

# ScaleShift

オンプレミス / クラウド で実現する機械学習環境

June, 2019



# ScaleShift

Docker ベース、オープンソースの **Web クライアント** アプリケーションです

- **モデル構築フェーズ**

- NGC / 自社リポジトリから機械学習 Docker イメージをワンクリックで取得
- その任意の Docker イメージを Jupyter notebook コンテナとして起動

- **モデル学習フェーズ**

- 構築に利用したライブラリごと Docker イメージに固めリポジトリへ保存
- クリックだけで Kubernetes クラスタ / Rescale へ大規模計算タスクを送信

# 基本的な動き

How does it work?

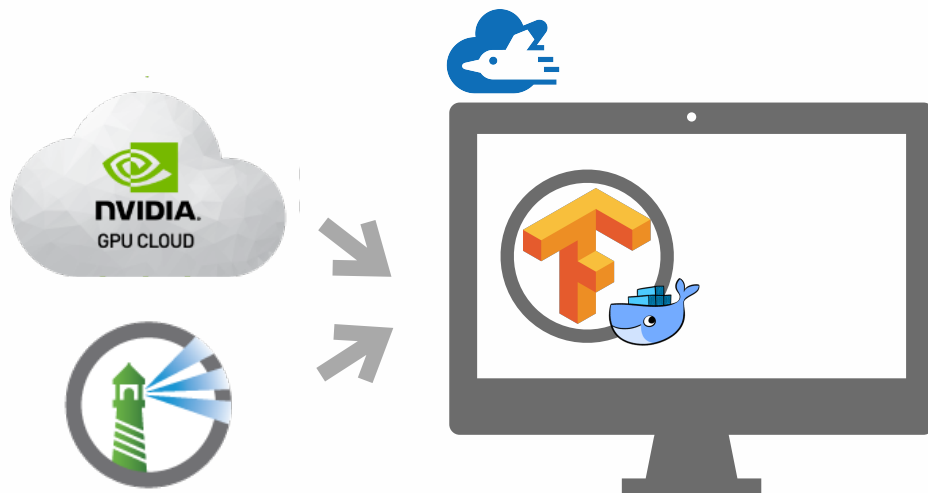
# ScaleShift の起動

ローカルに Web サーバーが立ち上がります



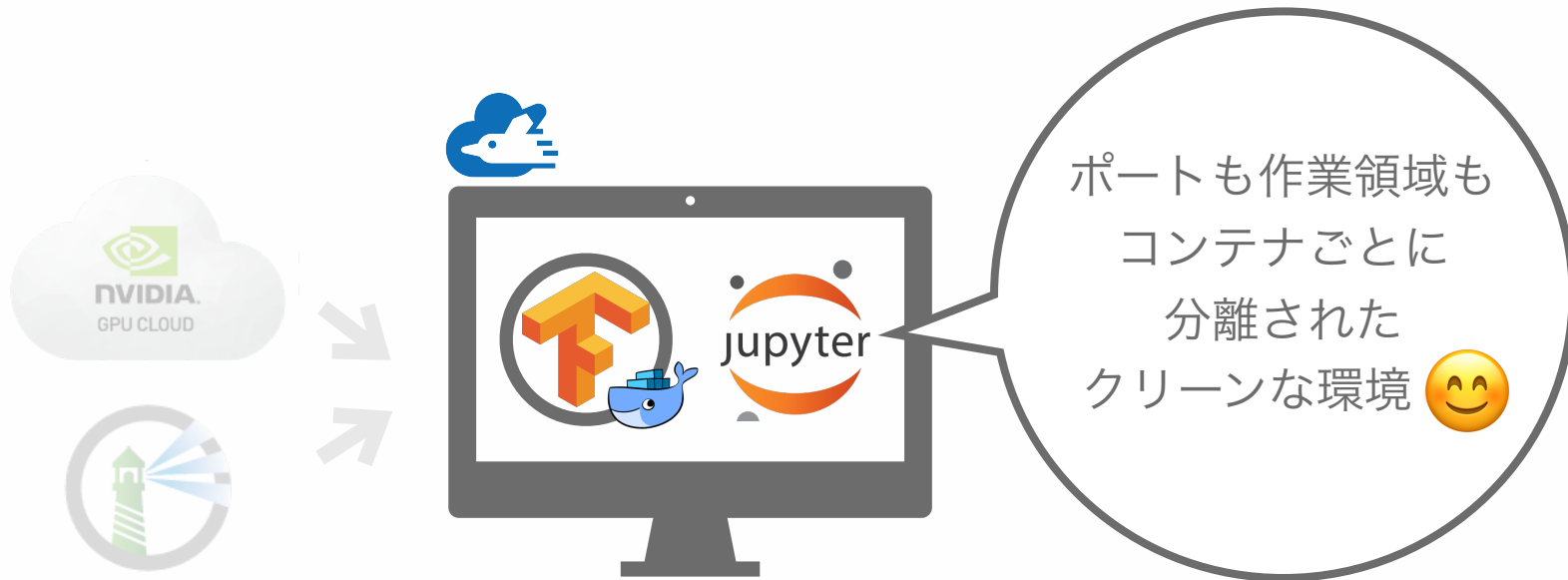
# 機械学習ソフトウェアのインストール

NGC / プライベートレジストリ からワンクリックでダウンロード



