

# Ecosystem

For more information contact:

- Matt Harasymczuk
- [ecosystem@haras.pl](mailto:ecosystem@haras.pl)
- <https://github.com/MattAgile/ecosystem-workshop>

- Environment Setup
  - Server Access
  - Environment Setup
  - Locale
    - TODO
- PostgreSQL
  - Download Page
  - Installation
  - Configuration
    - `/etc/postgresql/9.3/main/pg_hba.conf`
    - `/etc/postgresql/9.3/main/postgresql.conf`
- Artifactory
  - Documentation
  - Download Page
  - Installation
  - API Documentation
- Jira
  - Documentation
  - Download Page
  - Installation
    - Pre Install
    - Install
  - API Documentation
  - JIRA User Server
- Confluence
  - Documentation
  - Download Page
  - Installation
    - Pre Install
    - Install
  - API Documentation
  - Set JIRA User Directory
- Jenkins
  - Documentation
  - Download Page
  - Installation
    - Pre Install
    - Install
    - Post Install
  - API Documentation
  - Configuration
  - Connect Jenkins with Stash
  - Pull Request Build Configuration
  - Master Branch Build Configuration
  - Feature Branch Build Configuration
  - Bugfix Branch Build Configuration
- Sputnik
  - Download Page
  - Configuration
    - `pom.xml`
    - Jenkins Job Configuration Pre Step - Execute Shell
- Stash
  - Documentation
  - Download Page
  - Installation
    - Pre Install
    - Install
  - API Documentation
  - Configuration

- [Set JIRA User Directory](#)
- [Sonar](#)
  - [Documentation](#)
  - [Download Page](#)
  - [Installation](#)
    - [Pre Install](#)
    - [Install](#)
    - [Post Install](#)
  - [API Documentation](#)
  - [Demo Project](#)

## Environment Setup

Host IP address 52.28.55.156

## Server Access

```
ssh -i workshop.pem -l ubuntu HOST_IP_ADDRESS
```

## Environment Setup

```
apt-get install --yes git vim nmap htop wget curl unzip
```

## Locale

```
echo 'LANG="en_US.UTF-8"' >> /etc/default/locale
echo 'LC_ALL="en_US.UTF-8"' >> /etc/default/locale
echo 'LANG="en_US.UTF-8"' >> /etc/default/locale
locale-gen en_US.UTF-8
dpkg-reconfigure locales
```

## TODO

- ☐ Enable password login in `/etc/ssh/sshd_config` (easy for windows users with putty)
- ☐ `sudo passwd ubuntu`

## PostgreSQL

Host IP address 52.28.55.156

You can access PostgreSQL at 5432

# Documentation

- <http://www.postgresql.org/docs/9.3/static/index.html>

## Download Page

- <http://www.postgresql.org/download/>

## Installation

```
apt-get install --yes postgresql
```

## Configuration

### /etc/postgresql/9.3/main/pg\_hba.conf

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	local	all	postgres		peer
	local	all	all		peer
	host	all	all	127.0.0.1/32	md5
	host	all	all	0.0.0.0/0	md5
	host	all	all	:::1/128	md5

### /etc/postgresql/9.3/main/postgresql.conf

```
data_directory = '/var/lib/postgresql/9.3/main'
hba_file = '/etc/postgresql/9.3/main/pg_hba.conf'
ident_file = '/etc/postgresql/9.3/main/pg_ident.conf'
external_pid_file = '/var/run/postgresql/9.3-main.pid'
listen_addresses = '*'
port = 5432
max_connections = 100
unix_socket_directories = '/var/run/postgresql'
ssl = true
ssl_cert_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem'
ssl_key_file = '/etc/ssl/private/ssl-cert-snakeoil.key'
shared_buffers = 128MB
log_line_prefix = '%t '
log_timezone = 'UTC'
datestyle = 'iso, mdy'
timezone = 'UTC'
lc_messages = 'en_US.UTF-8'
lc_monetary = 'en_US.UTF-8'
lc_numeric = 'en_US.UTF-8'
lc_time = 'en_US.UTF-8'
default_text_search_config = 'pg_catalog.english'
```

