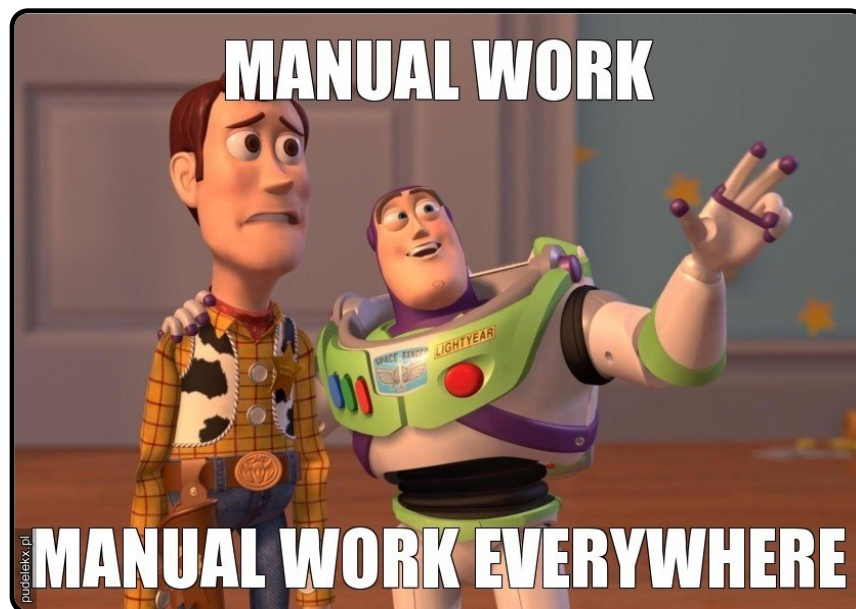


```
FROM openjdk:8-jdk-stretch RUN apt-get update && apt-get install -y git curl && rm -rf /var/lib/apt/lists/* ARG user=jenkins ARG group=jenkins ARG uid=1000 ARG gid=1000 ARG http_port=8080 ARG agent_port=50000 ARG JENKINS_HOME=/var/jenkins_home ENV JENKINS_HOME $JENKINS_HOME ENV JENKINS_SLAVE_AGENT_P
ORT $(agent_port) RUN mkdir -p $JENKINS_HOME \ && chown $(uid):$(gid) $JENKINS_HOME \ && groupadd -g $(gid) $(group) \ && useradd -d "$JENKINS_HOME" -
u $(uid) -g $(gid) -m -s /bin/bash $(user) VOLUME $JENKINS_HOME RUN mkdir -p /usr/share/jenkins/ref/init.groovy.d ARG TINI_VERSION=v0.16.1 COPY tini pu
b.gpg $(JENKINS_HOME)/tini.pub.gpg RUN curl -fsSL https://github.com/krallin/tini/releases/download/$(TINI_VERSION)/tini-static-$(dpkg --print-architec
ture) -o /sbin/tini \ && curl -fsSL https://github.com/krallin/tini/releases/download/$(TINI_VERSION)/tini-static-$(dpkg --print-architecture).asc -
o /sbin/tini.asc \ && gpg --no-tty --import $(JENKINS_HOME)/tini.pub.gpg \ && gpg --verify /sbin/tini.asc \ && rm -rf /sbin/tini.asc /root/.gnupg \ &&
chmod +x /sbin/tini COPY init.groovy /usr/share/jenkins/ref/init.groovy.d/tcp-slave-agent-port.groovy ARG JENKINS_VERSION ENV JENKINS_VERSION $(JENK
INS_VERSION:-2.121.1) ARG JENKINS_SHA=5bb075b81a392ceada4e960049e37df5f15a1e3cfc9dc24d749858e70b48919 ARG JENKINS_URL=https://repo.jenkins-ci.org/publi
c/org/jenkins-ci/main/jenkins-war/$(JENKINS_VERSION)/jenkins-war.$(JENKINS_VERSION).war RUN curl -fsSL $(JENKINS_URL) -o /usr/share/jenkins/jenkins.w
ar \ && echo "$(JENKINS_SHA) /usr/share/jenkins/jenkins.war" | sha256sum -c - ENV JENKINS_UC https://updates.jenkins.io ENV JENKINS_UC EXPERIMENTAL=https://updates.jenkins.io/experimental ENV JENKINS_INCREMENTALS_REPO_MIRROR=https://repo.jenkins-ci.org/incrementals RUN chown -R $(user) "$JENKINS_HOME" /usr/share/jenkins/ref EXPOSE $(http_port) ENV COPY_REFERENCE_FILE_LOG $JENKINS_HOME/copy-reference-file.log USER $(user) COPY jenkins-support /usr/local/bin/jenkins-support COPY jenkins.sh /usr/local/bin/jenkins.sh COPY tini-shim.sh /bin/tini ENTRYPOINT ["$/sbin/tini", "-", "/usr/local/bin/jenkins.sh"] COPY plugins.sh /usr/local/bin/plugins.sh COPY install-plugins.sh /usr/local/bin/install-plugins.shFROM openjdk:8-jdk-stretch RUN apt-get update && apt-get install -y git curl && rm -rf /var/lib/apt/lists/* ARG user=jenkins ARG group=jenkins ARG uid=1000 ARG gid=1000 ARG http_port=8080 ARG agent_port=50000 ARG JENKINS_HOME=/var/jenkins_home ENV JENKINS_SLAVE_AGENT_PORT $(agent_port) RUN mkdir -p $JENKINS_HOME \ && chown $(uid):$(gid) $JENKINS_HOME \ && groupadd -g $(gid) $(group) \ && useradd -d "$JENKINS_HOME" -u $(uid) -g $(gid) -m -s /bin/bash $(user) VOLUME $JENKINS_HOME RUN mkdir -p /usr/share/jenkins/ref/init.groovy.d ARG TINI_VERSION=v0.16.1 COPY tini.pub.gpg $(JENKINS_HOME)/tini.pub.gpg RUN curl -fsSL https://github.com/krallin/tini/releases/download/$(TINI_VERSION)/tini-static-$(dpkg --print-architecture) -o /sbin/tini \ && curl -fsSL https://github.com/krallin/tini/releases/download/$(TINI_VERSION)/tini-static-$(dpkg --print-architecture).asc -o /sbin/tini.asc \ && gpg --no-tty --import $(JENKINS_HOME)/tini.pub.gpg \ && gpg --verify /sbin/tini.asc \ && rm -rf /sbin/tini.asc /root/.gnupg \ && chmod +x /sbin/tini COPY init.groovy /usr/share/jenkins/ref/init.groovy.d/tcp-slave-agent-port.groovy ARG JENKINS_VERSION ENV JENKINS_VERSION $(JENKINS_VERSION:-2.121.1) ARG JENKINS_SHA=5bb075b81a392ceada4e960049e37df5f15a1e3cfc9dc24d749858e70b48919 ARG JENKINS_URL=https://repo.jenkins-ci.org/public/org/jenkins-ci/main/jenkins-war/$(JENKINS_VERSION)/jenkins-war.$(JENKINS_VERSION).war RUN curl -fsSL $(JENKINS_URL) -o /usr/share/jenkins/jenkins.war \ && echo "$(JENKINS_SHA) /usr/share/jenkins/jenkins.war" | sha256sum -c - ENV JENKINS_UC https://updates.jenkins.io ENV JENKINS_UC EXPERIMENTAL=https://updates.jenkins.io/experimental ENV JENKINS_INCREMENTALS_REPO_MIRROR=https://repo.jenkins-ci.org/incrementals RUN chown -R $(user) "$JENKINS_HOME" /usr/share/jenkins/ref EXPOSE $(http port) ENV COPY_REFERENCE_FILE_LOG $JENKINS_HOME/copy-reference-file.log USER $(user) COPY jenkins-support /usr/local/bin/jenkins-support COPY jenkins.sh /usr/local/bin/jenkins.sh COPY tini-shim.sh /bin/tini ENTRYPOINT ["$/sbin/tini", "-", "/usr/local/bin/jenkins.sh"] COPY plugins.sh /usr/local/bin/plugins.sh COPY install-plugins.sh /usr/local/bin/install-plugins.shFROM openjdk:8-jdk-stretch RUN apt-get update && apt-get install -y git curl && rm -rf /var/lib/apt/lists/* ARG user=jenkins ARG group=jenkins ARG uid=1000 ARG gid=1000 ARG http_port=8080 ARG agent_port=50000 ARG JENKINS_HOME=/var/jenkins_home ENV JENKINS_SLAVE_AGENT_PORT $(agent port) RUN mkdir -p $JENKINS_HOME \ && chown $(uid):$(gid) $JENKINS_HOME \ && groupadd -g $(gid) $(group) \ && useradd -d "$JENKINS_HOME" -u $(uid) -g $(gid) -m -s /bin/bash $(user) VOLUME $JENKINS_HOME RUN mkdir -p /usr/share/jenkins/ref/init.groovy.d ARG TINI_VERSION=v0.16.1 COPY tini.pub.gpg $(JENKINS_HOME)/tini.pub.gpg RUN curl -fsSL https://github.com/krallin/tini/releases/download/$(TINI_VERSION)/tini-static-$(dpkg --print-architecture) -o /sbin/tini \ && curl -fsSL https://github.com/krallin/tini/releases/download/$(TINI_VERSION)/tini-static-$(dpkg --print-architecture).asc -o /sbin/tini.asc \ && gpg --no-tty --import $(JENKINS_HOME)/tini.pub.gpg \ && gpg --verify /sbin/tini.asc \ && rm -rf /sbin/tini.asc /root/.gnupg \ && chmod +x /sbin/tini COPY init.groovy /usr/share/jenkins/ref/init.groovy.d/tcp-slave-agent-port.groovy ARG JENKINS_VERSION ENV JENKINS_VERSION $(JENKINS_VERSION:-2.121.1) ARG JENKINS_SHA=5bb075b81a392ceada4e960049e37df5f15a1e3cfc9dc24d749858e70b48919 ARG JENKINS_URL=https://repo.jenkins-ci.org/public/org/jenkins-ci/main/jenkins-war/$(JENKINS_VERSION)/jenkins-war.$(JENKINS_VERSION).war RUN curl -fsSL $(JENKINS_URL) -o /usr/share/jenkins/jenkins.war \ && echo "$(JENKINS_SHA) /usr/share/jenkins/jenkins.war" | sha256sum -c - ENV JENKINS_UC https://updates.jenkins.io ENV JENKINS_UC EXPERIMENTAL=https://updates.jenkins.io/experimental ENV JENKINS_INCREMENTALS_REPO_MIRROR=https://repo.jenkins-ci.org/incrementals RUN chown -R $(user) "$JENKINS_HOME" /usr/share/jenkins/ref EXPOSE $(http port) ENV COPY_REFERENCE_FILE_LOG $JENKINS_HOME/copy-reference-file.log USER $(user) COPY jenkins-support /usr/local/bin/jenkins-support COPY jenkins.sh /usr/local/bin/jenkins.sh COPY tini-shim.sh /bin/tini ENTRYPOINT ["$/sbin/tini", "-", "/usr/local/bin/jenkins.sh"] COPY plugins.sh /usr/local/bin/plugins.sh COPY install-plugins.sh /usr/local/bin/install-plugins.shFROM openjdk:8-jdk-stretch RUN apt-get update && apt-get install -y git curl && rm -rf /var/lib/apt/lists/* ARG user=jenkins ARG group=jenkins ARG uid=1000 ARG gid=1000 ARG http_port=8080 ARG agent_port=50000 ARG JENKINS_HOME=/var/jenkins_home ENV JENKINS_SLAVE_AGENT_PORT $(agent port) RUN mkdir -p $JENKINS_HOME \ && chown $(uid):$(gid) $JENKINS_HOME \ && groupadd -g $(gid) $(group) \ && useradd -d "$JENKINS_HOME" -u $(uid) -g $(gid) -m -s /bin/bash $(user) VOLUME $JENKINS_HOME RUN mkdir -p /usr/share/jenkins/ref/init.groovy.d ARG TINI_VERSION=v0.16.1 COPY tini.pub.gpg $(JENKINS_HOME)/tini.pub.gpg RUN curl -fsSL https://github.com/krallin/tini/
```

