

2022/12 事前セットアップ

Infra Workshopに向けて以下を事前に準備してください。

1. VirtualBox + minikube + kubectlコマンド
2. Docker Desktop
3. DockerHubのアカウント
4. GitHubのアカウント+git コマンド

1. minikube+kubectl

・ VirtualBoxは以下からダウンロードしインストール

<https://www.virtualbox.org/>

・ minikube は以下のドキュメントを参照しインストール (1. Installationを参照)

<https://minikube.sigs.k8s.io/docs/start/>

・ kubectl は以下のドキュメントを参照しインストール

<https://kubernetes.io/ja/docs/tasks/tools/install-kubectl/>

上記3つのインストールが完了したら、以下のコマンドでminikubeの起動とアクセスを確認してください。

```
$ minikube start --kubernetes-version v1.25.3 --vm-driver virtualbox
```

起動に成功すると以下が表示されます。



終了しました！kubectl がデフォルトで「minikube」クラスターと「default」ネームスペースを使用するよう設定されました

kubectl コマンドでKubernetesのコンポーネントが起動しているかを確認して下さい。

```
$ kubectl get pod -n kube-system
```

以下のように全て"Running"となっていれば成功です。

NAME	READY	STATUS	RESTARTS
AGE			
coredns-565d847f94-rsgff	1/1	Running	3 (7d18h ago)
13d			
etcd-minikube	1/1	Running	3 (7d18h ago)
13d			
kube-apiserver-minikube	1/1	Running	3 (7d18h ago)
13d			
kube-controller-manager-minikube	1/1	Running	3 (7d18h ago)
13d			
kube-proxy-gf46c	1/1	Running	3 (7d18h ago)
13d			

kube-scheduler-minikube	1/1	Running	3 (7d18h ago)
13d			
storage-provisioner	1/1	Running	7 (2m25s ago)
13d			

2. Docker Desktop

・ Docker Desktop は以下からダウンロードしインストール
<https://www.docker.com/products/docker-desktop/>

インストールが完了したら、以下のコマンドで動作を確認します。

```
$ docker run hello-world
```

以下のようにコンテナの実行結果が返って来れば成功です。

```
Hello from Docker!  
This message shows that your installation appears to be working  
correctly.
```

```
To generate this message, Docker took the following steps:
```

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.

```
To try something more ambitious, you can run an Ubuntu container  
with:
```

```
$ docker run -it ubuntu bash
```

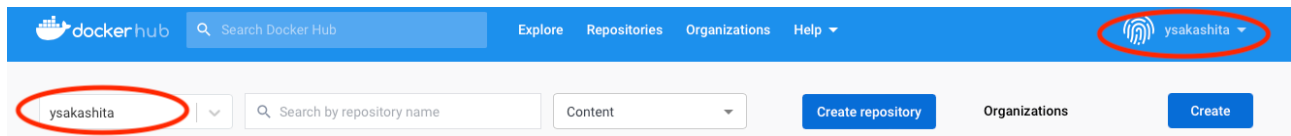
```
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/
```

```
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

3. DockerHubのアカウント

Webブラウザで <https://hub.docker.com/> にアクセスし、"Register"からアカウントを作成します。

アカウントを作成後、ログインし以下のように自分のアカウントになっていれば成功です。



4. GitHubのアカウント+git コマンド

Webブラウザで <https://github.com/> にアクセスし、アカウントを作成します。

以下のQiitaを参照しアカウントを作成してください。

<https://qiita.com/ayatokura/items/9eabb7ae20752e6dc79d>

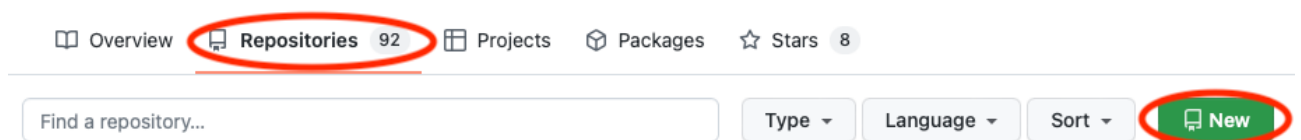
git コマンドは、OS標準でインストールされている場合がありますが、インストールされていない場合は、以下を参照しgit コマンドをインストールしてください。

<https://www.sejuku.net/blog/73444>

gitコマンドまでインストールが完了したら動作確認を行います。

Webブラウザで<https://github.com/> にアクセスしログインします。

“Repositories”->“New”をクリックし、新規のレポジトリを作成します。



以下の例では"sandbox"という名前のレポジトリを作成しています。

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *

 ysakashita ▾

Repository name *

sandbox ✓

Great repository names are short and memorable. Need inspiration? How about [fluffy-succotash](#)?

Description (optional)

☐ Public

Anyone on the internet can see this repository. You choose who can commit.

☒ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore


Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

License: None ▾


This will set  main as the default branch. Change the default name in your [settings](#).


 You are creating a private repository in your personal account.


Create repository

作成したレポジトリのページを開き、"code"をクリックします。

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


 main ▾


 1 branch

 0 tags


Go to file


Add file ▾

 Code ▾

 ysakashita Initial commit

61ab591 1 minute ago


 1 commit

 README.md

Initial commit

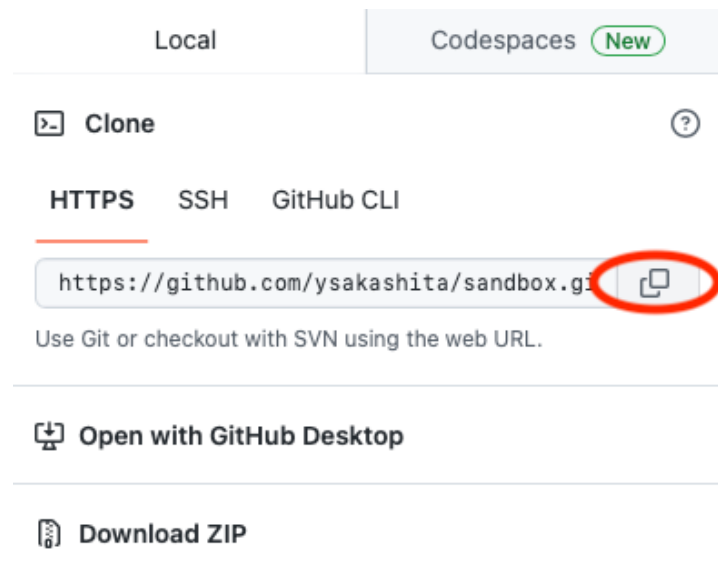
1 minute ago

README.md



sandbox

URIの右側のコピペのアイコンをクリックし、クリップボードへコピーします。



コンソールでgit cloneコマンドを実行します。(URIは上記でクリップボードにコピーされたものを使います)

```
$ git clone https://github.com/ysakashita/sandbox.git
```

cloneに成功し、自分のPCにレポジトリがダウンロード出来ていれば成功です。

```
$ cd sandbox/
```

次に、git へ pushできるかも確認します。

README.md ファイルを編集します。

```
$ vi README.md
```

編集したファイルをコミットしGitHubへpush(アップロード)します。

```
$ git add .  
$ git commit -m "test"  
$ git push origin main
```

コマンドが成功した後、WebブラウザでGitHubのレポジトリへアクセスし、README.mdファイルを確認します。編集した内容が表示されていれば成功です。

以上。