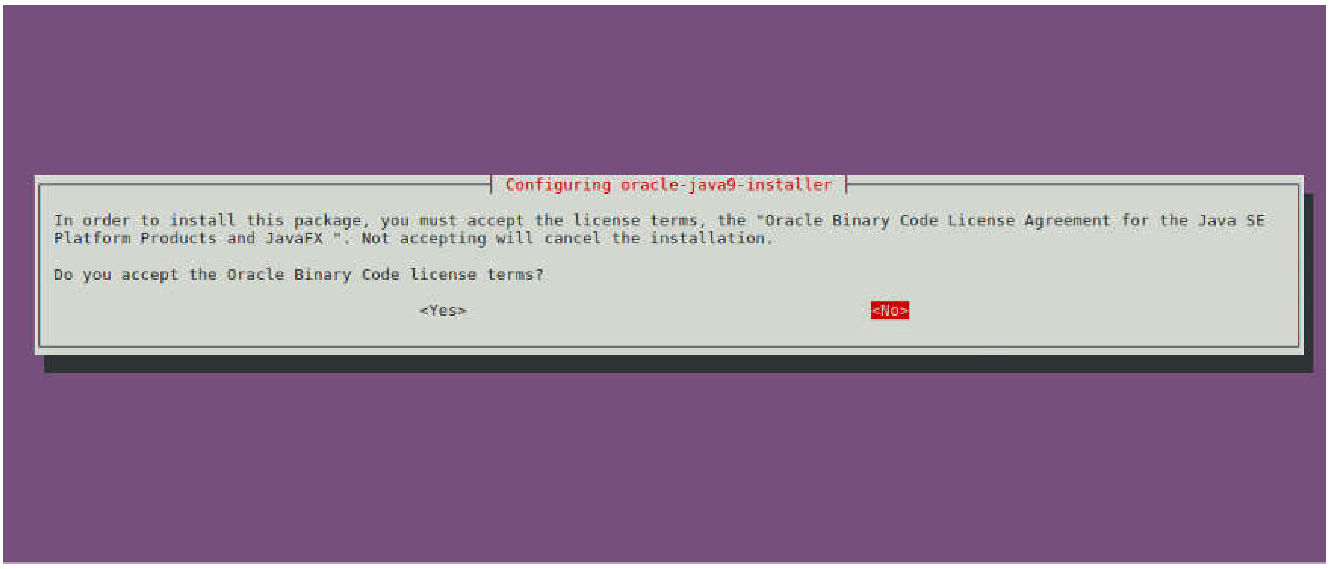
Install Java

Since Jenkins is a Java application, you'll need Java JDK installed. to install OpenJDK, run the commands below…

```
sudo apt update
sudo apt install openjdk-8-jdk
```

Currently, Jenkins might not be fully compatible with Java JDK 9, 10 or 11. For now stay with Java 8 when using Jenkins

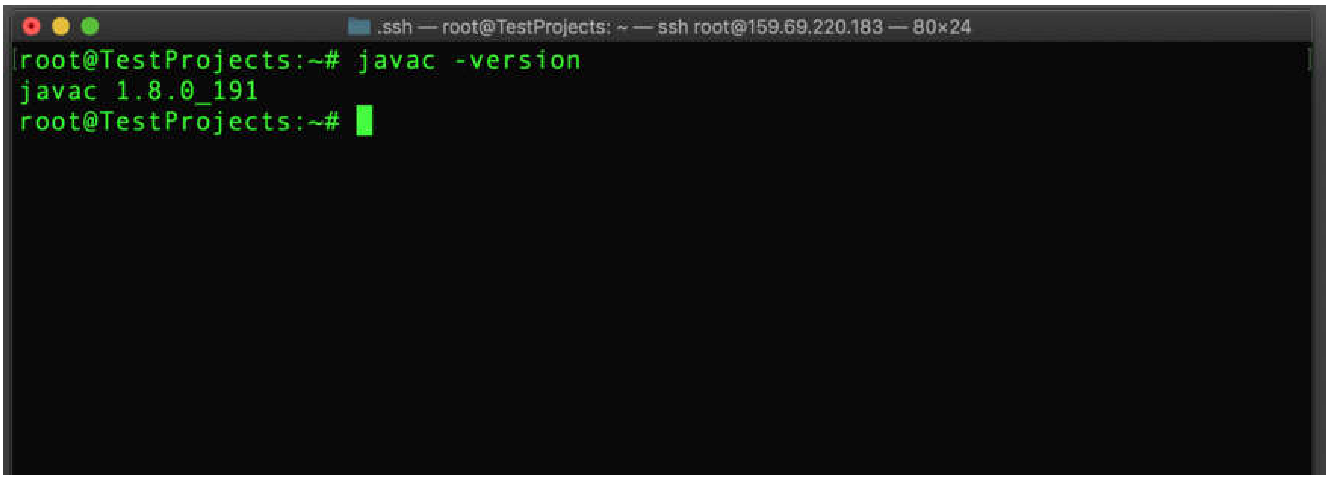


```
Configuring oracle-java9-installer

In order to install this package, you must accept the license terms, the "Oracle Binary Code License Agreement for the Java SE
Platform Products and JavaFX ". Not accepting will cancel the installation.

Do you accept the Oracle Binary Code license terms?

                    <Yes>                                  <No>
```

While you see this screen it may blink press the left arrow and select yes and press the enter key.

You can check you java version by running following command.

```
javac -version
```



.ssh — root@TestProjects: ~ — ssh root@159.69.220.183 — 80×24

```
root@TestProjects:~# javac -version
javac 1.8.0_191
root@TestProjects:~#
```

Install Jenkins

Now that Java is installed, follow the guide below to *install Jenkins*, First run the commands below to add Jenkins repository to your system… First add the repository key…

```
cd /tmp && wget -q -O - https://pkg.jenkins.io/debian-
stable/jenkins.io.key | sudo apt-key add -
```

run the commands below to add the repository

```
echo 'deb https://pkg.jenkins.io/debian-stable binary/' | sudo tee -a
/etc/apt/sources.list.d/jenkins.list
```

After that, run the commands below to install Jenkins

```
sudo apt update
sudo apt install jenkins
```

After installing Jenkins, the commands below can be used to stop, start and enable Jenkins to always start up when the server boots

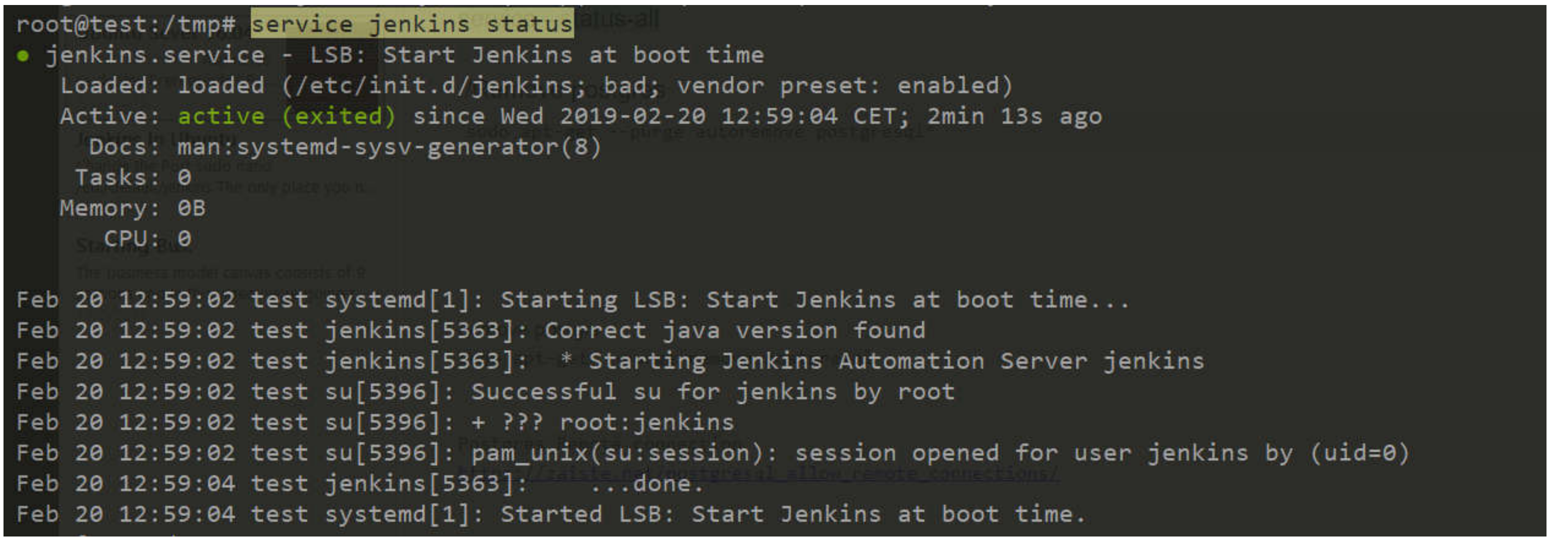
```
sudo systemctl stop jenkins.service
sudo systemctl start jenkins.service
sudo systemctl enable jenkins.service
```

