

Deep Learning計算演習用 サーバシステム 利用者様向け説明資料

2020年3月25日
富士通株式会社

●目次

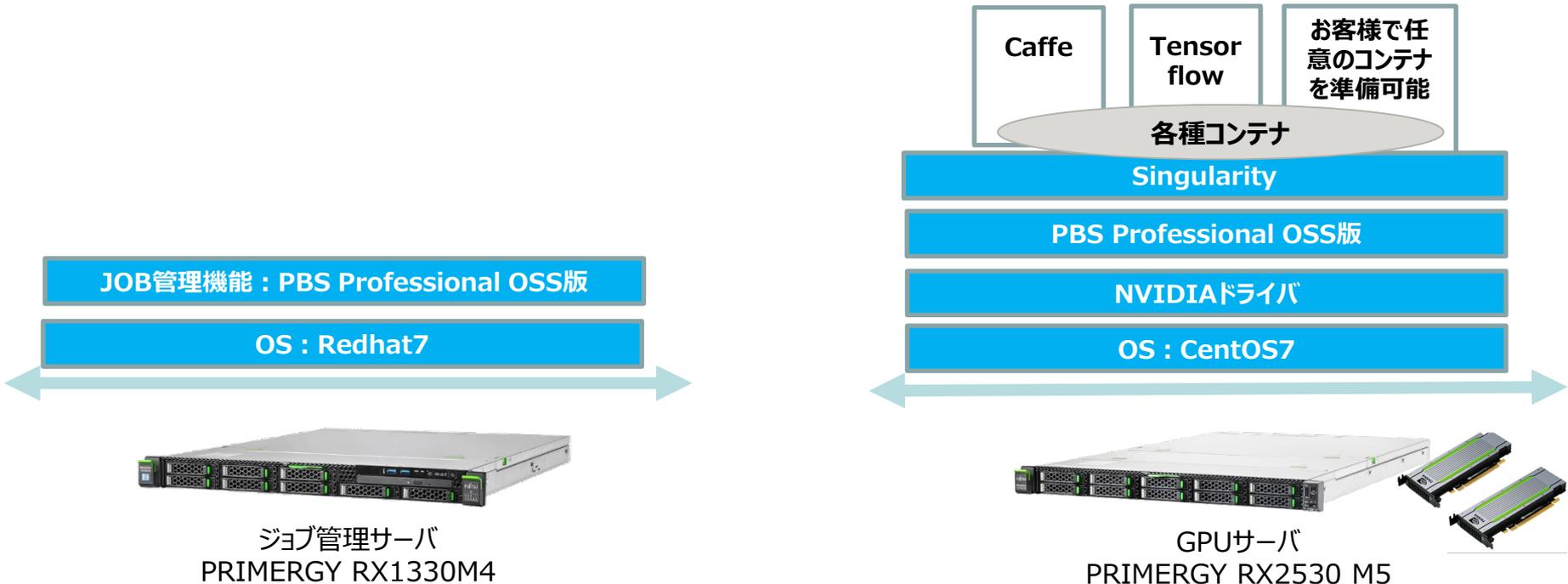
1. Deep Learningシステムの概要
 - 1-1. システムの特徴
 - 1-2. ハードウェア構成
 - 1-3. GPUサーバの概要
 - 1-4. Deep Learningシステム利用イメージ
2. Deep Learningシステムの利用方法
 - 2-1. ログイン方法
 - 2-2. JOBの投入
 - 2-3. JOBの確認
 - 2-4. JOBの取消し
 - 2-5. コンテナ利用時の留意点
 - 2-6. Singularityのコンテナのビルドと実行
 - 2-7. Nvidia GPU Cloud
 - 2-8. サンプルコンテナのダウンロードや利用方法
 - 2-9. サンプルコンテナをSingularityで実行する方法
 - 2-10. tensorflow+DIGITS
 - 2-11. nbodyコンテナのJOB起動

1. Deep Learningシステムの概要

1-1. システムの特徴

- NVIDIA社製の高性能GPU「Tesla T4」を2基搭載したGPU搭載サーバを1台提供
- 複数人での利用を考慮したジョブ管理機能を提供
- Tesla T4 スペック

