CONTAINERIZING FOREMAN DEPLOYMENTS

TAKE #42

$ whoami

Evgeni Golov

Principal Software Engineer at Red Hat

ex-Consultant at Red Hat

Debian Developer

♥ FOSS ♥

♥ automation ♥

- 2009: Foreman gets created
- 2013: Docker gets created
- 2019: Ohad adds a `Dockerfile` to `foreman.git`
- 2025: It's still not possible to run Foreman as a container in production

CLASSICAL FOREMAN DEPLOYMENT

- RPM and DEB packages
- orchestrated by Installer/Puppet
- users have lots of control (OS, plugins, etc)

EXISTING `Dockerfile`

- built from source
- no orchestration
- no plugins
- no control

EVERYONE WANTS KUBERNETES

EVERYONE WANTS KUBERNETES

or at least Podman

2024: THE RESEARCH

2024: THE RESEARCH

- let's throw *everything* away and start fresh

2024: THE RESEARCH

- let's throw *everything* away and start fresh
- maybe keep (RPM) packages?

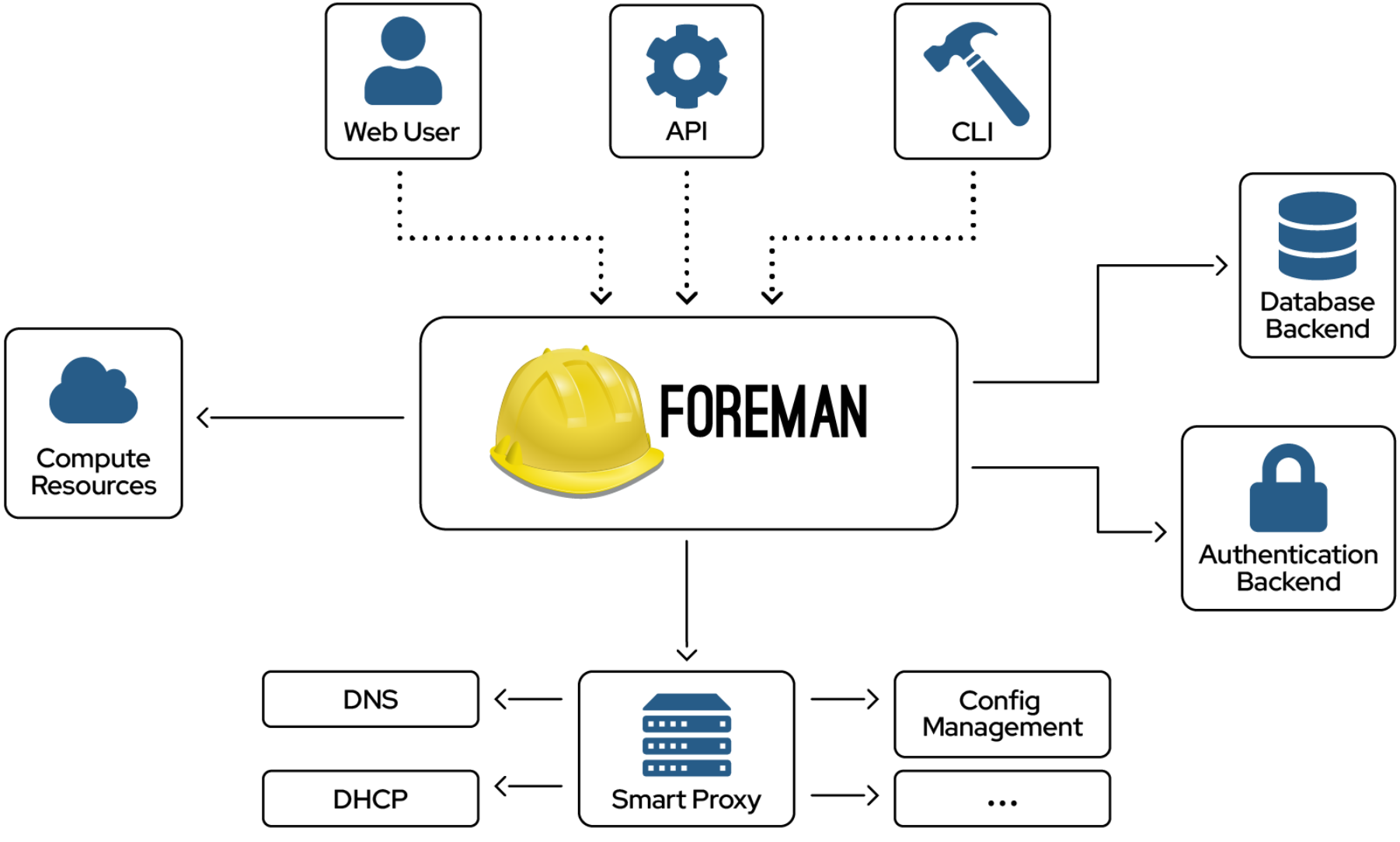
2024: THE RESEARCH

- let's throw *everything* away and start fresh
- maybe keep (RPM) packages?
- make it work with Podman, Kubernetes later

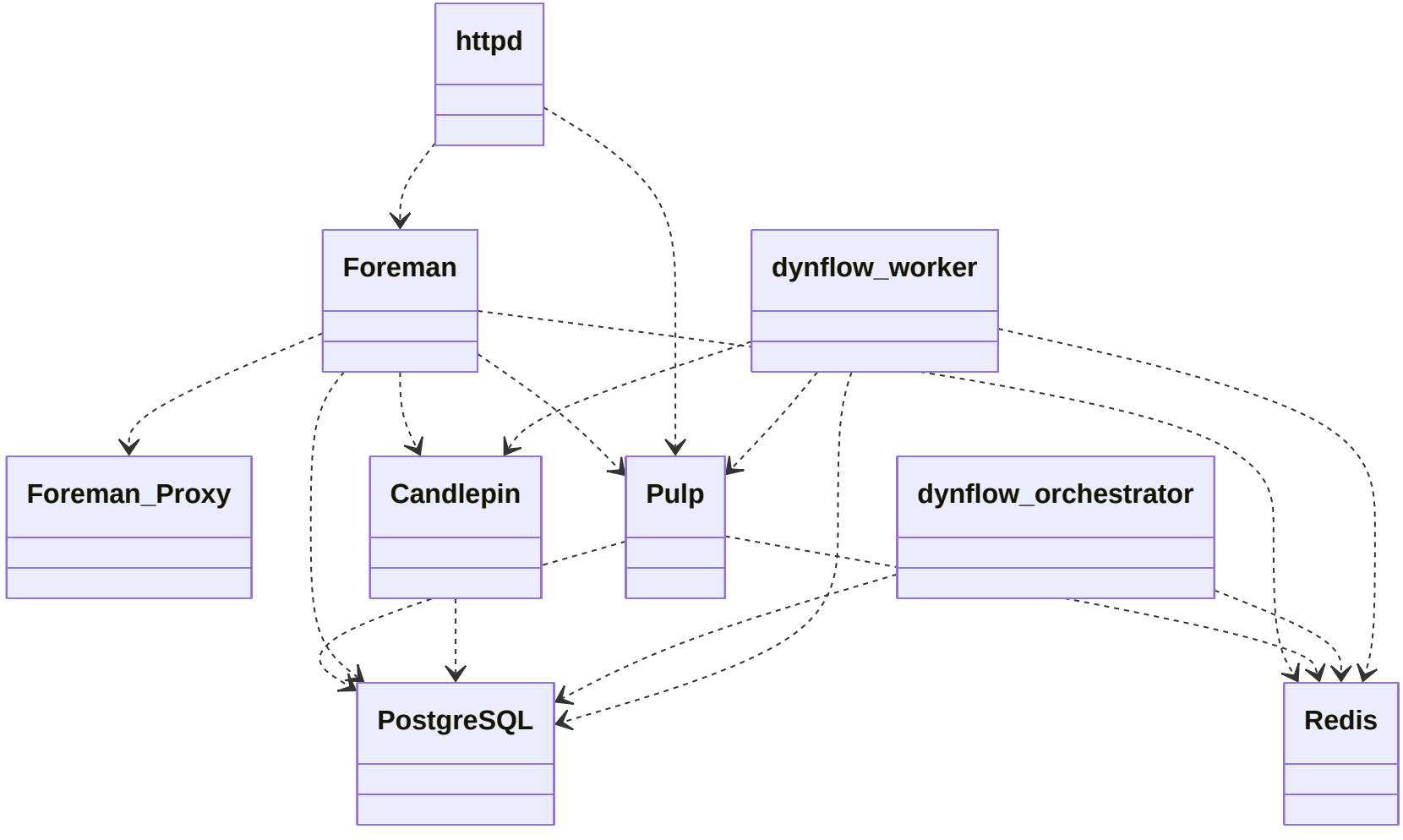
WHY PODMAN?

