



## Ceph, Docker, Heroku Slugs, CoreOS and Deis Overview



lorieri Nov/2014 @againstty0



$\alpha$

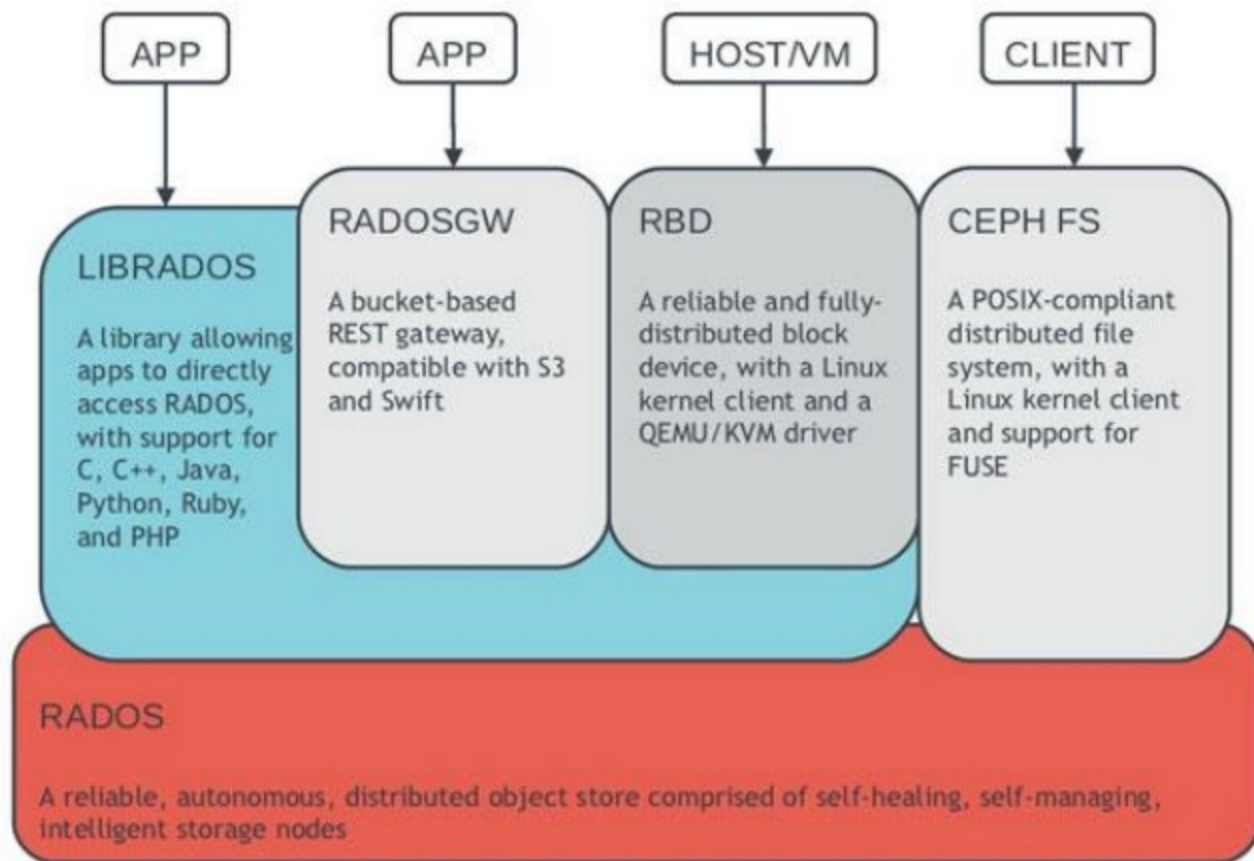
Welcome  
to  
the  
future

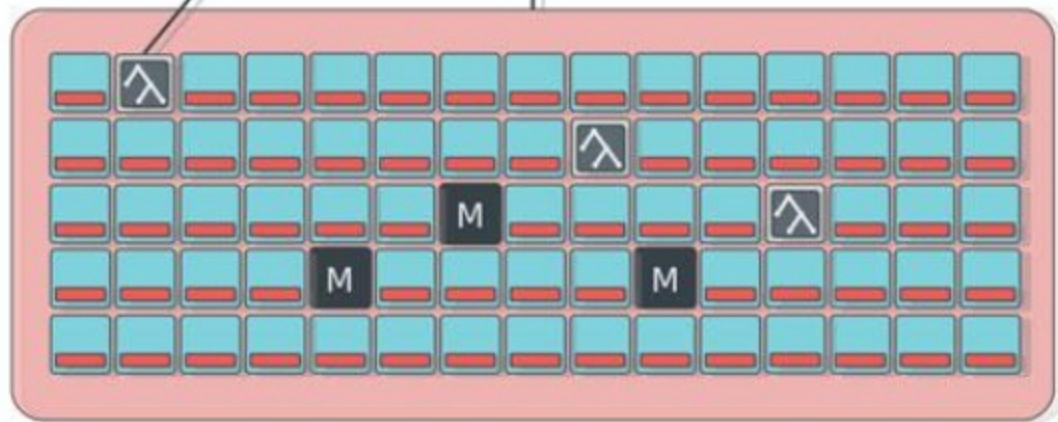
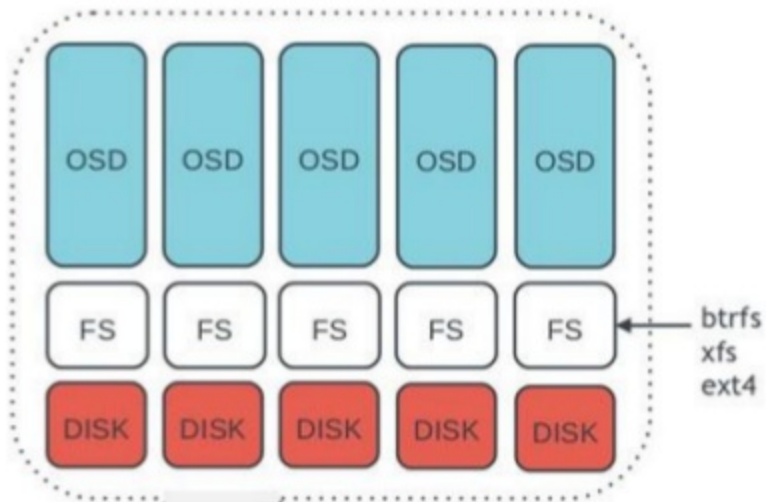
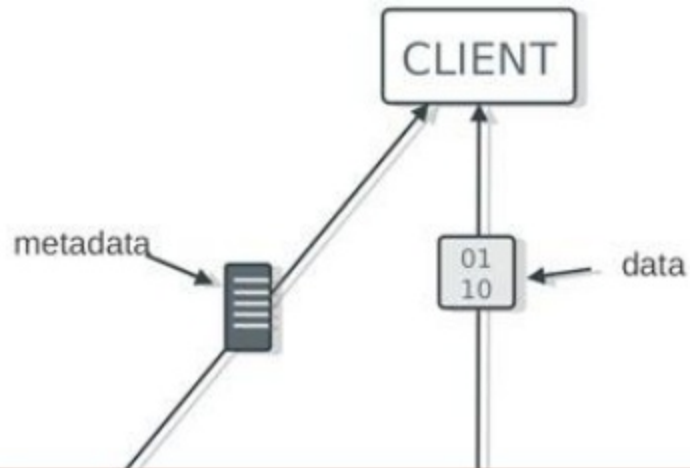


*“Ceph's main goals are to be completely distributed without a single point of failure, scalable to the exabyte level, and freely-available”*



- 100% distributed
- CephFS
  - For POSIX sharing
  - not really 100% (active-standby)
- LibRados
  - RBD
    - For Blocks
  - Rados Gateway
    - For REST Objects
    - S3 and Swift compatible