Tensorコア数	CUDAコア数	演算性能 (Tensor演算)	演算性能 (半精度)	演算性能 (単精度)	最大メモリ容量	メモリ性能
320 Tensorコア	2560 CUDAコア	65 TFLOPS	16.2 TFLOPS	8.1 TFLOPS	16GB (GDDR6)	320GB/s



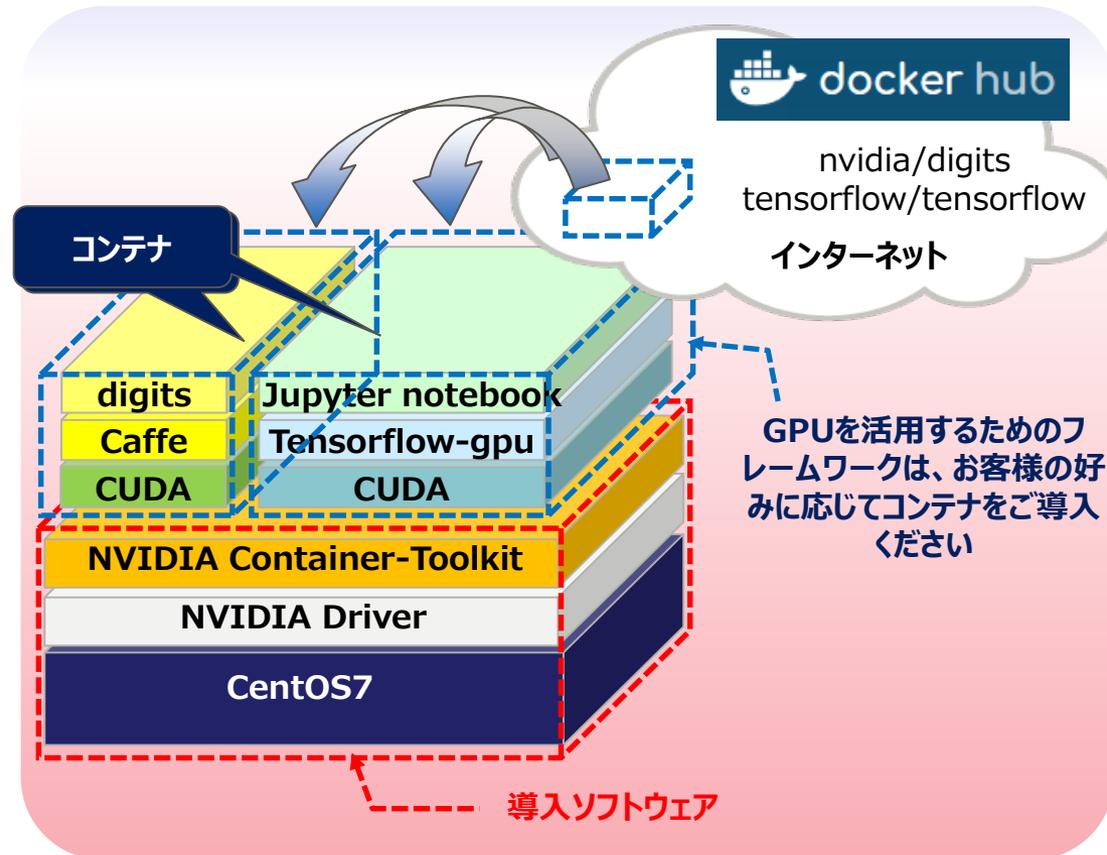
1-2. ハードウェア構成

■ ハードウェア構成

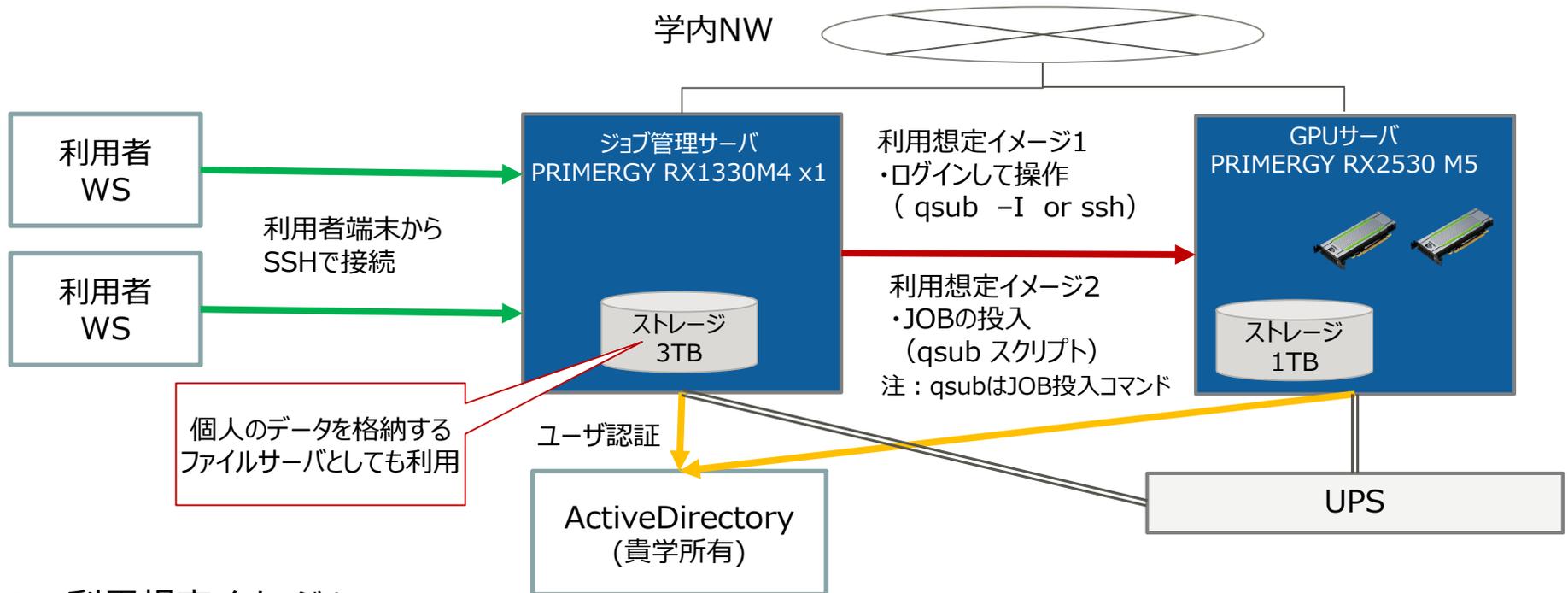
対象	ハードウェア構成	備考
GPUサーバ	PRIMERGY RX2530 M5 (NVIDIA Tesla T4 x 2搭載) x 1	電源 : 100V
	CPU : Xeon Sliver4110 (2.1GHz,8コア) x 2	
	メモリ : 96GB (メモリ-16GB x 6)	
	ハードディスク : 物理容量 1 TB (RAID1) x 2	
ジョブ管理サーバ	PRIMERGY RX1330M4 x 1	電源 : 100V
	CPU : Xeon E-2226G (3.4GHz,6コア) x 1	
	メモリ : 32GB (メモリ-8GB x 4)	
	ハードディスク : 物理容量 3 TB (RAID 5 + ホットスペア x 1) x 1	

1-3. GPUサーバの概要

- 動作検証済み環境のためすぐに利用可能
- ✓ 導入ソフトウェアは動作検証済みの組み合わせを利用することができ、お客様が試行錯誤する必要なく、すぐに利用することができます。
- ✓ Docker Hubに公開されている実績あるコンテナを導入し、すぐにディープラーニングを始められます
- ✓ GPU搭載サーバは、Ubuntu Server certified hardwareに認定されています。



1-4. Deep Learningシステム利用イメージ



➤ 利用想定イメージ1

- ① 教育用コンテナを作成し、GPU搭載サーバに配置します。
- ② ジョブ管理サーバから JOBスケジューラのInteractiveモードでGPUサーバにログインします。
- ③ ログイン後にGPUを確保していることを確認します。
- ④ 該当のコンテナをsingularityで実行します。

