

Utopia - Jenkins CI

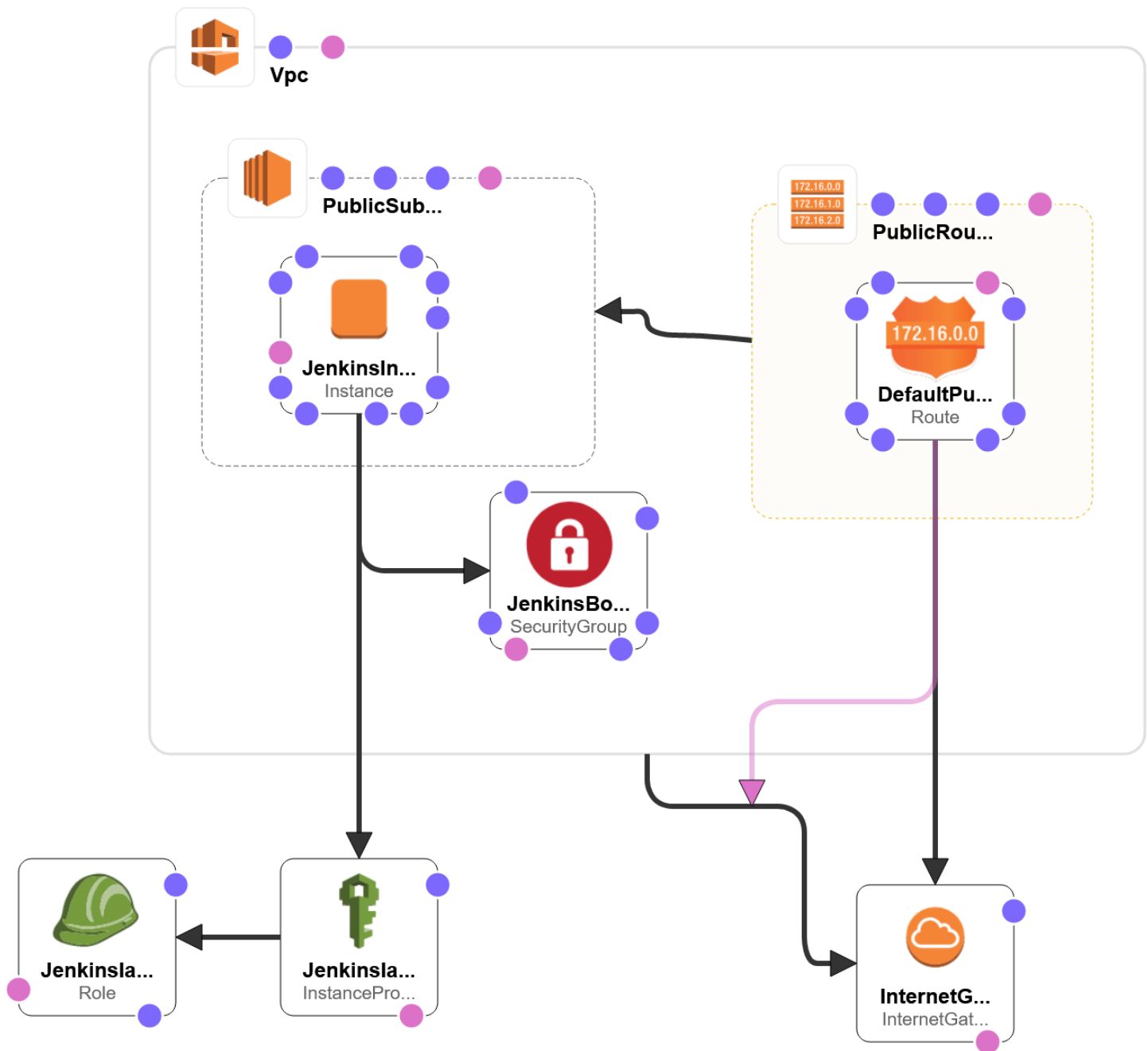
This repository contains all required elements for instantiating a fresh Jenkins continuous integration server with Sonarqube. This is currently deployed on AWS and available at <http://ec2-52-90-241-158.compute-1.amazonaws.com:8080>. Additionally, Sonarqube reports can be accessed at <http://ec2-52-90-241-158.compute-1.amazonaws.com:9000>.

