

# 神情協 先端技術研究会 「使えるJenkinsを構築する」 第3回

---

株式会社ジェイエスピー

A solid green horizontal bar at the bottom of the slide.

# 本日の予定

---

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

以下のアカウントを  
確認してください！

- Jenkins
- MagicPod
- GitHub
- 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

## 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



# 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	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。

# 本日の予定

---

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



# 前回について

---

## <目標>

- ① Jenkins でほかのツールと連携する。
- ② MagicPod で画面の自動テストをする。

キーワード : Checkstyle、FindBugs、MagicPod

# Jenkins に触ってみる

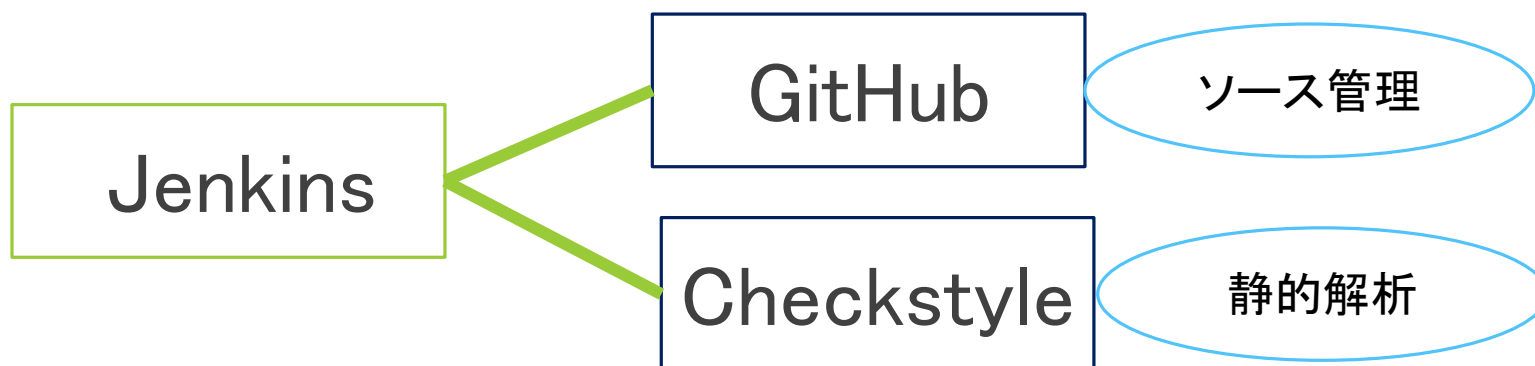
---

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
5. プライベートリポジトリを作る
6. ワークスペースを変える
7. GitHubからソースを取得する
8. Checkstyle で静的解析を行う

# これのできること

---

プラグインで連携可能！  
より開発環境に近い形に！



# MagicPod について

---

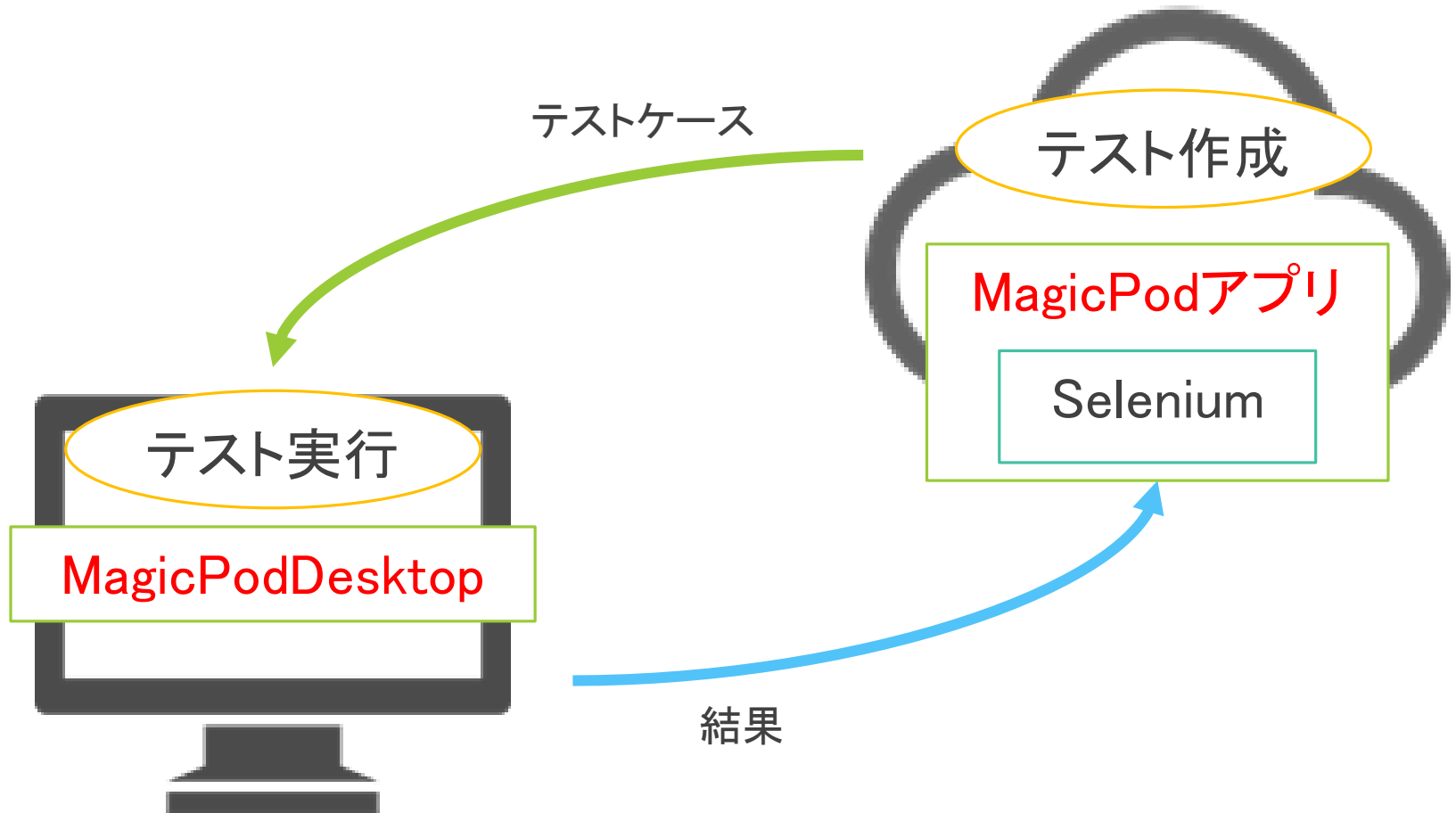
- Web画面の自動テストを行う
- 簡単操作でテストケースを作成
- ネットに接続できる環境であれば実行可能



Magic Pod

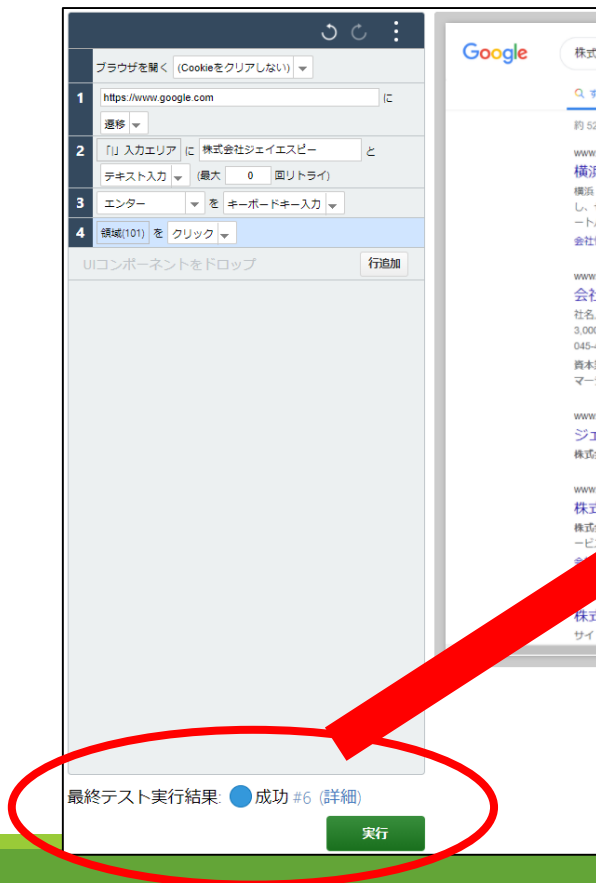
<https://www.magic-pod.com/> より引用

# MagicPodについて

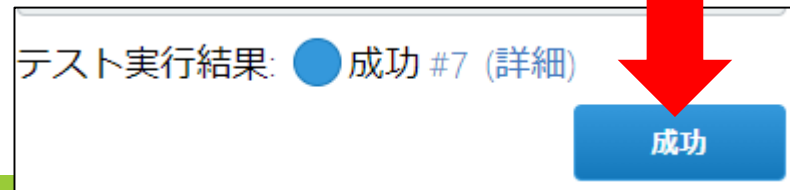
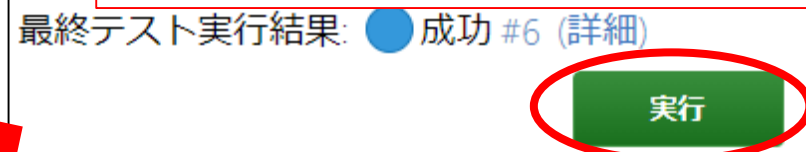