JAK PRZESTAĆ KLIKAĆ I ZACZAĆ PISAĆ! CAŁY JENKINS JAKO KOD

Jędrzej Andrykowski
 cinkciarz.pl

Jak
to najczęściej
jest?





1

Setup Jenkins

Instalacja

jenkins.io/doc/book/installing

Linux

Debian/Ubuntu

On Debian-based distributions, such as Ubuntu, you can install Jenkins through [apt](#).

Recent versions are available in [an apt repository](#). Older but stable LTS versions are in [this apt repository](#).

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins
```

This package installation will:

- Setup Jenkins as a daemon launched on start. See [/etc/init.d/jenkins](#) for more details.
- Create a 'jenkins' user to run this service.
- Direct console log output to the file [/var/log/jenkins/jenkins.log](#). Check this file if you are troubleshooting Jenkins.
- Populate [/etc/default/jenkins](#) with configuration parameters for the launch, e.g [JENKINS_HOME](#)
- Set Jenkins to listen on port 8080. Access this port with your browser to start configuration.



If your [/etc/init.d/jenkins](#) file fails to start Jenkins, edit the [/etc/default/jenkins](#) to replace the line `---- HTTP_PORT=8080----` with `----HTTP_PORT=8081----` Here, "8081" was chosen but you can put another port available.



2

Setup Jenkins

Post-Instalacja

Zaczynamy

Odblokuj Jenkinsa

Aby zapewnić, że Jenkins jest bezpiecznie uruchomiony przez administratora, hasło zostało zapisane do pliku logów ([nie masz pewności, gdzie go znaleźć?](#)) oraz w pliku na serwerze:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Skopiuuj hasło z jednej z powyższych lokalizacji i wklej poniżej.

Hasło administratorskie:



Kontynuuj



Dostosuj Jenkinsa

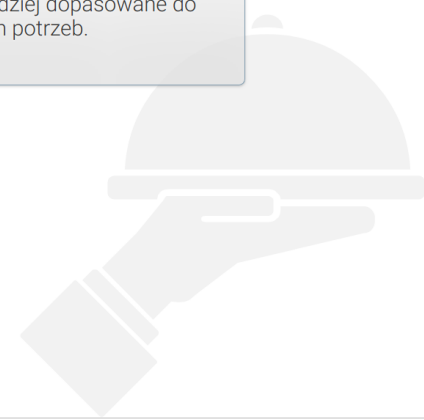
Wtyczki rozszerzające Jenkinsa o dodatkowe funkcjonalności, które zaspokajają wiele potrzeb.

Zainstaluj sugerowane wtyczki

Zainstaluj wtyczki, które społeczność Jenkinsa uznała za najbardziej przydatne.

Wybierz wtyczki do instalacji

Wybierz i zainstaluj wtyczki najbardziej dopasowane do Twoich potrzeb.



Instalacja wybranych wtyczek

Organization and Administration | Wszystkie | Żadne | Rekomer | Wybrane (20/57)

Zauważ, że poniżej nie jest wyświetlana pełna lista wtyczek. Dodatkowe wtyczki mogą być pobrane przez **Menedżera wtyczek**, kiedy konfiguracja Jenkinsa zostanie zakończona. [Sprawdź więcej informacji na Wiki](#).

Organization and Administration (2/3)

- Dashboard View** 10
Customizable dashboard that can present various views of job information.
- Folders**
This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.
- OWASP Markup Formatter** 9
Uses the [OWASP Java HTML Sanitizer](#) to allow safe-seeming HTML markup to be entered in project descriptions and the like.

Build Features (4/10)

- Build Name Setter** 11
This plug-in sets the display name of a build to something other than #1, #2, #3,...
- Build Timeout** 7
This plugin allows builds to be automatically terminated after the specified amount of time has elapsed.
- Config File Provider** 9
Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.
- Credentials Binding** 9
Allows credentials to be bound to environment variables for use from miscellaneous build steps.
- embeddable-build-status** 4
This plugin adds the embeddable build status badge to Jenkins so that you can easily hyperlink/show your build status from elsewhere.

Jenkins 2.150.3 Wróć Instaluj

Instalacja wtyczek

Instalacja wtyczek

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	⚙ Credentials Binding	Folders
⚙ Timestampers	⚙ Workspace Cleanup	⚙ Ant	⚙ Gradle	** JDK Tool
⚙ Pipeline	⚙ GitHub Branch Source	⚙ Pipeline: GitHub Groovy Libraries	⚙ Pipeline: Stage View	** Script Security
⚙ Git	⚙ Subversion	⚙ SSH Slaves	⚙ Matrix Authorization Strategy	** Command Agent Launcher
⚙ PAM Authentication	⚙ LDAP	⚙ Email Extension	⚙ Mailer	** Structs
				** Pipeline: Step API
				** bouncycastle API
				** SCM API
				** Pipeline: API
				** JUnit
				OWASP Markup Formatter
				** Token Macro
				Build Timeout
				** Credentials

** - zależności wymagane

Jenkins 2.150.3

Dodawanie pierwszego użytkownika

Stwórz pierwszego administratora

Login:	<input type="text" value="foo"/>
Hasło:	<input type="password" value="*****"/>
Powtórz hasło:	<input type="password" value="*****"/>
Pełna nazwa:	<input type="text" value="Foo Bar"/>
Adres e-mail:	<input type="text" value="foo@bar.com"/>

Jenkins 2.150.3

[Kontynuuj jako administrator](#)

[Zapisz i zakończ](#)

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Instalacja zakończona

Jenkins jest gotowy!

Konfiguracja Jenkinsa została zakończona.

Zaczynij korzystać z Jenkinsa

Jenkins 2.150.3

Instalacja zakończona

The screenshot shows the Jenkins web interface. At the top, a black navigation bar contains the Jenkins logo, a search bar with the text "Szukaj...", and links for "Foo Bar" and "Wyloguj". Below the navigation bar, a yellow banner displays "Instalacja zakończona". The main content area features a central welcome message "Witamy w Jenkinisie!" with a button to "Utwórz nowe zadanie". On the left, a sidebar lists various management options: "Nowy Projekt", "Użytkownicy", "Historia zadań", "Zarządzaj Jenkinsem", "Moje widoki", "Lockable Resources", "Credentials", and "New View". Below these are two expandable sections: "Kolejka zadań" (empty) and "Status wykonawców zadań" (showing two idle executors).

Jenkins 2.150.3



3 Setup Jenkins Konfiguracja

Jenkins

Foo Bar | Wyloguj

Jenkins >

- [Nowy Projekt](#)
- [Użytkownicy](#)
- [Historia zadań](#)
- Zarządzaj Jenkinsem**
- [Moje widoki](#)
- [Lockable Resources](#)
- [Credentials](#)
- [New View](#)

Kolejka zadań -

Nie ma zadań w kolejce

Status wykonawców zadań -

- 1 Bezczyanny
- 2 Bezczyanny

Zarządzaj Jenkinsem

Skonfiguruj system

Konfiguruj ustawienia globalne i ścieżki.

Konfiguruj ustawienia bezpieczeństwa

Zabezpiecz Jenkinsa: decyduj, kto ma do niego dostęp.

Configure Credentials

Configure the credential providers and types

