

神情協 先端技術研究会 「使えるJenkinsを構築する」 第1回

株式会社ジェイエスピー

はじめに

- 開催挨拶
- 技術委員会より挨拶
 - 神情協 技術委員会委員長 山口様
 - 神情協 技術委員会委員 三浦様

本日の予定

1. はじめに
2. 幹事会社紹介、スタッフ紹介
3. 自己紹介
4. この研究会について
5. 使用するツールについて
6. CIについて
7. Jenkinsに触れる
8. 次回以降について



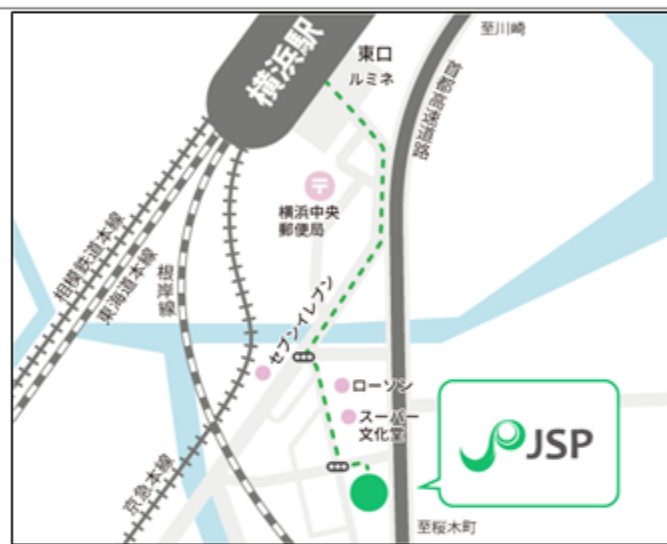
幹事会社紹介

株式会社ジェイエスピー

HP: <https://www.jspnet.co.jp/>

〒220-0011

神奈川県横浜市西区高島2-6-32
横浜東口ウィスパートビル11F



- ソフトウェア開発会社
- IoTの技術が強みに、システムの提案・開発を行う
- 自社製品も開発！

スタッフ紹介

- 講師 : 千田 駿人
- 助手 : 徳永 将太郎
- 質問等があった場合、チャットでコメントをお願いします。
→ Zoom の使い方は後程説明

自己紹介

参加者の皆様より自己紹介をしていただきます！

- 名前／会社名
- 普段の仕事
- 参加理由

この研究会について

目標

「Jenkinsを使えるものにする手段」を知り、
その利用価値を考える

成果物

Jenkinsで実現できることを提案
→ グループで1つ。デモも OK

この研究会について

目標 : PCにJenkinsを構築！

- ・いくつかのツールとの連携
- ・「自動テスト」「CI」を体験

→ 使える？使えない？を考えていただきます

成果物 : 使い方を学び、「生かす」ことを考える！

- ・プログラミング言語が「C#」や「python」なら？
- ・Linuxで動かすとしたら？
- ・Excel 操作も自動化できる？

→ 何を解消できるか？メリット／デメリットは？

「Jenkins実践入門」

この一部 + α を実施

[改訂第3版] Jenkins実践入門

—ビルド・テスト・デプロイを自動化する技術

川口耕介, 佐藤聖規 監修

佐藤聖規, 和田貴久, 新井雄介,

米沢弘樹, 山岸啓, 岩成祐樹 著



今後の予定

| | 日程 | 内容 | 形式 |
|-----|-------|------------------------|--------------|
| 第1回 | 10/10 | CIについて講義／Jenkinsに触れる | オンライン (Zoom) |
| 第2回 | 11/14 | プラグイン／MagicPodの使用 | オンライン (Zoom) |
| 第3回 | 12/12 | MagicPodとの連携／グループ話し合い | オンライン (Zoom) |
| 第4回 | 1/16 | Jenkinsのその他設定／グループ話し合い | オンライン (Zoom) |
| 第5回 | 2/13 | 成果発表／懇親会 | 弊社会議室 |

■ 時間：10:00～12:00

■ 形式について

新型コロナウイルスの影響で第4回までオンラインとする予定です。
第5回についても、
皆様のご要望によりオンラインとする可能性があります。

Zoomについて

第4回までZoomで実施の予定

① 「チャット」

質問等で使用



② 「手を挙げる」「手を下げる」

作業の区切り、質疑応答



Chatworkについて

<用途>

- ・ オンラインで何かあった場合
- ・ 質問等を気軽に行う
- ・ グループでの連絡手段

- 無料版を使用
- 14個までのグループ作成可
- 登録いただいているメールアドレスに招待メールを送ります。

MagicPod について

画面の自動テストを行う(詳細は次回)

外部サーバに接続するアプリを使用する

The screenshot displays the MagicPod web interface within a browser window. The browser's address bar shows the URL `https://www.google.com/?hl=ja`. The interface is divided into two main panels. The left panel, titled 'ブラウザを開く (Cookieをクリアしない)', contains a list of test steps: 1. Enter the URL 'https://www.google.com/?hl=ja' in the address bar. 2. Click the 'Google 検索' button. 3. Click the 'I'm Feeling Lucky' button. 4. Click the 'Google 検索' button again. The right panel shows the Google homepage with the search bar and the 'Google 検索' button. At the bottom of the interface, there is a green bar with the text '最終テスト実行結果: ● 成功 #50 (詳細)' and a green button labeled '実行'.

MagicPodのアカウント作成

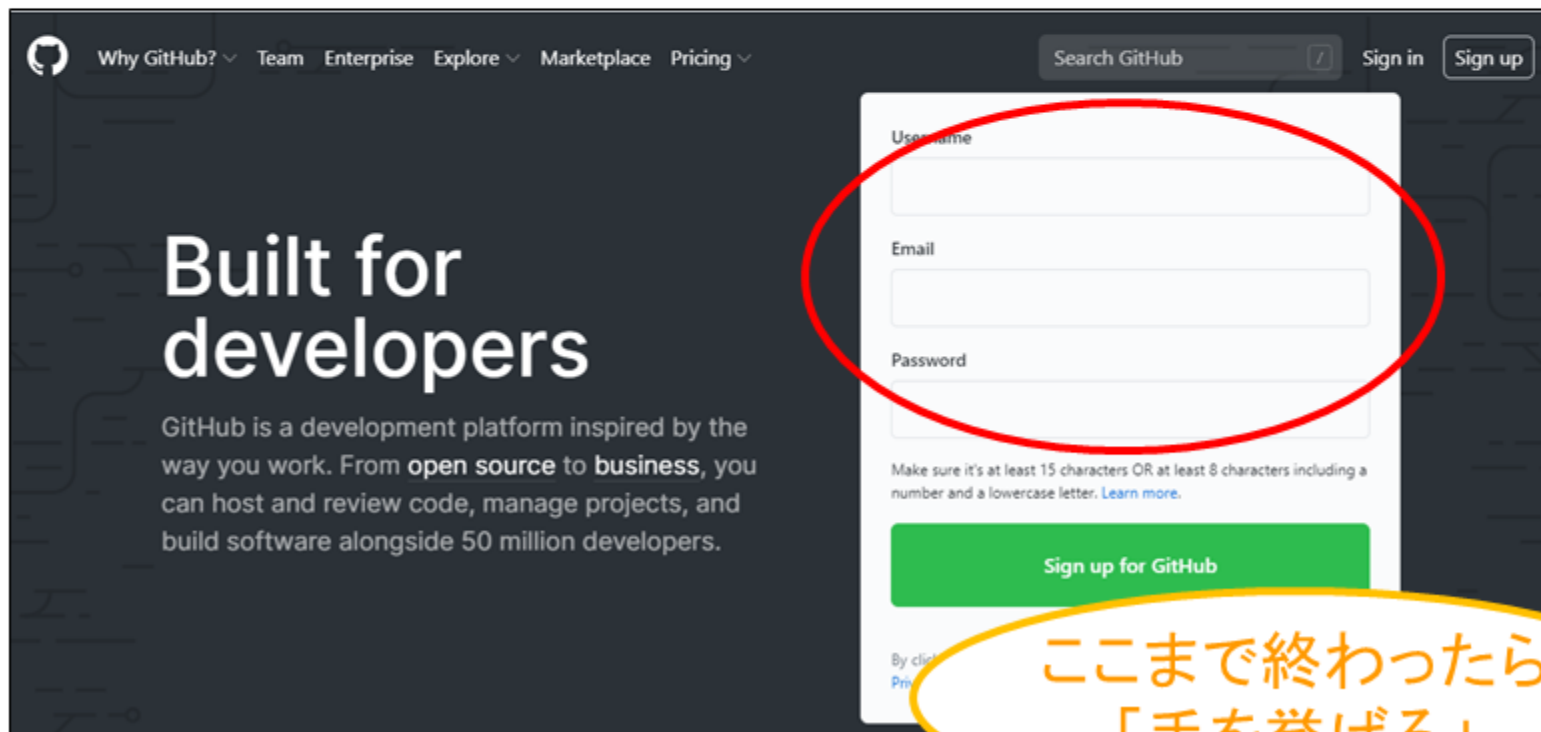
- ① <https://www.magic-pod.com/> にアクセス
- ② 「新規登録」で登録 ※会社名は自身の会社名でお願いします。
- ③ ユーザ名を教えてください！次回までに使えるようにします。



ここまで終わったら
「手を挙げる」

GitHubのアカウント作成

- ① <https://github.com> にアクセス
- ② ユーザ名、メールアドレス、パスワードを入力登録



The screenshot shows the GitHub homepage with a dark background. On the left, the text "Built for developers" is prominent, followed by a description of GitHub as a development platform. On the right, there is a white sign-up form. The form has three input fields: "Username", "Email", and "Password". These three fields are grouped together and circled in red. Below the "Password" field, there is a note: "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)". At the bottom of the form is a green button labeled "Sign up for GitHub". This button is circled in yellow. In the top right corner of the page, there are links for "Sign in" and "Sign up".