CephFS





**“Build, Ship and Run  
Any App, Anywhere”**





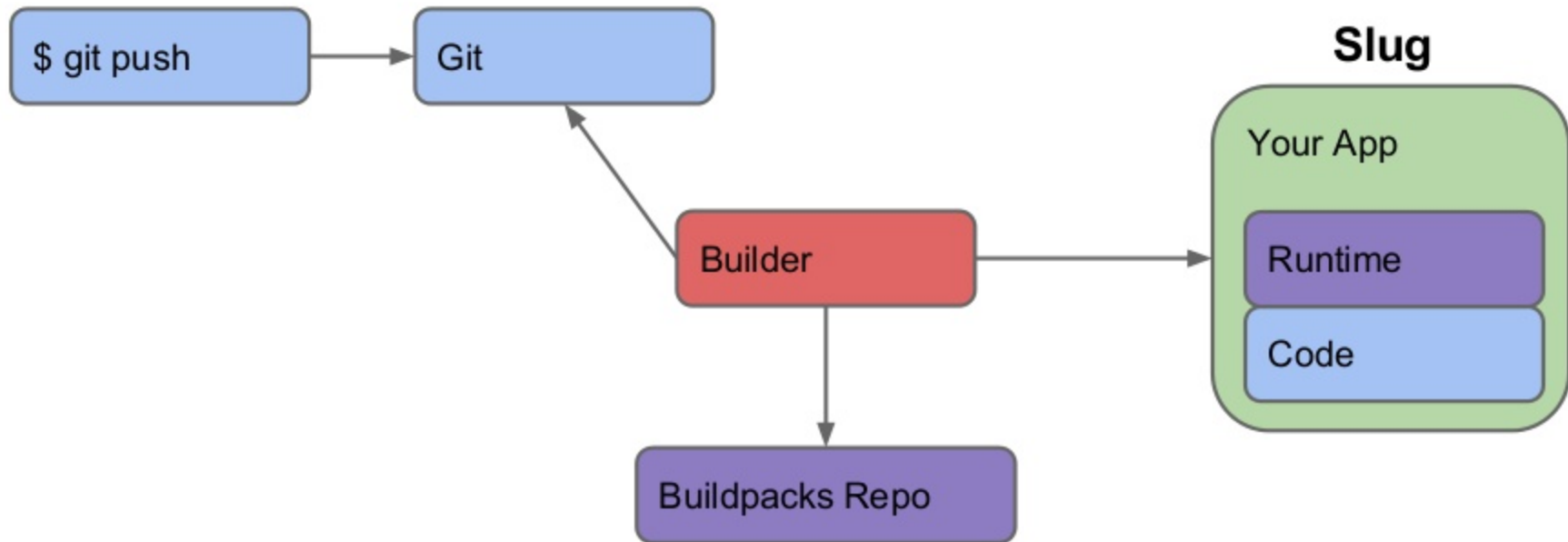
- **100% portable**
- **Easy API for LXC**
- **Public Repository**
- **Lightweight, 1 process (ideal)**
- **Layers (Union File System)**
  - **Shares read-only data**
  - **Incremental**
- **Volumes (not layered volumes)**





- **We will only talk about slugs**
  - (and twelve-factor later)

***“A slug is a bundle of your source, fetched dependencies, the language runtime, and compiled/generated output of the build system - ready for execution.”***





