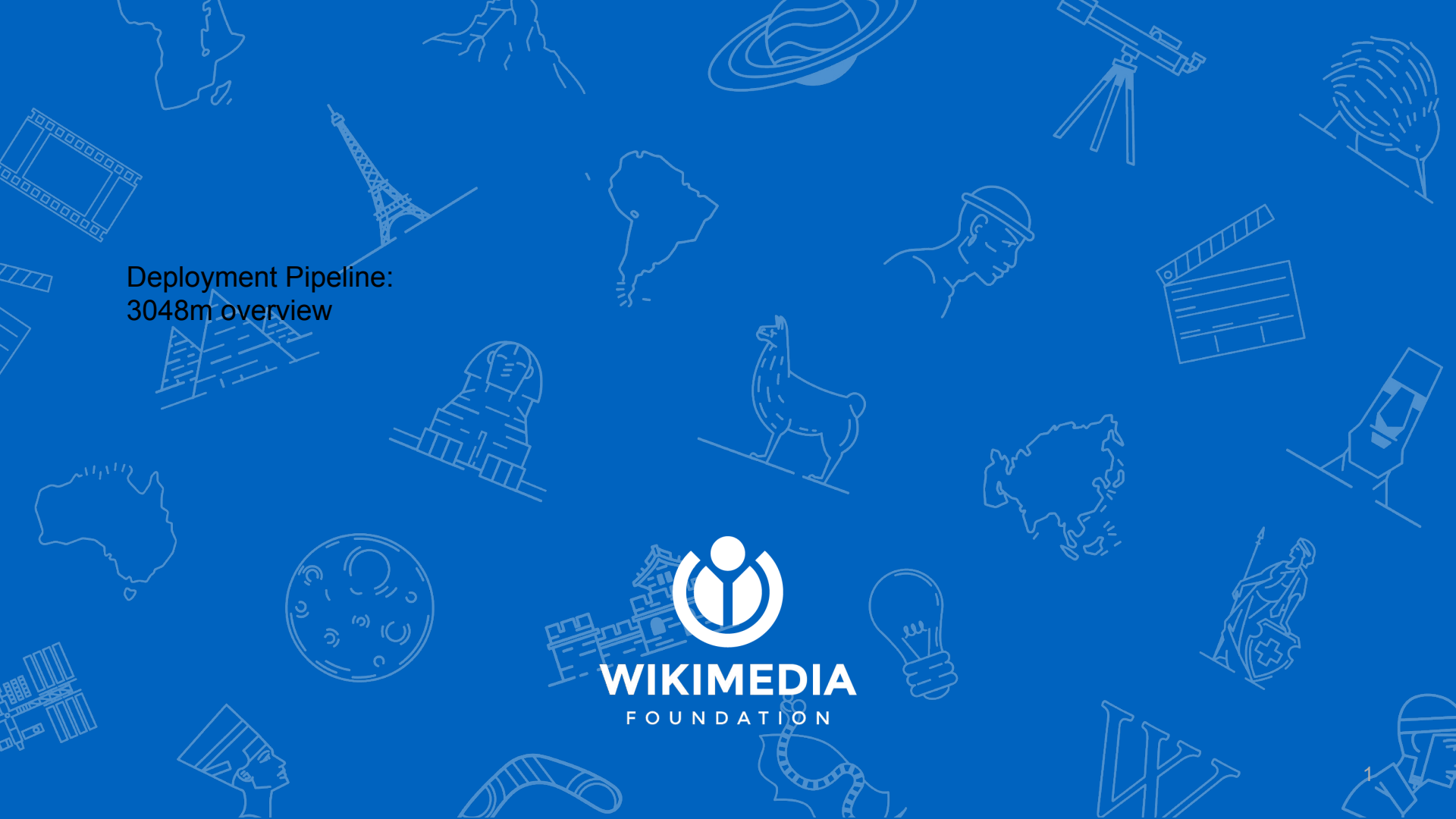
Deployment Pipeline:
3048m overview

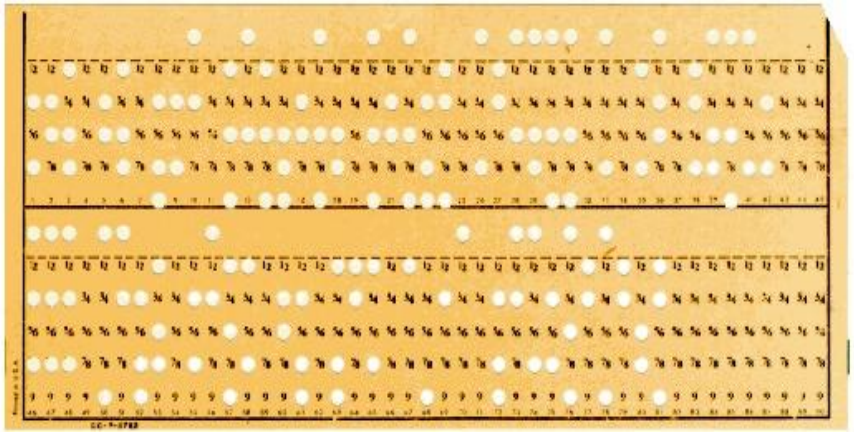


Problem statement

WIKIMEDIA
FOUNDATION

How does a user deploy or run code ?

WIKIMEDIA FOUNDATION



```
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[[:]attribs]] [/O[[:]sortord]]
    [/S] [/B] [/L] [/C[H]]

  [drive:][path][filename]   Specifies drive, directory, and/or files to list.
  /P       Pauses after each screenful of information.
  /W       Uses wide list format.
  /A       Displays files with specified attributes.
  attribs    D  Directories    R  Read-only files          H  Hidden files
             S  System files   A  Files ready to archive   -  Prefix meaning "not"
  /O       List by files in sorted order.
  sortord    N  By name (alphabetic)        S  By size (smallest first)
             E  By extension (alphabetic)   D  By date & time (earliest first)
             G  Group directories first     -  Prefix to reverse order
             C  By compression ratio (smallest first)
  /S       Displays files in specified directory and all subdirectories.
  /B       Uses bare format (no heading information or summary).
  /L       Uses lowercase.
  /C[H]    Displays file compression ratio; /CH uses host allocation unit size.

Switches may be preset in the DIRCMD environment variable.  Override
