

## Kako se povezati Jenkins sa svojim github-nalogom

Generisacemo token na nasem git hab nalogom i od ponudjenih opcija uzećemo samo

### Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> <b>repo</b>	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Regis
<input type="checkbox"/> read:packages	Download packages from GitHub Package

Vratim se sada u moj Jenkins i idem u Manage Jenkins pa u Configure System.

Idem dole sve do odseka sa Githabom.

**Manage Jenkins**

- + New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins**
- My Views

Warnings have been published for the following currently installed components:

**Build Pipeline Plugin 2.0.1:**  
Stored XSS vulnerability (no fix available)  
No fixes for these issues are available. It is recommended that you review the security analysis and fix the issue.

### System Configuration

**System**  
Configure global settings and paths.

**Tools**  
Configure tools, their automatic installers.

**Appearance**  
Configure the look and feel of Jenkins.

**Build Queue**  
No builds in the queue.

**Build Executor Status**  
built-in node (0 of 2 executors busy)

Ovde potrazim deo sa githubom

Kada instalirate predložene plaginove vi ste instalirali i plugin za github

Default notification URL

Notification URL ?  
Default

**GitHub**

GitHub Servers ?  
Add GitHub Server

Advanced Edited

Git hub plugin se automatski instalira

**GitHub API usage**

GitHub API usage rate limiting strategy ?  
Normalize API requests

**GitHub Enterprise Servers**

Add Save Apply

Kada dodjemo da izaberemo vezu sa našim nalogom na github-u izabracemo skriveni tekst gde ćemo token iskoristiti za povezivanjem sa našim nalogom.

Default

## GitHub

### GitHub Servers ?

#### GitHub Server ?

Name ?

GitHub

API URL ?

https://api.github.com

Credentials ?

- none -

+ Add ▾



Jenkins

Manage hooks

Advanced ▾

Test connection

O vde moram da  
ubacim nas token,  
pod skrivenim  
tekstom

Sign in [Jenkins] X Role-based Authoriza X System [Jenkins] X + ▾

localhost:8080/manage/configure

Dashboard > Manage Jenkins > System >

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Secret text

Treat username as secret

Password:

ID:

Description:

Jenkins Credentials Provider: Jenkins

Add GitHub

GitHub API URL: https://api.github.com

Credentials: - none -

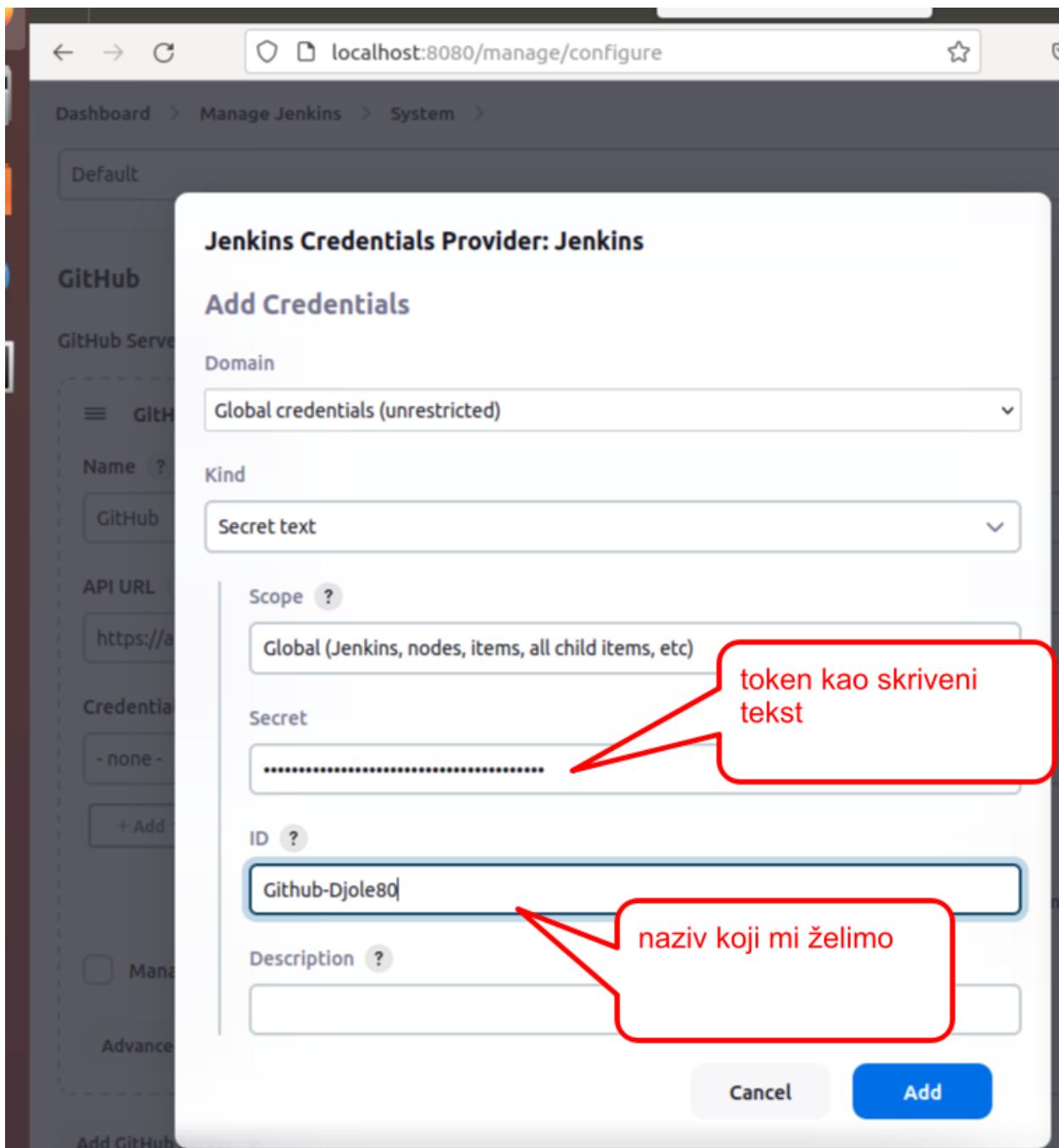
+ Add

Manage GitHub

Advanced

Add GitHub

A screenshot of the Jenkins 'Add Credentials' dialog. The dialog title is 'Jenkins Credentials Provider: Jenkins' and the sub-section is 'Add Credentials'. Under 'Domain', it shows 'Global credentials (unrestricted)'. In the 'Kind' dropdown, 'Secret text' is selected, highlighted with an orange background. Other options in the dropdown include 'Username with password', 'GitHub App', 'SSH Username with private key', 'Secret file', and 'Certificate'. Below the dropdown are fields for 'Treat username as secret', 'Password', 'ID', and 'Description'. On the left side of the dialog, there's a sidebar with sections for GitHub, API URL, and Credentials, each with their own 'Add' buttons.

