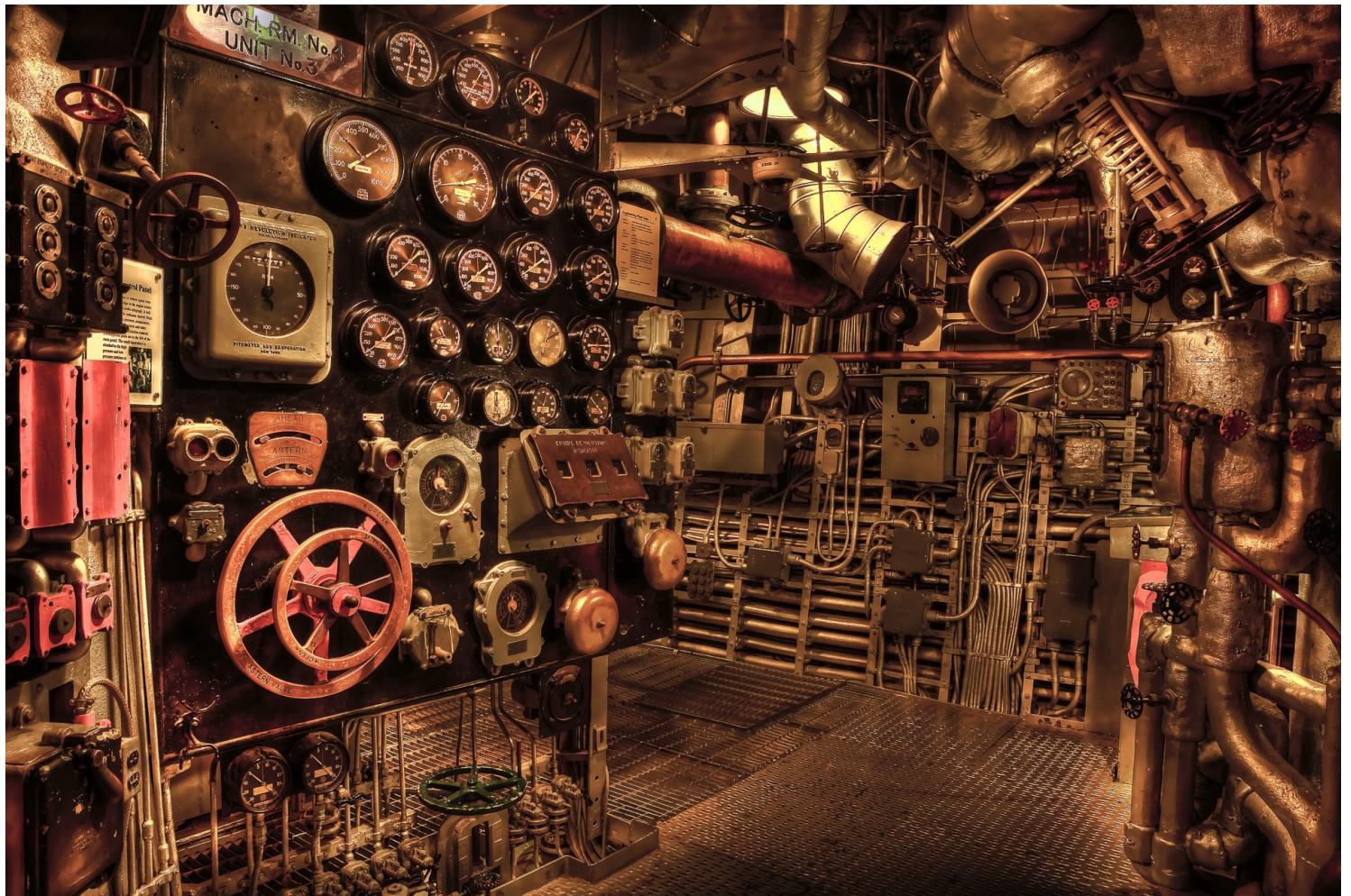


Diego Garber [Follow](#)

System Engineer at OLX, passionate about technologies, sustainability and sports.

Jan 13 · 4 min read



Automation is not enough, start automating automation

It all started on my holiday break, since I was in charge of creating deployment pipelines for new components, I was pretty used to the routine, I even had it documented and all:

- Copy one example YAML spec file and adapt it for the new component
- Add the new component to my [Ansible](#) inventory

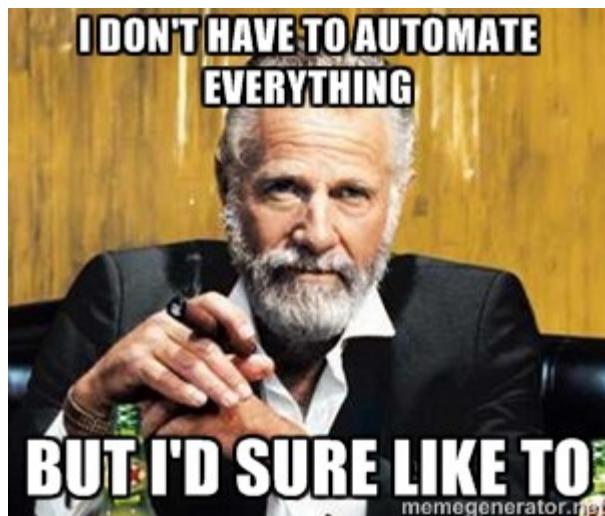
- Copy & paste init scripts from another similar component
- Create a new Jenkins job (also copy & paste it from a similar component)
- Add a new line to my jenkins config-package-creation job
- Commit & Push
- Test, fail, correct and fail a few more times until done!

