

30.04.2019



Kolla

Project onboarding

Mark Goddard | mgoddard | Kolla Train PTL



What does Kolla do?

Kolla's mission is to provide production-ready containers and deployment tools for operating OpenStack clouds.

Project background

- Founded during the Kilo release of OpenStack
- Joined the big tent in Liberty cycle
- Diverse contributor & user base
- 135 contributors during Stein cycle
- User Survey results (English responses only)
 - 125 Kolla Ansible deployments
 - 25 deployments with custom tooling (Puppet, Nomad, etc.)

Deliverables

- *openstack/kolla*
 - Docker container images for OpenStack services
 - Used by TripleO and Kolla Ansible
- *openstack/kolla-ansible*
 - Ansible-based tool to deploy Kolla images
- *openstack/kolla-cli*
 - Command Line Interface (CLI) and Python API to Kolla Ansible

Kolla



Overview

- ➔ Docker image build tool
- ➔ Highly customizable using Jinja2
 - Images built from source code or binary packages
 - Support for multiple OS distributions
 - CentOS, RHEL, Ubuntu, Oracle Linux and Debian
- ➔ Multiple architectures
 - x86_64, aarch64 and ppc64le

Combinations

- Image type
 - Source or binary (2)
- Container OS
 - CentOS, RHEL, Ubuntu, Debian, OracleLinux (5)
- Image
 - `mariadb`, `glance-api`, `nova-compute`, etc. (275)
- Question: How many images?

Images

almanach-api	barbican-worker	ceph-base	cinder-base	cron
almanach-base	base	cephfs-fuse	cinder-scheduler	designate-api
almanach-collector	bifrost-base	ceph-mds	cinder-volume	designate-backend-bind9
aodh-api	bifrost-deploy	ceph-mgr	cloudkitty-api	designate-base
aodh-base	blazar-api	ceph-mon	cloudkitty-base	designate-central
aodh-evaluator	blazar-base	ceph-nfs	cloudkitty-processor	designate-mdns
aodh-expirer	blazar-manager	ceph-osd	collectd	designate-pool-manager
aodh-listener	ceilometer-base	ceph-rgw	congress-api	designate-producer
aodh-notifier	ceilometer-central	certmonger	congress-base	designate-sink
barbican-api	ceilometer-compute	chrony	congress-datasource	designate-worker
barbican-base	ceilometer-ipmi	cinder-api	congress-policy-engine	...
barbican-keystone-listener	ceilometer-notification	cinder-backup	freezer-api	

... and more images

- Incredible breadth of support for OpenStack & related services
- Difficult to maintain
- Test coverage poor (but improving) outside of core services
- Rely heavily on testing by users of less common services
- And yet, does any other deployment tool support such a range?

Dockerhub

→ Images published to Dockerhub daily under `kolla` namespace

- <https://hub.docker.com/r/kolla/>
- CentOS, Ubuntu, OracleLinux images

→ Tagged using release name or `master` (development)

- `docker pull kolla/centos-binary-placement-api:stein`



kolla [Edit profile](#)

Community Organization None Worldwide <https://docs.openstack.org/developer/kolla>
 Joined September 25, 2014

Repositories

Displaying 25 of 1835 repositories



kolla/ubuntu-binary-ptp
By [kolla](#) • Updated 10 hours ago

1.0K
Downloads

Container



kolla/ubuntu-binary-kube-base
By [kolla](#) • Updated 10 hours ago

1.9K
Downloads

Container



kolla/oraclelinux-binary-prometheus-memcached-exporter
By [kolla](#) • Updated 10 hours ago

507
Downloads

Container



kolla/ubuntu-binary-mariadb
By [kolla](#) • Updated 10 hours ago

50K+
Downloads

Components

 kolla-build Command Line Interface (CLI)

