CoreOS Paris UG Meetup #5

CoreOS Linux - 101
OS insides + cluster bootstrapping

Agenda

1 - What is CoreOS ?

2 - Clustered by design ?

3 - The update process

4 - Demos

- Cluster bootstrapping
- Update process management

5 - Docs & Opening

1 - What is CoreOS ?

Core OS

1 - What is CoreOS ?

A ***lightweight*** *Linux operating system* ***designed for clustered deployments*** *providing:*

- *Automation,*
- *Security,*
- *and scalability*

for your most critical applications.

https://coreos.com/why/

1 - What is CoreOS ?

- *“Lightweight”:*
- *“**Clustered**”:*
- *"Automation", "security" and "scalability"*

- OK
- How ? What does it mean ?
- This sounds like a dream,

  Tell me more !

1 - What is CoreOS ?

It is not “**just**” an OS

Open Source Projects for Linux Containers

1 - What is CoreOS ?

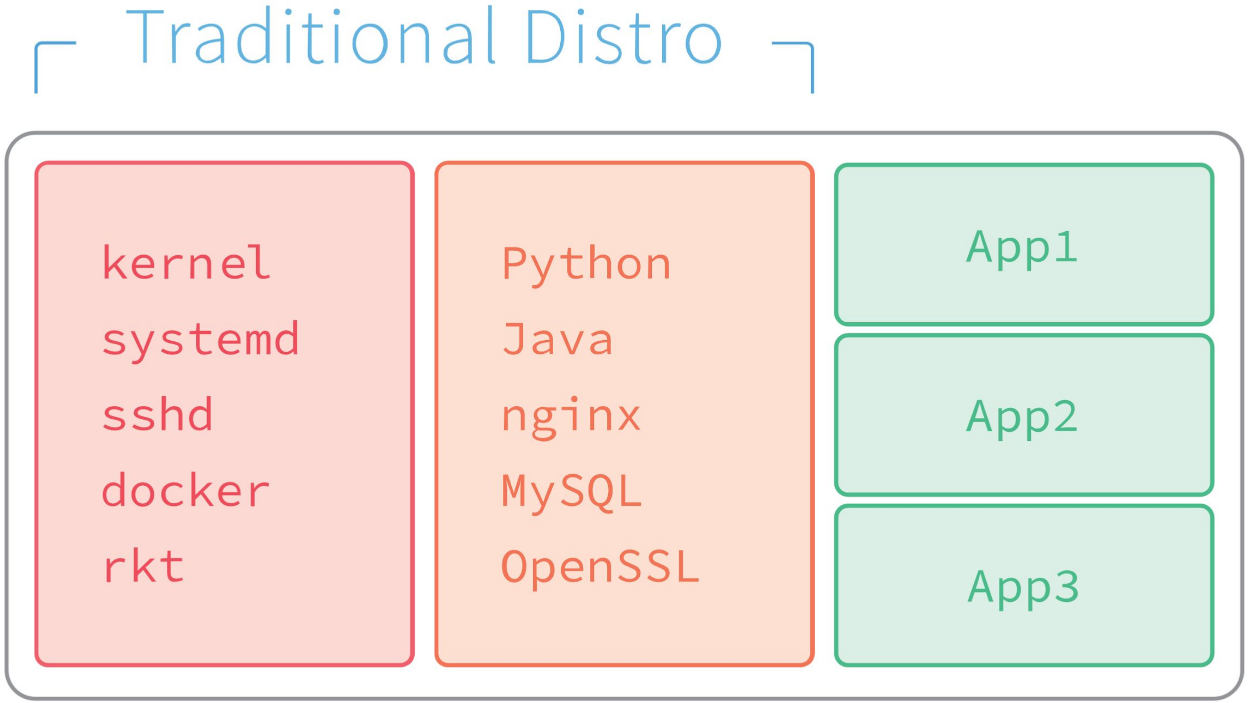
It is not “just” an OS

Projects: https://github.com/coreos/

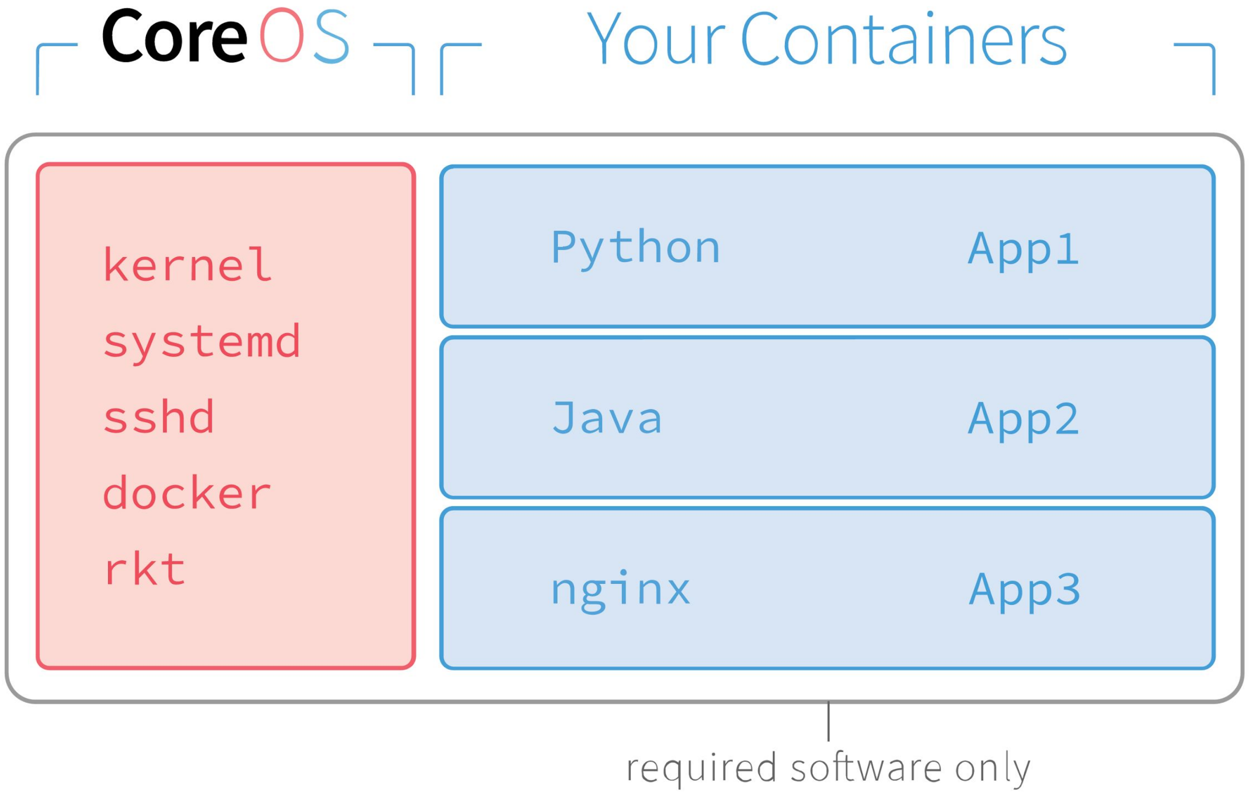
- CoreOS Linux
- Ectd
- Fleet
- Update-engine + Locksmith
- Flannel
- Rkt
- ...

1 - What is CoreOS ?

1.1 - CoreOS Linux: Redefining the Linux Distro



1.1 - CoreOS Linux: Redefining the Linux Distro



1.1 - CoreOS Linux: Redefining the Linux Distro

Features:

- **Read-only** RootFS + Overlay /etc
- **No package manager**
- **Simple** toolbelt utils, the “Unix way”

=> Separation of concerns: OS vs Apps

1 - What is CoreOS ?

1.2 - The CoreOS Update Philosophy

---

https://coreos.com/why/#updates

*CoreOS **automates software updates** to ensure better security and reliability of machines and containers running in large-scale clusters.*

- What does it mean ?
- What is the difference with croned :

`` `apt update && apt upgrade` `` ?

1.2 - The CoreOS Update Philosophy

---

https://coreos.com/why/#updates

Operating system updates and security patches are regularly ***pushed*** *to CoreOS Linux machines* ***without requiring intervention by administrators****.*

What !?

1.2 - The CoreOS Update Philosophy

https://coreos.com/why/#updates