Many of you might be familiarized with some of this steps, some of us use *Jenkins*, others *Bamboo*; for scripting some use *Ansible*, others *Salt*, *bash* or whatever, but we **ALL** copy & paste things from well working examples and re-adapt them accordingly.

And I think, I speak for all of us, when I say that once you start copy-pasting too much, you start feeling an itch on some anti-pattern gland of our geeky body.

The thing is, I might have ignored or overlooked this itch before my break, since all these steps never took me more than a day or two to get my new pipeline up and running, and it was a fair trade off since we didn't get as many new components requests to consider this a problem.

. . .



BUT, I left off, for my family holidays, pretty confident everything was going to be A-OK!. Well, kind of.. on my return I found myself with a couple of criticisms from my peers telling me that the whole process was too complicated, too many steps, etc... I have to admit I didn't take it very well at first, I mean, what was soooo complicated? but then it hit me, someone else might not share the same background, have the same context and no matter how clear the steps were documented, It was a pain! More so in an increasingly larger and complex infrastructure. Now I found myself with the opportunity to improve something, and rather than throwing more documentation at it, doing it the Devops way: Automating!.

. . .

Getting to work

So the first thing to do was getting rid of all that copy+pasting stuff, and the best way I know is refactoring, I won't bore you with all the details of my own implementation, but basically once you manually perform a task more than 5, 6 times you can easily pinpoint the repetitive stuff, and then you consolidate it in a single place. Ansible is very good at this, so good that I could even remove most of my group_vars files.

Both the refactoring part and the re-engineering were facilitated thanks to one simple but essential concept for automation: *Convention!* and that's why my new motto is always **Convention over Configuration.**

It took me some time but I finally got to a stage in my company where everybody is convinced that getting too creative is very nice for advertising people, but for us it should almost be banned! OK! ok, not that much, but conventions save a lot of time and give us a framework where we can feel safe that following some simple rules, things will work, and without so much effort.

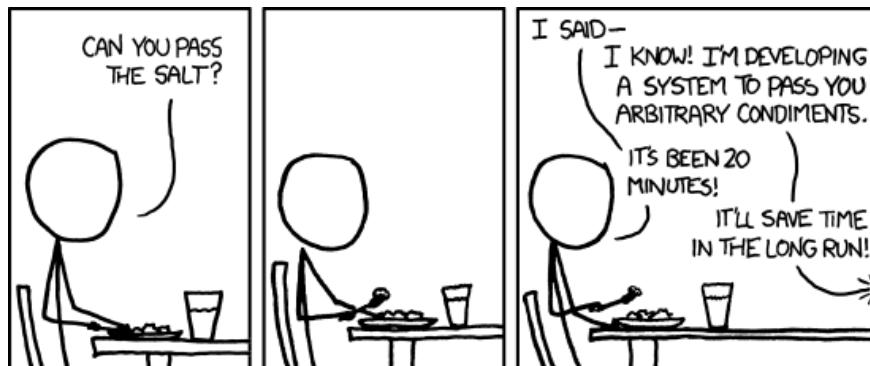
If we could guarantee that the same component name will apply for:

- Code Repository
- Config Repository

- Instance name and tags (AWS)
- DEB Package name
- Jenkins jobs

We could save lot of time and lines of code.

So I did, and never looked back ever since!!



Courtesy by xkcd

The new “automated automation” script, is basically a Jenkins job that commits a couple of lines to a couple of repos and hacks the very same Jenkins it’s running on, since it needs to add a new commit hook to another job, *it's kinda nasty but it works!*

Another approach we took for creating the Building part of the Pipeline was using a Jenkins plugin called “Job Generator”, which is very useful when you need to replicate a model job, we use it as a **job template pipeline** and saves us a lot of effort.

This is just one example, but it illustrates another big concept, **Automation is the new Documentation**, at least for us DevOps. It's always funner to do and easier to execute: a Jenkins job than a Documentation page. From backups, file encryptions, cache flushes, or even DNS configurations... you name it!

• • •

Concluding

In the age of the big microservices hype and cloud-based infrastructures this kind of automation becomes more and more important, it's no longer acceptable to have a big overhead for new services deployment. When you work in a company where you can have dozens (our case) or even hundreds (like Netflix, Airbnb, etc..) of services, doing things manually simply does NOT scale anymore, even the creation of automated processes should be laid in the hands of automation.

• • •

Did you like what you just read?

Recommend this story (by pressing the ❤️button) so other people can read it!

You can also react on Twitter: [@diegarber](#) or [@OlxTech](#)