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Policy](#).

ここまで終わったら
「手を挙げる」

本日の予定

1. はじめに
2. 幹事会社紹介、スタッフ紹介
3. 自己紹介
4. この研究会について
5. 使用するツールについて
6. CIについて
7. Jenkinsに触れる
8. 次回以降について

ココから本題！

第1回について

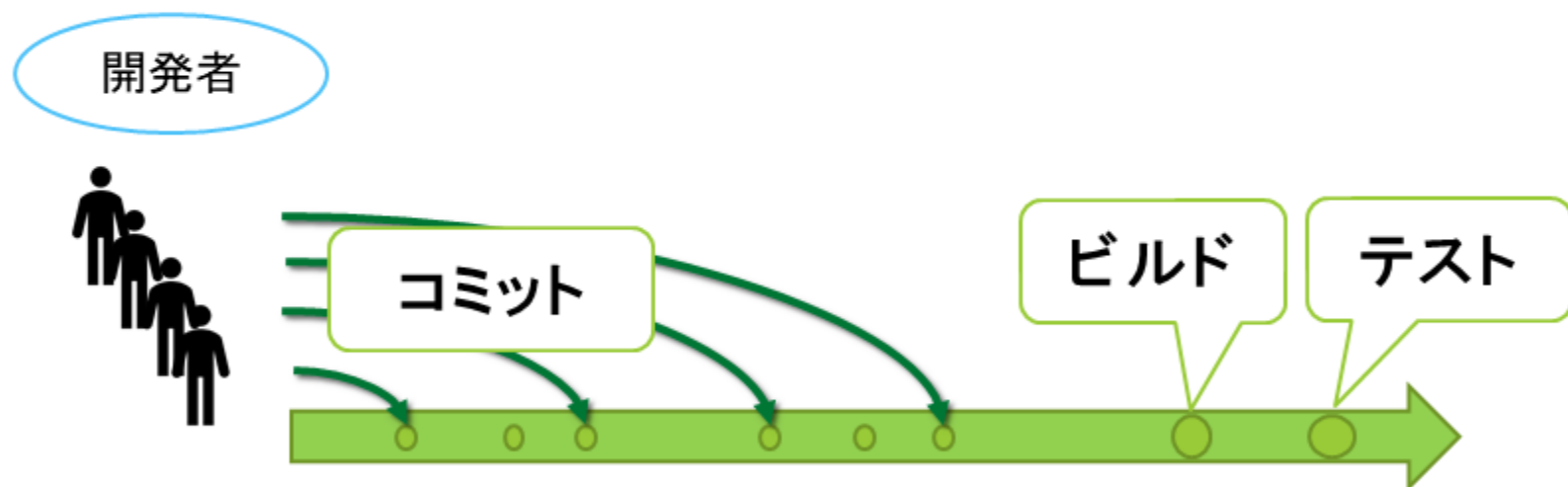
<目標>

- ① CI と Jenkins について知る
- ② Jenkins で簡単な自動テストをする

キーワード : CI、自動テスト、Jenkins、プラグイン

どんな開発形式ですか？

例えば、以下のような開発をしてませんか？



CI とは？

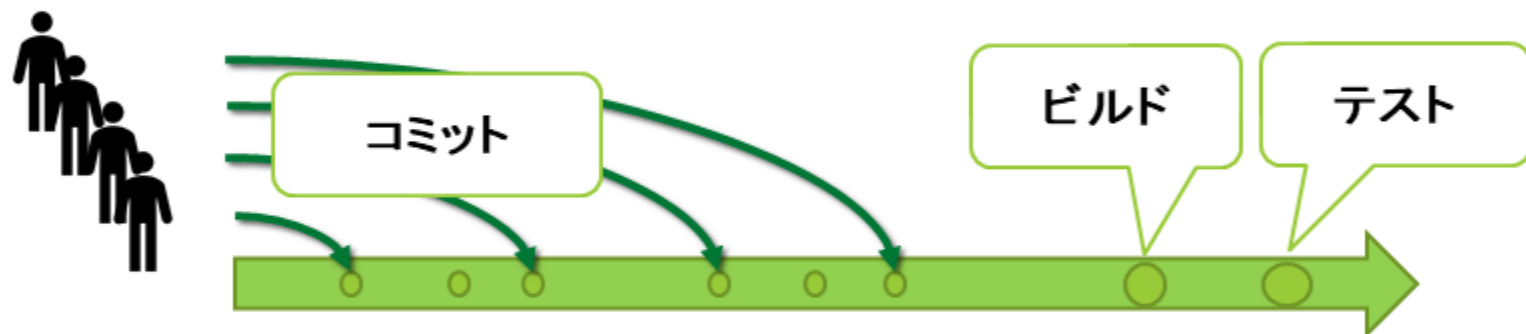
Continuous Integration（継続的インテグレーション）

ソフトウェア開発において以下を実現すること

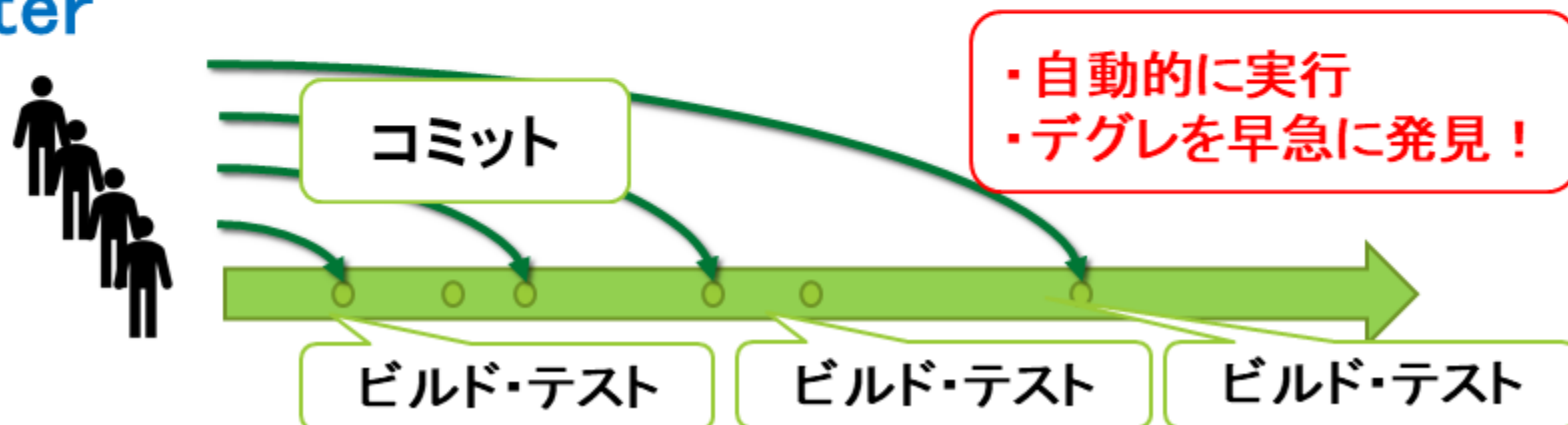
1. 定期的な自動ビルドと自動テスト
2. バグの早期発見と修正
3. 開発の品質と生産性の向上