preset switches by prefixing any switch with - (hyphen)--for example, /-W.

C:\>_
```

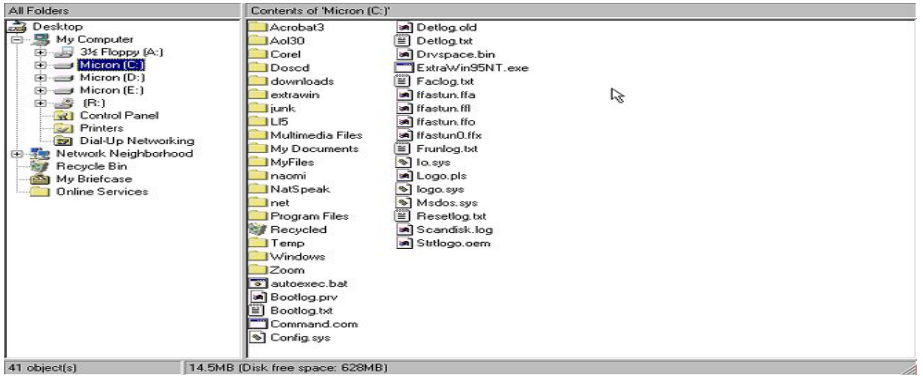
```
Last login: Thu Aug 18 16:08:43 2011 from c-67-180-28-216
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
------------------------------------------------------------------------
                    W E L C O M E   T O   S T A R
                brought to you by Instructional Support Group
------------------------------------------------------------------------
 - Star is an 8 core (1.6 Ghz) UltraSparc T2 running Solaris 10.
 - Intel binaries will not run here.
 - See the green bulletin boards by 199 Cory, 271/330 Soda for general info.
 - For more info on labs please see http://inst.eecs.berkeley.edu
 - To find scheduled office hours for staff please finger inst@inst
 - Send questions/problems to inst@EECS.Berkeley.EDU
 - SunRay Server for 271 Soda

 - Any processes left running for over 24 hours will be killed.

 - Ongoing COMPUTER HELP SESSIONS: see http://www.CSUA.Berkeley.EDU
------------------------------------------------------------------------
Date                                Notices
-----   ---------------------------------------------------------------

(type "more /etc/motd" to repeat this message)

star [119] ~ #
```



WIKIMEDIA FOUNDATION

How many things did the user have to care about ?