 Dockerfile templates

kolla-build CLI

- Kolla Command Line Interface (CLI)
- Written in Python
- Interacts with Docker Engine to build and push images
- Configured via `kolla-build.conf` and/or CLI arguments
- Match images based on a regular expression or *profile*

kolla-build CLI Examples

Build all images:

```
kolla-build
```

Build Ubuntu images from source:

```
kolla-build --type source --base ubuntu
```

Build images matching a regular expression:

```
kolla-build ^mariadb$ nova
```

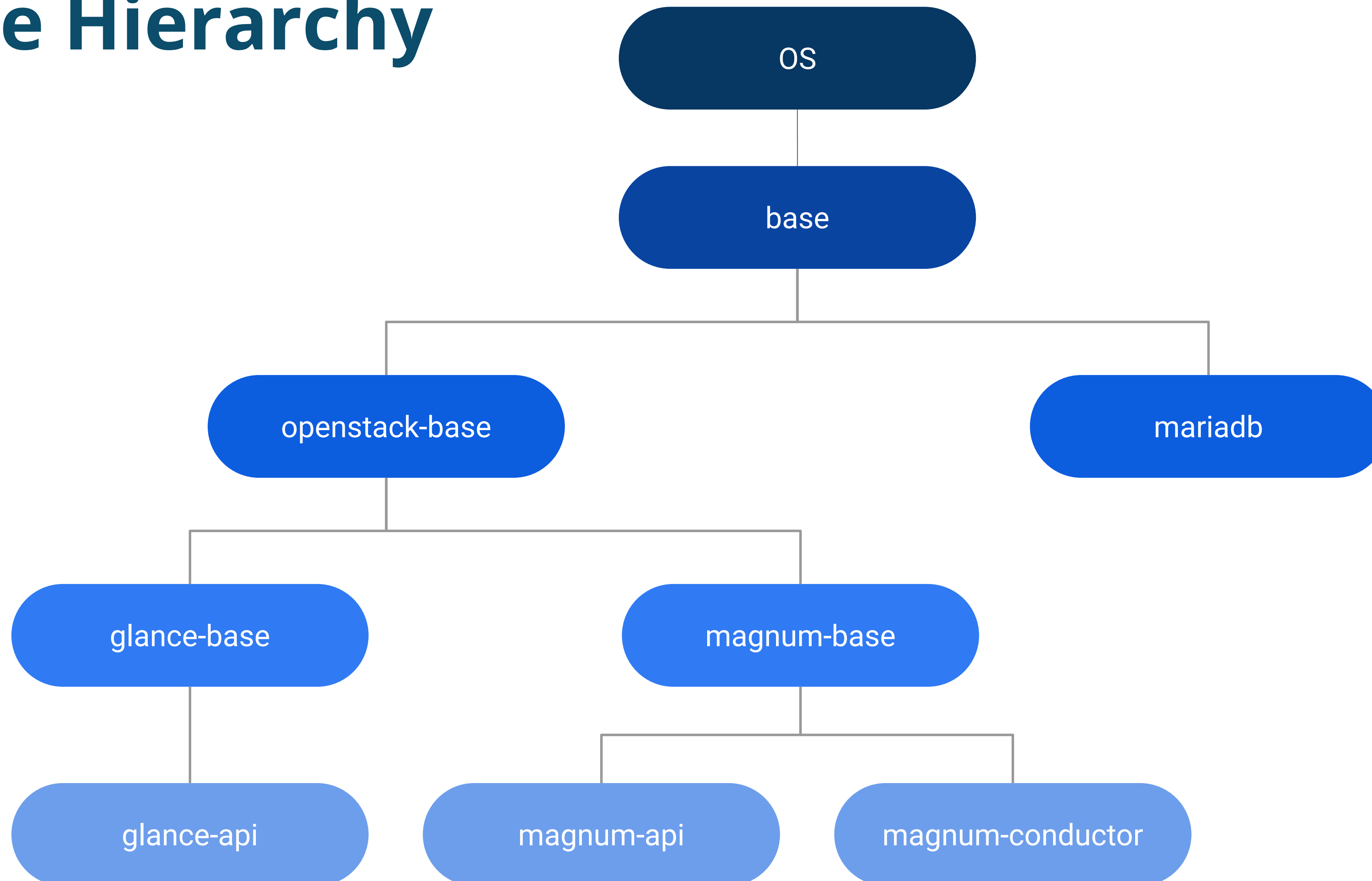
Build images in the *main* profile, push to a registry:

```
kolla-build --profile main --push
```

