



# レポジトリの説明

**Riku Mochizuki**

moz at sfc.keio.ac.jp

# ディレクトリ構成

- `docs/` : プロジェクトのドキュメントを含むディレクトリです。プロジェクトの概要や使用方法などが記述されています。
- `raft/` : プロジェクトのソースコードを含むディレクトリです。ここには、メインの実行ファイルやユーティリティ関数などが含まれます。

# 主要なファイル構成

- `run_node.py` : ノードを起動するためのスクリプトファイル
- `Dockerfile` : Dockerイメージをビルドするための設定ファイル。
- `docker-compose.yml` : 複数のDockerコンテナを定義し、ノード間のネットワークを設定するためのファイル。
- `generate_docker_compose.py` : 指定されたノード数に基づいて `docker-compose.yml` を動的に生成するPythonスクリプト。
- `raft/*.py` : Raftを実装するためのコードです。基本的に `raft/state.py` を編集することになります。

ひな形は用意してありますが、それぞれどのような実装になっているのか目を通してみよう!

## 必要な環境

- Docker
- Docker Compose
- Python 3.x (with pyyaml)

# セットアップ手順

## 1. リポジトリをクローン

```
git clone https://github.com/mzhkz/raft-on-weekends.git  
cd raft-on-weekends
```

# セットアップ手順 (続き)

## 2. 必要なライブラリをインストール

```
pip install pyyaml
```

## 3. docker-composeファイル (Raftクラスターの構成ファイル) の作成

```
python generate_docker_compose.py <ノードの数>
```

## セットアップ手順 (続き)

### 4. docker-composeを用いてクラスター（ノード群）を作成&起動

```
docker compose up -d --build
```

### 5. ある特定のノードのログ（開発ログ等）を見る場合

```
docker logs node{1~ノードの数} -f
```

## セットアップ手順 (続き)

### 6. クラスターの停止(削除含む)

```
docker compose down
```