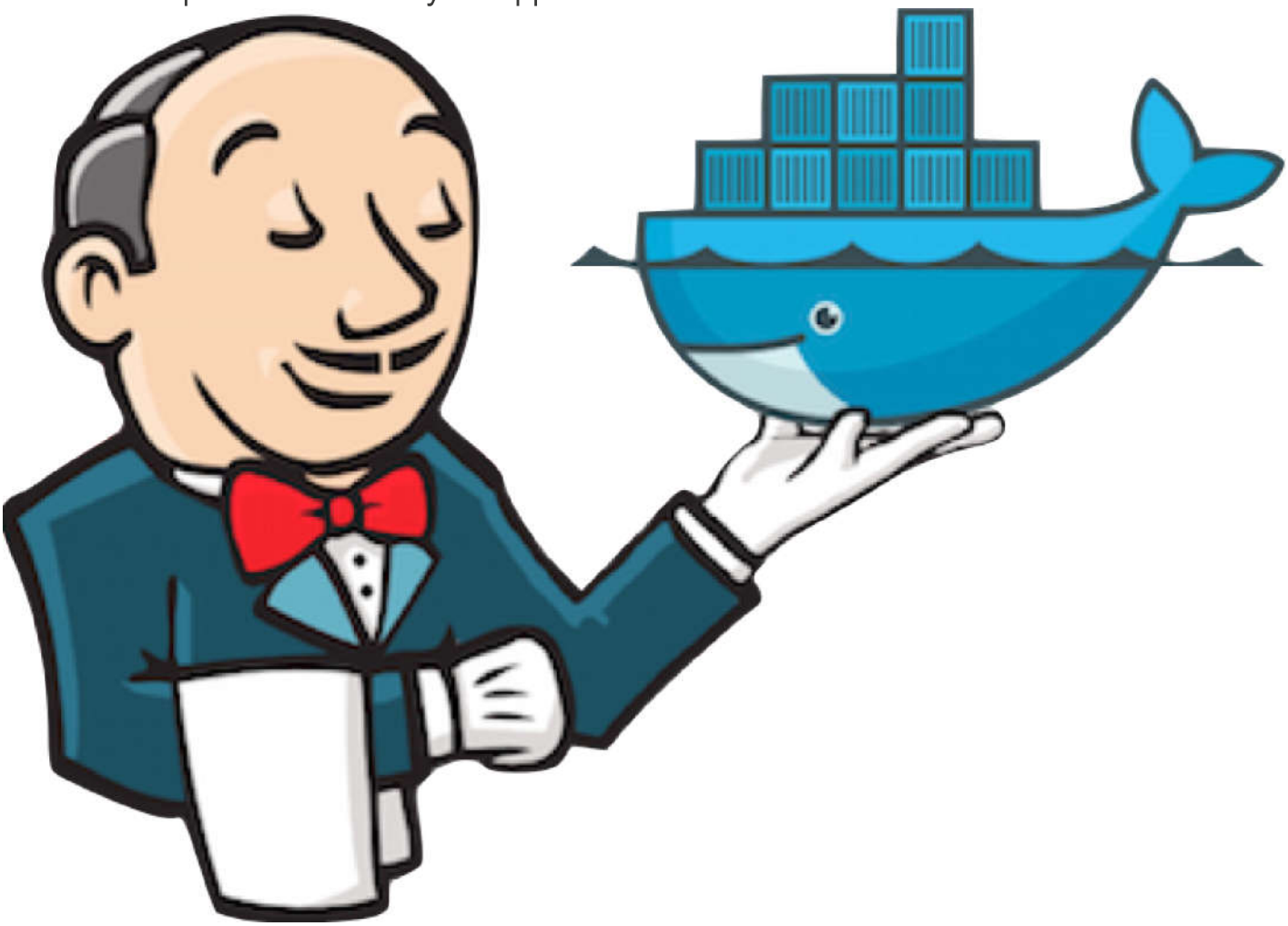
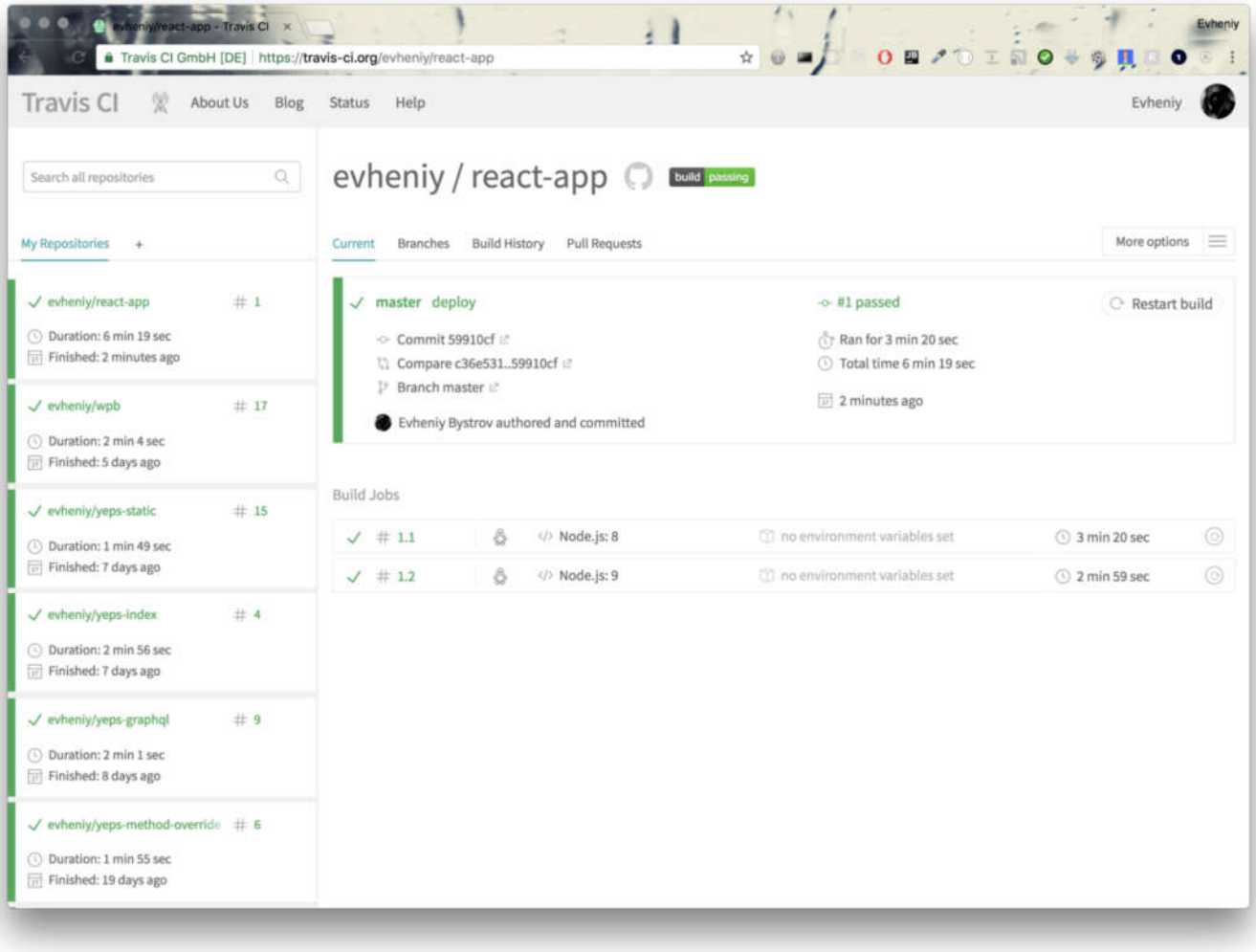
create a CD process to make your app better from scratch.



Before we start I'll put useful links.
First part of article:
React app from scratch
*It's a first part of tutorial where I'm going to show how to create react app from scratch.*medium.com
Other my article—how to start using Docker:
Making right things using Docker
*In this article I want to show how to use docker for development and testing. To show that now is time to switch from…*hackernoon.com
If your project is open source—you can use Travis CI. You just need to create .travis.yml like this:

```
language: node_js
node_js:
  - "8"
  - "9"
script:
  - npm test
```

And see result:



But I want to show you how to create CI/CD environment from scratch.
We will use **Jenkins** with **Docker.** Jenkins has official image on docker hub.
All configs you can find on github: https://github.com/evheniy/react-app/tree/master/jenkins

Jenkins

To run Jenkins using official image from docker hub we need to run next command:

```
docker run --name jenkins -p 8080:8080 jenkins
```

But we need to store our data if we need to update image or restart container. So we need to map volume to host machine:

```
docker run -p 8080:8080 -v $PWD/jenkins:/var/jenkins_
```

We will use docker not only for running Jenkins. With docker we can run docker registry and we can test and create images for our react app.

Registry

To run docker registry using docker image on docker hub run:

```
docker run -d -p 5000:5000 --restart always --name re
```

To push a new image to registry use next commands:

```
docker pull ubuntu
docker tag ubuntu localhost:5000/ubuntu
docker push localhost:5000/ubuntu
```

Docker compose

To run registry with Jenkins I'll use docker-compose.
But before I'll create Dockerfile for Jenkins. We need it for running docker inside docker.

```
touch Dockerfile
```

And put:

```
FROM jenkins/jenkins:lts

USER root
```

Here I use latest image of Jenkins and run it as root.
Now we are ready to use docker-compose.
Let's create **docker-compose** file for running Jenkins and docker repository in one command:

```
touch docker-compose.yml
```

And one more. As we need to run docker inside docker we need to add more volumes. But first run command:

```
which docker
```

And put next code:

```
version: '3'

services:

  jenkins:
    build: .
    container_name: jenkins
    privileged: true
    restart: always
    ports:
      - 8080:8080
    volumes:
      - ./jenkins_home:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
      - /usr/local/bin/docker:/usr/bin/docker

  registry:
    image: registry
    container_name: registry
    restart: always
    ports:
      - 5000:5000
```

To run it use

```
docker-compose up -d
```

And to stop:

```
docker-compose stop
```

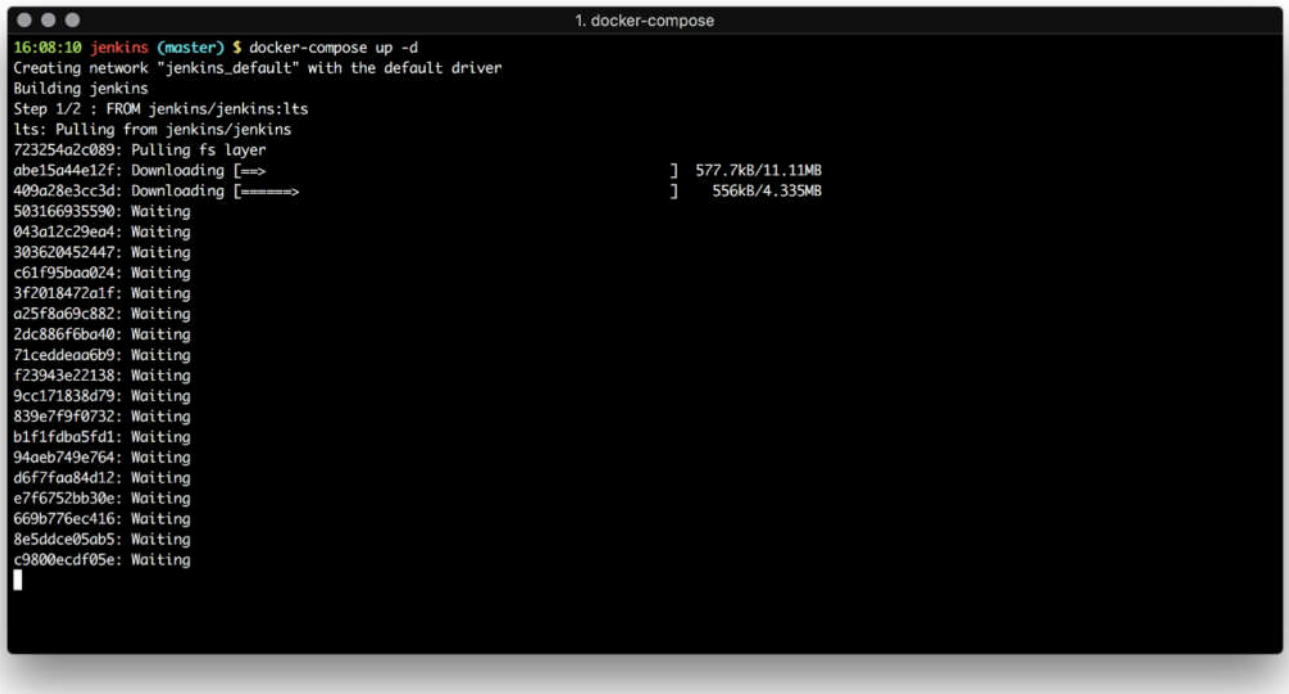
To bring everything down (with volumes):

```
docker-compose down --volumes
```

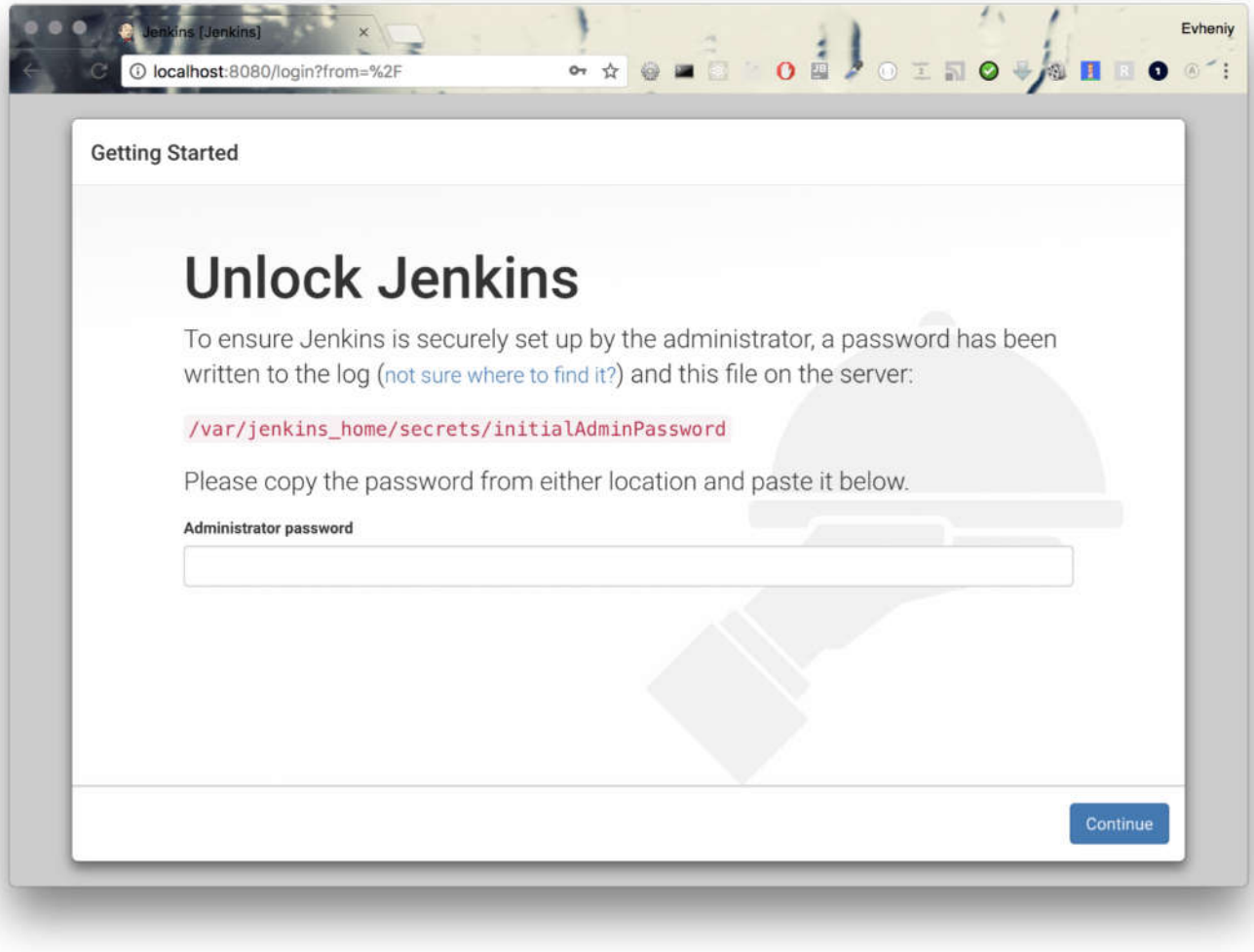
Or if you want to remove docker images:

```
docker-compose down --rmi all
```

So let's run it and configure Jenkins to use pipeline.

