

# 神情協 先端技術研究会 「使えるJenkinsを構築する」 第4回

---

株式会社ジェイエスピー

# 本日の予定

---

1. はじめに
2. 前回の復習
3. Jenkins のその他設定
4. グループディスカッション
5. 次回について

以下のアカウントを  
確認してください！

- Jenkins
- Chatwork

# スタッフ紹介

---

- 講師 : 千田 駿人
- 質問等があった場合、  
基本的にZoomのチャットでコメントをお願いします。

# Zoom について

## 「バーチャル背景」



# Zoom について

## 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



# 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

# 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 <b>オンライン座談会</b>	<b>オンライン (Zoom)</b>

■ 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5回**までオンラインとする予定です。

# 前回について

---

<目標>

Jenkins と MagicPod の連携方法を知る

キーワード : MagicPod、Powershell



# 前回の内容

---

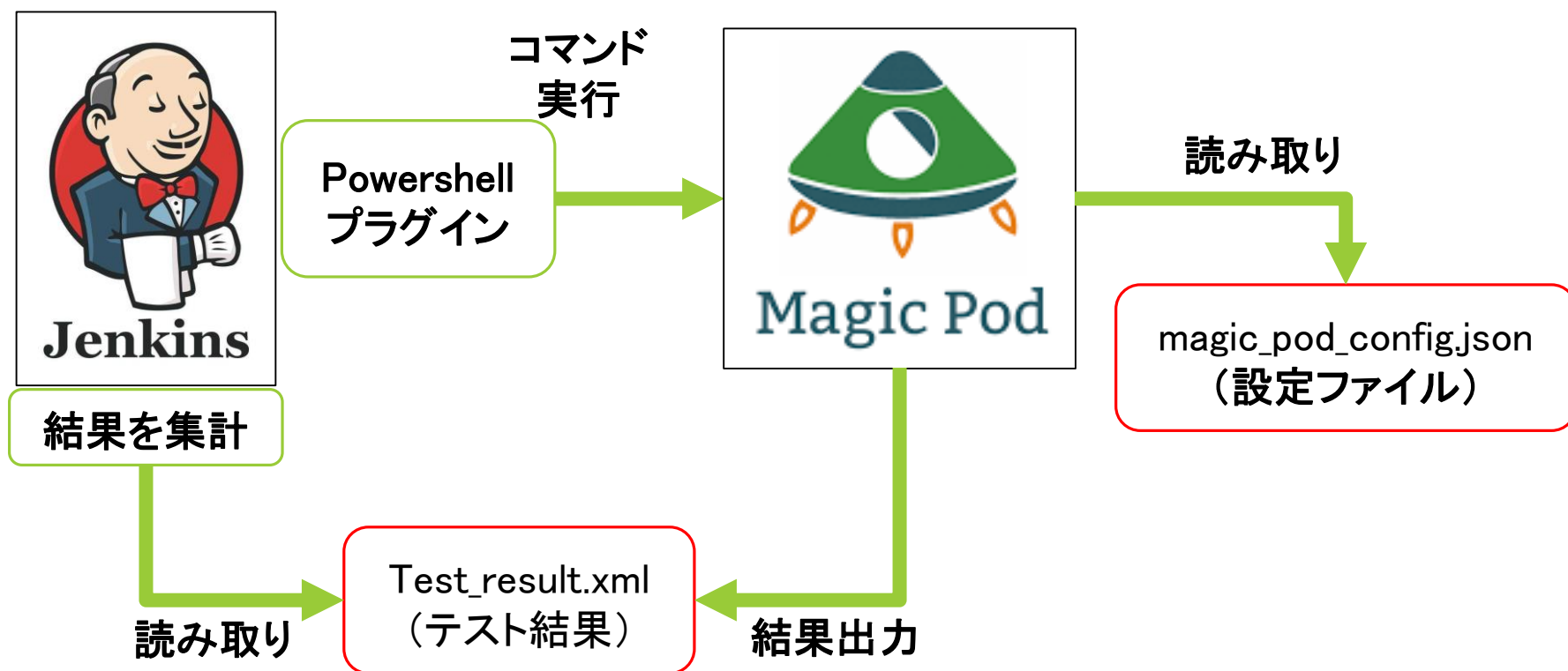
1. はじめに
2. 前回の復習
3. Jenkins と MagicPod の連携
4. グループディスカッション
5. 次回以降について

# Jenkins と MagicPod の連携

---

1. 設定ファイルの確認 (MagicPod)
2. 設定ファイルを編集 (MagicPod)
3. Powershellでコマンド実行 (MagicPod)
4. Powershellプラグインをインストール (Jenkins)
5. ジョブの設定と実行 (Jenkins、MagicPod)
6. テスト結果の表示 (Jenkins、MagicPod) 【課題】

# Jenkins と MagicPod の連携



# 本日の予定

---

1. はじめに
2. 前回の復習
3. Jenkins のその他設定
4. グループディスカッション
5. 次回について

# 今回について

---

<目標>

Jenkins と MagicPod の連携方法を知る

キーワード : MagicPod、Powershell

# Jenkins のその他設定

---

1. ジョブの変更差分
2. Excelの操作
3. 権限の設定【おまけ】

このマークのところまで来たら、  
「手を挙げる」を押してください」 →

ここまで終わったら  
「手を挙げる」

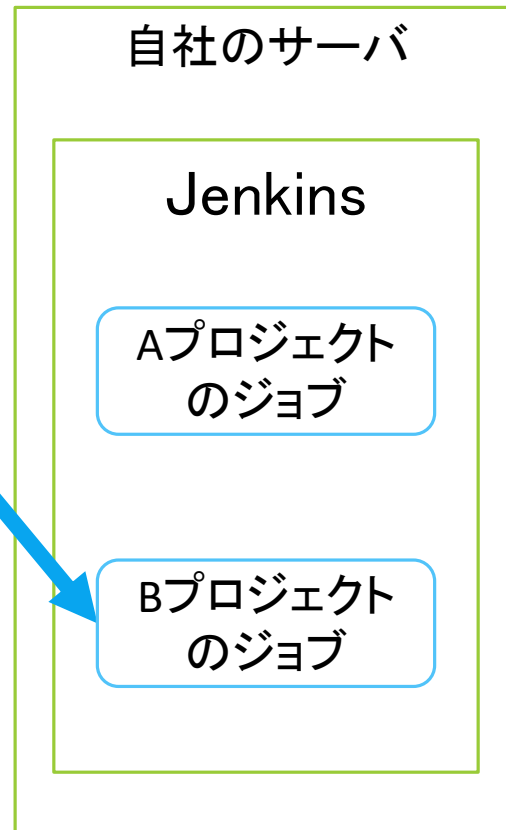
# ジョブの変更差分／権限の設定

## Case

他プロジェクトの人が  
誤ってジョブを修正



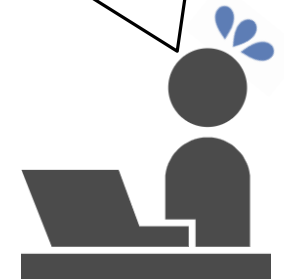
- ・バックアップ
- ・権限の設定 →【おまけ】  
があるとうれしい！



失敗が続くぞ！？



結果が分からないぞ！？



# ジョブの変更差分

- どこが変更されたが差分を見れる(XML形式)
- Gitと同様にコメントを残すことが可能

設定の差分

Hide Version Changes

### SampleDiff

Older Change	Newer Change
Date: 2021-01-13 19-10-44	Date: 2021-01-13 19-14-29
Operation: 変更	Operation: 変更
User: hayato_senda	User: hayato_senda
Change Reason: 新規作成	Change Reason: バッチを追加

Older Change	Newer Change
1 <?xml version="1.0" encoding="UTF-8"?><project>	1 <?xml version="1.0" encoding="UTF-8"?><project>
2 <blockBuildWhenDownstreamBuilding>false</blockBuildWhenDownstreamBuilding>	2 <actions/>
3 <blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>	3 <blockBuildWhenDownstreamBuilding>false</blockBuildWhenDownstreamBuilding>
4 <builders/>	4 <blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>
5 <buildWrappers/>	5 <builders>
6 <canRoam>true</canRoam>	6 <udson.tasks.BatchFile>
7 <concurrentBuild>false</concurrentBuild>	7 <command>echo "Hello World"</command>
	8 </udson.tasks.BatchFile>
	9 </builders>
	10 <buildWrappers/>
	11 <canRoam>true</canRoam>
	12 <concurrentBuild>false</concurrentBuild>

< Expand Diff      Shrink Diff >      < Prev      < Shrink Diff



# ジョブの変更差分

プラグインをインストール

オフラインの場合は  
22ページへ

The screenshot shows the Jenkins dashboard. The left sidebar contains the following items: 新規ジョブ作成 (New Job), 開発者 (Developer), ビルド履歴 (Build History), プロジェクト相関関係 (Project Relationships), フォール指紋チェック (Fingerprinting), **Jenkinsの管理** (Manage Jenkins), My Views, Lockable Resources, and New View. The 'Jenkinsの管理' item is circled in red. The main content area displays a table of jobs with columns: S, W, 名前 (Name), 最新の成功ビルド (Last Successful Build), 最新の失敗ビルド (Last Failed Build), ビルド所要時間 (Build Time), and # Issues. The table lists several jobs including GroupingJob, MagicPodTest, MagicPodTest\_2, MagicPodTestSub, SampleJob, SampleJob2, and SampleRepositoryJob. At the bottom, there are links for 'Atom feed 全ビルド', 'Atom feed 失敗ビルド', and 'Atom feed 最新ビルドのみ'.

S	W	名前 ↓	最新の成功ビルド	最新の失敗ビルド	ビルド所要時間	# Issues
		<a href="#">GroupingJob</a>	1 ヶ月 29 日 - <a href="#">#17</a>	2 ヶ月 0 日 - <a href="#">#14</a>	13 秒	-
		<a href="#">MagicPodTest</a>	1 ヶ月 1 日 - <a href="#">#7</a>	1 ヶ月 1 日 - <a href="#">#6</a>	22 秒	-
		<a href="#">MagicPodTest_2</a>	3 ヶ月 19 日 - <a href="#">#11</a>	3 ヶ月 19 日 - <a href="#">#9</a>	16 秒	-
		<a href="#">MagicPodTestSub</a>	3 ヶ月 29 日 - <a href="#">#4</a>	—	16 秒	-
		<a href="#">SampleJob</a>	3 ヶ月 7 日 - <a href="#">#23</a>	3 ヶ月 7 日 - <a href="#">#20</a>	8.2 秒	-
		<a href="#">SampleJob2</a>	1 ヶ月 29 日 - <a href="#">#15</a>	2 ヶ月 2 日 - <a href="#">#10</a>	12 秒	19
		<a href="#">SampleRepositoryJob</a>	3 ヶ月 11 日 - <a href="#">#1</a>	3 ヶ月 7 日 - <a href="#">#2</a>	14 秒	-

アイコン: [S](#) [M](#) [L](#)

凡例 Atom feed 全ビルド Atom feed 失敗ビルド Atom feed 最新ビルドのみ

# ジョブの変更差分

## プラグインをインストール

The screenshot shows the Jenkins web interface. At the top, there is a search bar and a 'ログアウト' (Logout) button. The left sidebar contains navigation links: '新規ジョブ作成' (Create New Job), '開発者' (Developer), 'ビルド履歴' (Build History), 'プロジェクト相関関係' (Project Relationships), 'ファイル指紋チェック' (File Fingerprint Check), 'Jenkinsの管理' (Jenkins Management), 'My Views', 'Lockable Resources', and 'New View'. The main content area is titled 'Jenkinsの管理' (Jenkins Management) and includes a notification about a new version (2.263.1) available for download. Below this, the 'System Configuration' section is visible, containing 'システムの設定' (System Settings), 'Global Tool Configuration', and 'プラグインの管理' (Manage Plugins). The 'プラグインの管理' option is highlighted with a red circle. It includes a green puzzle piece icon and text indicating that plugins can be added, removed, disabled, or enabled, and that there are updates available. The bottom of the interface shows a 'ビルドキュー' (Build Queue) section with a list of builds and a 'Security' section with links to 'Global Security Settings', 'Manage Credentials', and 'Configure Credentials'.