### Restart PostgreSQL Server

```
service postgresql restart
```

## Artifactory

Host IP address 52.28.55.156

You can access Artifactory at port 8081

## Documentation

- <https://www.jfrog.com/confluence/display/RTF/Artifactory+User+Guide>

## Download Page

- <http://www.jfrog.com/open-source/>

## Installation

```
useradd artifactory

cd /opt/
wget http://dl.bintray.com/jfrog/artifactory/artifactory-3.3.1.zip

unzip artifactory-3.3.1.zip
rm -fr artifactory-3.3.1.zip
chown -R artifactory:artifactory artifactory-3.3.1/

cd artifactory-3.3.1/bin/
su artifactory

screen
./artifactory.sh

(detach screen)
```

## API Documentation

- <http://www.jfrog.com/confluence/display/RTF/Artifactory+REST+API>

## Jira

Host IP address 52.28.55.156

You can access JIRA at port 8080

## Documentation

- <https://confluence.atlassian.com/display/JIRA/JIRA+Documentation>

## Download Page

- <https://www.atlassian.com/software/jira/download?b=a#allDownloads>

## Installation

### Pre Install

```
CREATE USER jira WITH PASSWORD 'jira';
CREATE DATABASE jira;
GRANT ALL PRIVILEGES ON DATABASE jira TO jira;
```

### Install

```
wget
https://www.atlassian.com/software/jira/downloads/binary/atlassian-jira-6.4.2-x64.bin
chmod +x atlassian-jira-6.4.2-x64.bin
./atlassian-jira-6.4.2-x64.bin

rm -fr atlassian-jira-6.4.2-x64.bin
echo "jira.websudo.is.disabled = true" >>
/var/atlassian/application-data/jira/jira-config.properties
service jira stop
service jira start
```

## API Documentation

- <https://docs.atlassian.com/jira/REST/latest/>
- <https://jira.atlassian.com/plugins/servlet/restbrowser#/>

## JIRA User Server

1. Go to Jira User Server (g+g and type JIRA User Server)
2. Add application
3. Set application name, password and IP Addresses (paste addresses from instances which you want connect with Jira User Server)

# Confluence

Host IP address 52.28.55.156

You can access Confluence at port 8090

## Documentation

- <https://confluence.atlassian.com/display/DOC/Confluence+Documentation+Home>

## Download Page

- <https://www.atlassian.com/software/confluence/download>

## Installation

### Pre Install

```
CREATE USER confluence WITH PASSWORD 'confluence';
CREATE DATABASE confluence;
GRANT ALL PRIVILEGES ON DATABASE confluence TO confluence;
```

### Install

```
wget
https://www.atlassian.com/software/confluence/downloads/binary/atlassian-confluence-5.7.3-x64.bin
chmod +x atlassian-confluence-5.7.3-x64.bin
./atlassian-confluence-5.7.3-x64.bin
rm -fr atlassian-confluence-5.7.3-x64.bin
```

## API Documentation

- <https://docs.atlassian.com/atlassian-confluence/REST/latest/>
- <https://confluence.atlassian.com/plugins/servlet/restbrowser#/>

## Set JIRA User Directory

1. Go to User Directories
2. Add directory
3. Choose directory type: 'Atlassian JIRA'
4. Set
  - a. directory name
  - b. paste jira url

- c. application name (application name from [Jira User Server](#))
- d. application password (application password from [Jira User Server](#))
- 5. Test connection
- 6. Save configuration
- 7. Synchronize directory

## Jenkins

Host IP address 52.28.55.156

You can access Jenkins at port 8081

## Documentation

- <https://wiki.jenkins-ci.org/display/JENKINS/Use+Jenkins>

## Download Page

- <http://jenkins-ci.org/changelog>

## Installation

### Pre Install

```
wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -  
echo "deb http://pkg.jenkins-ci.org/debian binary/" >> /etc/apt/sources.list  
apt-get update
```

### Install

```
apt-get install --yes jenkins  
sudo su - jenkins  
ssh-keygen  
cat ~/.ssh/id_rsa.pub  
exit
```

### Post Install

```
service jenkins stop  
sed -i 's/HTTP_PORT=8080/HTTP_PORT=8081/g' /etc/default/jenkins  
service jenkins start
```

## API Documentation

- <https://wiki.jenkins-ci.org/display/JENKINS/Remote+access+API>

## Configuration

1. Add Jenkins user pubkey (~/.ssh/id\_rsa.pub) generated during install to Stash repository access keys ([http://HOST\\_IP\\_ADDRESS:7990/plugins/servlet/ssh/projects/ECO/repos/workshop/keys](http://HOST_IP_ADDRESS:7990/plugins/servlet/ssh/projects/ECO/repos/workshop/keys))
2. In Jenkins Select **Credentials** from the menu at the left side
3. Select **Global credentials**
4. **Add Credential**

