

Ochrona Centrów Danych

Raport 2



Politechnika Wrocławska

Anna Płecha, 241446

Wrocław, 2021

1. Budowa projektu GitHub przy pomocy Jenkinsa

W tej części należało ustawić automatyczny dostęp do repozytorium GitHub dla serwisu Jenkins. Przykładowy kod źródłowy umieszczony w stworzonym na potrzeby repozytorium testowym powinien być pobierany automatycznie.

Na początku stworzono testowe repozytorium GitHub.

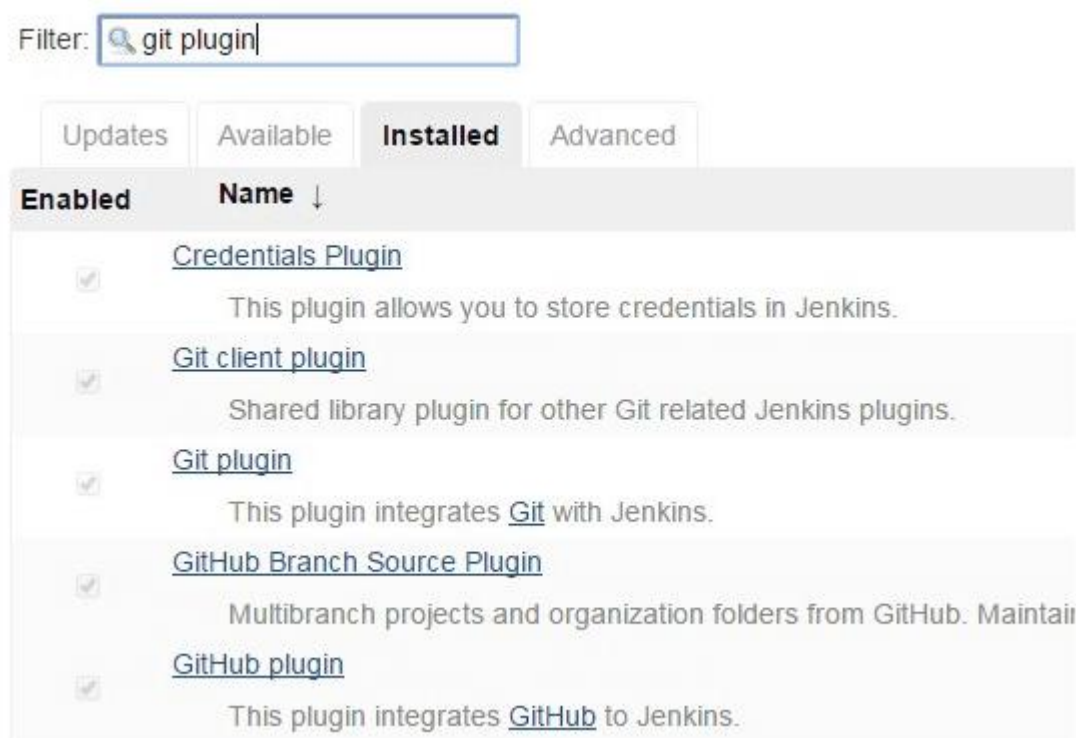


Następnie zmieniono jego widoczność na prywatny.

Następnie w repozytorium utworzono plik z testowym kodem źródłowym - „Hello World” w języku C.

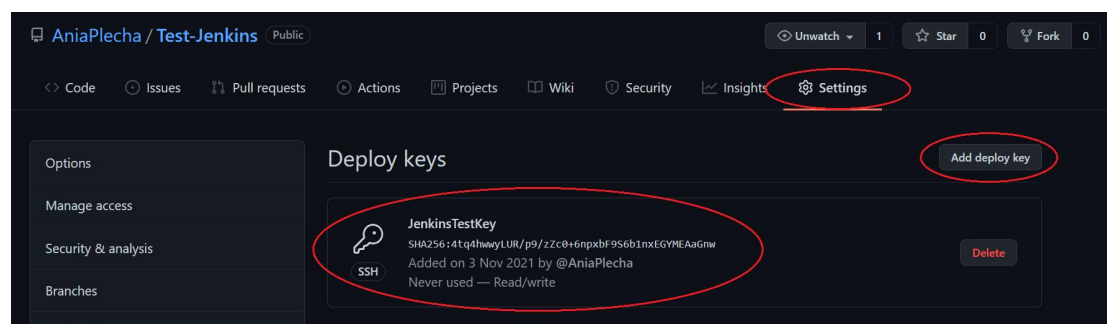


Na maszynie wirtualnej z zainstalowanym Jenkinsem na początku odpowiednio skonfigurowano Jenkinsa - utworzono konto administratora, domyślny adres serwisu zmieniono na adres maszyny wirtualnej. Następnie zainstalowano wszystkie wymagane wtyczki:



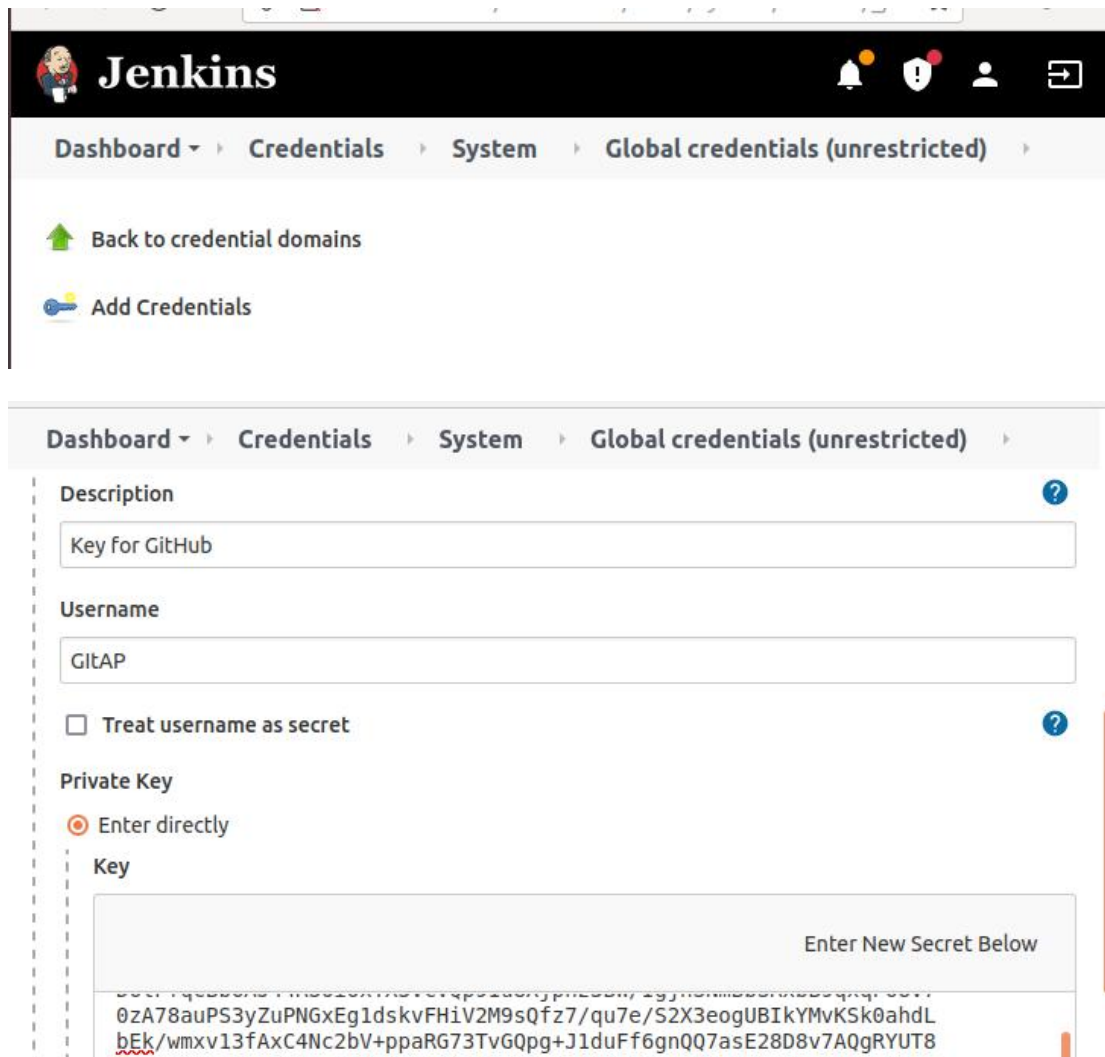
By dostęp do testowego repo był automatyczny, łączność powinna odbywać się za pomocą pary kluczy. Do łączności użyto kluczy wgenerowanych za pomocą polecenia „ssh-keygen -t rsa -b 4096 -f github_jd_rsa.