The ***isolation*** *of all* ***application code and dependencies*** *in* ***containers*** *means these* ***frequent OS updates*** *can deliver the latest features and security fixes* ***without risk to the apps running above.***

1.2 - The CoreOS Update Philosophy

https://coreos.com/why/#updates

The ***isolation*** *of all* ***application code and dependencies*** *in* ***containers*** *means these* ***frequent OS updates*** *can deliver the latest features and security fixes* ***without risk to the apps running above.***

The ***decoupling*** *of the application from the system and library dependencies layer* ***is the force*** *driving containers in the enterprise.*

1.2 - The CoreOS Update Philosophy

https://coreos.com/why/#updates

*The **isolation** of all **application code and dependencies** in **containers** means these **frequent OS updates** can deliver the latest features and security fixes **without risk to the apps running above.***

*The **decoupling** of the application from the system and library dependencies layer **is the force** driving containers in the enterprise.*

*CoreOS applies these lessons to the container support layer, the operating system, minimizing it and **formalizing the semantics of updates**.*

1.2 - The CoreOS Update Philosophy

https://coreos.com/why/#updates

*The **isolation** of all **application code and dependencies** in **containers** means these **frequent OS updates** can deliver the latest features and security fixes **without risk to the apps running above.***

*The **decoupling** of the application from the system and library dependencies layer **is the force** driving containers in the enterprise.*

*CoreOS applies these lessons to the container support layer, the operating system, minimizing it and **formalizing the semantics of updates**.*

=> The key concept: **Separate concerns between the OS vs App !**

1.2 - The CoreOS Update Philosophy

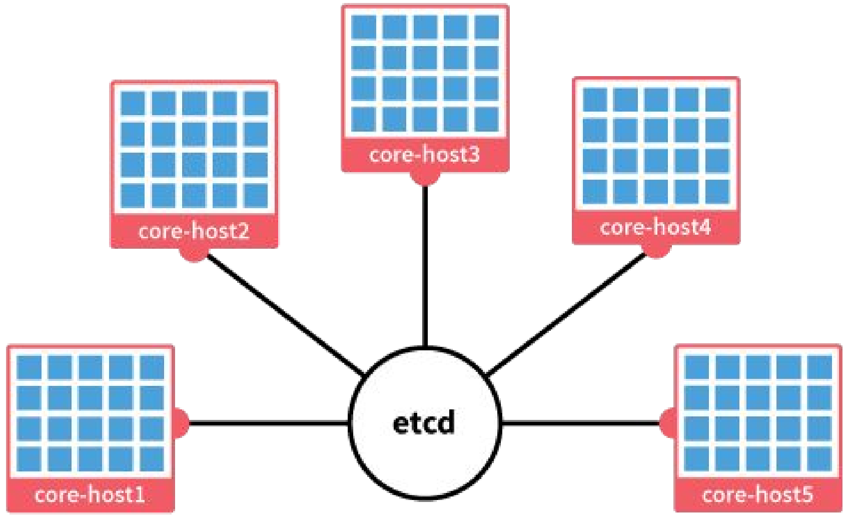
https://coreos.com/why/#updates

*Because, **even today** with VMs, **workflow ties the OS directly to the apps** on the box.*