CI を導入するなら

Before



After



CI を導入するなら



メリットは？デメリットは？

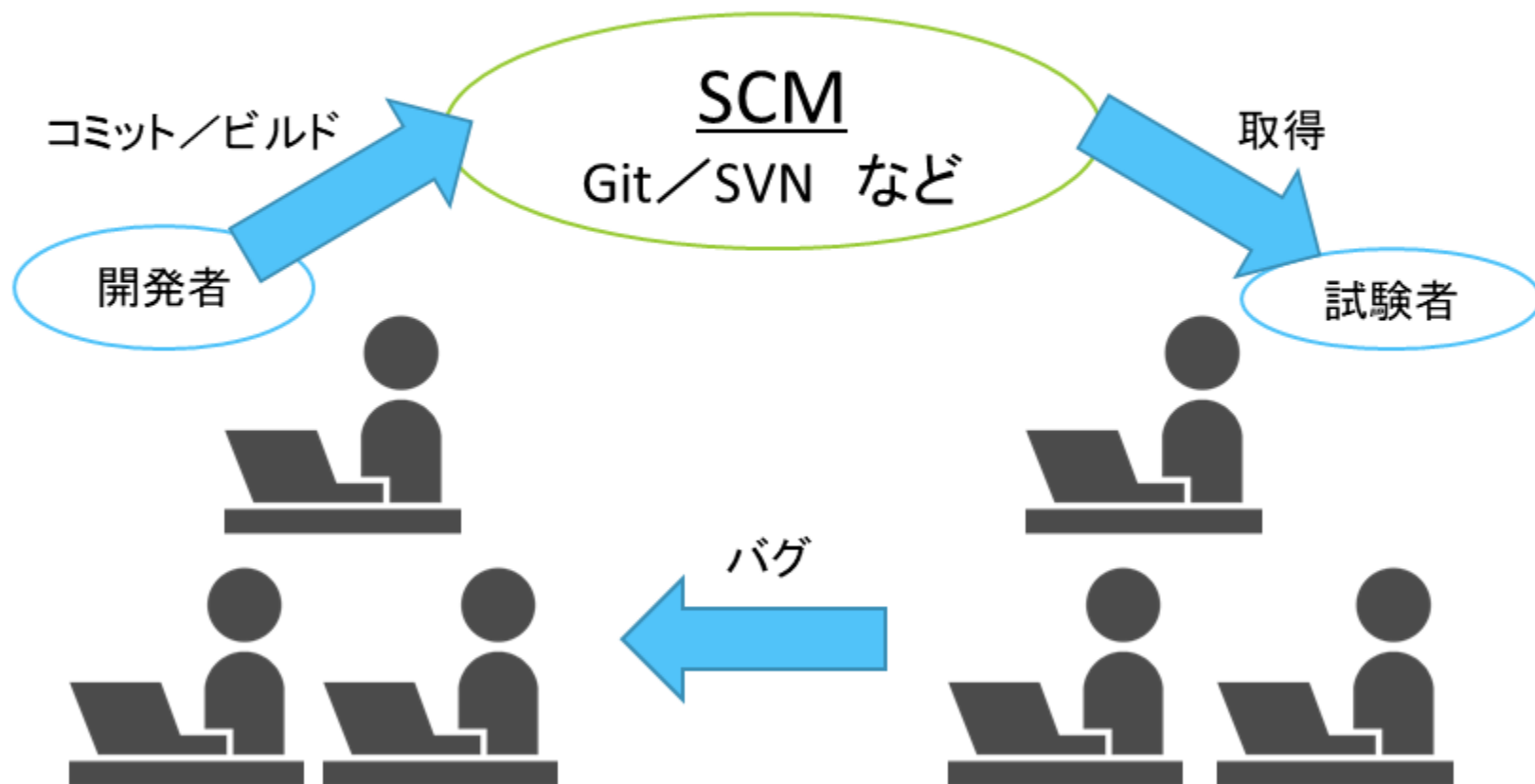
<メリット>

- ・テスト自動化でデグレを防ぐ = 品質向上！
- ・人の手によるテスト時間を減らせる = 生産性向上！

<デメリット>

- ・環境を整えるコストがかかる
- ・導入ですべてが「よくなるわけではない」
 - 自動テストは効果の高いものを選択すべき

どんな開発環境ですか？