Dockerfile Templates

- One Jinja2 template per image
- Typically bundled with files & scripts to copy into image
- Templating allows for customisation based on
 - Type (source, binary)
 - OS distro
 - Other `kolla-build` configuration
 - User customisation
- Reuse of base images

Image Hierarchy



Base Image

- ➔ Package repository setup
- ➔ Common packages
- ➔ Users & groups
- ➔ Entry point
 - `dumb-init`
 - `kolla_start`
- ➔ Kolla image API scripts
 - `kolla_set_configs`

OpenStack Base Image

- ➔ Binary images
 - Common OpenStack package dependencies
- ➔ Source images
 - Python build dependencies
 - OpenStack requirements project

Customisation

- Jinja2 template `--template-override`
- Override blocks in Dockerfile templates
- Modify package lists
- Add headers & footers
- Plugins & extensions
- <https://docs.openstack.org/kolla/latest/admin/image-building.html>

Example Dockerfile template - Glance API

```
FROM {{ namespace }}/{{ image_prefix }}glance-base:{{ tag }}
LABEL maintainer="{{ maintainer }}" name="{{ image_name }}" build-date="{{ build_date }}"

{% block glance_api_header %}{% endblock %}

{% import "macros.j2" as macros with context %}

{% if base_distro in ['centos', 'oraclelinux'] %}
    {% set glance_api_packages = ['qemu-img-ev'] %}
{% elif base_distro == 'rhel' %}
    {% set glance_api_packages = ['qemu-img'] %}
{% elif base_distro in ['debian', 'ubuntu'] %}
    {% set glance_api_packages = ['qemu-utils'] %}
{% endif %}

{{ macros.install_packages(glance_api_packages | customizable("packages")) }}

COPY extend_start.sh /usr/local/bin/kolla_glance_extend_start
RUN chmod 755 /usr/local/bin/kolla_glance_extend_start

{% block glance_api_footer %}{% endblock %}
{% block footer %}{% endblock %}

USER glance
```

Kolla Image API

- ➔ Environment variables
 - e.g. `KOLLA_CONFIG_FILE`, `KOLLA_BOOTSTRAP`
- ➔ JSON configuration file
 - Default is `/var/lib/kolla/config_files/config.json`
 - Command to run
 - e.g. `nova-compute`
 - Configuration files to copy
- ➔ https://docs.openstack.org/kolla/latest/admin/kolla_api.html

Repository Layout

- `contrib/`
 - Build overrides templates
 - Not tested in CI, not supported
 - Examples for third party builds
- `doc/`
 - Documentation
- `docker/`
 - Dockerfile templates
- `kolla/`
 - kolla-build CLI

