

Diary of a rkt developer

Iago López Galeiras

Iago López Galeiras

- Living in Berlin
- *Kinvolk* co-founder
- Worked together with *CoreOS* on *rkt* the last 20 months
- A *rkt* maintainer



github: <https://github.com/iaguis>
twitter: @iaguis



Berlin-based software company building
foundational Linux technologies

Some examples of what we work on...





Find out more about us...

Blog: <https://kinvolk.io/blog>

Github: <https://github.com/kinvolk>

Twitter: <https://twitter.com/kinvolkio>

Email: hello@kinvolk.io

Plan

What is rkt?

What tasks involve working on rkt?

How do we develop and run tests?

How do we work with other communities?

How do we stay current?

Warning!

This section includes stolen slides!

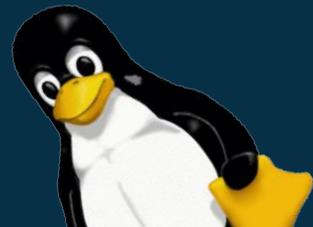


a modern, secure, composable
container runtime

rkt

simple CLI tool
golang + Linux
self-contained

init system/distro agnostic





an implementation of appc
(image format, execution environment)



a standard application container

open specification

associated tooling



appc

github.com/appc/spec ("appc spec")

github.com/appc/acbuild

github.com/appc/docker2aci

github.com/appc-containernetworking/cni

[github.com/appc/...](https://github.com/appc/)

appc spec in a nutshell

- Image Format (ACI)
 - what does an application consist of?
- Image Discovery
 - how can an image be located?
- Pods
 - how can applications be grouped and run?
- Executor (runtime)
 - what does the execution environment look like?

simple CLI tool

no (mandatory) daemon:

apps run directly under spawning process

swappable stage1s: nspawn, KVM, fly

How did rkt start

- Not your typical open-source project
- Well known from the start
- Bay Area startup environment
 - Hacker News
 - Hype
 - Startup wars
 - Container wars
 - Not what we're used to in Berlin

How do we communicate?

- Distributed team
- People in San Francisco and in Berlin
 - CoreOS team in SF
 - Kinvolk & CoreOS team in Berlin
- Not very easy at the beginning
 - Needed to learn a new project & tech
 - Time difference

Communication tools

- **GitHub**
- **Slack (internal)**
- **IRC (external) #rkt and #rkt-dev @ Freenode**
- **rkt-dev mailing list**
- **Internal syncs on Hangouts**
- **Community syncs on Zoom (before May Hangouts On Air)**

