on **what a developer knows**
  - Proposed elements & mapping to kubernetes/ansible
  - filespec
- Standardised via OCI or CNCF
- Supported by key partners (Google, Microsoft, CoreOS)

Naming conventions

- Naming conventions specified for containers
- Naming conventions specified for “OpenCompositions” of containers

Loosely orchestrated containers

- Red Hat provides an Ansible-based engine that can run opencompose.yaml
  - Currently ansible-compose
- Exposes 4 commands/operations to developer
  - Start
  - Stop
  - Restart/Refresh
  - Destroy
- Part of Red Hat Developer product line

OpenShift

- “Sensible Defaults” applied to run directly in OpenShift, with no questions asked
- <vendor specific overlay> file can be provided to override any defaults, including things such as replication, routes and more
- s2i extended to support open compose - multiple orchestrated containers from source code

Library

- Aimed at tool authors (e.g. Eclipse Che, VS Code)
    - Provides a high level abstraction
- Supports multiple implementations
    - Initial targets: OpenShift, Ansible