Note: Check if the service is running or not

```
service jenkins status
```

Show all running services by running the following command

```
service --status-all
```

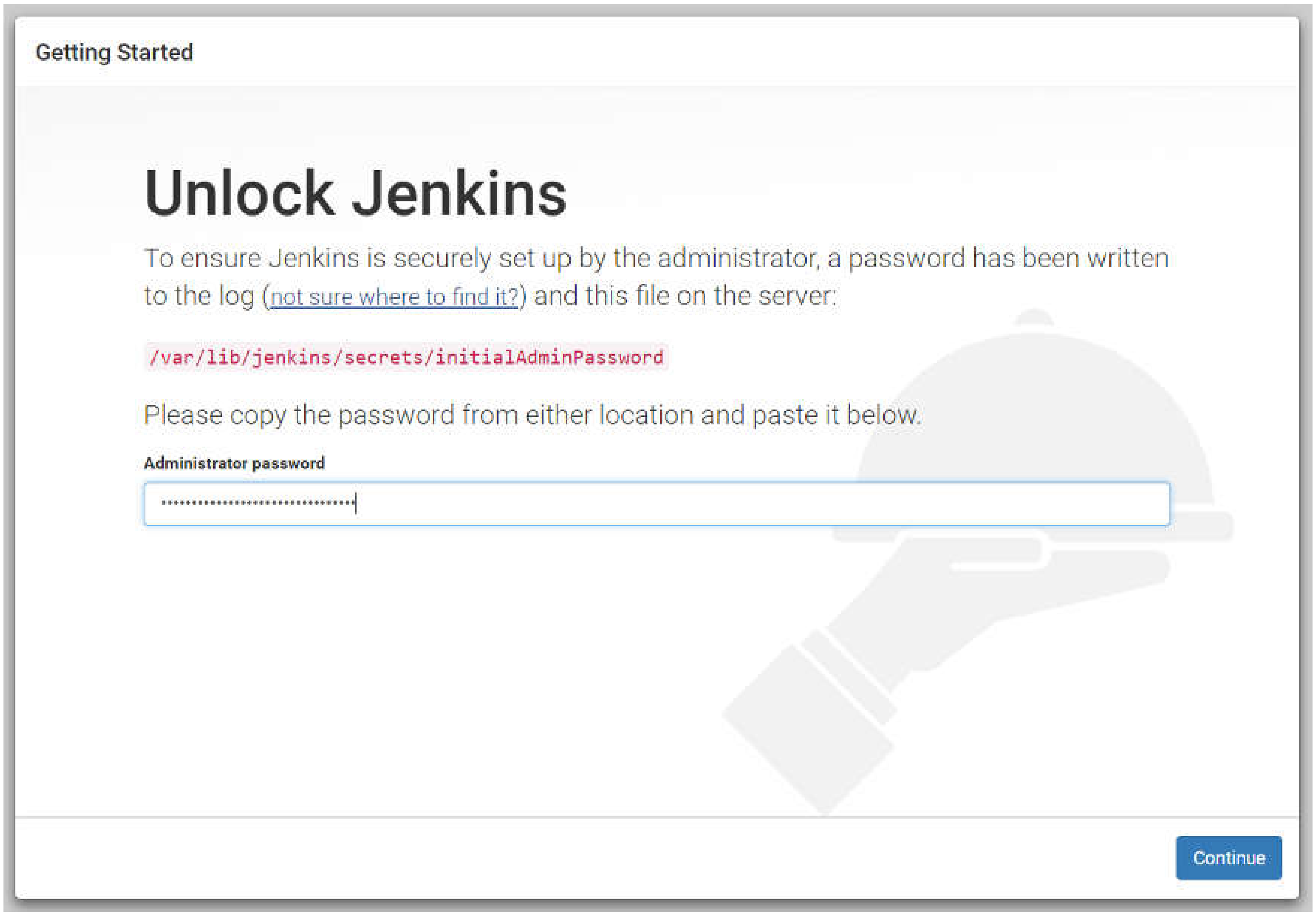


```
root@test:/tmp# service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)
   Active: active (exited) since Wed 2019-02-20 12:59:04 CET; 2min 13s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0
   Memory: 0B
      CPU: 0

Feb 20 12:59:02 test systemd[1]: Starting LSB: Start Jenkins at boot time...
Feb 20 12:59:02 test jenkins[5363]: Correct java version found
Feb 20 12:59:02 test jenkins[5363]:  * Starting Jenkins Automation Server jenkins
Feb 20 12:59:02 test su[5396]: Successful su for jenkins by root
Feb 20 12:59:02 test su[5396]: + ??? root:jenkins
Feb 20 12:59:02 test su[5396]: pam_unix(su:session): session opened for user jenkins by (uid=0)
Feb 20 12:59:04 test jenkins[5363]:    ...done.
Feb 20 12:59:04 test systemd[1]: Started LSB: Start Jenkins at boot time.
```

Next, open your browser and browse to the server hostname or IP address followed by port # **8080**

http://localhost:8080

http://IPAddress:8080 (Maybe Ip address your remote matchine)



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

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

Please copy the password from either location and paste it below.

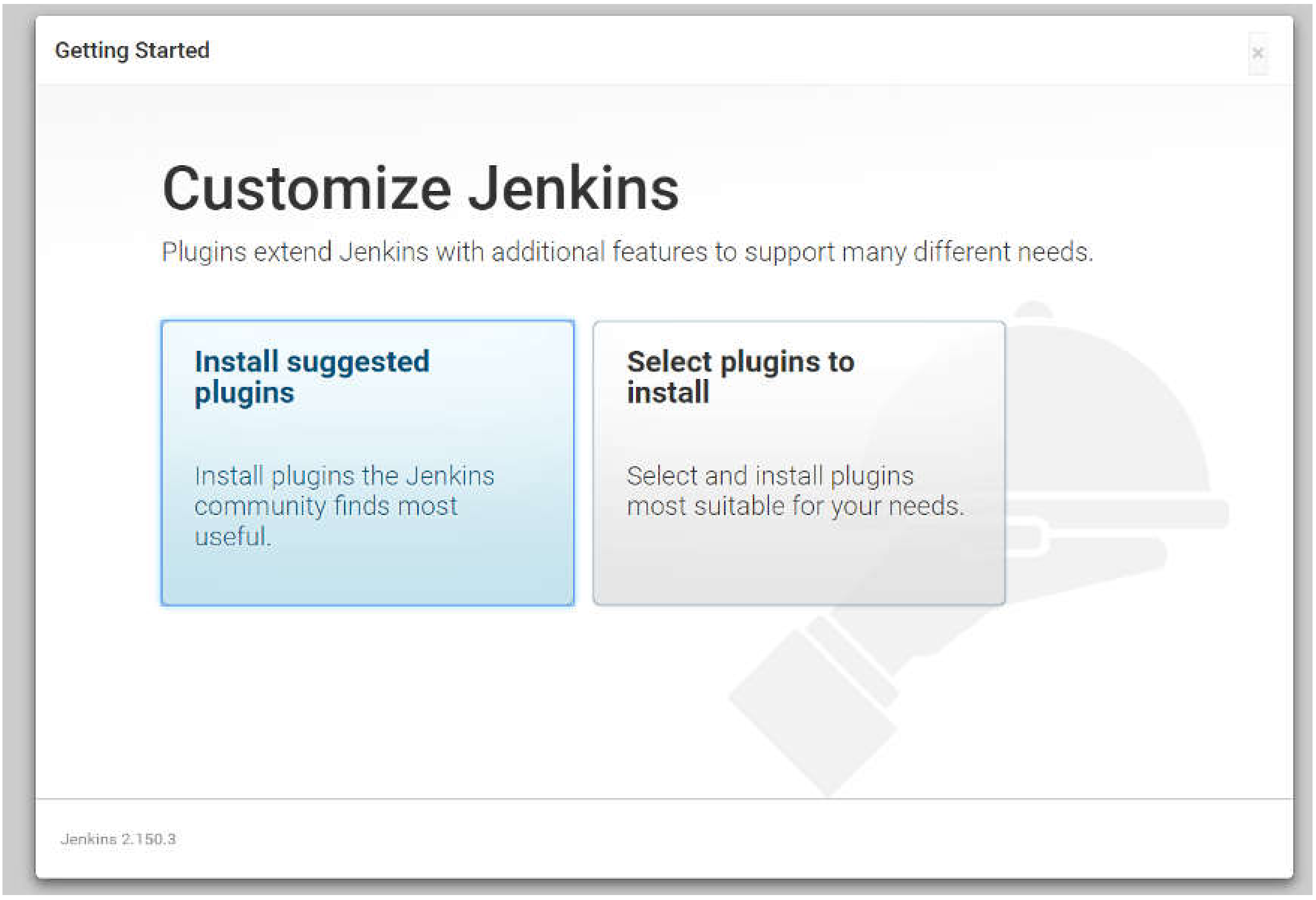
Administrator password

••••••••••••••••••••••••••••••••

When you that, you'll get a prompt to enter the **initial admin password**... run the commands below to view it on the system

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

Select the customized one it is most recent setting. It will install some dependencies for you.