- We still need "one VM all services" deployment
- Podman (since 5.0) has very good systemd integration
- Can use Kubernetes YAML as input

WHICH SERVICES?



WHICH SERVICES?



CANDLEPIN CONTAINER

```
1 FROM quay.io/centos/centos:stream9
2 RUN dnf -y install candlepin
3 CMD ["/usr/libexec/tomcat/server", "start"]
```

CANDLEPIN CONTAINER

```
1 FROM quay.io/centos/centos:stream9
2 RUN dnf -y install candlepin
3 CMD ["/usr/libexec/tomcat/server", "start"]
```

CANDLEPIN CONTAINER

```
1 FROM quay.io/centos/centos:stream9
2 RUN dnf -y install candlepin
3 CMD ["/usr/libexec/tomcat/server", "start"]
```

CANDLEPIN CONTAINER

```
1 FROM quay.io/centos/centos:stream9
2 RUN dnf -y install candlepin
3 CMD ["/usr/libexec/tomcat/server", "start"]
```

CANDLEPIN CONTAINER

```
# cat /etc/containers/systemd/candlepin.container
[Container]
ContainerName=candlepin
HostName=quadlet.example.com
Image=quay.io/ehelms/candlepin:4.4.14
Network=host
Secret=candlepin-ca-cert,
       target=/etc/candlepin/certs/candlepin-ca.crt,
       mode=0440,type=mount
…
Secret=candlepin-candlepin-conf,
       target=/etc/candlepin/candlepin.conf,
       mode=0440,type=mount
…
```

CANDLEPIN CONTAINER

```
# cat /etc/containers/systemd/candlepin.container
[Container]
ContainerName=candlepin
HostName=quadlet.example.com
Image=quay.io/ehelms/candlepin:4.4.14
Network=host
Secret=candlepin-ca-cert,
       target=/etc/candlepin/certs/candlepin-ca.crt,
       mode=0440,type=mount
…
Secret=candlepin-candlepin-conf,
       target=/etc/candlepin/candlepin.conf,
       mode=0440,type=mount
…
```

CANDLEPIN CONTAINER

```
# cat /etc/containers/systemd/candlepin.container
[Container]
ContainerName=candlepin
HostName=quadlet.example.com
Image=quay.io/ehelms/candlepin:4.4.14
Network=host
Secret=candlepin-ca-cert,
       target=/etc/candlepin/certs/candlepin-ca.crt,
       mode=0440,type=mount
…
Secret=candlepin-candlepin-conf,
       target=/etc/candlepin/candlepin.conf,
       mode=0440,type=mount
…
```

CANDLEPIN CONTAINER

```
# cat /etc/containers/systemd/candlepin.container
[Container]
ContainerName=candlepin
HostName=quadlet.example.com
Image=quay.io/ehelms/candlepin:4.4.14
Network=host
Secret=candlepin-ca-cert,
       target=/etc/candlepin/certs/candlepin-ca.crt,
       mode=0440,type=mount
…
Secret=candlepin-candlepin-conf,
       target=/etc/candlepin/candlepin.conf,
       mode=0440,type=mount
…
```