For credentials, please contact [Stephen Gerkin](#).

AWS Infrastructure

The stack created for this instance is fully self-contained and provides its own VPC, Subnet, routing, and EC2 instance with CloudFormation. The template for this is available at [./jenkins-stack.yaml](#). Not included in this repository are the parameters that provide SSH access to the EC2 instance. This can either be provided as a JSON file in `.secret/jenkins-params.json` to be run with the `create.ps1` script for creating a new CF stack, or provided during creation with the AWS CLI.

The template design creates the following stack:



Creation

The [create.ps1](#) file contains the AWS CLI command for creating a new stack. The stack name for this is hardcoded in the script and meant only as a reference for initial creation. If additional copies of this stack should be created, use the following command (substituting as necessary) in a bash terminal:

```
aws cloudformation create-stack \
  --stack-name ${STACK_NAME} \
  --region=us-east-1 \
  --template-body file:///jenkins-stack.yaml \
  --parameters file:///secret/jenkins-params.json \
  --capabilities "CAPABILITY_IAM" "CAPABILITY_NAMED_IAM"
```

Or

```
aws cloudformation create-stack `
  --stack-name $env:STACK_NAME `
  --region=us-east-1 `
  --template-body file://jenkins-stack.yaml `
  --parameters file://.secret/jenkins-params.json `
  --capabilities "CAPABILITY_IAM" "CAPABILITY_NAMED_IAM"
```

Both commands assume the presence of an included JSON file for the parameters that should be written as:

```
[
  {
    "ParameterKey": "AccessCidr",
    "ParameterValue": "FULL CIDR FOR SSH ACCESS"
  },
  {
    "ParameterKey": "SSHKeyName",
    "ParameterValue": "EXISTING SSH KEY NAME"
  }
]
```

The stack uses a custom AMI that includes all Utopia projects as pipeline projects within Jenkins as of 2021-03-16. Future revisions of the AMI will need to be provided as a parameter to the stack. The keyname for this property is `ImageId` and can be included in the above mentioned JSON file:

```
[
  {
    "ParameterKey": "ImageId",
    "ParameterValue": "ami-____"
  }
]
```

Jenkins UI

The Jenkins instance includes an install of the Blue Ocean UI for simpler pipeline management. Additional pipelines can be included by entering the Blue Ocean UI and clicking `New Pipeline`. From there, follow the prompts as appropriate and click `Create Pipeline`:

Jenkins

PipelinesAdministration

Logout

Create PipelineClassic Item Creation

✓

Where do you store your code?

Bitbucket Cloud

Bitbucket Server

GitHub

GitHub Enterprise

Git

✓

Which organization does the repository belong to?

sgerkin-ss

jms-smoothstack-utopia

○

Choose a repository

Loaded 11 repositories

Search...