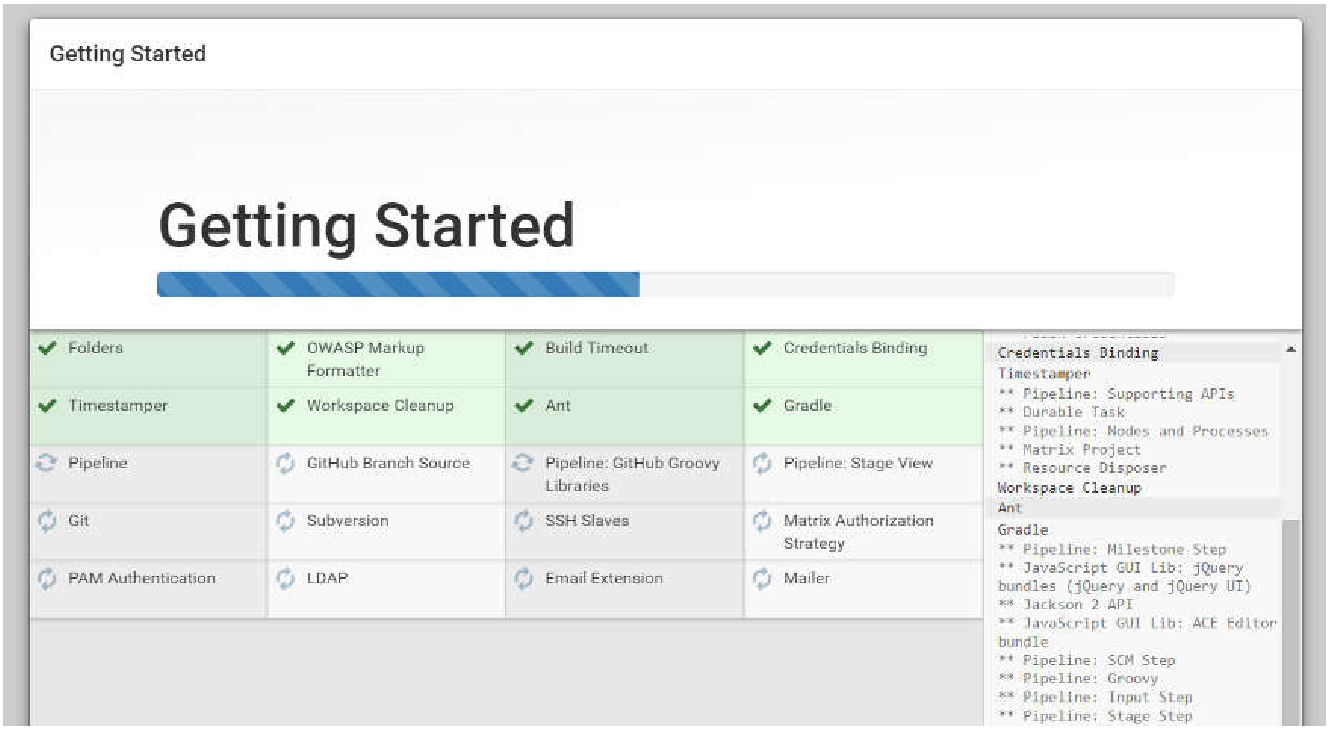
Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins Install plugins the Jenkins community finds most useful.	**Select plugins to install** Select and install plugins most suitable for your needs.

Jenkins 2.150.3

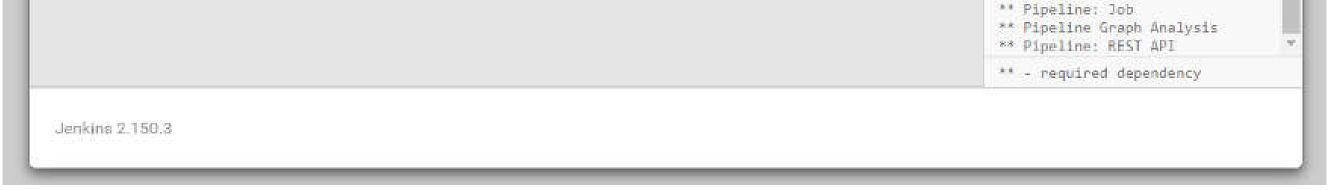


Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	Subversion	SSH Slaves	Matrix Authorization Strategy
PAM Authentication	LDAP	Email Extension	Mailer

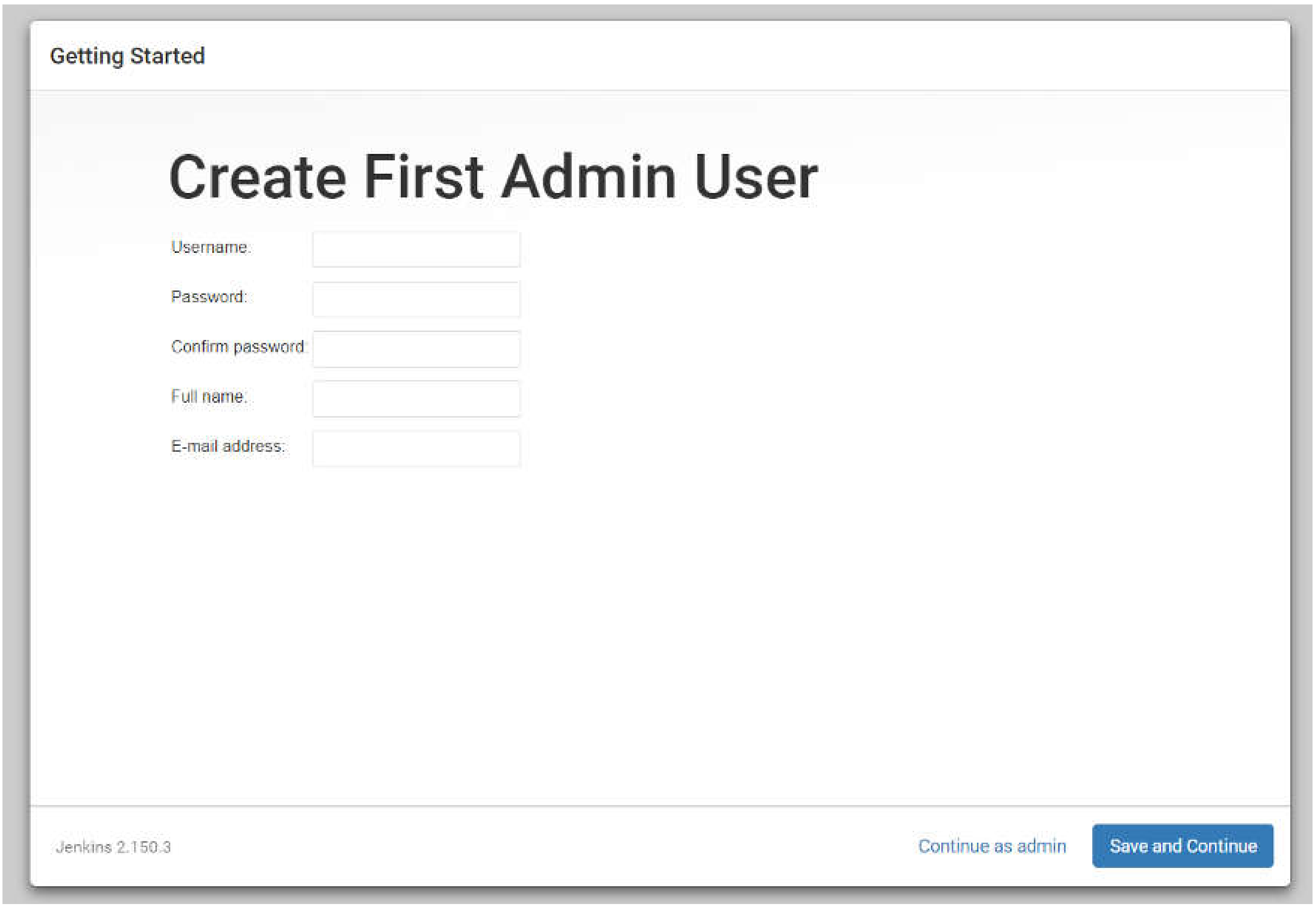
```
Credentials Binding
Timestamper
** Pipeline: Supporting APIs
** Durable Task
** Pipeline: Nodes and Processes
** Matrix Project
** Resource Disposer
Workspace Cleanup
Ant
Gradle
** Pipeline: Milestone Step
** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI)
** Jackson 2 API
** JavaScript GUI Lib: ACE Editor bundle
** Pipeline: SCM Step
** Pipeline: Groovy
** Pipeline: Input Step
** Pipeline: Stage Step
```



```
** Pipeline: Job
** Pipeline Graph Analysis
** Pipeline: REST API

** - required dependency
```