➤ 利用想定イメージ2

- ① ジョブスクリプトでJOB投入する場合は、JOB管理サーバから、qsubコマンドで投入します。
- ② qstatコマンドでJOBの完了を確認します。
- ③ JOBの結果を確認します。

2. Deep Learningシステムの使用方法

2-1. ログイン方法

Deep LearningシステムへログインするID・パスワードは、情報基盤センター様のPC、メール環境等にログインする時と同じものとなります。

また、接続にはTeraTerm等のSSHクライアントが必要となります。

- 作業端末でTeraTermを起動し、ジョブ管理サーバに接続します

接続先 : `yzaijs.yz.yamagata-u.ac.jp`

次ページ以降のJOBの投入やコンテナの起動を行う場合、ジョブ管理サーバのホームディレクトリヘデータ等を転送しておく必要があります。その際はWinSCP等のツールを使用して転送を行ってください。

ジョブ管理サーバへ接続後、qsubコマンドを実行しJOBの投入を行います。

利用想定イメージ1,2どちらの方法を使用するかでqsubコマンド実行時に指定するオプションが変わります。

1. 利用想定イメージ1(GPUサーバへログインして操作を行う場合)

```
$ qsub -l select=1:ncpus=2:ngpus=1 -I
```

コマンド実行時に「-I(大文字のi)」オプションを指定するとGPUサーバへのログインが行われます。
ログイン後は通常のLinuxサーバと同じように操作し処理を行います。
「exit」コマンドを実行することでJOBは終了されます。

2. 利用想定イメージ2(ジョブスクリプトを使用しジョブを自動化する場合)

```
$ qsub JobScript.sh
```

使用するジョブスクリプトを指定してqsubコマンドを実行します。
ジョブスクリプトには要求するリソースの量やJOBの投入後に実行したいコマンドを記載します。

JOBの状態はqstatコマンドで確認します。

```
$ qstat
Job id          Name          User          Time Use S Queue
-----
100.yzaijs     Job          fujitsu       00:00:00 R default_q
```

JOB投入時に出力される番号(JOB ID) から自分が投入したJOBを特定出来ます。
上記は「fujitsu」ユーザが投入したID「100」番のJOBの出力例です。
右から2つ目の項目はジョブの状態を示します。「Q」は実行待ち状態で、「R」は実行中です。

実行中のJOBだけではなく実行待ちのJOBも出力されるため、
システム全体の利用状況を確認出来ます。
処理が完了したJOBは出力結果に表示されなくなるため、
qstatコマンドの実行結果でJOBが出力されない場合はシステムが空いていると判断できます。

2-4. JOBの取消し

ジョブスケジューラで投入したJOBの状態はqdelコマンドで取消しが行えます。

JOBを2つ投入してしまった、何らかの問題でJOBの中止を行いたい場合等、JOBの正常終了を待たずにJOBを終了させたい場合に使用します。

```
$ qdel 100
```

上記はJOB ID100番のJOBの取消しを行う場合の実行例です。

2-5. コンテナ利用時の留意点

Dockerの利用にはroot資格が必要です。
Docker実行後の残dockerプロセスの再利用、回収は利用者の責任です。
Dockerの利用についてはNvidiaの各種ドキュメント、Web上の情報を参照してください。
本システムのDockerは nvidia-container-toolkit です。

Singularityの利用は一般ユーザ資格で可能ですが、
コンテナのビルド時はroot資格が必要な場合があります。
Singularityの詳細については、Sylab社のドキュメントを参照してください。
Singularityの版数は最新の**3.5.3**をセットアップしています。