**CoreOS**

***“Linux for Massive Server Deployments***

***CoreOS enables warehouse-scale  
computing on top of a minimal, modern  
operating system.”***



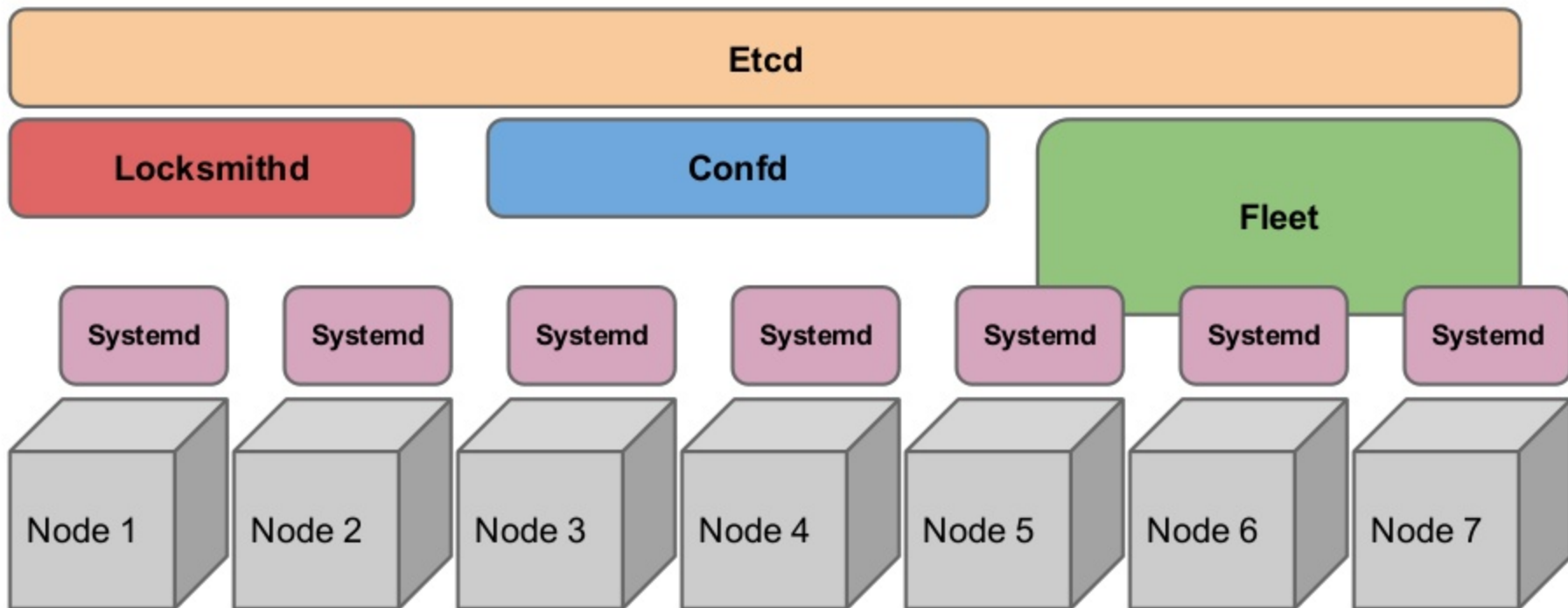
# Core OS

- 100% distributed
- Lightweight
- cloud-init for every boot
- Automatic Updates
  - 2 boot partitions