Jenkins 2.150.3

Register your account



Getting Started

Create First Admin User

Username:

Password:

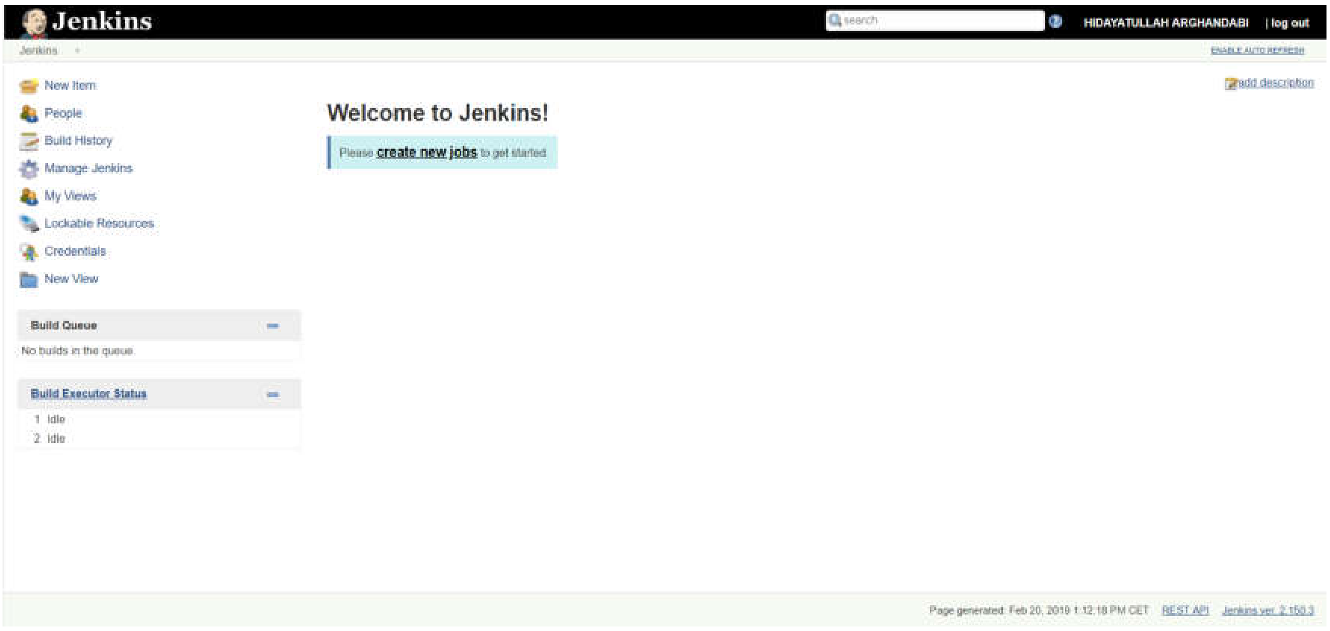
Confirm password:

Full name:

E-mail address:

Jenkins 2.150.3

After this it will be ready … to use



Welcome to Jenkins!

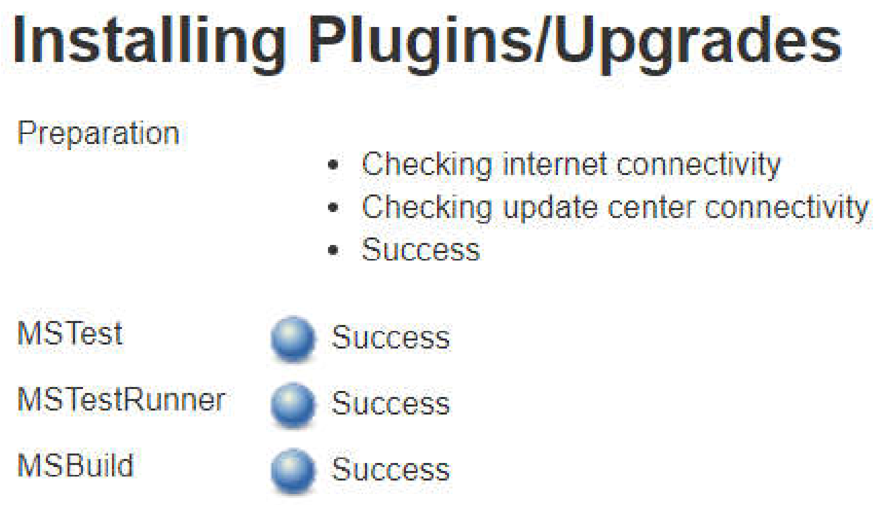
Please **create new jobs** to get started

No builds in the queue.

1 Idle
2 Idle

Page generated: Feb 20, 2019 1:12:18 PM CET REST API Jenkins ver. 2.150.3

Go to the **Manage Jenkins** and then **Plugin Manger** and Install the following packages `msBuild`, `msTest` and `msTest Runner`.


Installing Plugins/Upgrades

Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

MSTest	Success
MSTestRunner	Success
MSBuild	Success

Frequently Asked Questions

Change the Jenkins Port

Open the Jenkins setting

```
sudo nano /etc/default/jenkins
```

The only place you need to change is:

```
#port for HTTP connector (default 8080; disable with -1)
Http_port = 8080
```

There you change to the desired port. For example:

```
HTTP_PORT = 8081
```

Finally, restart Jenkins with the following command:

```
sudo service jenkins restart
```

Install the Prerequisites

Register Microsoft key and feed

```
wget -q https://packages.microsoft.com/config/ubuntu/16.04/packages-
microsoft-prod.deb

sudo dpkg -i packages-microsoft-prod.deb
```

Install the .NET SDK

```
sudo apt-get install apt-transport-https

sudo apt-get update

sudo apt-get install dotnet-sdk-2.2
```

To confirm your installation and to check the version of dotnet cli installed on the machine type the following command. You should get an output

```
dotnet —-version
```

Install NuGet Packing

We have dependencies in our project for that we need to install the NuGet Packing CLI command.

```
sudo apt install nuget
```

Install Nginx

NGINX is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability.

```
sudo apt-get install nginx
```

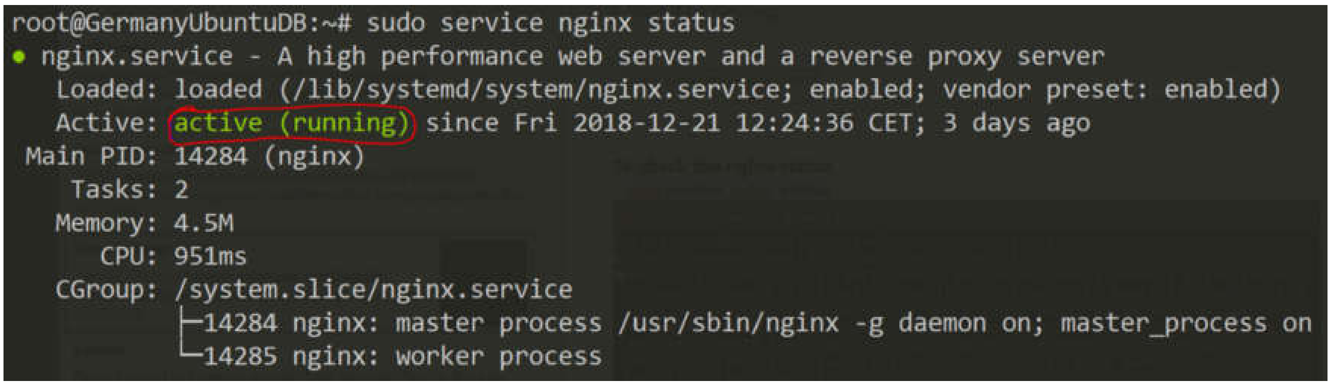
Start Nginx Server

```
sudo service nginx start
```

Server Status: Check the server status if running

```
sudo service nginx status
```

You can see that the server is active



```
root@GermanyUbuntuDB:~# sudo service nginx status
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2018-12-21 12:24:36 CET; 3 days ago
 Main PID: 14284 (nginx)
    Tasks: 2
   Memory: 4.5M
      CPU: 951ms
   CGroup: /system.slice/nginx.service
           ├─14284 nginx: master process /usr/sbin/nginx -g daemon on; master_process on
           └─14285 nginx: worker process
```

Configure Nginx

To configure Nginx as a reverse proxy to forward requests to our ASP.NETCore app. Modify (nano) /etc/nginx/sites-available/default.

```
nano /etc/nginx/sites-available/default
```

Or Open it in a text editor, and replace the contents with the following: This Nginx configuration file forwards incoming public traffic from port 80 to port 5000.

```
server {

listen 80;

location / {

proxy_pass http://localhost:5000;

proxy_http_version 1.1;

proxy_set_header Upgrade $http_upgrade;

proxy_set_header Connection keep-alive;

proxy_set_header Host $http_host;

proxy_cache_bypass $http_upgrade;

}

}
```