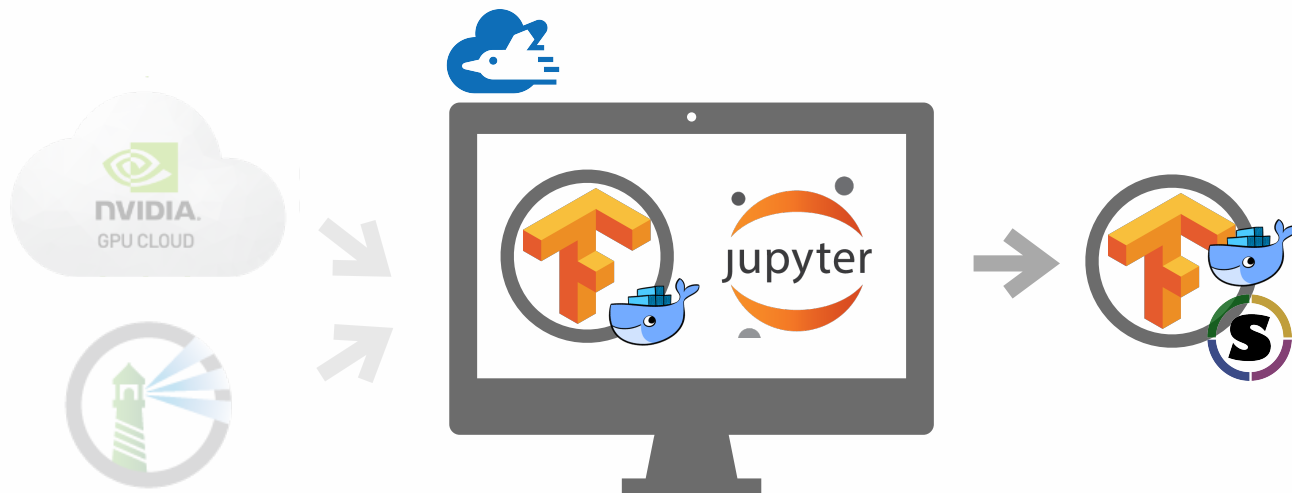
# Jupyter notebook でのモデル構築

Jupyter でラップしたコンテナがかんたんに起動



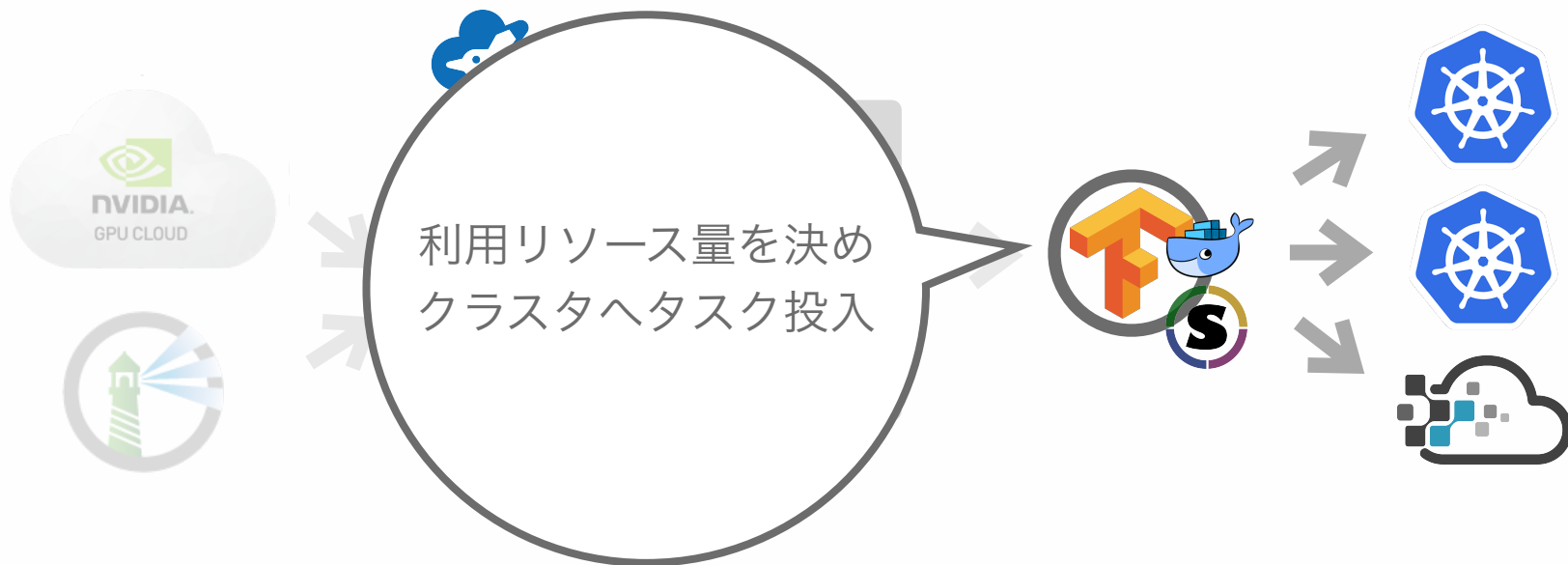
# 大規模計算のためのラッピング

依存ライブラリやソースコード群をまとめ、ひとつのイメージに固めます



# 社内クラスタ / クラウドへ計算タスク投入

投入先に応じて必要な API が実行されます





# Kubernetes 連携

Integration with a [kubernetes](#) cluster

# 機械学習 と Kubernetes

Web 界隈を中心にコンテナオーケストレーションのデファクトになった k8s。  
機械学習の文脈でもコンテナ利用が盛んになり、応用事例が増えています。

- NVIDIA が公式にサポートを表明 [ GTC 2018 Keynote, March 27 ]
- Mercari ML Ops Night Vol.1 [ 株式会社 メルカリ / May 23, 2018 ]  
