

Docker セミナー 2021

Docker を用いたOCTOPUS での ジョブ実行方法

福井大学 情報・メディア工学講座
渡場 康弘

コンテンツ

- コンテナ技術概要
- CMC で利用可能なコンテナ技術
 - Docker とSingularity
 - Docker を用いたジョブの実行方法
- Docker 利用の申請手順

コンテンツ

- コンテナ技術概要
- CMC で利用可能なコンテナ技術
 - Docker とSingularity
 - Docker を用いたジョブの実行方法
- Docker 利用の申請手順

高性能計算機への要求の変化

- 高性能計算機利用の多様化
 - 従来: 自作プログラムによる利用
 - ⇒ OSS のような公開プログラムの活用の増加
 - 特に、ビッグデータ解析、AI関係のライブラリ、フレームワーク等において開発が活発
- 公開プログラム活用における問題
 - デバイスドライバやライブラリの特定のバージョンを必要とする場合がある
 - 高性能計算機環境は多くの場合、いわゆる「枯れた技術」で構成されていて最新のライブラリ等が使用できない場合がある

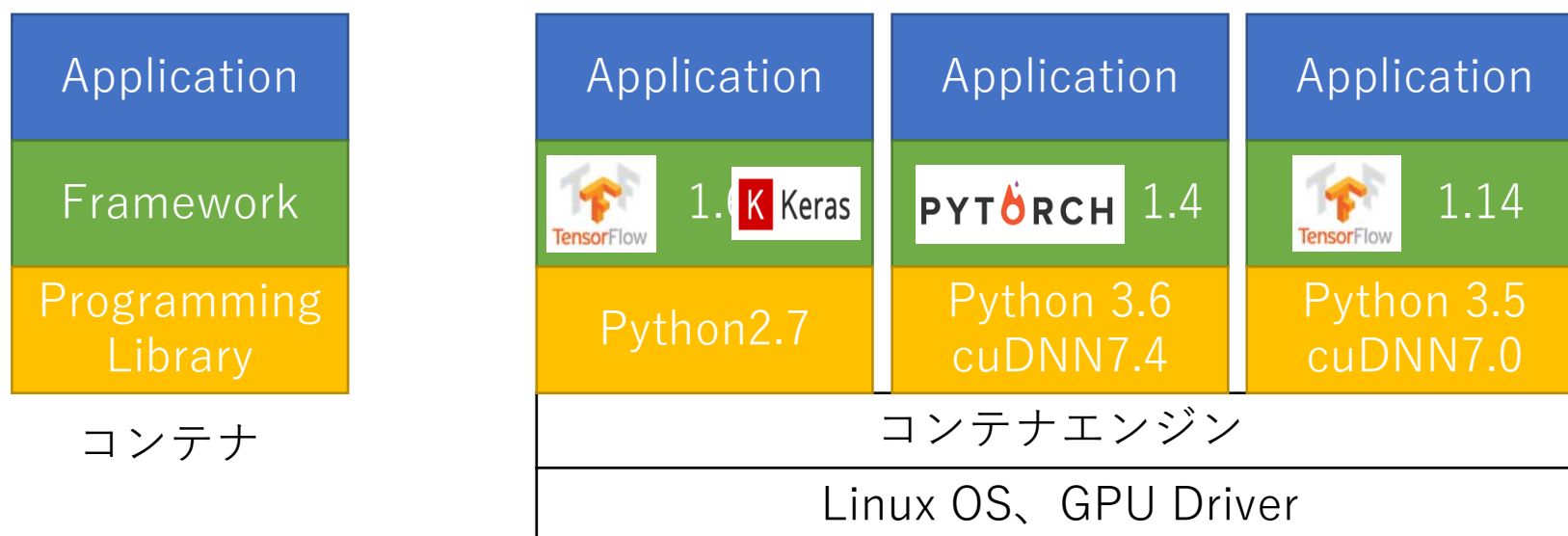
利用するソフトウェアの要件を満たした環境の構築が必要

コンテナ型仮想化技術の活用

• コンテナ利用の利点

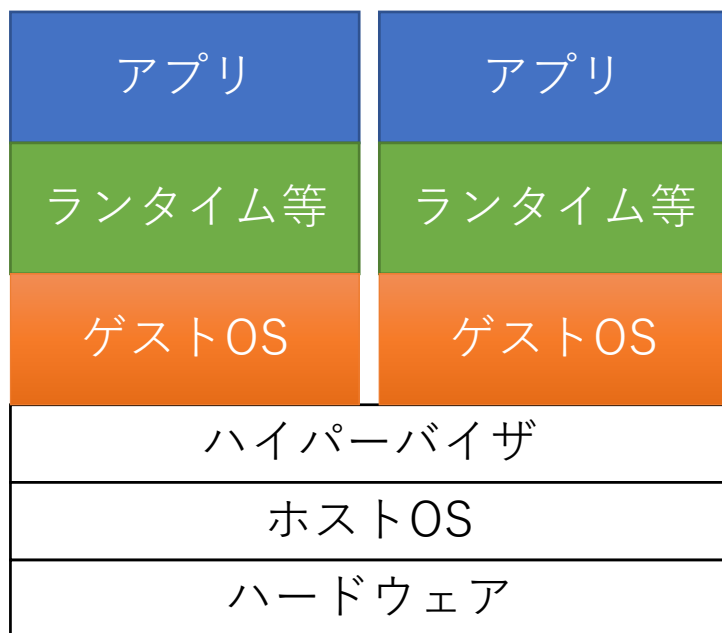
- 様々な種類やバージョンを組み合わせた、実行可能検証済みの環境が提供
- 単一環境で異なるソフトウェア環境を実行可能

⇒ ユーザは適切な環境構成を手間をかけずに利用可能

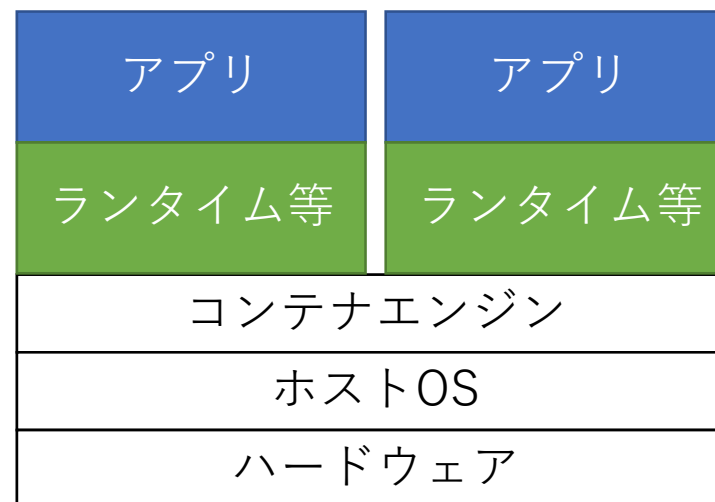


仮想マシンとコンテナの違い

- 主な違いはゲストOSの有無
 - コンテナ: ホストOSの機能を利用
=> 軽量な環境構成



仮想マシン



コンテナ

コンテンツ

- コンテナ技術概要
- **CMC で利用可能なコンテナ技術**
 - Docker とSingularity
 - Docker を用いたジョブの実行方法
- Docker 利用の申請手順