***Moving dependencies out of the OS and into a container** dramatically **reduces complexity** for systems administrators.*

2 - Clustered by design ?

https://coreos.com/os/docs/latest/cluster-discovery.html



2 - Clustered by design ?

---

https://coreos.com/os/docs/latest/cluster-discovery.html

CoreOS uses ***etcd****, a service running* ***on each machine****, to handle coordination between software running on the cluster.*

For a group of CoreOS machines ***to form a cluster****, their* ***etcd instances need to be connected****.*

2 - Clustered by design ?

---

https://coreos.com/os/docs/latest/cluster-discovery.html

*CoreOS uses **etcd**, a service running **on each machine**, to handle coordination between software running on the cluster.*

*For a group of CoreOS machines **to form a cluster**, their **etcd instances need to be connected.***

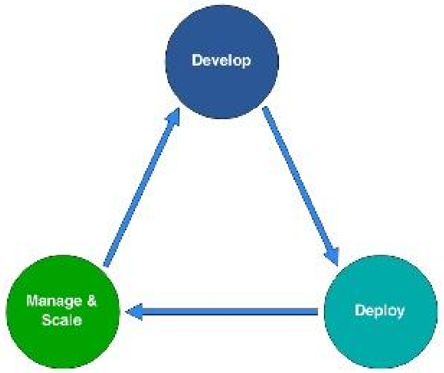
=> Defined, setup and configured **at provisioning** (By Ignition / cloud-config)

3 - Update process

https://coreos.com/os/docs/latest/update-strategies.html

omaha-server

High Fidelity, High Velocity Deployments in the Cloud



3 - Update process

Chrome inspired Update-engine (Omaha protocol)

- https://github.com/**google/omaha**
- http://www.slideshare.net/DmitriyLyfar/**omaha-google-update-server**

Features:

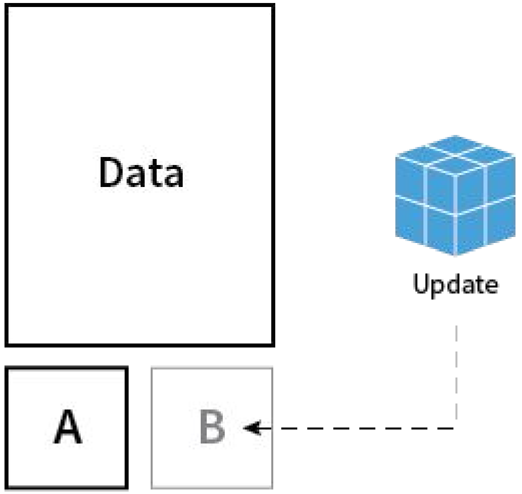
- Atomic
- Coordinated
- Automatic rollback

3 - Update process

Update-engine: https://github.com/**coreos/update_engine**

Features:

- Atomic
- Coordinated
- Automatic rollback



1. The **runtime** and mounted partition is **A**

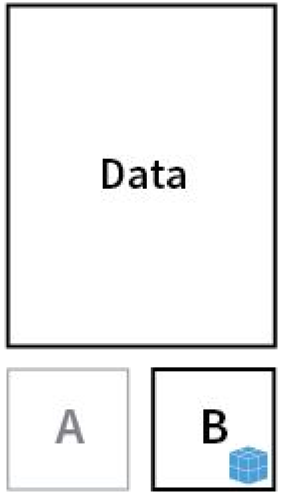
2. The update image is **flushed** on partition B

3 - Update process

Update-engine: [https://github.com/**coreos/update_engine**](https://github.com/coreos/update_engine)

Features:

- **Atomic**
- Coordinated
- Automatic rollback



3. The **partition table** is updated
    a. Flags *"Tries=1"*, *"Successful=0"* on partition B
    b. "prioritize" the boot order for B
4. The server **reboots**

3 - Update process

Update-engine: https://github.com/**coreos/update_engine**

Features:

- Atomic
- Coordinated
- Automatic rollback

1. **GRUB** select the **1st partition (new)**
2. If fails, GRUB select the 2nd partition (old)
3. When the boot succeed, **check the flags** and mark the “good Root” as “successfully” booted