> *NOTICE: The localhost port can change upon your project*

After the modification we need to verify the syntax of the configuration file

```
sudo nginx -t
```

If the configuration file test is successful, force Nginx to pick up the changes by running

```
sudo nginx -s reload
```

Or start it with:

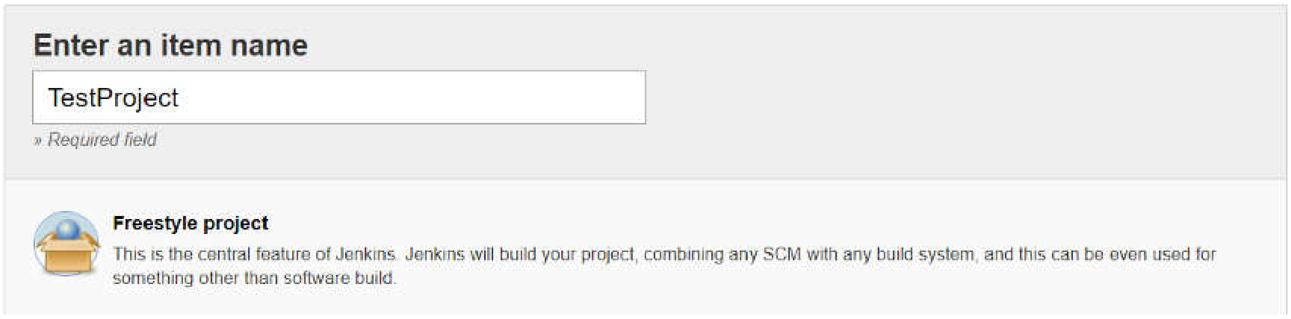
```
sudo service nginx start
```

Now your go to your nginx server adress and you see that you application has started working.

Install NuGet Package Management Tools

```
sudo apt install nuget
```

Create new (a Jenkins Freestyle Project)



Enter an item name

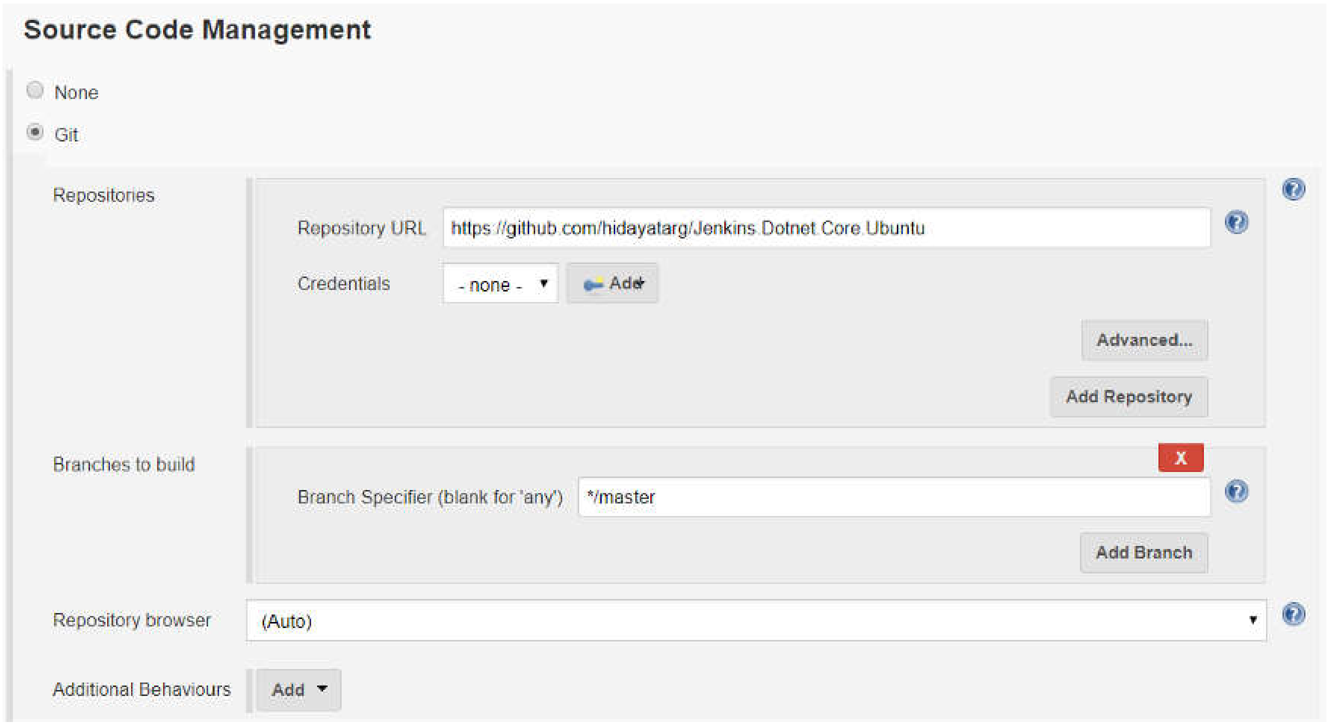
TestProject

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

In the source code management choose git (Put the **credential** if you need)



Source Code Management

None

Git

Repositories

Repository URL: https://github.com/hidayatarg/Jenkins.Dotnet.Core.Ubuntu

Credentials: - none -

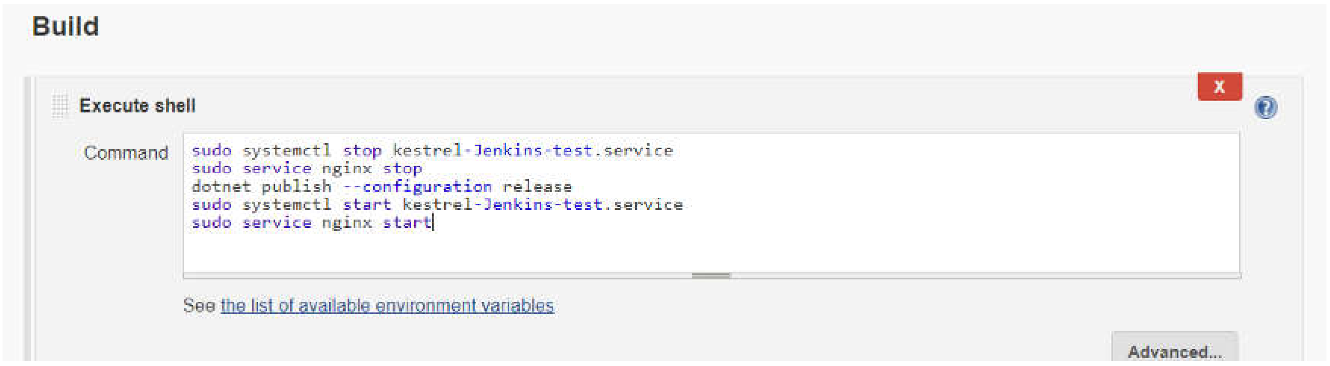
Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Additional Behaviours

and



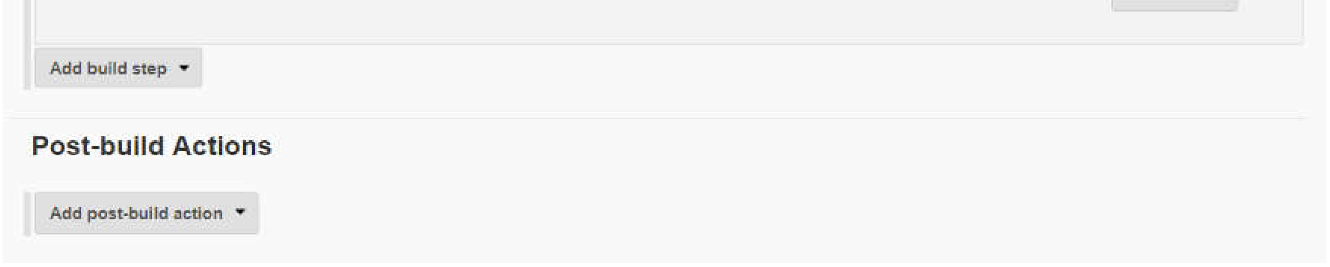
Build

Execute shell

Command

```
sudo systemctl stop kestrel-Jenkins-test.service
sudo service nginx stop
dotnet publish --configuration release
sudo systemctl start kestrel-Jenkins-test.service
sudo service nginx start
```

See the list of available environment variables



then save the changes and build the task.