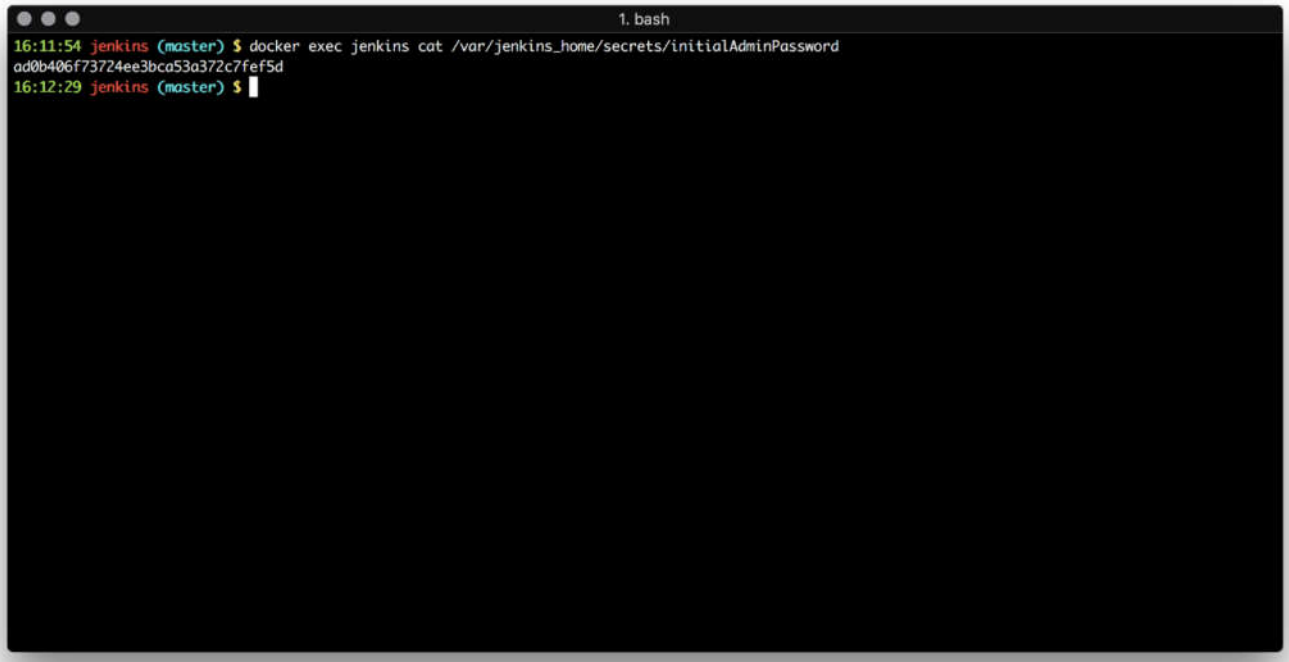


Open http://localhost:8000/:

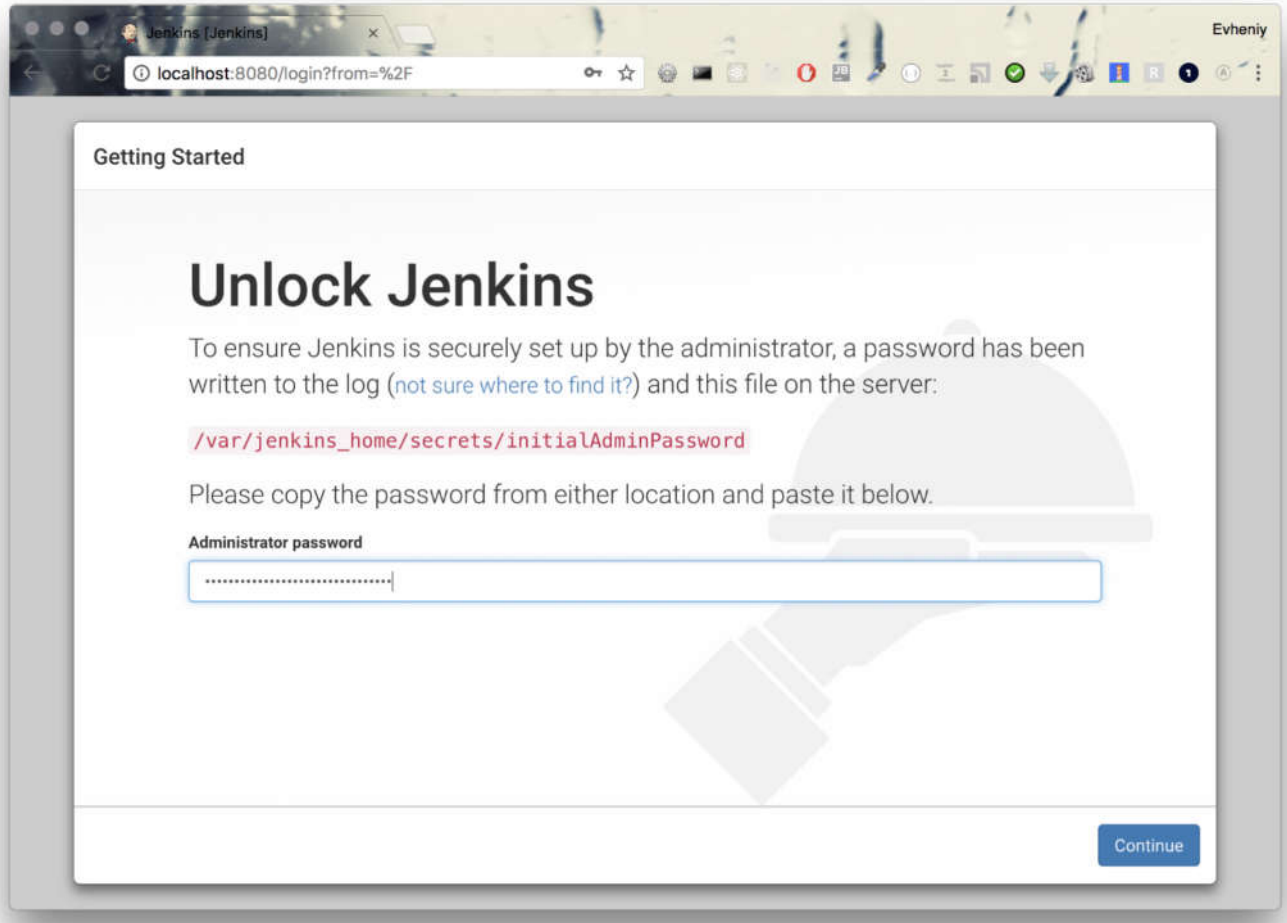


To see the password just run (we need it only once):

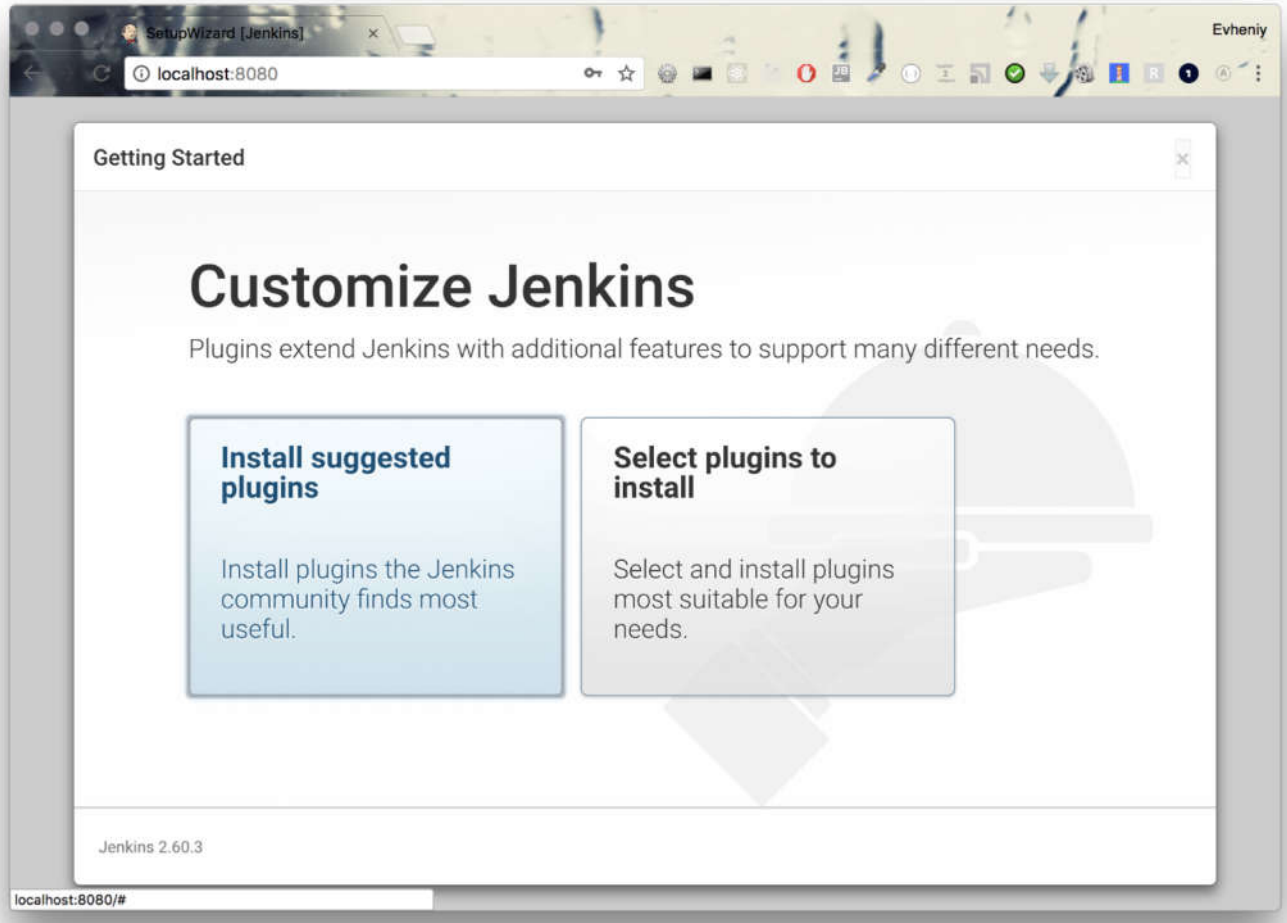
```
docker exec jenkins cat /var/jenkins_home/secrets/ini
```



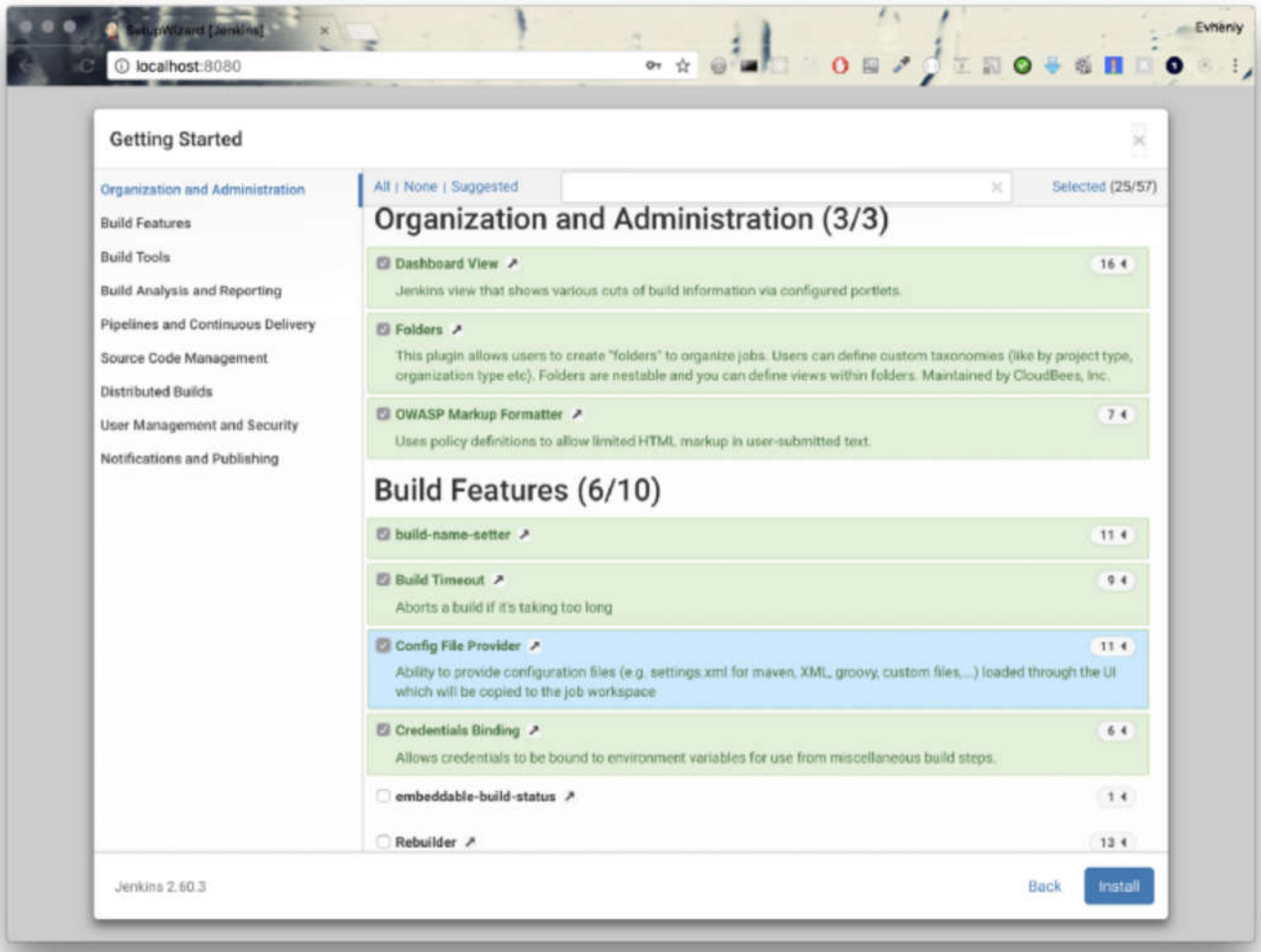
And enter it on page:

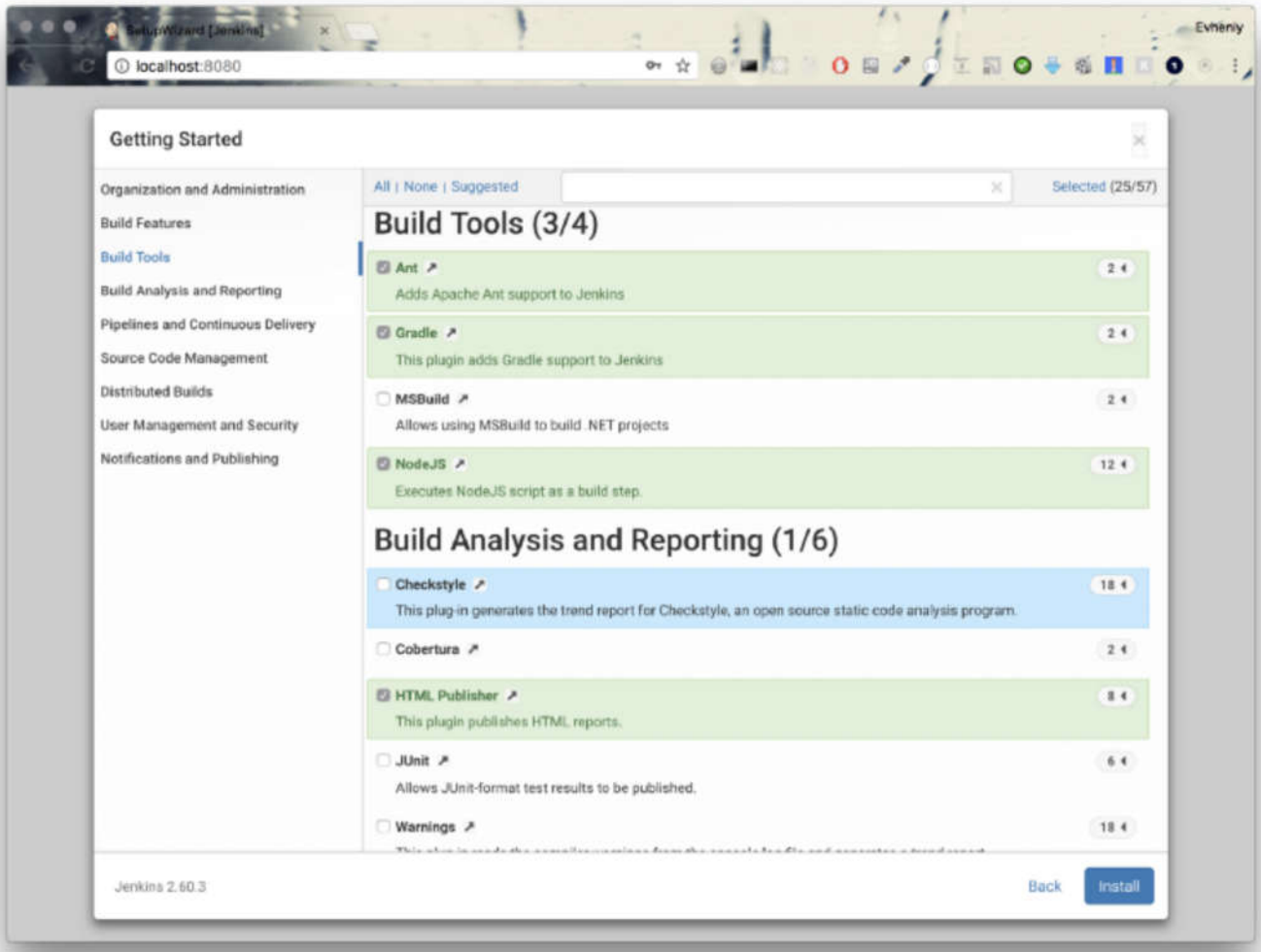


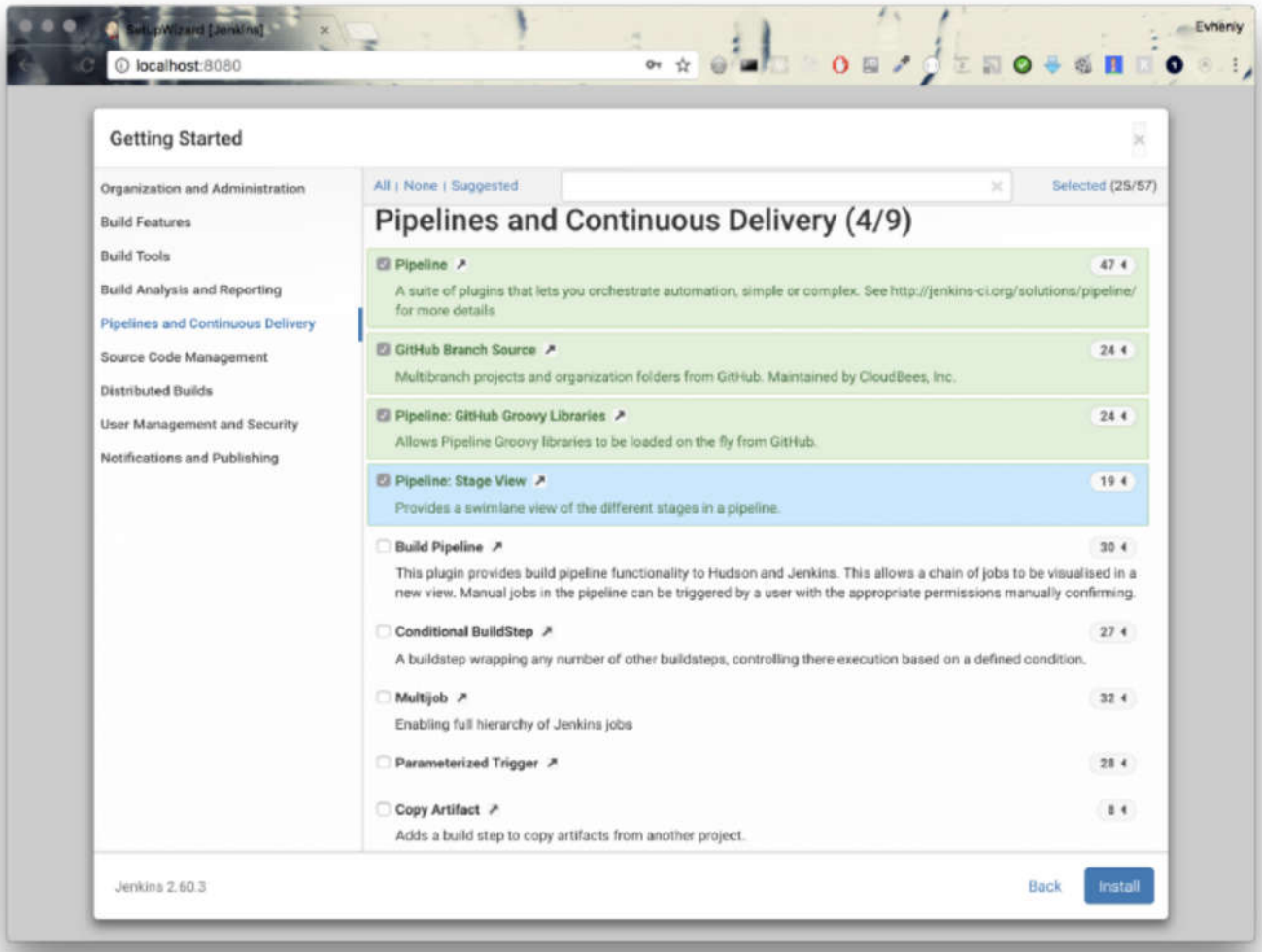
And as it’s our first run we see configuration page:

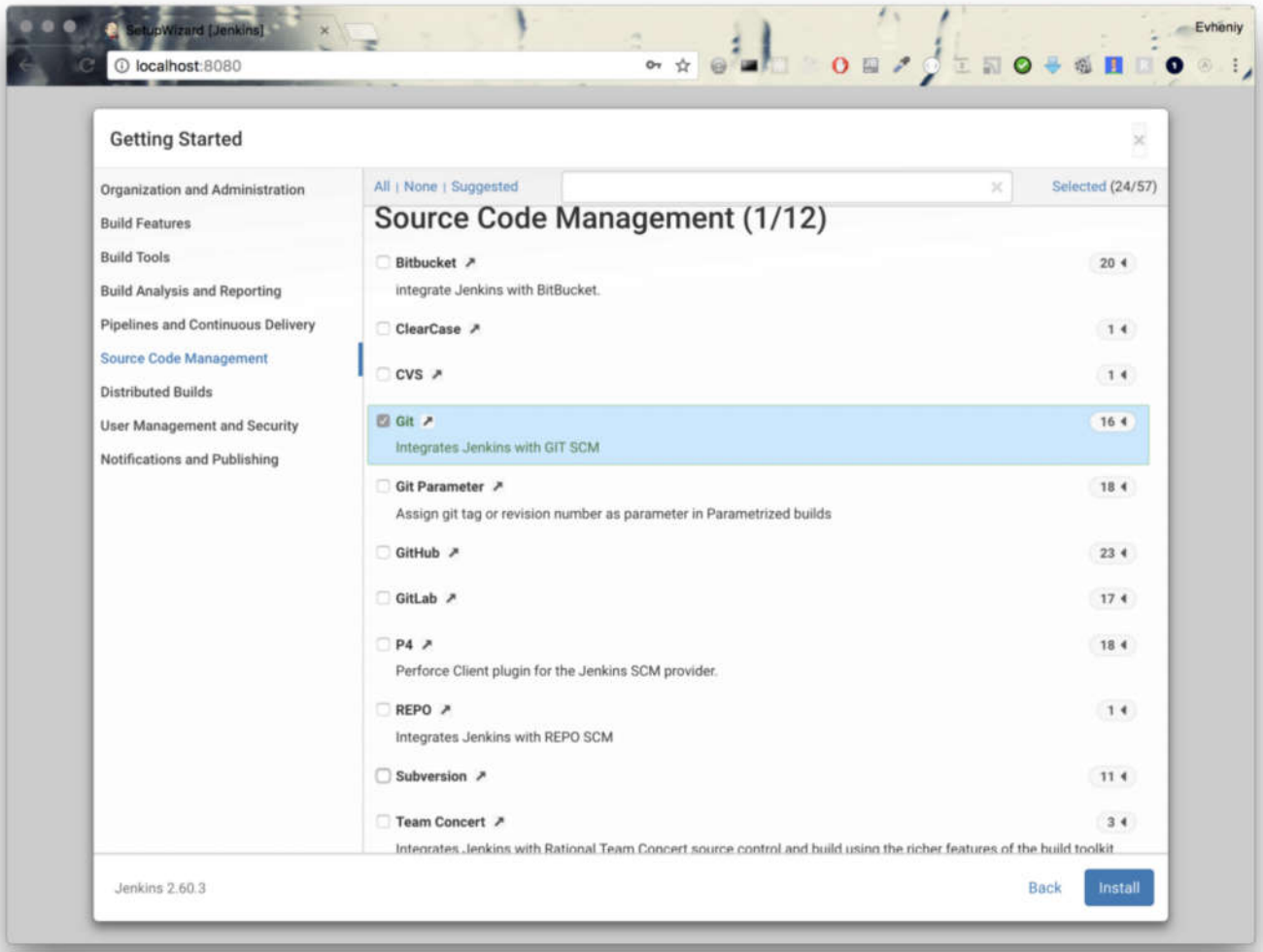


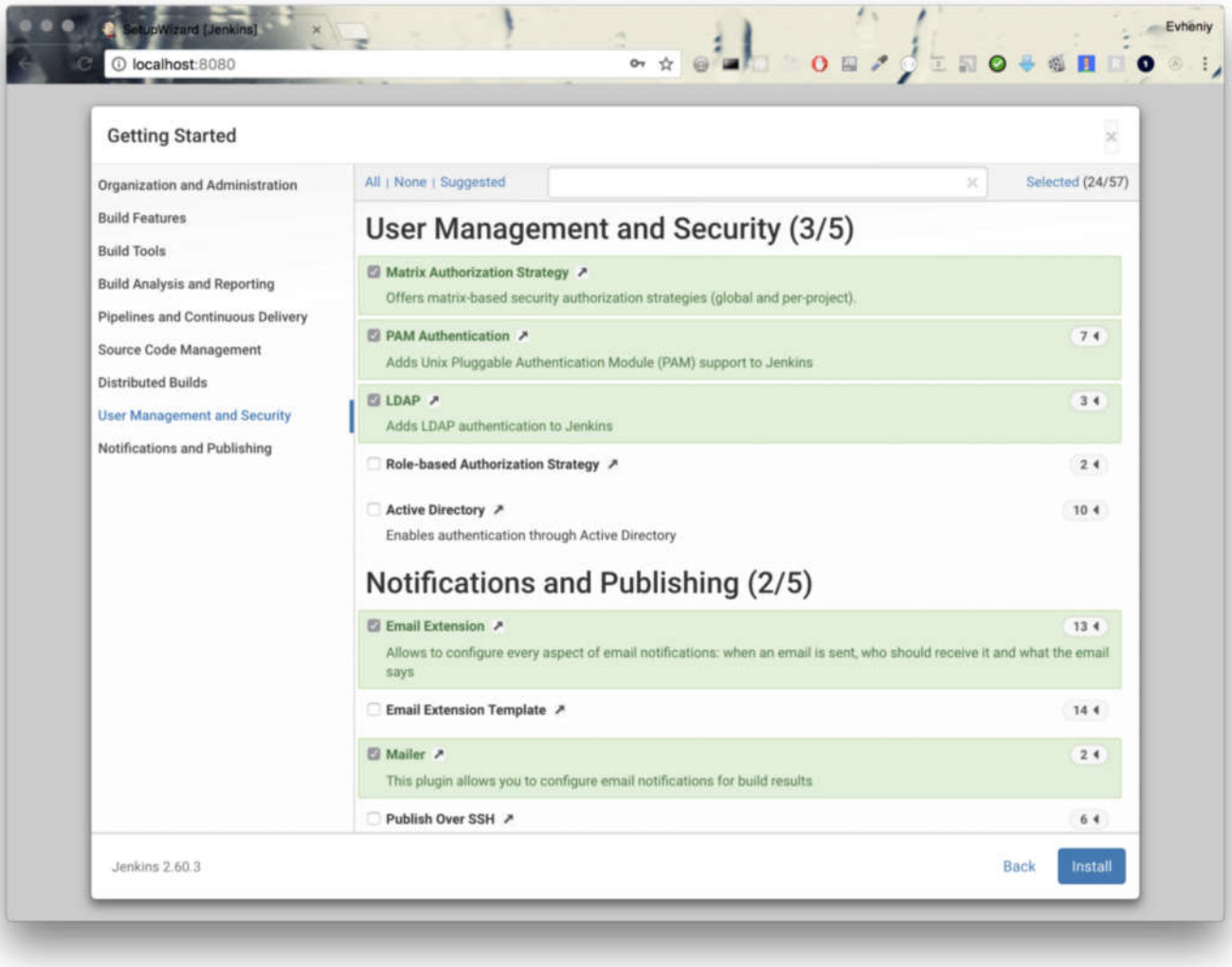
I choose the second point and check next plugins:


Setup Wizard [Jenkins]
Evheniy
localhost:8080
Getting Started
Organization and Administration
Build Features
Build Tools
Build Analysis and Reporting
Pipelines and Continuous Delivery
Source Code Management
Distributed Builds
User Management and Security
Notifications and Publishing
All | None | Suggested
Selected (25/57)
Organization and Administration (3/3)
Dashboard View
Jenkins view that shows various cuts of build information via configured portlets.
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
Uses policy definitions to allow limited HTML markup in user-submitted text.
Build Features (6/10)
build-name-setter
Build Timeout
Aborts a build if it's taking too long
Config File Provider
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
Allows credentials to be bound to environment variables for use from miscellaneous build steps.
embeddable-build-status
Rebuilder
Jenkins 2.60.3
Back
Install

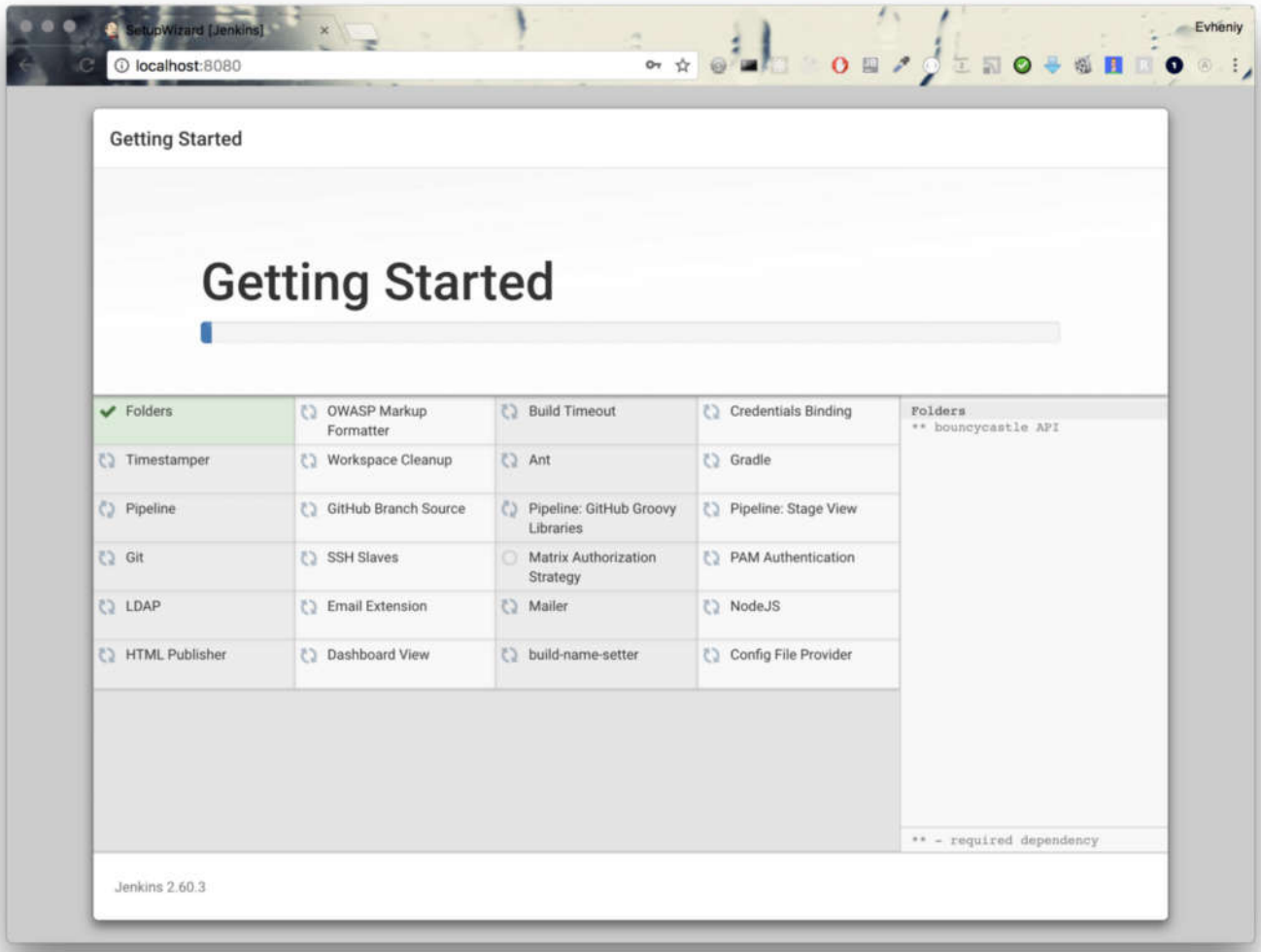

Setup Wizard [Jenkins]
localhost:8080
Evheniy
Getting Started
Organization and Administration
Build Features
Build Tools
Build Analysis and Reporting
Pipelines and Continuous Delivery
Source Code Management
Distributed Builds
User Management and Security
Notifications and Publishing
All | None | Suggested
Selected (25/57)
Build Tools (3/4)
Ant
Adds Apache Ant support to Jenkins
Gradle
This plugin adds Gradle support to Jenkins
MSBuild
Allows using MSBuild to build .NET projects
NodeJS
Executes NodeJS script as a build step.
Build Analysis and Reporting (1/6)
Checkstyle
This plug-in generates the trend report for Checkstyle, an open source static code analysis program.
Cobertura
HTML Publisher
This plugin publishes HTML reports.
JUnit
Allows JUnit-format test results to be published.
Warnings
Jenkins 2.60.3
Back
Install


localhost:8080
Getting Started
Organization and Administration
Build Features
Build Tools
Build Analysis and Reporting
Pipelines and Continuous Delivery
Source Code Management
Distributed Builds
User Management and Security
Notifications and Publishing
All | None | Suggested
Selected (25/57)
Pipelines and Continuous Delivery (4/9)
Pipeline
A suite of plugins that lets you orchestrate automation, simple or complex. See http://jenkins-ci.org/solutions/pipeline/ for more details
GitHub Branch Source
Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.
Pipeline: GitHub Groovy Libraries
Allows Pipeline Groovy libraries to be loaded on the fly from GitHub.
Pipeline: Stage View
Provides a swimlane view of the different stages in a pipeline.
Build Pipeline
This plugin provides build pipeline functionality to Hudson and Jenkins. This allows a chain of jobs to be visualised in a new view. Manual jobs in the pipeline can be triggered by a user with the appropriate permissions manually confirming.
Conditional BuildStep
A buildstep wrapping any number of other buildsteps, controlling there execution based on a defined condition.
Multijob
Enabling full hierarchy of Jenkins jobs
Parameterized Trigger
Copy Artifact
Adds a build step to copy artifacts from another project.
Jenkins 2.60.3
Back
Install