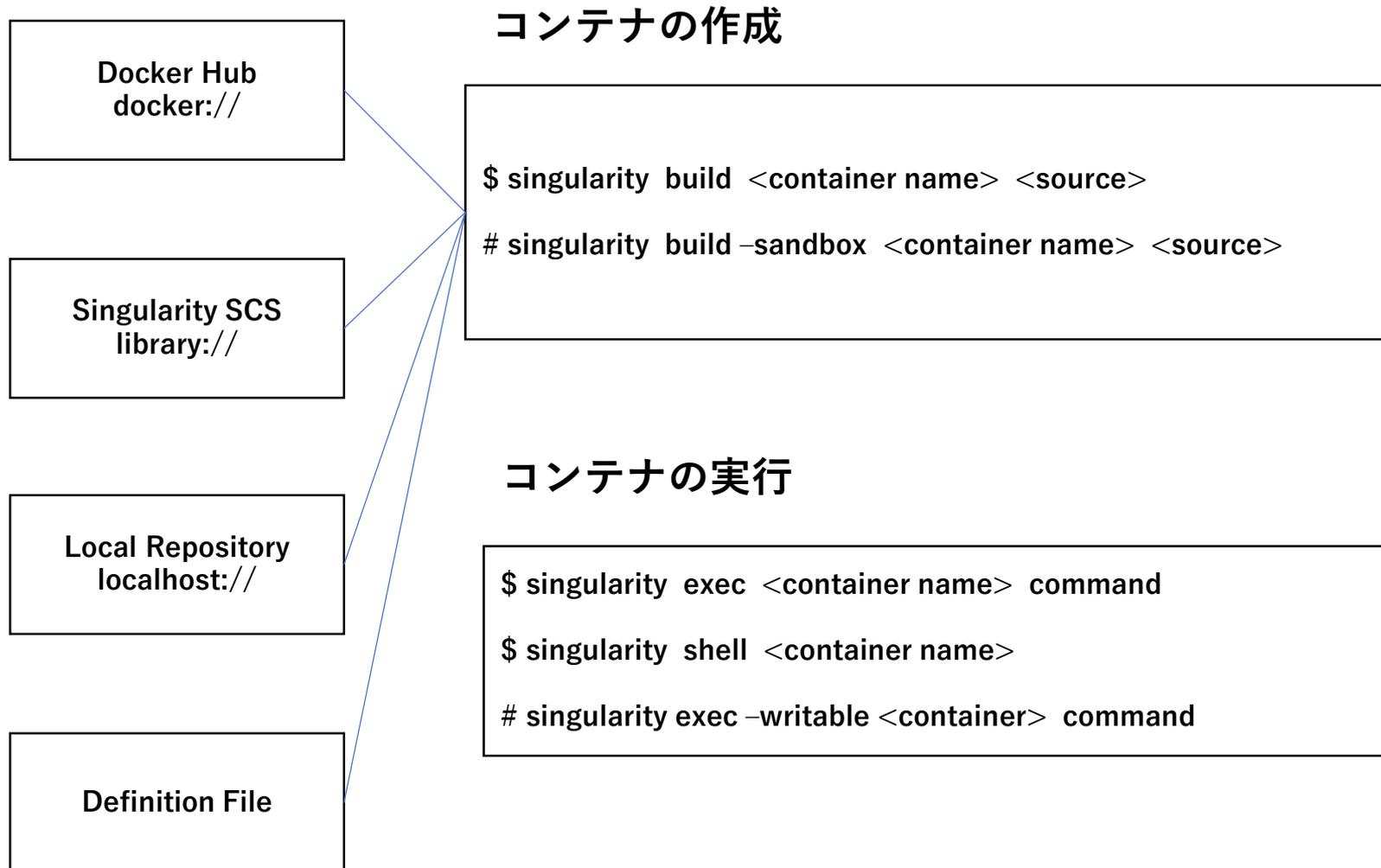
**Singularityの実行コマンドはPATHが通っていない状況です。
コマンドを以下のフルパスで記載するか、PATH環境変数を設定して実行を行います。**

```
/usr/local/singularity/bin/singularity
```

また、Singularityの利用時には環境変数等の設定が必要なため、
以下のコマンドを実行して設定を行います。
このコマンドは「.bashrc」ファイル等に記載し、ログイン時に自動実行することも可能です。

```
$ source /usr/local/singularity/etc/bash_completion.d/singularity
```