ss-utopia-customers

ss-utopia-email-lambda

ss-utopia-eureka

ss-utopia-flights

ss-utopia-jenkins

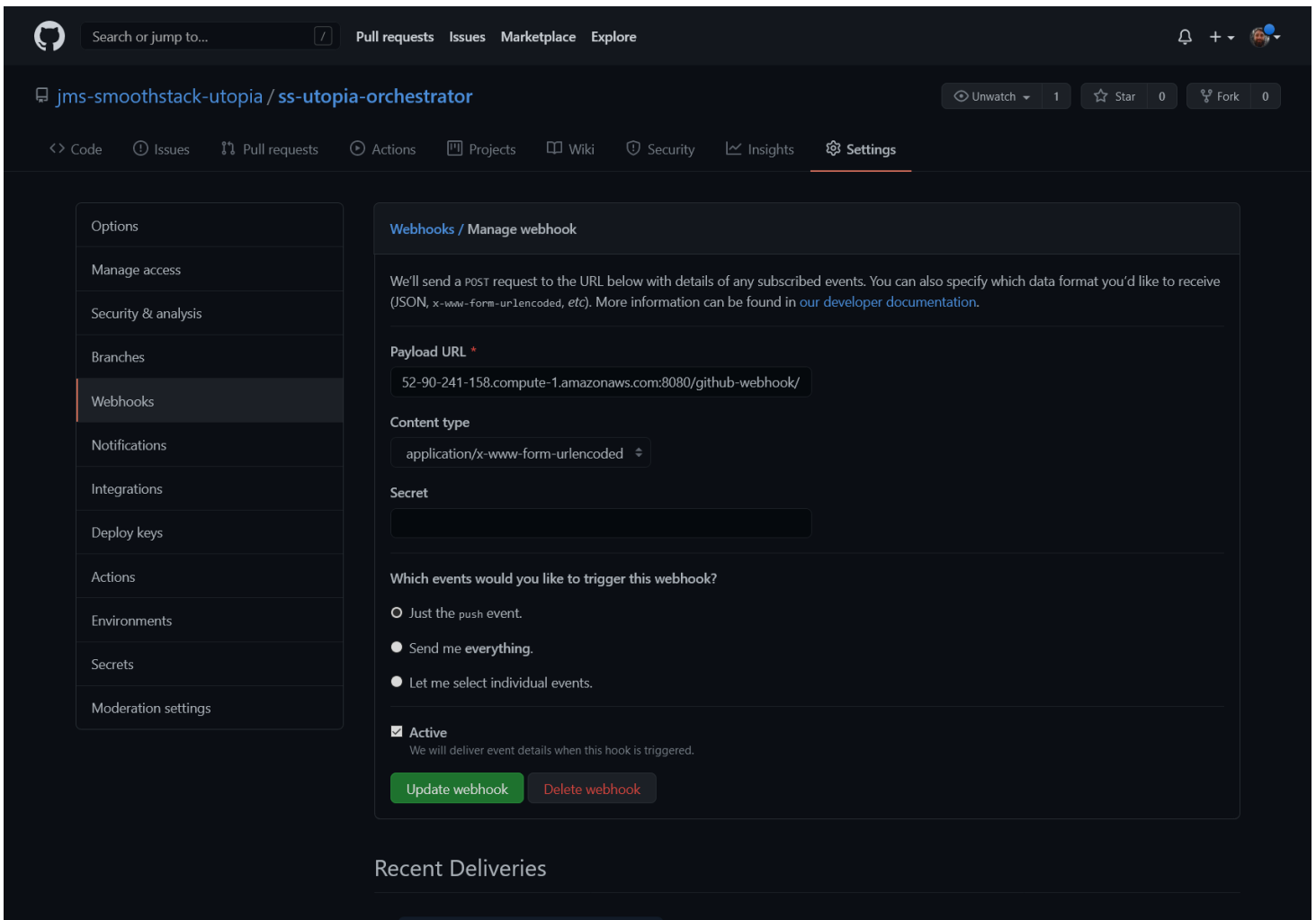
Create Pipeline

○

Complete

1.24.4 · Core 2.277.1 · d692e74 · 22nd January 2021 03:24 PM

Additionally, a webhook must be created on GitHub to allow it to push changes to the repository with a payload URL of `http://ec2-52-90-241-158.compute-1.amazonaws.com:8080/github-webhook/`



Unconfigured Instance

For a fresh, unconfigured instance with no credentials initialized (useful for creating a new AMI) can be created with the included [userdata-unconfigured.sh](#) file. This will create a new Jenkins and Sonarqube instance with no configuration.

The userdata file additionally includes the creation of a service daemon that will automatically turn on Jenkins and Sonarqube should the instance be rebooted or stopped and restarted for any reason.

After creating an EC2 with the userdata file, SSH into the instance to get the initial admin password for Jenkins by executing

```
$ docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword
```

Then open the Jenkins UI (on port 8080) to configure the instance with the username `admin` and the aforementioned password.

Sonarqube will also need administrative configuration and can be accessed with the default `admin:admin` username and password (on port 9000).

Once configured, a new AMI can be created from the configured settings and the [jenkins-stack.yaml](#) template can be used with the new AMI ID.

Example Build

Using the Blue Ocean UI, the full build pipeline can be clearly visualized. Each stage of the build can be inspected and log output for that individual stage can be reviewed for feedback for problems.