Developer setup

- My setup:
 - Arch Linux
 - Golang
 - Vim
 - i3wm
 - Chromium
 - gnome-terminal



```
type ProcAttr struct {
    // If Dir is non-empty, the child changes into the directory before
    // creating the process.
    Dir string
    // If Env is non-nil, it gives the environment variables for the
    // new process in the form returned by Environ.
    // If it is nil, the result of Environ will be used.
    Env []string
    // Files specifies the open files inherited by the new process. The
    // first three entries correspond to standard input, standard output, and
    // standard error. An implementation may support additional entries,
    // depending on the underlying operating system. A nil entry corresponds
    // to that file being closed when the process starts.
    Files []*file
    // Operating system-specific process creation attributes.
    // Note that setting this field means that your program
    // may not execute properly or even compile on some
    // operating systems.
    Sys *syscall.SysProcAttr
}

ProcAttr holds the attributes that will be applied to a new process started by StartProcess.

type Process struct {
    // pid int
    // contains filtered or unexported fields
}

Process stores the information about a process created by StartProcess.

func FindProcess

func FindProcess(pid int) (*Process, error)

FindProcess looks for a running process by its pid.

The Process it returns can be used to obtain information about the underlying operating system
process.

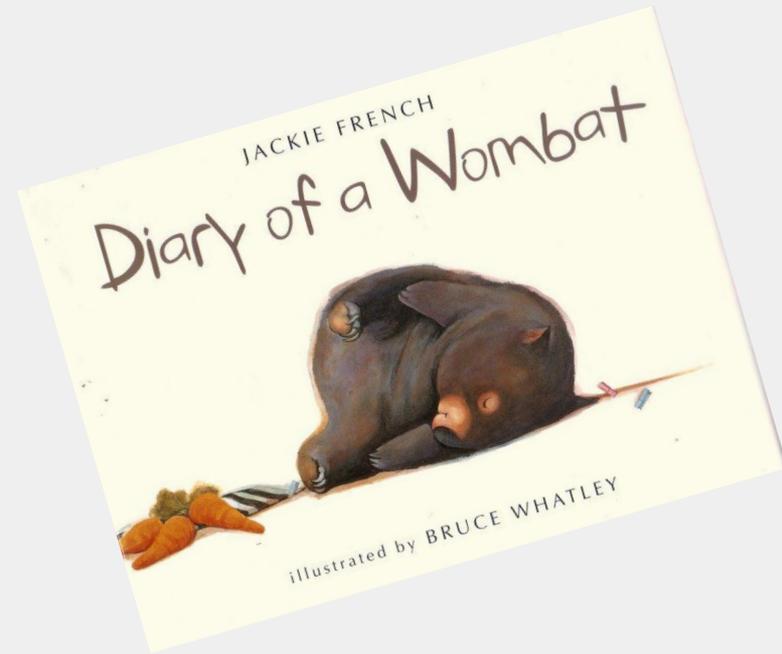
On Unix systems, FindProcess always succeeds and returns a Process for the given pid, regardless
of whether the process exists.

func StartProcess
```

```
149 var cmdRkt = &cobra.Command{
150     Use: "rkt [command]",
151     Short: cliDescription,
152     Long: "A CLI for running app containers on Linux.
153
154 To get the help on any specific command, run "rkt help command".",
155     BashCompletionFunction: bash_completion_func,
156     Run: runMissingCommand,
157 }
158
159
160 func init() {
161     sf, err := rktflag.NewSecFlags("none")
162     if err != nil {
163         fmt.Fprintf(os.Stderr, "rkt: problem initializing: %v", err)
164     }
165     os.Exit(1)
166 }
167
168 globalFlags.InsecureFlags = sf
169
170 cmdRkt.PersistentFlags().BoolVar(&globalFlags.Debug, "debug", f
171     else, "print out more debug information to stderr")
172 cmdRkt.PersistentFlags().Var((*absDir)(&globalFlags.Dir), "dir"
173     "rkt data directory")
174 cmdRkt.PersistentFlags().Var((*absDir)(&globalFlags.SystemConfi
175     gDir), "system-config", "system configuration directory")
176 cmdRkt.PersistentFlags().Var((*absDir)(&globalFlags.LocalConfig
177     Dir), "local-config", "local configuration directory")
178 cmdRkt.PersistentFlags().Var((*absDir)(&globalFlags.UserConfigD
179     ir), "user-config", "user configuration directory")
180 cmdRkt.PersistentFlags().Var(&globalFlags.InsecureFlags, "insecu
181     re-options",
182     "fmt.Sprintf("comma-separated list of security features to d
183     isable. Allowed values: %s",
184     globalFlags.InsecureFlags.PermisibleString()))
185 cmdRkt.PersistentFlags().BoolVar(&globalFlags.TrustKeysFromHTTP
186     S, "trust-keys-from-https",
187     "automatically trust gpg keys fetched from https")
188 }
```

NORMAL master /rkt.go init() < utf-8 go 41% 160:1

Diary of a rkt developer



Monday

Morning: Slept

Afternoon: Slept

Evening: Watched YouTube

Night: Filed PR!