Kolla Ansible



Overview

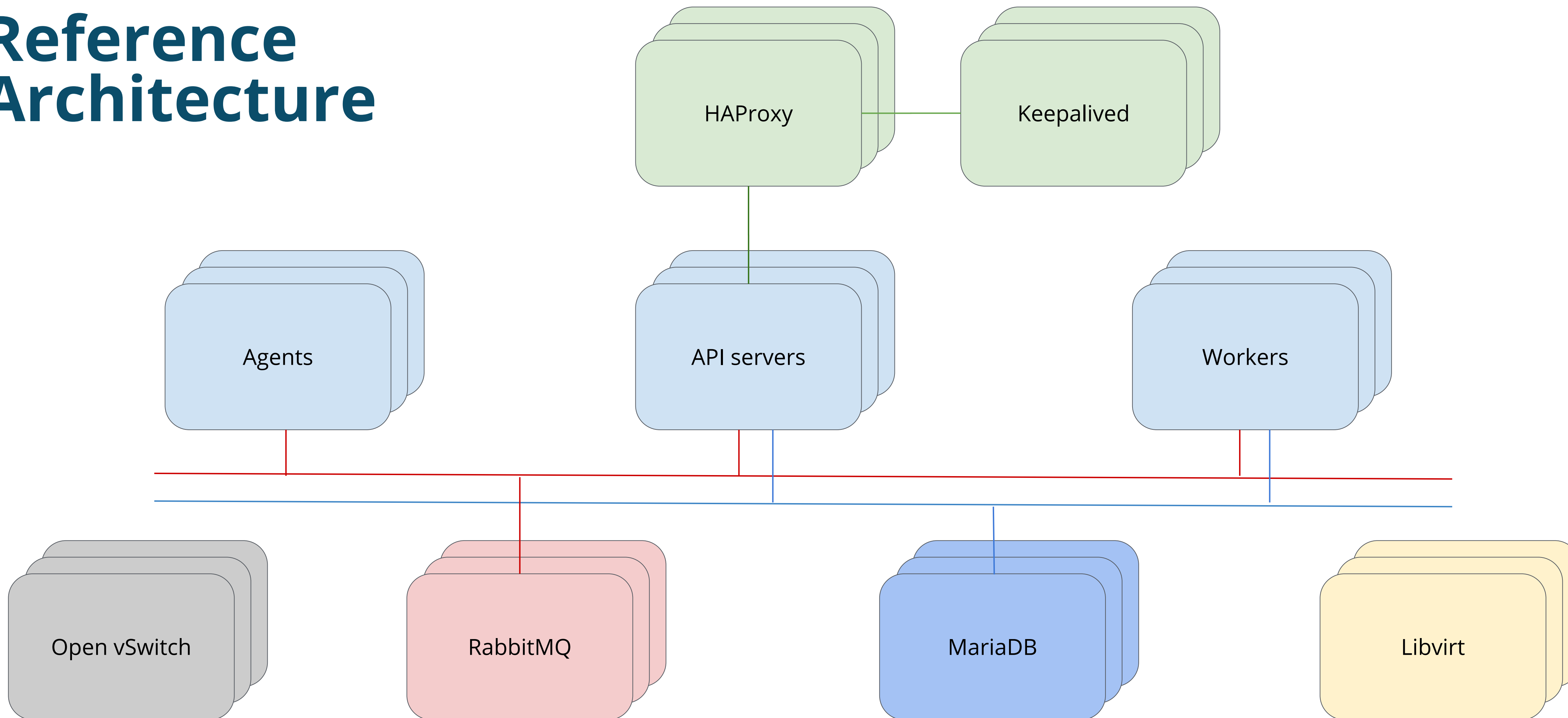
- Ansible-based deployment tool for Kolla images
- Highly available and scalable
- More than 50 different services supported
- Highly customizable by operators
- Fast deployments and upgrade

Flexibility

- ➔ Configure any option*
- ➔ Ability to choose different solutions
 - Monitoring stack
 - Networking
 - Storage
- ➔ Perform targeted operations
 - By service
 - By host

*well, almost any

Reference Architecture



Actions

 prechecks

 deploy

 reconfigure

 upgrade

 pull

 check

More Actions

 `genconfig`

 `stop`

 `destroy`

 `mariadb_recovery`

 `mariadb_backup`

Even More Actions

 bootstrap-servers

 bifrost-deploy

 deploy-servers

Components

Command Line Interface (CLI) tools

- `kolla-ansible`
- `kolla-genpwd`
- `kolla-mergepwd`

Ansible playbooks, roles and plugins

Command Line Interfaces (CLIs)



