AiDO
DevOps and more

posti Messaging

Version control (git) in the world of DevOps...

first part „introduction” (1/2)

$ whoami

- Dawid Sadecki
- Tooling Engineer at Atos - things related with global monitoring environment – nagios etc.
- Free software and Unix-like systems enthusiast (mainly Linux and FreeBSD)
- Also interested in automation and whole related ~~shit~~ stuff

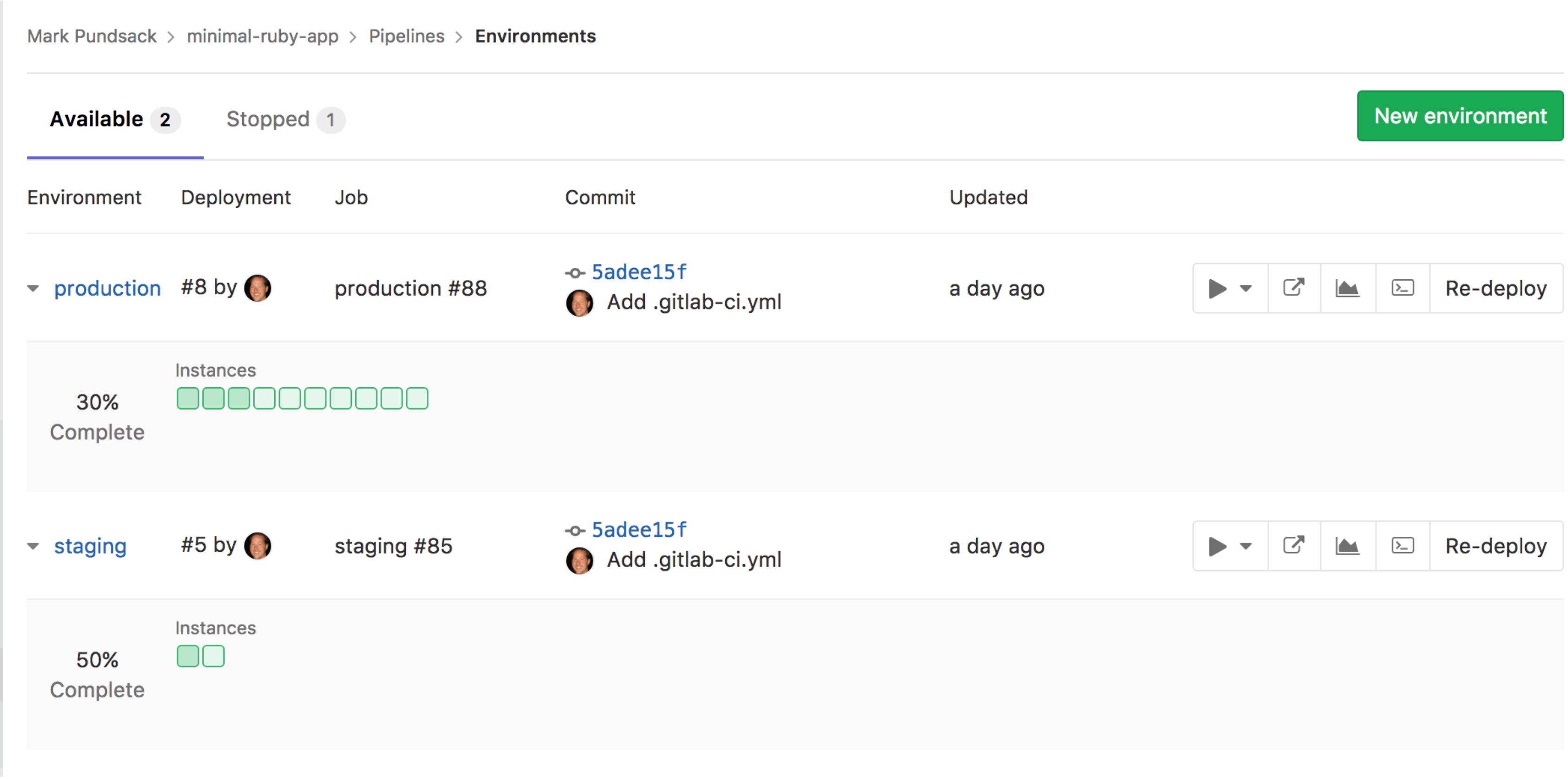
- LinkedIn: linkedin.com/in/dsadecki/
- Email: dawid.sadecki@protonmail.com

1. A bit of theory

what people want?

- Product manager / owner - nice **green** charts
- Developer - one clickable **CI/CD** environment
- SysAdmin - central register of recent **changes**
- „Lucky” man on standby shift - restful **sleep** and night
- Scrum master – I have **no** idea

what people want?