# ジョブの変更差分

## プラグインをインストール

①



「利用可能」をクリック

②



検索で「configuration history」を入力

# ジョブの変更差分

## プラグインをインストール

③



「Job Configuration History」  
にチェックを入れる

④

再起動せずにインストール

ダウンロードして再起動

「再起動せずにインストール」  
をクリック

# ジョブの変更差分

## プラグインをインストール

### プラグインのインストール/アップグレード

準備

- インターネットとの接続をチェックします。
- jenkins-ci.orgとの接続をチェックします。
- 成功

Job Configuration History

● 成功

Loading plugin extensions

● Success

成功すれば完了！

➡ [ページの先頭へ戻る](#)  
(すぐにインストールしたプラグインを使用できます)

➡ ☐ インストール完了後、ジョブがなければJenkinsを再起動する

# ジョブの変更差分

---

以下でプラグインをダウンロード。

- Job Configuration History

<http://updates.jenkins-ci.org/download/plugins/jobConfigHistory/>

プラグインのインストールは次ページ

# ジョブの変更差分

①

アップデート   利用可能   インストール済み   **高度な設定**

## HTTP Proxyの設定

サーバー

## プラグインのアップロード

.hpiファイルをアップロードして、プラグインリポジトリ以外からプラグインをインストールできます。

② ファイル: **ファイルを選択** 選択 **前ページダウンロードしたファイルを選択**

③ **アップロード**

# ジョブの変更差分

変更したことが分かるように新しくジョブを作成



The screenshot shows the Jenkins dashboard. The left sidebar contains a list of navigation items: '新規ジョブ作成' (Create New Job), '開発者' (Developer), 'ビルド履歴' (Build History), 'プロジェクト相関関係' (Project Relationships), 'ファイル指紋チェック' (File Fingerprint Check), 'Jenkinsの管理' (Manage Jenkins), 'My Views', 'Lockable Resources', '設定の変更履歴' (Configuration History), and 'New View'. The '新規ジョブ作成' button is circled in red, and a red arrow points to it from the text box above. The main content area displays a table of jobs with the following columns: 'S' (Status), 'W' (Icon), '名前 ↓' (Name), '最新の成功ビルド' (Latest Successful Build), and '最新の失敗ビルド' (Latest Failed Build). The table lists several jobs, including 'GroupingJob', 'MagicPodTest', 'MagicPodTest\_2', 'MagicPodTestSub', 'SampleJob', 'SampleJob2', and 'SampleRepositoryJob'. At the bottom, there are links for 'アイコン: SML' and '凡例 Atom feed 全ビルド'.

S	W	名前 ↓	最新の成功ビルド	最新の失敗ビルド
●	☁	<a href="#">GroupingJob</a>	2ヶ月前 - #17	2ヶ月前 - #17
●	☁	<a href="#">MagicPodTest</a>	1ヶ月前 - #7	1ヶ月前 - #7
●	☀	<a href="#">MagicPodTest_2</a>	3ヶ月前 - #11	3ヶ月前 - #11
●	☀	<a href="#">MagicPodTestSub</a>	4ヶ月前 - #4	—
●	☁	<a href="#">SampleJob</a>	3ヶ月前 - #23	3ヶ月前 - #23
●	☀	<a href="#">SampleJob2</a>	2ヶ月前 - #15	2ヶ月前 - #15
●	☁	<a href="#">SampleRepositoryJob</a>	3ヶ月前 - #1	3ヶ月前 - #1



# ジョブの変更差分

変更したことが分かるように新しくジョブを作成

①

Enter an item name

SampleDiff

» Required field

ジョブ名「SampleDiff」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は様々な作業の自動化に利用することができます。

「フリースタイル～」をクリック



パイプライン

複数のビルドステップ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.



フォルダ

階層的にアイテムを格納するコンテナを作成します。プロジェクトフォルダは個別に名前空間を作成しますので、異なるプロジェクトであるビューとは違

③

OK

「OK」をクリック

# ジョブの変更差分

## ジョブの設定

ここでは特に設定はしない

何かコメントを入れる  
例)「新規作成」

The screenshot shows the Jenkins Job Configuration page for a new job. The tabs at the top are: General, ソースコード管理, ビルド・トリガ, ビルド環境, ビルド, and ビルド後の処理. The General tab is selected. It contains a large text area for the job name and description, with a placeholder text "説明". Below this is a link "[HTMLをエスケープ] プレビュー". There are several checkboxes for build options: "Throttle builds", "ビルドのパラメータ化", "古いビルドの破棄", "ビルド無効化", and "ビルドを並行実行". A "高度な設定..." button is located to the right of these checkboxes. Below the checkboxes is the "ソースコード管理" section, which has a radio button selected for "なし". At the bottom are "保存" and "Apply" buttons. On the right side, there is a "JobConfigHistory change message:" section with a text area containing the text "新規作成". This section is circled in red.

# ジョブの変更差分

ジョブの設定(差をつけるため、再度設定を行う。)

Jenkins > SampleDiff

プロジェクト SampleDiff

ワークスペース

変更履歴

永続リンク

ビルド履歴 推移

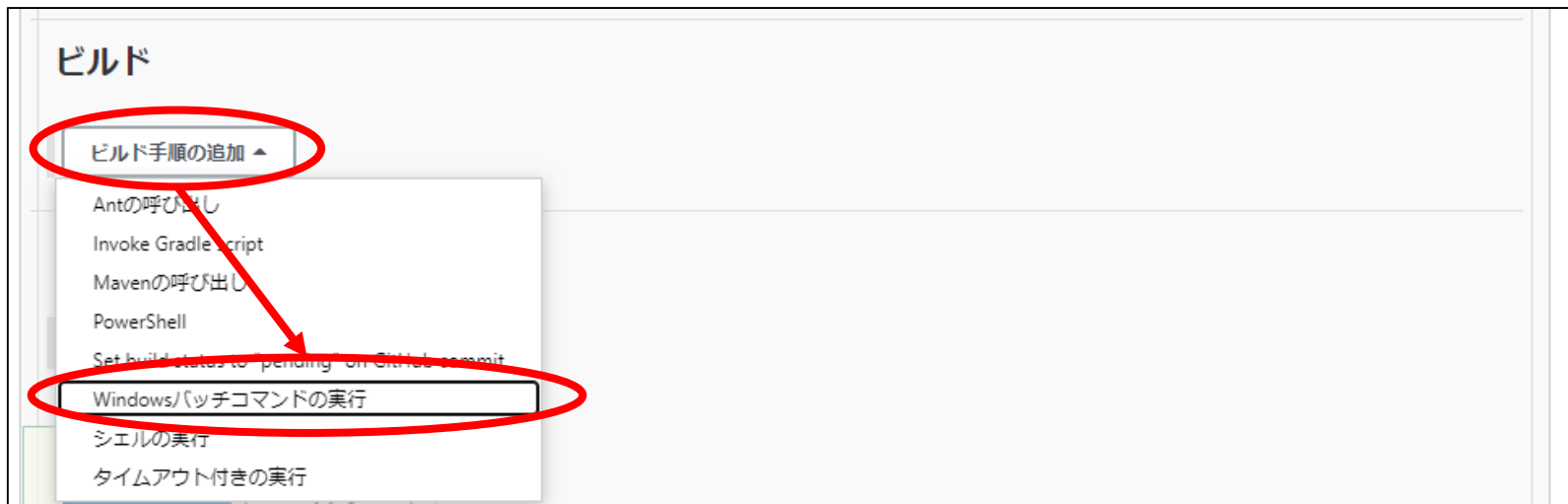
find x

Atom feed 全ビルド Atom feed 失敗ビルド

# ジョブの変更差分

## ジョブの設定

ビルドで「Windowsバッチコマンドの実行」を選択する



# ジョブの変更差分

## ジョブの設定

The screenshot shows the 'Build' configuration page with the following elements:

- ① 「echo "Hello World"」を入力**: A red box highlights the 'コマンド' (Command) input field containing the text 'echo "Hello World"'. Below it is a link 'ビルドから利用可能な環境変数の一覧' and a button '高度な設定...'. At the bottom left of the main panel is a button 'ビルド手順の追加'.
- ② 何かコメントを入れる例)「バッチを追加」**: A red box highlights the 'JobConfigHistory change message:' section on the right, which contains the text 'バッチを追加'.
- ③ 「保存」をクリックする**: A red box highlights the '保存' (Save) button at the bottom left of the page, next to an 'Apply' button.

# ジョブの変更差分

差分を見る

The screenshot shows the Jenkins web interface for the 'SampleDiff' project. The left sidebar contains a list of links: 'ダッシュボードへ戻る' (Return to Dashboard), '状態' (Status), '変更履歴' (Change History), 'ワークスペース' (Workspace), 'ビルド実行' (Build Execution), 'プロジェクトの削除' (Delete Project), '設定' (Settings), '設定の変更履歴' (Change History of Settings), and 'Rename'. A red box highlights the '差分を見る' (View Diff) link, and a red arrow points to it. The main content area displays 'プロジェクト SampleDiff' (Project SampleDiff) with links for 'ワークスペース' (Workspace) and '変更履歴' (Change History). Below this, there is a section for '永続リンク' (Permanent Link) and a 'ビルド履歴' (Build History) section with a search bar and Atom feed links for '全ビルド' (All Builds) and '失敗ビルド' (Failed Builds).