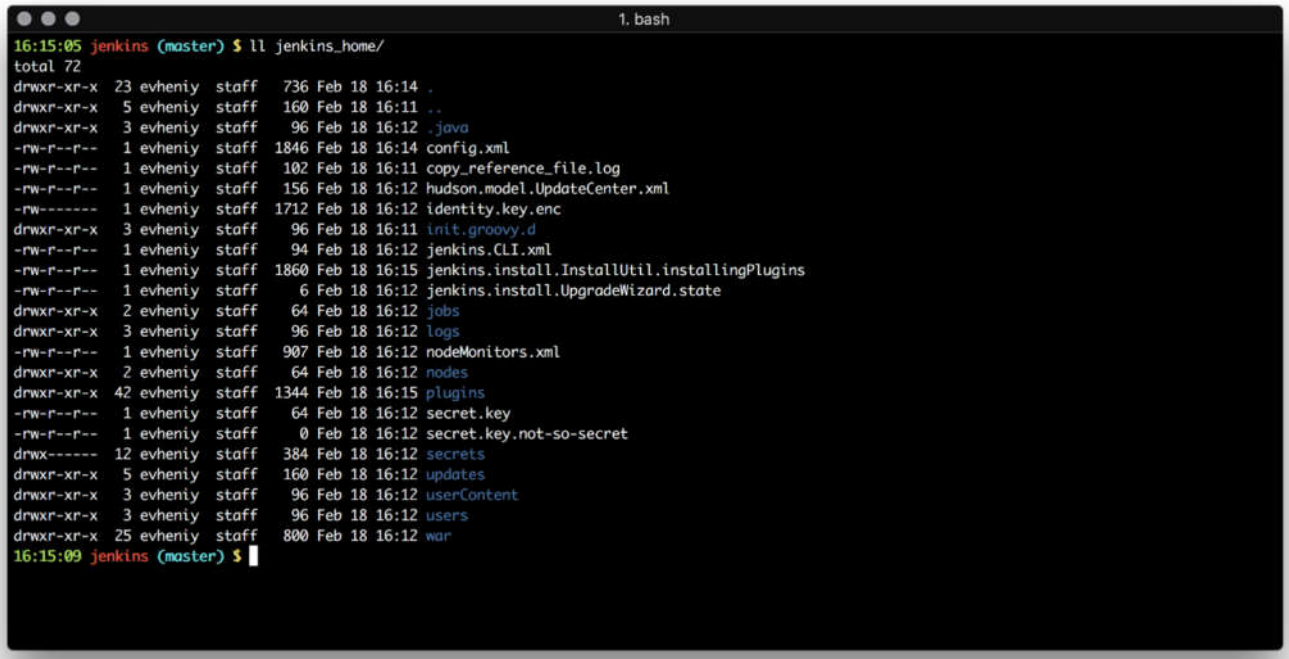

SetupWizard [Jenkins]
Evheniy
localhost:8080
Getting Started
Organization and Administration
Build Features
Build Tools
Build Analysis and Reporting
Pipelines and Continuous Delivery
Source Code Management
Distributed Builds
User Management and Security
Notifications and Publishing
All | None | Suggested
Selected (24/57)
Source Code Management (1/12)
Bitbucket
20
integrate Jenkins with BitBucket.
ClearCase
1
CVS
1
Git
16
Integrates Jenkins with GIT SCM
Git Parameter
18
Assign git tag or revision number as parameter in Parametrized builds
GitHub
23
GitLab
17
P4
18
Perforce Client plugin for the Jenkins SCM provider.
REPO
1
Integrates Jenkins with REPO SCM
Subversion
11
Team Concert
3
Integrates Jenkins with Rational Team Concert source control and build using the richer features of the build toolkit
Jenkins 2.60.3
Back
Install



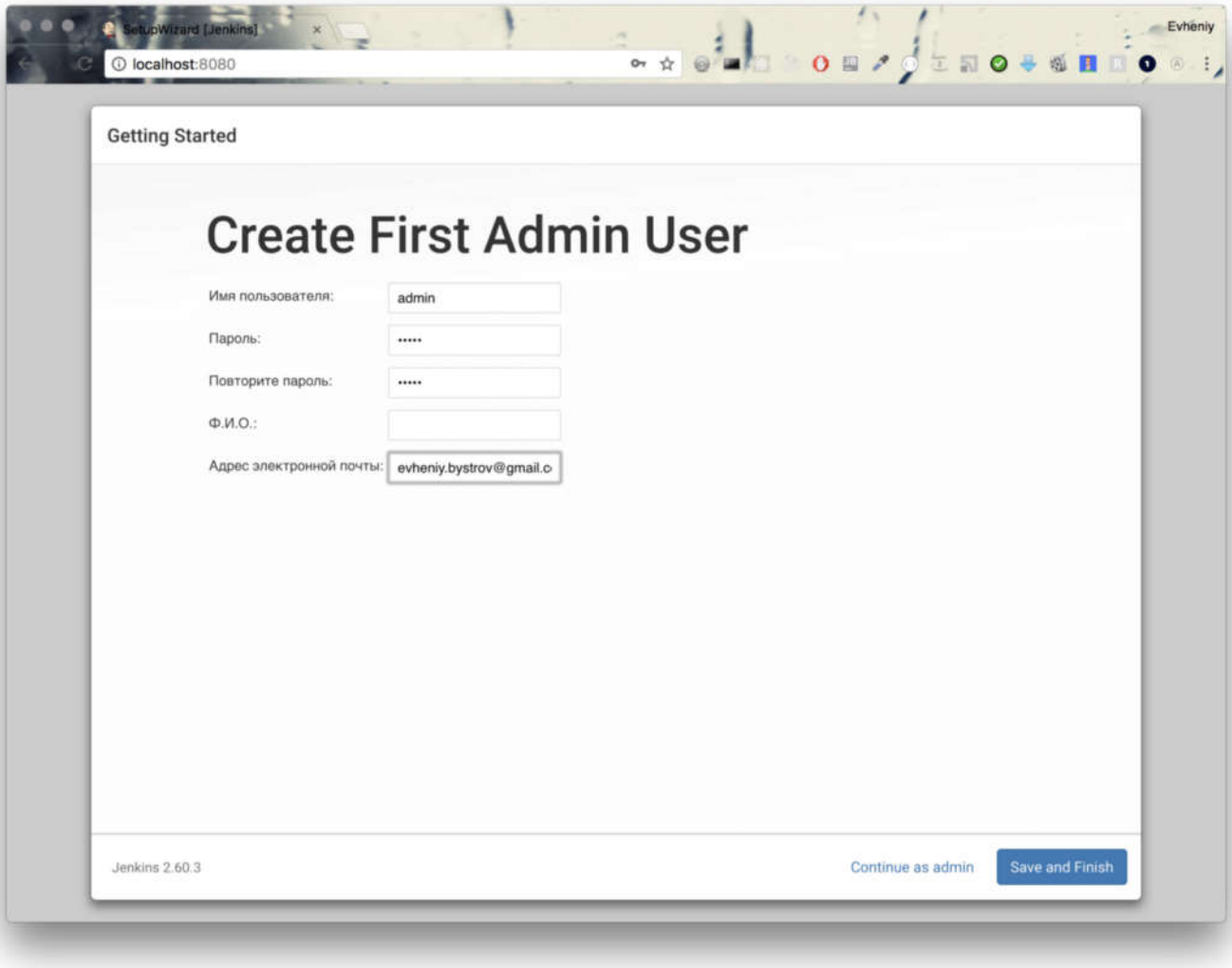
Click **install** and wait:



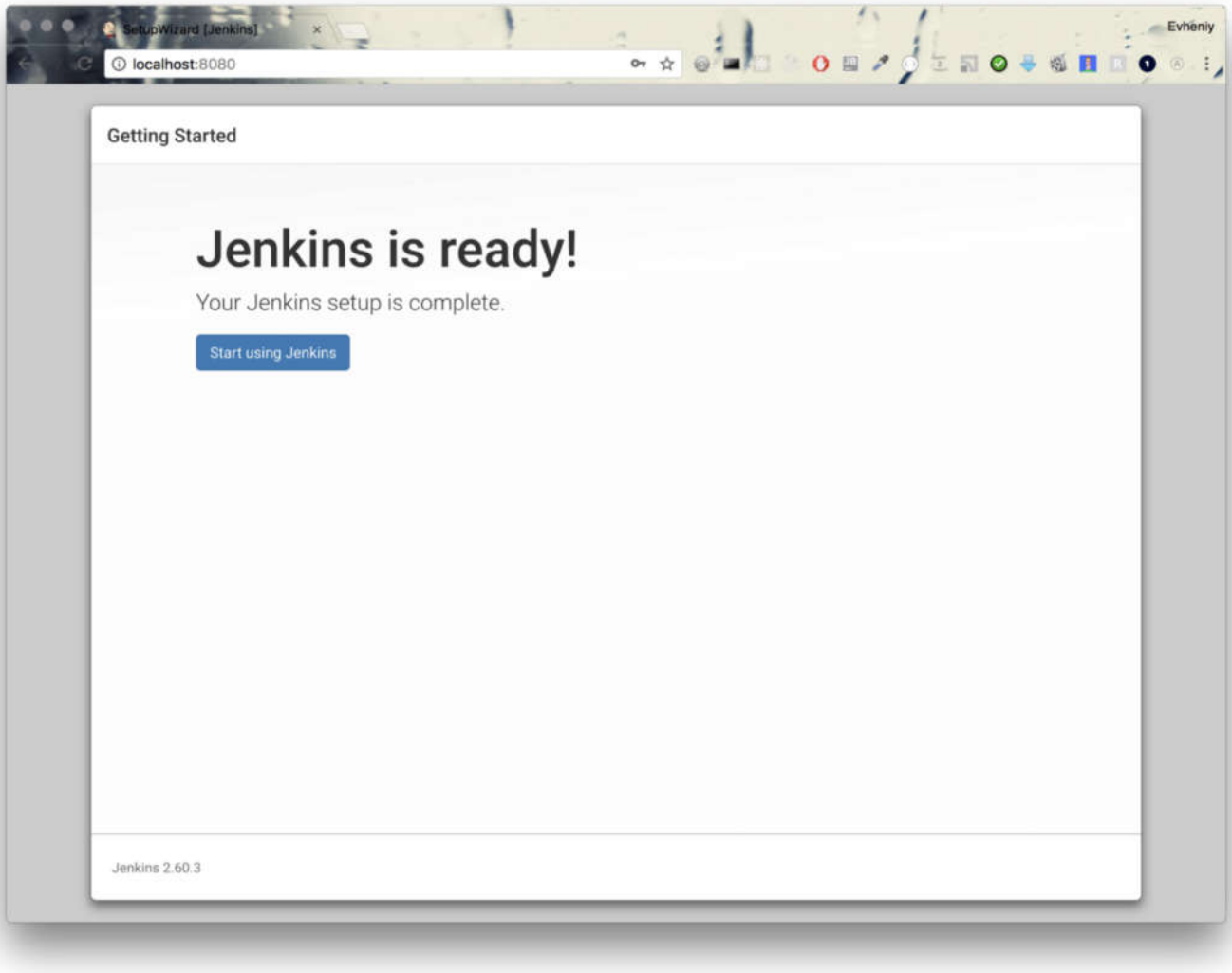
If you check directory we have a lot of Jenkins files:



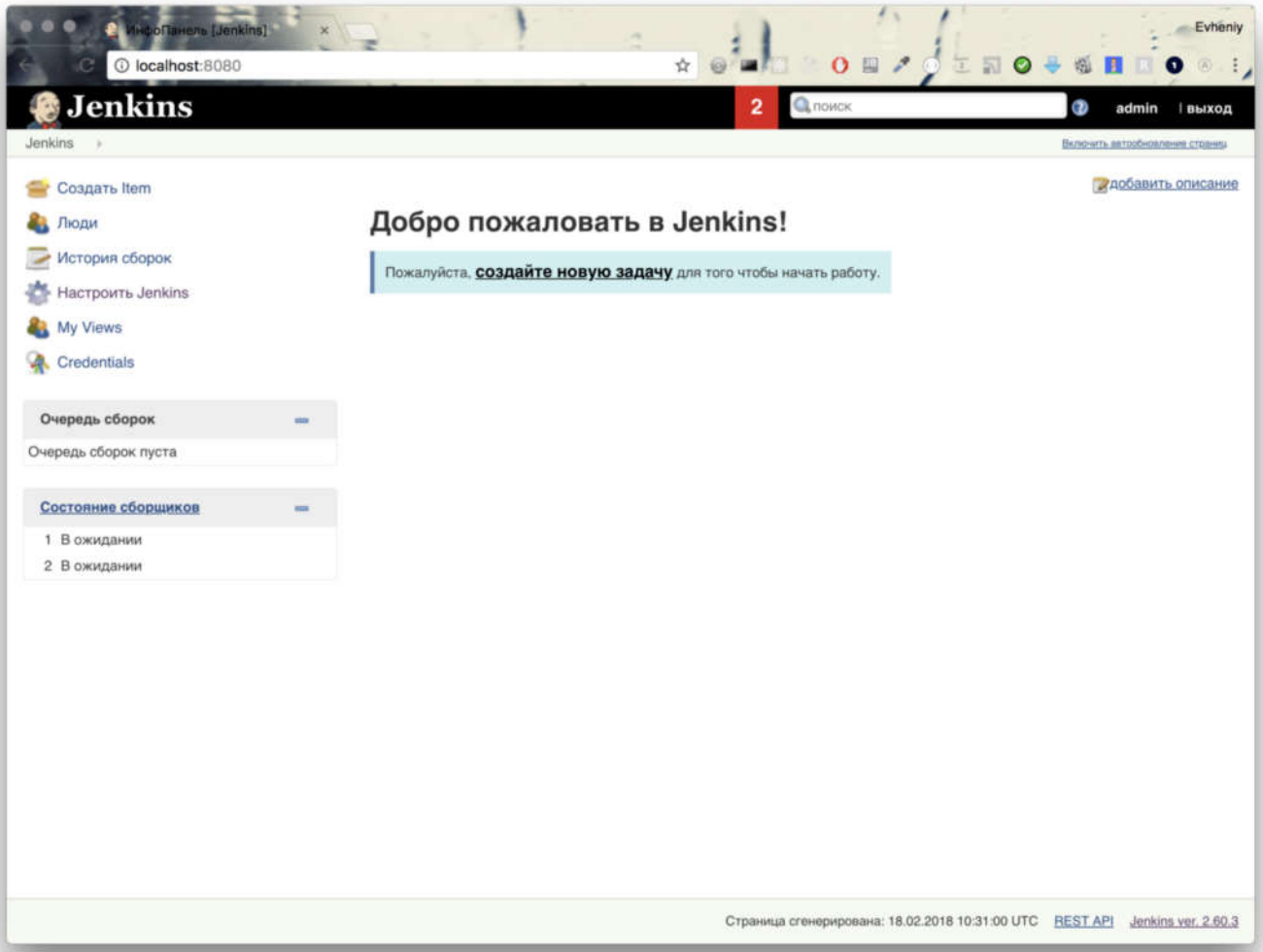
After you need to create a new user:



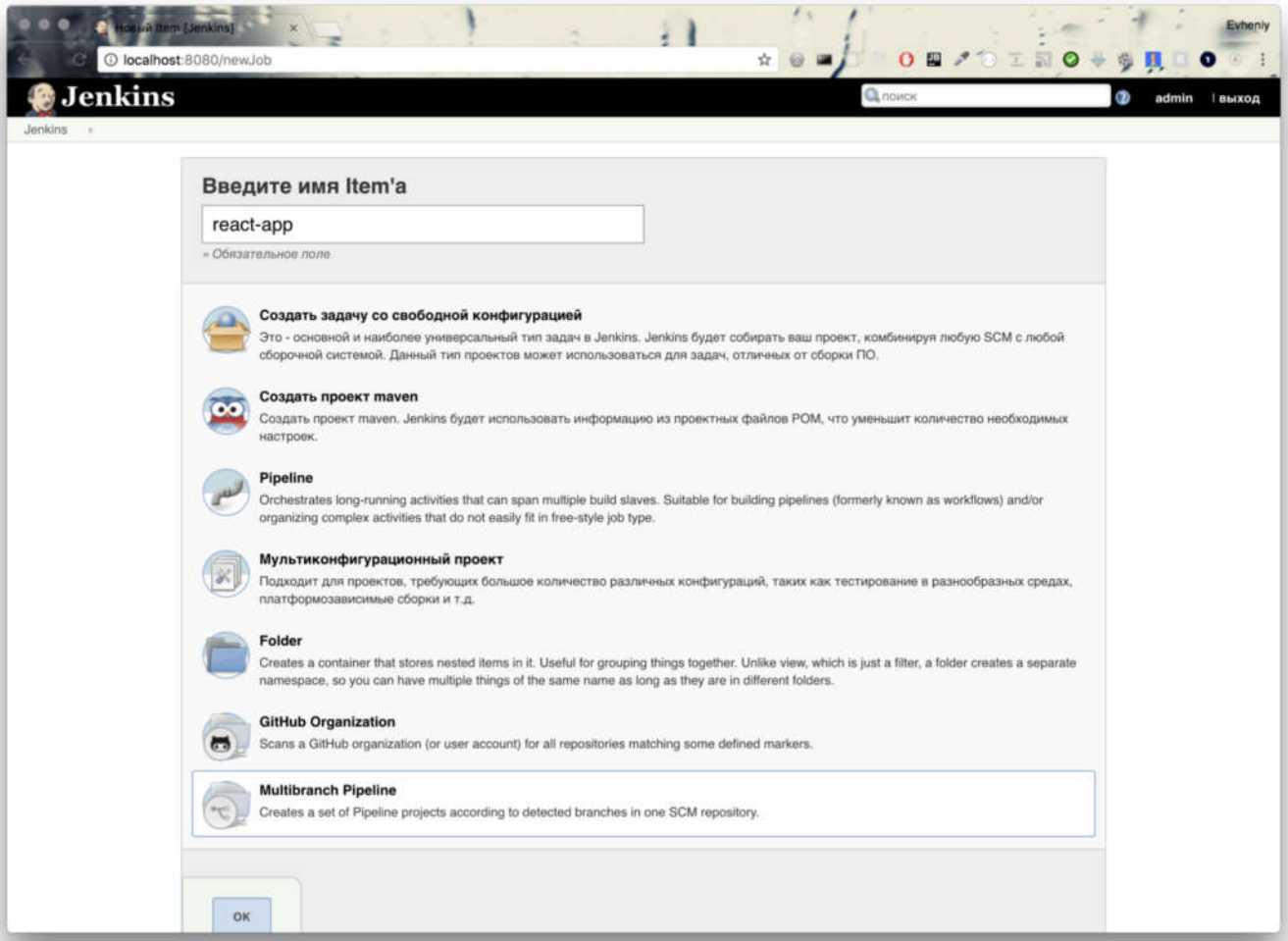
And that’s all. Jenkins is ready.