Globalne narzędzia do konfiguracji

Konfiguruj narzędzia, ścieżki do nich i automatyczne instalatory

Odczytaj ponownie konfigurację z dysku

Porzuć wszystkie dane załadowane w pamięci i załaduj wszystko z systemu plików. Opcja użyteczna gdy zmodyfikowano pliki konfiguracyjne bezpośrednio z dysku.

Zarządzaj wtyczkami

Dodaj, usuń, wyłącz lub włącz wtyczki które mogą rozszerzyć funkcjonalność Jenkinsa.

Informacje o systemie

Wyświetla wiele środowiskowych informacji pomocnych przy rozwiązywaniu problemów.

Dziennik systemowy

Dziennik systemowy gromadzi wywołania `java.util.logging` powiązane z Jenkinsem.

Statystyki obciążenia

Sprawdź obciążenie zasobów systemowych i dowiedz się, czy nie potrzebujesz więcej maszyn do uruchamiania zadań.

Wiersz poleceń Jenkinsa

Zarządzaj Jenkinsem poziomu z wiersza poleceń lub systemu.

Konsola skryptów

Wykonuje dowolny skrypt pomocny do celów administracyjnych, usuwania usterek i diagnostyki.

8

Jenkins konfiguracja

[Nowy Projekt](#)
[Użytkownicy](#)
[Historia zadań](#)
[Zarządzaj Jenkinsem](#)
[Moje widoki](#)
[Lockable Resources](#)
[Credentials](#)
[New View](#)

Kolejka zadań
 Nie ma zadań w kolejce

Status wykonawców zadań
 1 Bezczynny
 2 Bezczynny

Katalog domowy
 Komunikat systemowy

of executors: 2
 Labels:
 Plan wykorzystania: Wykorzystuj ten węzeł tak bardzo, jak to tylko możliwe
 Quiet period: 5
 SCM checkout retry count: 0

Restrict project naming

Global properties
 Disable deferred wipeout on this node
 Lokalizacja narzędzi
 Zmienne środowiskowe

Pipeline Speed/Durability Settings
 Pipeline Default Speed/Durability Level: None: use pipeline default (MAX_SURVIVABILITY)

Usage Statistics
 Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project.

Timestamper
 System clock time format: 'HH:mm:ss'
 Elapsed time format: 'HH:mm:ss.S'
Enabled for all Pipeline builds

[Zapisz](#) [Zastosuj](#)

Jenkins Szukaj... Foo Bar | Wyloguj

Konfiguruj ustawienia bezpieczeństwa

Konfiguruj ustawienia bezpieczeństwa

Enable security
 Disable remember me

Access Control

Security Realm

- Delegate to servlet container
- LDAP
- Unix user/group database
- Własna baza danych Jenkinsa
 - Pozwól użytkownikom na rejestrowanie się

Authorization

- Każdy użytkownik może wszystko
- Legacy mode
- Logged-in users can do anything
- Matrix-based security

User/group	Overall	Credentials			Agent	Job				Run	View	Repozytorium kodu	Lockable Resources										
	Administer	Create	Delete	Update	Connect	Configure	Build	View	Cancel	Build	Configure	Create	Read	Move	Discover	Delete	Read	Tag	Reserve	Unlock			
Anonymous Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Add user or group...

- Project-based Matrix Authorization Strategy

Markup Formatter

Save Zastosuj

Jenkins **Globalne narzędzia do konfiguracji** Szukaj... Foo Bar | Wyloguj

Jenkins » Globalne narzędzia do konfiguracji

[Back to Dashboard](#)
[Manage Jenkins](#)

Globalne narzędzia do konfiguracji

Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

JDK

JDK instalacji: Dodaj

List of JDK installations on this system

Git

Git installations

Git
Name: Default
Path to Git executable: git
<input type="checkbox"/> Install automatically
Delete Git

Add Git

Gradle

Gradle instalacji: Dodaj

List of Gradle installations on this system

Ant

Ant instalacji: Dodaj

List of Ant installations on this system

Maven

Maven instalacji

[Save](#) [Apply](#)







4 Setup Jenkins

Konfiguracja zadań

- Nowy Projekt
- Użytkownicy
- Historia zadań
- Zarządzaj Jenkinsem
- Moje widoki
- Lockable Resources
- Credentials
- New View

[dodaj opis](#)

Witamy w Jenkinisie!

Utwórz **nowe zadanie**, aby rozpocząć pracę.

Kolejka zadań

Nie ma zadań w kolejce

Status wykonawców zadań

- 1 Bezczyanny
- 2 Bezczyanny

Podaj nazwę projektu

» Pole wymagane



Ogólny projekt

To jest podstawowa funkcja Jenkinsa. Jenkins stworzy projekt łączący dowolny SCM z dowolnym systemem budującym, może to być również wykorzystane do czegoś innego niż budowanie oprogramowania.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



GitHub Organization

Scans a GitHub organization (or user account) for all repositories matching some defined markers.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

General Repozytorium kodu Wyzwalacze zadania Środowisko do uruchomienia Budowanie Akcje po zadaniu

Opis

[Niesformatowany tekst] [Podgląd](#)

- GitHub project
- Porzuć stare zadania ?
- This build requires lockable resources
- Throttle builds ?
- To zadanie jest sparametryzowane ?
- Zablokuj zadania ?
- Wykonuj zadania współbieżnie, jeśli zajdzie potrzeba ?

Zaawansowane...

Repozytorium kodu

- Brak
- Git

Repositories

Repository URL ?

Credentials

Zaawansowane...

Jenkins Szukaj... Foo Bar | Wyloguj

Jenkins WŁĄCZ AUTOMATYCZNE ODŚWIEŻANIE

[Nowy Projekt](#) [dodaj opis](#)
[Użytkownicy](#)
[Historia zadań](#)
[Zarządzaj Jenkinsem](#)
[Moje widoki](#)
[Lockable Resources](#)
[Credentials](#)
[New View](#)

Wszystkie +						
S	P	Nazwa ↓	Ostatni sukces	Ostatni błąd	Czas trwania	
		example	—	—	nd.	

Ikona: [S](#) [M](#) [L](#)
[Legenda](#)
[RSS Dla wszystkich](#)
[RSS Tylko niepowodzenia](#)
[RSS Tylko dla najnowszych](#)

Kolejka zadań -

Nie ma zadań w kolejce

Status wykonawców zadań -

- Bezczynny
- Bezczynny





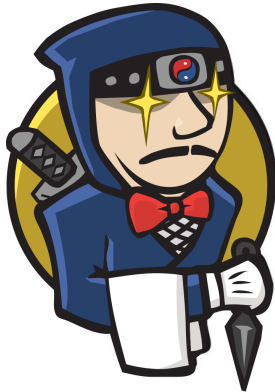


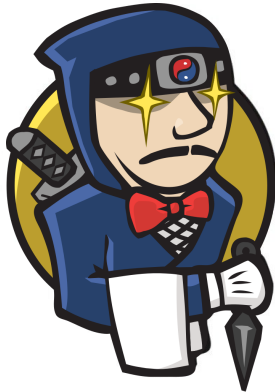








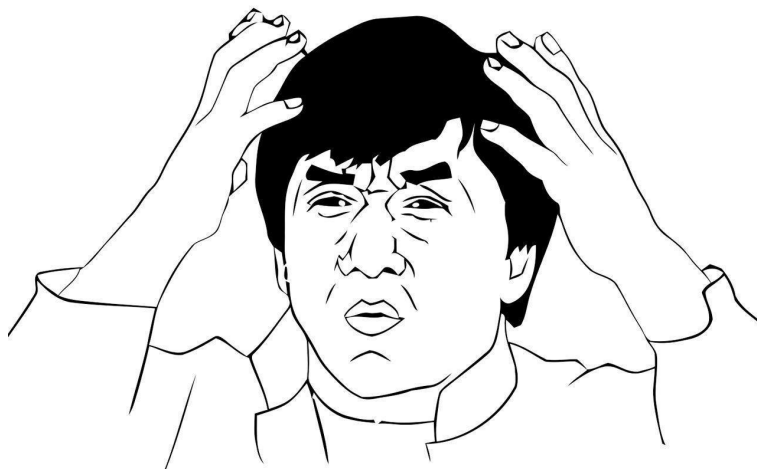






UP TO DATE DOCUMENTATION

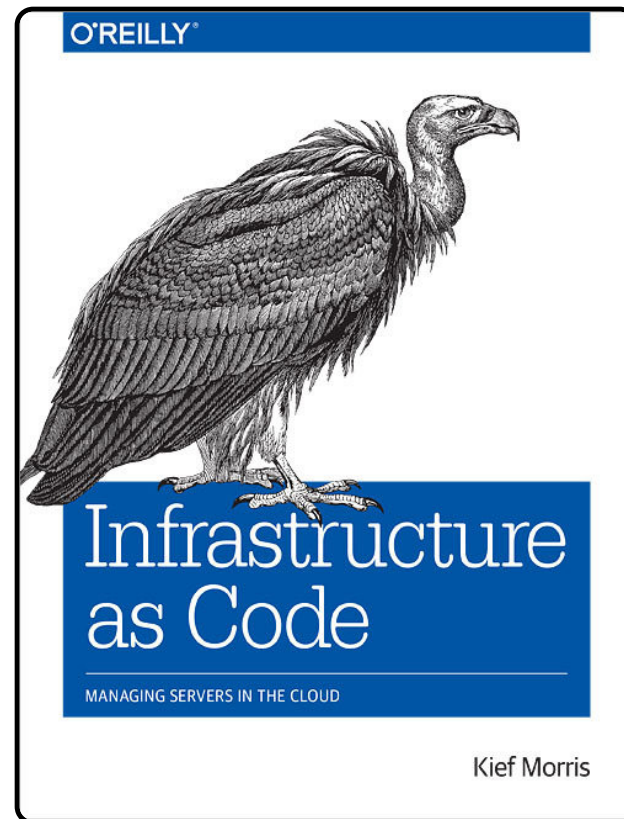
**NOW THERE IS SOMETHING I
HAVEN'T SEEN IN A LONG TIME.**



Co

można

zrobić?



Linux

Debian/Ubuntu

On Debian-based distributions, such as Ubuntu, you can install Jenkins through [apt](#).

Recent versions are available in [an apt repository](#). Older but stable LTS versions are in [this apt repository](#).

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -  
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'  
sudo apt-get update  
sudo apt-get install jenkins
```