# ジョブの変更差分

設定の変更履歴							
↓上が最近更新したジョブ				SampleDiff			
日時 ↑	操作	ユーザ	ファイルを表示	Restore old config	ファイル A	① ↓更新後のファイル	
2021-01-13_19-14-29	変更 ⓘ	<a href="#">hayato_senda</a>	<input type="checkbox"/> XMLで表示 (プレインテキストで表示)		<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>
2021-01-13_19-10-44	変更 ⓘ	<a href="#">hayato_senda</a>	<input type="checkbox"/> XMLで表示 (プレインテキストで表示)	⏮	<input checked="" type="radio"/>	<input type="radio"/>	<input type="checkbox"/>
2021-01-13_19-09-55	新規作成	<a href="#">hayato_senda</a>	<input type="checkbox"/> XMLで表示 (プレインテキストで表示)	⏮	② ↑更新前のファイル		
2021-01-13_19-09-54	変更 ⓘ	<a href="#">hayato_senda</a>	<input type="checkbox"/> XMLで表示 (プレインテキストで表示)	⏮	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>
Entries per page: 10 25 100 250 all							③ 差分を表示

# ジョブの変更差分

設定の差分

Hide Version Changes

更新前 SampleDiff 更新後

Older Change Restore this configuration

Date: 2021-01-13 19-10-44  
Operation: 変更  
User: hayato\_senda  
Change Reason: 新規作成

Newer Change Restore this configuration

Date: 2021-01-13 19-14-29  
Operation: 変更  
User: hayato\_senda  
Change Reason: パッチを追加

1	<?xml version="1.0" encoding="UTF-8"?><project>	1	<?xml version="1.0" encoding="UTF-8"?><project>
2	<blockBuildWhenDownstreamBuilding>false</blockBuildWhenDownstreamBuilding>	2	<actions/>
3	<blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>	3	<blockBuildWhenDownstreamBuilding>
4	<builders/>	4	<blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>
5	</build>	5	<builders>
		6	<hudson.tasks.BatchFile>
		7	<command>echo "Hello World"</command>
		8	</hudson.tasks.BatchFile>
		9	</builders>
		10	</build>

↑「echo "Hello World"」を追加したことが分かる

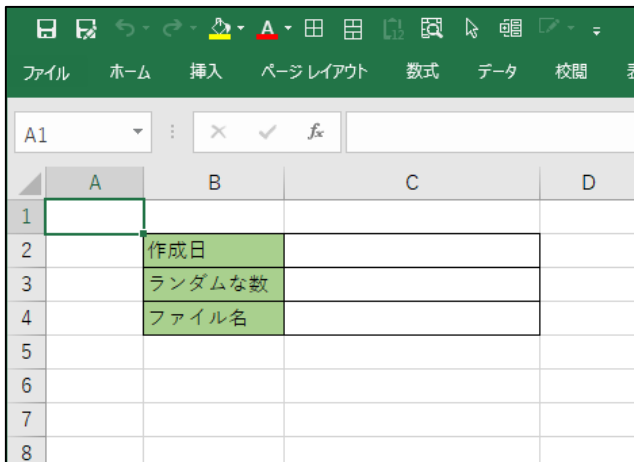
< Expand Diff Shrink Diff > Prev

ここまで終わったら  
「手を挙げる」



# Excel操作

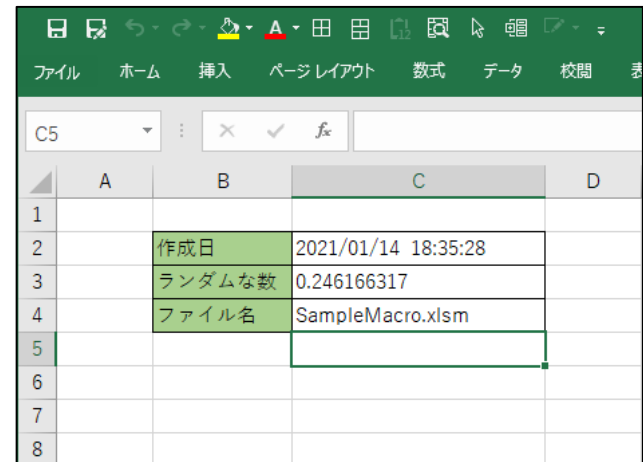
- Jenkinsを使って、定期的に Excel を操作できる。
- プラグインは「ない」。
- バッチからマクロを実行する。
- Excel がインストールされていることが前提。



Excel screenshot showing a template with headers for date, random number, and filename.

	A	B	C	D
1				
2		作成日		
3		ランダムな数		
4		ファイル名		
5				
6				
7				
8				

実行



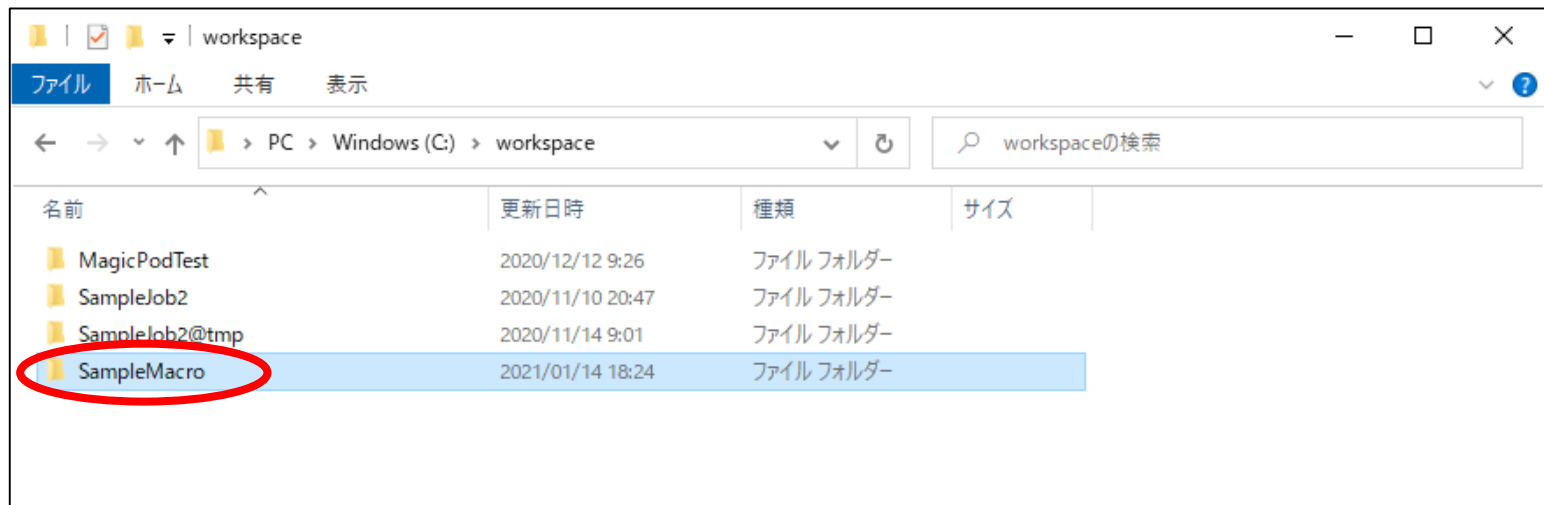
Excel screenshot showing the template with data filled in.

	A	B	C	D
1				
2		作成日	2021/01/14 18:35:28	
3		ランダムな数	0.246166317	
4		ファイル名	SampleMacro.xlsm	
5				
6				
7				
8				

# Excel操作

Jenkinsの設定をする前に準備  
フォルダを用意する

- C:¥workspace に新しく「SampleMacro」というフォルダを作成する

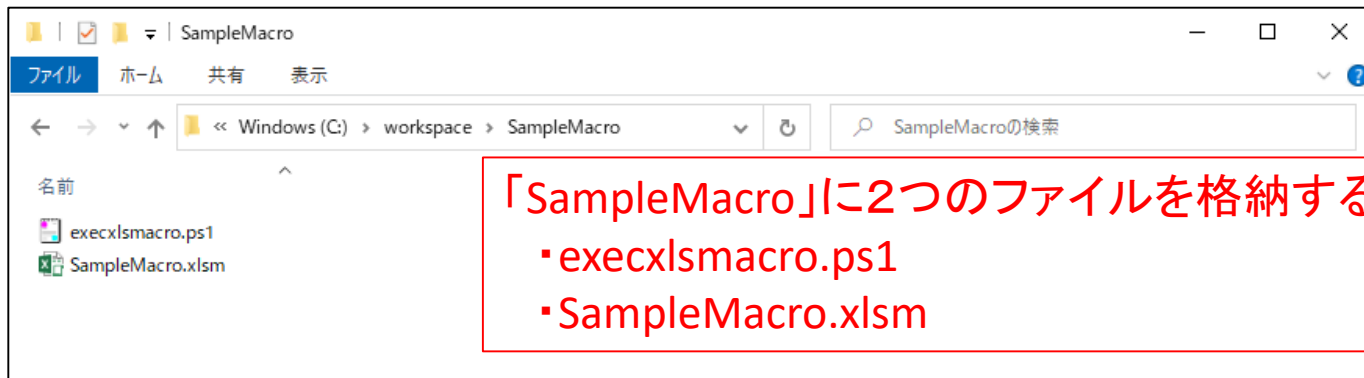


# Excel操作

Jenkinsの設定をする前に準備  
ファイルを取得する

- 以下URLから取得する(方法は次ページから)
- Gitに慣れている人はでクローンまたはプルをする

URL: <https://github.com/jspnet/jenkins-study-group>



# Excel操作

Jenkinsの設定をする前に準備



# Excel操作

Jenkinsの設定をする前に準備  
ファイルを取得する

「SampleMacro.xlsm」をクリックする

SampleMacro.xlsm

マクロと実行ファイルを追加

execxlsmacro.ps1

マクロと実行ファイルを追加

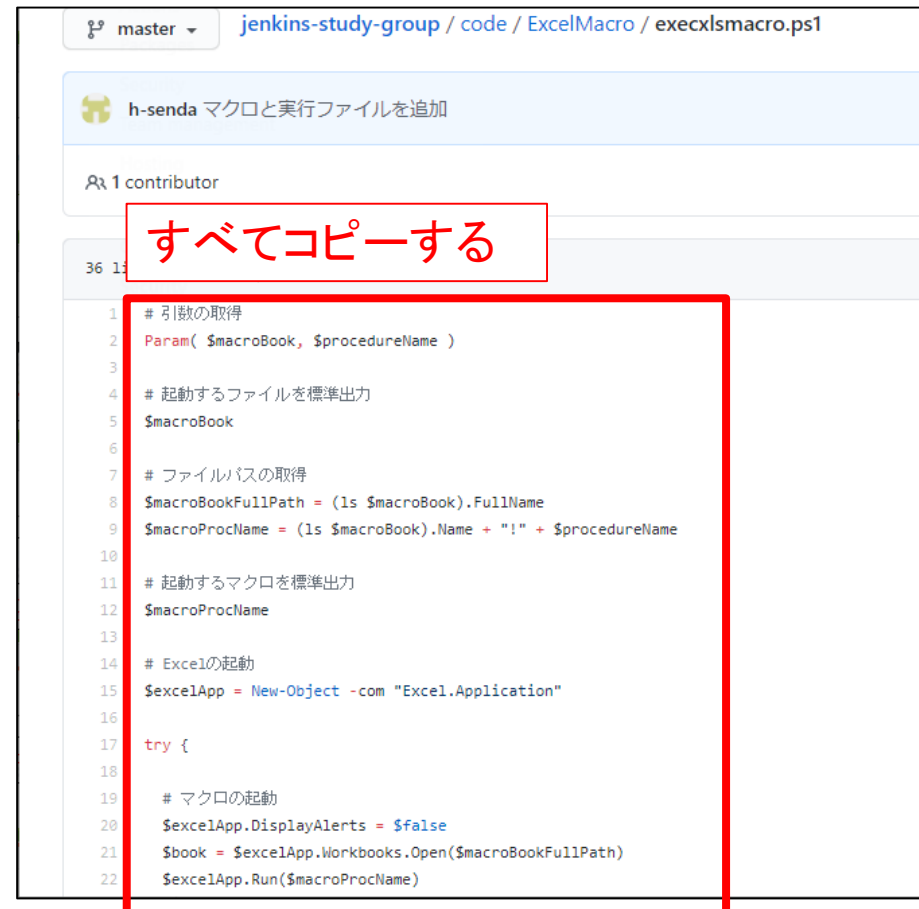
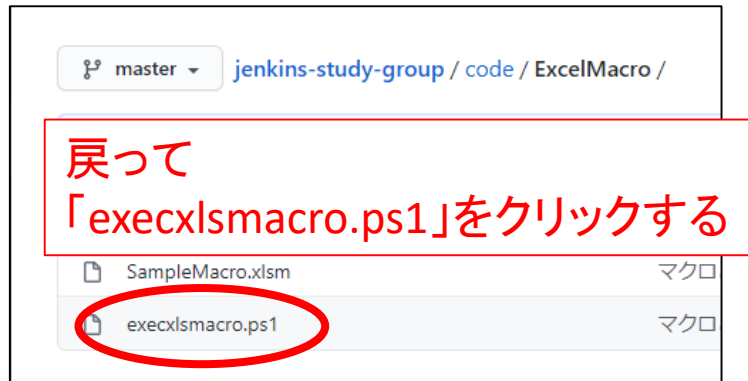
「Download」をクリックする