CANDLEPIN CONTAINER

```
# cat /etc/containers/systemd/candlepin.container
[Container]
ContainerName=candlepin
HostName=quadlet.example.com
Image=quay.io/ehelms/candlepin:4.4.14
Network=host
Secret=candlepin-ca-cert,
       target=/etc/candlepin/certs/candlepin-ca.crt,
       mode=0440,type=mount
…
Secret=candlepin-candlepin-conf,
       target=/etc/candlepin/candlepin.conf,
       mode=0440,type=mount
…
```

CANDLEPIN CONTAINER

```
# systemctl status candlepin
● candlepin.service
     Loaded: loaded
             (/etc/containers/systemd/candlepin.container;
             generated)
     Active: active (running)
   Main PID: 1330 (conmon)
     Memory: 975.1M
        CPU: 19.567s
     CGroup: /system.slice/candlepin.service
             ├─libpod-payload-ef42219c81f60b6287df9caff
             │ └─1346 /usr/lib/jvm/jre-17/bin/java …
             └─runtime
               └─1330 /usr/bin/conmon …
```

FOREMAN CONTAINER

```
FROM quay.io/centos/centos:stream9

RUN dnf install -y foreman foreman-postgresql\
    foreman-service foreman-redis foreman-dynflow-sidekiq

ARG FOREMAN_PLUGINS="foreman-tasks\
                     foreman_remote_execution\
                     katello"

RUN for PLUGIN in ${FOREMAN_PLUGINS}; do … ; done

CMD /usr/share/foreman/bin/rails db:migrate &&\
    /usr/share/foreman/bin/rails db:seed &&\
    /usr/share/foreman/bin/rails server -e production
```

FOREMAN CONTAINER

```
1  FROM quay.io/centos/centos:stream9
2
3  RUN dnf install -y foreman foreman-postgresql\
4      foreman-service foreman-redis foreman-dynflow-sidekiq
5
6  ARG FOREMAN_PLUGINS="foreman-tasks\
7                       foreman_remote_execution\
8                       katello"
9
10 RUN for PLUGIN in ${FOREMAN_PLUGINS}; do … ; done
11
12 CMD /usr/share/foreman/bin/rails db:migrate &&\
13     /usr/share/foreman/bin/rails db:seed &&\
14     /usr/share/foreman/bin/rails server -e production
```

FOREMAN CONTAINER

```
 1 FROM quay.io/centos/centos:stream9
 2
 3 RUN dnf install -y foreman foreman-postgresql\
 4     foreman-service foreman-redis foreman-dynflow-sidekiq
 5
 6 ARG FOREMAN_PLUGINS="foreman-tasks\
 7                       foreman_remote_execution\
 8                       katello"
 9
10 RUN for PLUGIN in ${FOREMAN_PLUGINS}; do … ; done
11
12 CMD /usr/share/foreman/bin/rails db:migrate &&\
13     /usr/share/foreman/bin/rails db:seed &&\
14     /usr/share/foreman/bin/rails server -e production
```

FOREMAN CONTAINER

```
1  FROM quay.io/centos/centos:stream9
2
3  RUN dnf install -y foreman foreman-postgresql\
4      foreman-service foreman-redis foreman-dynflow-sidekiq
5
6  ARG FOREMAN_PLUGINS="foreman-tasks\
7                       foreman_remote_execution\
8                       katello"
9
10 RUN for PLUGIN in ${FOREMAN_PLUGINS}; do … ; done
11
12 CMD /usr/share/foreman/bin/rails db:migrate &&\
13     /usr/share/foreman/bin/rails db:seed &&\
14     /usr/share/foreman/bin/rails server -e production
```

FOREMAN CONTAINER

```
# cat /etc/containers/systemd/foreman.container
[Container]
ContainerName=foreman
HostName=quadlet.example.com
Image=quay.io/evgeni/foreman-rpm:nightly
Network=host
Secret=foreman-database-url,type=env,target=DATABASE_URL
Secret=foreman-settings-yaml,type=mount,
       target=/etc/foreman/settings.yaml
…
Secret=foreman-client-cert,type=mount,
       target=/etc/foreman/client_cert.pem
…
```