WIKIMEDIA FOUNDATION

Let’s see now how a developer deploys/executes code in WMF

- **Writes the code**
- **Tests it locally**
- **Pushes to gerrit**
- **Waits for CI**
- **Gets code merged**
- **Builds a “deploy” repo manually**
  - Running composer install, pip install, npm install, whatever
- **Deploys deploy repo to production**
  - At best with a blue/green method
- **Tries to figure out why machine XYZ does not work**
  - Is it CPU contention ?
  - Did Out of Memory exception occur ?
  - Is the machine out of disk space ?
  - Is the machine dead ?
  - Is the code in production the same they tested X days ago anymore ?

- **Writes the code**
- **Tests it locally**
- **Pushes to gerrit**
- **Waits for CI**
- **Gets code merged**
- **Builds a “deploy” repo manually**
  - Running composer install, pip install, npm install, whatever
- **Deploys deploy repo to production**
  - At best with a blue/green method
- **Tries to figure out why machine XYZ does not work**
  - Is it CPU contention ?
  - Did Out of Memory exception occur ?
  - Is the machine out of disk space ?
  - Is the machine dead ?
  - Is the code in production the same they tested X days ago anymore ?

WIKIMEDIA FOUNDATION

Meanwhile

- No elasticity (can’t react to increased demand)
- Can’t run multiple versions of code simultaneously
- Colocating services leads to shared dependencies (e.g. all in nodejs6)
- No hardware fault tolerance
- development environment != deploy environment
- First deployment takes forever
  - Find hardware
  - Find out how to create deploy repo
  - Figure out how to configure deployment tool (scap)
  - And do it in beta cluster first
  - Talk with >1 teams to get code deployed

WIKIMEDIA
FOUNDATION

Abstract some problems away

- A scheduler to assign workloads (aka code) to hosts
- Decouple the service from the workloads
  - Allowing actually canary deployments
  - We kind of have this already via pybal
- Build the “deploy repo” automatically
  - Yes, that’s the docker image
- Use the deploy repo for development as well

WIKIMEDIA FOUNDATION

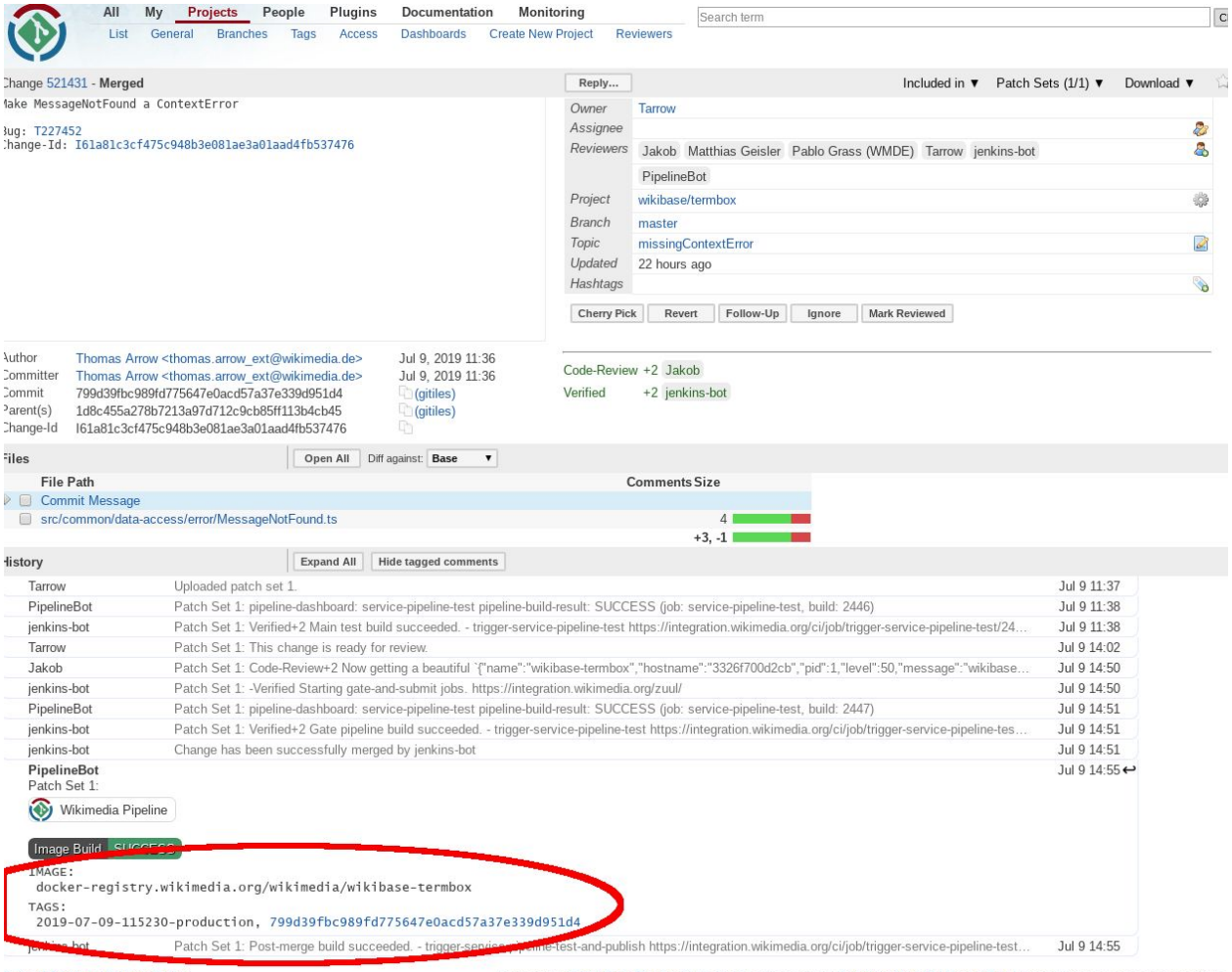
Build the “deploy repo”