「C:¥workspace¥SampleMacro」に保存する

Download

# Excel操作

Jenkinsの設定をする前に準備

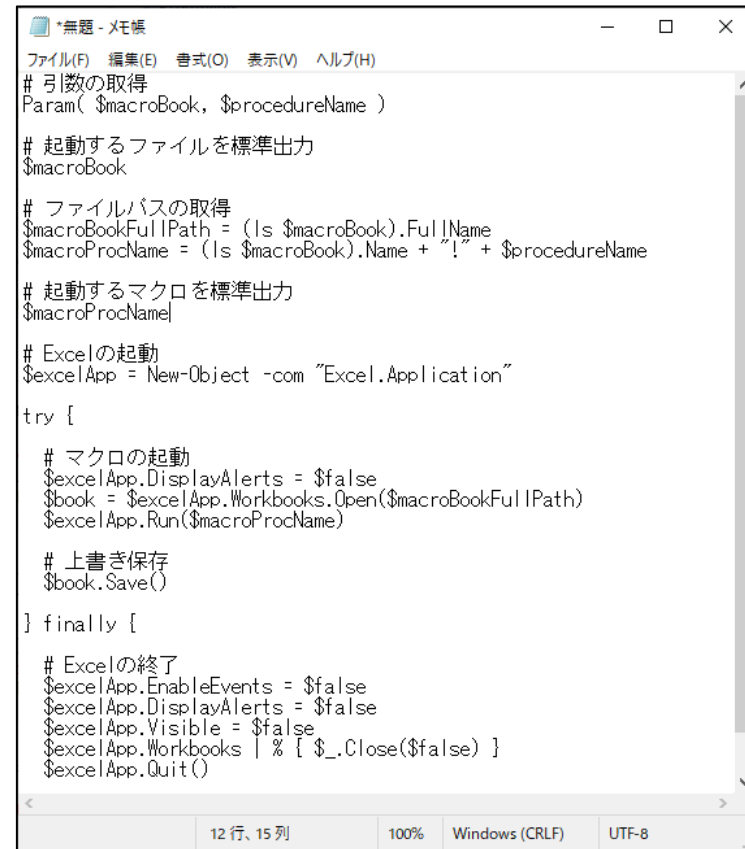


# Excel操作

Jenkinsの設定をする前に準備

メモ帳を開いて、貼り付ける

「C:¥workspace¥SampleMacro」に  
「execxlsmacro.ps1」  
という名前で保存する



```
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
# 引数の取得
Param( $macroBook, $procedureName )

# 起動するファイルを標準出力
$macroBook

# ファイルパスの取得
$macroBookFullPath = (ls $macroBook).FullName
$macroProcName = (ls $macroBook).Name + ".!" + $procedureName

# 起動するマクロを標準出力
$macroProcName

# Excelの起動
$excelApp = New-Object -com "Excel.Application"

try {
    # マクロの起動
    $excelApp.DisplayAlerts = $false
    $book = $excelApp.Workbooks.Open($macroBookFullPath)
    $excelApp.Run($macroProcName)

    # 上書き保存
    $book.Save()
} finally {
    # Excelの終了
    $excelApp.EnableEvents = $false
    $excelApp.DisplayAlerts = $false
    $excelApp.Visible = $false
    $excelApp.Workbooks | % { $_.Close($false) }
    $excelApp.Quit()
}
```