In case error:

```
sudo: no tty present and no askpass program specified
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

Solution

Running shell scripts that have contain sudo commands in them from jenkins might not run as expected. To fix this, follow along Simple steps:

1. On Ubuntu based systems, run `sudo visudo`
2. It will open `/etc/sudoers` file.
3. If your Jenkins user is already in that file, then modify to look like this: `jenkins ALL=(ALL) NOPASSWD: ALL`
4. save the file by doing Ctrl+O (dont save in tmp file. save in /etc/sudoers, confirm overwrite)
5. Exit by doing Ctrl+X
6. Relaunch your jenkins job
7. You shouldnt see that error message again :)

(Special thanks: Imran Haydar)

Find the application deployment place

Nginx Configuration

Create the service file

```
sudo nano /etc/systemd/system/kestrel-Jenkins-test.service
```

and paste

```
[Unit]

Description=Example .NET Web API App running on Ubuntu

[Service]

WorkingDirectory=/var/lib/jenkins/workspace/TestProject/JenkinsTest

ExecStart=/usr/bin/dotnet
/var/lib/jenkins/workspace/TestProject/JenkinsTest/bin/Release/netcoreap

Restart=always

# Restart service after 10 seconds if the dotnet service crashes:

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=dotnet-example

User=www-data

Environment=ASPNETCORE_ENVIRONMENT=Production

Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false

[Install]

WantedBy=multi-user.target
```

Register the service

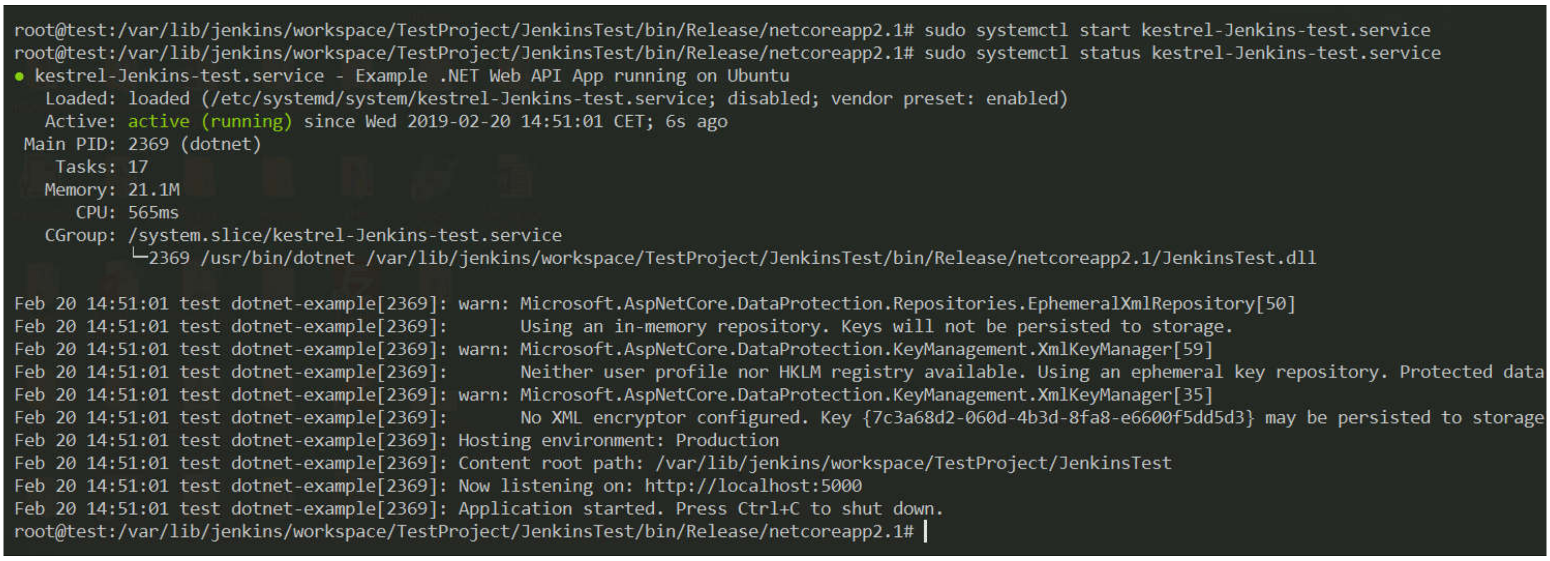
```
sudo systemctl enable kestrel-Jenkins-test.service
```

Start the service and verify that it’s running.

```
sudo systemctl start kestrel-Jenkins-test.service

sudo systemctl start kestrel-Jenkins-test.service
```

To Stop the service

```
sudo systemctl stop kestrel-Jenkins-test.service
```



```
root@test:/var/lib/jenkins/workspace/TestProject/JenkinsTest/bin/Release/netcoreapp2.1# sudo systemctl start kestrel-Jenkins-test.service
root@test:/var/lib/jenkins/workspace/TestProject/JenkinsTest/bin/Release/netcoreapp2.1# sudo systemctl status kestrel-Jenkins-test.service
● kestrel-Jenkins-test.service - Example .NET Web API App running on Ubuntu
   Loaded: loaded (/etc/systemd/system/kestrel-Jenkins-test.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-02-20 14:51:01 CET; 6s ago
 Main PID: 2369 (dotnet)
    Tasks: 17
   Memory: 21.1M
      CPU: 565ms
   CGroup: /system.slice/kestrel-Jenkins-test.service
           └─2369 /usr/bin/dotnet /var/lib/jenkins/workspace/TestProject/JenkinsTest/bin/Release/netcoreapp2.1/JenkinsTest.dll

Feb 20 14:51:01 test dotnet-example[2369]: warn: Microsoft.AspNetCore.DataProtection.Repositories.EphemeralXmlRepository[50]
Feb 20 14:51:01 test dotnet-example[2369]:       Using an in-memory repository. Keys will not be persisted to storage.
Feb 20 14:51:01 test dotnet-example[2369]: warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[59]
Feb 20 14:51:01 test dotnet-example[2369]:       Neither user profile nor HKLM registry available. Using an ephemeral key repository. Protected data
Feb 20 14:51:01 test dotnet-example[2369]: warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
Feb 20 14:51:01 test dotnet-example[2369]:       No XML encryptor configured. Key {7c3a68d2-060d-4b3d-8fa8-e6600f5dd5d3} may be persisted to storage
Feb 20 14:51:01 test dotnet-example[2369]: Hosting environment: Production
Feb 20 14:51:01 test dotnet-example[2369]: Content root path: /var/lib/jenkins/workspace/TestProject/JenkinsTest
Feb 20 14:51:01 test dotnet-example[2369]: Now listening on: http://localhost:5000
Feb 20 14:51:01 test dotnet-example[2369]: Application started. Press Ctrl+C to shut down.
root@test:/var/lib/jenkins/workspace/TestProject/JenkinsTest/bin/Release/netcoreapp2.1# |
```

Check the server logs

