

KubeCon EU SIG-ContribX Deep Dive Developer Guide

Problem Statement

- Developer guide today is extremely outdated (avg 3-4mo's ~= forever)
 - <https://git.k8s.io/community/contributors/devel>
 - <https://github.com/kubernetes/community/issues/1919>
- Lacks details of dev/build/test cycle
- Growing community of dev's need good info
- Tragedy of commons
- Need active/sustained volunteers...group ownership

K8s 2016



kubernetes

Kubernetes
<http://kubernetes.io>

Repositories  People **64**

Filters ▾

kubernetes.github.io HTML ★ 27 📄 81
Website/documentation repo
Updated Mar 23, 2016

kubernetes Go ★ 13,454 📄 3,898
Container Cluster Manager from Google
Updated Mar 23, 2016

People **64**



K8s 2018



Kubernetes

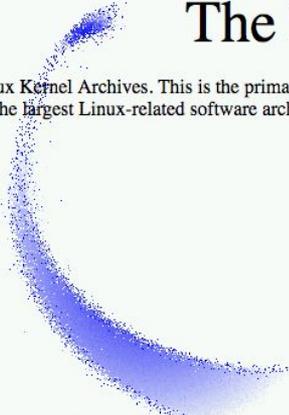
Kubernetes
<http://kubernetes.io>

 **Repositories** 56  **People** 657  **Teams** 304  **Projects** 2

Pinned repositories

<p>kubernetes</p> <p>Production-Grade Container Scheduling and Management</p> <p> Go  35.5k  12.6k</p>	<p>features</p> <p>Features tracking repo for Kubernetes releases</p> <p> 301  170</p>	<p>community</p> <p>Kubernetes con</p> <p> Go  1.8k</p>
<p>website</p> <p>Kubernetes website and documentation repo:</p> <p> HTML  467  2.4k</p>	<p>test-infra</p> <p>Test infrastructure for the Kubernetes project.</p> <p> Go  342  417</p>	<p>examples</p> <p>Kubernetes app</p> <p> Shell  7k</p>

An analogy



The Linux Kernel Archives

Welcome to the Linux Kernel Archives. This is the primary site for the Linux kernel source, but it has much more than just kernels - we have 20 GB of disk space set aside for mirroring the largest Linux-related software archives.

Protocol	Location
HTTP	http://ftp.kernel.org/pub
FTP	ftp://ftp.kernel.org/pub
NFS	ftp.kernel.org:/pub
SMB/CIFS	\\ftp.kernel.org\pub

What is Linux?

Linux is a Unix clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net. It aims towards POSIX compliance.

It has all the features you would expect in a modern fully-fledged Unix, including true multitasking, virtual memory, shared libraries, demand loading, shared copy-on-write executables, proper memory management, and TCP/IP networking.

Linux was first developed for 386/486-based PCs. These days it also runs on DEC Alphas, SUN SPARC, M68000 machines (like Atari and Amiga), MIPS, and the PowerPC. Additional ports (such as ARM) are in progress.

More Information

There is much [information about Linux on the web](#).

An analogy

Cool Linux Sites

These are Linux sites maintained by people with lots of free time.

[Linux Documentation Project](#)

The canonical set of Linux online and printed documentation.

[linux.org](#)

Linux information

[uk.linux.org](#)

Linux information from Great Britain (very good!)