- The major work of the pipeline
  - Powered by jenkins and blubber
- Upload a change
- Tests are run
- Merge it
- Obtain the image
- Optionally run integration tests
- Deploy it

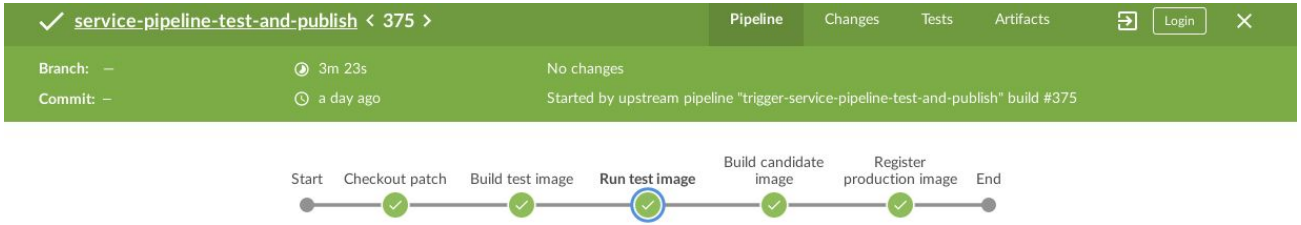
WIKIMEDIA
FOUNDATION

Lights! Camera! Action!