```
sudo journalctl -fu kestrel-helloapp.service
```

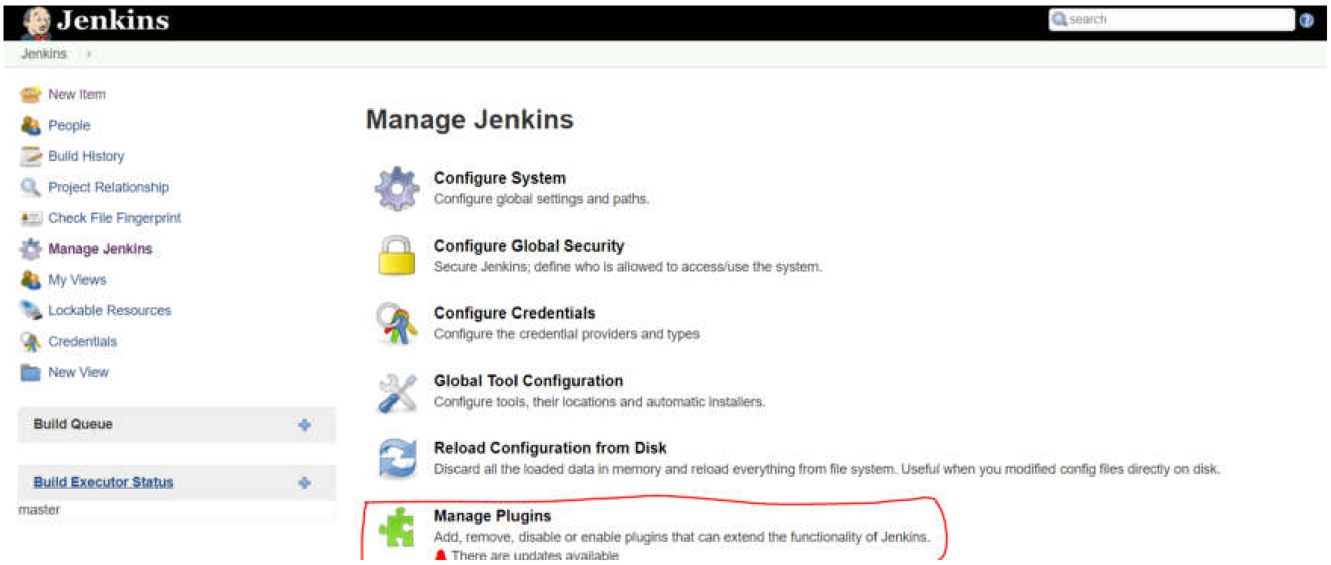
Your application is successfully running in Nginx server.

Now if you update you repository, Sign in to the Jenkins and deploy the recent version with one-click.

You can also make the git repository to alert Jenkins to deploy, whenever new commit are placed.

Install The GitHub Extension to Jenkins Server

Go to Manage Jenkins > Manage Plugins



Manage Jenkins

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

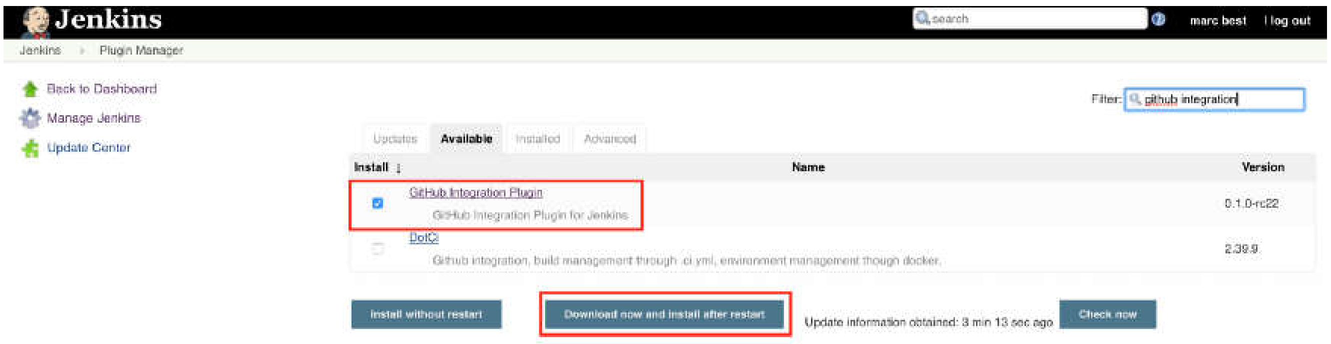
Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
There are updates available

System Information
Displays various environmental information to assist trouble-shooting.

Install GitHub Integration Plugin



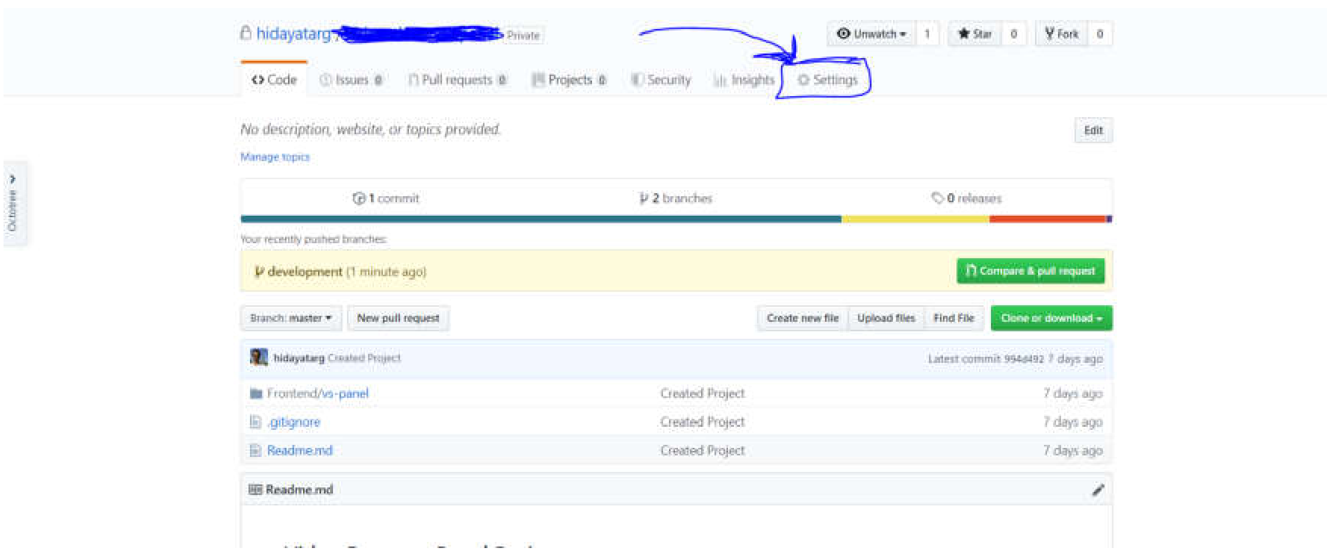
Filter: github integration

Install	Name	Version
✓	GitHub Integration Plugin GitHub Integration Plugin for Jenkins	0.1.0-rc22
☐	DotCi Github integration, build management through .ci.yml, environment management though docker.	2.39.9

Update information obtained: 3 min 13 sec ago

GitHub Hook

Go to repository setting



hidayatarg/[redacted] Private

No description, website, or topics provided.

1 commit · 2 branches · 0 releases

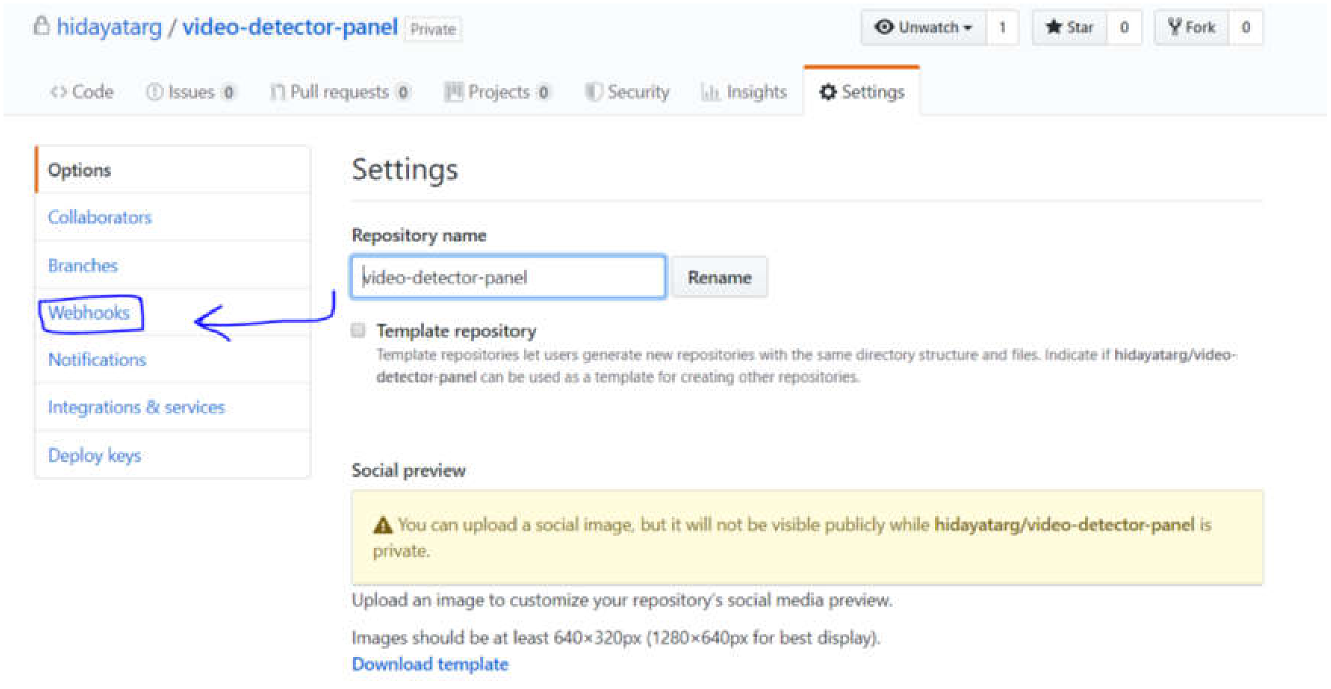
Your recently pushed branches:

development (1 minute ago)

hidayatarg Created Project — Latest commit 994d492 7 days ago

Frontend/vs-panel	Created Project	7 days ago
.gitignore	Created Project	7 days ago
Readme.md	Created Project	7 days ago

Readme.md



hidayatarg / video-detector-panel Private

Settings

Repository name: video-detector-panel

☐ Template repository
Template repositories let users generate new repositories with the same directory structure and files. Indicate if **hidayatarg/video-detector-panel** can be used as a template for creating other repositories.

Social preview

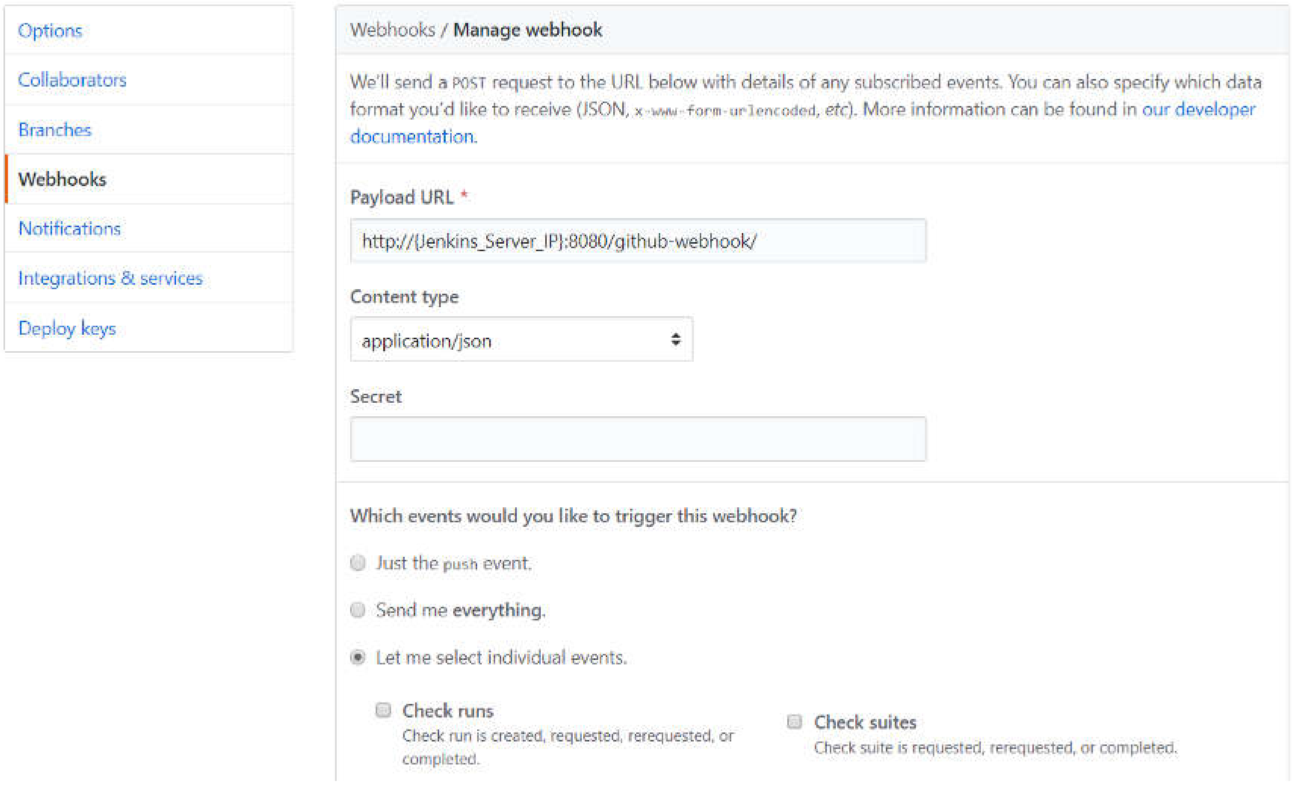
You can upload a social image, but it will not be visible publicly while **hidayatarg/video-detector-panel** is private.

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Download template

Click the Add webhook



Options
Collaborators
Branches
Webhooks
Notifications
