

# Automating the automation

By Stacy Naboikina



Passion for **Technology**

**SRSLY???**



# (CI)CD

(Continuous integration/Continuous Delivery)



---

# CI is...

---

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

---

# CD is...

---

**Continuous delivery (CD or CDE)** is a [software engineering](#) approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, doing so manually.<sup>[1][2]</sup> It aims at building, testing, and releasing software with greater speed and frequency. The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production. A straightforward and repeatable deployment process is important for continuous delivery.

---

# Our goal

- 
- Run unit tests
  - Build UI
  - Deploy app
  - Run automated tests

---

# Our goal

- 
- Run unit tests
  - Build UI
  - Deploy app
  - Run automated tests

ONE BUTTON CLICK!

# Our goal

- 
- Run unit tests ➤ Pass
  - Build UI ➤ Pass
  - Deploy app ➤ Fail
  - Run automated tests ➤ No run

ONE BUTTON CLICK!





Meet Jenkins!



## Meet Jenkins!

- Runs jobs for you
- Reports build results
- Works 24/7

---

# Usual way

---

Create jobs for each stage

---

# Usual way

---

Create jobs for each stage

OR:

Create jobs for each stage and make them trigger each other

# Usual way

---

---

Create jobs for each stage

OR:

Create jobs for each stage and make them trigger each other

OR:

Create a pipeline job that will run unit tests, build the app, run the automated tests

# Usual way

---

Create jobs for each stage

OR:

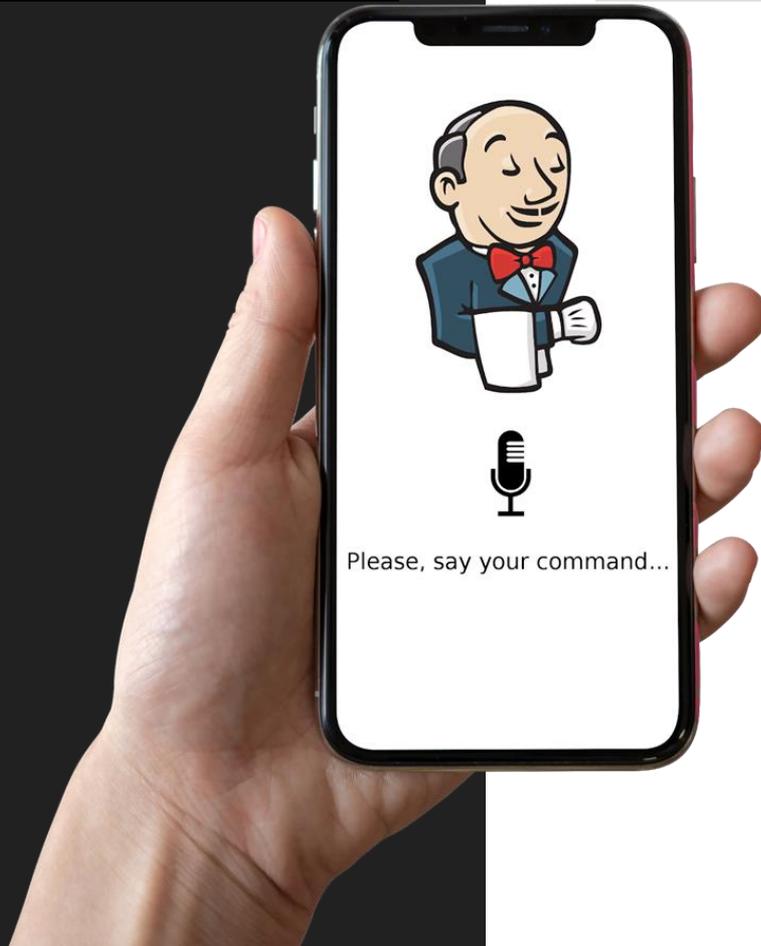
Create jobs for each stage and make them trigger each other

OR:

Create a pipeline job that will run unit tests, build the app, run the automated tests

OR:

Create a job for each stage (run unit tests, build the app etc.) and combine them into pipeline.



Ok, Jenkins...

- Please run the pipeline to:
  - run unittests
  - build UI
  - deploy app
  - run automated tests

## Annoying part

Creating and maintaining jobs takes time and can be very annoying...





# **Demo #1 - jenkins usual config**

---

# Automate it!

---

Automate creating jobs instead of clicking it through manually

---

# Technologies

- 
- Groovy
  - DSL Plugin
  - Jenkins pipelines

---

# Technologies

- 
- Groovy
  - DSL Plugin
  - Jenkins pipelines

--

Additionally:

- Credentials plugin
- Git plugin
- Pip, python3, npm, vue.js, virtualenv

---

# General approach

- 
1. Create ONLY ONE job using DSL plugin.
  2. Retrieve groovy script from git using Jenkins job.
  3. Run groovy script file.