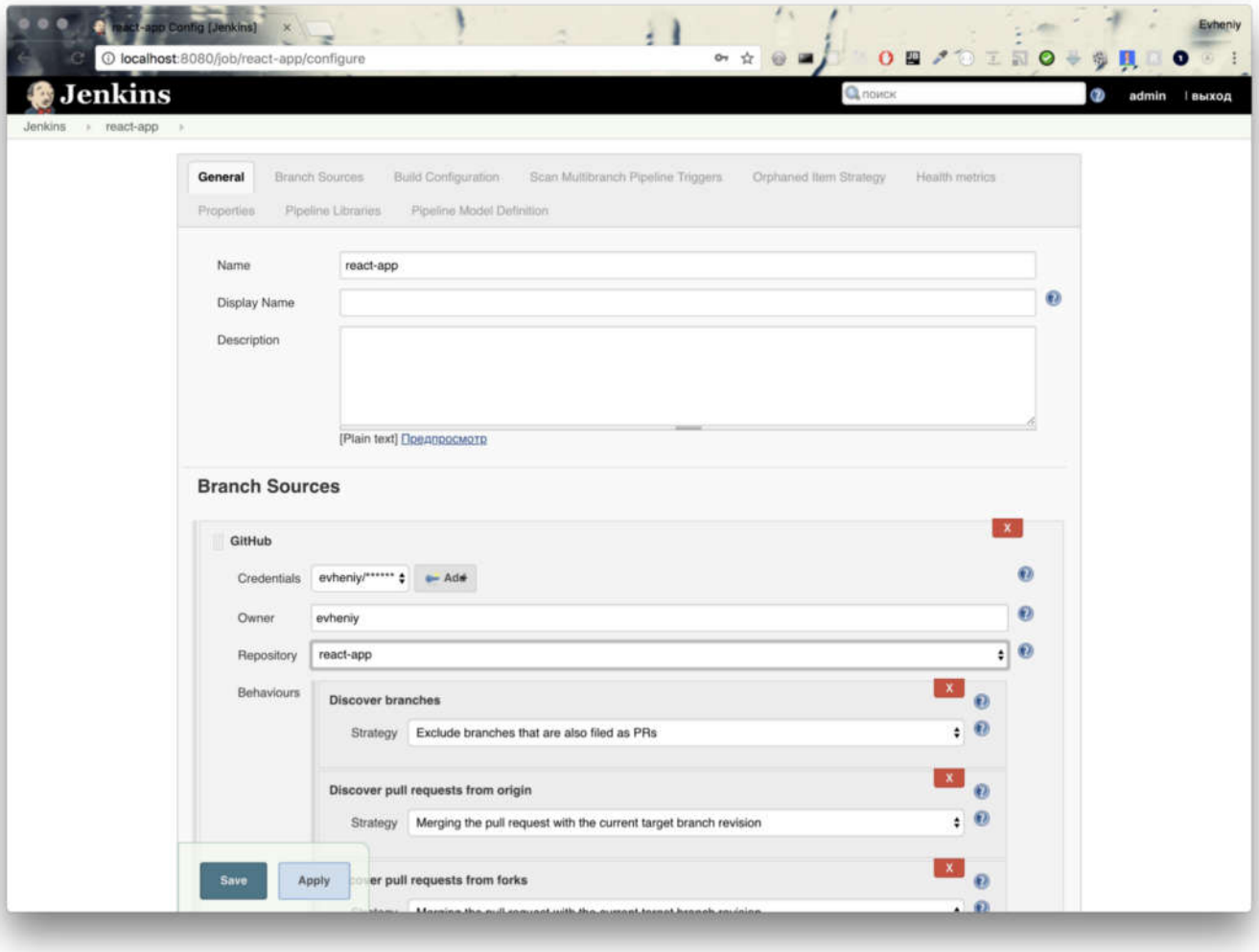
Next we need to create a new build. You can make it from start page:



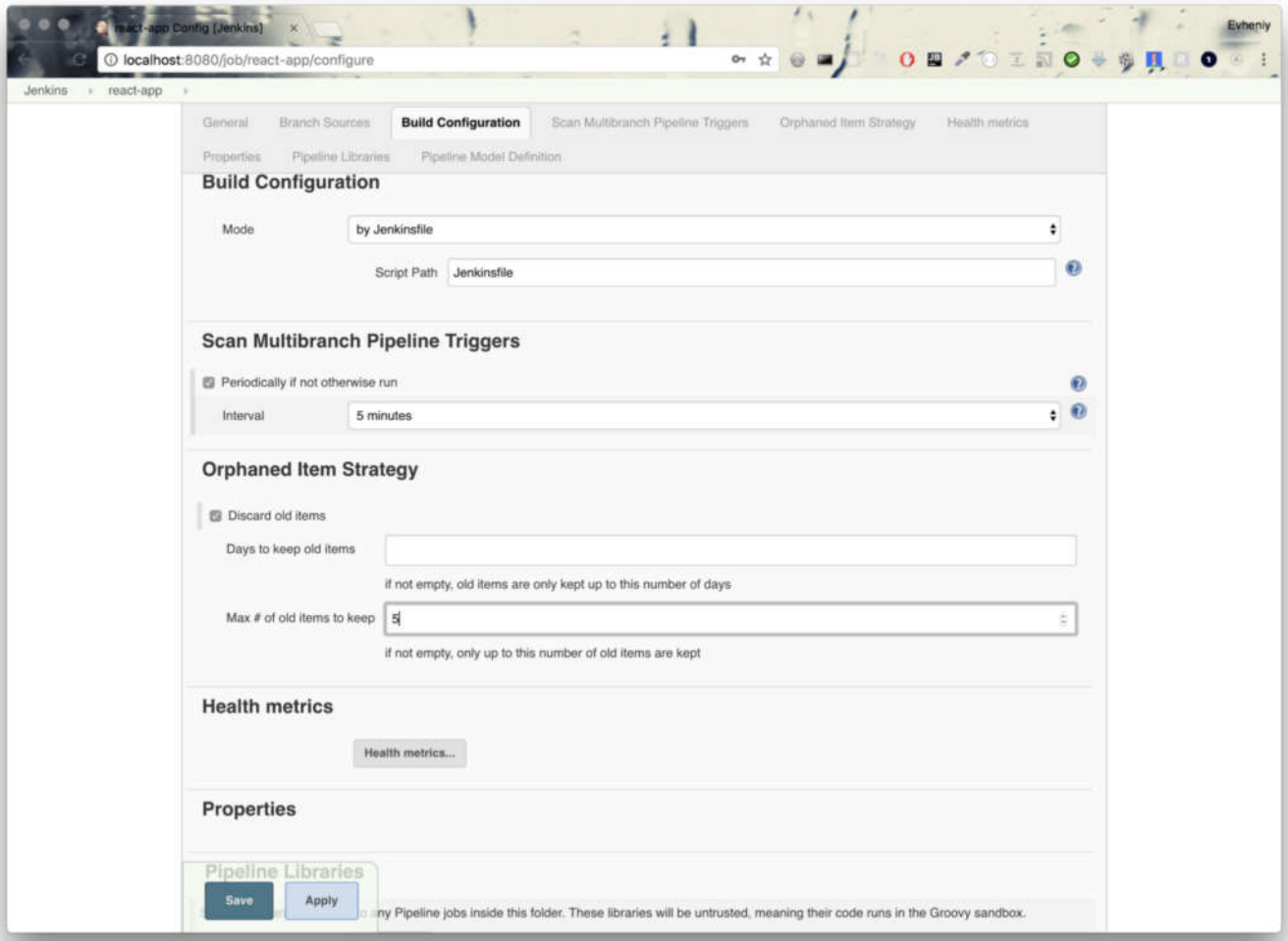
On first step we need to enter name and choose type of our build configurations:



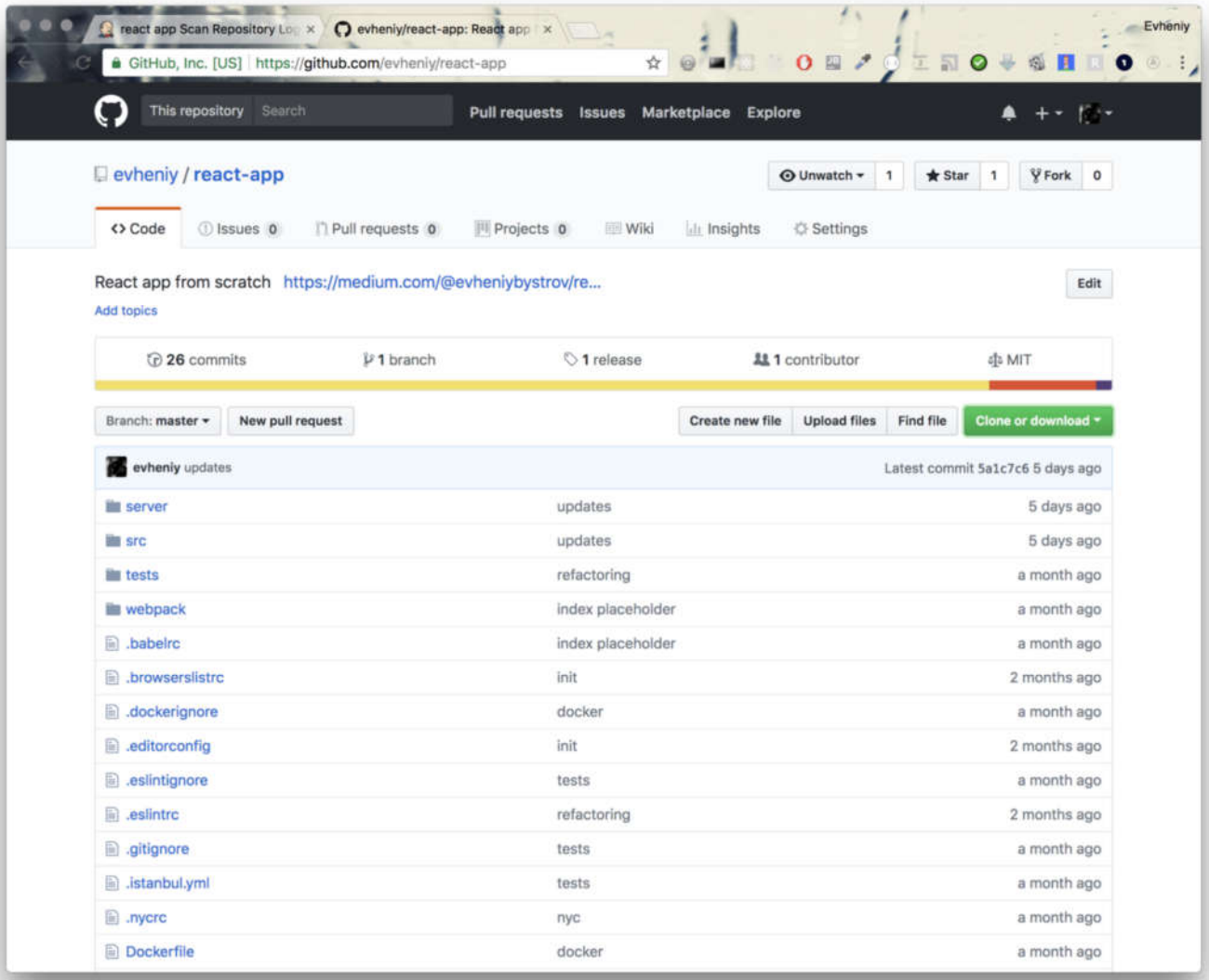
I use multibranch pipeline configuration.
Next we need to configure our build (name, access to github)



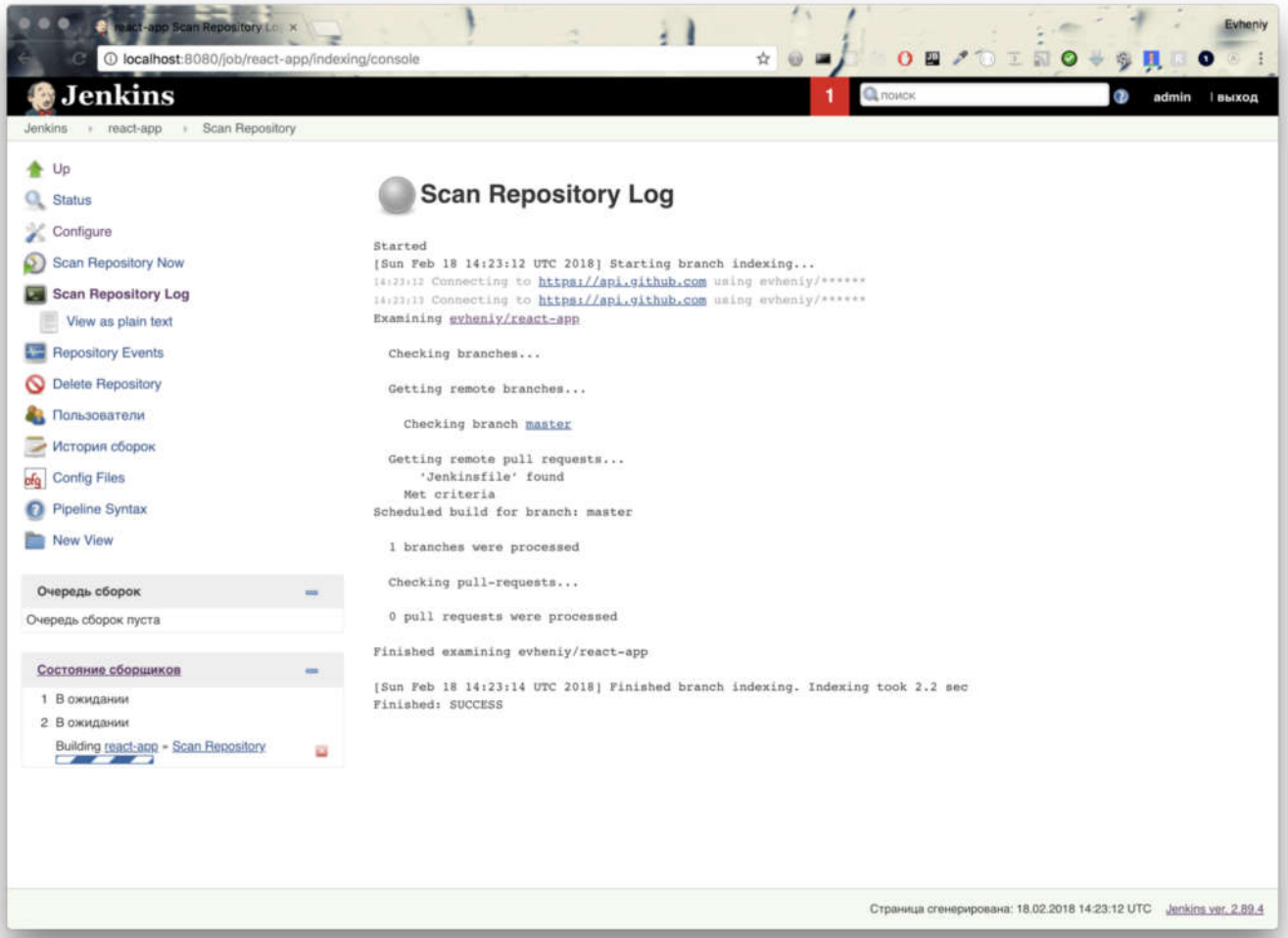
And some other configs like cleaning and scanning time.