# テスト実行

前回作成したテストを実行して成功になることを確認する



「実行」ボタンを押す。  
アプリの更新があれば更新しましょう！



# 本日の予定

---

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

---

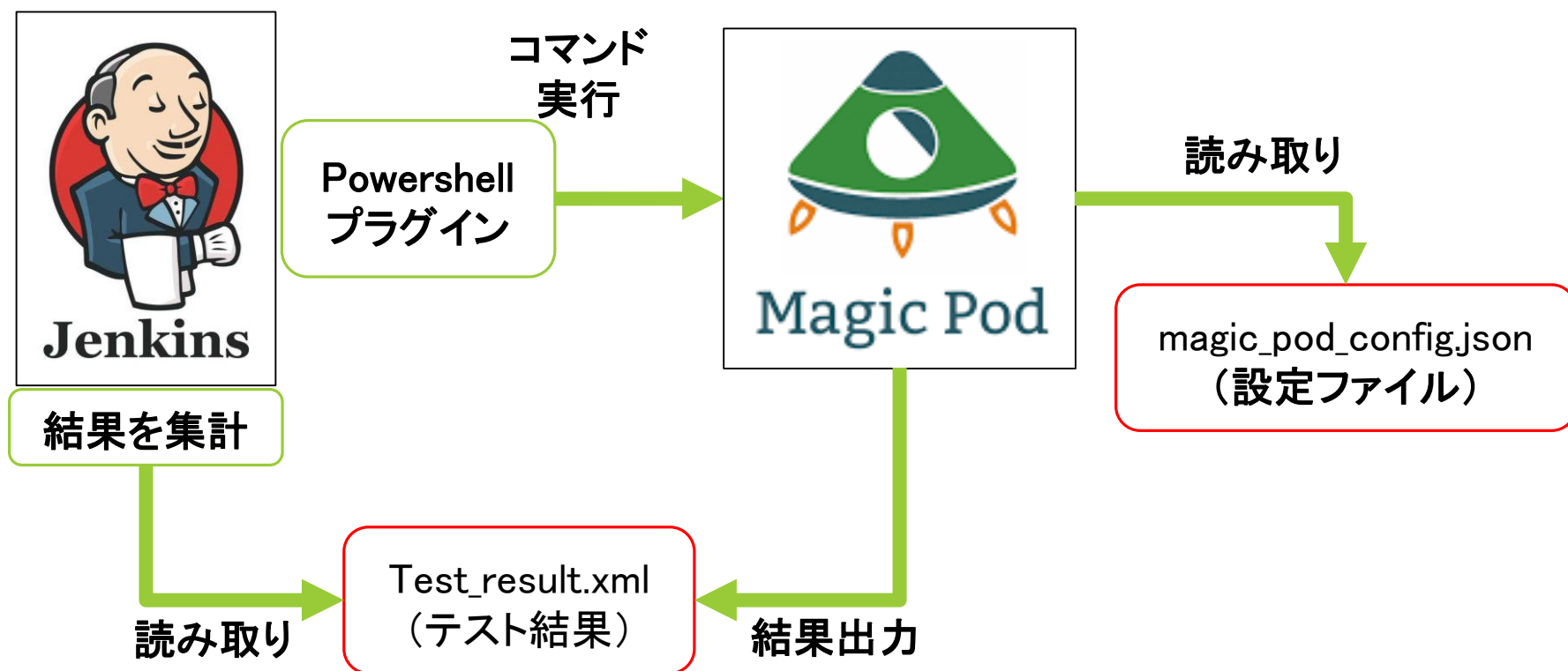
1. 設定ファイルの確認 (MagicPod)
2. 設定ファイルを編集 (MagicPod)
3. Powershellでコマンド実行 (MagicPod)
4. Powershellプラグインをインストール (Jenkins)
5. ジョブの設定と実行 (Jenkins、MagicPod)
6. テスト結果の表示 (Jenkins、MagicPod) 【課題】

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」



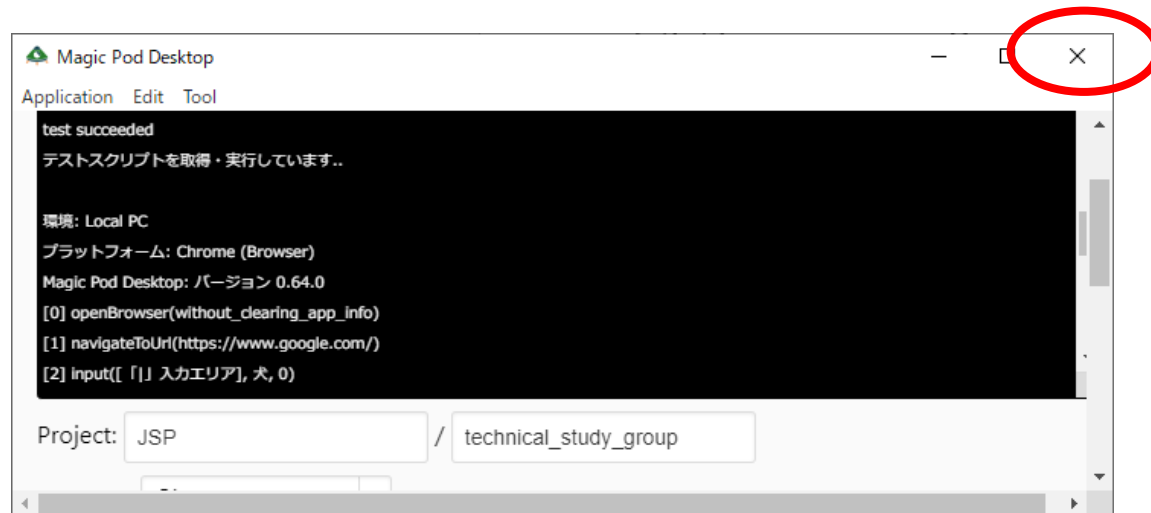
# Jenkins と MagicPod の連携(概要)



# 設定ファイルの確認

JenkinsとMagicPodとの連携で使うファイルを確認します。

## ① MagicPodDesktop を閉じる

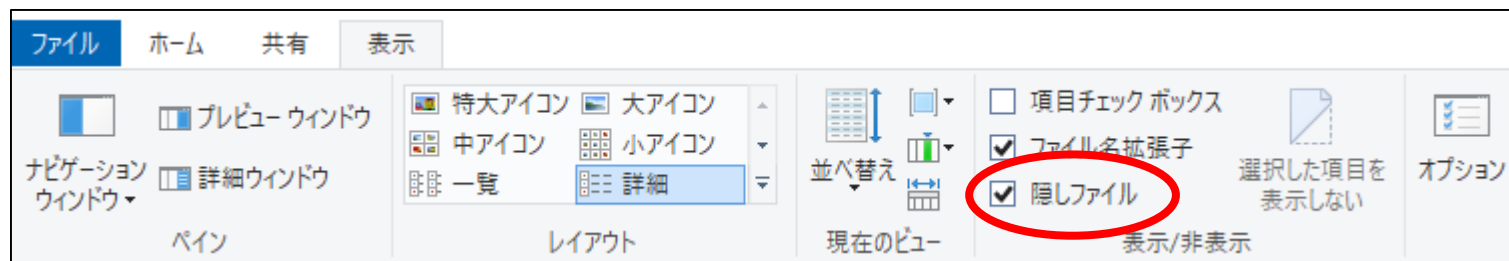


# 設定ファイルの確認

② 以下のディレクトリを開く。

C:¥Users¥(ユーザ名)¥AppData¥Roaming¥magic\_pod\_desktop

※隠しファイルの表示をONにする



# 設定ファイルの確認

③ 「magic\_pod\_config.json」を開く。

```
1  ↓
2  "owner": "JSP", ↓
3  "capabilities": [ ↓
4    "browserName": "chrome" ↓
5  ], ↓
6  "excludedTestCaseLabels": [], ↓
7  "sharedDataPatternRows": [], ↓
8  "capabilities": [ ↓
9    "browserName": "chrome" ↓
10 ], ↓
11 "baseUrl": "https://www.jspnet.co.jp/", ↓
12 "envVars": [], ↓
13 "xmlTestOutput": true, ↓
14 "workDir": "C:\\¥¥Tools", ↓
15 "authTokenFilePath": "C:\\¥¥Users\\[redacted].magic_pod_token", ↓
16 "proxyServerUrl": "auto", ↓
17 "proxyServerAuthType": "none", ↓
18 "proxyServerAuthUser": "", ↓
19 "proxyServerAuthPassword": "", ↓
20 "logLevel": "beginner", ↓
21 "testCondition": { ↓
22   "device_type": "desktop", ↓
23   "browser": "chrome" ↓
24 }, ↓
25 "captureType": "on_each_step", ↓
26 "stepCaptureScope": "html", ↓
27 "sendMail": false, ↓
28 "retryCount": 0 ↓
29 } [EOF]
```