Roles

- **Developer**
- **Maintainer**
 - **Tester**
 - **Release manager**

Diary of a rkt developer

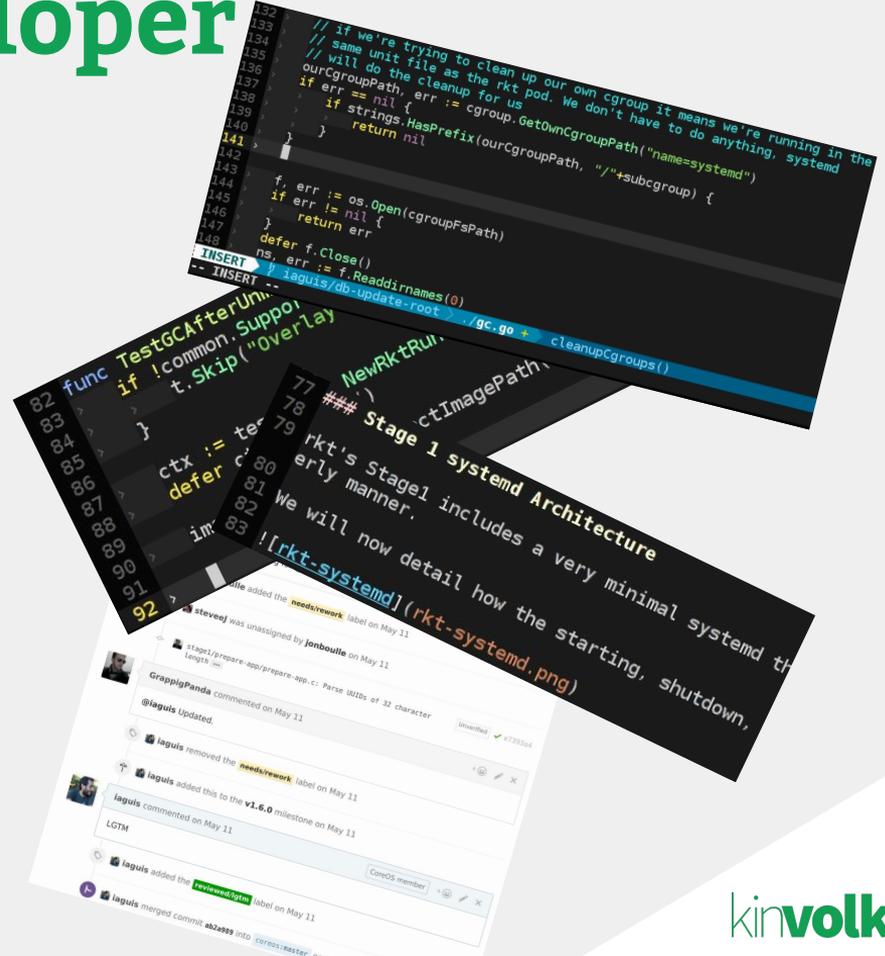
Monday

Morning: Wrote code

Afternoon: Wrote tests

Evening: Wrote docs

Night: Helped new contributors



Diary of a rkt maintainer

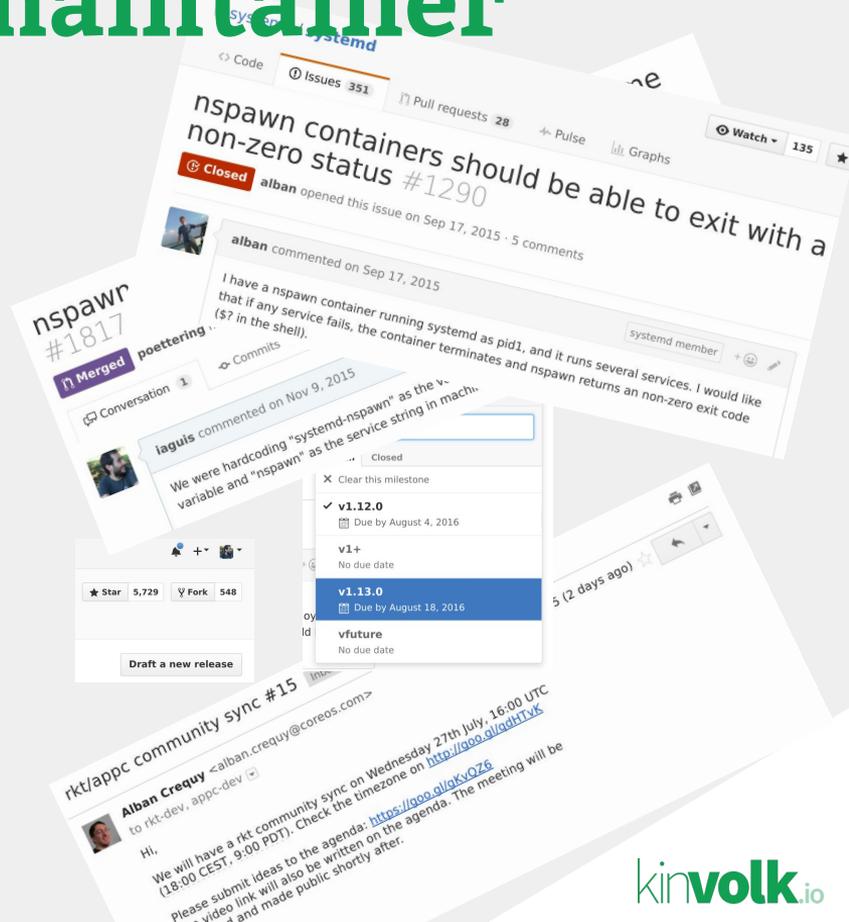
Tuesday

Morning: Filed/fixed issues on projects rkt depends on

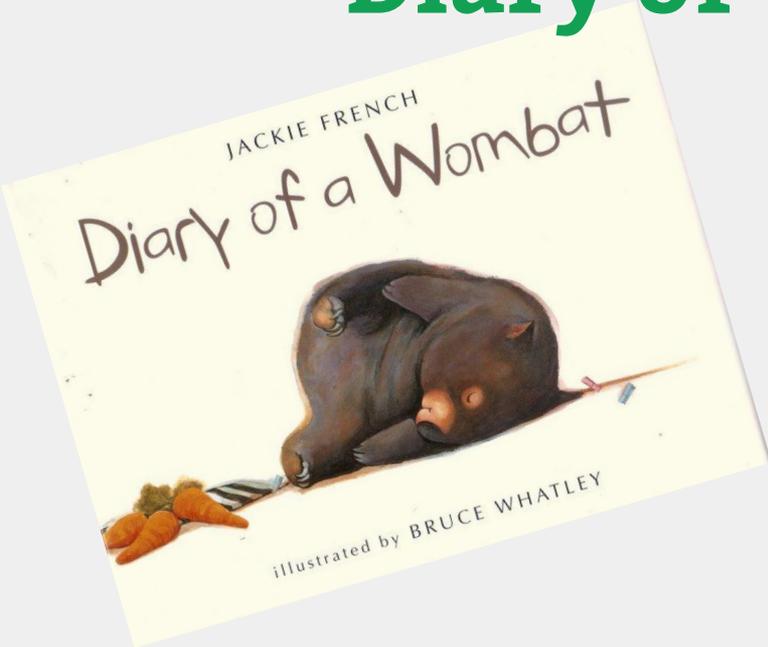
Afternoon: Triaged issues

Evening: Cut a release

Night: Organized community sync



Diary of a rkt maintainer



Tuesday

Morning: Gave talk about rkt

Afternoon: Slept

Evening: Slept

Night: Slept

Feature/bugfix workflow

- Somebody files an issue asking for a feature or complaining about a bug
- We tag and prioritize

Allow changing port numbers for `serviceActivated` containers #2135

Closed tv42 opened this issue on Feb 5 · 2 comments



tv42 commented on Feb 5

Earlier discussion in <https://groups.google.com/forum/#!topic/rkt-dev/JD0jwJy74R0> says that the port number is used to match the socket to the right application. Reading between the lines, the `name` of the protocol in the ACI metadata is ignored, and the `rkt run option --port=NAME:HOSTPORT` is ignored. To keep things easier to talk about, I'm going to call the systemd listening socket `sysport`, a port in the ACI metadata `aciport`, with name `aciportname`.

Right now (as far as I understand it), there is *no way* to change the port a `serviceActivated` container listens on without rebuilding the ACI. This is clearly not as nice as with the non-`serviceActivated` containers.

Instead of the current logic (matching `sysport` to `aciport`), I would like to propose the following:

1. if `--port` is not given, behave just like now: match `sysport` to `aciport`
2. with `--port=NAME:HOSTPORT`, match `sysport` to `HOSTPORT`, `NAME` to `aciportname`, and the related `aciport` to find what app to pass the listening fd to.

stevej added `kind/enhancement` `area/usability` labels on Feb 5

iaguis added this to the **v1.1.0** milestone on Feb 5

iaguis self-assigned this on Feb 5



iaguis commented on Feb 5

CoreOS member + 🗨️ ✎️ ✕

This was actually a bug, the way you describe it is how it's supposed to work.

Here's a fix: #2138



iaguis added the **priority/P0** label on Feb 5



iaguis commented on Feb 5

CoreOS member + 🗨️ ✎️ ✕

We're still discussing the right UX for this. Stay tuned...