利用可能なコンテナ技術

- Docker

- 個人環境で広く利用
 - イメージやドキュメントが豊富
- 利用にroot 権限が必要



- Singularity

- HPC サービス向けコンテナ技術
- ユーザ権限でコンテナ起動が可能
- HPC系ソフトウェアスタックが利用可能



利用可能なコンテナ技術

- **Docker**

- 個人環境で広く利用
 - イメージやドキュメントが豊富
- 利用にroot 権限が必要



- Singularity

- HPC サービス向けコンテナ技術
- ユーザ権限でコンテナ起動が可能
- HPC系ソフトウェアスタックが利用可能



<http://www.hpc.cmc.osaka-u.ac.jp/system/manual/squid-use/singularity/>

各システムでの対応状況

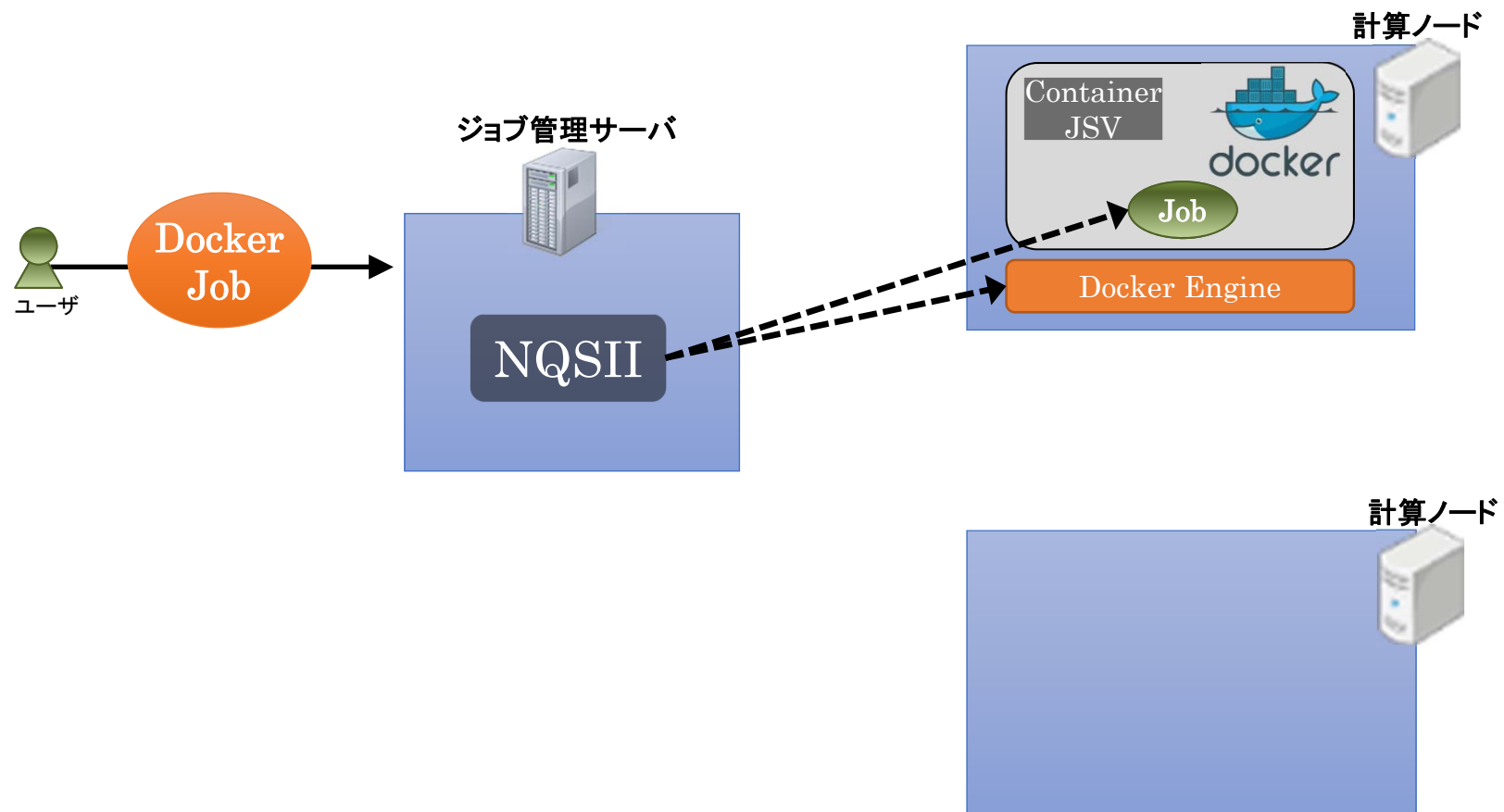
	OCTOPUS	SQUID
Docker	○ ※1	△ ※2
Singularity	△ ※3	◎