https://github.com/coreos/update_engine/blob/master/coreos-setgoodroot

=> Send feedback to the Update server

3 - Update process

Update-engine: https://github.com/**coreos/update_engine**

https://github.com/coreos/**locksmith**

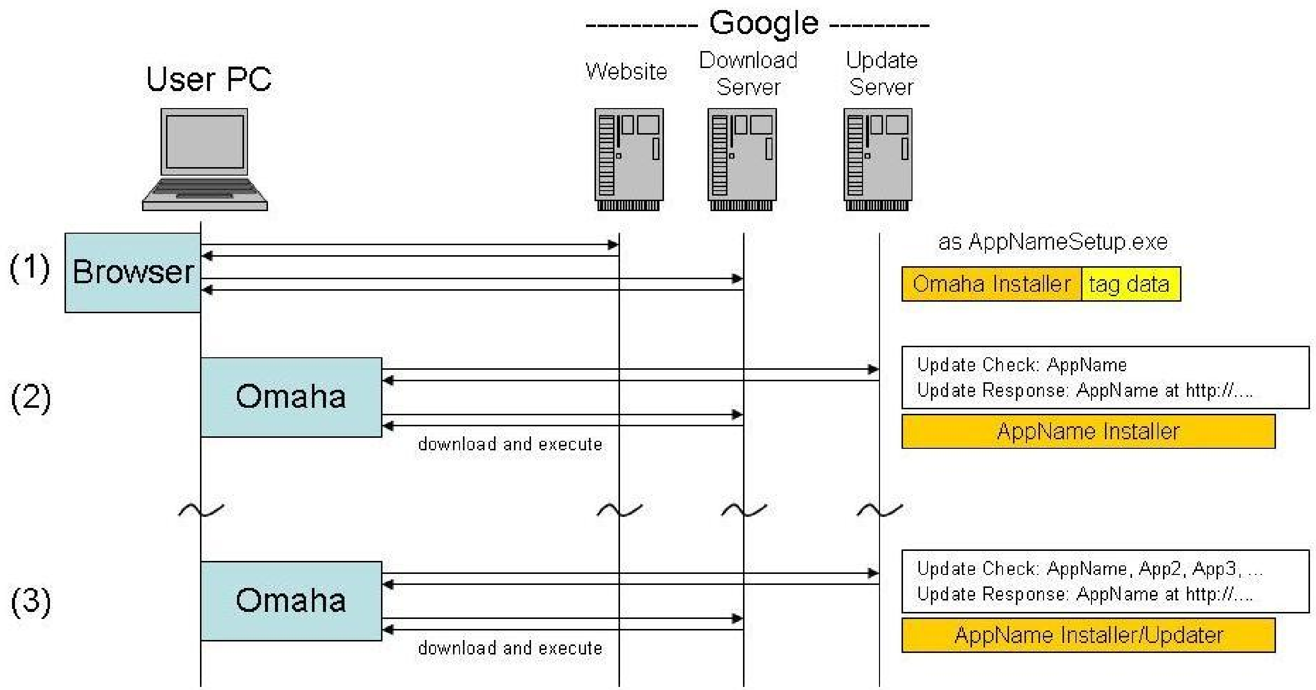
Features:

- Atomic
- **Coordinated**
- Automatic rollback

The update-strategies

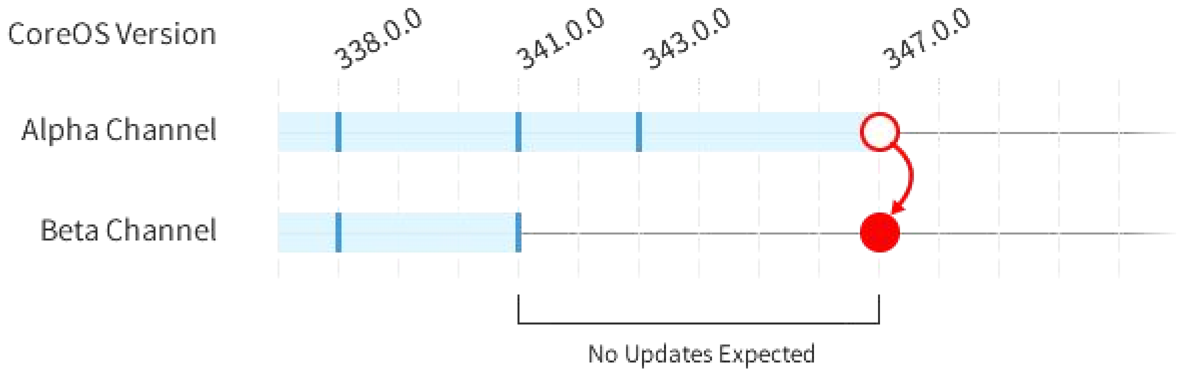
- Etcd-lock
- Reboot
- Best-effort
- off

3 - Update process



3 - Update process

Release Channels : https://coreos.com/os/docs/latest/update-strategies.html



3 - Update process

Summary

```
$ cat /etc/coreos/update.conf


GROUP=stable

REBOOT_STRATEGY=etcd-lock
```

```
$ cat /etc/os-release


NAME=CoreOS

ID=coreos

VERSION=835.9.0

VERSION_ID=835.9.0

BUILD_ID=

PRETTY_NAME="CoreOS 835.9.0"
```

3 - Update process

Summary

```
$ cat /etc/coreos/update.conf


GROUP=stable

REBOOT_STRATEGY=etcd-lock
```

```
$ cat /etc/lsb-release


DISTRIB_ID=CoreOS

DISTRIB_RELEASE=835.9.0

DISTRIB_CODENAME="Red Dog"

DISTRIB_DESCRIPTION="CoreOS

835.9.0"
```

4 - DEMOs

4.1 - Cluster bootstrapping

4.2 - Update process management

4 - Demos

4.1 - Cluster bootstrapping

4.2 - Update process management

4.1 - Cluster bootstrapping

OK, great, now I want to play with CoreOS :D

How do I bootstrap an entire cluster ?