Integrations & services
Deploy keys

Webhooks / **Manage webhook**

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *
http://{Jenkins_Server_IP}:8080/github-webhook/

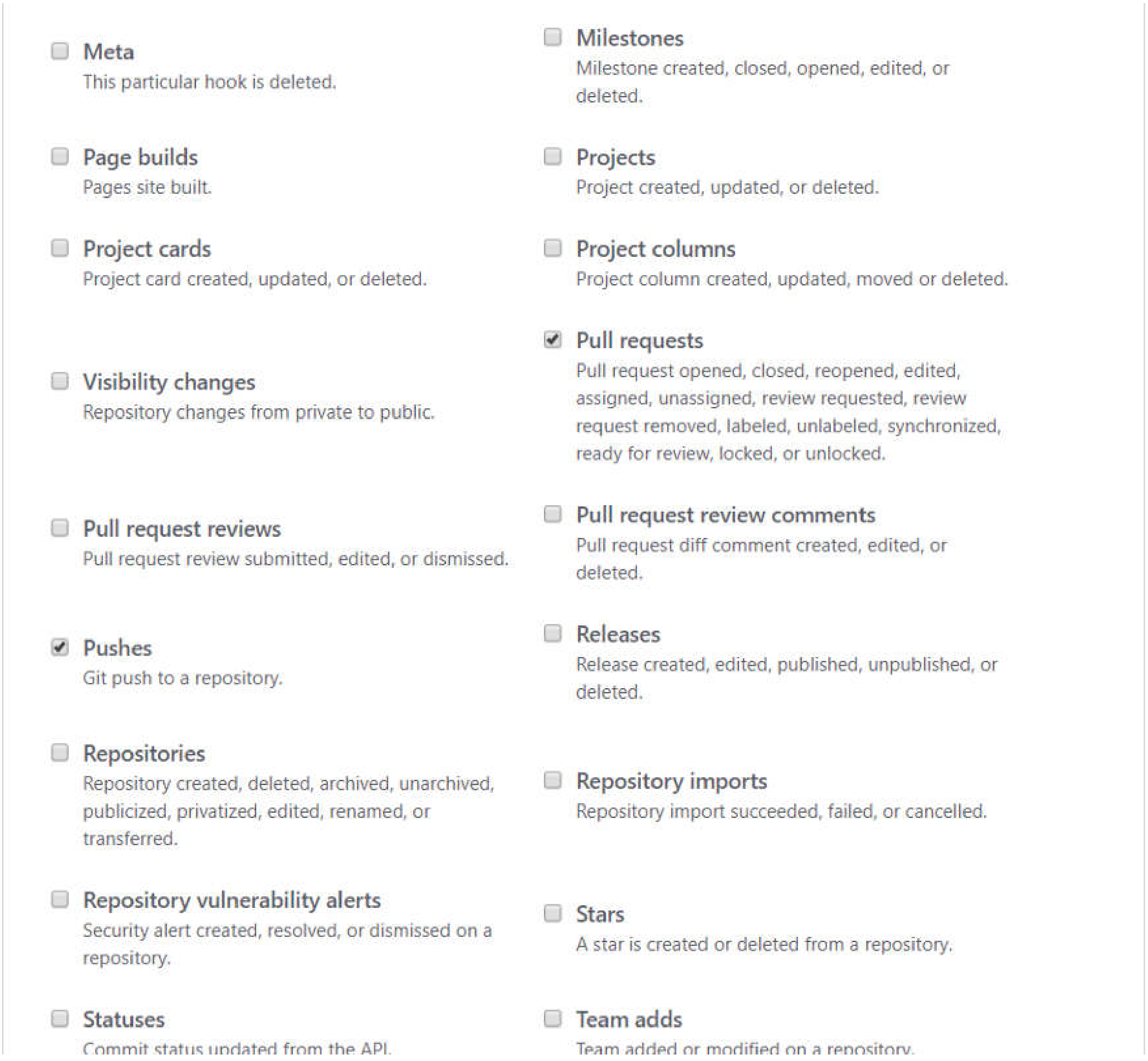
Content type
application/json

Secret

Which events would you like to trigger this webhook?

- [ ] Just the `push` event.
- [ ] Send me **everything**.
- [✓] Let me select individual events.

- [ ] **Check runs** Check run is created, requested, rerequested, or completed.
- [ ] **Check suites** Check suite is requested, rerequested, or completed.



- [ ] **Meta** This particular hook is deleted.
- [ ] **Milestones** Milestone created, closed, opened, edited, or deleted.
- [ ] **Page builds** Pages site built.
- [ ] **Projects** Project created, updated, or deleted.
- [ ] **Project cards** Project card created, updated, or deleted.
- [ ] **Project columns** Project column created, updated, moved or deleted.
- [ ] **Visibility changes** Repository changes from private to public.
- [✓] **Pull requests** Pull request opened, closed, reopened, edited, assigned, unassigned, review requested, review request removed, labeled, unlabeled, synchronized, ready for review, locked, or unlocked.
- [ ] **Pull request reviews** Pull request review submitted, edited, or dismissed.
- [ ] **Pull request review comments** Pull request diff comment created, edited, or deleted.
- [✓] **Pushes** Git push to a repository.
- [ ] **Releases** Release created, edited, published, unpublished, or deleted.
- [ ] **Repositories** Repository created, deleted, archived, unarchived, publicized, privatized, edited, renamed, or transferred.
- [ ] **Repository imports** Repository import succeeded, failed, or cancelled.
- [ ] **Repository vulnerability alerts** Security alert created, resolved, or dismissed on a repository.
- [ ] **Stars** A star is created or deleted from a repository.
- [ ] **Statuses** Commit status updated from the API.
- [ ] **Team adds** Team added or modified on a repository.

Commit status updated from the API.
Team added or modified on a repository.

After this process GitHub will send a hook with each commit to the repository. This is how we implemented the CI and CD.