※1: 試行サービスとして導入

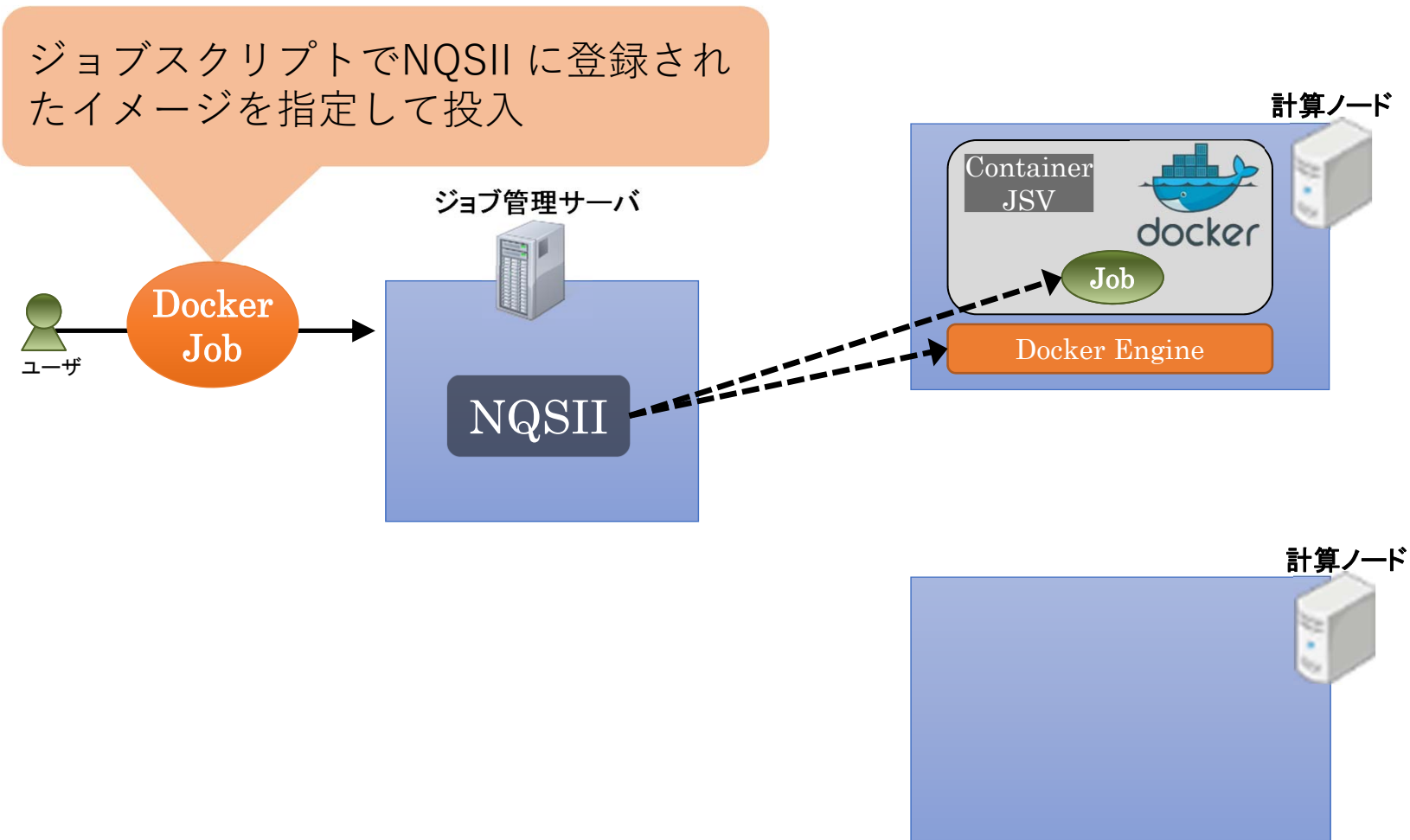
※2: サービスとして提供されていない

※3: テスト導入のため利用に制限あり

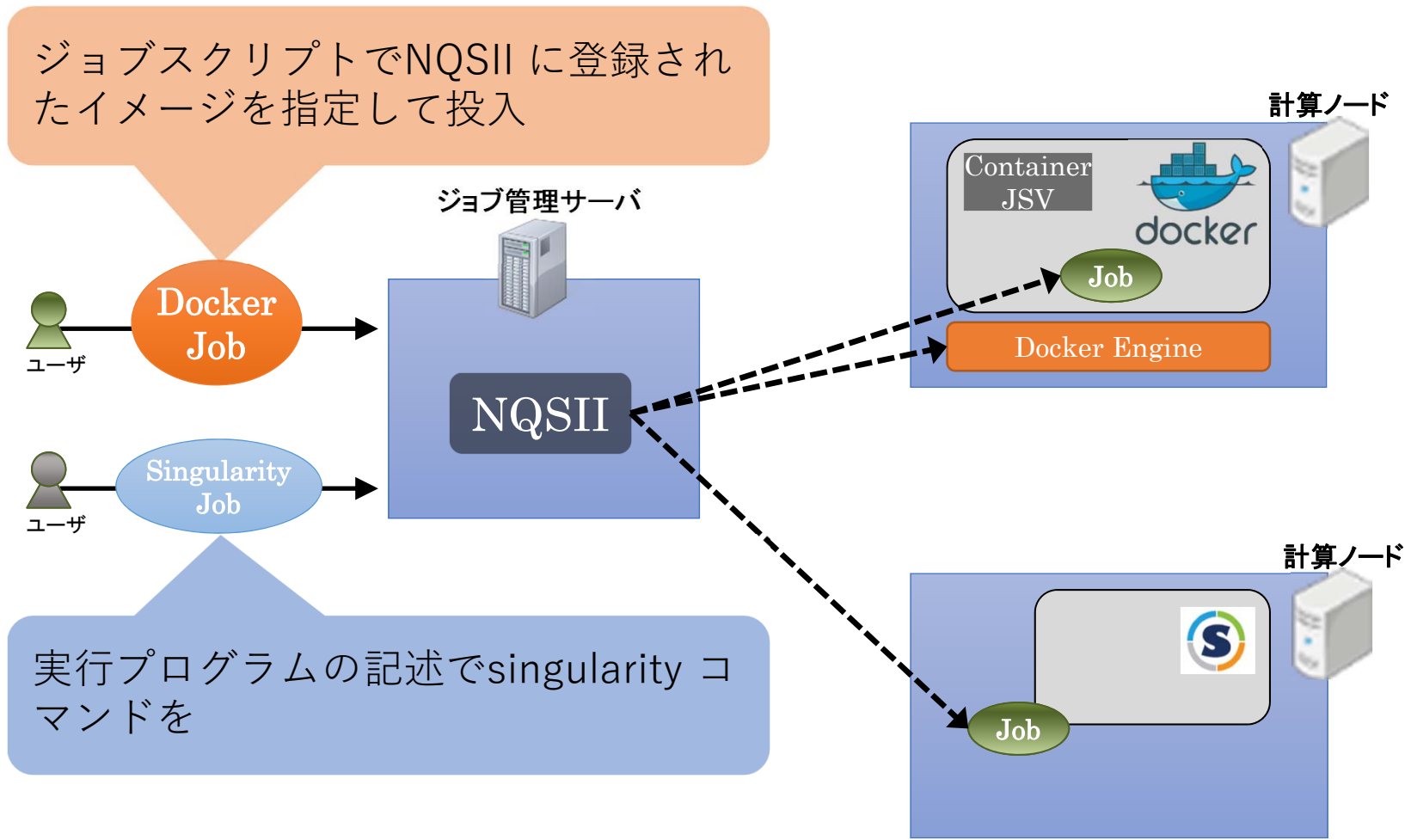
ジョブの実行形態の違い



ジョブの実行形態の違い



ジョブの実行形態の違い



ジョブスクリプトの違い

Docker 利用

```
#!/bin/bash

#----- qsub option -----
#PBS -q ODT
#PBS -l elapstim_req=00:30:00
#PBS --template=template_name

#----- Program execution -----
cd $PBS_O_WORKDIR
./a.out
```

Singularity 利用

```
#!/bin/bash

#----- qsub option -----
#PBS -q SQUID
#PBS -l elapstim_req=00:30:00

#----- Program execution -----
cd $PBS_O_WORKDIR
singularity exec image.sif ./a.out
```

ジョブスクリプトの違い

Docker 利用

```
#!/bin/bash

#----- qsub option -----
#PBS -q ODT
#PBS -l elapstim_req
#PBS --template=template_name

#----- Program execution -----
./a.out
```

Singularity 利用

```
#!/bin/bash