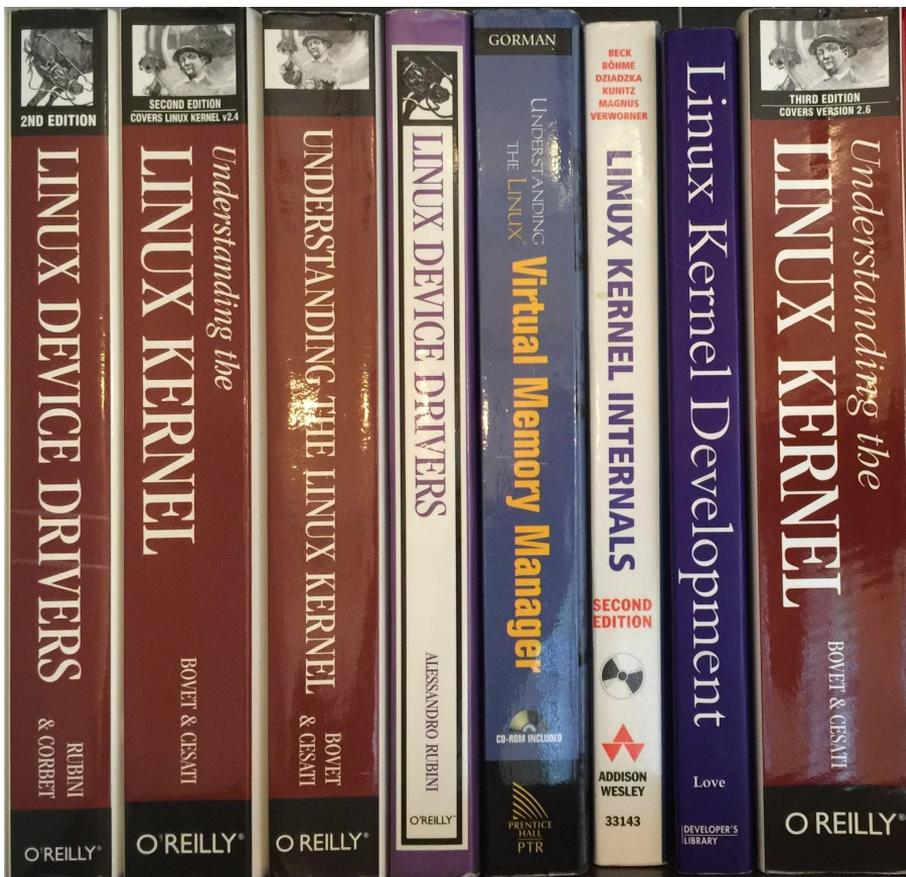
[Linux International](#)

An organization for promoting the use of Linux.

[Linux v2 Information Headquarters](#)

Linux 2.0 upgrade information and patches, along with other info.

An analogy



K8s Internals Developer Online Docs Today

<https://git.k8s.io/community/contributors/devel>

The process of developing and contributing code to the K8s

- **Contributor Guide** ([Please start here](#)) to learn about how to contribute to Kube
- **GitHub Issues** ([issues.md](#)): How incoming issues are triaged.
- **Pull Request Process** (</contributors/guide/pull-requests.md>): When and why p
- **Getting Recent Builds** ([getting-builds.md](#)): How to get recent builds including
- **Automated Tools** ([automation.md](#)): Descriptions of the automation that is runni

Setting up your dev environment, coding, and debugging

- **Development Guide** ([development.md](#)): Setting up your development environr
- **Testing** ([testing.md](#)): How to run unit, integration, and end-to-end tests in your

K8s Internals Developer Online Docs Today

Secure | <https://kubernetes.io/docs/home/?path=contributors&persona=code-contributor&level=found.>



Tutorials

- | | | |
|------------------------------|-----------------------------|---------------|
| 01 - Kubernetes Basics | 05 - Kubernetes 201 | 09 - Clusters |
| 02 - Online Training Courses | 06 - Configuration | 10 - Services |
| 03 - Hello Minikube | 07 - Stateless Applications | |
| 04 - Kubernetes 101 | 08 - Stateful Applications | |

Reference

- | | | |
|----------------------------|----------------------------|-------------------------------------|
| 01 - Standardized Glossary | 04 - Federation API | 07 - Command-line Tools |
| 02 - Using the API | 05 - kubectl CLI | Reference |
| 03 - API Reference | 06 - Setup Tools Reference | 08 - Kubernetes Issues and Security |

K8s Internals Developer Bookshelf Today

K8s Internals Developer Bookshelf Today ...is not



How do we get to:

5. Kernel Synchronization	189	System Call Handler and Service Routines	399
How the Kernel Services Requests	189	Entering and Exiting a System Call	401
Synchronization Primitives	189	Parameter Passing	409
Synchronizing Accesses to Kernel Data Structures	194	Kernel Wrapper Routines	418
Examples of Race Condition Prevention	217		
	222		
6. Timing Measurements	227	11. Signals	420
Clock and Timer Circuits	228	The Role of Signals	420
The Linux Timekeeping Architecture	232	Generating a Signal	433
Updating the Time and Date	240	Delivering a Signal	439
Updating System Statistics	241	System Calls Related to Signal Handling	450
Software Timers and Delay Functions	244		
System Calls Related to Timing Measurements	252	12. The Virtual Filesystem	456
		The Role of the Virtual Filesystem (VFS)	456
7. Process Scheduling	258	VFS Data Structures	462
Scheduling Policy	258	Filesystem Types	481
The Scheduling Algorithm	262	Filesystem Handling	483
Data Structures Used by the Scheduler	266	Pathname Lookup	495
Functions Used by the Scheduler	270	Implementations of VFS System Calls	505
Runqueue Balancing in Multiprocessor Systems	284	File Locking	510
System Calls Related to Scheduling	290		
		13. I/O Architecture and Device Drivers	519
8. Memory Management	294	I/O Architecture	519
Page Frame Management	294	The Device Driver Model	526
Memory Area Management	323	Device Files	536
Noncontiguous Memory Area Management	342	Device Drivers	540
		Character Device Drivers	552
9. Process Address Space	351	14. Block Device Drivers	560
The Process's Address Space	352	Block Devices Handling	560
The Memory Descriptor	353	The Generic Block Layer	566
	357	The I/O Scheduler	572
		Block Device Drivers	585
		Opening a Block Device File	595