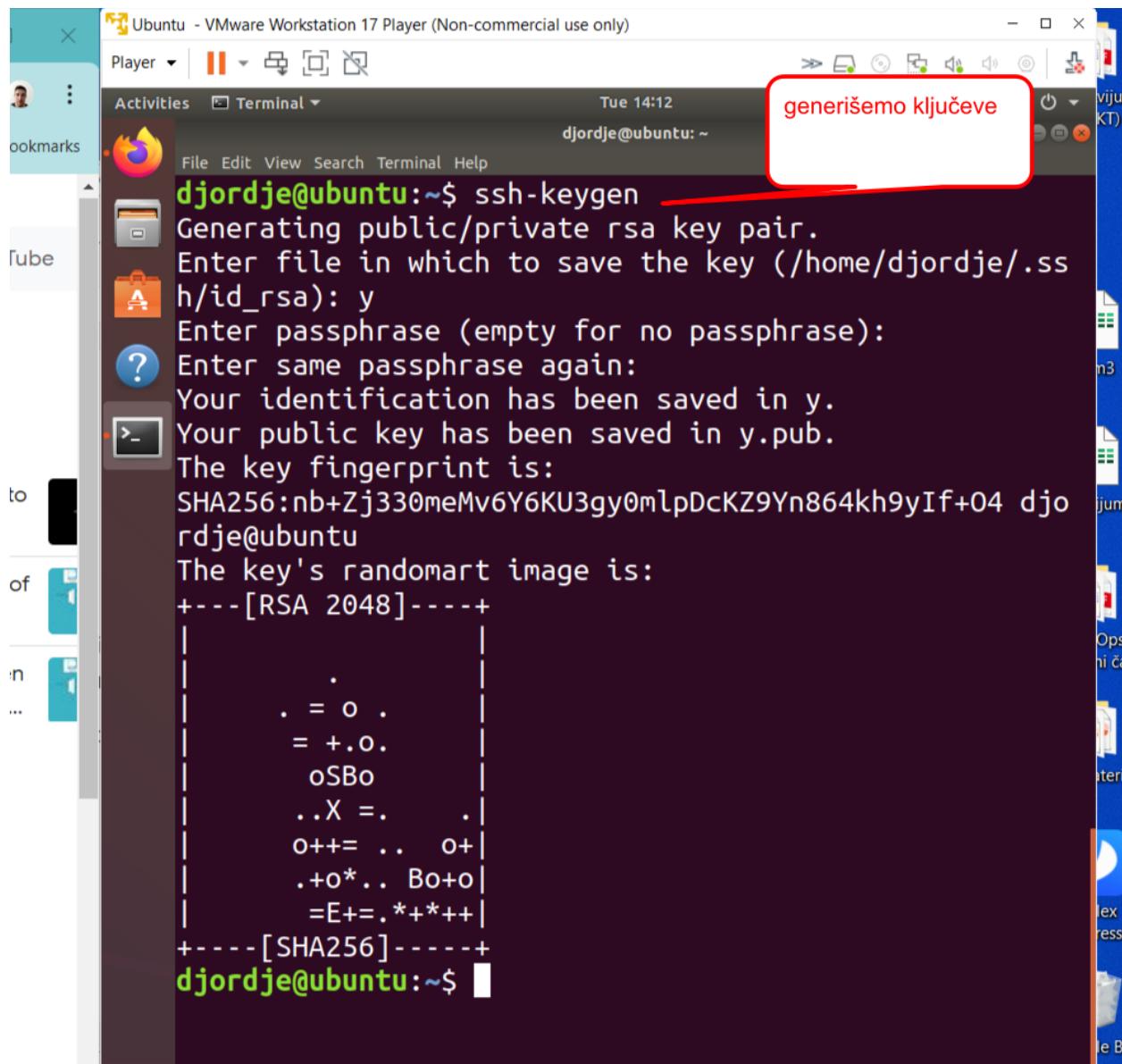


Proveriomo sada podešavanjem na našu konekciju da li ona rati . To možemo pritiskom na test connection taster.

## Sledeće što ćemo uraditi je da se povežemo sa SSH ključem

Na taj način moćićemo da cloniramo repozitorijum na Jenkinso lokalnom prostoru. Takođe gde god moj Github ima pristup imaće i Jenkins.

Da vidimo kako to možemo , moramo prvo u terminalu da generišemo privatni i javni ključ ssh.



```
Ubuntu - VMware Workstation 17 Player (Non-commercial use only)
Player Activities Terminal Tue 14:12 djordje@ubuntu:~ generišemo ključeve
File Edit View Search Terminal Help
djordje@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/djordje/.ssh/id_rsa): y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in y.
Your public key has been saved in y.pub.
The key fingerprint is:
SHA256:nb+Zj330meMv6Y6KU3gy0mlpDcKZ9Yn864kh9yIf+04 djordje@ubuntu
The key's randomart image is:
+---[RSA 2048]---+
| . |
| . = o . |
| = +.o. |
| oSBo |
| ..X =. . |
| o++= ... o+ |
| .+o*.. Bo+o |
| =E+=.*+*++ |
+---[SHA256]---+
djordje@ubuntu:~$
```

Da vidimo sada javni (public key ) ključ koji je

```
rw-r--r-- 2 djordje80 djordje80 4096 Feb 10  
rwxrwxr-x 4 djordje80 djordje80 4096 Mar 20  
rw-rw-r-- 1 djordje80 djordje80 199 Apr 29  
jordje@ubuntu:/home/djordje80$ cd .ssh  
jordje@ubuntu:/home/djordje80/.ssh$ ls  
d: no input files  
jordje@ubuntu:/home/djordje80/.ssh$ ls  
d_ed25519 id_ed25519.pub id_rsa id_rsa.pub  
jordje@ubuntu:/home/djordje80/.ssh$ id_rsa.pub  
d_rsa.pub: command not found  
jordje@ubuntu:/home/djordje80/.ssh$ cat id_rsa  
sh-rsa AAAAB3NzaC1yc2EAAAQABAAQCuDf6tjZS  
qWL9DPFNmmOBWni5qilOWtbFrmDWBnAz8NiTkUxIQpCHRxl  
7EijwM6lHl7/Lpl9TBSQmT6mLz2040buawbqvCdSSCr0ON  
kXTXUWb4uZqaX djolemladenovic@gmail.com  
jordje@ubuntu:/home/djordje80/.ssh$
```