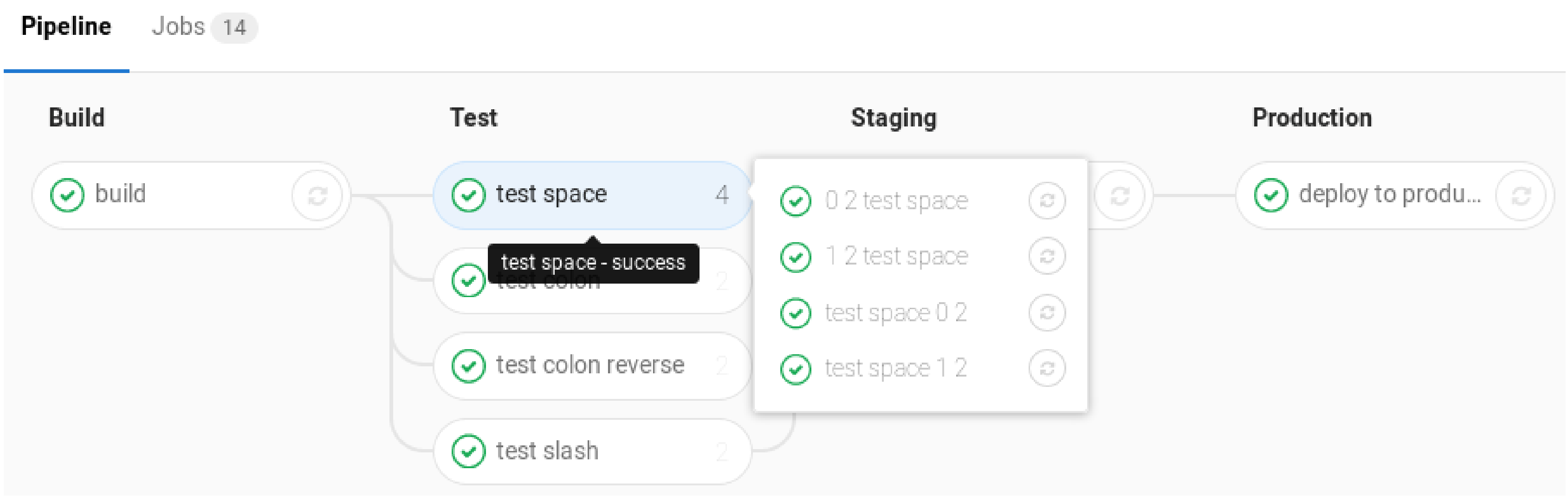


Mark Pundsack > minimal-ruby-app > Pipelines > **Environments**

Environment	Deployment	Job	Commit	Updated	
production	#8 by	production #88	5adee15f Add .gitlab-ci.yml	a day ago	Re-deploy
30% Complete	Instances 🟩🟩🟩⬜⬜⬜⬜⬜⬜⬜				
staging	#5 by	staging #85	5adee15f Add .gitlab-ci.yml	a day ago	Re-deploy
50% Complete	Instances 🟩⬜				

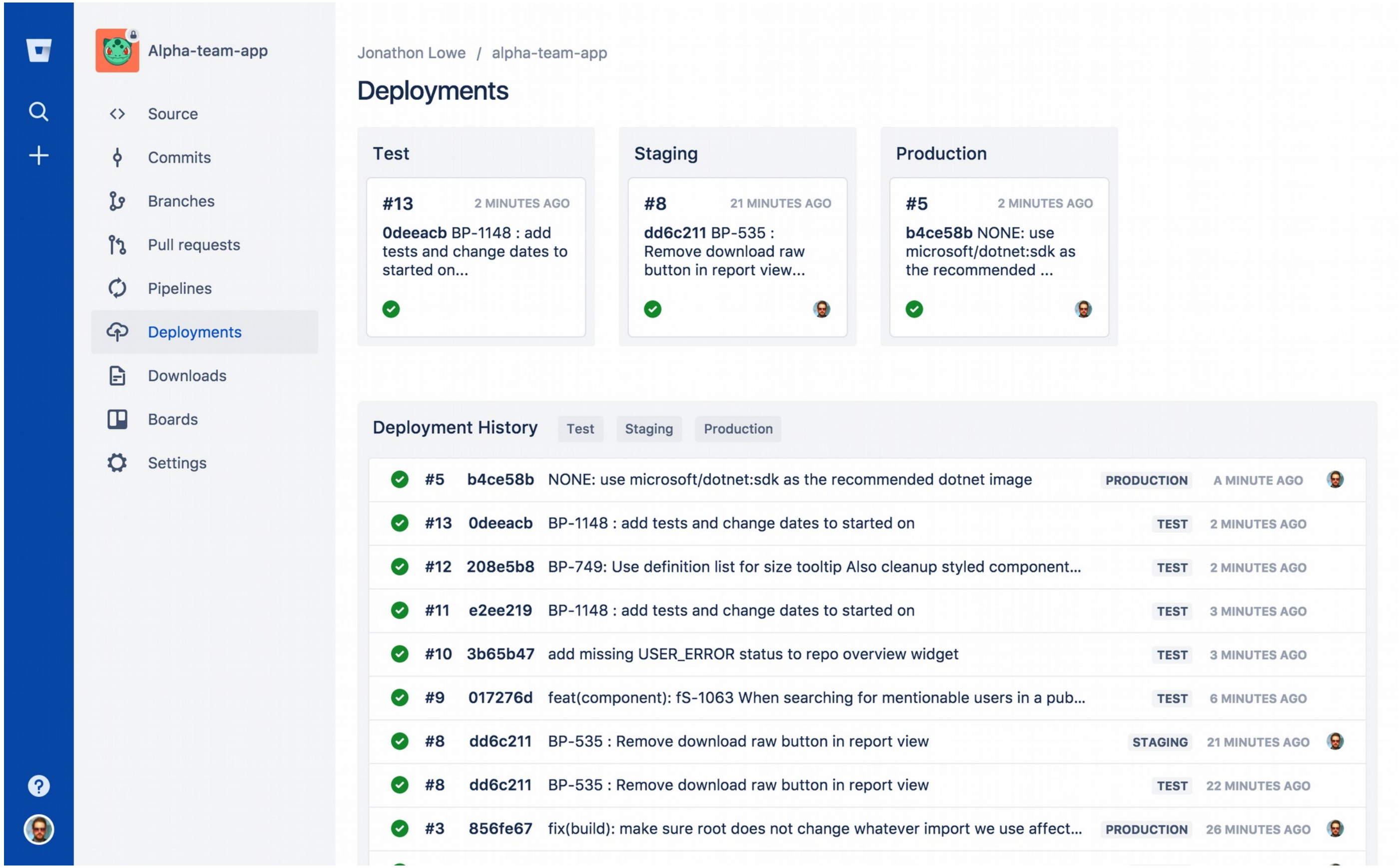
https://about.gitlab.com/blog/2017/10/04/devops-strategy/