WIKIMEDIA FOUNDATION



- Upload to gerrit
- Tests run
- Merge
- Get image



Run test image - 1m 2s

Shell Script 1m 2s

```
+ exec docker run --rm 28fd41b1a82c

> wikibase-termbox@0.1.0 test /opt/lib
> npm-run-all test:*


> wikibase-termbox@0.1.0 test:lint /opt/lib
> vue-cli-service lint --no-fix . && stylelint --syntax scss 'src/**/*.(vue|scss)'

 DONE  No lint errors found!
Browserslist: caniuse-lite is outdated. Please run next command `npm update caniuse-lite browserslist`

> wikibase-termbox@0.1.0 test:unit /opt/lib
> vue-cli-service test:unit

PASS tests/unit/server/data-access/WaitingForLanguageWikibaseContentLanguagesRepo.spec.ts (10.432s)
PASS tests/unit/store/entity/getters.spec.ts (10.821s)
PASS tests/unit/common/TermboxServices.spec.ts (10.821s)
PASS tests/unit/client/data-access/EntityRepository.spec.ts (11.162s)
PASS tests/unit/server/axios/axiosFactory.spec.ts (11.383s)
PASS tests/unit/client/axios/editTokenRequestInterceptor.spec.ts (11.414s)
PASS tests/unit/store/language/mutations.spec.ts (11.562s)
PASS tests/unit/client/init.spec.ts (11.639s)
PASS tests/unit/server/data-access/ContentLanguagesLanguageTranslationRepo.spec.ts
PASS tests/unit/client/data-access/AxiosWritingEntityRepository.spec.ts (12.046s)
PASS tests/unit/store/entity/actions.spec.ts (12.077s)
PASS tests/unit/server/assertAndGetConfig.spec.ts (12.154s)
PASS tests/unit/client/data-access/StringMWCookieStore.spec.ts
PASS tests/unit/client/axios/axiosFactory.spec.ts (12.218s)
PASS tests/unit/server/data-access/AxiosSpecialPageEntityRepo.spec.ts (12.417s)
PASS tests/unit/server/data-access/AxiosWikibaseContentLanguagesRepo.spec.ts (12.437s)
PASS tests/unit/store/entity/mutations.spec.ts (12.485s)
PASS tests/unit/server/data-access/ContentLanguagesLanguageRepo.spec.ts
PASS tests/unit/common/EntityInitializer.spec.ts
PASS tests/unit/store/language/getters.spec.ts
```

Steps are (going to be) configurable

Dockerfiles are hard

- Non pinned/frozen dependencies
- Changes to source code invalidate builds
- Running as root
- End up in bloated images
- Use external services
- Paths are not standardized

WIKIMEDIA
FOUNDATION

Decision: Abstract Dockerfile

- Blubber
- A declarative tool (vs Dockerfile imperative nature) in Golang
- Configuration in YAML
- Supports slimming down images using multi-stage Dockerfiles
- Supports policies
- Source code files/dirs in images are not owned by root
- Code is not executed as root
- https://wikitech.wikimedia.org/wiki/Blubber

WIKIMEDIA FOUNDATION

Blubber HelloWorld

```
version: v3
base: docker-registry.wikimedia.org/wikimedia-stretch

variants:
  hello:
    entrypoint: [echo, "Hello, world!"]
```

WIKIMEDIA FOUNDATION

Blubber HelloWorld

```
FROM docker-registry.wikimedia.org/wikimedia-stretch
USER "root"
ENV HOME="/root"
RUN groupadd -o -g "65533" -r "somebody" && useradd -o -m -d "/home/somebody" -r -g
"somebody" -u "65533" "somebody" && mkdir -p "/srv/app" && chown "65533":"65533"
"/srv/app" && mkdir -p "/opt/lib" && chown "65533":"65533" "/opt/lib"
RUN groupadd -o -g "900" -r "runuser" && useradd -o -m -d "/home/runuser" -r -g
"runuser" -u "900" "runuser"
USER "somebody"
ENV HOME="/home/somebody"
WORKDIR "/srv/app"
COPY --chown=65533:65533 [".", "."]
USER "runuser"
ENV HOME="/home/runuser"
ENTRYPOINT ["echo", "Hello, world!"]
```

WIKIMEDIA FOUNDATION

Decisions: Pipeline images

- Images are available for anyone to use under docker-registry.wikimedia.org
- But only the pipeline can push images
- Production images will always be based on base images provided internally
  - Yes, that means no Dockerhub (nor any other registry out there)
  - Debian based (Jessie, Stretch, Buster) to match production
- Tooling will be built to ensure as simple as possible upgrades
- **We don’t tag images as latest**

WIKIMEDIA FOUNDATION

Why no Dockerhub/gcr.io/etc?

- https://vulnerablecontainers.org/official/
  - 17 for nodejs image
  - 1 for php, 23 for php-zendserver
  - **876** for rails
  - **470** for django
  - **329** for java
- In 2018, 17 dockerhub images were found with cryptominers in them

WIKIMEDIA FOUNDATION

But also (more importantly)!