Klucz publiczny został umieszczony na serwisie GitHub za pomocą funkcji *Add deploy key*.



Klucz prywatny z kolei został umieszczony na Jenkinsie w następujący sposób. Na Jenkins należało wejść kolejno w ścieżkę jak na screenie i za

pomocą opcji „Add credentials” utworzyć nowe uwierzytelnianie kluczem prywatnym.



The screenshot shows the Jenkins web interface. At the top, the Jenkins logo and navigation bar are visible. The breadcrumb trail is: Dashboard > Credentials > System > Global credentials (unrestricted). Below the breadcrumb, there are two buttons: 'Back to credential domains' and 'Add Credentials'. The 'Add Credentials' button is highlighted. Below it, the 'Add Credentials' form is shown. The form has a title bar with the same breadcrumb trail. The form fields are: 'Description' (text input with 'Key for GitHub'), 'Username' (text input with 'GitAP'), 'Treat username as secret' (checkbox, unchecked), 'Private Key' (radio button selected, 'Enter directly'), and 'Key' (text area with a long alphanumeric string). A vertical orange bar is on the right side of the form.

Dashboard > Credentials > System > Global credentials (unrestricted)

Back to credential domains

Add Credentials

Description

Key for GitHub

Username

GitAP

☐ Treat username as secret

Private Key

☒ Enter directly

Key

Enter New Secret Below

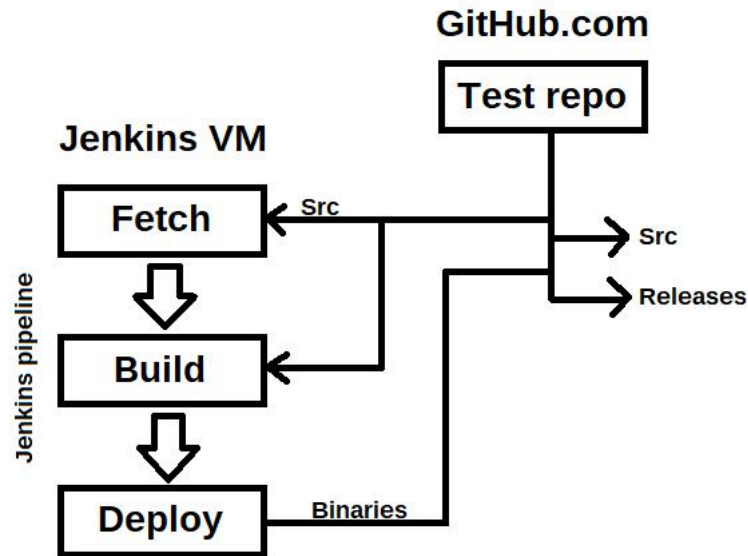
0zA78auPS3yZuPNGxEg1dskvFHiV2M9sQfz7/qu7e/S2X3eogUBIKYmKSk0ahdL
bEk/wmxv13fAxC4Nc2bV+ppaRG73TvGQpg+J1duFf6gnQQ7asE28D8v7AQgRYUT8

W ten sposób utworzono automatyczny szyfrowany dostęp Jenkinsa do GitHub.

2. Kompilacja do postaci binarnej. Tryb potokowy (Jenkins pipeline)

Następnie kod źródłowy z GitHub powinien zostać skompilowany do postaci binarnej i umieszczony na nim powrotnie. Kroki potoku powinny wyglądać w sposób następujący:

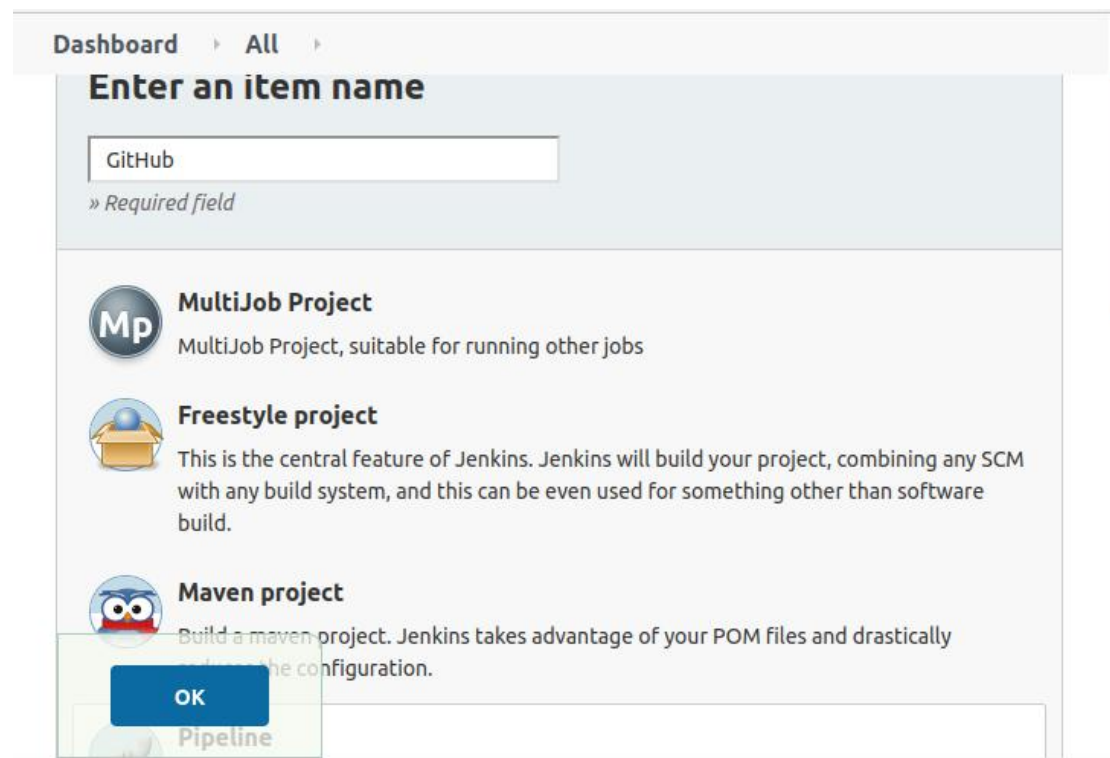
Pobranie kodu z repozytorium -> kompilacja -> testowe uruchomienie
-> zapisanie artefaktów na GitHub



Rys 1. Schemat kolejnych kroków potoku.

Wykonano następujące kroki:

Utworzono nowy projekt w Jenkinsie w kategorii Pipeline



Następnie w edycji projektu, w zakładce Pipeline ustawiono pobieranie kodu z repozytorium testowego na GitHub i połączenie za pomocą zdefiniowanego wcześniej uwierzytelniania kluczem.

Dashboard > GitHub >

General Build Triggers Advanced Project Options **Pipeline**

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

git@github.com:AniaPlecha/Test-Jenkins.git

Credentials

GitAP (Key for GitHub) Add

Save Apply Advanced...

Additional Behaviours

Add

Script Path

helloworldind

☒ Lightweight checkout

Help Pipeline Syntax

Po uruchomieniu projektu powinny wykonać się określone zadania w ramach pipeline, czego efekt powinien być widoczny w logach. Niestety na moment obecny ten etap kończy się poniższym błędem:

Console Output

Uruchomiono przez użytkownika [Admin](#)

hudson.plugins.git.GitException: Command "git fetch --tags --progress --prune -- origin +refs/heads/master:refs/remotes/origin/master" returned status code 128:

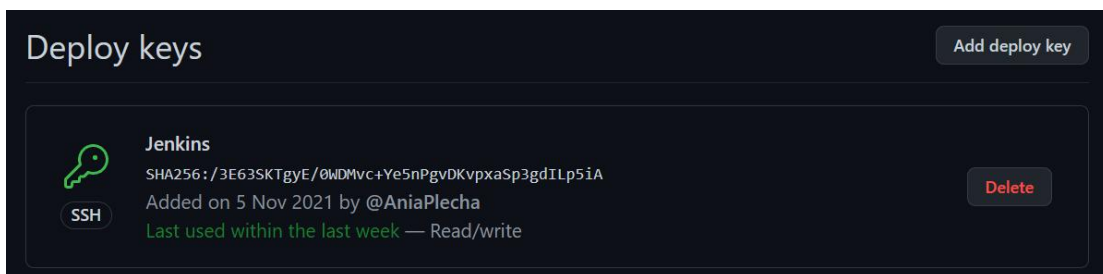
stdout:

stderr: fatal: **Couldn't find remote ref refs/heads/master**


```
    at org.jenkinsci.plugins.gitclient.CliGitAPIImpl.launchCommandIn(CliGitAPIImpl.java:2681)
    at
org.jenkinsci.plugins.gitclient.CliGitAPIImpl.launchCommandWithCredentials(CliGitAPIImpl.java:2102)
    at org.jenkinsci.plugins.gitclient.CliGitAPIImpl.access$500(CliGitAPIImpl.java:86)
    at org.jenkinsci.plugins.gitclient.CliGitAPIImpl$1.execute(CliGitAPIImpl.java:624)
    at jenkins.plugins.git.GitSCMFileSystem$BuilderImpl.build(GitSCMFileSystem.java:366)
    at jenkins.scm.api.SCMFileSystem.of(SCMFileSystem.java:197)
    at jenkins.scm.api.SCMFileSystem.of(SCMFileSystem.java:173)
    at org.jenkinsci.plugins.workflow.cps.CpsScmFlowDefinition.create(CpsScmFlowDefinition.java:114)
    at org.jenkinsci.plugins.workflow.cps.CpsScmFlowDefinition.create(CpsScmFlowDefinition.java:68)
    at org.jenkinsci.plugins.workflow.job.WorkflowRun.run(WorkflowRun.java:310)
    at hudson.model.ResourceController.execute(ResourceController.java:99)
    at hudson.model.Executor.run(Executor.java:431)
```

Finished: FAILURE

Łączność Jenkinsa z GitHub za pomocą klucza jest prawidłowa, ponieważ na GitHub widzimy, że klucz został użyty:



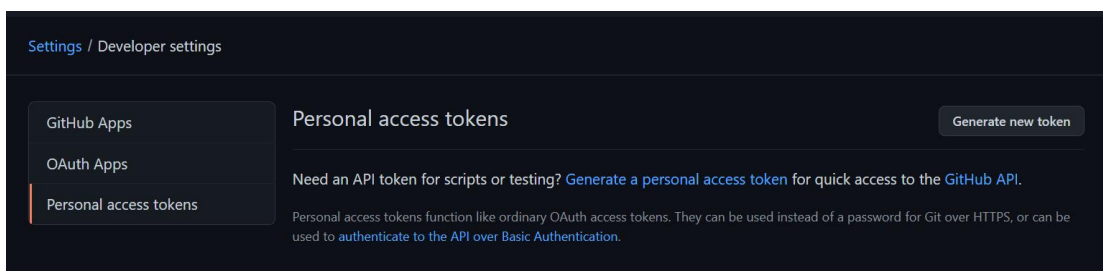
Deploy keys [Add deploy key](#)

 **Jenkins**
SHA256:/3E63SKTgyE/0WDMvc+Ye5nPgvDKVpxaSp3gdIlp5iA
Added on 5 Nov 2021 by @AniaPlecha
Last used within the last week — Read/write [Delete](#)

3. Publikacja artefaktów binarnych w testowym repozytorium GitHub

W tym punkcie należało dodatkowo ustawić tokeny dostępowe do funkcji GitHub Releases.

Tokeny utworzyć można w zakładce Settings -> Developer settings -> Personal Access Tokens -> Generate new token



[Settings](#) / [Developer settings](#)

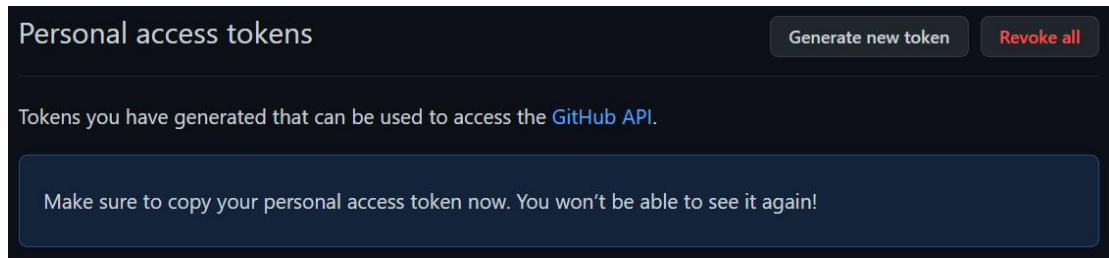
[GitHub Apps](#)
[OAuth Apps](#)
[Personal access tokens](#)

Personal access tokens [Generate new token](#)

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Token należy gdzieś zapisać, ponieważ będzie jedynie raz wyświetlony.



Ważne: zawartości kluczy i tokenów nie mogą być ujawnione w logach!