Key	Value
Kind	SSH Username with private key
Scope	Global
Username	jenkins
Private Key	From the Jenkins master ~/.ssh

## Connect Jenkins with Stash

1. Install Stash Notifier Plugin in Jenkins
2. In **Configure System** - Global Jenkins System Configuration set:

Key	Value
Stash Root Url	<a href="http://HOST_IP_ADDRESS:7990/">http://HOST_IP_ADDRESS:7990/</a>
Stash User	jenkins
Stash Password	jenkins
Keep repeated builds in Stash	True - checked

## Pull Request Build Configuration

Dashboard -> New Item -> wpisujemy project name z ponizszej tabelki i wybieramy np. "Freestyle project"

Section	Key	Value
	Project name	Ecosystem - Pull Request
Source Code Management	Source Code Management	GIT
Source Code Management	Repository URL	<a href="ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git">ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git</a>
Source Code Management	Credentials	jenkins
Source Code Management	[Advanced] -> Refspec	+refs/pull-requests/*/from:refs/remotes/origin/pr/*
Source Code Management	Branch Specifier	**/pr/*
Build Triggers	Schedule	* * * * *
Post-build Actions	Notify Stash Instance	

## Master Branch Build Configuration



Analogicznie - New Item

Section	Key	Value
	Project name	Ecosystem - Master
Source Code Management	Source Code Management	GIT
Source Code Management	Repository URL	<a href="ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git">ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git</a>
Source Code Management	Credentials	jenkins
Source Code Management	[Advanced] -> Refspec	+refs/pull-requests/*/*from:refs/remotes/origin/pr/*
Source Code Management	Branch Specifier	**/*master
Build Triggers	Schedule	* * * * *
Post-build Actions	Notify Stash Instance	

## Feature Branch Build Configuration

Analogicznie - New Item

Section	Key	Value
	Project name	Ecosystem - Feature
Source Code Management	Source Code Management	GIT
Source Code Management	Repository URL	<a href="ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git">ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git</a>
Source Code Management	Credentials	jenkins
Source Code Management	Branch Specifier	*/feature/*
Build Triggers	Schedule	* * * * *
Post-build Actions	Notify Stash Instance	

## Bugfix Branch Build Configuration

Analogicznie - New Item

Section	Key	Value
	Project name	Ecosystem - Bugfix
Source Code Management	Source Code Management	GIT
Source Code Management	Repository URL	<a href="ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git">ssh://git@HOST_IP_ADDRESS:7999/eco/workshop.git</a>
Source Code Management	Credentials	jenkins
Source Code Management	Branch Specifier	*/bugfix/*
Build Triggers	Schedule	* * * * *
Post-build Actions	Notify Stash Instance	

## Sputnik

Host IP address 52.28.55.156

---

## Download Page

- <https://github.com/TouK/sputnik>
- <https://github.com/ingwarsw/sputnik-maven-plugin>

## Configuration

**pom.xml**

```

<profiles>
  <profile>
    <id>sputnik</id>
    <build>
      <plugins>
        <plugin>
          <groupId>org.codehaus.mojo</groupId>
          <artifactId>build-helper-maven-plugin</artifactId>
          <executions>
            <execution>
              <id>regex-property-sputnik</id>
              <phase>initialize</phase>
              <goals>
                <goal>regex-property</goal>
              </goals>
              <configuration>
                <name>sputnik.pullRequestId</name>
                <value>${env.GIT_BRANCH}</value>
                <regex>^origin/pr/([0-9]+)$</regex>
                <replacement>$1</replacement>
                <failIfNoMatch>true</failIfNoMatch>
              </configuration>
            </execution>
          </executions>
        </plugin>
        <plugin>
          <groupId>org.eu.ingwar.maven</groupId>
          <artifactId>sputnik-maven-plugin</artifactId>
          <version>0.0.1</version>
          <executions>
            <execution>
              <id>sputnik-default</id>
              <goals>
                <goal>stash</goal>
              </goals>
            </execution>
          </executions>
          <configuration>
            <checkstyleEnabled>true</checkstyleEnabled>

<checkstyleConfigurationFile>file:${project.build.directory}/resources-shared/shared/c
heckstyle/checkstyle_client.xml</checkstyleConfigurationFile>

            <stashHost>HOST_IP_ADDRESS</stashHost>
            <stashUsername>stash</stashUsername>
            <stashProjectKey>ECO</stashProjectKey>
            <stashRepositorySlug>workshop</stashRepositorySlug>
          </configuration>
        </plugin>
      </plugins>
    </build>
  </profile>
</profiles>