`kolla-ansible`

- Executes `ansible-playbook`
- `tools/kolla-ansible`



`kolla-genpwd`

- Generates passwords, SSH keys, etc.
- `kolla_ansible/cmd/genpwd.py`



`kolla-mergepwd`

- Merge existing and new passwords
- `kolla_ansible/cmd/mergepwd.py`

Ansible Playbooks

→ **ansible/site.yml**

→ ansible/bifrost.yml

→ ansible/certificates.yml

→ ansible/destroy.yml

→ ansible/detect-release.yml

→ ansible/gather-facts.yml

→ ansible/kolla-host.yml

→ ansible/mariadb_backup.yml

→ ansible/mariadb_recovery.yml

→ ansible/post-deploy.yml

Ansible Roles

- `ansible/roles/<service>`
- `ansible/roles/baremetal`
- `ansible/roles/common`
- `ansible/roles/haproxy-config`
- `ansible/roles/module-load`
- `ansible/roles/prechecks`
- `ansible/roles/service-stop`

Ansible Plugins

- ➔ Modules (`ansible/library/`)
 - **kolla_docker** - Interacts with Docker daemon
 - `bslurp`
 - `kolla_ceph_keyring`
 - `kolla_container_facts`
 - `kolla_toolbox`
- ➔ Action plugins (`ansible/action_plugins/`)
 - **merge_configs**
 - `merge_yaml`

Variables & Inventory



Group variables

- Set global defaults
- `ansible/group_vars/all.yml`



Inventory

- All-in-one
 - `ansible/inventory/all-in-one`
- Multinode
 - Requires modification
 - `ansible/inventory/multinode`

Anatomy of a Role

`ansible/roles/<service>/`

`defaults/main.yml`

Variable defaults

`files/`

Files to copy

`handlers/main.yml`

Handler tasks

`meta/main.yml`

Role dependencies

`tasks/main.yml`

Tasks

`templates/`

Template source files

Anatomy of a Kolla Ansible Role - Defaults

```
ansible/roles/<service>/defaults/main.yml
```

```
glance_services:
  glance-api:
    container_name: glance_api
    group: glance-api
    enabled: true
    image: "{{ glance_api_image_full }}"
    environment: "{{ container_proxy }}"
    volumes:
      - "{{ node_config_directory }}/glance-api/:{{ container_config_directory }}/:ro"
      - "/etc/localtime:/etc/localtime:ro"
      - "{{ glance_file_datadir_volume }}:/var/lib/glance/"
      - "kolla_logs:/var/log/kolla/"
    dimensions: "{{ glance_api_dimensions }}"
    haproxy: <omitted>

glance_api_image: ...
```

Anatomy of a Kolla Ansible Role - Tasks

➔ Roles support all actions for a single service

➔ Use (mostly) consistent patterns

```
ansible/roles/<service>/
```

```
tasks/main.yml
```

```
tasks/<action>.yml
```

```
includes {{ kolla_action }}.yml
```

```
config, deploy, pull, reconfigure, upgrade
```