Sada idem na moj Github nalog da postavim taj javni ključ.

github.com/settings/ssh/new

djmladenovic (djmladenovic)  
Your personal account

Add new SSH Key

Title: Jenkins-public-Key

Key type: Authentication Key

Key:

```
ssh-rsa  
AAAAB3NzaC1yc2EAAAQABAAQCuDf6tjZS...  
3VdnihlaDCaGqWL9DPFNmmOBWni5qilOWtbFrmDWBnAz8NiTkUxIQpCHRxBKJH...  
TMqHUF18mjezc9+18sW4Mddd9IPs7EijwM6lHl7/Lpl9TBSQmT6mLz2040buawbqvCdSSCr0ON7lokMhf3ml9:  
qiA4K1trPfVhkm4Tb4b6LltTDKSYQT2OD1eGKDqukXTXUWb4uZqaX djolemladenovic@gmail.com
```

Public profile Account Appearance Accessibility Notifications

Billing and plans Emails Password and authentication Sessions SSH and GPG keys Organizations

Idemo sada da iskopiramo naš private key u Jenkinsu gde treba.

```

[...]
/GsVjIGJS22JYHcSxs22/Tqq+8g9Ll9BzmDJvJ0mi5hxxbijahpJPm9L VF4Wt+10KmJwHx
GRu5dDqUuTgD42K5C3GqqSWLnd60mklmlcF6562akQx+XUnxpPeQfM67poDgafkrN0vd3Q
oDIgb6K1Xjt djordje@ubuntu
djordje@ubuntu:~/.ssh$ ^C
djordje@ubuntu:~/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
EpQIBAAKCAQEAbA0vKwQJCoEGXVoVF5tfLVr3xl0N/Euv2vBl0eRZttWDPu
let8oqHBrwB39C5Ra/wXI7wjvD9No+a/m5ochrFYyBiUttiWBoXEsbn tv06qvV
5fQc5gybydJouYccW4o2oaST5vS1ReFrftdCpiVh8XAPmgIamFUk1YFDpggBv
:1DnbaSHYMAiiK8CGXr0OY0w59AX0UBkbuXQ6llk4A+NiuQtxqqkli53etJpJZ
SeuetmpEMfl1J8aT3kHz0u6aA4Gn5KzdL3d0NArNc0/7SS3EI/9crOZEwfYx2b
IEHJIHd3IRLHoSC78NOAefDw6AyIG+itVybQIDAQABoIBAQCW8BEs0wzeNfUm
/HyQ9ejuotkN2gEL2xuySxMSIx6bBJztXFLRodG++/shcMDJGCPp6AK2BzqHZ
>PecSlpNa0pPZ/XdesYu1Mnnh3B3cn84nsuqtBwqFi9hwjCZGxQGPbxjhMLidY
/Zm1sn+BbkllKhML2Caw0EwGgZeGG00oL67vkEsyWobv701gd6/AfXLkU3PuL7
'MadHL /0?fh0RFcDn0Pi v4n6/vA7cIIIV77hM+1/C7h9/7n8r9hEWDRr11 c7wtt+

```

Sada u Jenkinsu idemo u **Manage Jenkins** pa u deo o sigurnosti pronađemo **Manage Credentials**

T	P	Store ↓	Domain	ID	Name
		System	(global)	Github-Djole80	Github-Djole80
		System	(global)	Github-jenkins	Github-jenkins

P	Store ↓	Domains
	System	(global)

Icon: S M L

Jenkins

Search (CTRL+K)

Dashboard > Manage Jenkins > Credentials > System >

System

Domain ↓ Description

Global credentials (unrestricted) Credentials that should be available irrespective of domain specification to requirements matching.

Add credentials

Icon: S M L

izabralo ovo

localhost:8080/manage/credentials/store/system/domain/\_newCredentials

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

SSH Username with private key

Scope ? Global (Jenkins, nodes, items, all child items, etc)

ID ? Private-key-github

Description ? naziv po izboru

Username

Treat username as secret ?

Private Key

Enter directly

Key

postavimo opciju za direktno unošenje privatnog ključa, tu ga iskopiramo!!!!