Result: generated jobs



**Demo #2 - let's do it!**

---

# Where to start

- 
1. Collect information about your product release lifecycle (from app to testing)
  2. If any manual steps are still needed, make them executable from command line (or through Jenkins plugins)
  3. Find a clean environment
  4. Run each step one-by-one and fix any failures
  5. Put it into groovy and commit to repo
  6. Configure Jenkins env variables, credentials etc
  7. Create DSL job and configure it
  8. Run the job to generate jobs

---

# Bottlenecks & challenges

- 
- Generating too much jobs :D
  - Pipelines and DSLs use groovy, but are different
  - Including pipeline definitions into DSLs - syntax can be tricky
  - Not all information can be found in the official documentation
  - In progress scripts approval issue

---

# Useful links

- 
- General documentation about DSLs:

<https://jenkinsci.github.io/job-dsl-plugin/>

<https://github.com/jenkinsci/job-dsl-plugin>

- [ALM plugin configuration](#)

---

# Link to repository

---

<https://bitbucket.org/stacyengineering/>

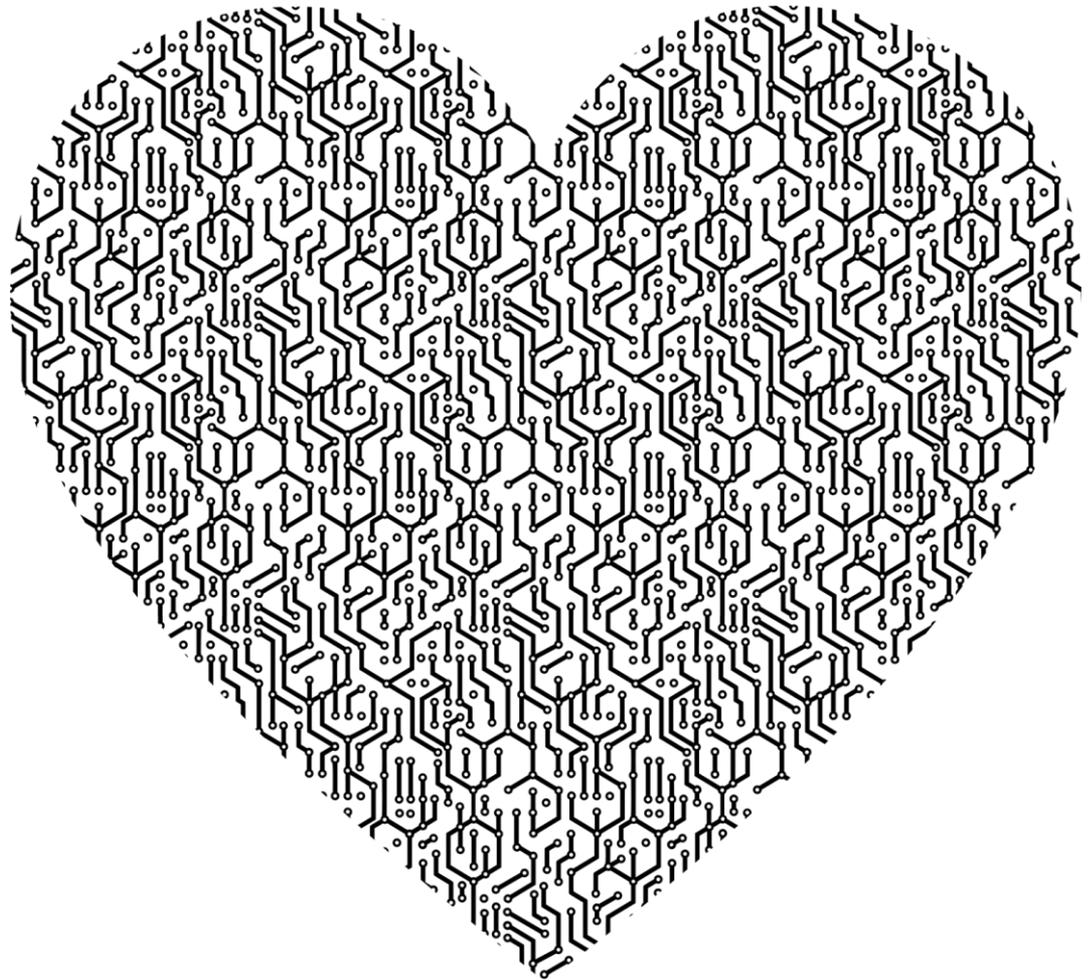
---

# Contact me

- 
- <https://www.facebook.com/people/Stacy-Naboikina>
  - [elderes11@gmail.com](mailto:elderes11@gmail.com)



**Thank you!**





**Q&A**