what people want?



https://docs.gitlab.com/ee/ci/pipelines.html

what people want?



Jonathon Lowe / alpha-team-app

Deployments

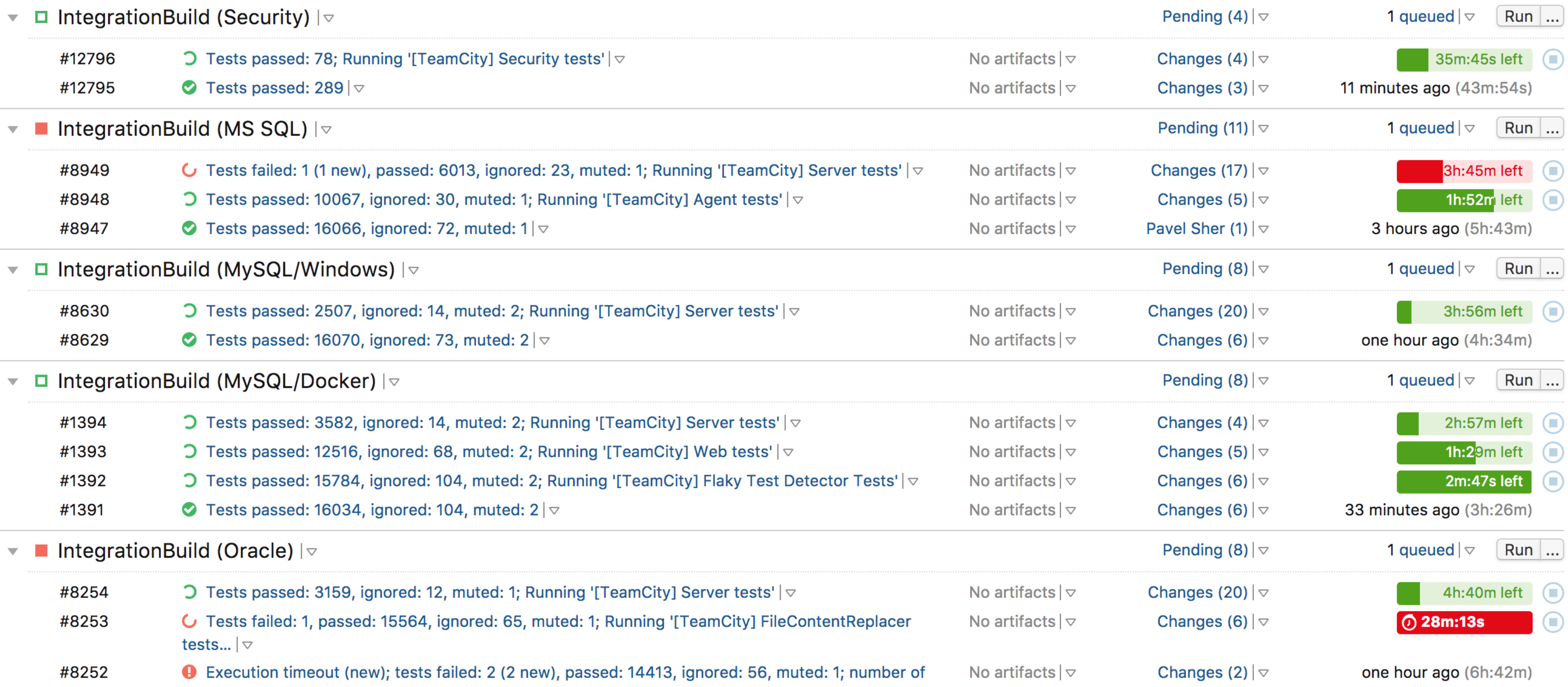
Test	Staging	Production
#13 2 MINUTES AGO 0deeacb BP-1148 : add tests and change dates to started on...	#8 21 MINUTES AGO dd6c211 BP-535 : Remove download raw button in report view...	#5 2 MINUTES AGO b4ce58b NONE: use microsoft/dotnet:sdk as the recommended ...