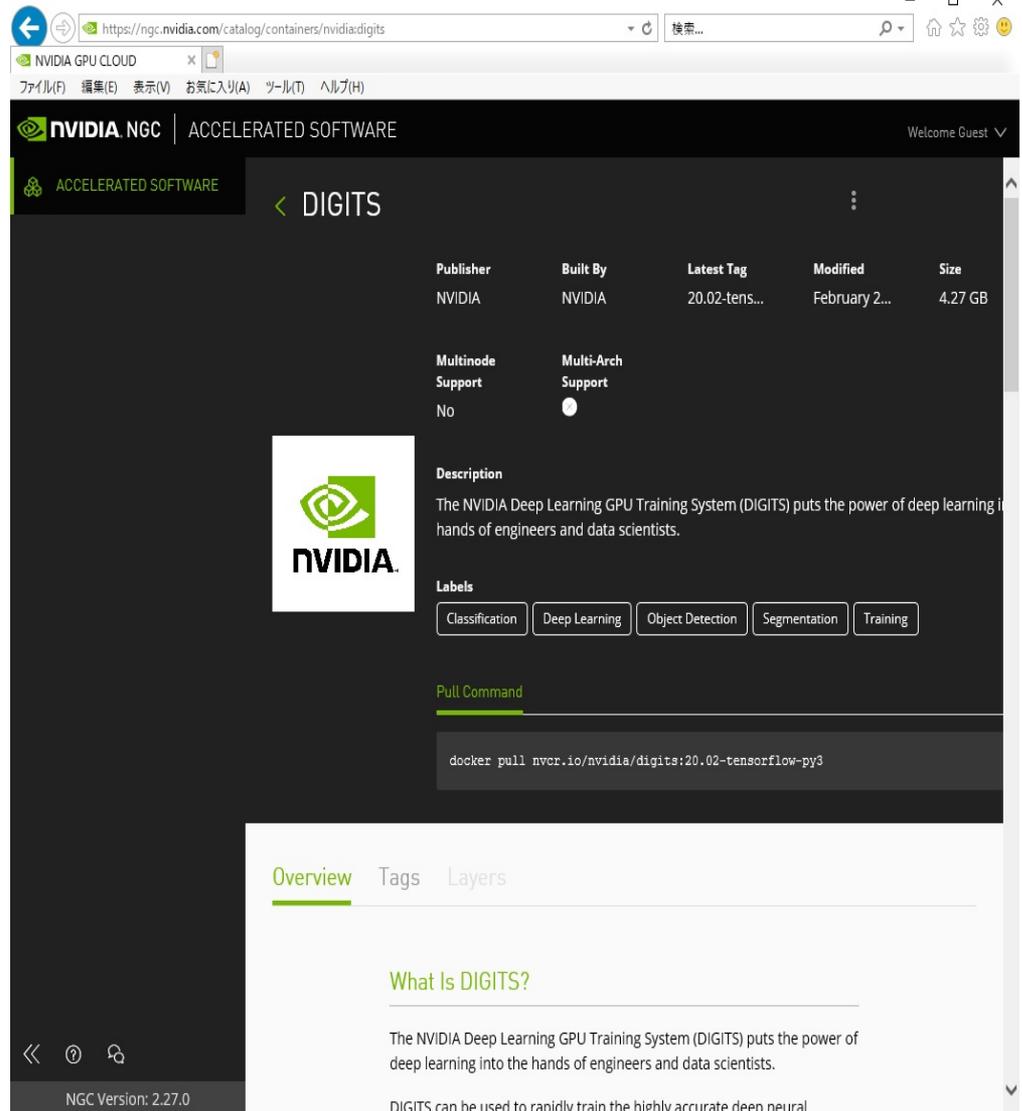
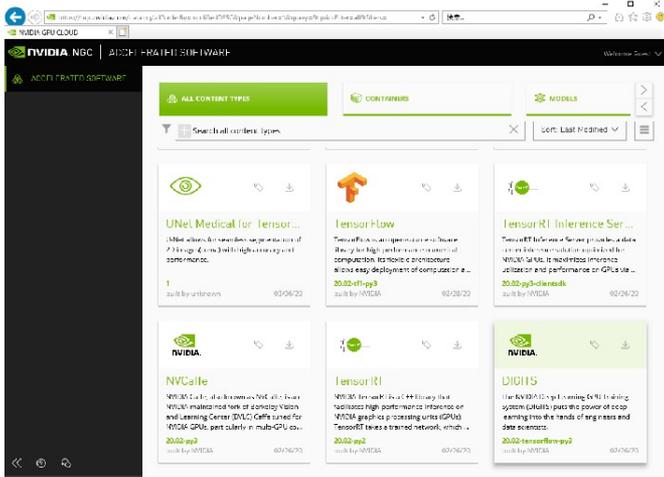
2-6. Singularityのコンテナのビルドと実行