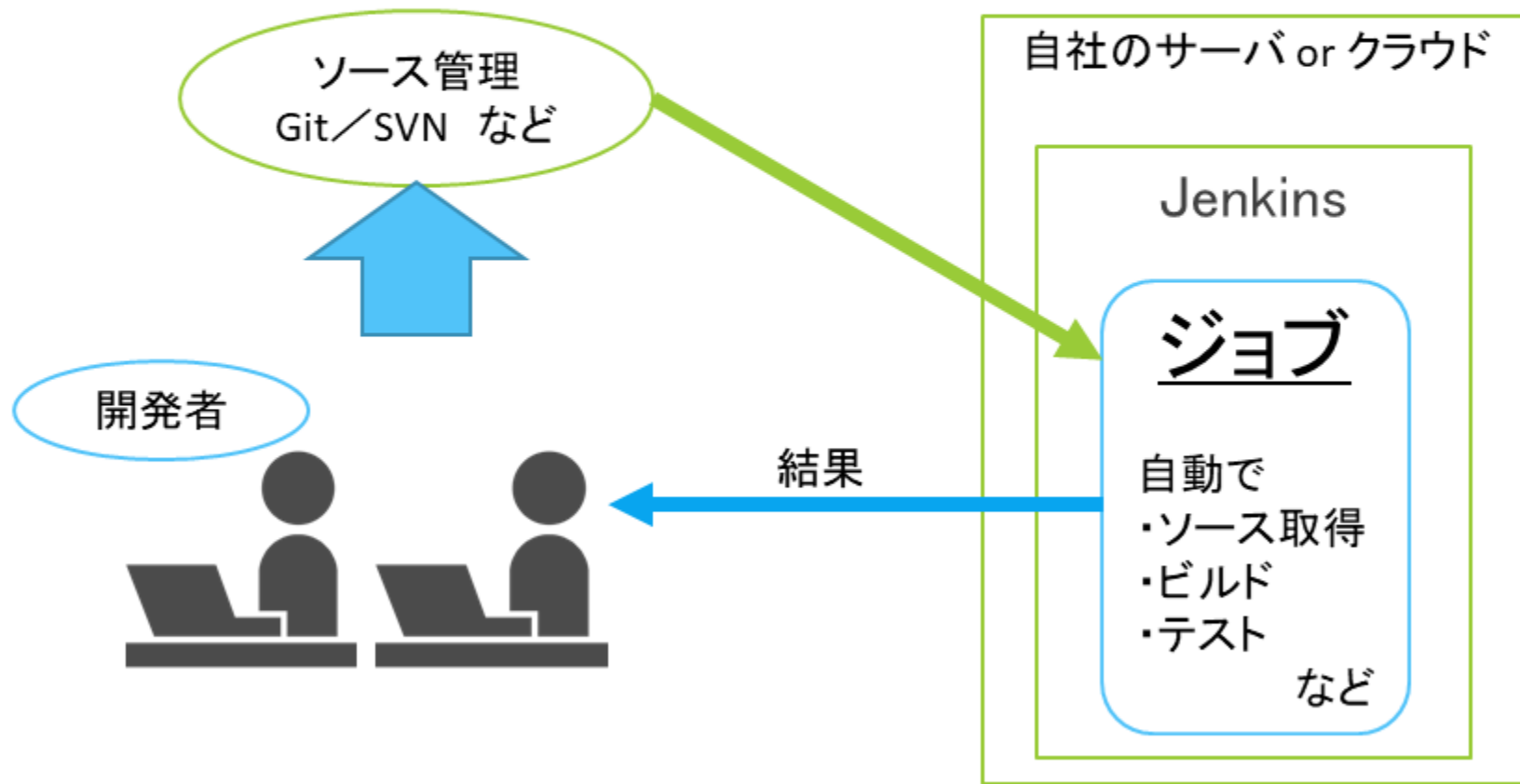


Jenkinsについて

CIを実現するツール

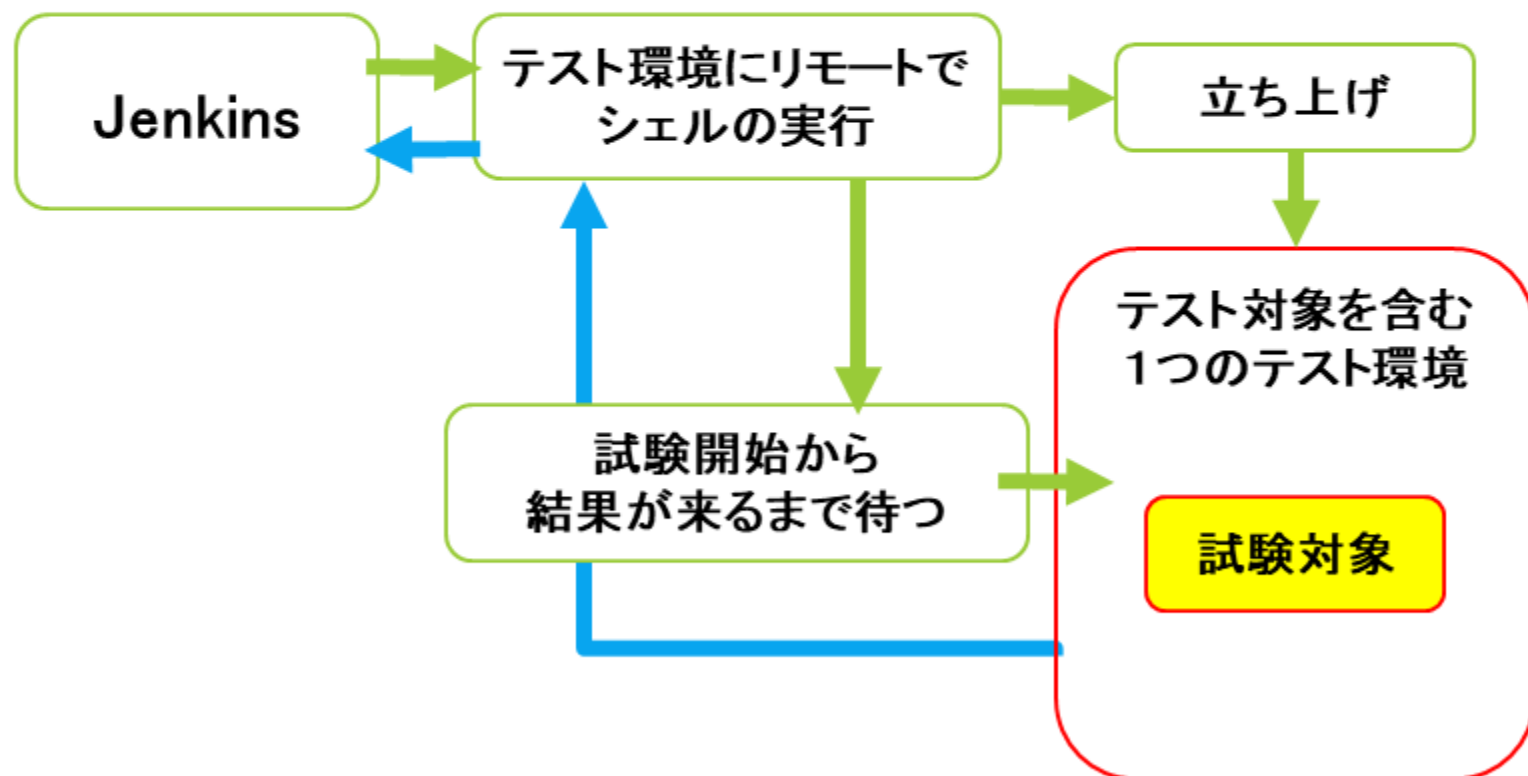
- ・Javaで作られたOSS
- ・分かりやすいUI
- ・プラグインが多い
 - いろいろなツールと連携可能

よく使われている構成

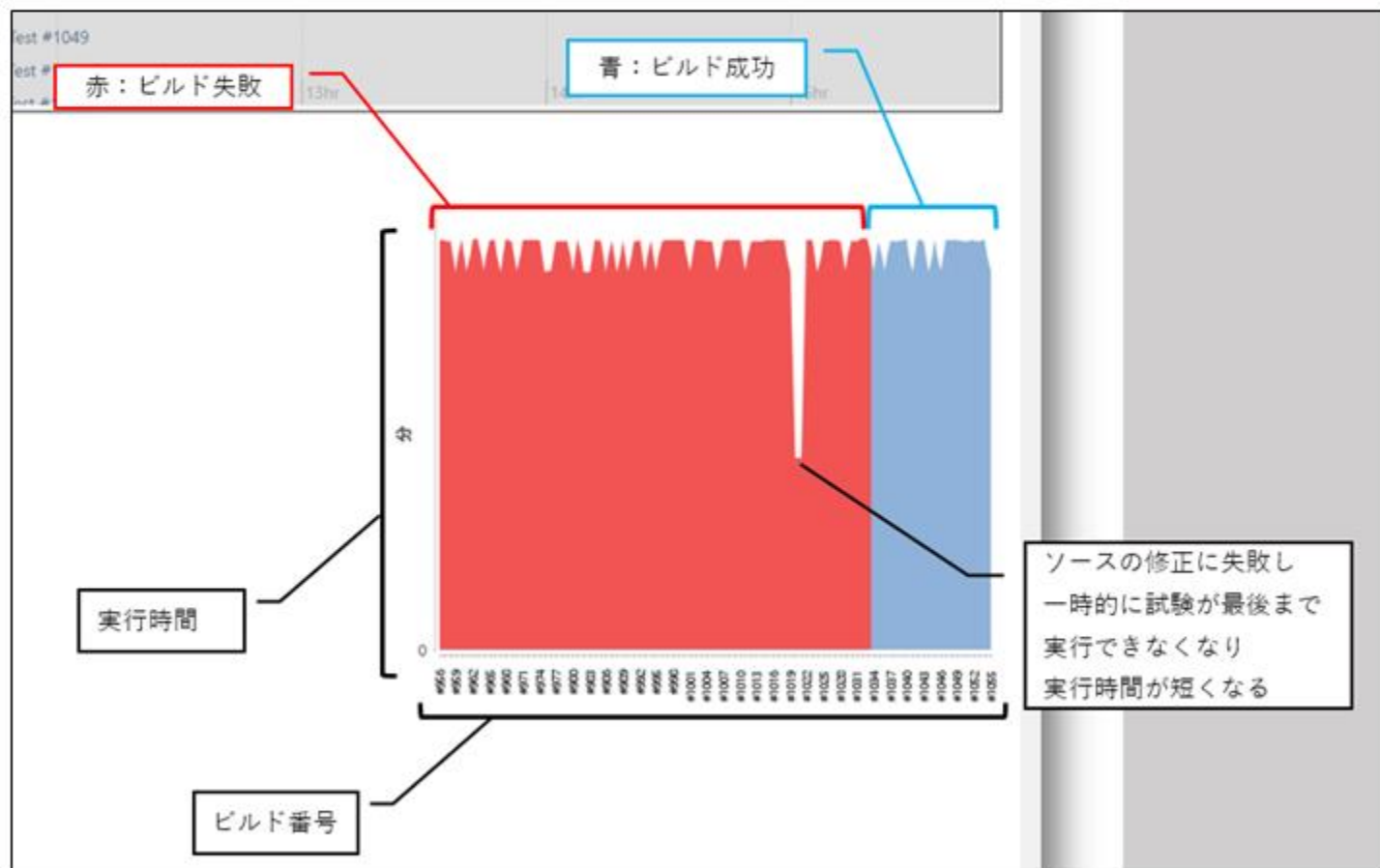


実例

弊社のある案件の試験にて...