Feature workflow (1/5)

- Then somebody makes a PR for it

stage1: fix socket activation port matching #213

Merged **iaguis** merged 5 commits into `coreos:master` from `kinvolk:iaguis/fix-socketactivation` on Feb 18

Conversation 12 Commits 5 Files changed 5

iaguis commented on Feb 5 CoreOS member

We were using the port of the image manifest to set up the socket unit file inside the pod. This means it only worked if the port in the image manifest and the pod in the pod manifest (set by rkt at runtime) were the same.

The reason is that systemd inside the host matches based on the socket port number, and since the socket is created on the host, it will have the host port number. If this number is different from the one in the image manifest, systemd will not assign the socket to the correct app in the pod because it will expect a socket with the image manifest port number.

The solution is using the host port number for the socket unit file.

Fixes #2135

Feature workflow (2/5)

- We review
 - General feedback about the PR
 - Code review. Ask for a test. Ask for doc changes

Documentation/using-rkt-with-systemd.md [View full outdated diff](#)

178	178	KillMode=mixed
179	179	...
180	180	
181		-Finally, start the socket unit:
	181	+Finally, instantiate the socket unit. In this case, with port 8080:

 **jonboule** added a note on Feb 8 CoreOS member + 😊 ✎ ✕

It's confusing and a bit magical why the port is different here - could do with some more detailed explanation of how 80 is the default in the aci, but can be overridden,etc

tests/rkt_socket_activation_test.go [View full outdated diff](#)

```
... @ -64,7 +64,7 @@ func TestSocketActivation(t *testing.T) {
64     echoImage := patchTestACI("rkt-inspect-echo.aci",
65         "--exec=/echo-socket-activated",
66         fmt.Sprintf("--ports=%d-tcp,protocol=tcp,port=%d,socketActi
67     + fmt.Sprintf("--ports=test-port,protocol=tcp,port=80,socketA
```

 **alban** added a note on Feb 18 CoreOS member + 😊 ✎ ✕

No need to use `fmt.Sprintf` if there is no `%d`

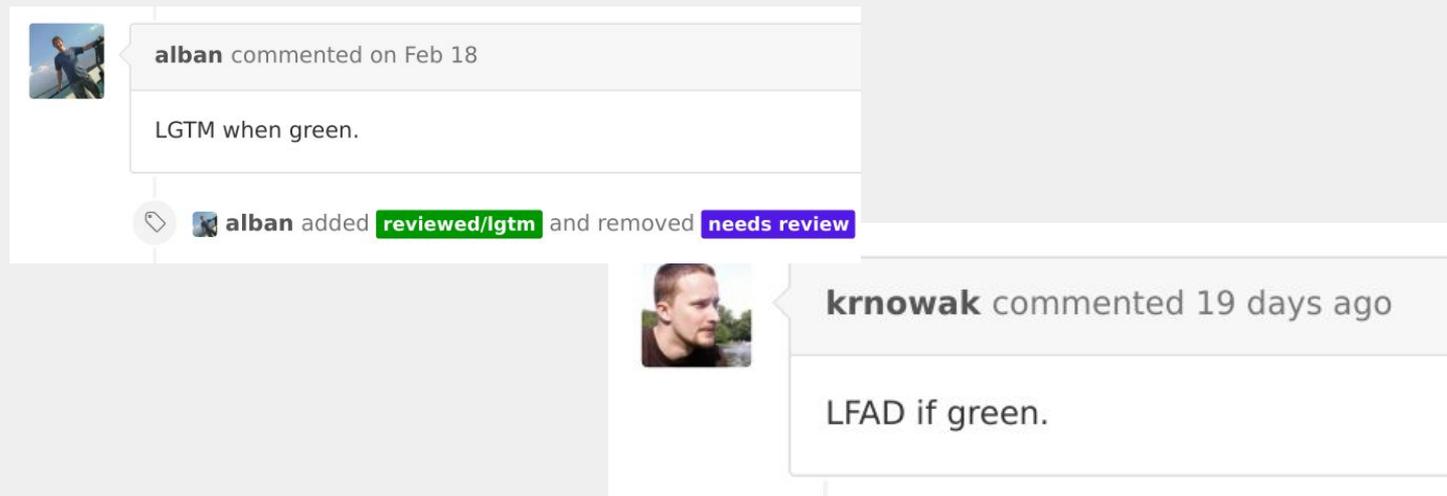
Feature workflow (3/5)

- The developer force-pushes to the branch
- Writes an “Updated” message
- We repeat this n times. Until...



