Advanced CI/CD Workshop

Part 1: Jenkins jobs DSL

Nir Koren
DevOps CI/CD Team Leader, SAP Gigya Israel

Goal of part 1

Each one of you will understand:

How to setup Jenkins (basic)

Install plugins (DSL)

Know the DSL API

Connect to Github Repo

Agenda

09:00 – 10:30: Advanced Jenkins – Nir Koren

10:30 – 11:00: Coffee Break 

11:00 – 12:30: Progressive Delivery with Argo Rollouts – Anton Weiss

12:30 – 13:30: Lunch 

13:30 – 15:00: Integration Testing with Docker and Testcontainers – Kevin Wittek

15:00 – 15:30: Coffee Break 

15:30 – 17:00: Expert Panel – Nir Koren, Anton Weiss, Kevin Wittek

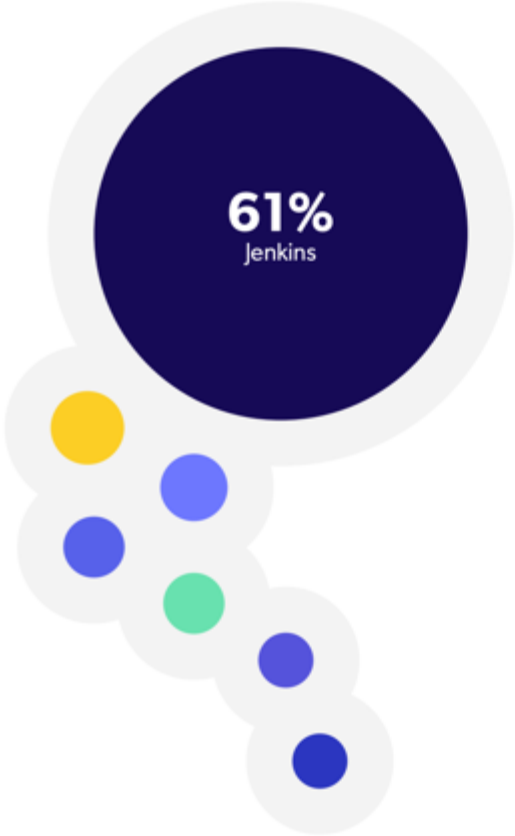
$ whoami

Nir Koren, DevOps CI/CD Team Leader, SAP Gigya Israel
Doing and implementing DevOps and CI / CD for 10+ years

https://il.linkedin.com/in/nirkoren

Introduction to
Jenkins CI Server

Which CI/CD technologies are you using?



Jenkins **61%**

Bamboo **12%**

Travis CI **11%**

N/A **10%**

Other **10%**

TeamCity **9%**

Circle CI **9%**

Let’s install Jenkins first

- Download Jenkins (Preferred LTS) https://jenkins.io/
- Open up a terminal in the download directory.
- Run java -jar jenkins.war --httpPort=9090.
- Browse to http://localhost:9090.
- Follow the instructions to complete the installation.

Docker?

$ docker run -d -p 9090:8080 -u root -v /Users/nirk/jenkins_home:/var/jenkins_home jenkins/jenkins:lts

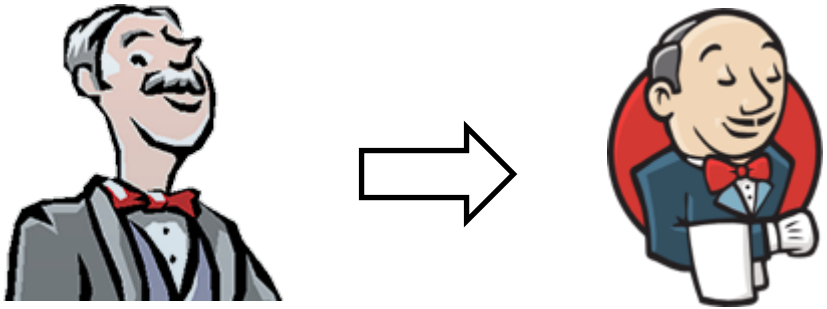
What is Jenkins?