いろいろ書いてありますが、これを使います！

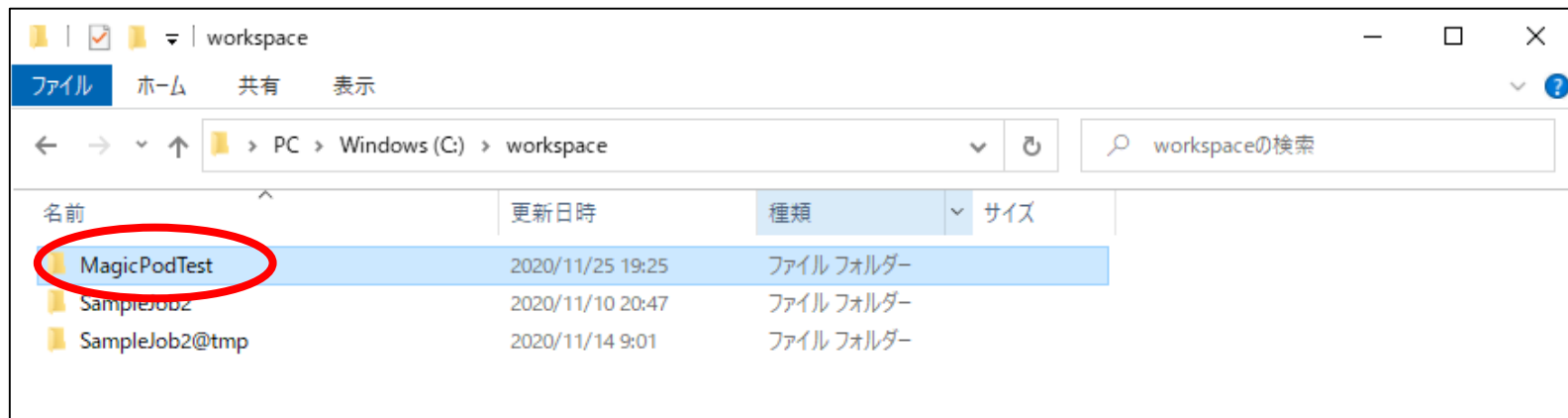
ここまで終わったら  
「手を挙げる」

# 設定ファイルの編集

エクスプローラを開き、前回作成した「 C:\workspace 」の直下に移動。

「MagicPodTest」というフォルダを作成する。

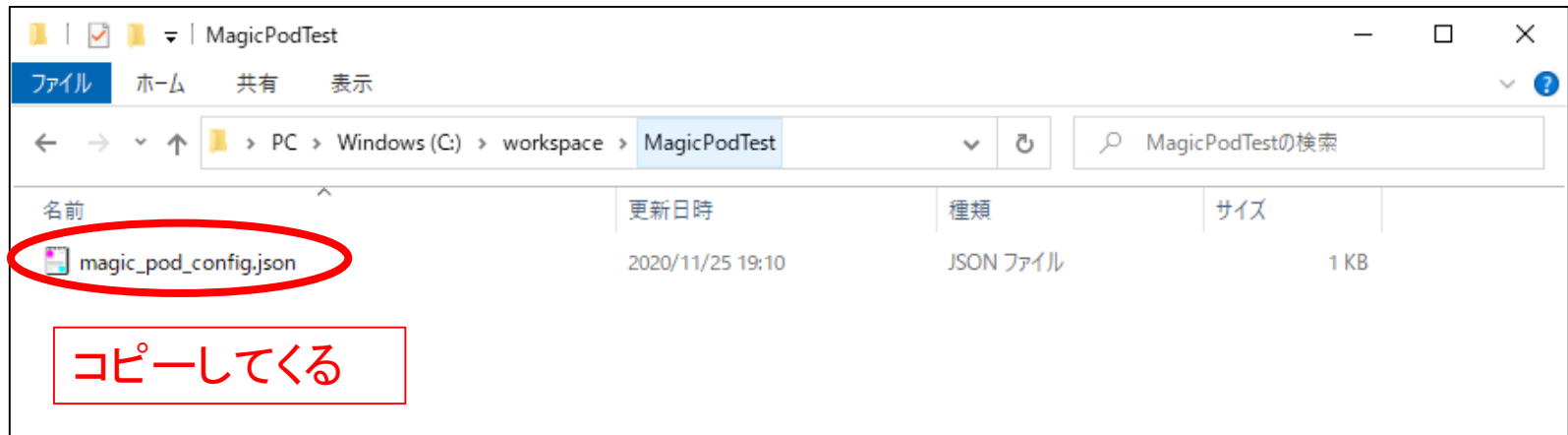
※「 workspace 」というフォルダを作っていない場合は、作成する。



# 設定ファイルの編集

C:\workspace¥ MagicPodTest に

設定ファイル「magic\_pod\_config.json」をコピーする。



# 設定ファイルを編集

```
{
  "owner": "JSP",
  "project": "(プロジェクト名)",
  "testNumber": "(テスト番号)",
  "capabilities": {
    "browserName": "chrome"
  },
  "baseUrl": "(指定のURL)",
  "xmlTestOutput": false,
  "workDir": "C:¥¥Users¥¥(ユーザ名)/Desktop/magicPod",
  "authTokenFilePath": "C:¥¥Users¥¥(ユーザ名)/.magic_pod_token",
  "logLevel": "beginner",
  "testCondition": {},
  "captureType": "on_each_step",
  "stepCaptureScope": "html",
  "sendMail": false,
  "retryCount": 0
}
```

①

②

③

④

Powershell で実行するために  
編集する箇所は 3 か所

+ 注意点

# 設定ファイルを編集

---

## ① ブラウザの設定(赤字部分を追加)

```
"capabilities": {  
  "browserName": "chrome",  
  "goog:chromeOptions": {  
    "args": [ "--disable-features=VizDisplayCompositor" ]  
  }  
},
```

「,」を忘れないように



# 設定ファイルを編集

---

## ② テスト結果をXML形式で出力(赤字部分を修正)

"xmlTestOutput": **true**,

True にするとXML形式で出力してくれる。  
このファイルをJenkinsに読ませる。

# 設定ファイルを編集

---

## ③ ワークスペースの指定(赤字部分を修正)

→ テスト結果のファイルがここに作られる。

“workDir”: “C:¥¥workspace¥¥MagicPodTest”,

「¥」では読まれないため、「¥¥」としておく

# 設定ファイルを編集

---

## 【注意点】 ④ メール送信設定

"sendMail": false,

↑ 必ず「 false 」にしてください  
このメンバ全員にメールが飛んでしまいます。

ここまで終わったら  
「手を挙げる」

# Powershellで確認

---

Jenkins では MagicPod をWeb経由ではなく、  
**Powershell** を使って、コマンド実行をします。  
(コマンドが厄介…)