12 行、15 列 100% Windows (CRLF) UTF-8

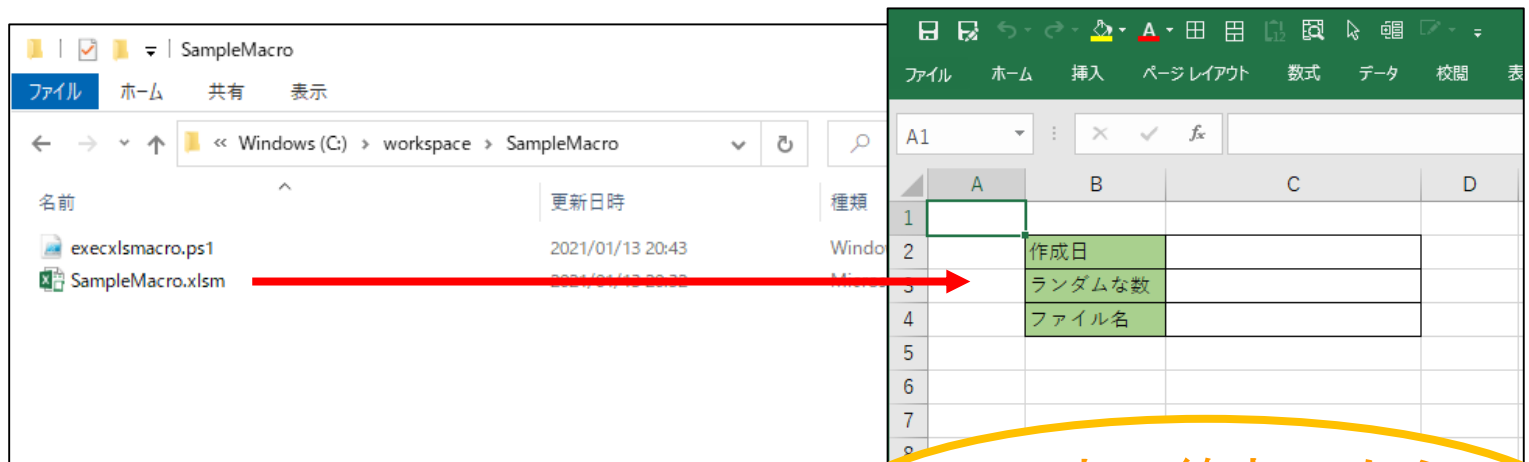
# Excel操作

Jenkinsの設定をする前に準備

「SampleMacro」に2つのファイルはありますか？

- execxlsmacro.ps1
- SampleMacro.xlsm

→一度開いて右図のように入力されてないことを確認

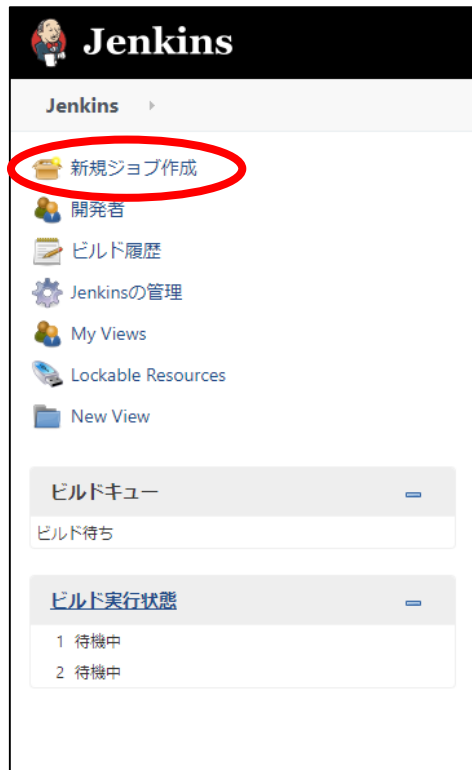


ここまで終わったら  
「手を挙げる」



# Excel操作

ここからジョブの作成  
以前作ったジョブをコピーせずに、新規に作成する



ジョブ名 「SampleMacro」

フリースタイル～ を選択

# Excel操作

## ワークスペースを変える

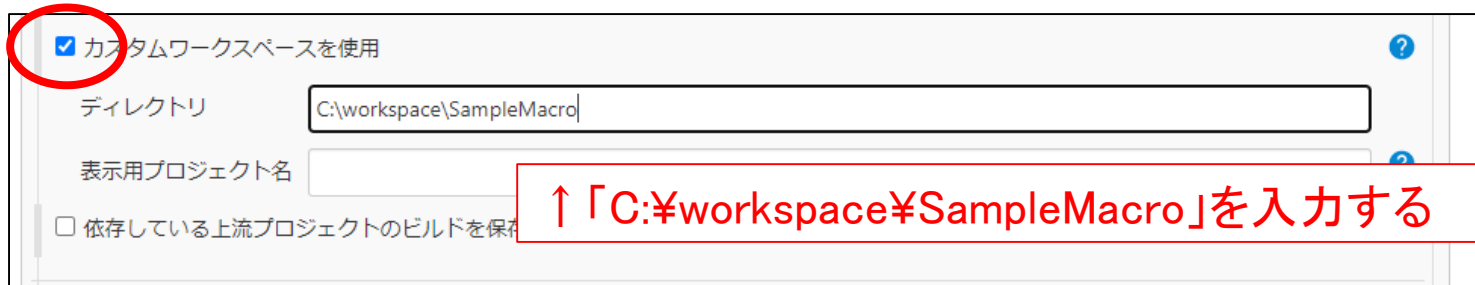


隠れている項目を表示するために、  
「高度な設定」を押す ↓

# Excel操作

ワークスペースを変える

↓「カスタムワークスペースを使用」にチェックを入れる



☒ カスタムワークスペースを使用

ディレクトリ

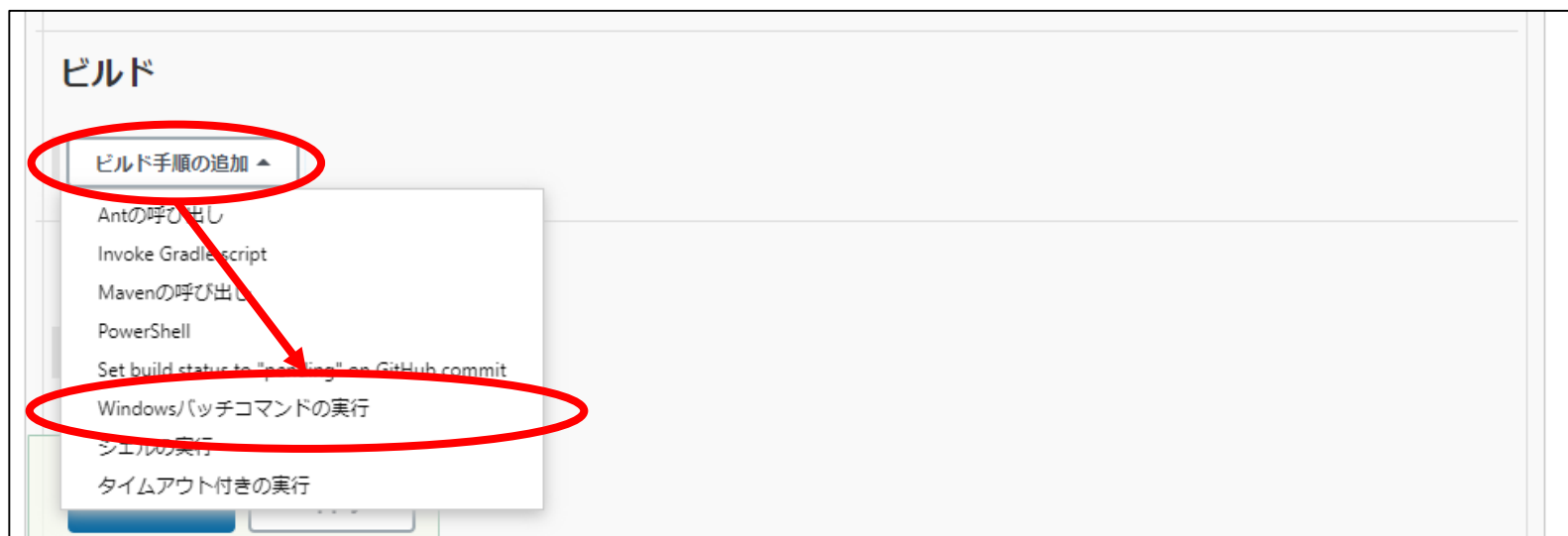
表示用プロジェクト名

☐ 依存している上流プロジェクトのビルドを保持

↑「C:¥workspace¥SampleMacro」を入力する

# Excel操作

バッチコマンドを入力する



# Excel操作

バッチコマンドを入力する

↓コピーして「コマンド」に貼り付ける

```
mkdir C:\Windows\system32\config\systemprofile\Desktop  
powershell -NoProfile -ExecutionPolicy Unrestricted .\execxlsmacro.ps1 $env:xls_macro_file $env:xls_macro_function
```



## ※解説

mkdir ~ : ファルダを作るコマンド。「Desktop」のフォルダを作成しなければ、エラーが出るため、必ず作成する。

Powershell : Powershellをコマンドプロンプトで呼び出すコマンド。

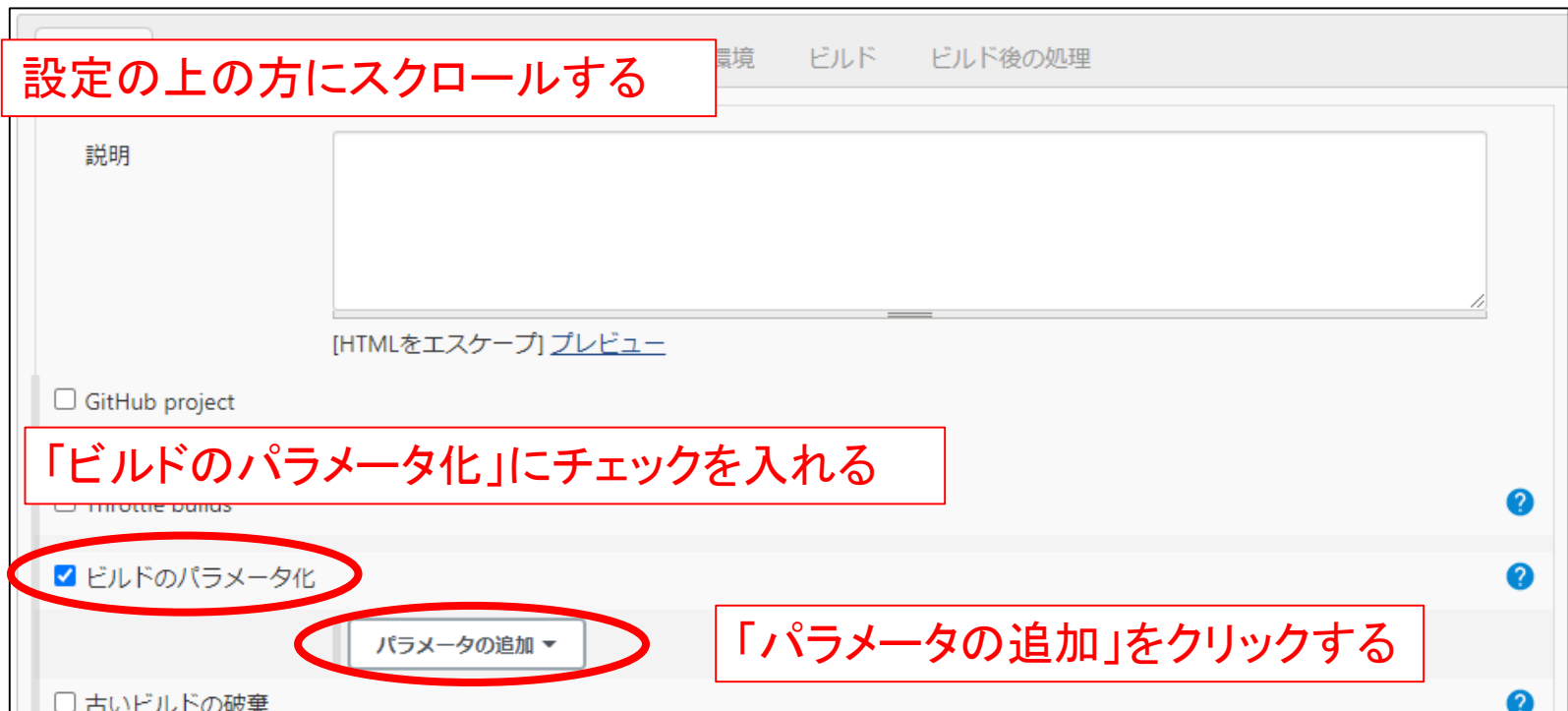
-NoProfile -ExecutionPolicy Unrestricted : 一時的にスクリプトが実行の設定を変えるオプション。

( Powershell ではデフォルトでスクリプトが実行できないため)

# Excel操作

パラメータを作る  
ファイル名とマクロ関数名をパラメータにする(環境変数として登録できる)

設定の上の方にスクロールする



# Excel操作

パラメータを作る(表の値をそれぞれ設定に入力する)

名前	xls_macro_file
デフォルト値	SampleMacro.xlsm
説明	エクセルのファイル名

文字列

名前 xls\_macro\_file

デフォルト値 SampleMacro.xlsm

説明 エクセルのファイル名

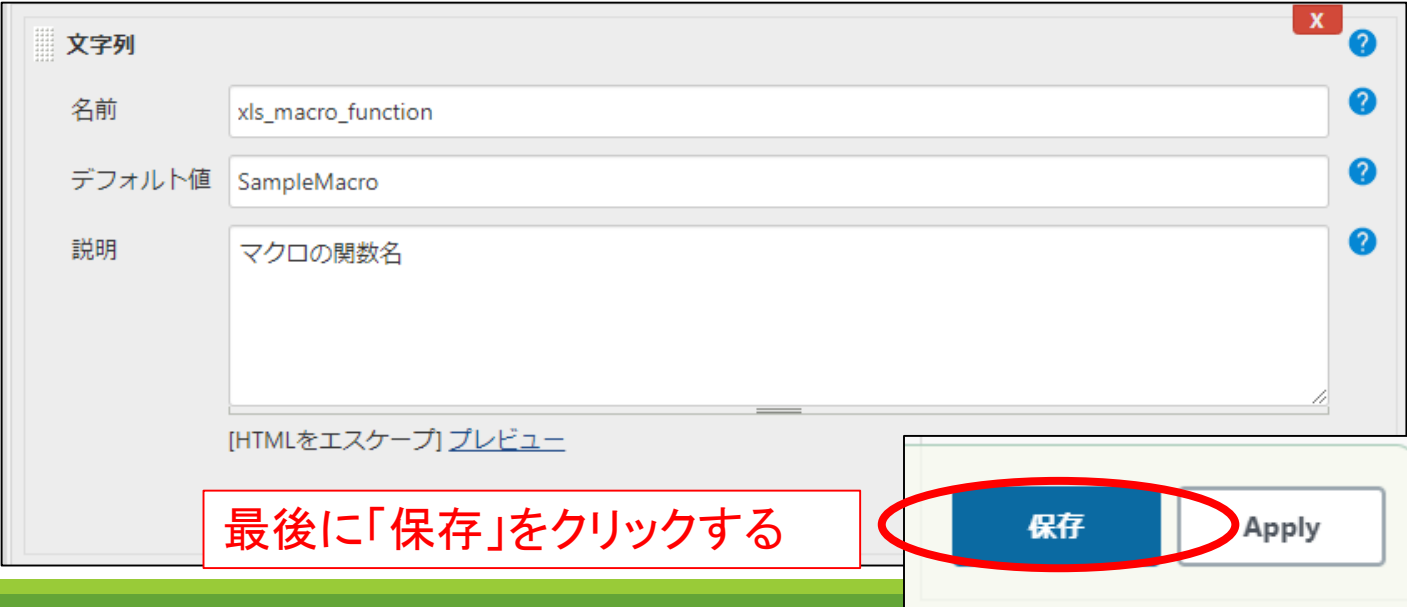
[HTMLをエスケープ] [プレビュー](#)

☐ Trim the string

# Excel操作

パラメータを作る(表の値をそれぞれ設定に入力する)

名前	xls_macro_function
デフォルト値	SampleMacro
説明	マクロの関数名



The screenshot shows the 'Text' dialog box in Excel. The '名前' (Name) field contains 'xls\_macro\_function', the 'デフォルト値' (Default value) field contains 'SampleMacro', and the '説明' (Description) field contains 'マクロの関数名'. At the bottom, there is a blue button labeled '保存' (Save) and a grey button labeled 'Apply'. The '保存' button is circled in red. Below the dialog box, there is a red box with the text '最後に「保存」をクリックする' (Click 'Save' at the end).