Jenkins is **open source automation server** which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

History of Jenkins

- Funded by **Kohsuke Kawaguchi** as Hudson CI (in that time he was a developer in Sun Microsystems).
- First Hudson release was in 2004.
- Oracle acquired Sun in 2010 and claimed the right to the ”**Hudson**” name and applied for a trademark in December 2010.
- Kohsuke forked Hudson open source and created the name “**Jenkins”** – the rest is history.



JOB DSL

- DSL stands for **Domain Specific Language**
- You can describe your jobs in Jenkins using a **Groovy** Based Language.
- Groovy-- It’s similar to Java but simpler because it’s much more dynamic. It's Scripting Language based in the JDK.

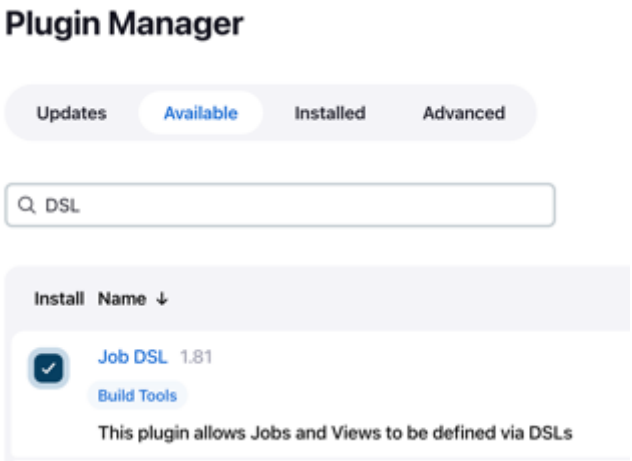
Jenkins job DSL plugin was designed to make it easier to manage jobs.

- If you don’t have a lot of jobs, using the UI is the easiest way.
- When the number of jobs grows, maintaining becomes difficult and a lot of work.

JOB DSL – How to?

Install the plugin:

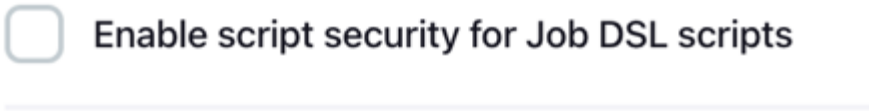
Manage Jenkins > Manage Plugins > “Available” Tab



JOB DSL – How to?

For this workshop

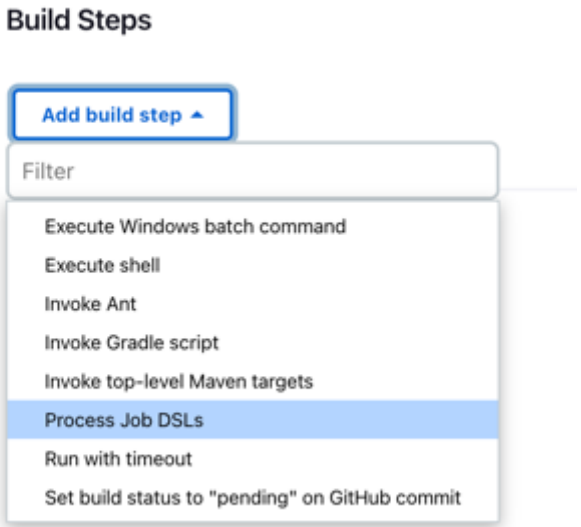
Manage Jenkins > Configure Global Security > “Uncheck”



Job Configuration

Freestyle Job

- Create new Freestyle project



Job Configuration

Code your DSL using the API viewer

http://localhost:9090/plugin/job-dsl/api-viewer/index.html

HANDS-ON AGENDA

- Setup Jenkins
- Install the plugin
- Know the API documentation
- Write the first code that creates pipeline
- Connect it to Github
- Add Datasource code to generate it