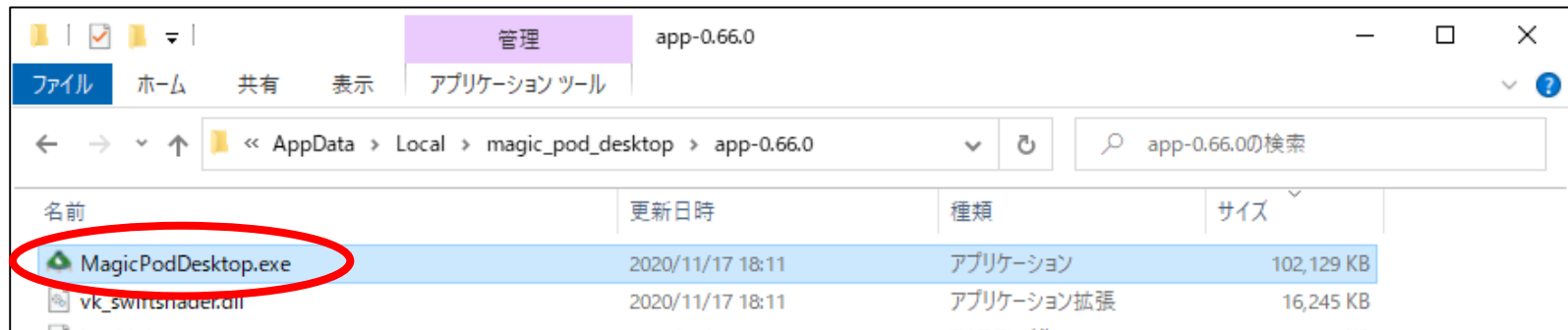
Windows の メモ帳 を開いてください！

# MagicPodDesktop.exeのパス

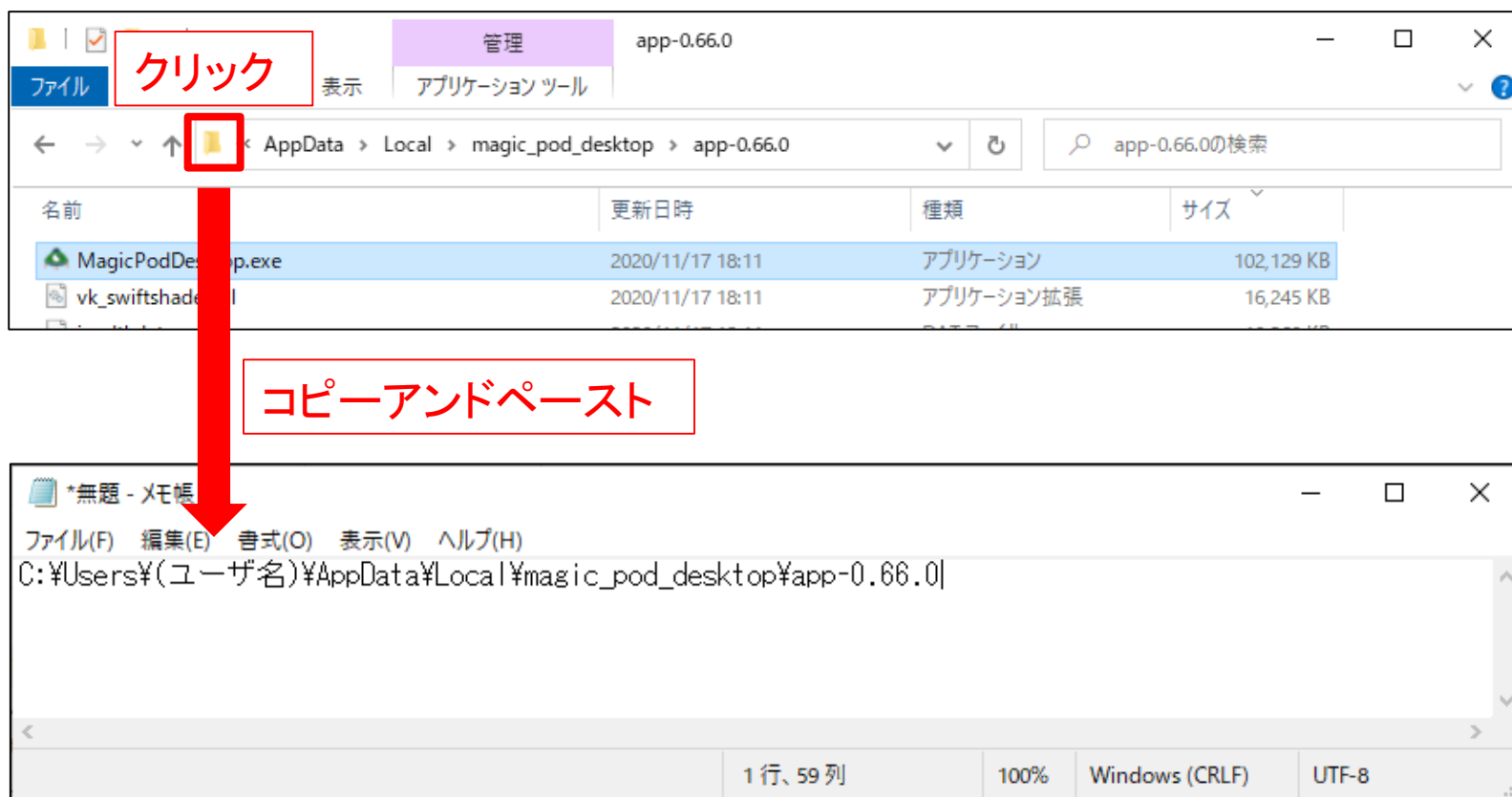
↓にあるはず.....なければ教えてください(隠しファイルを表示)

C:¥Users¥(ユーザ名)¥AppData¥Local¥magic\_pod\_desktop¥app-0.67.0¥MagicPodDesktop.exe

※MagicPodのバージョンによって名前が異なるので注意



# MagicPodDesktop.exeのパス



# MagicPodDesktop.exeのパス

最後に「¥MagicPodDesktop.exe」を付ける



# Powershellで確認

---

## コマンド

↓メモ帳にコピーアンドペーストします(改行はしない！)

```
& "(MagicPodDesktop.exeのパス)" run --magic_pod_config="C:¥workspace¥MagicPodTest¥magic_pod_config.json"; Wait-Process -Name MagicPodDesktop -Timeout 1200
```

拡大したもの:

```
& "(MagicPodDesktop.exeのパス)" run --  
magic_pod_config="C:¥workspace¥MagicPodTest¥magic_pod_config.json";  
Wait-Process -Name MagicPodDesktop -Timeout 1200
```

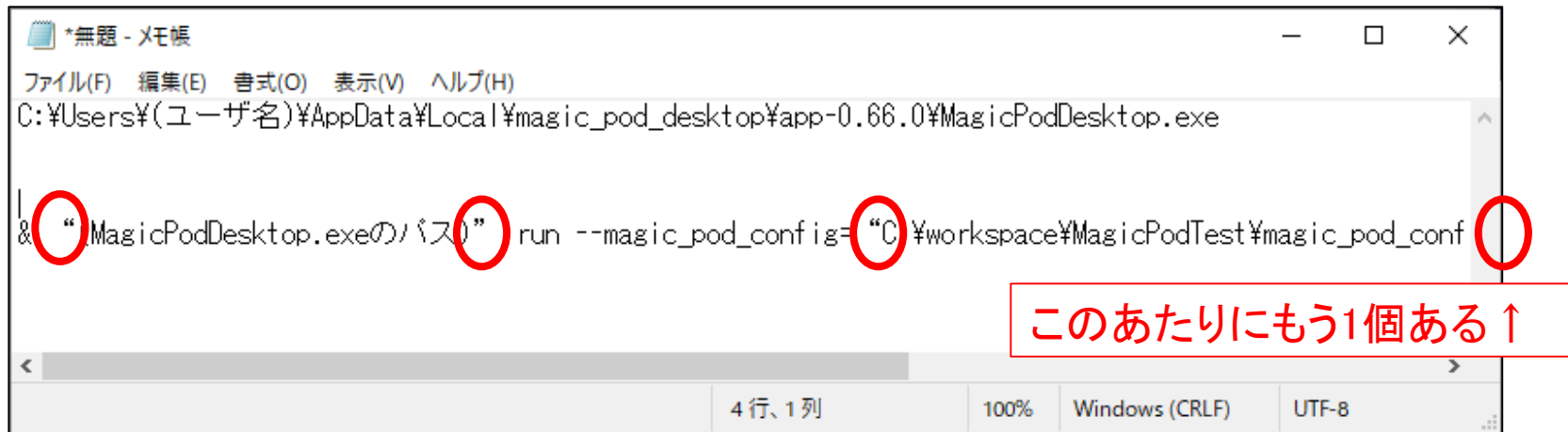
次ページでこのコマンドを編集します。



# Powershellで確認

## コマンド作成

ダブルクォーテーションが全角になっているので、すべて半角に直す



The screenshot shows a Notepad window titled "\*無題 - メモ帳". The menu bar includes "ファイル(F)", "編集(E)", "書式(O)", "表示(V)", and "ヘルプ(H)". The text area contains the following command:

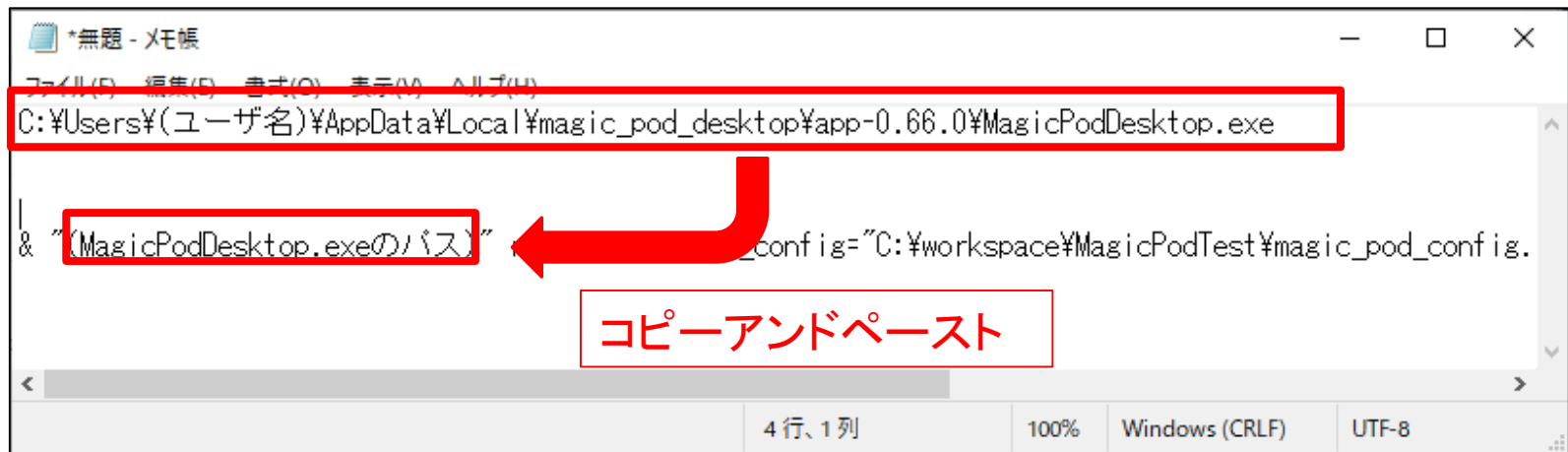
```
C:¥Users¥(ユーザ名)¥AppData¥Local¥magic_pod_desktop¥app-0.66.0¥MagicPodDesktop.exe  
|  
& “.MagicPodDesktop.exeのパス” run --magic_pod_config= “C:¥workspace¥MagicPodTest¥magic_pod_conf
```

Four red circles are drawn around the full-width double quotes in the command. A red box with the text "このあたりにもう1個ある↑" points to the end of the command line.

At the bottom of the window, the status bar shows "4 行、1 列", "100%", "Windows (CRLF)", and "UTF-8".

# Powershellで確認

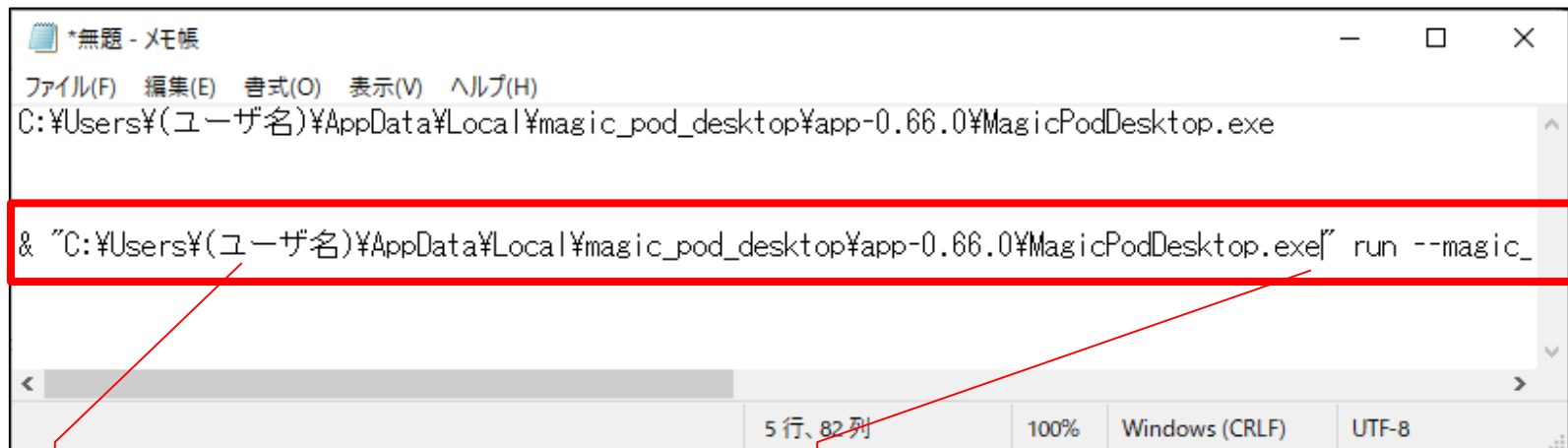
## コマンド作成



```
*無題 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
C:\Users\ユーザー\AppData\Local\magic_pod_desktop\app-0.66.0\MagicPodDesktop.exe
| & "(MagicPodDesktop.exeのパス)" _config="C:\workspace\MagicPodTest\magic_pod_config."
4行、1列 100% Windows (CRLF) UTF-8
```

# Powershellで確認

## コマンド作成



```
*無題 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
C:¥Users¥(ユーザ名)¥AppData¥Local¥magic_pod_desktop¥app-0.66.0¥MagicPodDesktop.exe
& "C:¥Users¥(ユーザ名)¥AppData¥Local¥magic_pod_desktop¥app-0.66.0¥MagicPodDesktop.exe" run --magic_
5行、82列 100% Windows (CRLF) UTF-8
```

PCのユーザ名になってますか？

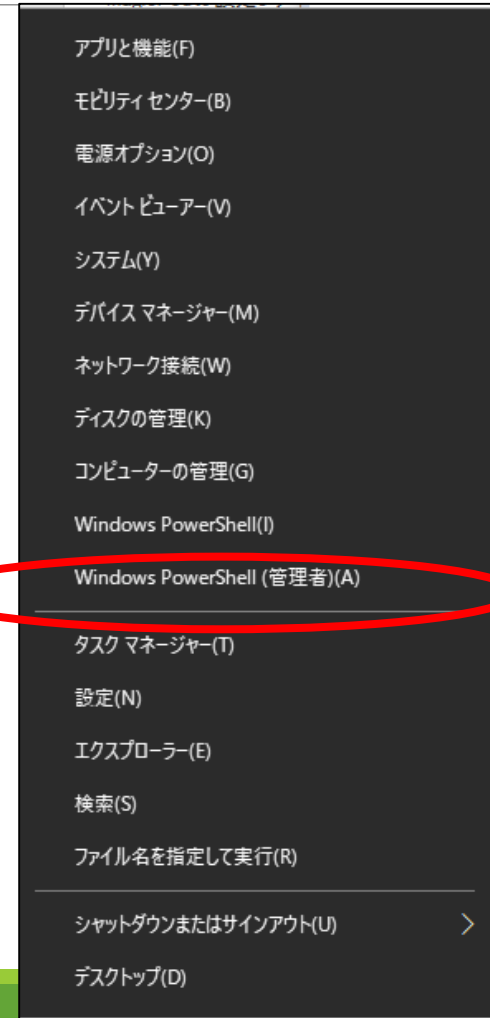
余計な空白はありませんか？

# Powershellで確認



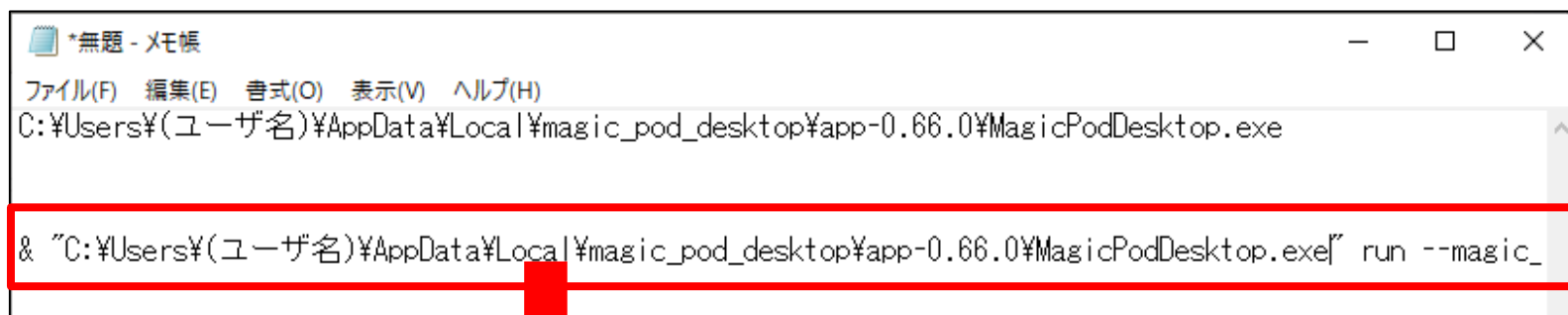
右クリック

クリック！  
許可が求められるので、  
「はい」を選ぶ



# Powershellで確認

## コマンド実行



A screenshot of a Notepad window titled "\*無題 - メモ帳". The menu bar includes "ファイル(F)", "編集(E)", "書式(O)", "表示(V)", and "ヘルプ(H)". The text area contains the command: `C:¥Users¥(ユーザ名)¥AppData¥Local¥magic_pod_desktop¥app-0.66.0¥MagicPodDesktop.exe`. Below this, a red rectangular box highlights the command: `& "C:¥Users¥(ユーザ名)¥AppData¥Local¥magic_pod_desktop¥app-0.66.0¥MagicPodDesktop.exe" run --magic_`. A large red arrow points from this box down to the PowerShell window below.