文字列

名前 xls\_macro\_function

デフォルト値 SampleMacro

説明 マクロの関数名

[HTMLをエスケープ] [プレビュー](#)

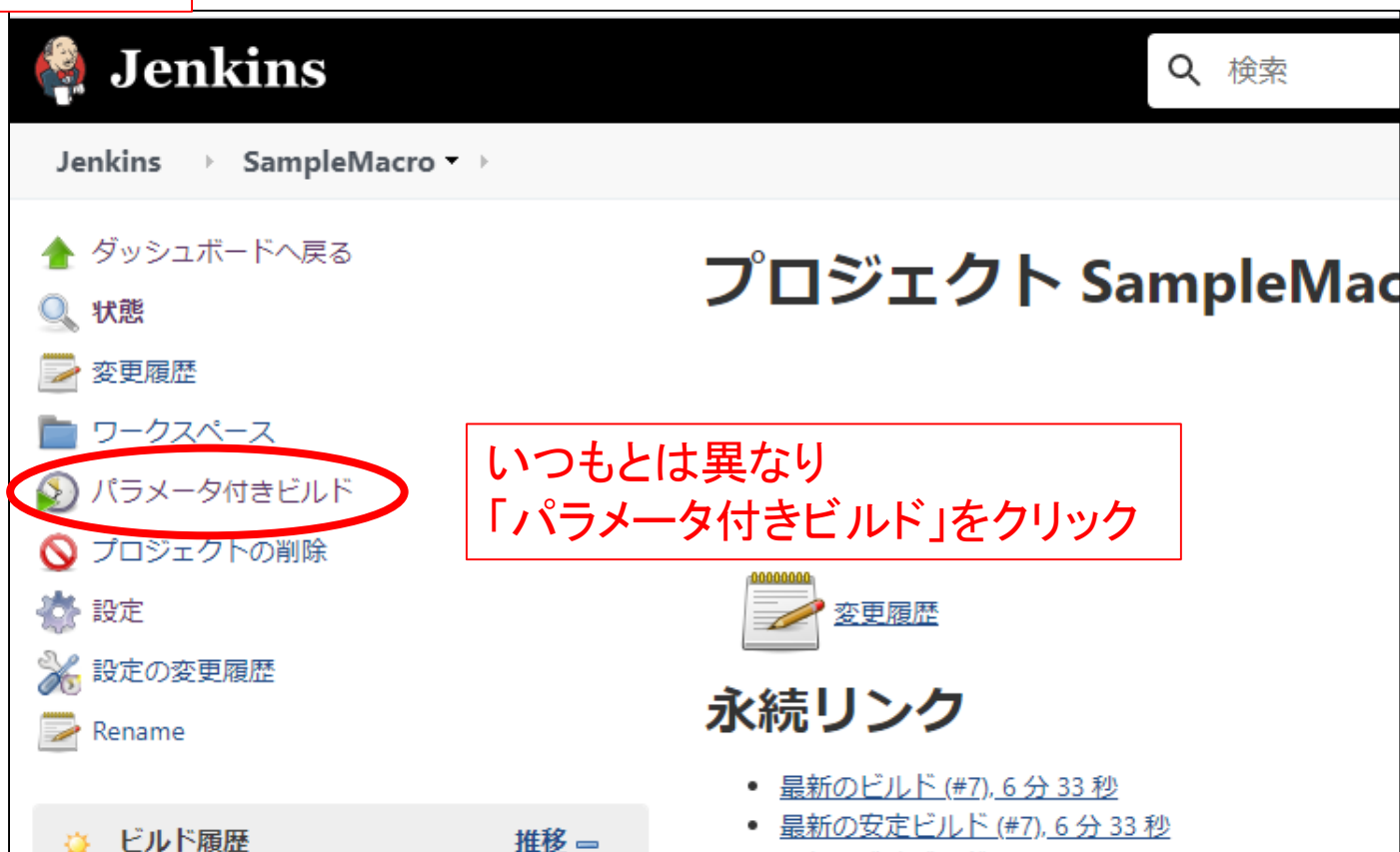
最後に「保存」をクリックする

保存 Apply



# Excel操作

## ビルド実行



Jenkins

検索

Jenkins ▸ SampleMacro ▾ ▸

ダッシュボードへ戻る

状態

変更履歴

ワークスペース

**パラメータ付きビルド**

プロジェクトの削除

設定

設定の変更履歴

Rename

ビルド履歴

プロジェクト SampleMacro

変更履歴

永続リンク

- 最新のビルド (#7), 6 分 33 秒
- 最新の安定ビルド (#7), 6 分 33 秒

いつもとは異なり  
「パラメータ付きビルド」をクリック

# Excel操作

## ビルド実行

何も変更せずに「ビルド」をクリックする  
(ファイル名やマクロ関数名が異なる場合はここを変更する)

### プロジェクト SampleMacro

このビルドはパラメータが必要です。

xls\_macro\_file SampleMacro.xlsm

エクセルのファイル名

xls\_macro\_function SampleMacro

マクロの関数名

ビルド

# Excel操作

## ビルド実行

実行が終わったら  
結果を見る



「コンソール出力」をクリック

# Excel操作

ビルド実行  
以下のように表示されればOK

## コンソール出力

ユーザー [hayato\\_senda](#) が実行

Running as SYSTEM

ビルドします。 ワークスペース: C:\workspace\SampleMacro

[SampleMacro] \$ cmd /c call C:\Windows\TEMP\jenkins4242006290870778332.bat

C:\workspace\SampleMacro>mkdir C:\Windows\system32\config\systemprofile\Desktop

サブディレクトリまたはファイル C:\Windows\system32\config\systemprofile\Desktop は既に存在します。

C:\workspace\SampleMacro>powershell -NoProfile -ExecutionPolicy Unrestricted .\execxlsmacro.ps1 \$env:xls\_macro\_file

\$env:xls\_macro\_function

SampleMacro.xlsm

SampleMacro.xlsm!SampleMacro

C:\workspace\SampleMacro>exit 0

Finished: SUCCESS

# Excel操作

結果を確認

File Explorer window showing the contents of the 'SampleMacro' folder. The files listed are 'execxlsmacro.ps1' and 'SampleMacro.xlsm'. A red arrow points from 'SampleMacro.xlsm' to the Excel spreadsheet below.

Excel spreadsheet showing the results of the build process. The data is as follows:

	A	B	C
1			
2		作成日	2021/01/13 21:44:09
3		ランダムな数	0.651056911
4		ファイル名	SampleMacro.xlsm
5			
6			
7			
8			
9			
10			

ビルドを実行した時間近くになっ  
ていれば成功！  
↓

ここまで終わったら  
「手を挙げる」

# Excel操作

## 補足

### ■ 定期実行するなら...

[ジョブの設定] → [ビルド・トリガ] → [定期的に実行]  
にチェックと cron の設定を入力

### ■ パラメータはどのように使われているか...

パラメータ	env:xls_macro_file (SampleMacro.xlsm)	xls_macro_function (SampleMacro)	環境変数
コマンド	\$env:xls_macro_file	\$env:xls_macro_function	
execxlsmacro.ps1	Param( \$macroBook, \$procedureName )		引数

# 本日の予定

---

1. はじめに
2. 前回の復習
3. Jenkins のその他設定
4. グループディスカッション
5. 次回について

# 成果発表について

---

## 課題

### Jenkinsを使って困ったことを解決する方法を提案

- グループで資料やデモを作成
- 2月13日の第5回で発表(1グループ15分発表、質疑応答5分)
- 必ず Jenkins を使うこと(GitHubやMagicPodを使用してもよい)
- ハードやツールの構成も発表に入れること
- 複数作成してもOK
- 途中でもOK、「本当はこうしたい」「ここが課題」など



# グループディスカッション

---

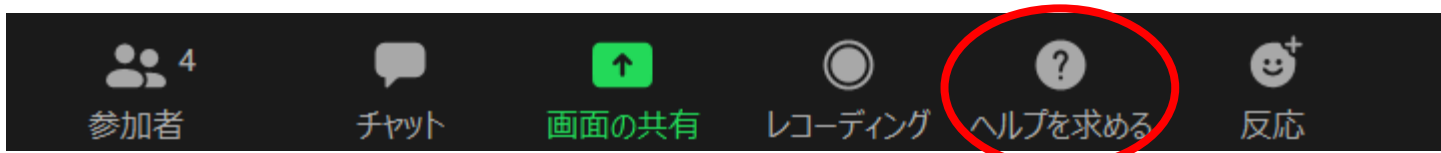
## ■ 内容

1. 何を発表するか
2. 成果発表までのTODOと担当

## ■ Chatworkのグループチャットに書く

## ■ タスク機能を使ってもOK！

## ■ ディスカッション中、聞きたいことがあればヘルプで千田を呼ぶ



# 次回について

	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／オンライン座談会	オンライン (Zoom)

## ■ 時間:

- 10:00～ 開始／発表について説明
- 10:10～ グループごとに発表 (1グループあたり15分発表／5分質疑応答)
- 11:10～ オンライン座談会 (詳細は検討中)
- 12:00～ 終了

## ■ 課題: 成果発表の資料作成

# 皆様お疲れ様でした。

次回の成果発表を楽しみにしています！

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。

---

# おまけ

---

# 権限の設定

プラグインをインストール

オフラインの場合は  
66ページへ

The screenshot shows the Jenkins dashboard. The left sidebar contains the following links: 新規ジョブ作成, 開発者, ビルド履歴, プロジェクト相関関係, フォールトトレランスチェック, **Jenkinsの管理** (circled in red), My Views, Lockable Resources, and New View. Below these are sections for 'ビルドキュー' (Build Queue) and 'ビルド実行状態' (Build Execution Status).

The main content area displays a table of jobs. The table has columns: S, W, 名前 ↓, 最新の成功ビルド, 最新の失敗ビルド, ビルド所要時間, and # Issues. The jobs listed are: GroupingJob, MagicPodTest, MagicPodTest\_2, MagicPodTestSub, SampleJob, SampleJob2, and SampleRepositoryJob.

At the bottom of the table, there is a legend for icons: 凡例, Atom feed 全ビルド, Atom feed 失敗ビルド, and Atom feed 最新ビルドのみ.

# 権限の設定

## プラグインをインストール

The screenshot shows the Jenkins web interface. At the top, there's a search bar and a 'ログアウト' (Logout) button. The left sidebar contains navigation links: '新規ジョブ作成' (Create new job), '開発者' (Developer), 'ビルド履歴' (Build history), 'プロジェクト相関関係' (Project dependencies), 'ファイル指紋チェック' (File fingerprint check), 'Jenkinsの管理' (Manage Jenkins), 'My Views', 'Lockable Resources', and 'New View'. The main content area is titled 'Jenkinsの管理' (Manage Jenkins) and includes a notification about a new version (2.263.1) available for download. Below this, the 'System Configuration' section is visible, containing links for 'システムの設定' (Configure System), 'Global Tool Configuration', and 'プラグインの管理' (Manage Plugins). The 'プラグインの管理' link is circled in red. It includes a description: '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。' (Add, delete, disable, and enable plugins to extend Jenkins functionality.) and a red bell icon indicating 'アップデートあり' (Updates available). At the bottom, the 'Security' section is partially visible, with links for 'グローバルセキュリティの設定' (Configure Global Security), 'Manage Credentials', and 'Configure Credential'.