This package installation will:

- Setup Jenkins as a daemon launched on start. See [/etc/init.d/jenkins](#) for more details.
- Create a 'jenkins' user to run this service.
- Direct console log output to the file [/var/log/jenkins/jenkins.log](#). Check this file if you are troubleshooting Jenkins.
- Populate [/etc/default/jenkins](#) with configuration parameters for the launch, e.g [JENKINS_HOME](#)
- Set Jenkins to listen on port 8080. Access this port with your browser to start configuration.



If your [/etc/init.d/jenkins](#) file fails to start Jenkins, edit the [/etc/default/jenkins](#) to replace the line `---- HTTP_PORT=8080----` with `----HTTP_PORT=8081----` Here, "8081" was chosen but you can put another port available.

```
- name: Add Jenkins key
  apt_key:
    url: https://pkg.jenkins.io/debian/jenkins.io.key
    state: present
  become: true

- name: Add Jenkins repository
  apt_repository:
    repo: deb http://pkg.jenkins.io/debian-stable binary/
    state: present
    filename: jenkins
  become: true

- name: Update cache
  apt:
    update_cache: yes
  become: true

- name: Install Jenkins
  apt:
    name: jenkins
  become: true
```



TERRAFORM



{JSON}



YAML



VCS



VCS

Identyfikowalność



VCS

Identyfikowalność

Wersjonowanie



VCS

Identyfikowalność

Wersjonowanie

Korelacja



VCS

Identyfikowalność

Wersjonowanie

Korelacja

Widoczność



VCS

Identyfikowalność

Wersjonowanie

Korelacja

Widoczność



Wykonalność

```
$ cat goss.yaml
package:
  jenkins:
    installed: true
service:
  jenkins:
    enabled: true
    running: true
http:
  {{ .Vars.jenkins_host }}/login:
    status: 200
```

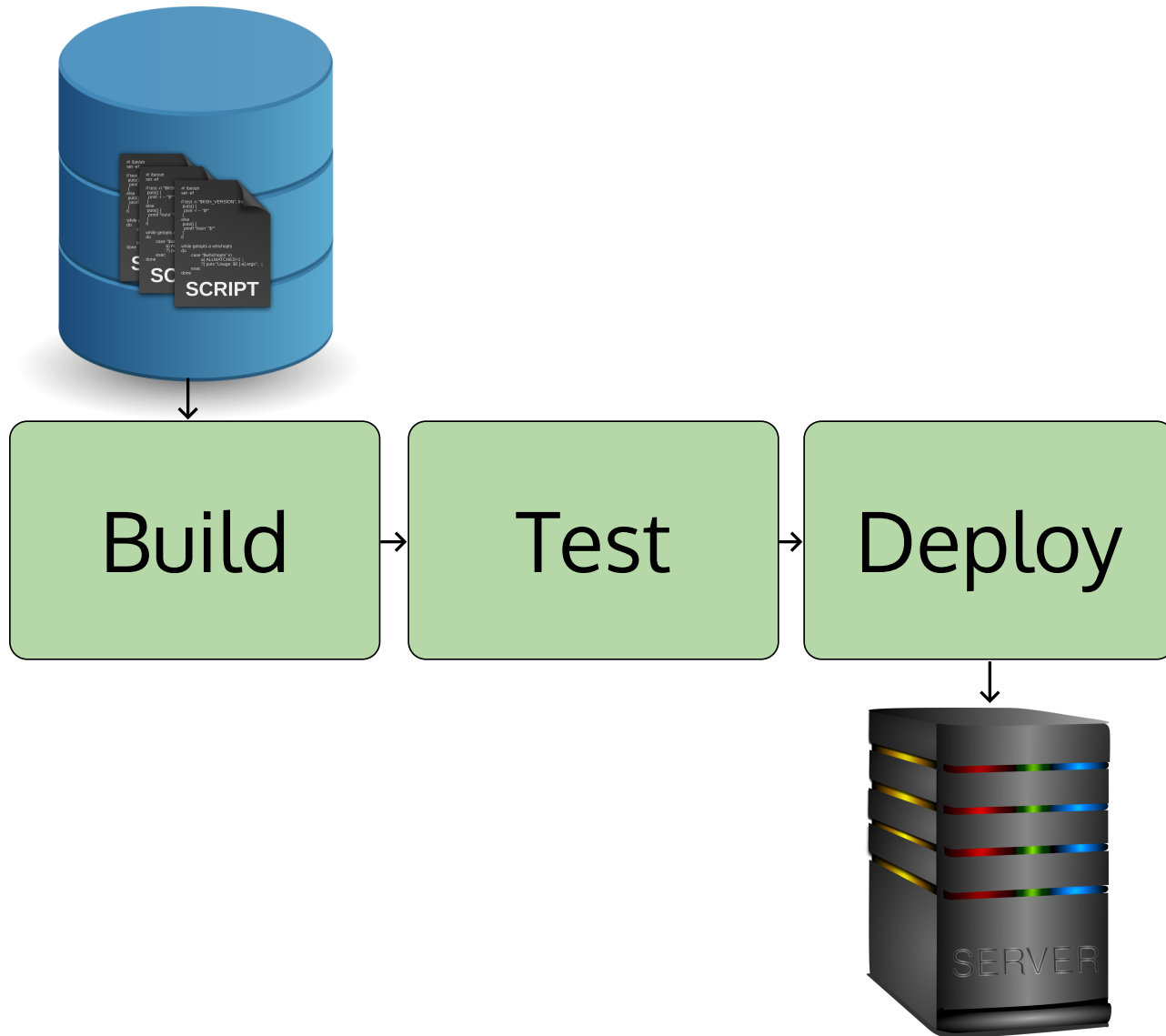
```
$ cat goss.yaml
package:
  jenkins:
    installed: true
service:
  jenkins:
    enabled: true
    running: true
http:
  {{ .Vars.jenkins_host }}/login:
    status: 200
```

```
$ goss --vars vars.yaml validate --format documentation
```

```
Service: jenkins: enabled: matches expectation: [true]
Service: jenkins: running: matches expectation: [true]
Package: jenkins: installed: matches expectation: [true]
HTTP: http://localhost:8080/login: status:
```

```
Expected
  <int>: 503
to equal
  <int>: 200
```

```
Total Duration: 1.404s
Count: 4, Failed: 1, Skipped: 0
```





Wykonałem
zmianę poza
automatyzacją

Mój serwer jest
niespójny

STRACH!