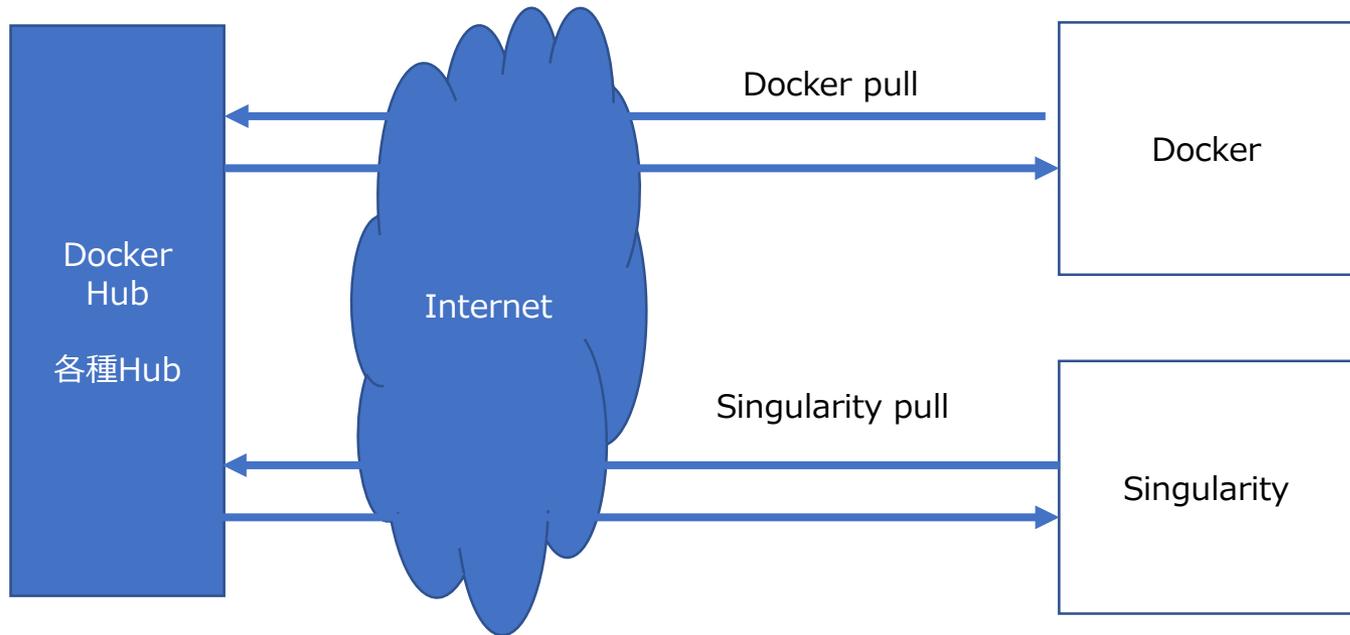
2-6. Singularityのコンテナのビルドと実行

Command	説明	使い方
build	buildを使用して、既存のコンテナをダウンロードしたり、フォーマットのコンテナを変換したり、レシピからコンテナを作成することができます。	singularity build centos7.sif library://centos singularity build centos8.simg docker://centos singularity build lolcow.simg library://sylabs-jms/testing/lolcow
exec	コンテナイメージ内の任意のコマンドの実行ができます。Docker execとは異なり、コンテナを使用する前にコンテナを実行する必要がありません	singularity exec centos7.sif cat /etc/os-release singularity exec centos8.simg cat /etc/os-release
shell	コンテナ内のシェルを実行します。	singularity shell centos7.sif singularity shell centos8.simg
run	コンテナがファイル名で直接実行または実行されたときに実行されるカスタムアクションを定義できます。	singularity run lolcow.sif ./lolcow.sif
pull	Docker HubやSingularity Hub からコンテナのイメージを取得します	singularity pull centos7.sif library://centos singularity pull centos8.simg docker://centos