Feature workflow (4/5)

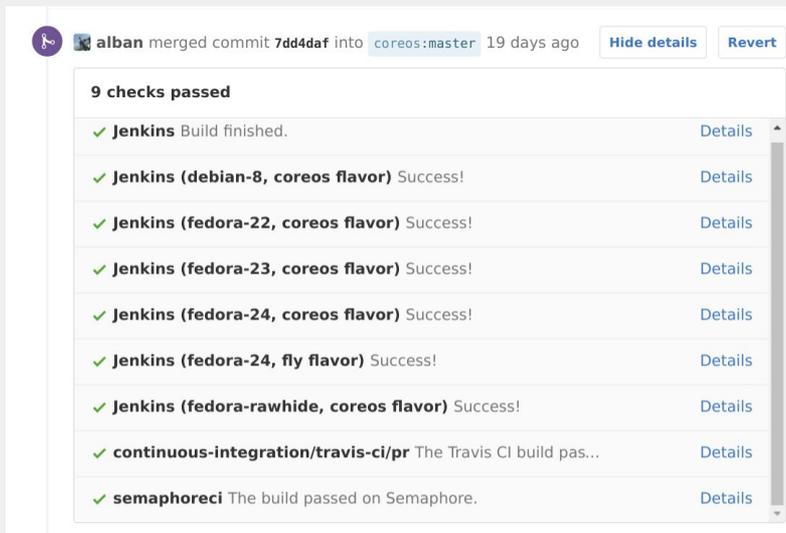
- A maintainer writes LGTM (or LFAD :P)



The screenshot displays a GitHub commit history entry. On the left, a small profile picture of a person is visible. The main content area shows a comment by user **alban** dated Feb 18, which reads "LGTM when green." Below the comment, a status change is recorded: **alban** added the **reviewed/igtm** label (highlighted in green) and removed the **needs review** label (highlighted in purple). To the right, a separate comment box shows a profile picture of user **krnowak** with the text "krnowak commented 19 days ago" and "LFAD if green."

Feature workflow (5/5)

- We wait for CI to finish and...
- We merge!



The screenshot shows a GitHub commit page for user 'alban' with commit hash '7dd4daf' merged into the 'coreos:master' branch, 19 days ago. The commit message is 'alban merged commit 7dd4daf into coreos:master 19 days ago'. Below the commit information, there are two buttons: 'Hide details' and 'Revert'. The main content area displays '9 checks passed' and a list of CI checks, all of which are successful. Each check entry includes a green checkmark, the name of the check, a brief description, and a 'Details' link.

Check Name	Status	Details
Jenkins	Build finished.	Details
Jenkins (debian-8, coreos flavor)	Success!	Details
Jenkins (fedora-22, coreos flavor)	Success!	Details
Jenkins (fedora-23, coreos flavor)	Success!	Details
Jenkins (fedora-24, coreos flavor)	Success!	Details
Jenkins (fedora-24, fly flavor)	Success!	Details
Jenkins (fedora-rawhide, coreos flavor)	Success!	Details
continuous-integration/travis-ci/pr	The Travis CI build pas...	Details
semaphoreci	The build passed on Semaphore.	Details

You mentioned CI...

- Unit testing on Travis
- Works fine
- rkt is complicated software
 - Relies a lot on other software
- We needed functional testing

Functional testing

- rkt is a special kind of software
- Interacts a lot with the host and requires a lot of privileges on it
- We didn't want to have to create our own CI server
- @alban did an initial analysis of CI services:
 - <https://github.com/coreos/rkt/issues/600#issuecomment-87655911>

CI service analysis

- Travis: CAP_SYS_ADMIN ✗
- Drone: Changing mount options ✗
- Codeship: root ✗
- Semaphore: Changing mount options ✗
- Wercker: Changing mount options ✗
- CircleCI: mknod ✗

Nothing works, but...

- Semaphore runs tests on containers
- But they gave us access to a beta image that uses VMs instead
- After using that, we had all the privileges we needed to test rkt! Yay!!