Running CoreOS

CoreOS runs on most cloud providers, virtualization platforms and bare metal servers. Running a local VM on your laptop is a great dev environment. Following the **Quick Start guide** is the fastest way to get set up.

Official Platforms

Cloud Providers

- Amazon EC2 Container Service
- Amazon EC2
- DigitalOcean
- Microsoft Azure
- Google Compute Engine
- Rackspace Cloud

Bare Metal

- Booting with iPXE
- Booting with PXE
- Installing to Disk

Other Platforms

- OpenStack
- Vagrant
- ISO

Community-Supported Platforms

Cloud Providers

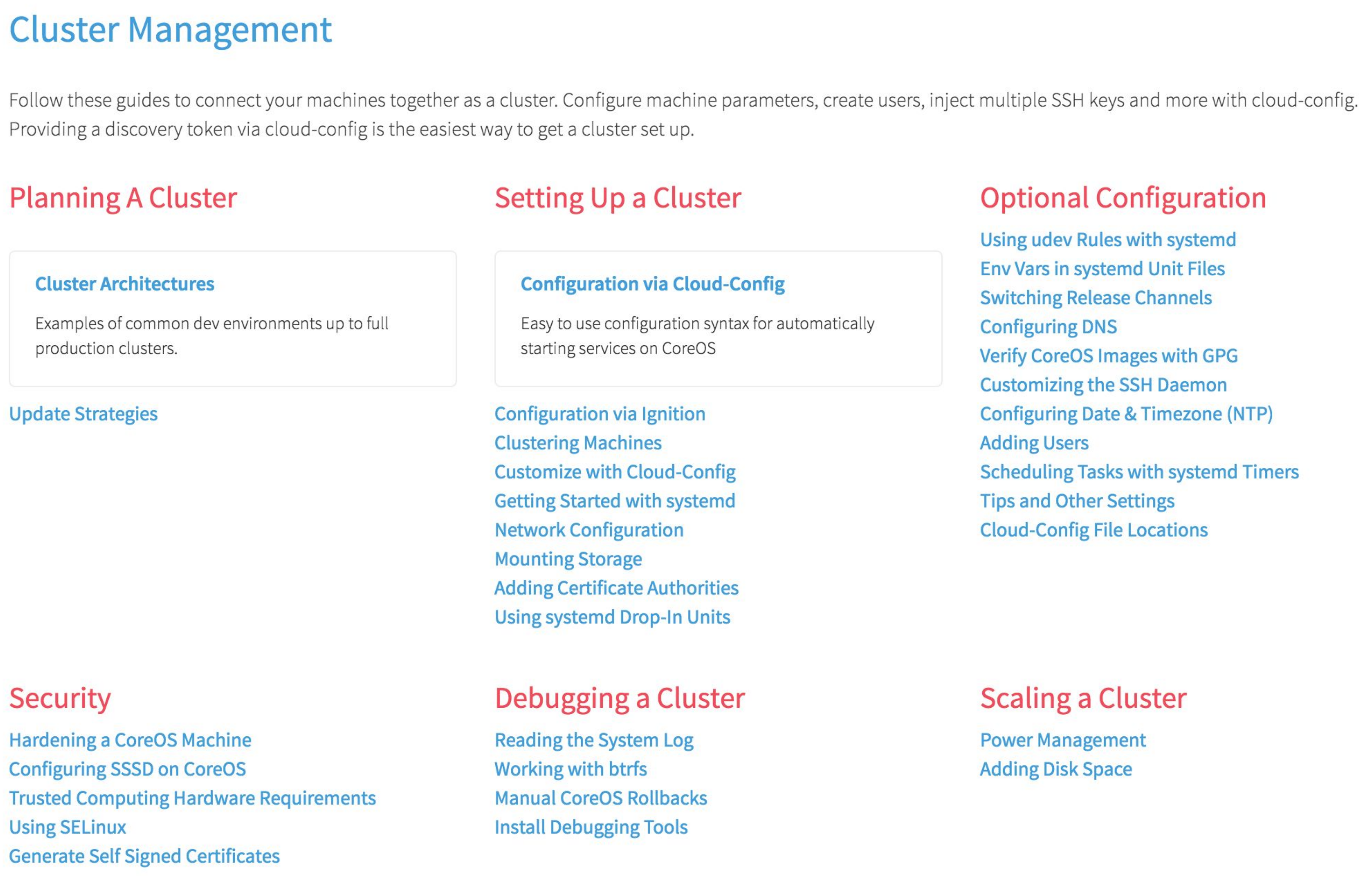
- Packet
- AURO Cloud
- VEXXHOST Cloud
- NIFTY Cloud
- Cloud.ca
- Exoscale
- Interoute VDC
- RimuHosting LaunchtimeVPS
- Vultr VPS
- Brightbox Cloud

Other Platforms

- CloudStack
- Eucalyptus
- VMware
- Libvirt
- QEMU
- VirtualBox

Uncategorized Documents

- nginx host cloud config
- debian interfaces
- booting on ikoula



Cluster Management

Follow these guides to connect your machines together as a cluster. Configure machine parameters, create users, inject multiple SSH keys and more with cloud-config. Providing a discovery token via cloud-config is the easiest way to get a cluster set up.

Planning A Cluster

Cluster Architectures

Examples of common dev environments up to full production clusters.