As I made it for react-app I use the same github repo (https://github.com/evheniy/react-app)



And after Jenkins scans it



Configs

Next step is creating Jenkinsfile, Dockerfile, Dockerfile.test and save it on github:

```
node {
  try {
    stage('Checkout') {
      checkout scm
    }
    stage('Environment') {
      sh 'git --version'
      echo "Branch: ${env.BRANCH_NAME}"
      sh 'docker -v'
      sh 'printenv'
    }
    stage('Build Docker test'){
     sh 'docker build -t react-test -f Dockerfile.tes
```

```
    }
    stage('Docker test'){
      sh 'docker run --rm react-test'
    }
    stage('Clean Docker test'){
      sh 'docker rmi react-test'
    }
    stage('Deploy'){
      if(env.BRANCH_NAME == 'master'){
        sh 'docker build -t react-app --no-cache .'
        sh 'docker tag react-app localhost:5000/react
        sh 'docker push localhost:5000/react-app'
        sh 'docker rmi -f react-app localhost:5000/re
      }
    }
  }
  catch (err) {
    throw err
  }
}
```

Dockerfile

```
# Extending image
FROM node:carbon

RUN apt-get update
RUN apt-get upgrade -y
RUN apt-get -y install autoconf automake libtool nasm

# Create app directory
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

# Versions
```

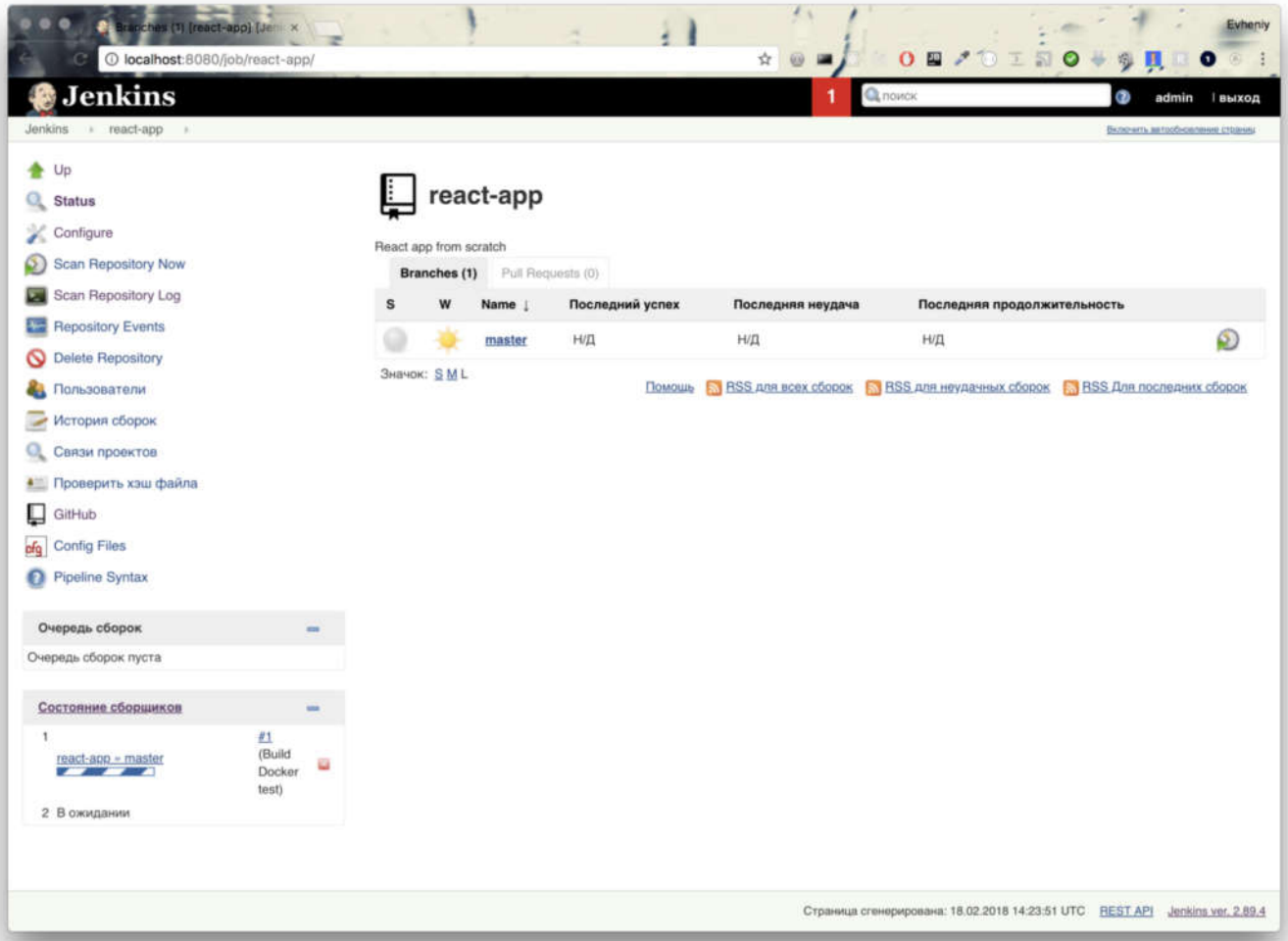
```
RUN npm -v
RUN node -v

# Install app dependencies
COPY package.json /usr/src/app/
COPY package-lock.json /usr/src/app/

RUN npm install

# Bundle app source
COPY . /usr/src/app

# Port to listener
EXPOSE 3000

# Environment variables
ENV NODE_ENV production
ENV PORT 3000
ENV PUBLIC_PATH "/"

RUN npm run start:build

# Main command
CMD [ "npm", "run", "start:server" ]
```

Dockerfile.test

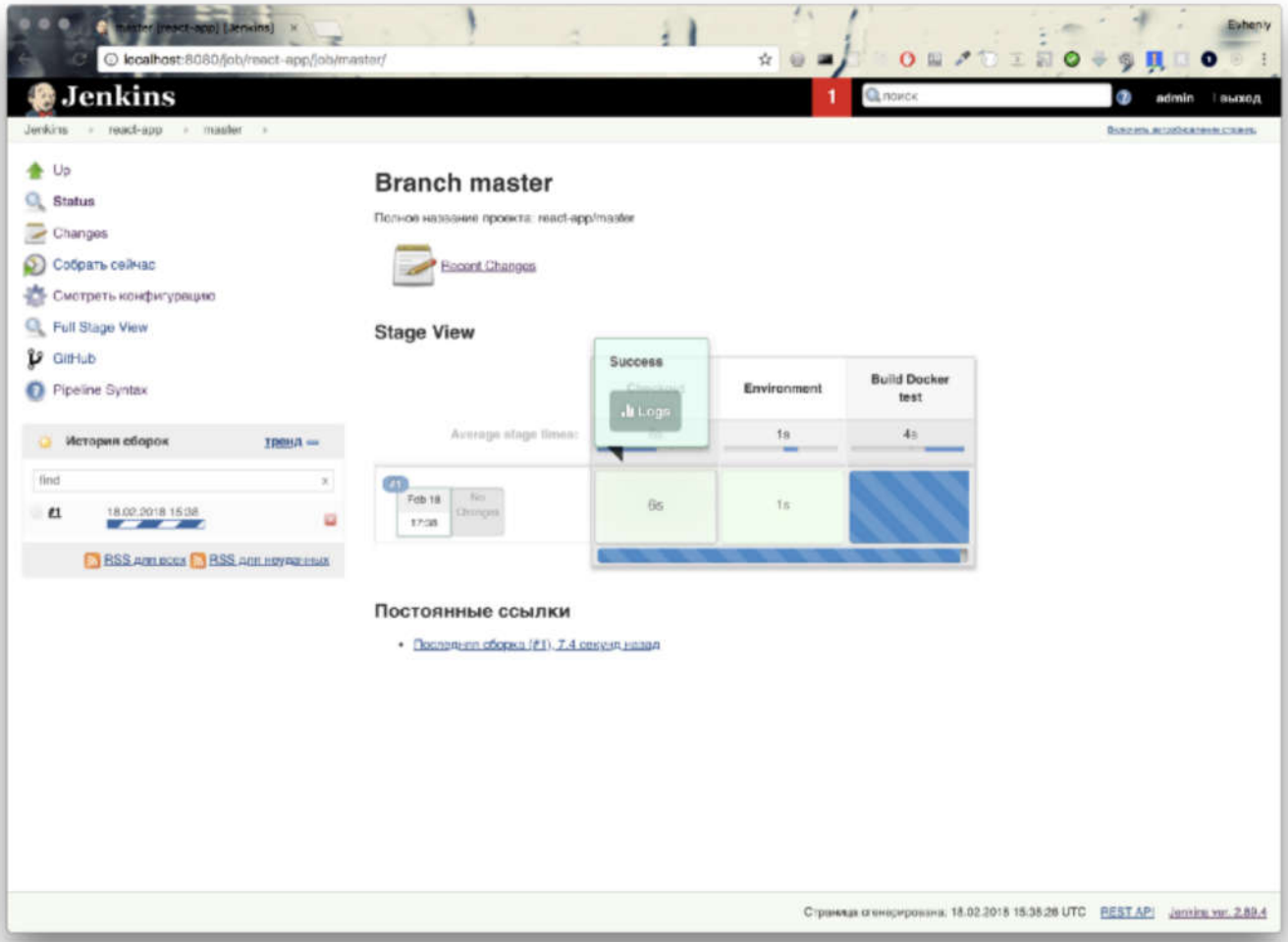
```
# Extending image
FROM node:carbon

RUN apt-get update
RUN apt-get upgrade -y
RUN apt-get -y install autoconf automake libtool nasm

# Create app directory
```

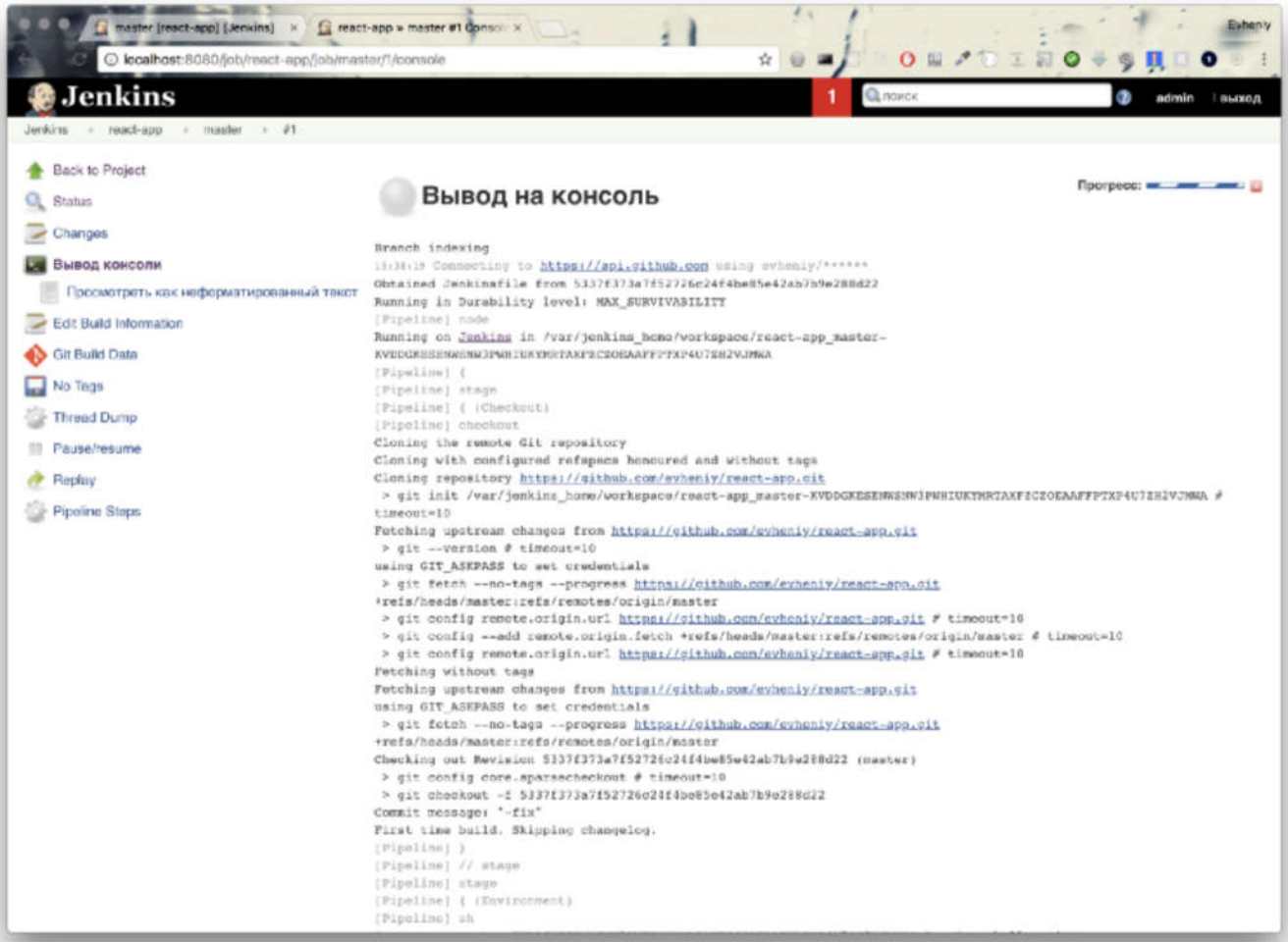
```
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

# Versions
RUN npm -v
RUN node -v

# Install app dependencies
COPY package.json /usr/src/app/
COPY package-lock.json /usr/src/app/

RUN npm install

# Bundle app source
COPY . /usr/src/app

# Environment variables
ENV NODE_ENV test

# Main command
CMD [ "npm", "test" ]
```

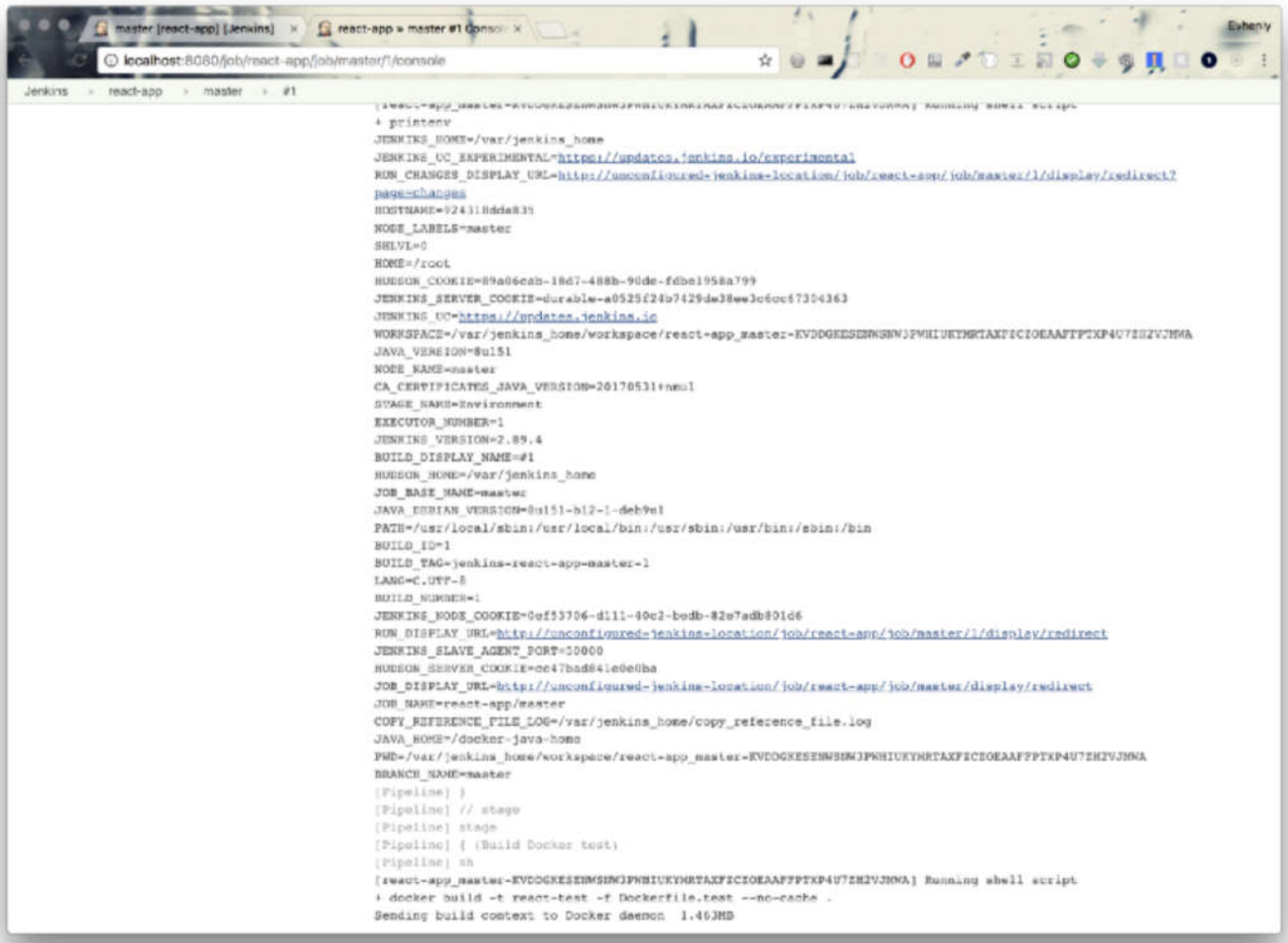
And after rescanning Jenkins checks it:


Jenkins
localhost:8080/job/react-app/
admin
выход
Jenkins
react-app
Up
Status
Configure
Scan Repository Now
Scan Repository Log
Repository Events
Delete Repository
Пользователи
История сборок
Связи проектов
Проверить хэш файла
GitHub
Config Files
Pipeline Syntax
react-app
React app from scratch
Branches (1)
Pull Requests (0)
S
W
Name
Последний успех
Последняя неудача
Последняя продолжительность
master
Н/Д
Очередь сборок
Очередь сборок пуста
Состояние сборщиков
react-app » master
#1
(Build Docker test)
2 В ожидании
Помощь
RSS для всех сборок
RSS для неудачных сборок
RSS Для последних сборок
Страница сгенерирована: 18.02.2018 14:23:51 UTC
REST API
Jenkins ver. 2.89.4

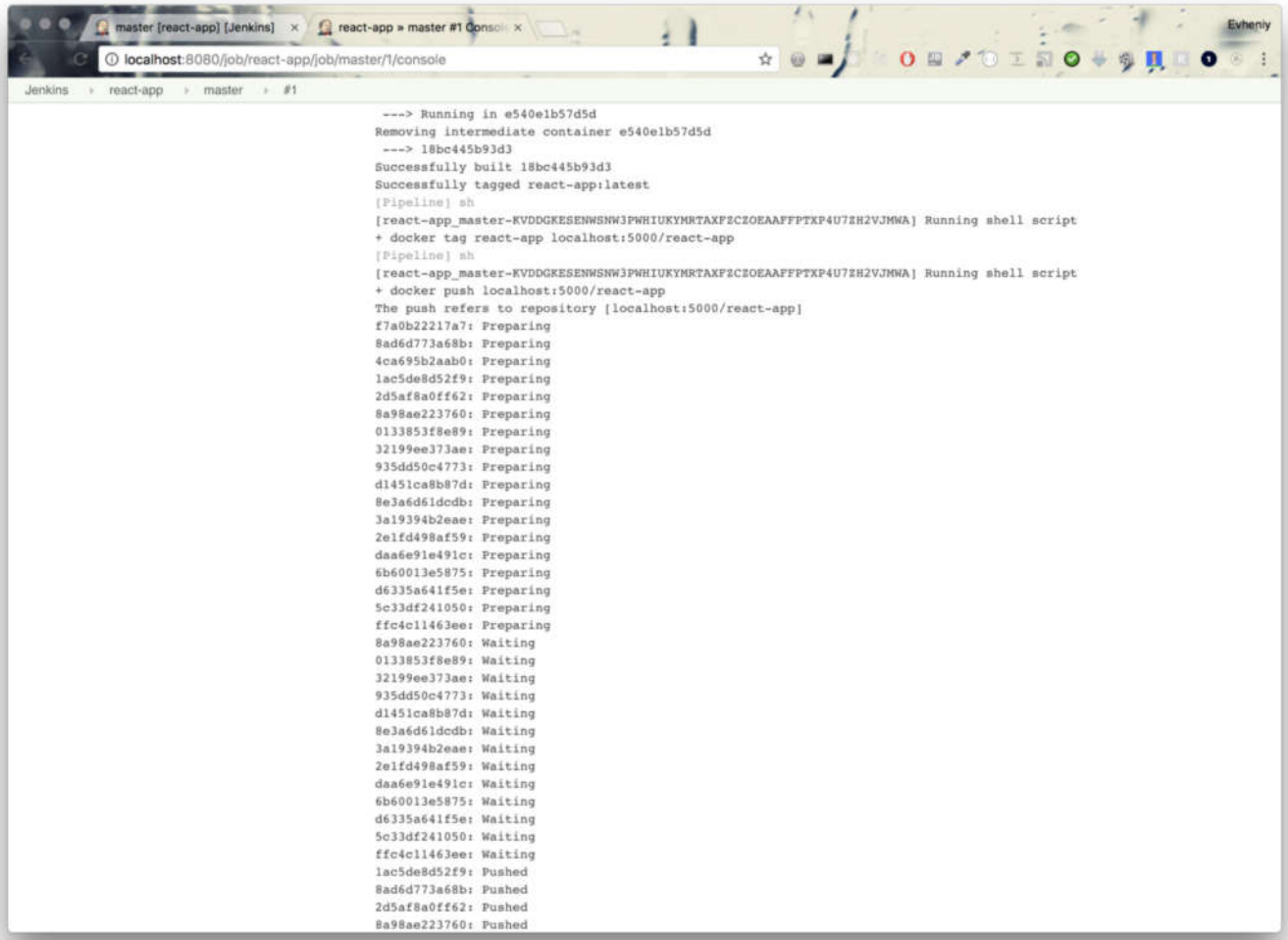
And logs:

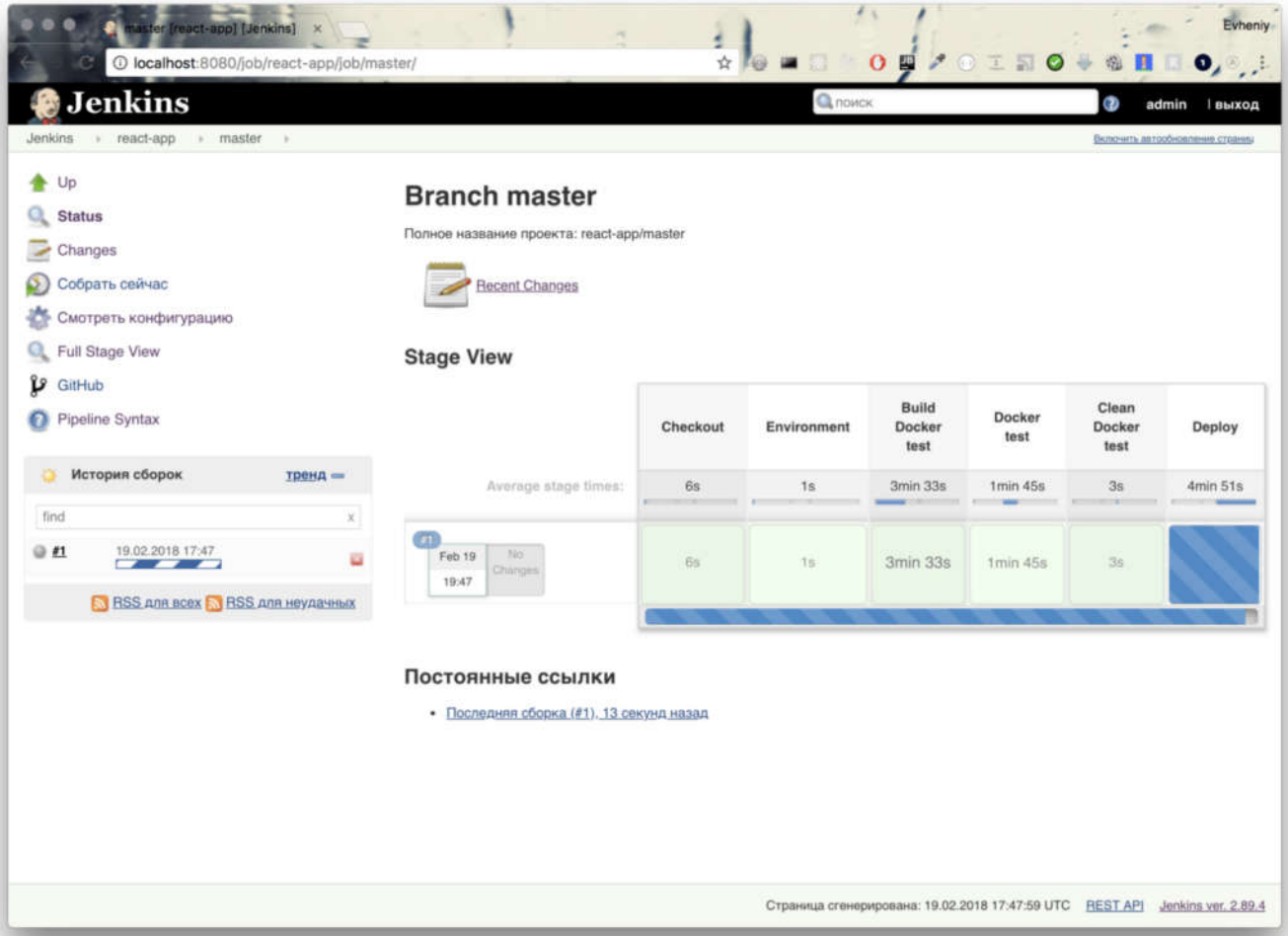

Jenkins
admin
выход
Jenkins » react-app » master »
Up
Status
Changes
Собрать сейчас
Смотреть конфигурацию
Full Stage View
GitHub
Pipeline Syntax
Branch master
Полное название проекта: react-app/master
Recent Changes
Stage View
Success
Logs
Environment
Build Docker test
Average stage times:
1s
4s
Feb 18
17:38
No Changes
6s
1s
История сборок
тренд
find
#1
18.02.2018 15:38
RSS для всех
RSS для неудачных
Постоянные ссылки
Последняя сборка (#1), 7.4 секунд назад
Страница сгенерирована: 18.02.2018 15:38:28 UTC
REST API
Jenkins ver. 2.89.4

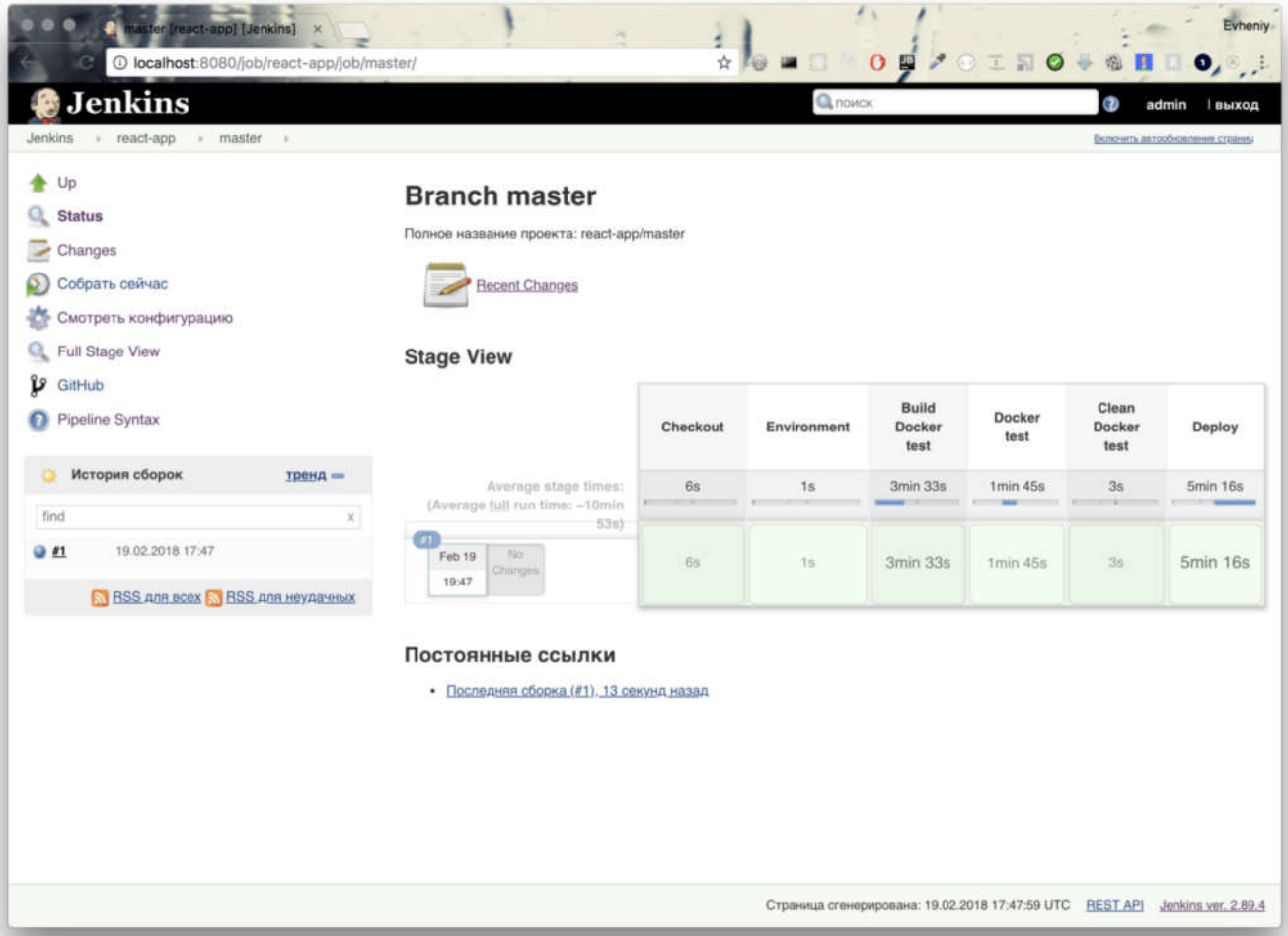

master [react-app] [Jenkins]
react-app » master #1 Consol
Evheny
localhost:8080/job/react-app/job/master/1/console
Jenkins
1
поиск
admin
выход
Jenkins » react-app » master » #1
Back to Project
Status
Changes
Вывод консоли
Просмотреть как неформатированный текст
Edit Build Information
Git Build Data
No Tags
Thread Dump
Pause/resume
Replay
Pipeline Steps
Вывод на консоль
Прогресс:
Branch indexing
15:38:19 Connecting to https://api.github.com using evheniy/******
Obtained Jenkinsfile from 5337f373a7f52726e24f4be85e42ab7b9e288d22
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/react-app_master-KVDDGKESENWSNW3PWHIUKYMR7AXF2CZOEAAFFPTXP4U7ZH2VJMWA
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] checkout
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/evheniy/react-app.git
> git init /var/jenkins_home/workspace/react-app_master-KVDDGKESENWSNW3PWHIUKYMR7AXF2CZOEAAFFPTXP4U7ZH2VJMWA # timeout=10
Fetching upstream changes from https://github.com/evheniy/react-app.git
> git --version # timeout=10
using GIT_ASKPASS to set credentials
> git fetch --no-tags --progress https://github.com/evheniy/react-app.git +refs/heads/master:refs/remotes/origin/master
> git config remote.origin.url https://github.com/evheniy/react-app.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/master:refs/remotes/origin/master # timeout=10
> git config remote.origin.url https://github.com/evheniy/react-app.git # timeout=10
Fetching without tags
Fetching upstream changes from https://github.com/evheniy/react-app.git
using GIT_ASKPASS to set credentials
> git fetch --no-tags --progress https://github.com/evheniy/react-app.git +refs/heads/master:refs/remotes/origin/master
Checking out Revision 5337f373a7f52726e24f4be85e42ab7b9e288d22 (master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 5337f373a7f52726e24f4be85e42ab7b9e288d22
Commit message: "-fix"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Environment)
[Pipeline] sh

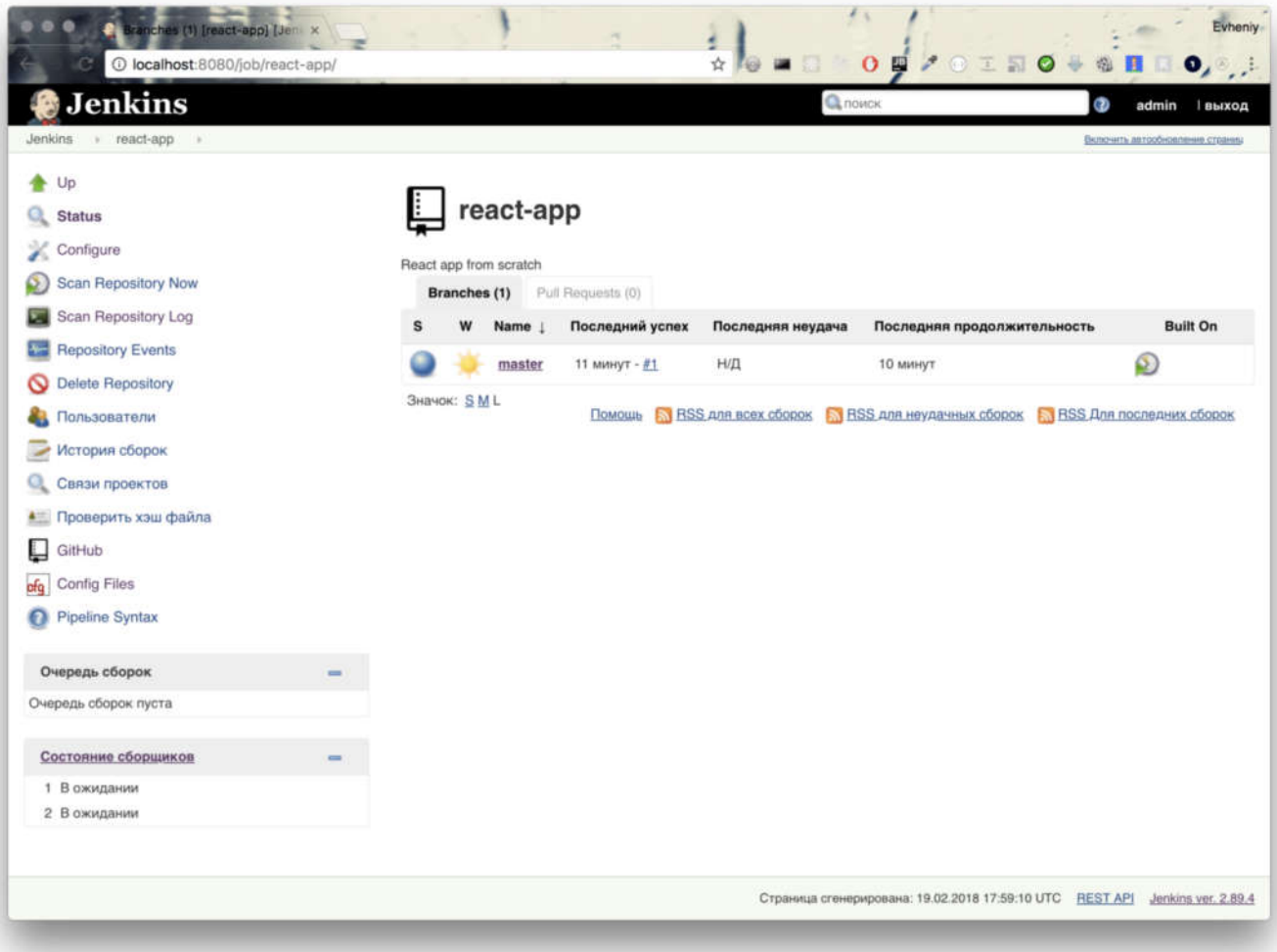


We see a lot of interesting information.