```

## Jenkins Job Configuration Pre Step - Execute Shell

```
mvn -N -B -X -U \  
-Pspunik initialize org.eu.ingwar.maven:sputnik-maven-plugin:1.1.0-SNAPSHOT:stash \  
-Dspunik.connector.projectKey=ECO \  
-Dspunik.connector.repositorySlug=workshop \  
-Dspunik.connector.host=HOST_IP_ADDRESS \  
-Dspunik.connector.username=stash \  
-Dspunik.connector.password=stash \  
-Dspunik.global.processTestFiles=false \  
-Dglobal.commentOnlyChangedLines=true \  
-Dspunik.global.maxNumberOfComments=20 \  
-Dspunik.pmd.enabled=true \  
  
-Dspunik.pmd.pmdRulesets='file:${project.build.directory}/resources-shared/shared/checkstyle/pmd_client.xml' \  
-Dspunik.findbugs.enabled=false \  
  
-Dspunik.findbugs.includeFilter='file:${project.build.directory}/resources-shared/shared/checkstyle/findbugs_client.xml' || true
```

## Stash

Host IP address 52.28.55.156

You can access Stash at 7090

## Documentation

- <https://confluence.atlassian.com/display/STASH/Stash+Documentation+Home>
- <https://confluence.atlassian.com/display/STASHKB/Troubleshooting+Installation>

## Download Page

- <https://www.atlassian.com/software/stash/download#allDownloads>

## Installation

### Pre Install

```
CREATE USER stash WITH PASSWORD 'stash';  
CREATE DATABASE stash;  
GRANT ALL PRIVILEGES ON DATABASE stash TO stash;
```

## Install

```
wget
https://www.atlassian.com/software/stash/downloads/binary/atlassian-stash-3.8.0-x64.bi
n
chmod +x atlassian-stash-3.8.0-x64.bin
./atlassian-stash-3.8.0-x64.bin

rm -fr atlassian-stash-3.8.0-x64.bin
```

## API Documentation

- <https://developer.atlassian.com/static/rest/stash/latest/stash-rest.html>

## Configuration

1. Create repository and enable Branching Model

## Set JIRA User Directory

1. Go to User Directories
2. Add directory
3. Choose directory type: 'Atlassian JIRA'
4. Set
  - a. directory name
  - b. paste jira url
  - c. application name (application name from [Jira User Server](#))
  - d. application password (application password from [Jira User Server](#))
5. Test connection
6. Save configuration
7. Synchronize directory

## Sonar

Host IP address 52.28.55.156

You can access Sonar at 9000

## Documentation

- <http://docs.codehaus.org/display/SONAR/Installing>

## Download Page

- <http://www.sonarqube.org/downloads/>

## Installation

## Pre Install

```
CREATE USER sonar WITH PASSWORD 'sonar';
CREATE DATABASE sonar;
GRANT ALL PRIVILEGES ON DATABASE stash TO sonar;
```

## Install

```
echo "deb http://downloads.sourceforge.net/project/sonar-pkg/deb binary/" >>
/etc/apt/sources.list
apt-get update
apt-get install --yes sonar
```

## Post Install

```
service sonar stop
sed -i 's(#sonar.jdbc.url=jdbc:postgresql(sonar.jdbc.url=jdbc:postgresql(g'
/opt/sonar/conf/sonar.properties
sed -i 's(sonar.jdbc.url=jdbc:h2(#sonar.jdbc.url=jdbc:h2(g'
/opt/sonar/conf/sonar.properties
sed -i 's(#sonar.jdbc.username=sonar(sonar.jdbc.username=sonar(g'
/opt/sonar/conf/sonar.properties
sed -i 's(#sonar.jdbc.password=sonar(sonar.jdbc.password=sonar(g'
/opt/sonar/conf/sonar.properties
service sonar start
```

## API Documentation

- [http://nemo.sonarqube.org/api\\_documentation](http://nemo.sonarqube.org/api_documentation)

## Demo Project

- <https://github.com/SonarSource/sonar-examples>