Update Strategies

Setting Up a Cluster

Configuration via Cloud-Config

Easy to use configuration syntax for automatically starting services on CoreOS

Configuration via Ignition
Clustering Machines
Customize with Cloud-Config
Getting Started with systemd
Network Configuration
Mounting Storage
Adding Certificate Authorities
Using systemd Drop-In Units

Optional Configuration

Using udev Rules with systemd
Env Vars in systemd Unit Files
Switching Release Channels
Configuring DNS
Verify CoreOS Images with GPG
Customizing the SSH Daemon
Configuring Date & Timezone (NTP)
Adding Users
Scheduling Tasks with systemd Timers
Tips and Other Settings
Cloud-Config File Locations

Security

Hardening a CoreOS Machine
Configuring SSSD on CoreOS
Trusted Computing Hardware Requirements
Using SELinux
Generate Self Signed Certificates

Debugging a Cluster

Reading the System Log
Working with btrfs
Manual CoreOS Rollbacks
Install Debugging Tools

Scaling a Cluster

Power Management
Adding Disk Space

4.1 - Cluster bootstrapping

CoreOS official **configuration tools**:

- [https://coreos.com/os/docs/latest/**cloud-config**.html](https://coreos.com/os/docs/latest/cloud-config.html)
- [https://coreos.com/**ignition**/docs/latest/](https://coreos.com/ignition/docs/latest/)

4.1 - Cluster bootstrapping

CoreOS official **configuration tools**:

- `https://coreos.com/os/docs/latest/`**`cloud-config`**`.html`
- `https://coreos.com/`**`ignition`**`/docs/latest/`

CoreOS-baremetal:

- `https://github.com/coreos/`**`coreos-baremetal`**`/blob/master/Documentation/`**`bootcfg`**`.md`
  - HTTP and gRPC service
  - Renders signed Ignition configs, cloud-configs
  - Renders **network boot configs** (PXE / iPXE / Grub), and metadata to machines
  - Groups match machines based on label
  - Based on **Templates**

4.1 - Cluster bootstrapping

Wow !

That looks like a lot of documentation to read

4.1 - Cluster bootstrapping

Introducing Mayu !

https://blog.giantswarm.io/**mayu-yochu-provisioning-tools-for-coreos-bare-metal**/

4.1 - Cluster bootstrapping

Mayu Features:

https://github.com/**giantswarm/mayu/**

- All in 1 - **bundled services** (`DHCP / PXE / TFTP / HTTP`)
- **`HTTP` service** for `iPXE / Cloud-config / Ignition` config files (based on templates)
- Manage clusters as “named” groups of nodes
- **`Git` backed-up history** (Track node state transition, versioning and auditing)
- `mayuctl` **client** for operations (Track and edit nodes)
- Metadata and Tags support

4.1 - Cluster bootstrapping

Mayu doc:

https://blog.giantswarm.io/**mayu-yochu-provisioning-tools-for-coreos-bare-metal/**

- https://github.com/**giantswarm/mayu/**
- https://github.com/giantswarm/mayu/blob/master/docs/**inside**.md
- https://github.com/giantswarm/mayu/blob/master/docs/**machine_state_transition**.md
- https://github.com/giantswarm/mayu/blob/master/docs/**mayuctl**.md

4.1 - Cluster bootstrapping

Demo !

4 - Demos

4.1 - Cluster bootstrapping

4.2 - Update process management

4.2 - Update process management

OK, now I have 1 cluster of CoreOS nodes running.

What about the updates?

4.2 - Update process management

CoreOS pushes the updates…

But,

- How can I control **which update** is pushed to my clusters ?
- How can I control **when** the updates are applied ?
- How can I **see and visualize** the state of the nodes ?

4.1 - Cluster bootstrapping

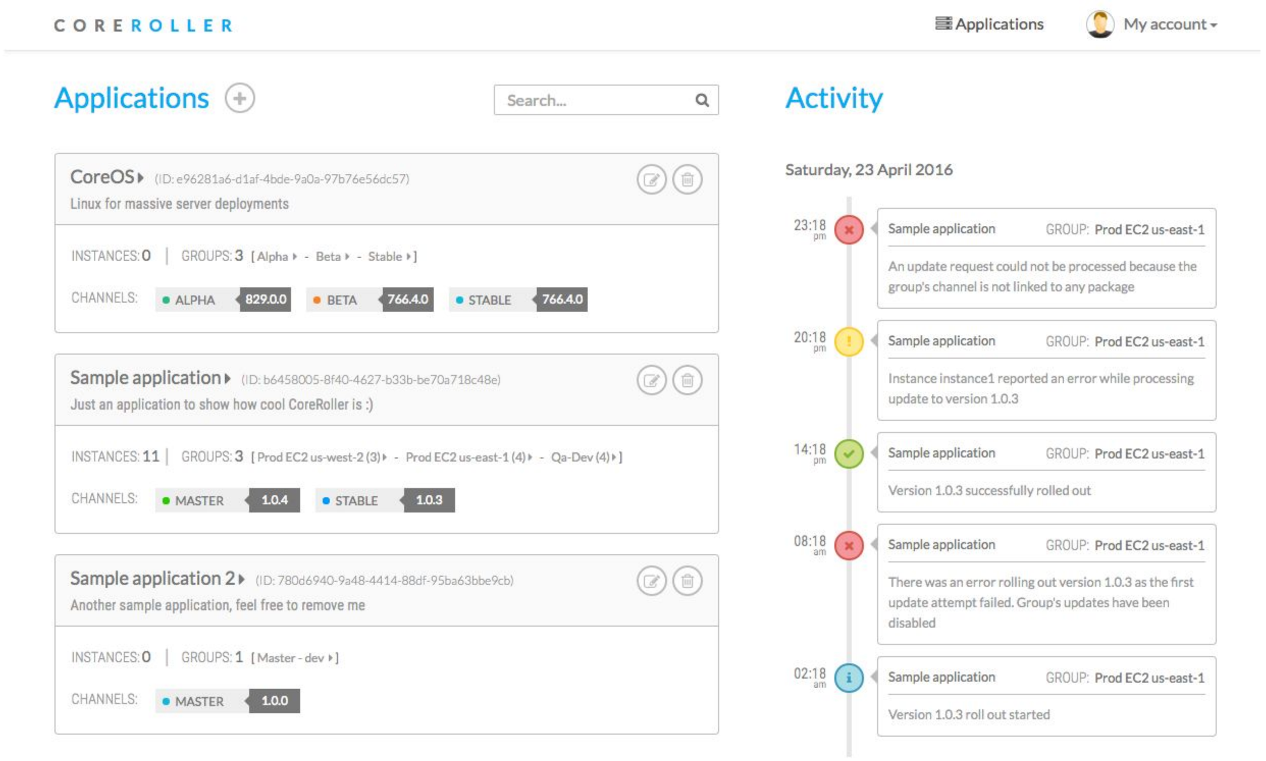
Introducing CoreRoller !