Jenkins

検索

ログアウト

Jenkins

新規ジョブ作成

開発者

ビルド履歴

プロジェクト相関関係

ファイル指紋チェック

Jenkinsの管理

My Views

Lockable Resources

New View

ビルドキュー

ビルド待ち

ビルド実行状態

1 待機中

2 待機中

### Jenkinsの管理

Jenkinsの新しいバージョン(2.263.1)をダウンロードできます (変更履歴).

自動でアップグレード

### 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。  
🔔 アップデートあり

### 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。

### Security

グローバルセキュリティの設定

Manage Credentials

Configure Credential

# 権限の設定

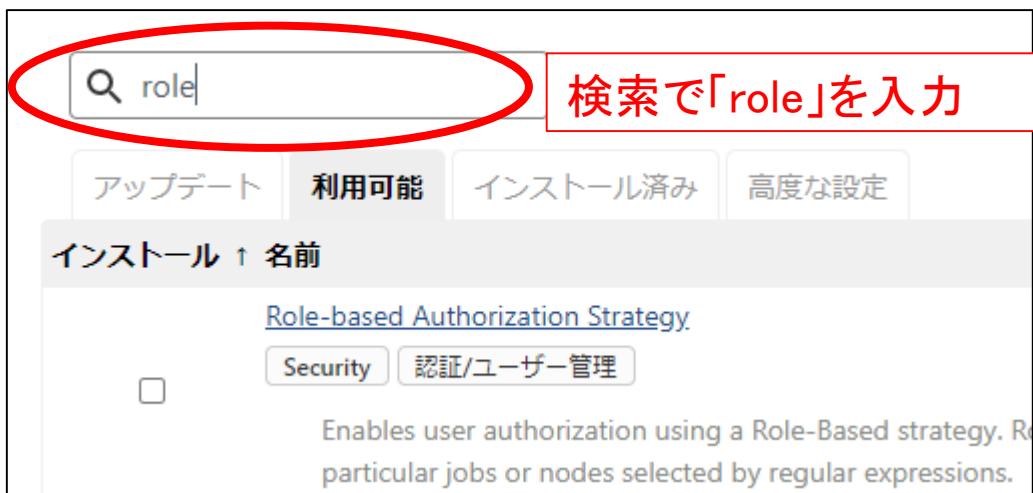
## プラグインをインストール

①



「利用可能」をクリック

②



検索で「role」を入力

# 権限の設定

## プラグインをインストール

③



「Role-based Authorization Strategy」  
にチェックを入れる

④

再起動せずにインストール

「再起動せずにインストール」をクリック



# 権限の設定

## プラグインをインストール

The screenshot shows the Jenkins Update Center interface. The main heading is 'プラグインのインストール/アップグレード' (Plugin Installation/Upgrade). A red box highlights the text '成功すれば完了！' (Success means completion!). Below this, a red oval encircles the status of two plugins: 'Role-based Authorization Strategy' and 'Loading plugin extensions', both showing a blue circle icon and the text '成功' (Success) and 'Success' respectively. To the right, a list of steps is shown: 'インターネットとの接続をチェックします。' (Check connection to the Internet), 'jenkins-ci.orgとの接続をチェックします。' (Check connection to jenkins-ci.org), and '成功' (Success). At the bottom, there are two green arrows with text: 'ページの先頭へ戻る (すぐにインストールしたプラグインを使用できます)' (Return to the top of the page (you can use the plugins you just installed immediately)) and 'インストール完了後、ジョブがなければJenkinsを再起動する' (After installation is complete, restart Jenkins if there are no jobs).

成功すれば完了！

プラグインのインストール/アップグレード

- インターネットとの接続をチェックします。
- jenkins-ci.orgとの接続をチェックします。
- 成功

Role-based Authorization Strategy 成功

Loading plugin extensions Success

ページの先頭へ戻る  
(すぐにインストールしたプラグインを使用できます)

インストール完了後、ジョブがなければJenkinsを再起動する

# 権限の設定

---

以下でプラグインをダウンロード。

- Role-based Authorization Strategy

<http://updates.jenkins-ci.org/download/plugins/role-strategy/>

プラグインのインストールは次ページ

# 権限の設定

① アップデート 利用可能 インストール済み **高度な設定**

## HTTP Proxyの設定

サーバー

## プラグインのアップロード

.hpiファイルをアップロードして、プラグインリポジトリ以外からプラグインをインストールできます。

② ファイル: **ファイルを選択** 選択 **前ページダウンロードしたファイルを選択**

③ **アップロード**

# 権限の設定

権限の設定を有効にする

The screenshot shows the Jenkins web interface. The left sidebar contains the following menu items: 新規ジョブ作成, 開発者, ビルド履歴, プロジェクト相関関係, フォール指紋チェック, **Jenkinsの管理** (highlighted with a red circle), My Views, Lockable Resources, and New View. The main content area displays a table of build jobs. The table has columns for S, W, 名前 ↓, 最新の成功ビルド, 最新の失敗ビルド, ビルド所要時間, and # Issues. The jobs listed are GroupingJob, MagicPodTest, MagicPodTest\_2, MagicPodTestSub, SampleJob, SampleJob2, and SampleRepositoryJob. Below the table, there is a legend for icons: 凡例, Atom feed 全ビルド, Atom feed 失敗ビルド, and Atom feed 最新ビルドのみ.

S	W	名前 ↓	最新の成功ビルド	最新の失敗ビルド	ビルド所要時間	# Issues
		<a href="#">GroupingJob</a>	1 ヶ月 29 日 - <a href="#">#17</a>	2 ヶ月 0 日 - <a href="#">#14</a>	13 秒	-
		<a href="#">MagicPodTest</a>	1 ヶ月 1 日 - <a href="#">#7</a>	1 ヶ月 1 日 - <a href="#">#6</a>	22 秒	-
		<a href="#">MagicPodTest_2</a>	3 ヶ月 19 日 - <a href="#">#11</a>	3 ヶ月 19 日 - <a href="#">#9</a>	16 秒	-
		<a href="#">MagicPodTestSub</a>	3 ヶ月 29 日 - <a href="#">#4</a>	—	16 秒	-
		<a href="#">SampleJob</a>	3 ヶ月 7 日 - <a href="#">#23</a>	3 ヶ月 7 日 - <a href="#">#20</a>	8.2 秒	-
		<a href="#">SampleJob2</a>	1 ヶ月 29 日 - <a href="#">#15</a>	2 ヶ月 2 日 - <a href="#">#10</a>	12 秒	19
		<a href="#">SampleRepositoryJob</a>	3 ヶ月 11 日 - <a href="#">#1</a>	3 ヶ月 7 日 - <a href="#">#2</a>	14 秒	-

アイコン: [S](#) [M](#) [L](#)

凡例 Atom feed 全ビルド Atom feed 失敗ビルド Atom feed 最新ビルドのみ

# 権限の設定

権限の設定を有効にする

The screenshot shows the Jenkins web interface. At the top, there's a search bar and a 'Logout' button. The main content area is titled 'Jenkinsの管理' (Jenkins Management). Below this, there's a notification about a new version (2.263.1) and a button to '自動でアップグレード' (Upgrade automatically). The 'System Configuration' section is visible, containing three main options: 'システムの設定' (System Configuration), 'Global Tool Configuration', and 'プラグインの管理' (Plugin Management). The 'システムの設定' option is further expanded to show 'ノードの管理' (Node Management) and 'グローバルセキュリティの設定' (Global Security Configuration). The 'グローバルセキュリティの設定' option is circled in red, indicating it's the target of the instruction. Below the 'Global Security Configuration' option, there's a description: 'Jenkinsのセキュリティを設定します。誰がシステムにアクセス、使用できるかなどを設定します。' (Configure Jenkins's security. Set who can access and use the system, etc.).

Jenkinsの管理

Jenkinsの新しいバージョン(2.263.1)をダウンロードできます (変更履歴)。 [自動でアップグレード](#)

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。  
▲ アップデートあり

ビルドキュー

ビルド実行状態

- 1 待機中
- 2 待機中

Security

- グローバルセキュリティの設定  
Jenkinsのセキュリティを設定します。誰がシステムにアクセス、使用できるかなどを設定します。
- Manage Credentials  
Configure credentials
- Configure Credential Providers  
Configure the credential providers and types

「グローバルセキュリティの設定」をクリック↓

# 権限の設定

権限の設定を有効にする

The screenshot shows the Jenkins 'Global Security Configuration' page. The page title is 'グローバルセキュリティの設定' (Global Security Configuration). Under the 'Authentication' section, the 'User Information' (ユーザー情報) is set to 'Jenkinsのユーザーデータベース' (Jenkins User Database). Under the 'Authorization' (権限管理) section, the 'Strategy' (権限管理 Strategy) is set to 'Role-Based Strategy'. The 'Save' (保存) button is highlighted with a red circle and the number 2. The 'Role-Based Strategy' option is also highlighted with a red circle and the number 1. A red arrow points to the 'Role-Based Strategy' option with the text '↓「Role-Based Strategy」を選択' (Select 'Role-Based Strategy').

① ↓「Role-Based Strategy」を選択

② 保存

「保存」をクリック

# 権限の設定

ユーザを追加する

The screenshot shows the Jenkins administration interface. At the top, there's a blue banner about downloading the latest version (2.263.1) and an '自動でアップグレード' button. Below that is a yellow warning box about build permissions. The main content is divided into two sections: 'System Configuration' and 'Security'. Under 'System Configuration', there are three options: 'システムの設定' (System Configuration), 'Global Tool Configuration', and 'プラグインの管理' (Plugin Management). Under 'Security', there are four options: 'グローバルセキュリティの設定' (Global Security Configuration), 'Manage Credentials', 'Configure Credential Providers', and 'In-process Script Approval'. The 'ユーザーの管理' (Manage Users) option is circled in red, and a red arrow points to it from the text 'ユーザを追加する' in the top left.

Jenkinsの新しいバージョン(2.263.1)をダウンロードできます (変更履歴)。 [自動でアップグレード](#)

Builds Jenkins run as the virtual Jenkins SYSTEM user with full Jenkins permissions by default. This can be a problem if some users have restricted or no access to some jobs, but can configure others. If that is the case, it is recommended to install a plugin implementing build authentication, and to override this default.

✗ No implementation of access control for builds is present. It is recommended that you install [Authorize Project Plugin](#) or another plugin implementing the [QueueItemAuthenticator](#) extension point.

## 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。  
🔴 アップデートあり

## Security

- グローバルセキュリティの設定**  
Jenkinsのセキュリティを設定します。誰がシステムにアクセス、使用できるかなどを設定します。
- Manage Credentials**  
Configure credentials
- Configure Credential Providers**  
Configure the credential providers and types
- In-process Script Approval**  
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.