Deployment History Test Staging Production

#5	b4ce58b	NONE: use microsoft/dotnet:sdk as the recommended dotnet image	PRODUCTION	A MINUTE AGO
#13	0deeacb	BP-1148 : add tests and change dates to started on	TEST	2 MINUTES AGO
#12	208e5b8	BP-749: Use definition list for size tooltip Also cleanup styled component...	TEST	2 MINUTES AGO
#11	e2ee219	BP-1148 : add tests and change dates to started on	TEST	3 MINUTES AGO
#10	3b65b47	add missing USER_ERROR status to repo overview widget	TEST	3 MINUTES AGO
#9	017276d	feat(component): fS-1063 When searching for mentionable users in a pub...	TEST	6 MINUTES AGO
#8	dd6c211	BP-535 : Remove download raw button in report view	STAGING	21 MINUTES AGO
#8	dd6c211	BP-535 : Remove download raw button in report view	TEST	22 MINUTES AGO
#3	856fe67	fix(build): make sure root does not change whatever import we use affect...	PRODUCTION	26 MINUTES AGO

https://bitbucket.org/product/features/pipelines

what people want?



IntegrationBuild (Security)			Pending (4)	1 queued · Run · ...
#12796	Tests passed: 78; Running '[TeamCity] Security tests'	No artifacts	Changes (4)	35m:45s left
#12795	Tests passed: 289	No artifacts	Changes (3)	11 minutes ago (43m:54s)
IntegrationBuild (MS SQL)			Pending (11)	1 queued · Run · ...
#8949	Tests failed: 1 (1 new), passed: 6013, ignored: 23, muted: 1; Running '[TeamCity] Server tests'	No artifacts	Changes (17)	3h:45m left
#8948	Tests passed: 10067, ignored: 30, muted: 1; Running '[TeamCity] Agent tests'	No artifacts	Changes (5)	1h:52m left
#8947	Tests passed: 16066, ignored: 72, muted: 1	No artifacts	Pavel Sher (1)	3 hours ago (5h:43m)
IntegrationBuild (MySQL/Windows)			Pending (8)	1 queued · Run · ...
#8630	Tests passed: 2507, ignored: 14, muted: 2; Running '[TeamCity] Server tests'	No artifacts	Changes (20)	3h:56m left
#8629	Tests passed: 16070, ignored: 73, muted: 2	No artifacts	Changes (6)	one hour ago (4h:34m)
IntegrationBuild (MySQL/Docker)			Pending (8)	1 queued · Run · ...
#1394	Tests passed: 3582, ignored: 14, muted: 2; Running '[TeamCity] Server tests'	No artifacts	Changes (4)	2h:57m left
#1393	Tests passed: 12516, ignored: 68, muted: 2; Running '[TeamCity] Web tests'	No artifacts	Changes (5)	1h:29m left
#1392	Tests passed: 15784, ignored: 104, muted: 2; Running '[TeamCity] Flaky Test Detector Tests'	No artifacts	Changes (6)	2m:47s left
#1391	Tests passed: 16034, ignored: 104, muted: 2	No artifacts	Changes (6)	33 minutes ago (3h:26m)
IntegrationBuild (Oracle)			Pending (8)	1 queued · Run · ...
#8254	Tests passed: 3159, ignored: 12, muted: 1; Running '[TeamCity] Server tests'	No artifacts	Changes (20)	4h:40m left
#8253	Tests failed: 1, passed: 15564, ignored: 65, muted: 1; Running '[TeamCity] FileContentReplacer tests...	No artifacts	Changes (6)	28m:13s
#8252	Execution timeout (new); tests failed: 2 (2 new), passed: 14413, ignored: 56, muted: 1; number of	No artifacts	Changes (2)	one hour ago (6h:42m)

https://www.jetbrains.com/teamcity/

what is / who is DevOps?