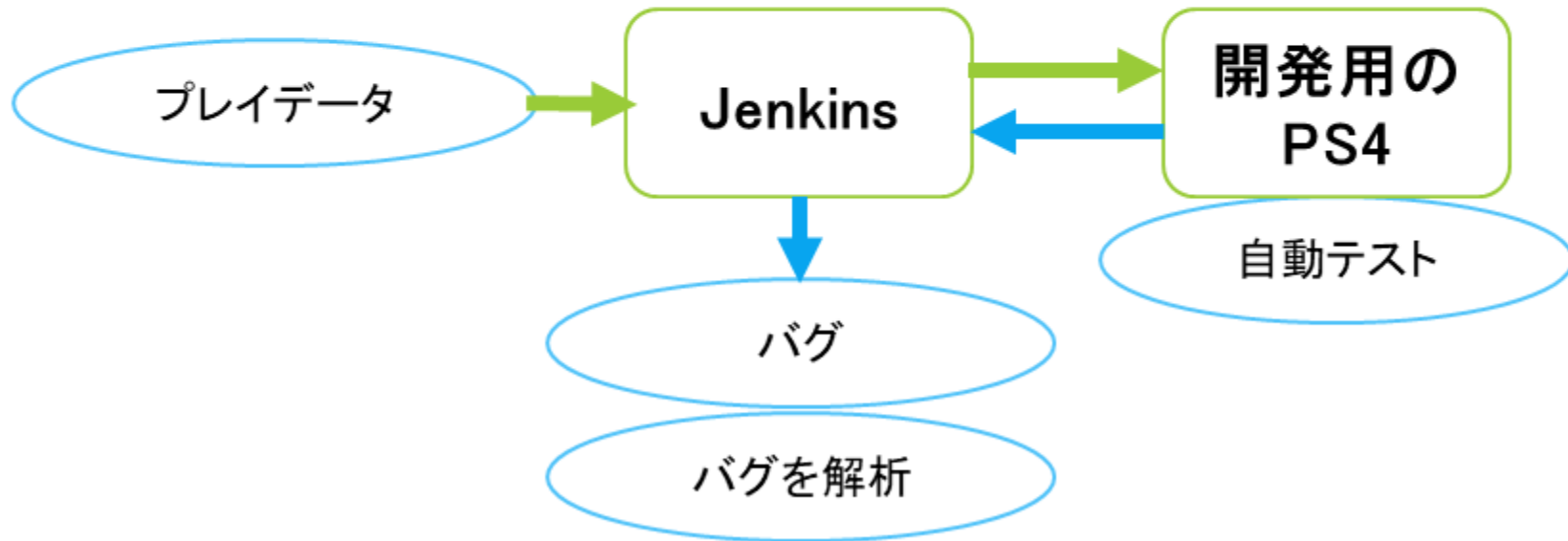
結果



ゲーム開発でも

『FFVII リメイク』は自動デバッグで、休日、夜間問わず、毎日数百回も
通しプレイ中。ゲームのバグを自動で検知するシステムを開発
【CEDEC 2020】

<https://www.famitsu.com/news/202009/07205335.html>



本日の予定

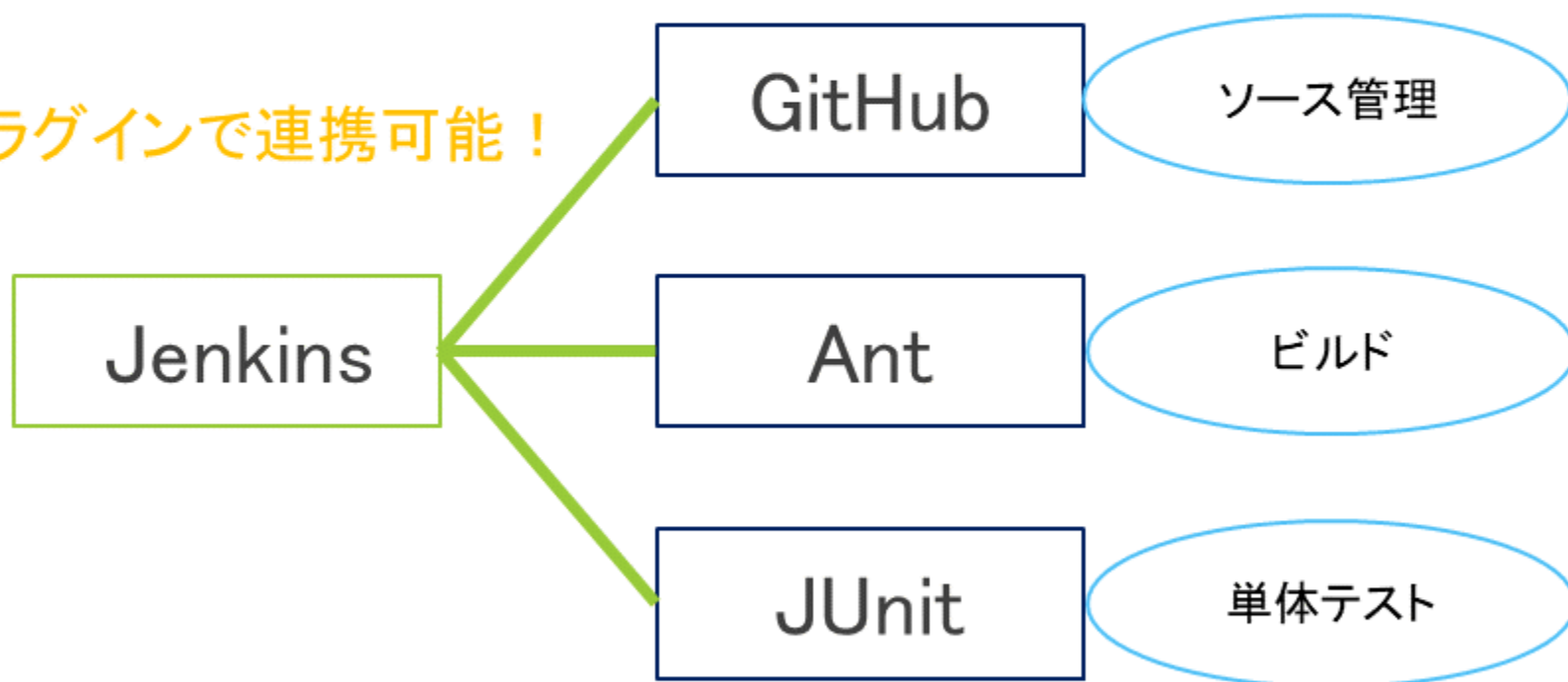
1. はじめに
2. 幹事会社紹介、スタッフ紹介
3. 自己紹介
4. この研究会について
5. 使用するツールについて
6. CIについて
7. Jenkinsに触れる
8. 次回以降について

Jenkins に触ってみる

1. ジョブの作成
2. GitHubからソースを取得する
3. Ant でビルドをする
4. JUnit でテストをする

これのでできること

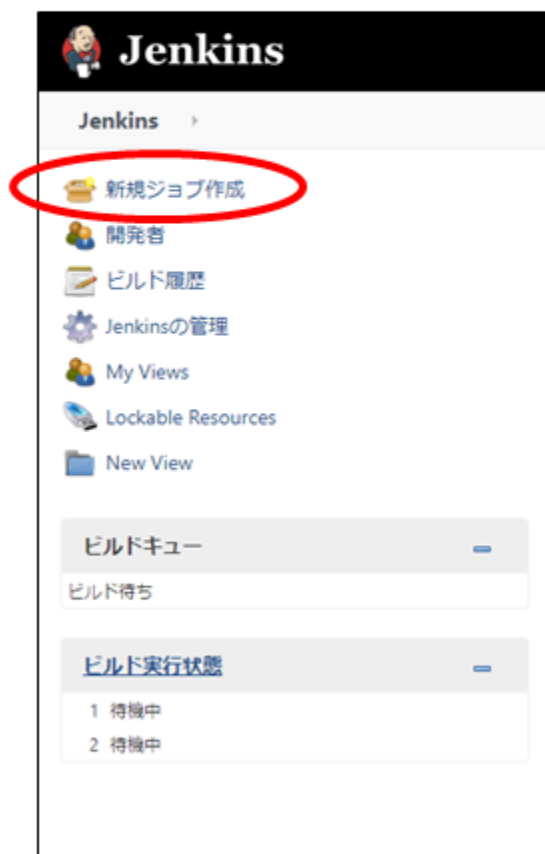
プラグインで連携可能！



Jenkins に触ってみる

1. ジョブの作成
2. GitHubからソースを取得する
3. Ant でビルドをする
4. JUnit でテストをする

ジョブ作成



ジョブの設定①

①

Enter an item name

SampleJob

» Required field

ジョブ名を入力

②



フリースタイル・プロジェクトのビルド

もっとも汎用性の高いJenkinsの中核機能です。任意のSCMからソースコードがビルドできます。往々にして、ソフトウェアのビルド以外にも様々な

ジョブの形式を選択



パイプライン

複数のビルドスレーブにまたがる長時間に渡る処理を編成します。（以前は、ワークフローとして知られていた）パイプラインの構築に適しており、フリースタイルジョブでは上手く扱えない複雑な処理を容易に編成することができます。



マルチ構成プロジェクトのビルド

複数の環境でのテストや、プラットフォームごとのビルドなどといった、多数の異なる構成が必要なプロジェクトに適しています。



GitHub Organization

Scans a GitHub organization (or user account) for all repositories matching some defined markers.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

アイテムを格納するコンテナを作成します。プロジェクトなどをまとめるのに便利です。単なるフィルタであるビューとは違

ジョブの設定

1. 【ビルド】を押す。
2. 【Windowsバッチコマンドの実行】を選択する。
3. 次を入力する。

echo "Hello World"



ビルド実行



ビルド結果表示



The screenshot shows the Jenkins web interface for a build. The breadcrumb navigation is 'Jenkins > SampleJob > #1'. On the left sidebar, the 'コンソール出力' (Console Output) link is circled in red, with a large red arrow pointing down to the second screenshot. The main content area shows 'ビルド #1 (2020/09/06 17:34:35)' with a blue sphere icon. Below this, it states '変更点はありません。' (No changes) and 'ユーザー hayato_senda が実行' (Executed by user hayato_senda). The sidebar also includes links like 'プロジェクトへ戻る', '状態', '変更履歴', '説明の編集', and 'Delete build '#1''.



The screenshot shows the Jenkins web interface for the console output of build #1. The breadcrumb navigation is 'Jenkins > SampleJob > #1'. The left sidebar is the same as the first screenshot, but 'コンソール出力' is now selected. The main content area shows 'コンソール出力' with a blue sphere icon. Below this, it states 'ユーザー hayato_senda が実行' (Executed by user hayato_senda), 'Running as SYSTEM', and 'ビルドします。ワークスペース: C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins-Jenkins\workspace\SampleJob' (Building. Workspace: C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins-Jenkins\workspace\SampleJob). The console output shows the command 'C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins-Jenkins\workspace\SampleJob [SampleJob] \$ cmd /c call C:\Windows\TEMP\Jenkins5729252959136084372.bat' and the result 'C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins-Jenkins\workspace\SampleJob "Hello World" Finished: SUCCESS'. A yellow oval highlights the bottom right of the console output area with the text 'ここまで終わったら「手を挙げる」' (After this is over, raise your hand).