Boję się, że
automatyzacja
coś mi popsuje

Wyko
zmiar
autom



ver jest
ójny

Doję się, że
automatyzacja
coś mi popsuje



Jak

to možna

zrobić?





1


Setup Jenkins

Instalacja

https://hub.docker.com/r/jenkins/jenkins/

The screenshot shows the Docker Hub page for the `jenkins/jenkins` image. The page has a blue header with the Docker Hub logo and a search bar. The main content area features a blue cube icon for the image, the name `jenkins/jenkins` with a star, and a pull count of 10M+. Below this is a navigation bar with 'Overview' and 'Tags' tabs. The main content area is divided into two columns. The left column contains the title 'Jenkins Continuous Integration and Delivery server.', a description 'This is a fully functional Jenkins server, based on the weekly and LTS releases.', the Jenkins logo, and a list of instructions for pulling the image. The right column contains a 'Docker Pull Command' section with the command `docker pull jenkins/jenkins` and an 'Owner' section showing the user 'jenkins'.

docker hub Search for great content (e.g., mysql) Explore Sign In Sign Up


 **jenkins/jenkins** ☆ Pulls 10M+

By `jenkins` • Updated 2 hours ago
The leading open source automation server
Container

Overview Tags

Jenkins Continuous Integration and Delivery server.

This is a fully functional Jenkins server, based on the weekly and LTS releases .

 **Jenkins**


- To use the latest LTS: `docker pull jenkins/jenkins:lts`
- To use the latest weekly: `docker pull jenkins/jenkins`
- Lighter alpine based image also available

[Read documentation for usage](#)

Docker Pull Command

```
docker pull jenkins/jenkins
```

Owner

 `jenkins`

Uruchomienie obrazu docker

```
docker-compose.yaml
```

```
version: '3.7'  
services:  
  jenkins:  
    image: "jenkins/jenkins"  
    ports:  
      - "8080:8080"  
    volumes:  
      - "./jenkins_home:/var/jenkins_home"
```

```
$ docker-compose up
```


Zaczynamy

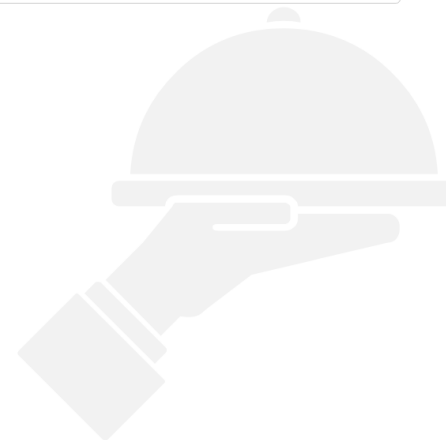
Odblokuj Jenkinsa

Aby zapewnić, że Jenkins jest bezpiecznie uruchomiony przez administratora, hasło zostało zapisane do pliku logów ([nie masz pewności, gdzie go znaleźć?](#)) oraz w pliku na serwerze:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Skopiuj hasło z jednej z powyższych lokalizacji i wklej poniżej.

Hasło administratorskie:



Kontynuuj

Instalowanie dodatkowych pakietów

Dockerfile

```
FROM jenkins/jenkins
USER root
RUN apt-get update && apt-get install -y maven
USER jenkins
```

docker-compose.yml

```
version: '3.7'
services:
  jenkins:
    image: "jenkins"
    build: .
    ports:
      - "8080:8080"
    volumes:
      - "./jenkins_home:/var/jenkins_home"
```

```
$ docker-compose up --build
```



2 Setup Jenkins

Post-Instalacja

Wyłączenie WizardSetup

```
docker-compose.yaml
```

```
version: '3.7'
services:
  jenkins:
    image: "jenkins"
    build: .
    environment:
      JAVA_OPTS: "-Djenkins.install.runSetupWizard=false"
    ports:
      - "8080:8080"
    volumes:
      - "./jenkins_home:/var/jenkins_home"
```

https://plugins.jenkins.io/

The screenshot shows the Jenkins Plugins website. The top navigation bar includes 'Jenkins', 'Blog', 'Documentation', 'Plugins' (underlined), 'Community', 'Sub-projects', 'About', and 'English'. Below this is a search bar with 'Browse' and a search icon, containing the text 'greenballs'. The left sidebar has 'Sort relevance' options: 'Relevance' (selected), 'Most installed', 'Trending', 'Title', and 'Release date'. Below that are 'Categories' with checkboxes for 'Platforms', 'iOS development', '.NET', 'Android development', and 'Ruby development'. The main content area shows a search result for 'greenballs' with '1 to 1 of 1' results. The result card for 'Green Balls' includes the text: 'Installs: 33423', 'Jenkins 1.440++', 'User Interface,', and 'Because green is better than blue! For color blind support configure user property.' At the bottom of the card, it says 'Asgeir Storesund Nilsen' and has a 'GB' icon.

https://plugins.jenkins.io/

The screenshot shows the Jenkins Plugins page for the 'Green Balls' plugin. The page has a dark blue header with the Jenkins logo and navigation links: Blog, Documentation, Plugins (underlined), Community, Sub-projects, About, and English. Below the header is a blue bar with a back arrow and the text 'Find plugins'. The main content area is white and features the plugin name 'Green Balls' in large black font, with the version '1.15' to its right. Below the name, it states 'Minimum Jenkins requirement: 1.440' and 'ID: greenballs'. There are three columns of information: 'Installs: 33423' with a 'GitHub →' link and 'Last released: 3 years ago'; 'Maintainers' listing 'Asgeir Storesund Nilsen'; and 'Dependencies' listing 'External Monitor Job Type v.1.0 (implied) (what's this?)', 'LDAP v.1.0 (implied) (what's this?)', 'PAM Authentication v.1.0 (implied) (what's this?)', 'Mailer v.1.2 (implied) (what's this?)', and 'Matrix Authorization Strategy v.1.0.2 (implied) (what's this?)'. On the right side, there is a sidebar with a 'Archives' section and a line chart showing install trends from February to July.

Jenkins Blog Documentation Plugins Community Sub-projects About English

← Find plugins

Green Balls ^{1.15}

Minimum Jenkins requirement: 1.440
ID: greenballs

Installs: 33423
[GitHub →](#)
Last released: 3 years ago

Maintainers
Asgeir Storesund Nilsen

Dependencies
[External Monitor Job Type v.1.0 \(implied\) \(what's this?\)](#)
[LDAP v.1.0 \(implied\) \(what's this?\)](#)
[PAM Authentication v.1.0 \(implied\) \(what's this?\)](#)
[Mailer v.1.2 \(implied\) \(what's this?\)](#)
[Matrix Authorization Strategy v.1.0.2 \(implied\) \(what's this?\)](#)

Archives
Get past

Month	Installs
Feb	28000
Mar	29000
Apr	29000
May	29000
Jun	29000
Jul	29000

Instalowanie pluginów

Dockerfile

```
FROM jenkins/jenkins
RUN /usr/local/bin/install-plugins.sh greenballs
USER root
RUN apt-get update && apt-get install -y maven
USER jenkins
```


- [Powrót do tablicy](#)
- [Zarządzaj Jenkinsem](#)

Filtruj:

- Aktualizacje
- Dostępne
- Zainstalowane**
- Zaawansowane

Włączone wtyczki	Nazwa	Wersja	Poprzednia zainstalowana wersja	Odinstaluj
<input checked="" type="checkbox"/>	Green Balls Because green is better than blue! For color blind support configure user property.	1.15		Odinstaluj



3 Setup Jenkins Konfiguracja

<https://github.com/jenkinsci/configuration-as-code-plugin>

Jenkins Configuration as Code Plugin

build passing plugin v1.6 chat on gitter



View the [wiki](#) page. See [presentation slides](#) from Jenkins World 2018.

Join our Jenkins Configuration as Code (JCasC) office hours meeting scheduled for every second Wednesday. Use the Hangout on Air link from our [Gitter](#) chat channel. As an alternative, use the link from the [invitation](#). See previous [meeting minutes](#).