#----- qsub option -----
#PBS -q SQUID
#PBS -l elapstim_req

#----- Program execution -----
singularity exec image.sif ./a.out
```

Docker イメージをtemplateとして
NQSII に登録する必要がある

利用できるテンプレートの確認

- コマンド: `qstat --template`
 - テンプレート名が切れている場合は「-l」オプションを付ける

```
=====
Template  L Image      CPU  Memory GPU  Custom      Comment
-----
tensorflow - tensorflow 24 190.0G 0 (none)     TensorFlow
centos     - centos     24 190.0G 4 (none)     CentOS
nvidia    - nvidia/cud 24 190.0G 4 (none)     Nvidia-Docker2
gromacs   - oct-gromac 24 190.0G 4 (none)     nvcr gromacs 2020.2
dent-openf - dent-openf 24 190.0G 0 (none)     dent openfoam7-param*
tensorflow - oct-tensor 10 16.0G 4 (none)     tensorflow 1.14
pytorch-0. - oct-pytorc 10 16.0G 4 (none)     pytorch 0.4
pytorch-1. - oct-pytorc 10 16.0G 4 (none)     pytorch 1.4
tensorflow - oct-tensor 10 16.0G 4 (none)     tensorflow 2.7
=====
```


Docker/Singularity の選択

- 利用に関しての違い

	Docker	Singularity
ジョブの実行	template で指定する 以外は通常通り	Singularityコマンド を介しての実行
イメージの編集	不可	可能

- CMC でのコンテナ利用の選び方

- Docker が適しているユーザ
 - 同じ環境で繰り返し計算を行う
 - プログラムの実行は従来通りに行いたい
- Singularity が適しているユーザ
 - 頻繁にイメージを編集して利用する

コンテンツ

- コンテナ技術概要
- CMC で利用可能なコンテナ技術
 - Docker とSingularity
 - Docker を用いたジョブの実行方法
- **Docker 利用の申請手順**

利用までの流れ

1. `system@cmc.osaka-u.ac.jp` 宛にメールで連絡
2. テンプレートとして登録するDocker イメージについての情報提供
3. テンプレート登録作業の実施
4. 登録完了の通知後から利用開始

Docker イメージの情報連絡

- 公開イメージをそのまま利用する場合
 - イメージ名[:タグ名]をお知らせください
 - docker pull で指定する引数
 - テンプレート名は基本的にイメージ名と同じとなります
- 独自イメージを利用する場合
 - 登録するイメージを docker save コマンドでアーカイブして送付してください
 - 登録テンプレート名をお知らせください
 - 公開イメージと同じ名称だと区別できないため