All was fine except

- Semaphore has
 - An old Ubuntu (14.04 LTS v1503)
 - An old kernel (3.13)
 - No users
 - No overlay fs
 - ...
 - Plus we want to test other distros

Enter Jenkins

- **Bite the bullet and set up our own CI**
- **Jenkins + AWS plugin**
- **Annoying UI**
- **Complicated to set up**
- **But we can test on arbitrary distros with arbitrary kernels**

Jenkins

- **Periodically test rkt**
 - With lots of configurations
- **Test each PR**
 - More limited testing
- **Distros**
 - Fedora 22-rawhide
 - Debian testing
 - More to come...

CI TODO

- **Fighting test flakes**
- **Make periodic test really pass**
 - **Still some problems with certain rkt configs**
- **Test KVM backend**
 - **Not possible on AWS**
 - **Still use Semaphore for that**
 - **Some flaky tests**

Work with related projects

- We depend/rely on several projects
- We need to communicate our needs
- Sometimes even implement them
- Here're some of the projects we collaborate with

systemd

- rkt uses and interfaces with
 - systemd-nspawn
 - systemd (PID 1)
- We use
 - The mailing list
 - GitHub
- We talk to Lennart
 - he's in Berlin too

man: nspawn is used in production these days, admit that

Previously, the man page suggested to only use nspawn for testing, building, and debugging things. However, it is nowadays used in production and used as building block for rocket, hence let's just admit that it's pretty much production ready.

 master  v231 ... v220

 **poettering** committed on May 5, 2015

1 parent [93c4747](#)

systemd.conf 2016

Sept. 28th - Oct. 1st @ betahaus in Berlin

Day 1: Hands on workshop session

Day 2 & 3: Presentations

Day 4: Hackfest

<https://conf.systemd.io>

systemd

- **Some examples of issues we fixed:**
 - Propagate exit status
 - notify-ready
 - InaccessiblePaths=
 - Cgroup issues
 - ...

Linux kernel

- Being a container runtime means
 - We use some of the latest features in Linux
 - Some are not so mature
- We use the mailing list of course
- Examples:
 - CONFIG_PROC_CHILDREN
 - cpuset issue
 - Use overlay fs and find some issues
 - Test cgroup v2

go libraries

- We depend on a lot of go libraries
- Go is hip -> GitHub
- Some examples
 - ql database
 - gexpect
 - docker2aci

Linux distributions

- We want to help packagers to get rkt everywhere
 - **Fedora...** and eventually CentOS/RHEL
 - **Debian**
 - **Arch Linux**

Kubernetes

- Abstraction over containers in clusters
- Became major driving force of rkt
- Pushing container runtime agnostic
- Needed several new features in rkt

- Mostly rktnetes team interacting with k8s
- They ask, we provide :)

Work with other companies

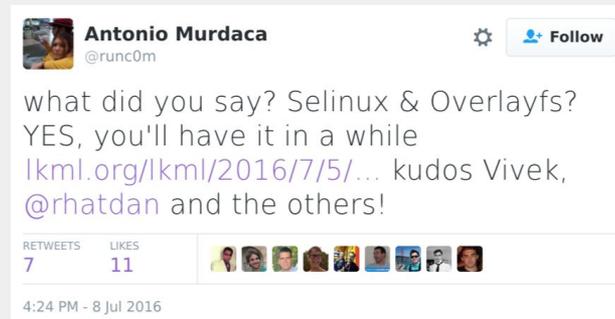
- **CoreOS**
 - From the beginning :)
- **Intel**
 - KVM stage1
 - Communication: IRC and community syncs
- **Blablacar**
 - Issues and conferences

How do we stay current

- We use a lot of cutting-edge technologies
- How do we stay current?
- Read
- Fiddle
- Read

How do we stay current

- LWN
 - Great in-depth articles
 - All the new features in the kernel
- LKML
 - When LWN is not enough
- We analyze new features in systemd to see how we can benefit
- Conferences
- Twitter :)



That's about it...

Timeline & blog

[go to the next tab]

Long form blog story coming within the next couple days.

Twitter: @kinvolkio

Thanks!

Get involved

Github: <https://github.com/coreos/rkt>

Mailing list: <https://groups.google.com/forum/#!forum/rkt-dev>

#rkt-dev @ Freenode



Find out more about us...

Blog: <http://kinvolk.io/blog>

Github: <https://github.com/kinvolk>

Twitter: <https://twitter.com/kinvolkio>

Email: hello@kinvolk.io

Questions?