**ユーザーの管理**  
Jenkinsにログインするユーザーの作成/削除/変更を実行します。

# 権限の設定

ユーザを追加する

①

「ユーザ作成」をクリック

②

ユーザ情報は入力  
あとで使用するので記録しておく

アドレスは何でもよい  
例) user@example.com

③

「ユーザ作成」をクリック

The image displays two screenshots of the Jenkins web interface. The top screenshot shows the 'Jenkinsのユーザーデータベース' (Jenkins User Database) page. In the left sidebar, the 'ユーザー作成' (Create User) option is circled in red. A red callout box points to it with the text '「ユーザ作成」をクリック' (Click 'Create User'). The main content area shows a table with one user entry: 'hayato\_senda'. The bottom screenshot shows the 'ユーザー作成' (Create User) form. The form fields are circled in red. A red callout box points to the form with the text 'ユーザ情報は入力あとで使用するので記録しておく' (Enter user information and record it for later use). Another red callout box points to the 'メールアドレス' (Email Address) field with the text 'アドレスは何でもよい 例) user@example.com' (Address can be anything, example: user@example.com). At the bottom of the form, the 'ユーザー作成' (Create User) button is circled in red, with a red callout box pointing to it with the text '「ユーザ作成」をクリック' (Click 'Create User').



# 権限の設定

ユーザを追加する

Jenkins ユーザーデータベース

ダッシュボードへ戻る  
Jenkinsの管理  
ユーザー作成

## ユーザー

これらのユーザーはJenkinsにログインすることができます。これは、Jenkinsにアクセスしていないけれどもいくつかのプロジェクトにコミットすると自動的に作成されるユーザーを含むこのリストの一部です。

User ID	名前
 <a href="#">hayato_senda</a>	<a href="#">hayato_senda</a> 
 <a href="#">user1</a>	 

← 追加されればOK

# 権限の設定

権限を追加する



Jenkins Jenkinsのユーザーデータベース

ダッシュボードへ戻る  
Jenkinsの管理  
ユーザー作成

## ユーザー

これらのユーザーはJenkinsにログインすることができます。これは、Jenkinsにアクセスしていないけれどもいくつかのプロジェクトにコミットすると自動的に作成されるユーザーを含む[このリスト](#)の一部です。

User ID	名前
 <a href="#">hayato_senda</a>	<a href="#">hayato_senda</a> 
 <a href="#">user1</a>	<a href="#">user1</a>  

# 権限の設定




## 権限を追加する

Jenkinsの新しいバージョン(2.263.1)をダウンロードできます (変更履歴)。 [自動でアップグレード](#)




Builds in Jenkins run as the virtual Jenkins SYSTEM user with full Jenkins permissions by default. This can be a problem if some users have restricted or no access to some jobs, but can configure others. If that is the case, it is recommended to install a plugin implementing build authentication, and to override this default.

✗ No implementation of access control for builds is present. It is recommended that you install [Authorize Project Plugin](#) or another plugin implementing the [QueueItemAuthenticator](#) extension point.

### 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。  
🔴 アップデートあり

### Security

-  **Manage and Assign Roles**  
Handle permissions by creating roles and assigning them to users/groups
-  **ユーザーの管理**  
Jenkinsにログインするユーザーの作成/削除/変更を実行します。
-  **In-process Script Approval**  
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.

↓「Manage and Assign Roles」をクリック

# 権限の設定

権限を追加する

Jenkins Manage and Assign Roles

新規ジョブ作成  
開発者  
ビルド履歴  
プロジェクト相関関係  
ファイル指紋チェック  
Jenkinsの管理  
My Views  
Lockable Resources  
New View

ビルドキュー  
ビルド待ち

ビルド実行状態  
1 待機中  
2 待機中

Manage Roles  
Assign Roles

↑「Manage Roles」をクリック

# 権限の設定

権限を追加する

**Manage Roles**

Global roles

Role	全体	認証情報				Agent				
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Role to add

1 developer

2 Add

権限名を入力する  
例) developer

↑「Add」をクリック

# 権限の設定

権限を追加する

追加した権限名に権限を設定する

※「admin」の権限は変更しないようにしてください！

John Doe

john.doe@example.jp

Manage Roles

Global roles

Role	全体	認証情報				Agent								ジョブ									
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Write
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
developer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

# 権限の設定

次にジョブ閲覧の権限を追加する

①

②

③

**Item roles**

Role	Pattern	認証情報	ジョブ
		Create Delete ManageDomains Update View Build Cancel Configure	Create Delete Discover Move Read W

Role to add: 権限名を入力 例) sample → sample

Pattern: ジョブ名を入力 例) Sample.\* → Sample.\*

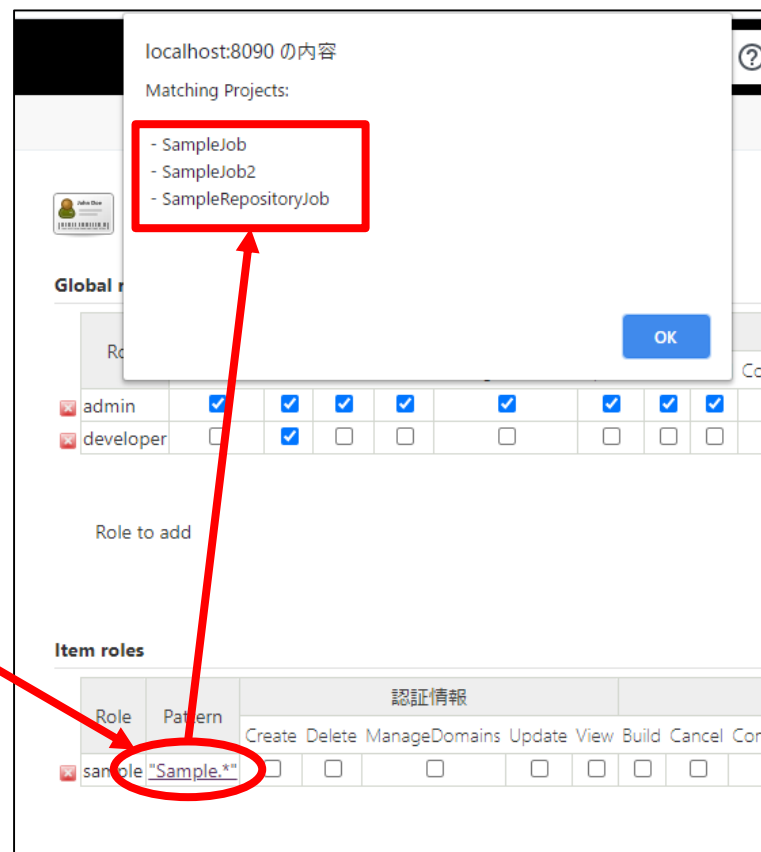
「Add」をクリック → Add

※「Sample.\*」とすると「Sample～」というジョブ名の閲覧が可能になる

# 権限の設定

ジョブ閲覧の権限を追加する

設定したジョブが実際にあるかどうかは、「Pattern」をクリックするを確認できる






# 権限の設定

ジョブ閲覧の権限を追加する

① 追加した権限名に権限を設定する

Item roles

Role	Pattern	認証情報					ジョブ							
		Create	Delete	ManageDomains	Update	View	Build	Cancel	Configure	Create	Delete	Discover	Move	Read
 sample	<a href="#">"Sample*"</a>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Role to add

sample

② 「Save」をクリックする

Save

Apply

# 権限の設定

追加したユーザに権限を設定する

Jenkins Manage and Assign Roles

新規ジョブ作成  
開発者  
ビルド履歴  
プロジェクト相関関係  
ファイル指紋チェック  
Jenkinsの管理  
My Views  
Lockable Resources  
New View

ビルドキュー  
ビルド待ち

ビルド実行状態  
1 待機中  
2 待機中

Manage and Assign Roles

Manage Roles  
Assign Roles  
Role Strategy Macros

↑「Assign Roles」をクリック

# 権限の設定

追加したユーザに権限を設定する

①

User/group	admin	developer
hayato_senda	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Anonymous	<input type="checkbox"/>	<input checked="" type="checkbox"/>

User/group to add

Add

User/group	sample
Anonymous	<input type="checkbox"/>

User/group to add

Add

②

③

操作ユーザ →「admin」にチェック  
Anonymous →追加した権限にチェック

72ページで追加したユーザ名を入力

「Add」をクリック

# 権限の設定

追加したユーザに権限を設定する

**Assign Roles**

**Global roles**

User/group	admin	developer
hayato_senda	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Anonymous	<input type="checkbox"/>	<input checked="" type="checkbox"/>

User/group to add:

**Item roles**

User/group	sample
Anonymous	<input type="checkbox"/>
user1	<input checked="" type="checkbox"/>

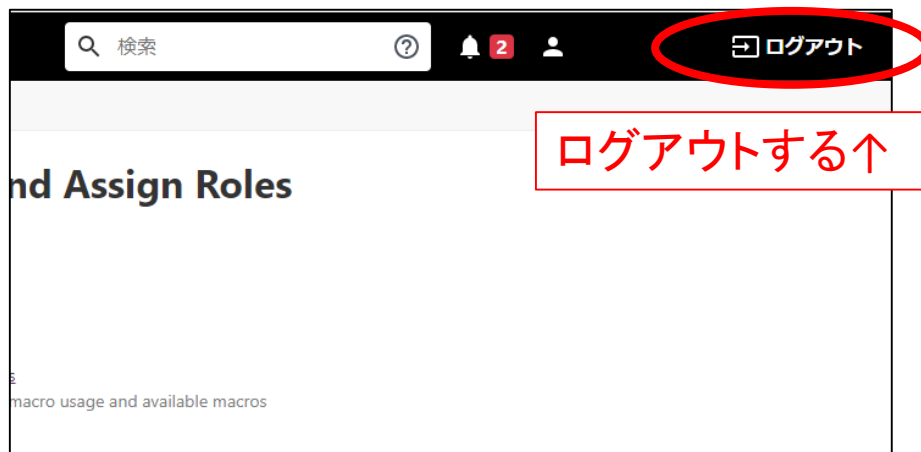
User/group to add:

ジョブ閲覧の権限にチェック

↓「Save」をクリック

# 権限の設定

権限の設定を確認する



ログアウトする↑



追加したユーザでログインする↑

# 権限の設定

権限の設定を確認する



The screenshot shows the Jenkins dashboard for a user named 'user1'. The left sidebar contains navigation links: 開発者, ビルド履歴, プロジェクト相関関係, ファイル指紋チェック, and My Views. The main area displays a table of jobs. A red circle highlights the first two jobs, 'SampleJob' and 'SampleJob2', indicating they are visible to the user. The table columns are: S, W, 名前 ↓, 最新の成功ビルド, 最新の失敗ビルド, ビルド所要時間, and # Issues.

S	W	名前 ↓	最新の成功ビルド	最新の失敗ビルド	ビルド所要時間	# Issues
		<a href="#">SampleJob</a>	3 ヶ月 8 日 - <a href="#">#23</a>	3 ヶ月 8 日 - <a href="#">#20</a>	8.2 秒	-
		<a href="#">SampleJob2</a>	2 ヶ月 0 日 - <a href="#">#15</a>	2 ヶ月 3 日 - <a href="#">#10</a>	12 秒	19
		<a href="#">SampleRepositoryJob</a>	3 ヶ月 12 日 - <a href="#">#1</a>	3 ヶ月 8 日 - <a href="#">#2</a>	14 秒	-

アイコン: [S](#) [M](#) [L](#)

凡例 [Atom feed 全ビルド](#) [Atom feed 失敗ビルド](#) [Atom feed 最新ビルドのみ](#)

設定通りにジョブが閲覧でき、  
設定ができないことが確認できればOK