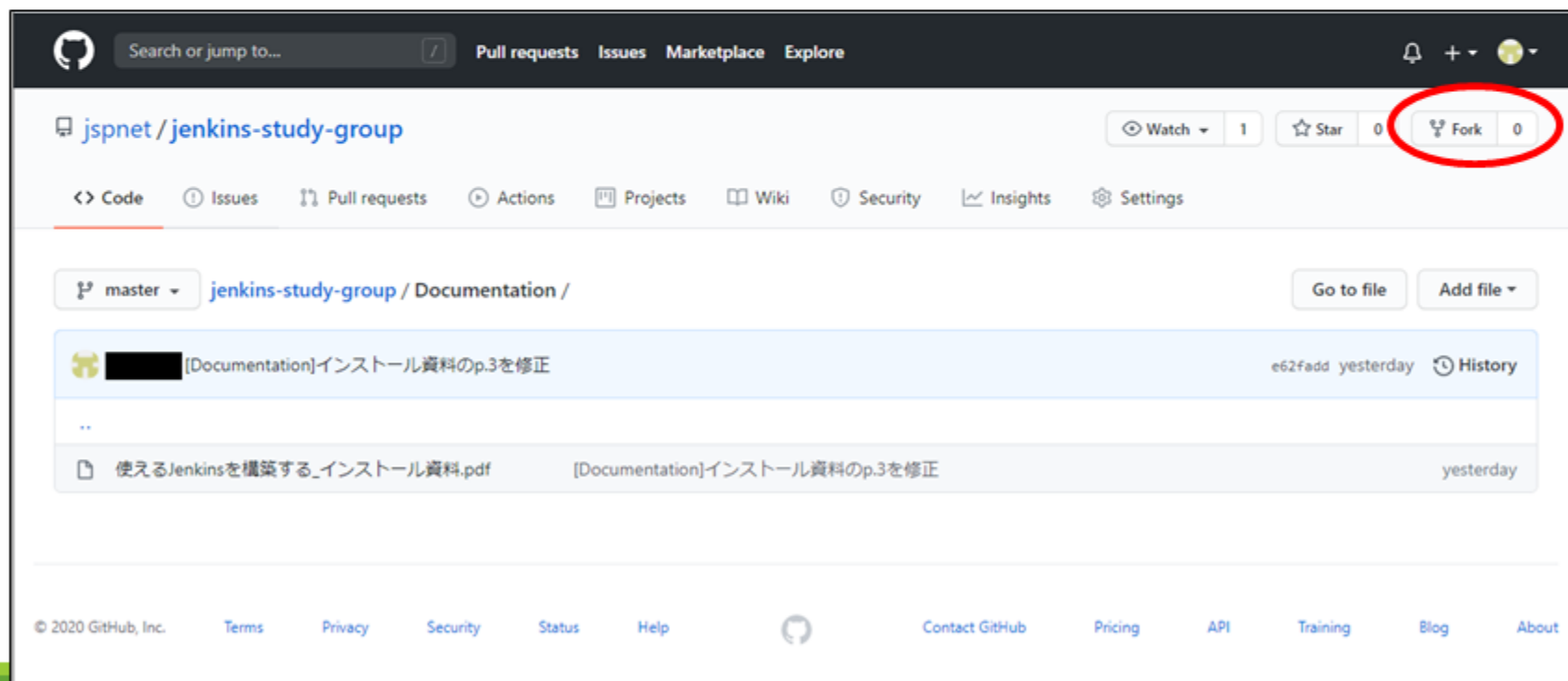
Jenkins に触ってみる

1. ジョブの作成
2. GitHubからソースを取得する
3. Ant でビルドをする
4. JUnit でテストをする

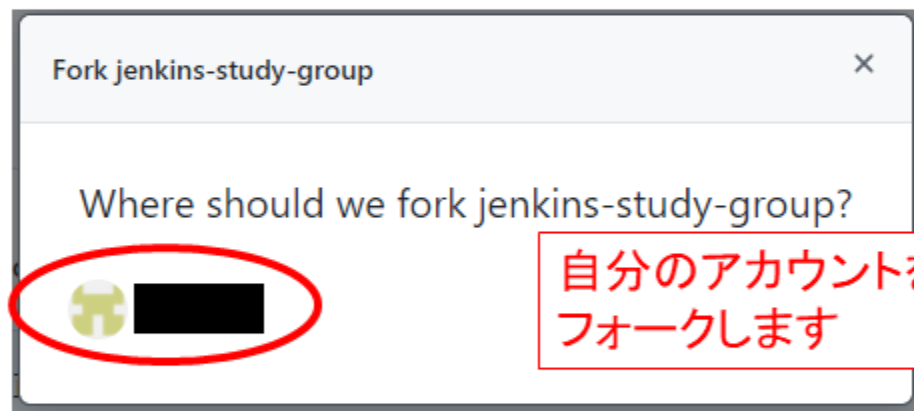
GitHubのアカウントを使います！

サンプルコードを取得①

- ① <https://github.com> にアクセスし、自分のアカウントにログイン
- ② <https://github.com/jspnet/jenkins-study-group> にアクセスする



サンプルコードを取得②



自分のアカウントをクリックし、
フォークします

サンプルコードを取得③

The screenshot shows the GitHub interface for a repository named 'jenkins-study-group'. The repository is forked from 'jspnet/jenkins-study-group'. The main content area displays the commit history for the 'master' branch, showing a merge of the 'master' branch from the parent repository. The commit history table lists the following files and their commit messages:

| File | Commit Message | Time |
|---------------|--------------------------------|------------|
| Documentation | [Documentation]インストール資料のp.3を修正 | yesterday |
| LICENSE | Initial commit | 3 days ago |
| README.md | 説明を追加しました | 2 days ago |

The README.md file is visible at the bottom of the page, showing the repository name 'jenkins-study-group'.

ジョブの設定①

ソースコード管理

☒ Git

リポジトリ

リポジトリURL

 Gitのリポジトリを入力してください。

認証情報

ビルドするブランチ

ブランチ指定子 (空欄はすべてを指定)

リポジトリ・ブラウザ

追加処理

☐ Subversion

ソースコード管理で「Git」を選択する

ジョブの設定②

GitHubのページに戻る

The screenshot shows the GitHub interface for a repository named 'jenkins-study-group'. The repository is forked from 'jspnet/jenkins-study-group'. The 'Code' tab is selected, showing the 'master' branch with 1 branch and 0 tags. A red box highlights the 'Code' button, which is labeled with a circled '1'. A dropdown menu is open, showing the 'Clone' section with the HTTPS URL 'https://github.com/[redacted]/jenkins-study-group'. A red box highlights the copy icon next to the URL, labeled with a circled '2'. A red box also highlights the text 'このリポジトリのURLをコピーする' (Copy the repository URL) overlaid on the file list.

Search or jump to... Pull requests Issues Marketplace Explore

[redacted] /jenkins-study-group
forked from jspnet/jenkins-study-group

<> Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is even with jspnet:master.

h-sen

Documentation [Documentation]インストール資料

LICENSE Initial commit

README.md 説明を追加しました

Clone
HTTPS SSH GitHub CLI
https://github.com/[redacted]/jenkins-study-group
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP

このリポジトリのURLをコピーする

ジョブの設定③

Jenkinsのページに戻る

ソースコード管理

☐ なし
☒ Git

リポジトリ

リポジトリURL

認証情報

ビルドするブランチ

ブランチ指定子 (空欄はすべてを指定)

このリポジトリのURLを貼り付ける

ジョブ実行

Jenkins > SampleRepositoryJob

プロジェクト Sa

「ビルドを実行」を押す
→ビルド履歴で成功が出ればOK!