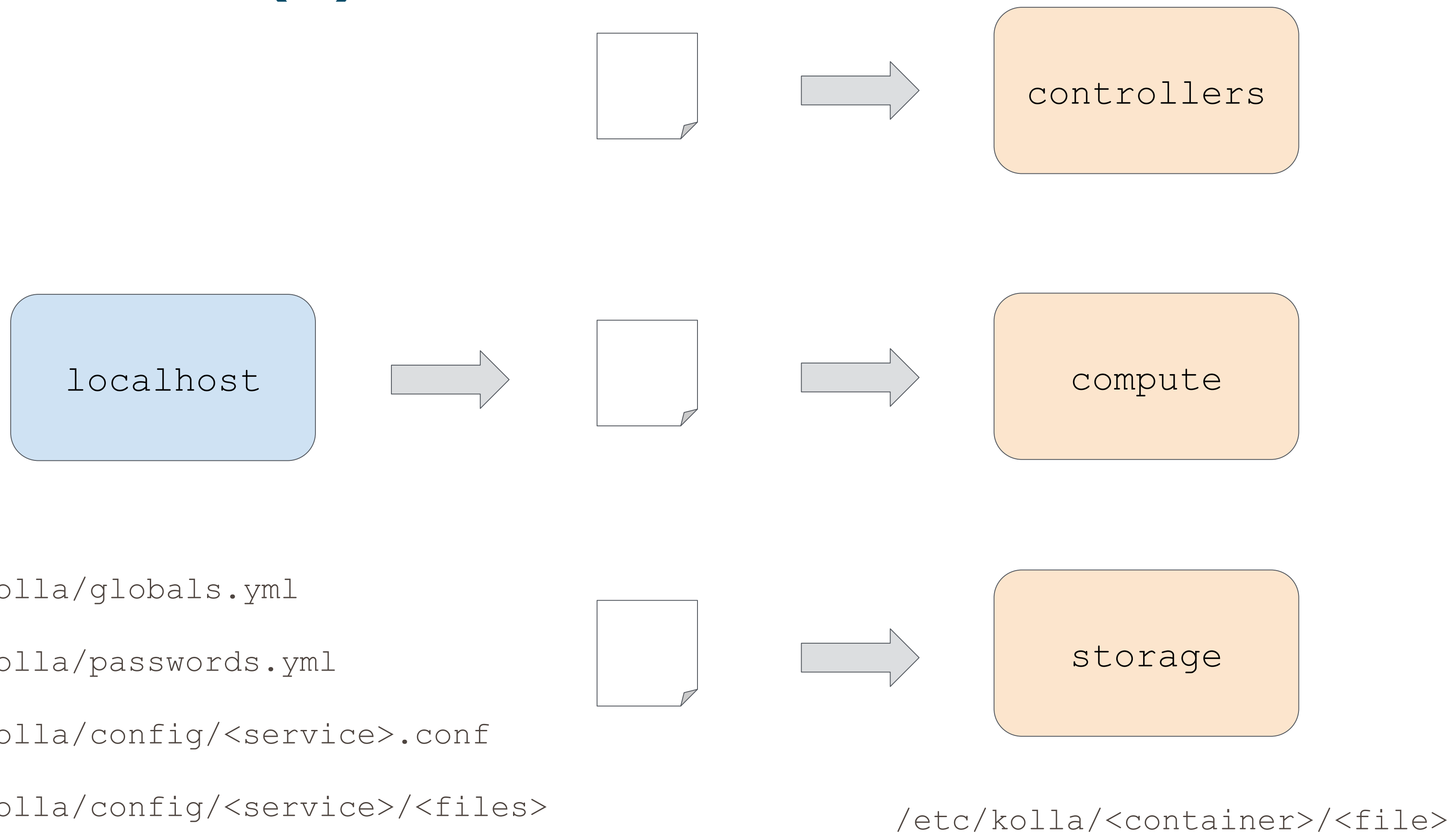
Anatomy of a Kolla Ansible Role - Deploy Action

- ➔ Register service, endpoints, users & roles in Keystone
 - `register.yml`
- ➔ Generate configuration files on remote hosts
 - `config.yml`
- ➔ Bootstrap - create DB & user
 - `bootstrap.yml`
- ➔ Bootstrap service - sync DB schemas
 - `bootstrap_service.yml`
- ➔ Flush handlers
 - Create, recreate or restart containers

Configuration

- Generate config files for each container on the remote hosts
- Contents may need to be different on different hosts
- Combination of defaults in Kolla Ansible and user configuration on `localhost`

Configuration (2)