## コピーアンドペースト



A screenshot of a Windows PowerShell window titled "管理者: Windows PowerShell". The text area shows the command being pasted: `& "C:¥Users¥[redacted]¥AppData¥Local¥magic_pod_desktop¥app-0.66.0¥MagicPodDesktop.exe" run --magic_pod_`. The command is partially visible, with the rest of the command from the previous screenshot being pasted. The command continues with `config="C:¥workspace¥MagicPodTest¥magic_pod_config.json"; Wait-Process -Name MagicPodDesktop -Timeout 1200`.

# Powershellで確認

The screenshot shows a Windows PowerShell window titled "管理者: Windows PowerShell". The terminal output includes:

```
[1606299731.084][INFO]: Waiting for pending navigations...
[1606299731.085][INFO]: Done waiting for pending navigations. Status: ok
[1606299731.085][INFO]: [4c651ccb9124e6a20601ae12307aebdf] RESPONSE SwitchToFrame
[1606299731.093][INFO]: [4c651ccb9124e6a20601ae12307aebdf] COMMAND Screenshot {
}
[1606299731.093][INFO]: Waiting for pending navigations...
[1606299731.094][INFO]: Done waiting for pending navigations. Status: ok
[1606299731.596][INFO]: Waiting for pending navigations...
[1606299731.624][INFO]: Done waiting for pending navigations. Status: ok
[1606299731.624][INFO]: [4c651ccb9124e6a20601ae12307aebdf] RESPONSE 4CAYAAAAY
au7NAAAAXNSROIARS4c6QAAIABJREFUeJzs3XusP8993/X37D1znkeonm...TSURKQQkA9EWKYITFEVShobIEUZTADUZSCHAA1pKQtJaKK
SlzEHQVRqGhKiOLtJI2S2sFpAk3tYPKL3fgWX..."
Sending test screen captures... 1 more captures need to be sent
Test result: https://magic-pod.com/JSP/technical_study_group/batch-run/2/
[info - 2020-11-25 19:22:15 - C:\Users\█████\AppData\Local\magic_pod_desktop$app-0.66.0$resources$app.asar$lib] driver.sto
PS C:\Windows\system32>
```

A red speech bubble overlay contains the Japanese text: **最後にこれが出ていればテスト完了！**

## ここまで終わったら 「手を挙げる」

# Powershellプラグインをインストール

Powershellのプラグインを入れる。

オフラインの場合は  
44ページへ



# Powershellプラグインをインストール

Powershellのプラグインを入れる。

## 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。



# Powershellプラグインをインストール

Powershellのプラグインを入れる。



The screenshot shows the PowerShell Gallery interface. At the top, there is a search bar with the placeholder text 'search'. Below the search bar, there are three tabs: 'アップデート' (Updates), '利用可能' (Available), and 'インストール済み' (Installed). The '利用可能' tab is highlighted with a red circle. A red box with the text '「利用可能」のタブを押す' (Press the 'Available' tab) points to this tab. Below the tabs, there is a table header with columns: 'インストール ↑ 名前' (Install ↑ Name), 'バージョン' (Version), and 'Released'. Below the header, there is a message: 'Use the search field above to search for available plugins.' At the bottom, there are three buttons: '再起動せずにインストール' (Install without restarting), 'ダウンロードして再起動後にインストール' (Download and install after restarting), and 'データ更新時刻: 8 時間 32 分' (Data update time: 8 hours 32 minutes). To the right of the update time, there is a '更新' (Update) button.

# Powershellプラグインをインストール

Powershellのプラグインを入れる。

The screenshot shows the IntelliJ IDEA plugin marketplace interface. A search bar at the top contains the text "powershell". Below the search bar, there are tabs for "アップデート", "利用可能", "インストール済み", and "高度な設定". The "インストール" tab is selected, showing a list of plugins. The "PowerShell" plugin is highlighted, and its checkbox is checked. The "Venafi CodeSign Protect" plugin is also visible below it. At the bottom, there are two buttons: "再起動せずにインストール" (Install without restarting) and "ダウンロード" (Download). The "再起動せずにインストール" button is circled in red.

① ② ③

「powershell」と入力

「PowerShell」にチェックを付ける

「再起動せずにインストール」を押す

インストール ↑	名前	バージョン	Released
<input checked="" type="checkbox"/>	<a href="#">PowerShell</a> ビルドツール .NET Dev This plugin allows Java and PowerShell to run on Linux.	1.4	9ヶ月 14日 ago
<input type="checkbox"/>	<a href="#">Venafi CodeSign Protect</a> Security This plugin enables code signing for JAR files, executables or PowerShell using the Venafi PKCS#11 Driver.	1.0.0	1ヶ月 18日 ago

# Powershellプラグインをインストール

Powershellのプラグインを入れる。

## 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

PowerShell

● 成功

成功すれば完了！

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

オフラインの場合

# Powershellプラグインをインストール

---

## ■ Powershell

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

プラグインのインストールは次ページ

# Powershellプラグインをインストール

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

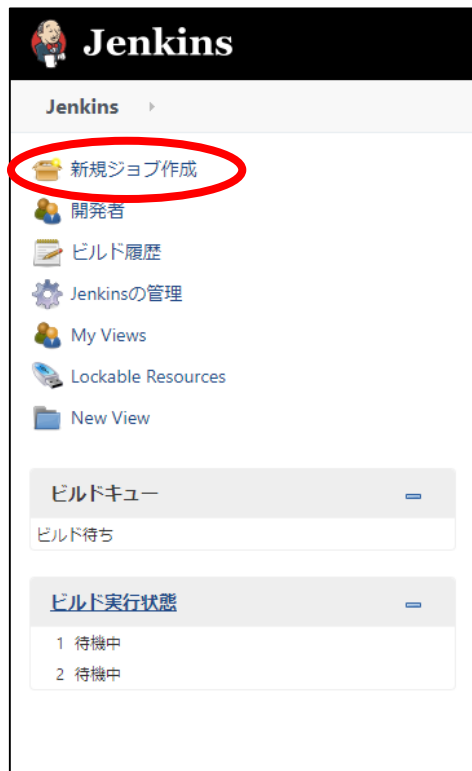
.hpiファイルをアップロードして、プラグインリポジトリ以外からプラグインをインストールできます。

② ファイル: **ファイルを選択** 選択 **先程ダウンロードしたファイルを選択**

③ **アップロード**

# ジョブの設定と実行

以前作ったジョブをコピーせずに、新規に作成する



# ジョブの設定と実行

## 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 ↓

# ジョブの設定と実行

チェック

①

☒ カスタムワークスペースを使用

②

ディレクトリ C:\workspace\MagicPodTest

表示用プロジェクト名

↑ 「C:¥workspace¥MagicPodTest」と入力する

☐ 依存している上流プロジェクトのビルドを保存

保存

Apply

一度保存し、ビルドを実行する  
成功すればOK



# ジョブの設定と実行

再度、設定画面を開く

プロジェクト MagicPodTest

ワークスペース

変更履歴

永続リンク

- 最新のビルド (#1), 48 秒
- 最新の安定ビルド (#1), 48 秒
- 最新の成功ビルド (#1), 48 秒
- 最新の完了ビルド (#1), 48 秒