What's missing

Comprehensive

- architectural big picture
- core component list
- per component details

Not too hard to draft at high level, but details take time and needs active volunteers.

Curating sufficiently detailed yet abstract text (to minimize ongoing maintenance requirement) is a challenge.

This could just be Subprojects List

<https://git.k8s.io/community/sigs.yaml>



The screenshot shows the GitHub interface for the file `community/sigs.yaml` in the `kubernetes` repository. The file is currently on the `master` branch. A pull request #2095, submitted by `evankanderson`, is being reviewed by `k8s-ci-robot`. The file has 1909 lines (1901 sloc) and is 74.3 KB in size. It is tracked by 96 contributors. The visible portion of the file's content is as follows:

```
1 sigs:
2   - name: API Machinery
3     dir: sig-api-machinery
4     mission_statement: >
5       Covers all aspects of API server, API registration and discovery, generic
6       API CRUD semantics, admission control, encoding/decoding, conversion,
7       defaulting, persistence layer (etcd), OpenAPI, CustomResourceDefinition,
8       garbage collection, and client libraries.
9     label: api-machinery
10    leadership:
11      chairs:
```

This could just be Subprojects List

...that's a yaml file

Yaml is not for humans.

Need us humans to build from this things that are human consumable:

- Contrib-X → automate yaml to web listing ToC?
- new repo requires SIG update their subprojects/repos list
- survey subprojects → do they have a README?

This could be community/contributors/devel

Today that is a tragedy of the commons.

Incomplete. Out of date. Even irrelevant. No organization

 adding-an-APIGroup.md	removed autogenerated munge analytics from files	6 months ago
 api-conventions.md	fix a bad punctuation	3 months ago
 api_changes.md	Add the requirement that GA/stable APIs must have conformance tests, ...	3 months ago
 arch-roadmap-1.png	Added architectural layers diagram, placeholder for Summary section.	6 months ago
 architectural-roadmap.md	Use git.k8s.io for links	4 months ago
 automation.md	Use git.k8s.io for links	4 months ago
 bazel.md	Use git.k8s.io for links	4 months ago
 cherry-picks.md	simplify the cherry pick process documentation	13 days ago
 client-libraries.md	removed autogenerated munge analytics from files	6 months ago
 collab.md	removed unnecessary link from guide README, also removed duplicate te...	4 months ago

This could be community/contributors/devel



carlpett commented on Mar 21

As a wishlist for how the docs could have been structured in a way that I would have found intuitive, something like this (sort of a step-by-step style):

- Prerequisites: Getting the code and development tools (not necessarily covering the "running" part here, just what you'll need to compile it)
- Navigating the code base (what components live where, the gotcha regarding k8s.io/vendor, ...)
- Making a small change (some simple demonstration about what is required to do some typical first change)
- Building your change (could have this before the change part, but since just compiling your change will take ~10-15 minutes, there's a high risk of losing people if they have to do it twice..)
- Running your changed Kubernetes (might be merged with the previous one? Will get long, though...)
- Adding a test
- Making a PR
- (Quick start: Alternative route for the impatient/short summary for reference - commands required to get up and running, etc)

community/contributors/level

This needs:

- Active owners overall for the guide
- Group consensus on the main top level ToC items that are needed
 - Do we top down this? With a list of ToC items we think are the most important?
 - Do we bottom up this? SIGs docs, if existing, link into some ToC?
- Incentives for content creation from SME's
- Incentives for review from SME's

Call to action

We've had a lot of talk on the basic idea

We don't have a set of people working on it

If you have opinions on this, share them with us PLEASE

- Are there docs you know need help/moved: open an issue..right now
- Do you disagree, think it's mostly good...come up here and share your screen and talk us through it
- Do you have automation ideas?
- Umbrella issue: <https://github.com/kubernetes/community/issues/1919>

If you have a vision for what it should look like? Get involved in improving it:

- SIG-ContribX -> Subproject Devel Guide

Timeline?

What's realistic?

Swarm on it the next 3 months?