Konfiguracja

```
conf-as-code.yaml
```

```
jenkins:  
  systemMessage: "IaC Jenkins"  
  securityRealm:  
    local:  
      allowsSignup: false  
      users:  
        - id: "${JENKINS_USER}"  
          password: "${JENKINS_PASS}"
```

Instalowanie CasC

Dockerfile

```
FROM jenkins/jenkins
RUN /usr/local/bin/install-plugins.sh greenballs
RUN /usr/local/bin/install-plugins.sh configuration-as-code
RUN /usr/local/bin/install-plugins.sh configuration-as-code-support
USER root
RUN apt-get update && apt-get install -y maven
USER jenkins
```

Podanie lokalizacji plików konfiguracyjnych

```
docker-compose.yaml
```

```
version: '3.7'
services:
  jenkins:
    image: "jenkins"
    build: .
    environment:
      JAVA_OPTS: "-Djenkins.install.runSetupWizard=false"
      CASC_JENKINS_CONFIG: "/var/jenkins_home/casc_configs"
      JENKINS_USER: "${JENKINS_USER}"
      JENKINS_PASS: "${JENKINS_PASS}"
    ports:
      - "8080:8080"
    volumes:
      - "./jenkins_home:/var/jenkins_home"
      - "./configuration:/var/jenkins_home/casc_configs"
```

W jaki sposób pisać pliki konfiguracyjne?

active-directory	jobs
alauda-devops-sync	keycloak
artifactory	kubernetes-helm
build_agents	kubernetes-secrets
config-file-provider	kubernetes
credentials	ldap
docker	mailer
ec2	mesos
embedded-userdatabase	plugins
git	role-strategy-auth
github-oauth	simple-theme-plugin
github	slack
gitlab	sonarqube
gitscm	statistics-gatherer
global-matrix-auth	terraform
google-login	tfs
graphite	warnings
jenkins	workflow-cps-global-lib

W jaki sposób pisać pliki konfiguracyjne?

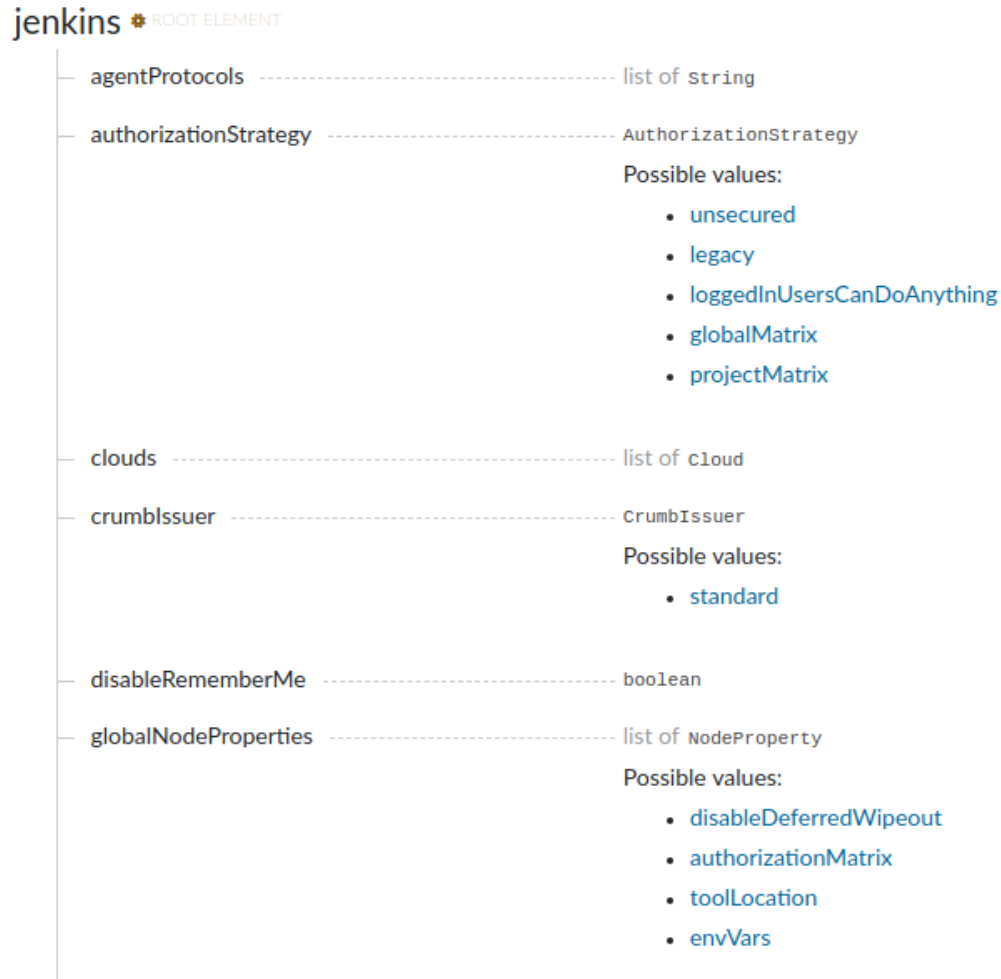


Configuration as Code

Reload your configuration or update configuration source

[Documentation](#)

W jaki sposób pisać pliki konfiguracyjne?



Bezpieczeństwo

Dockerfile

```
FROM jenkins/jenkins
RUN /usr/local/bin/install-plugins.sh greenballs
RUN /usr/local/bin/install-plugins.sh configuration-as-code
RUN /usr/local/bin/install-plugins.sh configuration-as-code-support
RUN /usr/local/bin/install-plugins.sh matrix-auth
USER root
RUN apt-get update && apt-get install -y maven
USER jenkins
```

Zablokowanie możliwość konfigurowania

```
conf-as-code.yaml
```

```
jenkins:  
  systemMessage: "IaC Jenkins"  
  authorizationStrategy:  
    globalMatrix:  
      grantedPermissions:  
        - "Overall/Read:authenticated"  
        - "Job/Build:authenticated"  
        - "Job/Cancel:authenticated"  
        - "Job/Read:authenticated"  
  securityRealm:  
    local:  
      allowsSignup: false  
      users:  
        - id: "${JENKINS_USER}"  
          password: "${JENKINS_PASS}"
```

ImmutableServer



Kief Morris
13 June 2013

Automated configuration tools (such as [CFEngine](#), [Puppet](#), or [Chef](#)) allow you to specify how servers should be configured, and bring new and existing machines into compliance. This helps to avoid the problem of fragile [SnowflakeServers](#). Such tools can create [PhoenixServers](#) that can be torn down and rebuilt at will. An Immutable Server is the logical conclusion of this approach, a server that once deployed, is never modified, merely replaced with a new updated instance.

Automated configuration tools are usually used with [ConfigurationSynchronization](#) where you leave a server running for a potentially long period of time, repeatedly applying configuration to bring it into line with the latest specification. In theory servers can be allowed to run indefinitely, and they'll be kept completely consistent and up to date. In practice it's not possible to manage a server's configuration completely, so there is considerable scope for configuration drift, and unexpected changes to running servers.



4 Setup Jenkins

Konfiguracja zadań

`https://github.com/jenkinsci/job-dsl-plugin`

Jenkins Job DSL / Plugin

The Jenkins "Job DSL / Plugin" is made up of two parts: The Domain Specific Language (DSL) itself that allows users to describe jobs using a Groovy-based language, and a Jenkins plugin which manages the scripts and the updating of the Jenkins jobs which are created and maintained as a result.

Background

Jenkins is a wonderful system for managing builds, and people love using its UI to configure jobs. Unfortunately, as the number of jobs grows, maintaining them becomes tedious, and the paradigm of using a UI falls apart. Additionally, the common pattern in this situation is to copy jobs to create new ones, these "children" have a habit of diverging from their original "template" and consequently it becomes difficult to maintain consistency between these jobs.

The Jenkins job-dsl-plugin attempts to solve this problem by allowing jobs to be defined with the absolute minimum effort in a programmatic form. The goal is for your team to be able to define all the jobs they wish to be related to their project, declaring their intent for the jobs programmatically, and leaving the common elements in each of them hidden behind the DSL.

For example, your project might require a unit test job, a nightly SonarQube build, an integration test job, and a promotion job. Permission to run the release job should be limited to certain users. Here's the example DSL script:

Job DSL

seed.groovy

```
job('example') {  
  triggers {  
    scm('*/*15 * * * *')  
  }  
  steps {  
    shell("echo 'Hello world!'")  
  }  
}
```

Dockerfile

```
FROM jenkins/jenkins
RUN /usr/local/bin/install-plugins.sh greenballs
RUN /usr/local/bin/install-plugins.sh configuration-as-code
RUN /usr/local/bin/install-plugins.sh configuration-as-code-support
RUN /usr/local/bin/install-plugins.sh matrix-auth
RUN /usr/local/bin/install-plugins.sh job-dsl
USER root
RUN apt-get update && apt-get install -y maven
USER jenkins
```