ビルド履歴

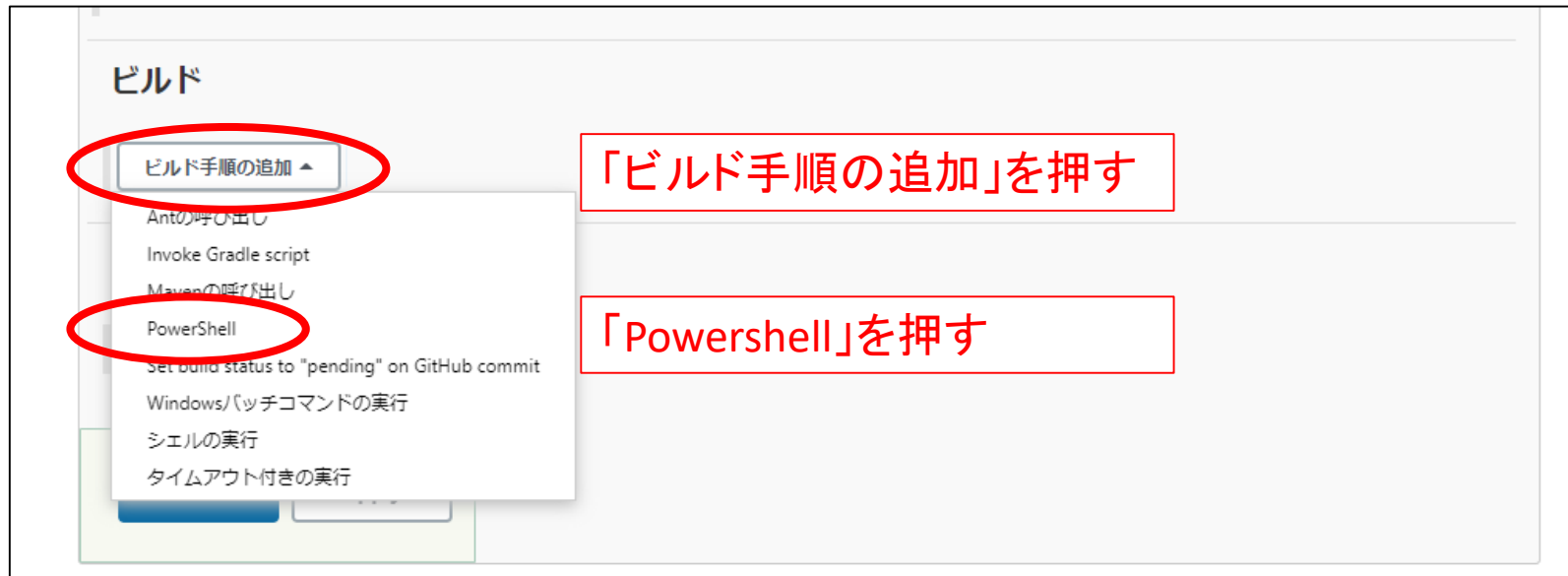
find

#1 2020/11/27 19:05

Atom feed 全ビルド Atom feed 失敗ビルド

# ジョブの設定と実行

## ビルドの項目



# ジョブの設定と実行

ビルド

PowerShell

Command

1 & "C:\Users\jsp\AppData\Local\magic\_pod\_desktop\app-0.66.0\MagicPodDesktop.exe" run --magic\_pod\_config

See [the list of available environment variables](#)

☒ Stop On Errors

☒ Use PowerShell profile

ビルド手順の追加 ▾

PowerShellで実行したものをそのまま入力する  
↓メモ帳からコピーアンドペースト

保存 Apply

保存

# ジョブの設定と実行

**Jenkins** 検索 ?

Jenkins > MagicPodTest

プロジェクト MagicPodTest

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

ワークスペース  
変更履歴

永続リンク

- 最新のビルド (#2), 1分 4 秒
- 最新の安定ビルド (#2), 1分 4 秒

ビルド履歴 推移

find	
<b>#2</b>	2020/11/27 19:13
#1	2020/11/27 19:05

Atom feed 全ビルド Atom feed 失敗ビルド

**ビルド実行**

**成功しても失敗しても  
とりあえず結果を見ましょう**

# ジョブの設定と実行

「コンソール出力」を押す



The screenshot displays the Jenkins web interface. At the top, the Jenkins logo and a search bar are visible. The breadcrumb navigation shows 'Jenkins > MagicPodTest > #2'. The left sidebar contains several links: 'プロジェクトへ戻る' (Return to project), '状態' (Status), '変更履歴' (Change history), 'コンソール出力' (Console output, circled in red), '説明の編集' (Edit description), 'Delete build '#2'', and '前のビルド' (Previous build). The main content area features a large blue sphere icon and the title 'ビルド #2 (2020/11/27 19:13:36)'. Below this, there is a message '変更点はありません。' (No changes) and a yellow diamond icon indicating the build was executed by the user 'hayato senda'.

# ジョブの設定と実行

Jenkins ▸ MagicPodTest ▸ #2

- プロジェクトへ戻る
- 状態
- 変更履歴
- コンソール出力
- ブレンテキスト表示
- 説明の編集
- Delete build '#2'
- 前のビルド

## 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\MagicPodTest  
[MagicPodTest] \$ powershell.exe -NonInteractive -ExecutionPolicy Bypass -File  
C:\Windows\TEMP\jenkins6528764170525228399.ps1

Starting ChromeDriver 87.0.4280.20 (c99e81631faa0b2a448e858c0dbd8311fb04ddbd-refs/branch-heads/4280@{#355}) on port 4723

Only local connections are allowed.

Please see <https://chromedriver.chromium.org/security-considerations> for suggestions on keeping ChromeDriver safe.

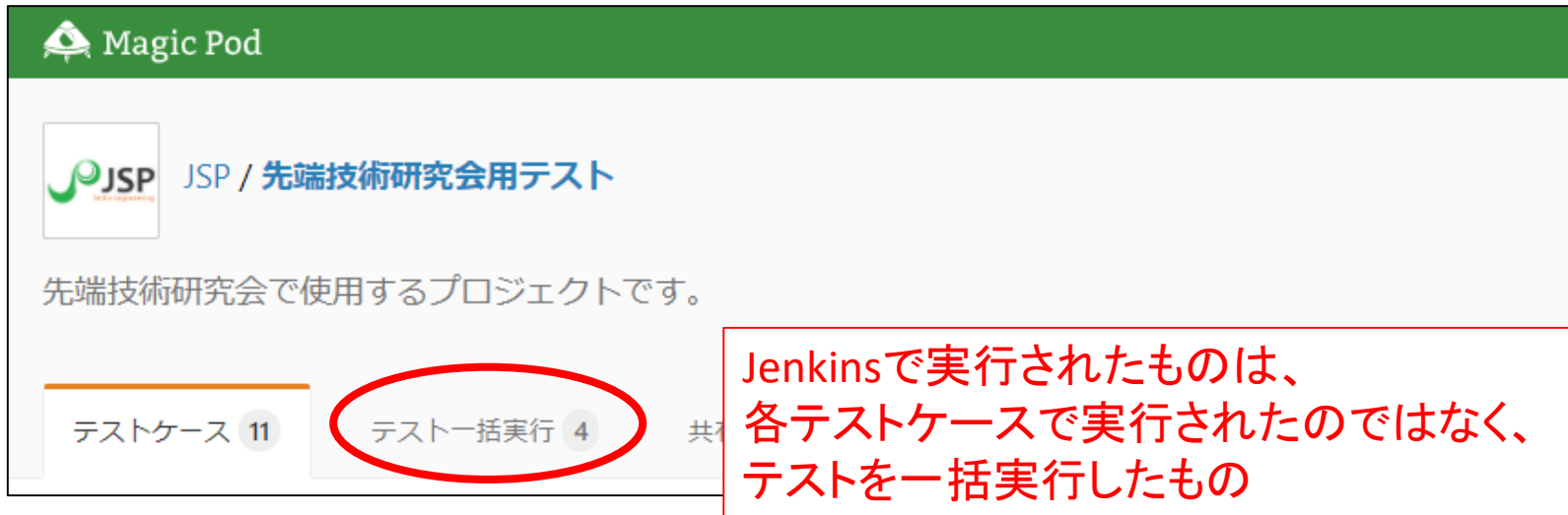
■  
■ 一番下までスクロール  
■

Sending test screen captures... 1 more captures needs to be sent  
Test result: [https://magic-pod.com/JSP/technical\\_study\\_group/batch-run/3/](https://magic-pod.com/JSP/technical_study_group/batch-run/3/)  
[info - 2020-11-27 19:13:58 - C:\Users\jisp\AppData\Local\magic\_pod\_desktop\app-0.66.0\resources\app.asar\lib] driver.stop  
Finish

↑ Powershell で実行した結果と同じ出力が出る

# ジョブの設定と実行

MagicPod側ではどうなっているか？  
Webの画面を見してみる



The screenshot shows the Magic Pod web interface. At the top is a green header with the Magic Pod logo and name. Below it, the job name "JSP / 先端技術研究会用テスト" is displayed next to a JSP logo. A description reads "先端技術研究会で使用するプロジェクトです。". At the bottom, there are two buttons: "テストケース 11" and "テスト一括実行 4". The "テスト一括実行 4" button is circled in red. To the right of the buttons, a red-bordered box contains the following text:

Jenkinsで実行されたものは、  
各テストケースで実行されたのではなく、  
テストを一括実行したもの

# ジョブの設定と実行

「テスト一括実行」の画面  
自分が実行したものを探してみる

テストケース 11

テスト一括実行 4

共有ステップ 0

メンバー 14

設定

設定: 4: デフォルト プライベート ▼

テストを一括実行

4件のデータ

↓結果に何を実行したかが書いてある

#		所要時間	開始時間	
●	4	Chrome 87	00:00:11	2020年11月27日19:29 >
●	3	Chrome 87	00:00:13	2020年11月27日19:13 >
●	2	Chrome 87	00:00:14	2020年11月25日19:22 >
●	1	Chrome 87	00:00:14	2020年11月25日19:19 >



# ジョブの設定と実行

「テスト一括実行結果」の画面  
自分が実行したテストの時間や開始時間を確認できる

テスト一括実行 / テスト一括実行結果 #4

Chrome 87	↓詳細やキャプチャはテストケース名をクリック			...
#	テストケース	所要時間	開始時間	
1	#1 テスト	00:00:10	2020年11月27日19:29	>

1件のテスト実行結果

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. 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

# 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

# 成果発表について

---

## ■ 困ったことの内容は自由

- C# で開発して Jenkins で自動ビルドをしたい。
- 画面試験を Jenkins から自動で実施してほしい。
- 結果を Slack に通知してほしい。
- 日報を自動生成してほしい。 etc...