Create

I dobili smo željene konekcije sa github-om

The screenshot shows the Jenkins Global credentials (unrestricted) page. It lists two credentials:

ID	Name	Kind
Github-Djole80	Github-Djole80	Secret text
Private-key-github	jenkins	SSH Username with private key

Icons for each credential are shown to the right of their respective rows.

On bi sada trebao da može da klonira repositorijum uz pomoć ssh ključa.

The screenshot shows the Jenkins New Item creation page. The search bar contains the text "gitprojekat2". Below the search bar, there are three project types listed:

- Freestyle project**: Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, archiving artifacts and sending email notifications.
- Maven project**: Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration required.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (pipelines) and/or organizing complex activities that do not easily fit in free-style job type.

Koristicu na githubu neki prosti primer gde imam samo jedan readme file. Uzeću njegovu ssh konekciju i to će iskopirati.

Jenkins-proba

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Pin Unwatch 1 Fork

main 1 Branch 0 Tags

Go to file + <> Code

Local

Clone

HTTPS SSH GitHub CLI

git@github.com:djmladenovic/Jenkins-proba.git

Use a password-protected SSH key.

Open with GitHub Desktop

Download ZIP

About

No description

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

U postavkama iz priloženog obratite pažnju na branch koji postavljate. Postavljen je master koji morate promeniti na main. Naravno zavisi gde je željeni program koji želite da preuzmete na Jenkinsu tj. Na kojij grani se nalazi željena verzija .

Source Code Management

None

Git

Repositories

Repository URL

https://github.com/djmladenovic/Jenkins-proba.git

Credentials

jenkins (privatni ključ)

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

/main

Save Apply

ssh konekciju sa githuba postavimo ovde

obratite pažnju da ovde master granu promenite na main

The screenshot shows the Jenkins job configuration interface for a job named "gitprojekat2". In the "Repository" section, the "Repository URL" is set to "https://github.com/djmladenovic/Jenkins-proba.git". Under "Credentials", the dropdown menu shows "jenkins (privatni ključ)" selected. A red box highlights this selection with the annotation "predhodno definisam privatni ključ" (I previously defined a private key). Below this, the "Branches to build" section shows a "Branch Specifier (blank for 'any')" set to "\*/\*". A red box highlights this field with the annotation "učestalost koda pregleda sa githuba" (frequency of code review from GitHub).

Dodali smo i opciju ( \* \* \* \* \* ) sa kojom možemo pratimo promene koje se dešavaju na GitHubu i njih bildujemo. Učestalost zavisi od nas i našeg podešavanja.

The screenshot shows the Jenkins job configuration interface for a job named "gitprojekat2". In the "Build Triggers" section, several options are listed: "Trigger builds remotely (e.g., from scripts)", "Build after other projects are built", "Build periodically", "GitHub Branches", "GitHub Pull Requests", and "GitHub hook trigger for GITScm polling". The "Poll SCM" option is checked, and its schedule is set to "\*/\* \* \* \* \*". A red box highlights this schedule with the annotation "učestalost koda pregleda sa githuba" (frequency of code review from GitHub). Below the schedule, a warning message states: "⚠ Spread load evenly by using 'H/5 \* \* \* \*' rather than '\*/\* \* \* \* \*'. Would last have run at Sunday, May 26, 2024 at 2:45:11 AM Pacific Daylight Time; would next run at Sunday, May 26, 2024 at 2:50:11 AM Pacific Daylight Time." A red box highlights this warning with the annotation "učestalost koda pregleda sa githuba" (frequency of code review from GitHub). At the bottom of the "Build Triggers" section, there is an unchecked checkbox for "Ignore post-commit hooks".

I kao što vidimo bild radi.

Console Output

## Console Output

```
Started by user Djordje Mladenovic
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/gitprojekat2
The recommended git tool is: NONE
using credential kljuc
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/gitprojekat2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/djmladenovic/Jenkins-proba.git # timeout=10
Fetching upstream changes from https://github.com/djmladenovic/Jenkins-proba.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
using GIT_SSH to set credentials privatni kljuc
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
> git fetch --tags --progress -- https://github.com/djmladenovic/Jenkins-proba.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 3747524ed59d77092a1d1a893ba91f98788595f3 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 3747524ed59d77092a1d1a893ba91f98788595f3 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 3747524ed59d77092a1d1a893ba91f98788595f3 # timeout=10
Finished: SUCCESS
```