> „**DevOps** is a set of practices that combines software development (Dev) and information-technology operations (Ops) which aims to shorten the systems development life cycle and provide continuous delivery with high software quality.”

https://en.wikipedia.org/wiki/DevOps

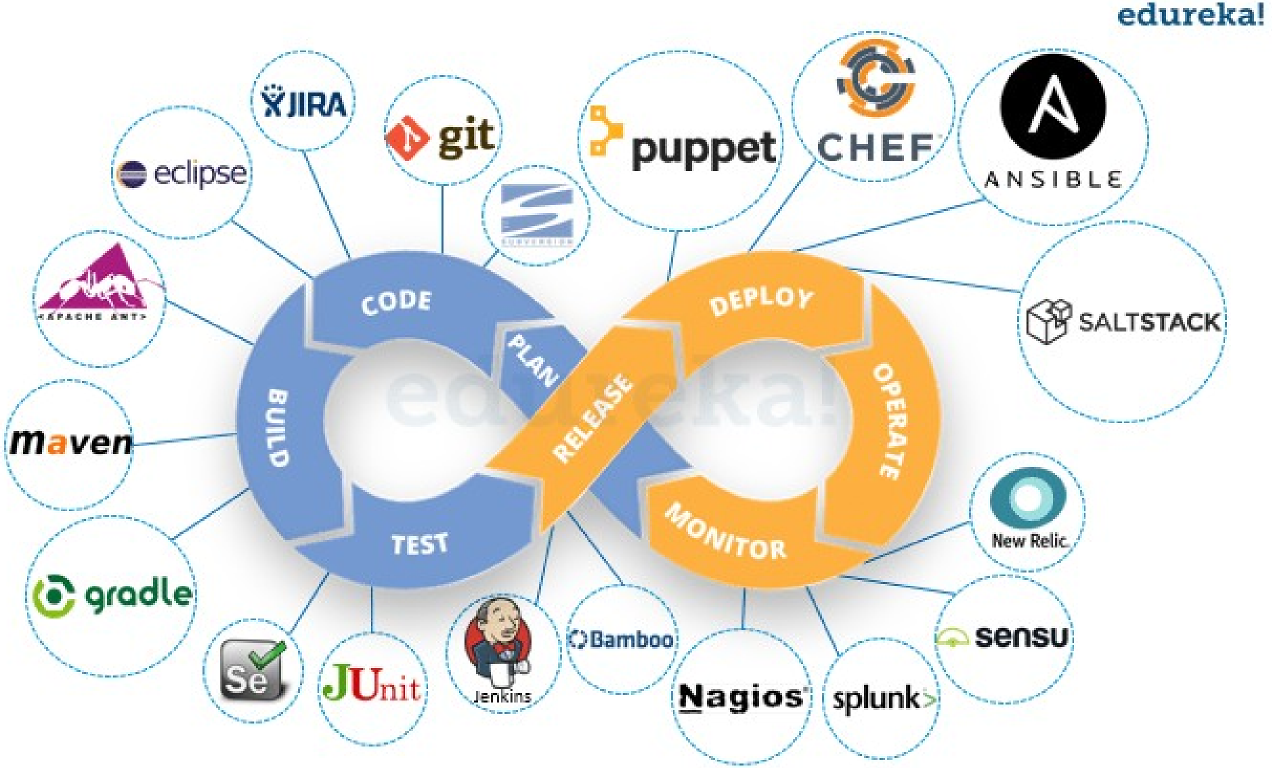
> „Making people happier while making apps better.”

https://www.jeffgeerling.com/blog/2019/real-world-devops

> ”A man sitting on explosive grenade”