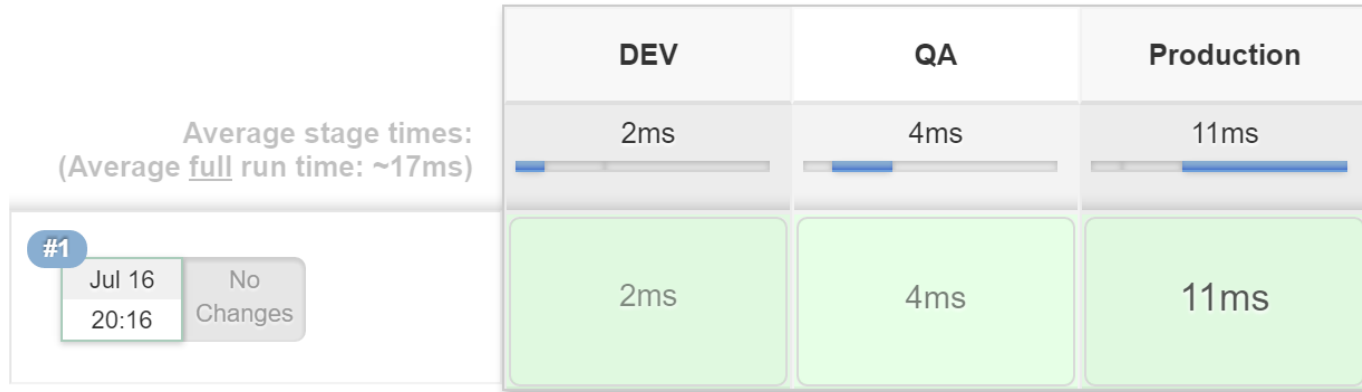
Konfiguracja Job DSL

```
conf-as-code-jobs.yaml
```

```
jobs:
  - script: >
      job('example') {
        triggers {
          scm('* /15 * * * *')
        }
        steps {
          shell("echo 'Hello world!'")
        }
      }
}
```

<https://github.com/jenkinsci/pipeline-plugin>

Stage View



Permalinks

- [Last build \(#1\), 2 hr 14 min ago](#)
- [Last stable build \(#1\), 2 hr 14 min ago](#)
- [Last successful build \(#1\), 2 hr 14 min ago](#)

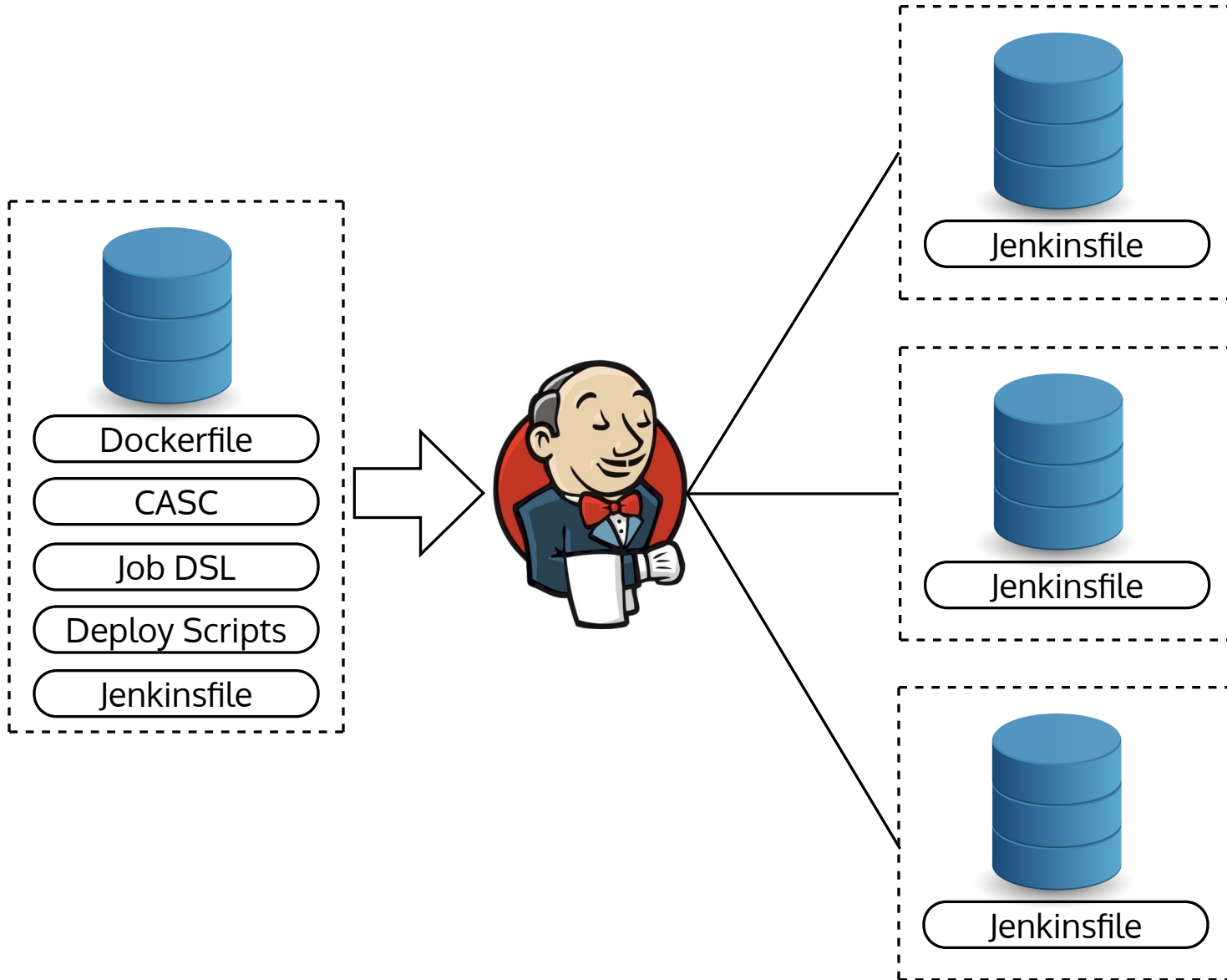
Deklaratywny pipeline

Jenkinsfile

```
pipeline {
  agent any
  stages {
    stage('Stage 1') {
      steps {
        echo 'Hello world!'
      }
    }
  }
}
```

Dockerfile

```
FROM jenkins/jenkins
RUN /usr/local/bin/install-plugins.sh greenballs
RUN /usr/local/bin/install-plugins.sh configuration-as-code
RUN /usr/local/bin/install-plugins.sh configuration-as-code-support
RUN /usr/local/bin/install-plugins.sh matrix-auth
RUN /usr/local/bin/install-plugins.sh job-dsl
RUN /usr/local/bin/install-plugins.sh workflow-aggregator
USER root
RUN apt-get update && apt-get install -y maven
USER jenkins
```





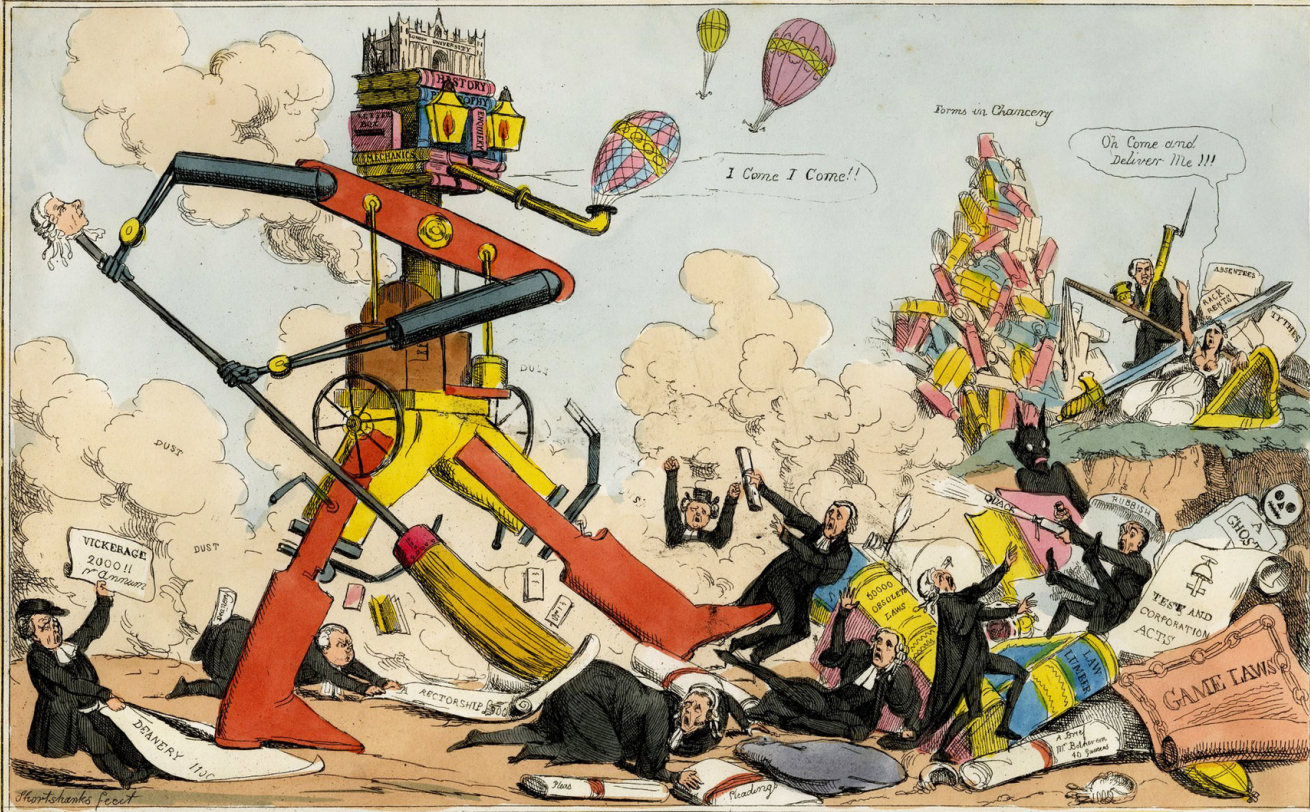
Technology Radar

JAN
2011

ADOPT ?