master [react-app] [Jenkins]
react-app » master #1 Console
Evheniy
localhost:8080/job/react-app/job/master/1/console
Jenkins › react-app › master › #1
---> Running in e540e1b57d5d
Removing intermediate container e540e1b57d5d
---> 18bc445b93d3
Successfully built 18bc445b93d3
Successfully tagged react-app:latest
[Pipeline] sh
[react-app_master-KVDDGKESENWSNW3PWHIUKYMRTAXFZCZOEAAFFPTXP4U7ZH2VJMWA] Running shell script
+ docker tag react-app localhost:5000/react-app
[Pipeline] sh
[react-app_master-KVDDGKESENWSNW3PWHIUKYMRTAXFZCZOEAAFFPTXP4U7ZH2VJMWA] Running shell script
+ docker push localhost:5000/react-app
The push refers to repository [localhost:5000/react-app]
f7a0b22217a7: Preparing
8ad6d773a68b: Preparing
4ca695b2aab0: Preparing
1ac5de8d52f9: Preparing
2d5af8a0ff62: Preparing
8a98ae223760: Preparing
0133853f8e89: Preparing
32199ee373ae: Preparing
935dd50c4773: Preparing
d1451ca8b87d: Preparing
8e3a6d61dcdb: Preparing
3a19394b2eae: Preparing
2e1fd498af59: Preparing
daa6e91e491c: Preparing
6b60013e5875: Preparing
d6335a641f5e: Preparing
5c33df241050: Preparing
ffc4c11463ee: Preparing
8a98ae223760: Waiting
0133853f8e89: Waiting
32199ee373ae: Waiting
935dd50c4773: Waiting
d1451ca8b87d: Waiting
8e3a6d61dcdb: Waiting
3a19394b2eae: Waiting
2e1fd498af59: Waiting
daa6e91e491c: Waiting
6b60013e5875: Waiting
d6335a641f5e: Waiting
5c33df241050: Waiting
ffc4c11463ee: Waiting
1ac5de8d52f9: Pushed
8ad6d773a68b: Pushed
2d5af8a0ff62: Pushed
8a98ae223760: Pushed

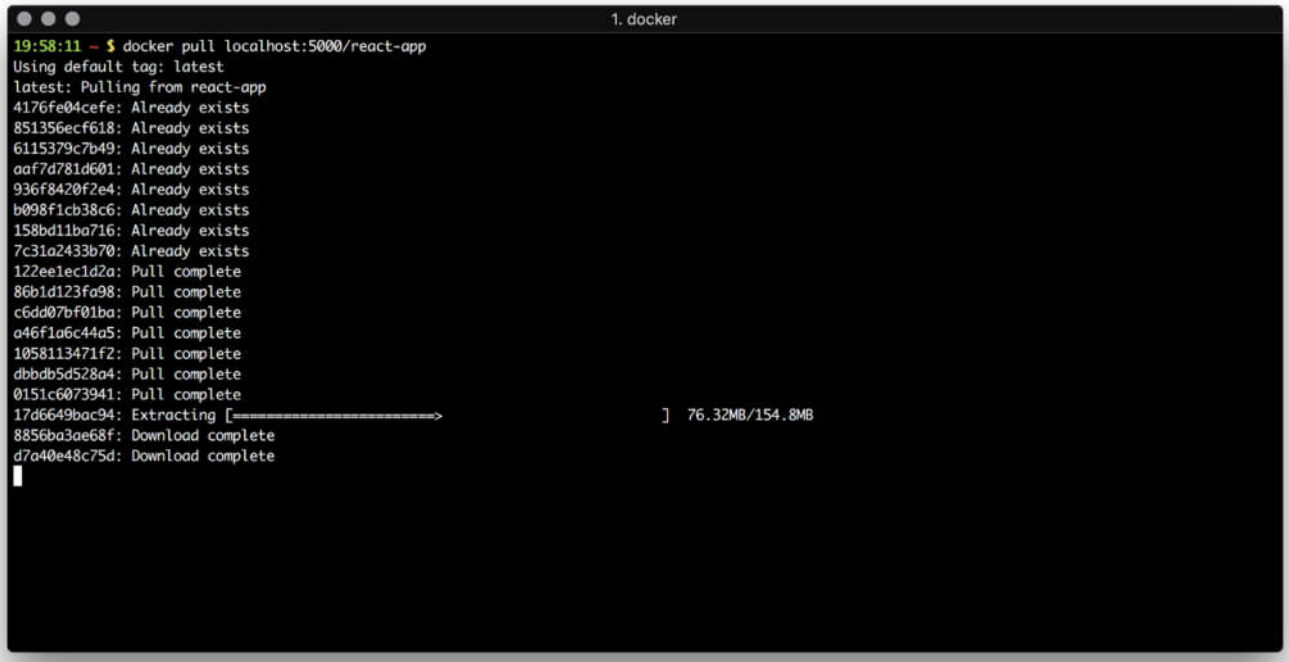

master [react-app] [Jenkins]
Evheniy
localhost:8080/job/react-app/job/master/
Jenkins
поиск
admin
| выход
Jenkins › react-app › master ›
Включить автообновление страниц
Up
Status
Changes
Собрать сейчас
Смотреть конфигурацию
Full Stage View
GitHub
Pipeline Syntax
История сборок
тренд
find
#1
19.02.2018 17:47
RSS для всех
RSS для неудачных
Branch master
Полное название проекта: react-app/master
Recent Changes
Stage View
Checkout
Environment
Build Docker test
Docker test
Clean Docker test
Deploy
Average stage times:
6s
1s
3min 33s
1min 45s
3s
4min 51s
#1
Feb 19
19:47
No Changes
6s
1s
3min 33s
1min 45s
3s
Постоянные ссылки
Последняя сборка (#1), 13 секунд назад
Страница сгенерирована: 19.02.2018 17:47:59 UTC
REST API
Jenkins ver. 2.89.4

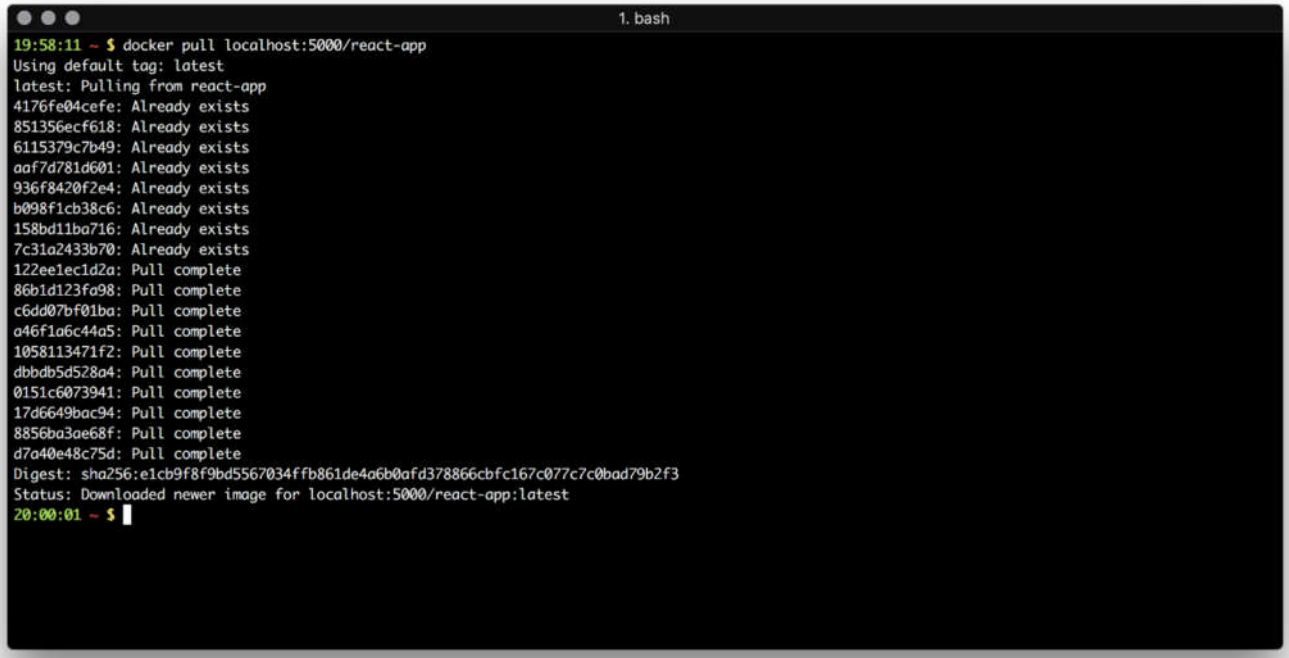

master [react-app] [Jenkins]
Evheniy
localhost:8080/job/react-app/job/master/
Jenkins
поиск
admin
| выход
Jenkins › react-app › master ›
Включить автообновление страниц
Up
Status
Changes
Собрать сейчас
Смотреть конфигурацию
Full Stage View
GitHub
Pipeline Syntax
История сборок
тренд
find
#1
19.02.2018 17:47
RSS для всех
RSS для неудачных
Branch master
Полное название проекта: react-app/master
Recent Changes
Stage View
Checkout
Environment
Build Docker test
Docker test
Clean Docker test
Deploy
Average stage times:
(Average full run time: ~10min 53s)
6s
1s
3min 33s
1min 45s
3s
5min 16s
#1
Feb 19
19:47
No Changes
6s
1s
3min 33s
1min 45s
3s
5min 16s
Постоянные ссылки
Последняя сборка (#1), 13 секунд назад
Страница сгенерирована: 19.02.2018 17:47:59 UTC
REST API
Jenkins ver. 2.89.4



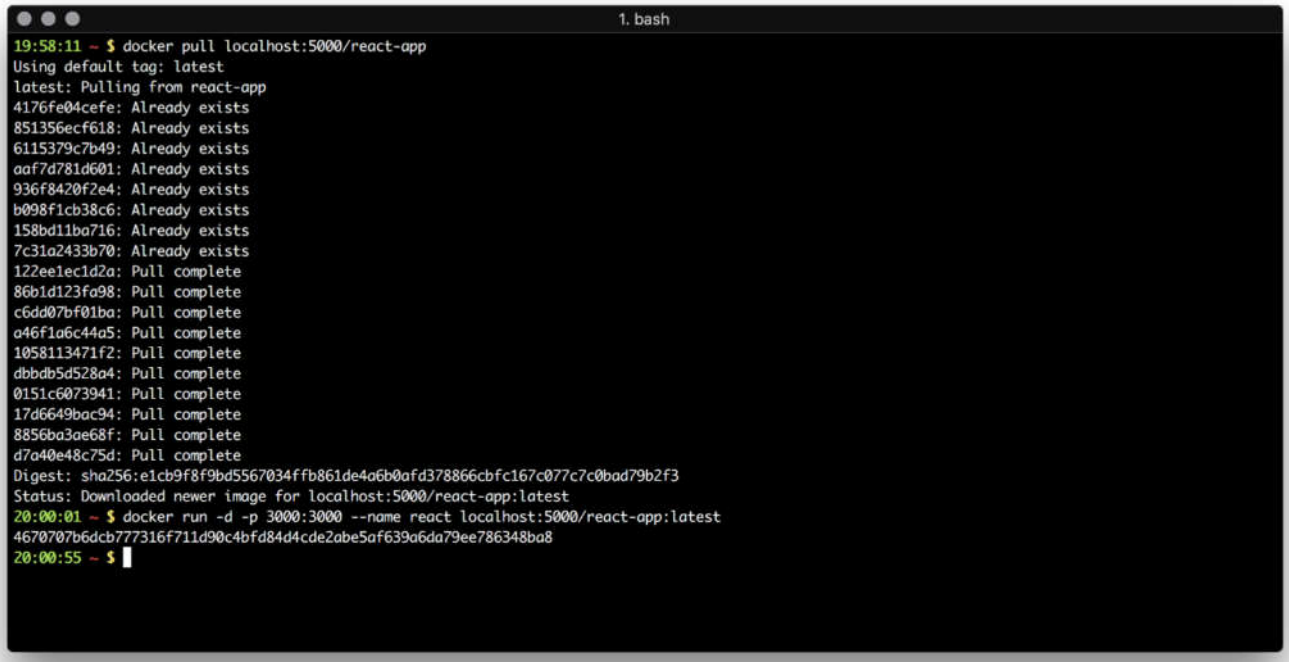
Now we can check that our image stored in our registry. Run command:

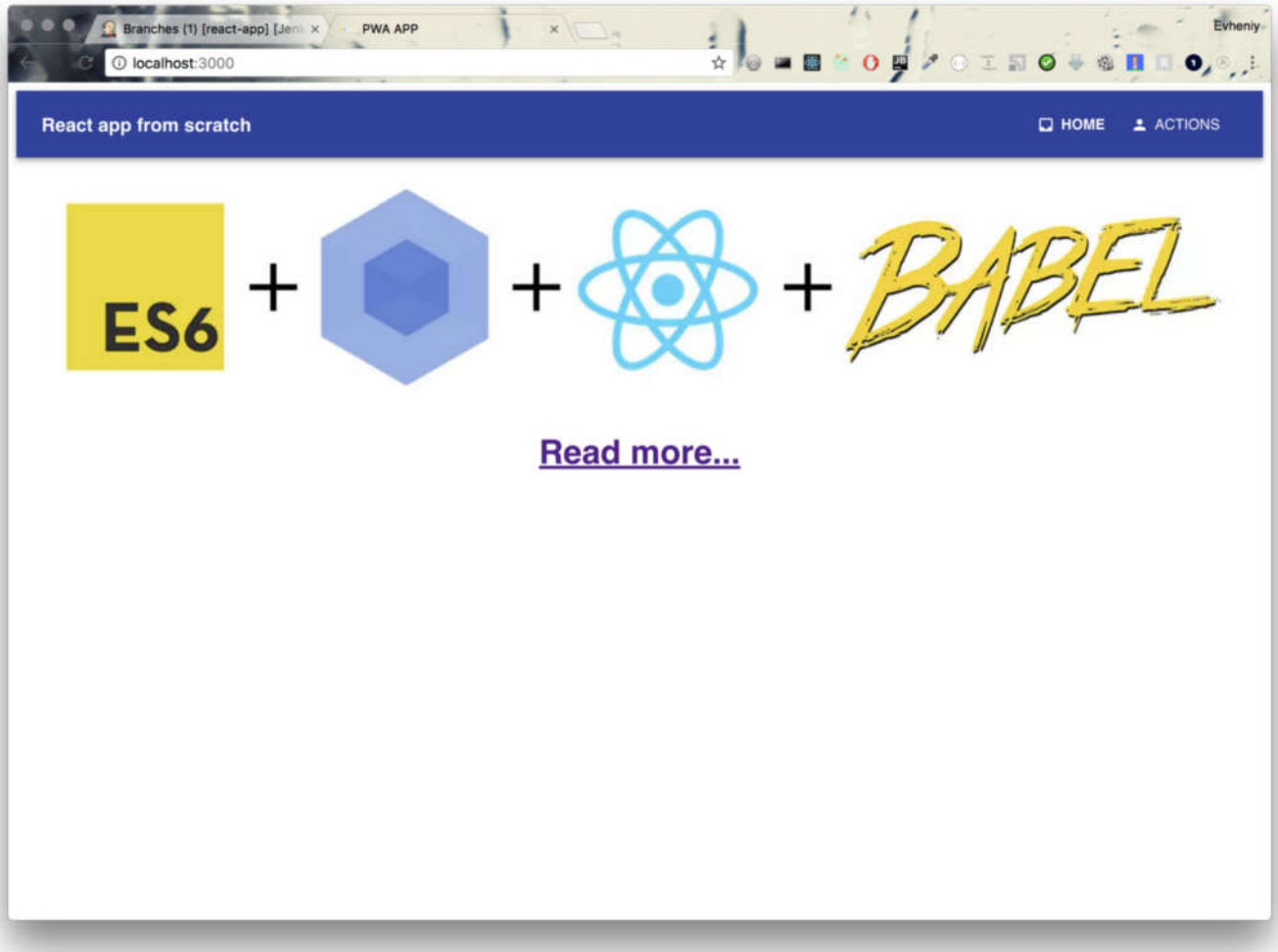
```
docker pull localhost:5000/react-app
```





Now you can run this image on production server.





In this article I created docker compose file for running Jenkins and Docker registry. I created Jenkinsfile and Dockerfile for testing and releasing our app.