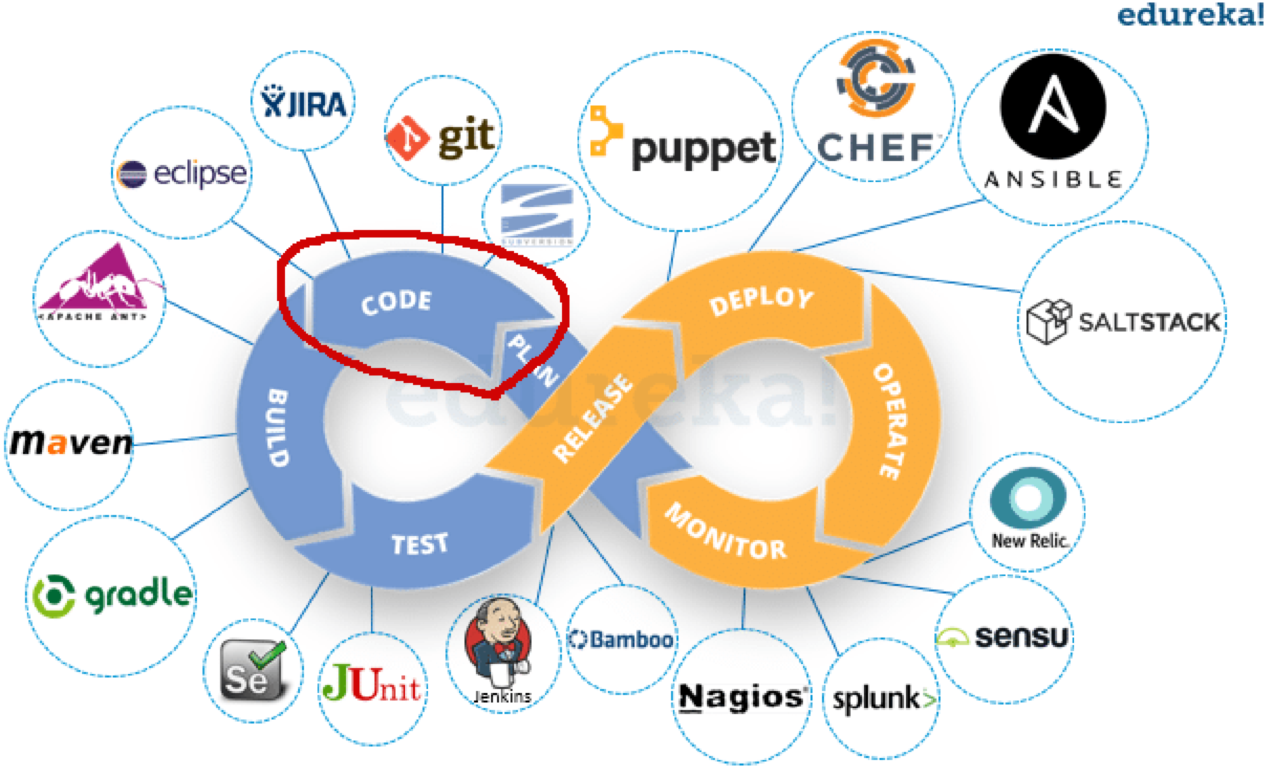
me ;)

from the beginning...



https://www.edureka.co/blog/devops-lifecycle/

from the beginning...



https://www.edureka.co/blog/devops-lifecycle/

what is source control?

VCS (version control system) - is used to track information about changes on indicated files. It provides such options as restoring changes, adding new modifications, or checking what changes have been made recently.

It allows you to revert selected files back to a previous state what also means easily recovering if needed.

In theory – calm and order ;)

types of vcs (version control systems)

- **Local** (VCS) - they only save data on the local computer.
- **Centralized** (CVCS) – main assumption is only one central repository (client – server architecture) which provides access to many clients. Requires continuous synchronization with the central server - offline work is impossible. Can be more efficient when working with large files – binaries.
- **Distributed** (DVCS) - the opposite of the centralized model. Each user has a local copy of the repository – central server fault tolerance. Remote work possible – only 2 commands require connection (push and pull).

types of vcs (version control systems)

Local:

- RCS,
- SCCS.

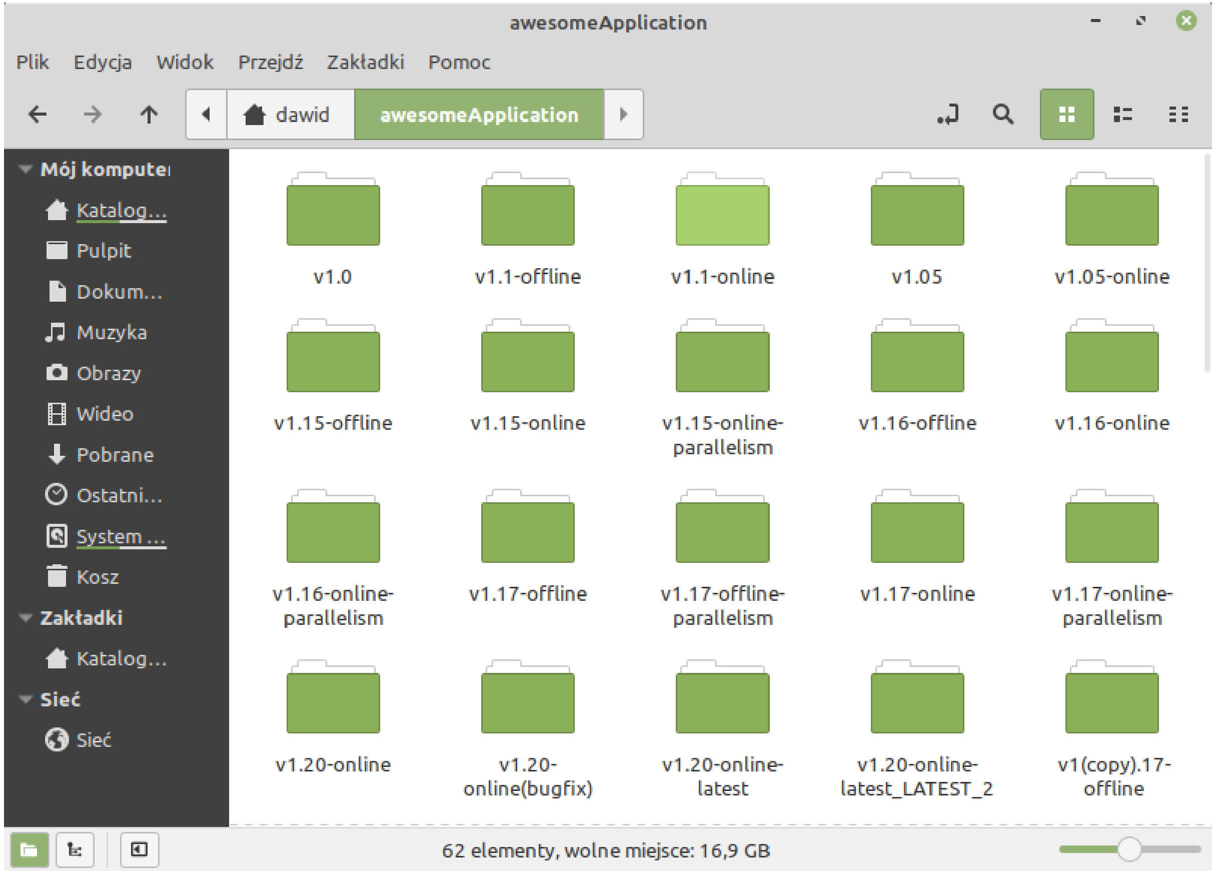
Centralized:

- SVN (Subversion),
- CVS.

Distributed:

- Git,
- Mercurial/HG,
- Fossil,
- GNU Bazaar.

alternatives (to vcs)?



v1.0
v1.1-offline
v1.1-online
v1.05
v1.05-online
v1.15-offline
v1.15-online
v1.15-online-parallelism
v1.16-offline
v1.16-online
v1.16-online-parallelism
v1.17-offline
v1.17-offline-parallelism
v1.17-online
v1.17-online-parallelism
v1.20-online
v1.20-online(bugfix)
v1.20-online-latest
v1.20-online-latest_LATEST_2
v1(copy).17-offline

62 elementy, wolne miejsce: 16,9 GB

alternatives (to vcs)?

https://unsplash.com/photos/9Eid2zc_Veo

for whom?

- Developer (mainly application source code),
- SysAdmin (simple scripts and config management),
- QA engineer / „normal” engineer,
- Scrum master (better understanding the project and current status),
- Project manager / project owner (those more aware)…

for whom?

- Developer (mainly application source code),
- SysAdmin (simple scripts and config management),
- QA engineer / „normal” engineer,
- Scrum master (better understanding the project and current status),
- Project manager / project owner (those more aware),
- **Each** IT junior / intern!

why git?

- Created by Linus Torvalds for Linux kernel development (in 2005).
- The most popular of the version control systems, using also by „Big Three” (GitHub, Bitbucket and GitLab).
- Released under GPL license (GNU General Public License).
- Big community - many guides, publications and support groups.
- The ability to work in two modes – local and remote.

advantages / disadvantages of git

Advantages:

- most operations are performed locally (we can work local),
- quite good performance (because offline).
- allows to create, organize and edit commits before send to the server.

Disadvantages:

- small performance for storing large binary files,
- no intuitive commits numbering (only SHA1),
- need to download the entire repository (always).

don't be like „typical” programmer

BRAGS ABOUT USING GIT IN TERMINAL
GIT PULL - GIT ADD . - GIT COMMIT - GIT PUSH
memegenerator.net

https://memegenerator.net/instance/65271689/scumbag-steve-brags-about-using-git-in-terminal-git-pull-git-add-git-commit-git-push

2. Time for practice

getting started

```
$ git version       # simple version check

$ git init          # create a new repository in the current directory

$ git clone https://github.com/torvalds/linux.git .
# clone the indicated repository to the current directory

$ git config user.name "Dawid Sadecki" # username setting

$ git config user.email "dawid.sadecki@protonmail.com" # user’s email setting
```

minimal glossary

repository	directory for all of project’s files (including revision history, refs and config)
commit	single point in the Git history (is represented as set of interrelated commits)
branch	active line of development, one repository can track any number of branches
tip	last (newest) commit in the branch
parent	previous commit, one commit can have many parents if it’s result of a merge
head	named reference to the commit at the tip of a branch
HEAD / @	commit being the source of current content of working copy (current commit)