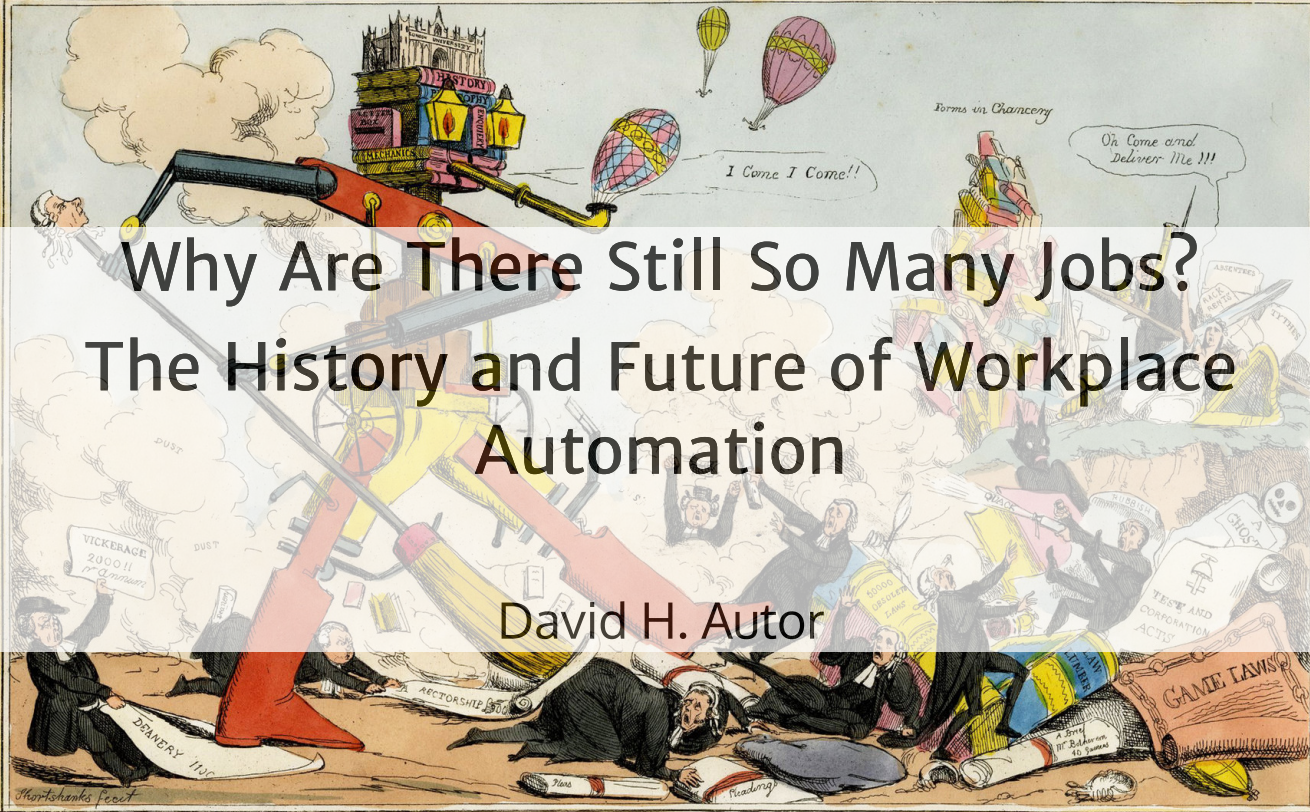
The large number of hosts and devices in a modern datacenter or cloud deployment have made manually installing and configuring infrastructure unwise. Infrastructure as code is an approach whereby infrastructure configuration is scripted or described by files that are stored in version control, and changes are pushed out to the datacenter in a controlled manner. This parallels the discipline of source control and build promotion used in software development, hence 'as code'. The two front-running open source tools for infrastructure automation are Chef and Puppet. They both use a textual DSL to script automation. Using this approach provides consistent and repeatable environment changes, reducing the manual effort involved, especially in troubleshooting environmental differences.

THE MARCH OF INTELLECT.



I saw a Vision, a Giant form appeared, its eyes were burning lights, even of Gays, and on its learned head it bore a Crown of many towers, Its Body was an Engine yea of Steam, its arms where iron, and the legs with which it stood like unto presses that men called printers use, from whence fell ever and anon small Books that fed the little people of the Earth, It rose and in its hand it took a Broom to sweep the rubbish from the face of the land, the Special pleaders & their wigs also & the Quack Doctors also and the Ghosts & those that wear Horns & the Crowns of those that set themselves above the laws & the Delays in Chancery it utterly destroyed, likewise it swept from the Clergy every Plurality, Nevertheless the Lawyers & the Parsons & divers others kickt up a great Dust !!!

THE MARCH OF INTELLECT.

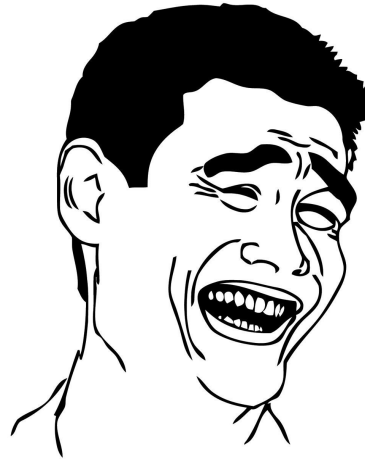
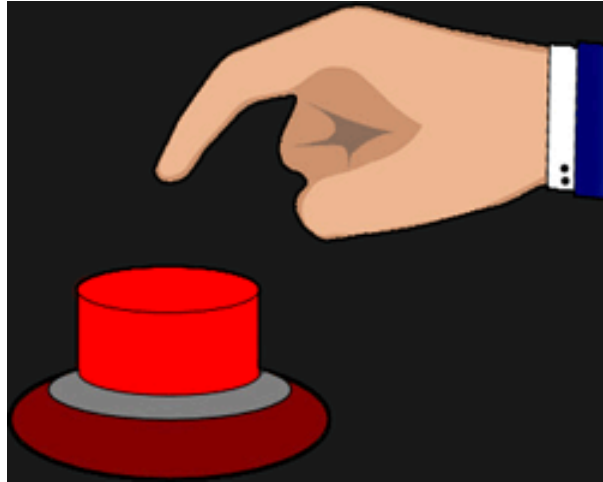


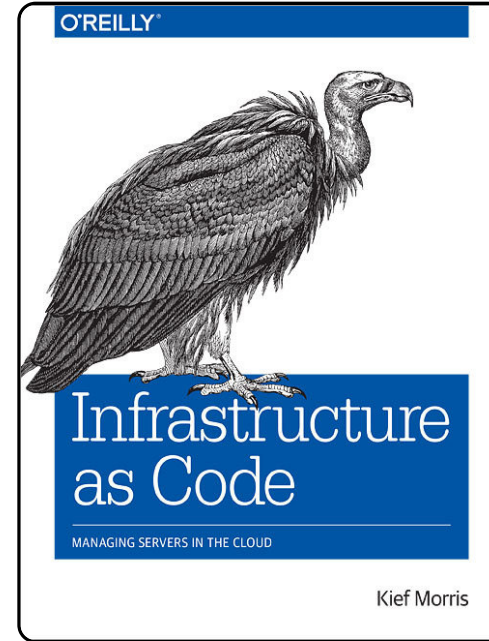
Why Are There Still So Many Jobs? The History and Future of Workplace Automation

David H. Autor

I saw a Vision, a Giant form appeared, its eyes were burning lights, even of Gays, and on its learned head it bore a Crown of many towers, Its Body was an Engine yea of Steam, its arms where iron and the legs with which it stood like unto presses that men called printers use, from whence fell ever and anon small Books that fed the little people of the Earth, It rose and in its hand it took a Broom to sweep the rubbish from the face of the land, the Special pleaders & their wigs also & the Quack Doctors also and the Ghosts & those that wear Horns & the Crowns of those that set themselves above the laws & the Delays in Chancery it utterly destroyed, likewise it swept from the Clergy every Plurality, Nevertheless the Lawyers & the Parsons & divers others kickt up a great Dust!!!







```
https://github.com/zielona-gora-jug/jenkins-iac
https://jenkins.io
https://hub.docker.com/r/jenkins/jenkins/
https://plugins.jenkins.io/
https://github.com/jenkinsci/configuration-as-code-plugin
https://github.com/jenkinsci/job-dsl-plugin
https://github.com/jenkinsci/pipeline-plugin
```



Dziękims!