https://github.com/**coreroller/coreroller**

4.2 - Update process management

CoreRoller features:

- **Omaha Server** based on the Omaha protocol developed by Google
- **Dashboard** to control and **monitor** your applications updates
- Dashboard for **statistics** about versions installed in your instances, updates progress status, etc
- Admin panel to **configure** the Omaha update protocol per group of agents
- **HTTP Restful** and Golang **APIs**


COREROLLER

Applications   My account

Applications

Search...

CoreOS (ID: e96281a6-d1af-4bde-9a0a-97b76e56dc57)
Linux for massive server deployments

INSTANCES: 0 | GROUPS: 3 [Alpha - Beta - Stable]

CHANNELS: ● ALPHA 829.0.0 ● BETA 766.4.0 ● STABLE 766.4.0

Sample application (ID: b6458005-8f40-4627-b33b-be70a718c48e)
Just an application to show how cool CoreRoller is :)

INSTANCES: 11 | GROUPS: 3 [Prod EC2 us-west-2 (3) - Prod EC2 us-east-1 (4) - Qa-Dev (4)]

CHANNELS: ● MASTER 1.0.4 ● STABLE 1.0.3

Sample application 2 (ID: 780d6940-9a48-4414-88df-95ba63bbe9cb)
Another sample application, feel free to remove me

INSTANCES: 0 | GROUPS: 1 [Master - dev]

CHANNELS: ● MASTER 1.0.0

Activity

Saturday, 23 April 2016

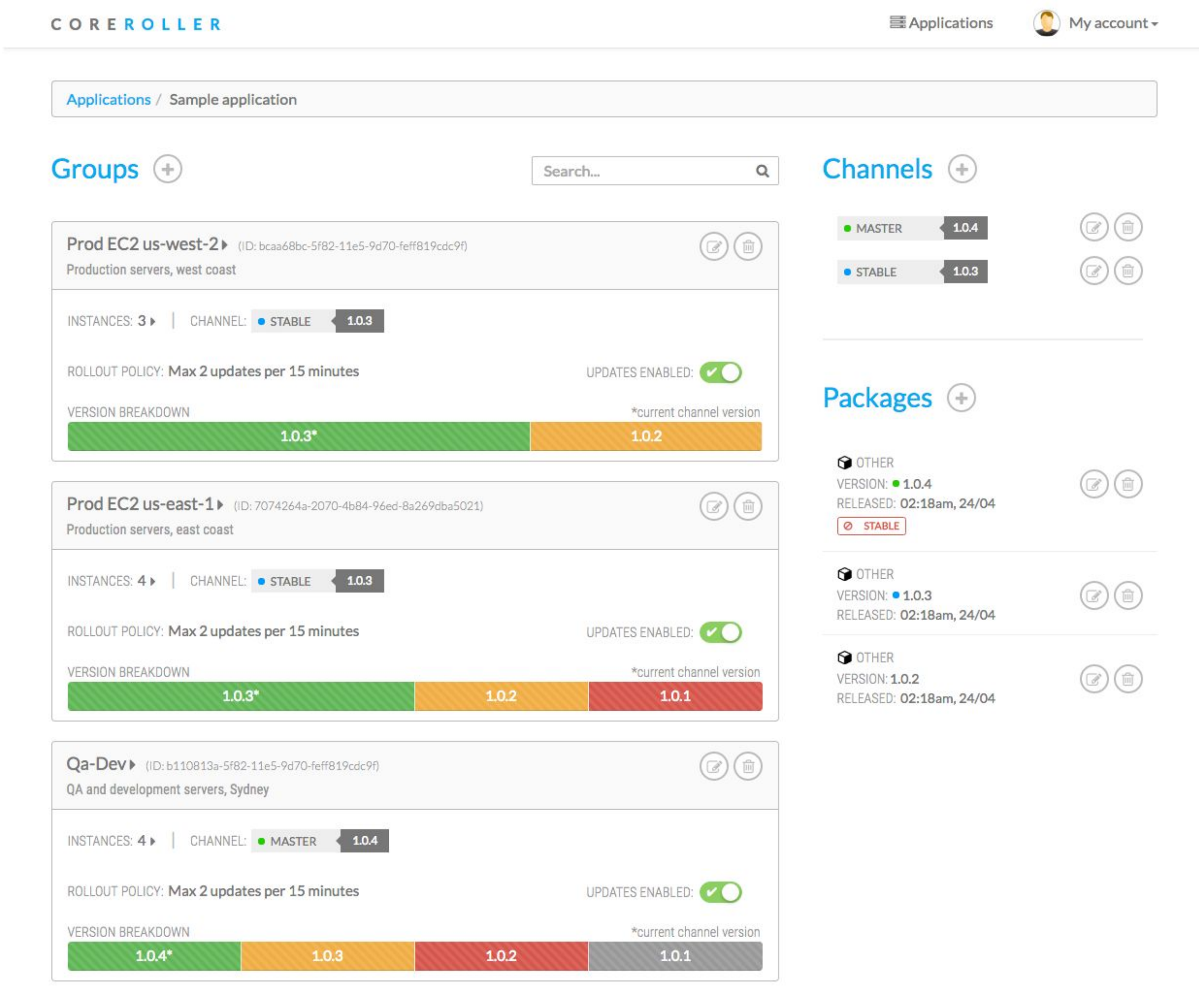
23:18 pm — Sample application — GROUP: Prod EC2 us-east-1
An update request could not be processed because the group's channel is not linked to any package