An example of a passing build is seen here:

The screenshot displays the Jenkins Blue Ocean UI for a build named 'ss-utopia-auth'. The top navigation bar includes tabs for Pipeline, Changes, Tests, Artifacts, and a Logout button. The build status is 'Success' (indicated by a green checkmark). The pipeline consists of 11 stages: Start, Clean target, Lint, Test and Package, Code Analysis: SpotBugs, Code Analysis: PMD, Code Analysis: Sonarqube, Await Quality Gateway, Build Docker image, Push image to repository, and End. All stages are marked with green checkmarks, indicating a successful build. The 'Push image to repository' stage is currently selected, showing its log output. The log output for this stage includes commands for logging into AWS ECR, pushing the Docker image, and cleaning the Docker environment. The build was completed on 2021-03-16T16:32:27Z.

```
Push image to repository - 18s

> aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 247293358719.dkr.ecr.us-east-1.amazonaws.com — Shell Script 4s
> mvn docker:push — Shell Script 14s
v mvn clean -Ddocker.removeMode=all docker:remove — Shell Script 5s

1 + mvn clean -Ddocker.removeMode=all docker:remove
2 [INFO] Scanning for projects...
3 [INFO]
4 [INFO] -----< com.github.jms-smoothstack-utopia:ss-utopia-auth >-----
5 [INFO] Building Utopia Auth Service 1.0
6 [INFO] -----[ jar ]-----
7 [INFO]
8 [INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ ss-utopia-auth ---
9 [INFO] Deleting /var/jenkins_home/workspace/ss-utopia-auth_main/target
10 [INFO]
11 [INFO] --- docker-maven-plugin:0.33.0:remove (default-cli) @ ss-utopia-auth ---
12 [INFO] DOCKER> ss-utopia/auth-service: Remove
13 [INFO] DOCKER> ss-utopia/auth-service:1.0: Remove
14 [INFO]
15 [INFO] BUILD SUCCESS
16 [INFO]
17 [INFO] Total time: 2.665 s
18 [INFO] Finished at: 2021-03-16T16:32:27Z
19 [INFO] -----

v docker system prune -f — Shell Script <1s

1 + docker system prune -f
2 Deleted Images:
3 deleted: sha256:b1a168540972a5a3ffeb205adccdf182c50e4a687bcc238f9cf2342bfa79682
4 deleted: sha256:13c5c7374890688fd17c51e0bb6ebc062fea6c9045459f0b388dd7e0e26eae40
5 deleted: sha256:e96030ef9baae3ddd92801a3cb40c1abe735117694c9c6364e5f299e448a0a
6 deleted: sha256:71486f02e84b51120ff9108e6e72e8804333b7f80e259e761055129a551c635b
7 deleted: sha256:2e1b3e344fa4126be3c4eba0f816c70a9fd9382b38d24263fa94d2d7a2688009
8 deleted: sha256:37f709c99cf8f71bc1af1cfb71bf52463d9c566db9c4ab9ea44c6f0514fa3a0
9
10 Total reclaimed space: 238.2MB
```

Should a build fail, we can inspect the specific stage for the specific causes for the failure:

ss-utopia-auth 1

PipelineChangesTestsArtifactsLogout

Branch: breakstuff

17s

No changes

Commit: --

2 days ago

Push event to branch breakstuff

StartClean targetLintTest and PackageCode Analysis: SpotBugsCode Analysis: PMDCode Analysis: SonarqubeAwait Quality GatewayBuild Docker imagePush image to repositoryEnd

Lint - 6s

Restart Link

mvn checkstyle:check -- Shell Script

6s

1+ mvn checkstyle:check

2[INFO] Scanning for projects...

3[INFO]

4[INFO] -----< com.github.jms-smoothstack-utopia:ss-utopia-auth >-----

5[INFO] Building Utopia Auth Service 1.0

6[INFO] -----[jar]-----

7[INFO]

8[INFO] --- maven-checkstyle-plugin:3.1.1:check (default-cli) @ ss-utopia-auth ---

9[INFO] Starting audit...

10[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/service/AccountActionTokenService.java:11:3: Missing a

11Javadoc comment. [MissingJavadocMethod]

12[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/entity/UserAccount.java:72:3: Missing a Javadoc comment.