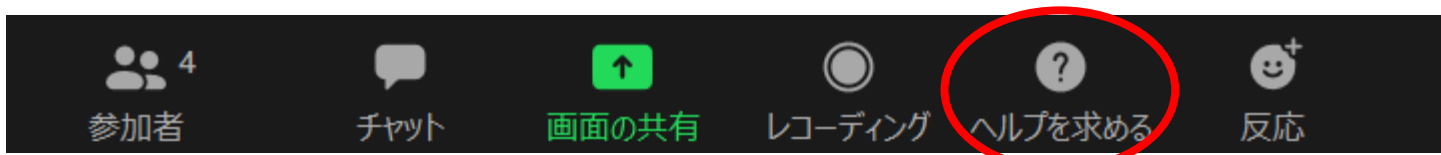
## ■ 様々なツール連携が可能。

## ■ こういうのを実現したいということであれば伺います。

# グループディスカッション

---

- グループで話してもらいたいことは3点
  1. 自己紹介
  2. 課題について(困ったことは？ 自動化したいことは？)
  3. 次回までのTODO
- Chatworkのグループチャットに書く
- タスク機能を使ってもOK！（期限を決める場合は1/16がおすすめ）
- 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



# 本日の予定

---

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

# 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。
- 課題： 65ページ以降の内容、本日チーム内で決めたこと

# 皆様お疲れ様でした。 よいお年を！

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。

---



# 課題

---

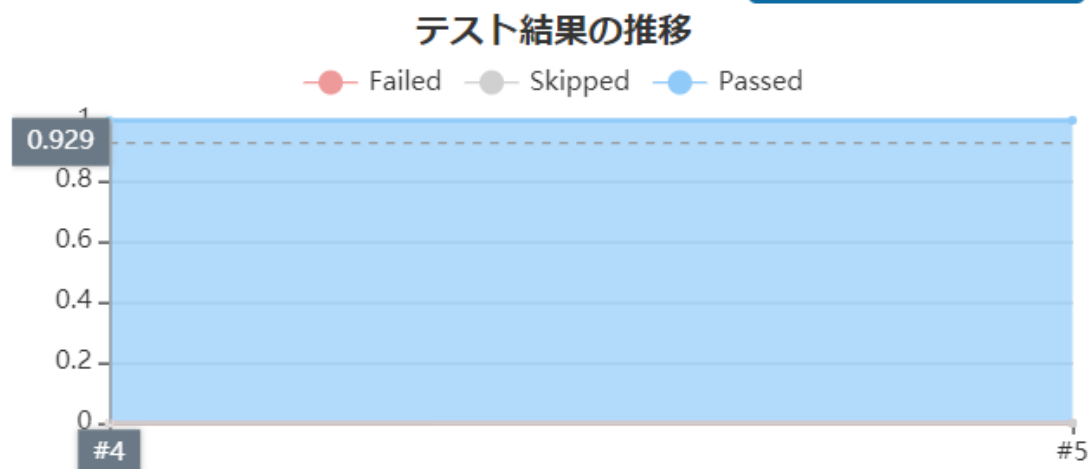
# テスト結果の表示

テストケースごとに成功失敗を視覚化できる。

- ・JUnit形式で出力
- ・テストケースの実行結果は「成功」「失敗」のみ  
(どこで失敗は分らない)
- ・テスト時間もわかる

 [説明を記入](#)

プロジェクトの無効化



# テスト結果の表示

再度、設定画面を開く

The screenshot shows the Jenkins web interface for the 'MagicPodTest' project. The left sidebar contains a list of navigation 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), and 'Rename'. The '設定' link, represented by a gear icon, is circled in red. The main content area is titled 'プロジェクト MagicPodTest' and includes links for 'ワークスペース' (Workspace) and '変更履歴' (Change History). Below this is a section titled '永続リンク' (Permanent Links) with a list of build links: '最新のビルド (#2), 1 分 4 秒', '最新の安定ビルド (#2), 1 分 4 秒', '最新の成功ビルド (#2), 1 分 4 秒', and '最新の完了ビルド (#2), 1 分 4 秒'. At the bottom, there are links for 'Atom feed 全ビルド' and 'Atom feed 失敗ビルド'.

Jenkins > MagicPodTest >

プロジェクト MagicPodTest

ワークスペース

変更履歴

永続リンク

- 最新のビルド (#2), 1 分 4 秒
- 最新の安定ビルド (#2), 1 分 4 秒
- 最新の成功ビルド (#2), 1 分 4 秒
- 最新の完了ビルド (#2), 1 分 4 秒

ビルド履歴

ビルド番号	時刻
#2	2020/11/27 19:13
#1	2020/11/27 19:05

Atom feed 全ビルド Atom feed 失敗ビルド

# テスト結果の表示

ビルド後の処理の項目（設定画面の一番下）



# テスト結果の表示

XMLファイルを読み取たいため、「\*.xml」を入力

**JUnitテスト結果の集計** X ?

テスト結果XML \*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 1.0 ?

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 ?

Skip publishing checks ☐ If unchecked, then issues will be published to SCM provider platforms ?

Checks name Tests ?

保存 Apply

保存

# テスト結果の表示

The screenshot shows the Jenkins web interface for the project 'MagicPodTest'. The sidebar on the left contains several links, with 'ビルド実行' (Build) circled in red. The main area displays the project name 'プロジェクト MagicPodTest' and a 'ビルド実行' button. Below this is a '永続リンク' (Permanent Link) section. At the bottom, there is a 'ビルド履歴' (Build History) table with a search bar and a list of builds. The build history table has a red circle around the first row, which is build #3. A red box with Japanese text is overlaid on the right side of the build history table.

ビルド実行

成功しても失敗しても  
とりあえず結果を見ましょう

find	
#3	2020/11/27 19:28
#2	2020/11/27 19:13
#1	2020/11/27 19:05

# テスト結果の表示

「コンソール出力」を押す



The screenshot displays the Jenkins web interface for a specific build. At the top, the Jenkins logo and name are visible on the left, and a search bar is on the right. Below the header, the breadcrumb navigation shows 'Jenkins > MagicPodTest > #3'. On the left sidebar, a list of links is provided: 'プロジェクトへ戻る' (Back to project), '状態' (Status), '変更履歴' (Change history), 'コンソール出力' (Console output), '説明の編集' (Edit description), 'Delete build '#3'', 'テスト結果' (Test results), and '前のビルド' (Previous build). The 'コンソール出力' link is circled in red. The main content area features a large blue sphere icon followed by the text 'ビルド #3 (2020/11/27 19:28:56)'. Below this, three status messages are shown: '変更点はありません。' (No changes), 'ユーザー hayato\_senda が実行' (Executed by user hayato\_senda), and 'テスト結果 (全て成功)' (Test results (all successful)).

# テスト結果の表示



Jenkins > MagicPodTest > #3

プロジェクトへ戻る  
状態  
変更履歴  
コンソール出力  
プレーンテキスト表示  
説明の編集  
Delete build '#3'  
テスト結果  
前のビルド

## 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\MagicPodTest  
[MagicPodTest] \$ powershell.exe -NonInteractive -ExecutionPolicy Bypass -File C:\Windows\TEMP\jenkins1451471636980526165.ps1

Starting ChromeDriver 87.0.4280.20 (c89e81631faa0b2a448e658c0dbd8311fb04ddbd-refs/branch-heads/4280@{#355}) on port 4723

Only local connections are allowed.

Please see <https://chromedriver.chromium.org/security-considerations> for suggestions on keeping ChromeDriver safe.

一番下までスクロール

Sending test screen captures... 1 more captures needs to be sent  
Test result: [https://magic-pod.com/JSP/technical\\_study\\_group/batch-run/4/](https://magic-pod.com/JSP/technical_study_group/batch-run/4/)  
[info - 2020-11-27 19:28:14 - C:\Users\jsp\AppData\Local\magic\_pod\_desktop\app-0.66.0\resources\app.asar\lib] driver.stop

テスト結果を保存中  
[Checks API] No suitable checks publisher found.  
Finished

↑テスト結果が保存されている！

確認出来たら、ジョブの画面に戻る



# テスト結果の表示

テスト結果が表示される  
テストケース1個分の結果のみ