変更履歴

永続リンク

最新のビルド (#1), 1 分 22 秒

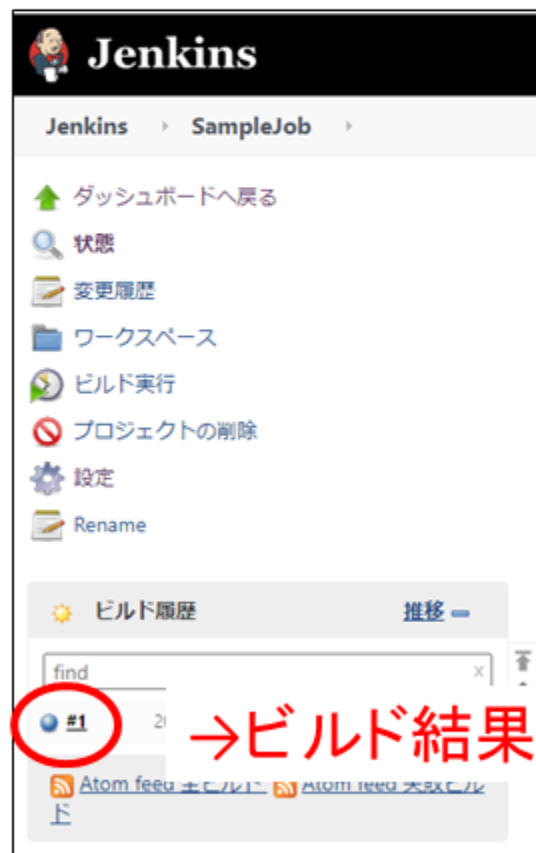
最新の安定ビルド (#1), 1 分 22 秒

どこにクローンされたか

ワークスペースにクローンされます。

ではワークスペースとはどこか？

ビルド結果から確認してみましょう！



ビルド結果結果表示

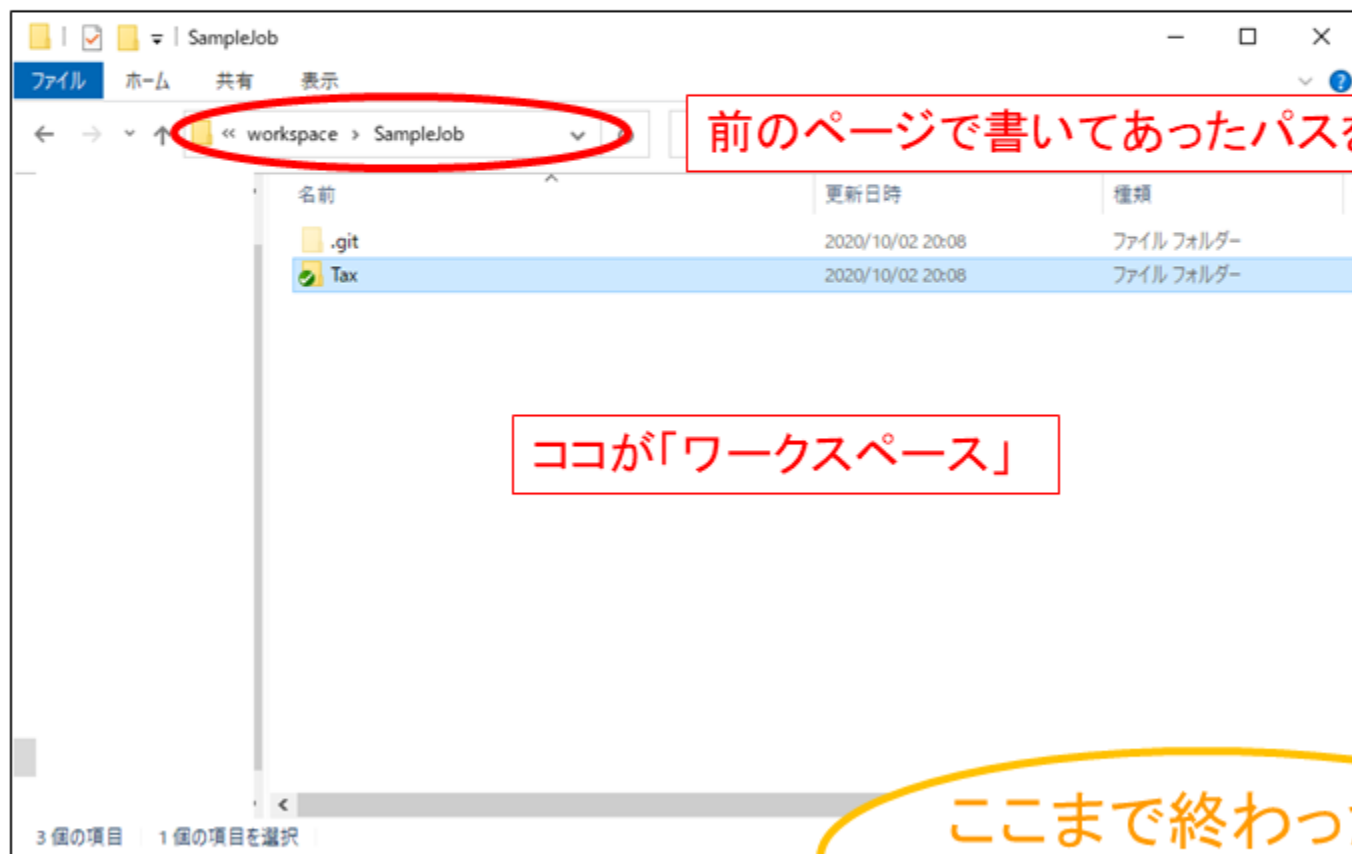


The screenshot shows the Jenkins web interface for a build. The breadcrumb navigation is 'Jenkins > SampleJob > #1'. On the left sidebar, the 'Console Output' link is circled in red, with a large red arrow pointing down to the second screenshot. The main content area shows 'ビルド #1 (2020/09/06 17:34:35)' with a blue sphere icon. Below this, it states '変更点はありません。' (No changes) and 'ユーザー hayato_senda が実行' (Executed by user hayato_senda).



The screenshot shows the Jenkins console output page. The breadcrumb navigation is 'Jenkins > SampleJob > #1'. The left sidebar shows 'コンソール出力' (Console Output) selected. The main content area shows 'コンソール出力' with a blue sphere icon. Below this, it states 'ユーザー hayato_senda が実行' (Executed by user hayato_senda). The console output text is circled in red, and a red box highlights the first line: 'ビルドします。ワークスペース: C:\Windows\System32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\SampleJob [SampleJob] > cd /d C:\Windows\TEMP\Jenkins5729252959136084372.bat'. The output continues with 'C:\Windows\System32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\SampleJob>echo "Hello World"' and 'C:\Windows\System32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\SampleJob>exit 0', ending with 'Finished: SUCCESS'.

実際に見てみましょう



Jenkins に触ってみる

1. ジョブの作成
2. GitHubからソースを取得する
3. Ant でビルドをする
4. JUnit でテストをする

ビルドツールの設定①



ビルドツールの設定②

Jenkinsの管理

System Configuration



システムの設定

システム全体の振る舞いやパスを設定します。



Global Tool Configuration

Configure tools, their locations and automatic installers.



プラグインの管理

Jenkinsの機能を拡張可能なプラグインの追加、削除、無効化および有効化を行います。



ノードの管理

Jenkinsがジョブを実行する様々なノードの追加、削除、制御および監視を行います。

ビルドツールの設定③

Antを入れます

Ant

インストール済みAnt

①

Ant追加

「Ant追加」を押す

Jenkinsで利用する、このシステムにインストールされたAntの一覧です。

Ant

インストール済みAnt

Ant追加

②

Ant

名前 DefaultAnt

☒ 自動インストール

Apacheからインストール

バージョン 1.10.8

③

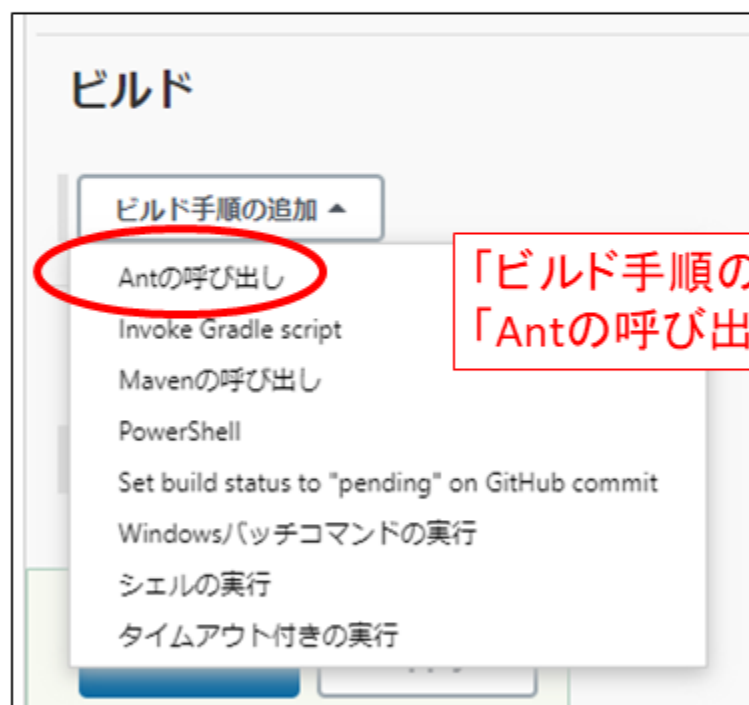
Save Apply

画面に表示される Ant の名前
→ジョブごとにバージョンを変えたい
場合は、異なる名前で追加する

インストーラの削除

ジョブの設定④

ビルドツールの設定



「ビルド手順の追加」から
「Antの呼び出し」を選択

ジョブの設定

ビルド

Antの呼び出し

① 使用するAnt DefaultAnt ビルドツールの名前を選択

② ターゲット all 「all」と入力

③ 高度な設定...