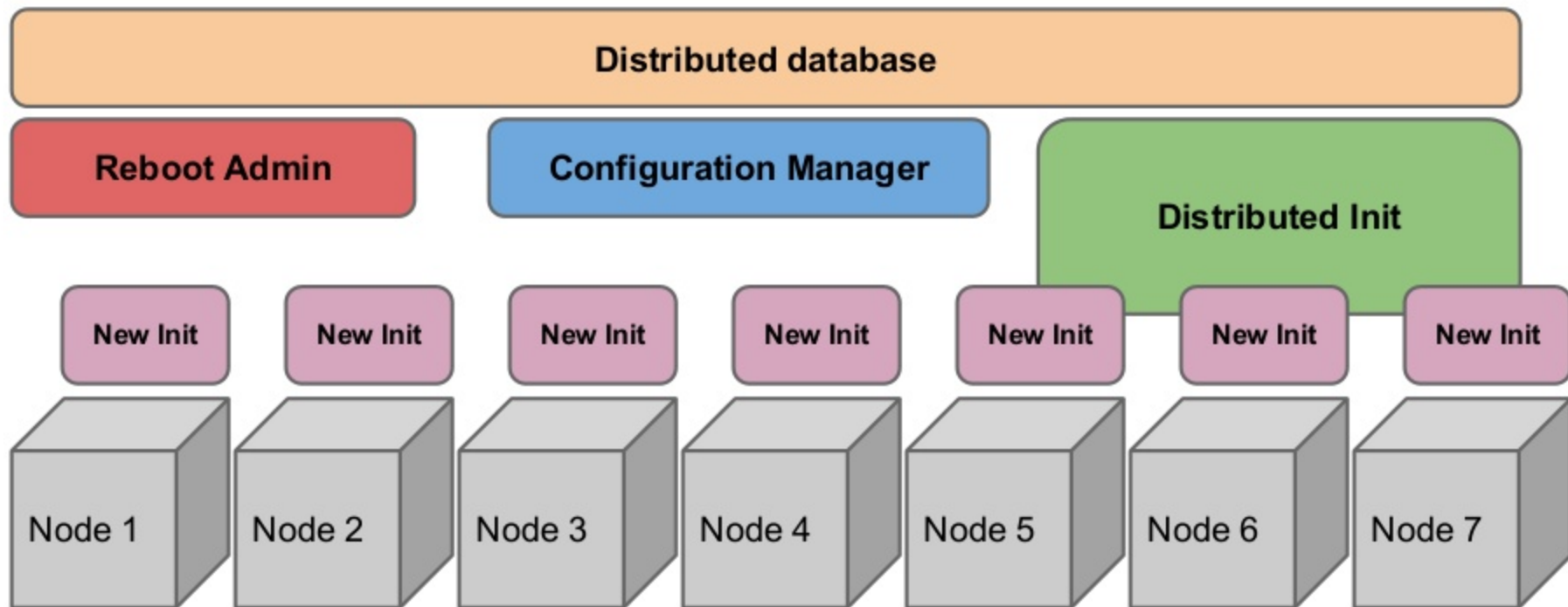


# CoreOS





# CoreOS

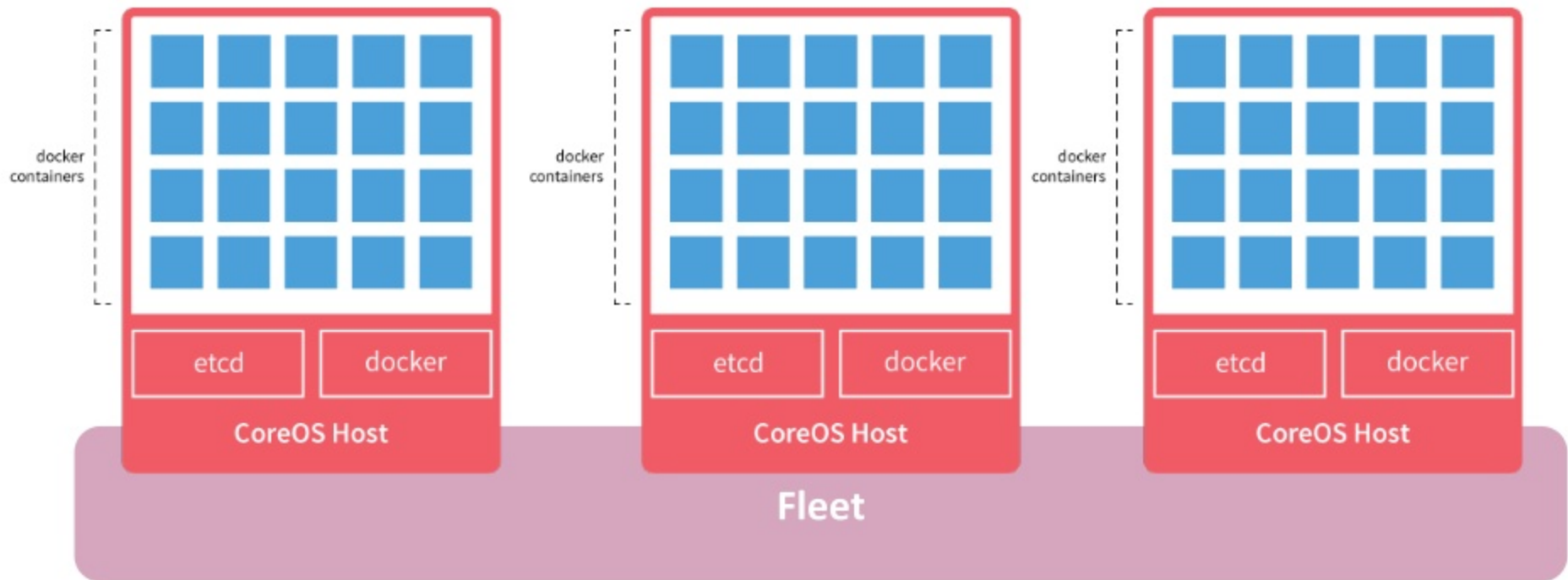




# CoreOS



# docker







- Fleet

## Distributed Services

```
core@coreos1 ~ $ fleetctl list-units
```

deis-builder.service	1b8e82ef.../10.1.1.4	active	running
deis-cache.service	1b8e82ef.../10.1.1.3	active	running
deis-controller.service	1b8e82ef.../10.1.1.3	active	running
deis-database.service	1b8e82ef.../10.1.1.4	active	running
deis-logger.service	1b8e82ef.../10.1.1.3	active	running
deis-logspout.service	1b8e82ef.../10.1.1.2	active	running
deis-logspout.service	1b8e82ef.../10.1.1.3	active	running
deis-logspout.service	1b8e82ef.../10.1.1.4	active	running
deis-publisher.service	1b8e82ef.../10.1.1.2	active	running
deis-publisher.service	1b8e82ef.../10.1.1.3	active	running
deis-publisher.service	1b8e82ef.../10.1.1.4	active	running
deis-registry.service	1b8e82ef.../10.1.1.4	active	running
deis-router@1.service	1b8e82ef.../10.1.1.3	active	running
deis-router@2.service	1b8e82ef.../10.1.1.4	active	running
deis-router@3.service	1b8e82ef.../10.1.1.2	active	running
deis-store-daemon.service	1b8e82ef.../10.1.1.2	active	running
deis-store-daemon.service	1b8e82ef.../10.1.1.3	active	running
deis-store-daemon.service	1b8e82ef.../10.1.1.4	active	running
deis-store-gateway.service	1b8e82ef.../10.1.1.3	active	running
deis-store-metadata.service	1b8e82ef.../10.1.1.2	active	running
deis-store-metadata.service	1b8e82ef.../10.1.1.3	active	running
deis-store-metadata.service	1b8e82ef.../10.1.1.4	active	running
deis-store-monitor.service	1b8e82ef.../10.1.1.2	active	running