20:18 pm — Sample application — GROUP: Prod EC2 us-east-1
Instance instance1 reported an error while processing update to version 1.0.3

14:18 pm — Sample application — GROUP: Prod EC2 us-east-1
Version 1.0.3 successfully rolled out

08:18 am — Sample application — GROUP: Prod EC2 us-east-1
There was an error rolling out version 1.0.3 as the first update attempt failed. Group's updates have been disabled

02:18 am — Sample application — GROUP: Prod EC2 us-east-1
Version 1.0.3 roll out started



COREROLLER

Applications My account

Applications / Sample application

Groups

Search...

Prod EC2 us-west-2 (ID: bcaa68bc-5f82-11e5-9d70-feff819cdc9f)
Production servers, west coast

INSTANCES: 3 | CHANNEL: ● STABLE 1.0.3

ROLLOUT POLICY: Max 2 updates per 15 minutes
UPDATES ENABLED: ✓

VERSION BREAKDOWN *current channel version

1.0.3* | 1.0.2

Prod EC2 us-east-1 (ID: 7074264a-2070-4b84-96ed-8a269dba5021)
Production servers, east coast

INSTANCES: 4 | CHANNEL: ● STABLE 1.0.3

ROLLOUT POLICY: Max 2 updates per 15 minutes
UPDATES ENABLED: ✓

VERSION BREAKDOWN *current channel version

1.0.3* | 1.0.2 | 1.0.1

Qa-Dev (ID: b110813a-5f82-11e5-9d70-feff819cdc9f)
QA and development servers, Sydney

INSTANCES: 4 | CHANNEL: ● MASTER 1.0.4

ROLLOUT POLICY: Max 2 updates per 15 minutes
UPDATES ENABLED: ✓

VERSION BREAKDOWN *current channel version

1.0.4* | 1.0.3 | 1.0.2 | 1.0.1

Channels

● MASTER 1.0.4

● STABLE 1.0.3

Packages

OTHER
VERSION: ● 1.0.4
RELEASED: 02:18am, 24/04
⊘ STABLE

OTHER
VERSION: ● 1.0.3
RELEASED: 02:18am, 24/04

OTHER
VERSION: 1.0.2
RELEASED: 02:18am, 24/04

4.2 - Update process management

Demo !

5 - Docs & Opening

1. PXE + DHCP Proxy
2. CoreOS Toolbox
3. Enterprise / On premise
4. More / Next ?

5 - Docs & Opening

5.1 - PXE + ProxyDHCP

http://download.intel.com/design/archives/wfm/downloads/**pxespec**.pdf

http://www.thekelleys.org.uk/**dnsmasq**/doc.html

https://en.wikipedia.org/wiki/Preboot_Execution_Environment#**Proxy_DHCP**

5 - Docs & Opening

5.2 - Tips and Tricks

***CoreOS Toolbox** is a small script, **shipped with CoreOS images**, that launches a container to let you bring in your favorite debugging or admin tools.*

https://thepracticalsysadmin.com/coreos-tips-and-tricks/

https://github.com/**coreos/toolbox**

5 - Docs & Opening

5.3 - Enterprise needs / on Premise services

CoreOS utils:

- `https://github.com/`**`coreos/discovery.etcd.io`**
- `https://github.com/yodlr/`**`CoreGI`**

Classic Linux softwares and config:

- `https://coreos.com/os/docs/latest/#running-coreos`
- `SSSD / SELinux / User Managements / Networking` (CoreOS official Doc link)

5 - Docs & Opening

5.4 - More / Next

Nomad cluster with Consul integration on top of CoreOS.

- Scheduling
- Service discovery
- Lightweight

https://github.com/pires/nomad-vagrant-coreos-cluster

5 - Docs & Opening

5.4 - More / Next

Kubernetes on top of CoreOS.

- Scheduling
- Service discovery
- Service orchestration
- Service delivery

https://github.com/**kubernetes/kubernetes**

That’s all !

CoreOS Paris UG Meetup #5

CoreOS Linux - 101
OS insides + cluster bootstrapping

2016-07-19

Xavier Krantz (xakraz)

https://github.com/xakraz

Systems engineer

SRE @Criteo