- We need to upgrade easily and quickly for the next heartbleed, shellshock, ghost, you name it
- And that can only happen if the Wikimedia community controls the entire supply chain of images

WIKIMEDIA
FOUNDATION

Decision: Image Orchestration

- Docker on its own is not great in orchestrating workloads
  - Networking can painful
  - Global state of a workload is difficult to discern
  - Metrics/logging need to be implemented
  - It’s pretty good at executing workloads though
- Kubernetes is the current de facto standard for orchestrating the deployment of workloads
- WMF is a member of CNCF of which Kubernetes is a graduated project
- SRE team already had some knowledge

Kubernetes has a scheduler

- You tell it the size of your workload
  - CPU
  - Memory
  - # of instances
  - Other stuff (GPU needs, workload spreading needs, etc)
- It will do the best it can to nicely spread workloads across multiple nodes
- It will reschedule workloads if a node fails

WIKIMEDIA FOUNDATION

Kubernetes decouples the service from workloads

- Based on a flexible tagging scheme
- Spawn X workloads, have Y of them serving production requests
  - Allows implementing canaries
- Monitors workload endpoints and depools failing ones
  - Until they are functional again, no production traffic
- Implements staged blue/green deployments
  - Have X workloads, upgrade them in batches (% at a time), stop/rollback if problems arise

Kubernetes is declarative

- Define
  - The version
  - The # of instances
  - Policies, configurations etc
- It will make it happen (assuming you defined stuff correctly)
- No more caring about what the previous version/state of a app was

WIKIMEDIA FOUNDATION

Kubernetes runtimes are pluggable

- There’s now a standard called Open Container Initiave (OCI)
- We are not locked in to Docker.
- Any OCI compliant container runtime engine (CRE) will do
- Examples:
  - **Docker**
  - CRI-O
  - containerd

WIKIMEDIA FOUNDATION

Kubernetes networking is pluggable

- It encouraged the creation of a standard called CNI that anyone can implement
- Cisco, AWS, Apstra, Cilium, Contiv, Contrail, Calico, Flannel, OpenVSwitch and many more all implement it

Decision: Calico

- After evaluation of a few CNI plugins, calico was chosen because:
  - It was compatible with our current networking setup
  - It avoided the complexity of an overlay network (e.g. VXLAN)
  - It supports Network policies
    - Required for setting up firewalling from/to workloads

WIKIMEDIA FOUNDATION

Kubernetes: A lot of YAML and new concepts

- Many many new concepts: Pod, StatefulSet, Deployment, ReplicaSet, Service, NetworkPolicy, Endpoint, ConfigMap, Volume (the list goes on)
  - 9 for workloads
  - 4 for services
  - 6 for configuration and storage
  - A ton more for metadata, cluster management etc
- All of them defined in YAML many many lines long each

Decision: Helm

- Helm is a package manager/deployment tool for the kubernetes environment
- Abstracts kubernetes deployments complexity away
- The de facto standard again in the kubernetes community
- Group your YAML, make it more configurable, deploy it

WIKIMEDIA FOUNDATION

Helm charts

- Group and template the various kubernetes resources
- Templates allow you to set values (e.g. version of app)
- Share the chart with the world
- Use it for local development as well (not just for production)
- Wikimedia has their own repo:
  - https://releases.wikimedia.org/charts/
  - https://gerrit.wikimedia.org/g/operations/deployment-charts/

WIKIMEDIA FOUNDATION

Decision: Provide Helm chart scaffolding

- git clone https://gerrit.wikimedia.org/g/operations/deployment-charts/
- ./create_new_service.sh
- Answer questions
- (Optional) edit chart
- Submit for review

WIKIMEDIA FOUNDATION

Helm chart tests

- Scaffolding helm tests rely on service-checker:
  - https://gerrit.wikimedia.org/g/operations/software/service-checker
- Probes:
  - OpenAPI/Swagger spec endpoint (https://swagger.io/specification/)
  - All the endpoints described in it
- But it’s really easy to add more tests/customize existing ones
- Pipeline integration tests run exactly that

WIKIMEDIA FOUNDATION

Pipeline status

- 9 services currently in the pipeline
  - Blubberoid, citoid, cxserver, eventgate-analytics, eventgate-main, mathoid, sessionstore, termbox, zotero
- Services to be added soon
  - Restrouter (part of RestBASE), changepop, cpjobqueue, wikifeeds (part of MCS)

WIKIMEDIA FOUNDATION

Roadmap

- Document the pipeline better
- Create a local development environment (based on minikube)
- Encourage/handhold other service owners to migrate their services to the pipeline
- Add tooling to automatically respond to increased/decreased demand
- Allow developers to create Highly Available endpoints for apps
- TLS demarcation
- Telemetry