FOREMAN CONTAINER

```
# cat /etc/containers/systemd/foreman.container
[Container]
ContainerName=foreman
HostName=quadlet.example.com
Image=quay.io/evgeni/foreman-rpm:nightly
Network=host
Secret=foreman-database-url,type=env,target=DATABASE_URL
Secret=foreman-settings-yaml,type=mount,
       target=/etc/foreman/settings.yaml
…
Secret=foreman-client-cert,type=mount,
       target=/etc/foreman/client_cert.pem
…
```

DYNFLOW CONTAINERS

```
# cat /etc/containers/systemd/dynflow-sidekiq@.container
[Container]
ContainerName=dynflow-sidekiq-%i
HostName=quadlet.example.com
Image=quay.io/evgeni/foreman-rpm:nightly
Network=host
Secret=…
Exec=/usr/libexec/foreman/sidekiq-selinux -e production \
     -r /usr/share/foreman/extras/dynflow-sidekiq.rb \
     -C /etc/foreman/dynflow/%i.yml
```

DYNFLOW CONTAINERS

```
# cat /etc/containers/systemd/dynflow-sidekiq@.container
[Container]
ContainerName=dynflow-sidekiq-%i
HostName=quadlet.example.com
Image=quay.io/evgeni/foreman-rpm:nightly
Network=host
Secret=…
Exec=/usr/libexec/foreman/sidekiq-selinux -e production \
     -r /usr/share/foreman/extras/dynflow-sidekiq.rb \
     -C /etc/foreman/dynflow/%i.yml
```

DYNFLOW CONTAINERS

```
# cat /etc/containers/systemd/dynflow-sidekiq@.container
[Container]
ContainerName=dynflow-sidekiq-%i
HostName=quadlet.example.com
Image=quay.io/evgeni/foreman-rpm:nightly
Network=host
Secret=…
Exec=/usr/libexec/foreman/sidekiq-selinux -e production \
     -r /usr/share/foreman/extras/dynflow-sidekiq.rb \
     -C /etc/foreman/dynflow/%i.yml
```