13[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/exception/IllegalCustomerAccountDeletionException.java:11:3:

14Missing a Javadoc comment. [MissingJavadocMethod]

15[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/exception/ExceptionHandlerAdvisor.java:22:3: Missing a

16Javadoc comment. [MissingJavadocMethod]

17[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/exception/ExceptionHandlerAdvisor.java:30:3: Missing a

18Javadoc comment. [MissingJavadocMethod]

19[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/exception/ExceptionHandlerAdvisor.java:50:3: Missing a

20Javadoc comment. [MissingJavadocMethod]

21[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/exception/ExceptionHandlerAdvisor.java:58:3: Missing a

22Javadoc comment. [MissingJavadocMethod]

23[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/exception/ExceptionHandlerAdvisor.java:67:3: Missing a

24Javadoc comment. [MissingJavadocMethod]

25[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/exception/ExceptionHandlerAdvisor.java:75:3: Missing a

26Javadoc comment. [MissingJavadocMethod]

27[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/client/email/AbstractUrlEmail.java:22:3: Missing a Javadoc

28comment. [MissingJavadocMethod]

29[ERROR] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/client/email/AccountConfirmationEmail.java:3:63:

30WhitespaceAround: '{' is not preceded with whitespace. [WhitespaceAround]

31[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/controller/UserAccountController.java:41:3: Missing a Javadoc

32comment. [MissingJavadocMethod]

33[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/controller/UserAccountController.java:51:3: Missing a Javadoc

34comment. [MissingJavadocMethod]

35[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/controller/UserAccountController.java:75:3: Missing a Javadoc

36comment. [MissingJavadocMethod]

37[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/controller/UserAccountController.java:85:3: Missing a Javadoc

38comment. [MissingJavadocMethod]

39[WARN] /var/jenkins_home/workspace/ss-utopia-auth_breakstuff/src/main/java/com/ss/utopia/auth/security/JwtAuthenticationFilter.java:31:3: Missing a Javadoc

40comment. [MissingJavadocMethod]

41Audit done.

42[INFO] -----

43[INFO] BUILD FAILURE

44[INFO] -----

45[INFO] Total time: 3.899 s

46[INFO] Finished at: 2021-03-16T01:21:01Z

47[INFO] -----

48[ERROR] Failed to execute goal org.apache.maven.plugins:maven-checkstyle-plugin:3.1.1:check (default-cli) on project ss-utopia-auth: Failed during

49checkstyle execution: There is 1 error reported by Checkstyle 8.29 with config/google_checks.xml ruleset. -> [Help 1]

50[ERROR]

51[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.

52[ERROR] Re-run Maven using the -X switch to enable full debug logging.

53[ERROR]

54[ERROR] For more information about the errors and possible solutions, please read the following articles:

55[ERROR] [Help 1] <http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException>

56script returned exit code 1

mvn clean -Ddocker.removeMode=all docker:remove -- Shell Script

5s

Sonarqube Results

After the build has gone through the Sonarqube Quality Gates, we can visually inspect the output and address potential security hotspots, code smells, and review code coverage for specific files.

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Search for projects and files...

+

A

Utopia Auth Service

master

March 16, 2021, 3:21 PMVersion 1.1-SNAPSHOT

OverviewIssuesMeasuresCodeActivityAdministration

Quality Gate

Passed

Reliability

Measures

0

A

started 2 days ago

Bugs

New code: since previous version

started 2 days ago

0

A

New Bugs

Security

Measures

0

A

Vulnerabilities

6

Security Hotspots

0

A

New Vulnerabilities

0

New Security Hotspots

Maintainability

Measures

0

A

Debt

0

Code Smells

0

A

New Debt

0

New Code Smells

Coverage

Measures

92.0%

Coverage

66

Unit Tests

—

Coverage on New Code

Duplications

Measures

0.0%

Duplications

0

Duplicated Blocks

0.0%

Duplications on

1 New Lines

About This Project

Utopia Authentication and Authorization Service

No tags

S

1.7k

Lines of Code

Java

1.5k

XML

259

Project Activity

March 16, 2021

1.1-SNAPSHOT

March 16, 2021

1.0

March 16, 2021

Project Analyzed

Show More

Quality Gate

(Default) Sonar way

Quality Profiles

(Java) Sonar way

(XML) Sonar way

External Links

Project's Website

Sources

Project Key

com.github.jms-smoothstack-utopia:

Copy

Get project badges