ビルド手順の追加 ▼

さらに設定することがあるため、
「高度な設定...」を押す

ジョブの設定

ワークスペースの下にある「build.xml」を含むフォルダのパスを入力する
例: (ワークスペース)¥Tax

ビルドファイル

[Redacted path]

プロパティ

[Redacted property]

Javaオプション

[Redacted Java options]

SampleJob

共有 表示

workspace > SampleJob

名前

.git

Tax



Tax

共有 表示

SampleJob > Tax

名前

ant

bin

libs

src

build.xml

ココのパスを入力する

ジョブの設定

最後に保存！

ビルドを中止する

ビルド

Antの呼び出し

使用するAnt: DefaultAnt

ターゲット: all

ビルドファイル: [REDACTED]

プロパティ: [REDACTED]

Javaオプション: [REDACTED]

ビルド手順の追加

ビルド後の処理

ビルド後の処理の追加

保存 Apply

ジョブ実行

Jenkins > SampleRepositoryJob

プロジェクト Sa

ダッシュボードへ戻る
状態
変更履歴
ワークスペース
ビルド実行
プロジェクトの削除
設定
Rename

「ビルドを実行」を押す
→ビルド履歴で成功が出ればOK！

変更履歴

永続リンク

ビルド履歴 推移

- 最新のビルド (#1), 1 分 22 秒
- 最新の安定ビルド (#1), 1 分 22 秒

ここまで終わったら
「手を挙げる」

ビルド実行で失敗する場合

BUILD FAILED

```
javac compiler;  
com.sun.tools.javac.Main is not on the classpath.  
Perhaps JAVA_HOME does not point to the JDK.  
It is currently set to "C:\Program Files\Java\jre1.8.0_261"
```

```
Total time: 0 seconds
Build step 'Antの呼び出し' marked build as failure
Finished: FAILURE
```

Google 翻訳: おそらく、JAVA HOMEはJDKを指していません。

正しくJDKを使えてない様子
→JDKを指定してあげましょう

JDK の設定①



JDK の設定②

Jenkinsの管理

System Configuration



システムの設定

システム全体の振る舞いやパスを設定します。



Global Tool Configuration

Configure tools, their locations and automatic installers.



プラグインの管理

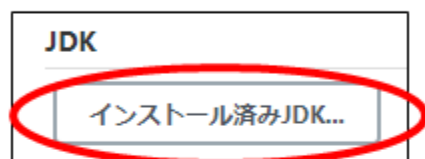
Jenkinsの機能を拡張可能なプラグインの追加、削除、無効化および有効化を行います。



ノードの管理

Jenkinsがジョブを実行する様々なノードの追加、削除、制御および監視を行います。

JDK の設定③



Jenkins に触ってみる

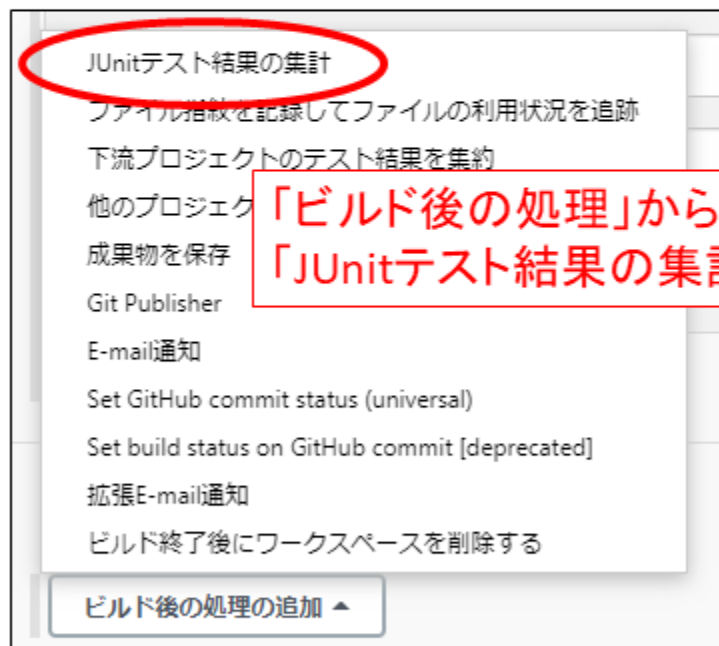
1. ジョブの作成
2. GitHubからソースを取得する
3. Ant でビルドをする
4. JUnit でテストをする

ジョブの設定①

①



②



「ビルド後の処理」から
「JUnitテスト結果の集計」を選択

ジョブの設定②

ビルド後の処理

JUnitテスト結果の集計

テスト結果XML

Antのファイルセットincludes属性の書式に従ってビルドから生成されるXMLレポートファイル群を指定します（例：myproject/target/test-reports/*.xml）。ワークスペースルートからの相対パスです。

☐ 長い標準/エラー出力をそのまま保持する

空模様の倍率

1%失敗すると、ビルド状態は99%になります。5%失敗すると、ビルド状態は95%になります。

Allow empty results ☐ Do not fail the build on empty test results

ビルド後の処理の追加 ▾

保存 Apply

最後に保存！

ジョブ実行

Jenkins > SampleRepositoryJob >

プロジェクト Sa

「ビルドを実行」を押す
→ビルド履歴で成功が出ればOK!

変更履歴

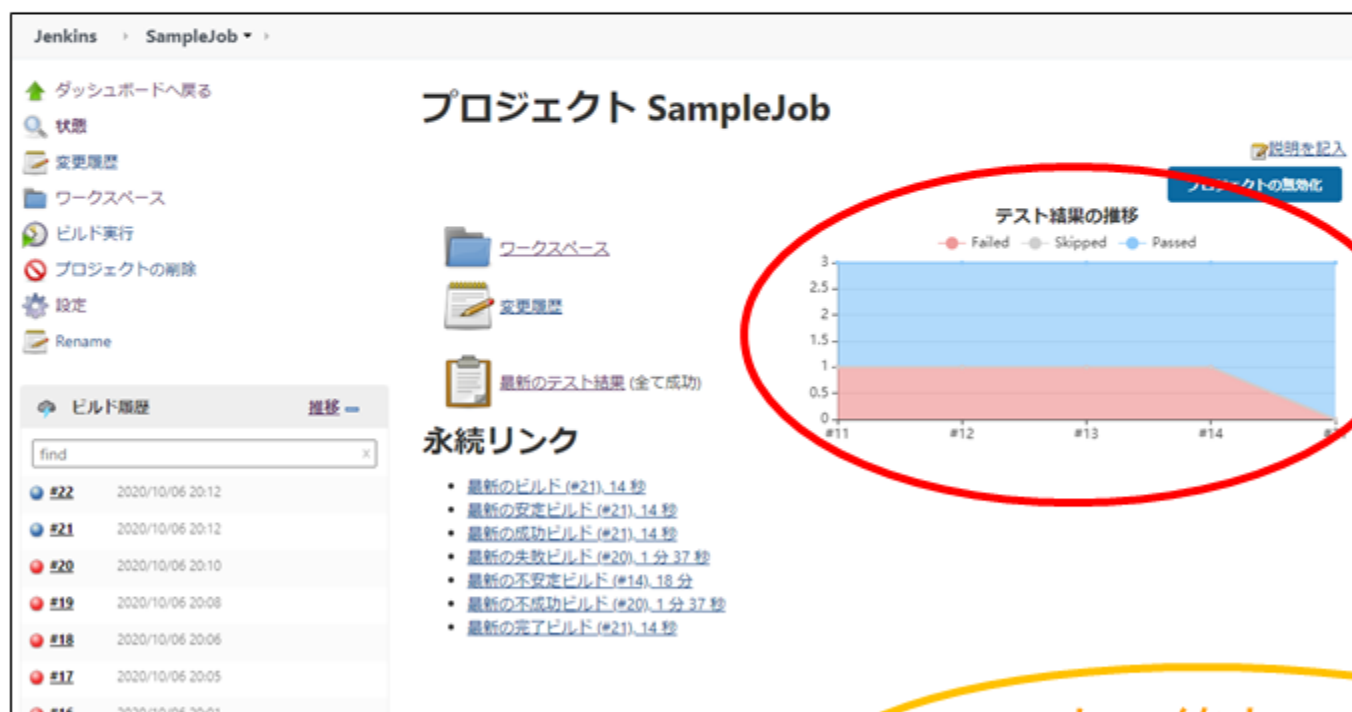
永続リンク

最新のビルド (#1) 1分 22 秒

最新の安定ビルド (#1) 1分 22 秒

ビルド結果結果表示

ビルド実行が1回だとわかりにくいので、もう一度実行



ここまで終わったら
「手を挙げる」

本日の予定

1. はじめに
2. 幹事会社紹介、スタッフ紹介
3. 自己紹介
4. この研究会について
5. 使用するツールについて
6. CIについて
7. Jenkinsに触れる
8. 次回以降について

次回以降について

| | 日程 | 内容 | 形式 |
|-----|-------|------------------------|--------------|
| 第1回 | 10/10 | CIについて講義／Jenkinsに触れる | オンライン (Zoom) |
| 第2回 | 11/14 | プラグイン／MagicPodの使用 | オンライン (Zoom) |
| 第3回 | 12/12 | MagicPodとの連携／グループ話し合い | オンライン (Zoom) |
| 第4回 | 1/16 | Jenkinsのその他設定／グループ話し合い | オンライン (Zoom) |
| 第5回 | 2/13 | 成果発表／懇親会 | 弊社会議室 |

- 時間: 10:00～12:00
- 詳細: 再度メールにてご連絡させていただきます。

皆様お疲れ様でした。

このミーティングは、13:00 まで開いております。
質問等がございましたら、何なりとお申し付けください。