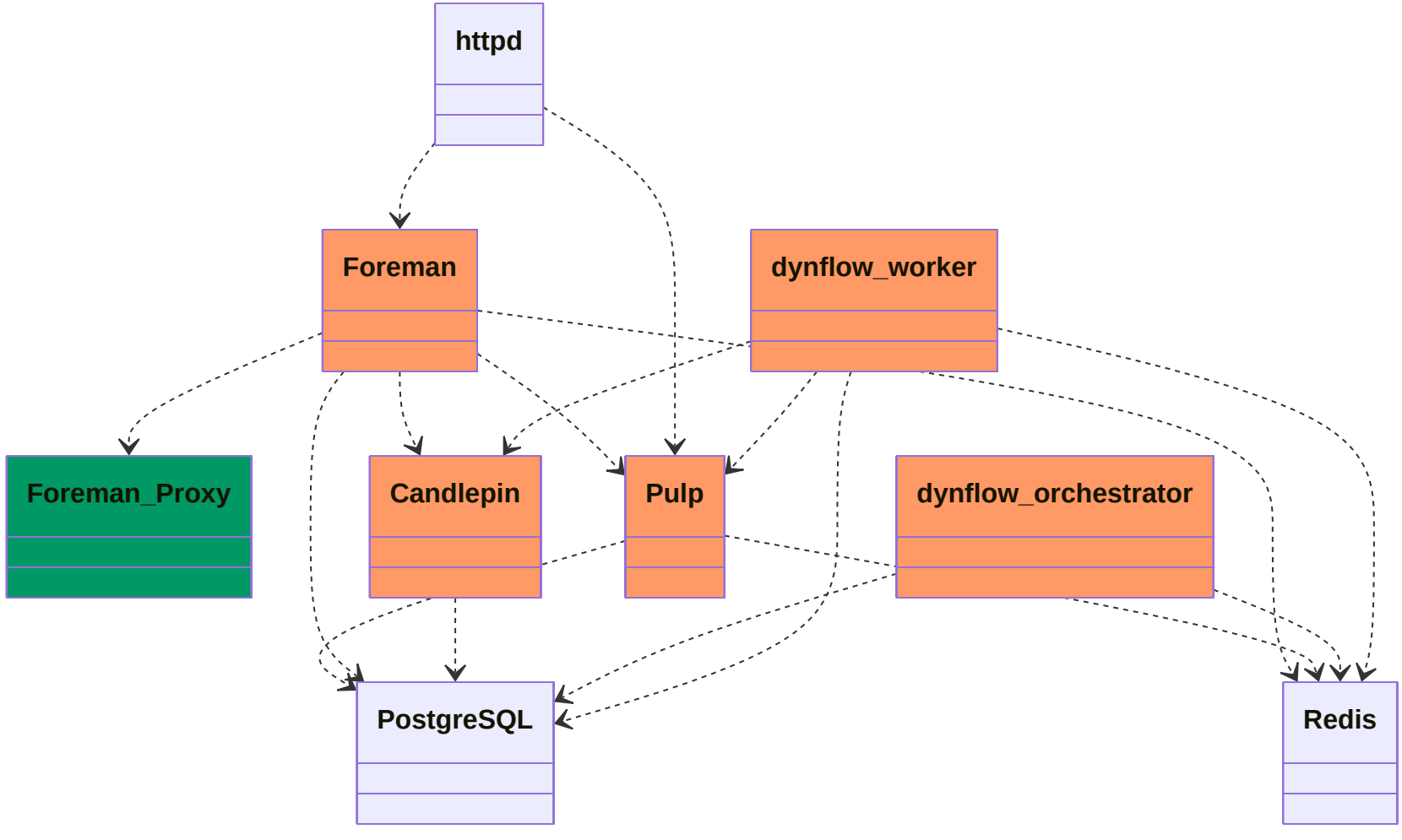
DYNFLOW CONTAINERS

```
# cat /etc/containers/systemd/dynflow-sidekiq@.container
[Container]
ContainerName=dynflow-sidekiq-%i
HostName=quadlet.example.com
Image=quay.io/evgeni/foreman-rpm:nightly
Network=host
Secret=…
Exec=/usr/libexec/foreman/sidekiq-selinux -e production \
     -r /usr/share/foreman/extras/dynflow-sidekiq.rb \
     -C /etc/foreman/dynflow/%i.yml
```

PULP CONTAINERS

- Built by the Pulp Project!
- All-in-One (incl PostgreSQL and Redis)
- API / Content / Worker

WHICH CONTAINERS?



ARE WE THERE YET?

ARE WE THERE YET?

No.

ARE WE THERE YET?

No. But we know which questions we need to answer.

INSTALLATION / CONFIGURATION

- our research used Ansible
- our current Installer is Puppet
- is this a chance to simplify?

MIGRATIONS / UPGRADES

- existing setups will require migration
- the installer decision will influence this

LOGGING

- Applications log to `/var/log`, but we don't mount that
- stdout/stderr is correctly collected by systemd

POSTGRESQL

- Should we move PostgreSQL into a container?
- How will we handle PostgreSQL upgrades?

HTTP INGRESS

- Today we use Apache httpd from the host
- Inherits crypto policies from the host
- Inherits Kerberos setup from the host

PACKAGING

- we're using RPM packages as the base for the containers
- the containers can run on Debian/Ubuntu
- containers *can* be build from source

INTEGRATION

- Foreman uses the Foreman Proxy to integrate with services
- We avoided that part in our research
- We'll also probably avoid it for the first real deployment

DEVELOPMENT SETUP

- "quickly apply this patch" doesn't work anymore
- neither does `bundle exec rails`
- we need to get dev and prod deployments closer together

YOU

- Pretty sure we didn't find all the questions
- There is an RFC
- We're also eager to hear ideas!

LINKS

- podman-systemd
- foreman-quadlet
- RFC: Foreman Production Installation via Containers and Podman Quadlets

THANKS!

 evgeni@golov.de

 die-welt.net

 @zhenech@chaos.social

 @evgeni