Configuration Patterns

- ➔ INI merge
 - Default INI template in Kolla Ansible
 - User may provide INI files to set or override arbitrary options
 - `merge_configs`
- ➔ YAML merge
 - Default YAML template in Kolla Ansible
 - User may provide YAML files to set or override sections
 - `merge_yaml`

Configuration Patterns (2)

- File Override
 - Default template in Kolla Ansible
 - User may provide a file to replace contents entirely
 - `with_first_found, template`

- Directory glob
 - User may provide files matching a pattern in a directory
 - Copied or templated to remote host
 - `copy, template`

Other Things

→ Contributed scripts & files, Vagrant configuration

- `contrib/`

→ Documentation source

- `docs/`

→ Configuration files

- `etc/`

→ Tools & scripts

- `tools/`

Kolla CLI



Kolla CLI

- Started by Oracle
- Python-based CLI
- Replaces `kolla-ansible` CLI
- Flexible inventory & variable manipulator
- So far not too much uptake
- Authors no longer active in community
- Maintainers needed

Kayobe

Kayobe

- Not under Kolla project governance
- Bare metal provisioning and configuration of control plane hosts
- Physical network configuration
- Version controlled configuration
- <https://kayobe.readthedocs.io>
- <https://www.slideshare.net/MarkGoddard2/to-kayobe-or-not-to-kayobe>
- **Try it at the “A Universe from Nothing” workshop today at 16:20 - 17:50**

The Kase for Kolla



Why (Docker) Containers?

- Isolation
 - Dependencies
 - File system
 - Processes
 - Not everything though - host networking
- Immutable
- Reduced privileges (where possible)
- Standard deployment model

Why Ansible?

- Simplicity & predictability
- Need to orchestrate container deployment
- Why not Kubernetes?
 - Complexity has to end somewhere
- Also good for general automation tasks

Help!



We Need Your Help!

- Kolla is a real community project
- Maintained by operators
- Review bandwidth of core team limited - no one is full time
- Lots of less frequent contributors
- Has enough momentum to survive
- More help needed to *thrive*

How Can I Help?

- There are many ways to help
- Cater to your skillset, availability & interests
- Do you fit one of the following profiles?

The Noob

- ➔ Just getting started? Great!
- ➔ Try out the all-in-one environment
- ➔ Jump up to a multi-node install
- ➔ Try going off the beaten track
- ➔ Ask in IRC for help
- ➔ Documentation out of date?
 - Raise a bug, or propose a fix

The Conscientious Operator

- ➔ Running a cloud deployed via Kolla? No time for regular contribution?
 - No problem
- ➔ Find a bug?
 - Report it on Launchpad, providing as much info as possible
- ➔ Fixed a bug?
 - Propose the fix via Gerrit
 - Or ask someone else to crank the handle for you
- ➔ Added a feature downstream?
 - Propose it via Gerrit, or if you have no time, code dump

The Part Time Upstreamer

- ➔ Consider watching for *new* patches in Gerrit
 - Review those in areas you use and/or understand
 - **Quality** over quantity
- ➔ Subscribe to Launchpad bug feeds
 - <https://launchpad.net/kolla> and <https://launchpad.net/kolla-ansible>
 - Monitor for issues in areas you use and/or understand
- ➔ Join IRC (`#openstack-kolla`)
- ➔ Attend weekly meetings - 1500UTC on Wednesdays in `#openstack-meeting-4`

The Part Time Upstreamer (2)

- ➔ Subscribe to openstack-discuss@openstack.org
 - Watch for [kolla] tag
- ➔ Attend the virtual PTG - <https://etherpad.openstack.org/p/kolla-train-ptg>
- ➔ Help with testing prior to releases

The Core Reviewer

- Cores have ability to approve patches
- There aren't strict rules for becoming core
- Demonstrate responsibility, understanding & care

The Project Team Lead

→ PTL elected for each cycle

→ It could be you!

THANKS.

Questions?



openstack



@OpenStack



openstack



OpenStackFoundation