deis-store-monitor.service	1b8e82ef.../10.1.1.3	active	running
deis-store-monitor.service	1b8e82ef.../10.1.1.4	active	running
deis-store-volume.service	1b8e82ef.../10.1.1.2	active	running
deis-store-volume.service	1b8e82ef.../10.1.1.3	active	running
deis-store-volume.service	1b8e82ef.../10.1.1.4	active	running

```
core@coreos1 ~ $
```



***“Your PaaS. Your Rules.”***

***An open source PaaS that makes it easy to  
deploy and manage applications on your  
own servers. Deis builds upon Docker and  
CoreOS to provide a lightweight PaaS  
with a Heroku-inspired workflow.”***



Developer



Build Images



Scheduler

Containers

Router

Cluster (Dev)

Scheduler

Containers

Router

Cluster (Test)

Scheduler

Containers

Router

Cluster (Prod)

Pull Images

Load Balancer



Application Consumers

Route Traffic

Monitoring

Logging

Backing Services



# DEIS

- **CoresOS + Docker + Ceph + Heroku**
- **Twelve-Factor**
  - for Deis: must be stateless (no wordpress)
- **Nginx Router + Wildcard DNS**
- **First release using Ceph**
  - more features coming soon
- **Limited**
  - twelve-factor
  - only HTTP port (non-http soon)
  - must expose **ONLY** one port



- **Installation:**
  - **Install CoreOS and ssh keys then:**

```
$ export DEISCTL_TUNNEL=coreos01
```

```
$ curl -sSL http://deis.io/deisctl/install.sh | sh
```

```
$ git clone https://github.com/deis/deis.git ; cd deis
```

```
$ deisctl config platform set \
```

```
    domain=mylocalpaas.com
```

```
$ deisctl install platform && deisctl start platform
```