2-7. Nvidia GPU Cloud <https://ngc.nvidia.com>



2-8. サンプルコンテナのダウンロードや利用方法



Docker Hubからコンテナをダウンロードする例

(注意：Docker Hubのコンテンツは変更が多いため、実行前にパスの確認が必要となります。)

```
# docker pull nvcr.io/nvidia/digits:latest
# docker pull nvcr.io/tensorflow/tensorflow:2.0.0-gpu-py3-jupyter

$ singularity build digits-latest.simg docker://nvcr.io/nvidia/digits:latest
$ singularity build tensorflow-gpu-py3-jupyter.simg docker://nvcr.io/tensorflow/tensorflow:2.0.0-gpu-py3-jupyter
```

2-9. サンプルコンテナをSingularityで実行する方法

GPUサーバへのログイン後、以下の操作を行うことでSingularityを実行できます。

注) 以下は「fujitsu」ユーザを使用する場合の例です。読み替えて実施してください。

1. Tensorflow+Jupyterコンテナ

```
$ export SINGULARITY_NOHTTPS=true  
$ singularity run --nv tensorflow-gpu-py3-jupyter.simg
```

画面に出力されるhttpアクセスを実施します。

2. Caffe+DIGITSコンテナ

```
$ cd /home/fujitsu/work/sing (コンテナのあるDirectoryに移動)  
$ mkdir digits-jobs  
$ echo > digits.log  
$ chmod 777 digits-jobs  
$ chmod 666 digits.log  
$ export DIGITS_LOGFILE_FILENAME=/home/fujitsu/work/sing/digits/digits.log  
$ export DIGITS_JOBS_DIR=/home/fujitsu/work/sing/sing/digits-jobs  
$ singularity run --nv digits-latest.simg
```

URL:<http://GPU搭載サーバIPアドレス:5000/> でアクセスします。

2-10. tensorflow+DIGITS

GPUサーバへのログイン後、以下の操作を行うことでSingularityを実行できます。

注) 以下は「fujitsu」ユーザを使用する場合の例です。読み替えて実施してください。

```
$ cd /home/fujitsu/work/sing (コンテナのあるDirectoryに移動)
$ singularity pull digits-tensorflow.simg docker://nvcr.io/nvidia/digits:20.02-tensorflow-py3
$ cd /home/fujitsu/work/sing (コンテナのあるDirectoryに移動)
$ mkdir digits-jobs
$ echo > digits.log
$ chmod 777 digits-jobs
$ chmod 666 digits.log
$ export DIGITS_LOGFILE_FILENAME=/home/fujitsu/work/sing/digits/digits.log
$ export DIGITS_JOBS_DIR=/home/fujitsu/work/sing/sing/digits-jobs
$ singularity run --nv digits-tensorflow.simg
```

[URL:http://GPU搭載サーバIPアドレス:5000/](http://GPU搭載サーバIPアドレス:5000/) でアクセスします。

2-11. nbodyコンテナのJOB起動

以下の内容でスクリプトを作成します。

注) 以下は「fujitsu」ユーザを使用する場合の例です。読み替えて実施してください。

```
#!/bin/sh

#PBS -N nbody-job
#PBS -l walltime=01:00:00
#PBS -l select=1:ncpus=1:ngpus=1
#PBS -j oe
#PBS -m abe
#PBS -M test@example.com

/usr/bin/nvidia-smi

singularity run --nv /home/fujitsu/sing/cuda_sample.sif nbody -benchmark -numdevices=1 -numbodies=1000000
```

作成したスクリプトを指定しJOBを投入します。

JOB投入が行われるとsingularityがcuda_sample.sifを使用して起動し、nbodyが実行されます。

```
$ qsub JobScript.sh
```



FUJITSU

shaping tomorrow with you