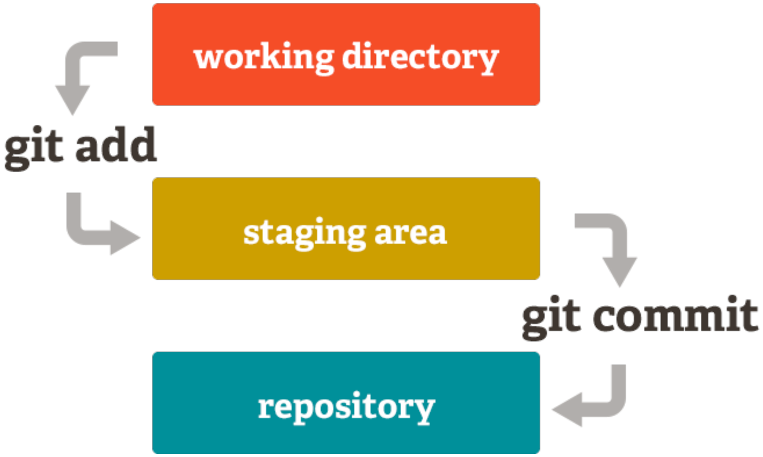
https://git-scm.com/docs/gitglossary

first commits

`$ git add fileName`	start tracking a new file / adding changes to commit index
`$ git commit`	saving changes (from index) to the repository
`$ git commit fileName`	commit indicated files without using the index
`$ git commit -m "message"`	commit without text editor opening
`$ git commit --allow-empty`	ability to create empty commits without any content
`$ git status`	displaying the state of the working directory and the index

index / staging area / cache

Index (staging area) is an intermediate area where commits can be formatted and reviewed before completing the commit.



https://git-scm.com/about/staging-area

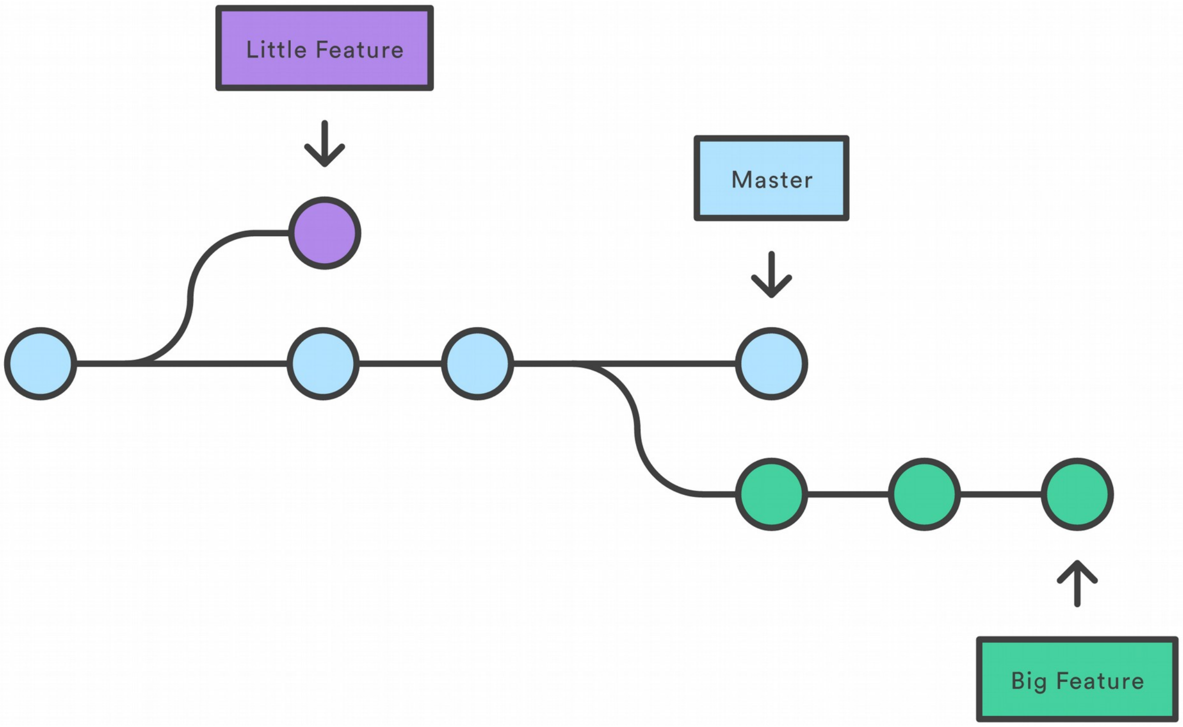
more details

$ git log	shows project history (commits)
$ git log --oneline	one-line commit summary
$ git log --graph	log with branches drawn
$ git show dab4d5f	shows indicated commit
$ git show HEAD	shows the current commit
$ git diff	differences between working directory and the index
$ git diff HEAD	differences between working directory and last commit

useful commands

$ git reset	clearing index content
$ git reset --hard	removes uncommited changes from index and also from disk!
$ git clean -n	shows the range of untracked files to delete
$ git clean -f	delete untracked files
$ git add -i	run in interactive mode
$ git add -p	run in path mode
$ git checkout dab4d5f	placing state of the given commit in the working directory

few words about branches



few words about branches

$ git branch	displaying existing branches
$ git branch -a	displaying all branches (also remotes)
$ git branch -vv	shows additional information

$ git branch _branch	new branch creation
$ git checkout -b new_branch	new branch creation and switch to it

$ git checkout branch_name	switching working directory to indicated branch
$ git checkout -	return to previous branch / commit

$ git branch -d branch_name	removal indicated branch

Many thanks for today's presence