<https://mercari.connpass.com/event/85931/presentation/>
- Jupyter だけで機械学習を実サービス展開できる基盤 [ 株式会社リクルートライフスタイル ]  
<https://engineer.recruit-lifestyle.co.jp/techblog/2018-10-04-ml-platform/>
- Kubernetesによる機械学習基盤への挑戦 [ 株式会社 Preferred Networks / Dec 4, 2018 ]  
<https://www.slideshare.net/pfi/kubernetes-125013757>

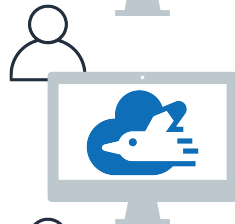
# ScaleShift + Kubernetes 構成例

NGC



DockerHub

研究 / 開発チーム

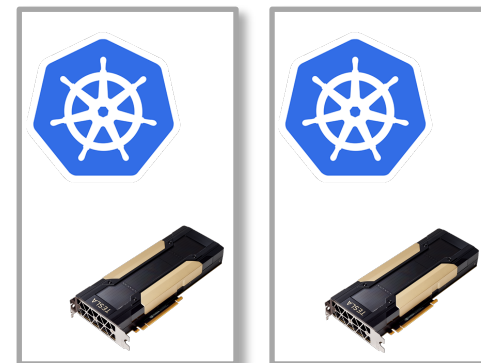


ScaleShift 入り  
ローカル端末

管理ノード



計算ノード

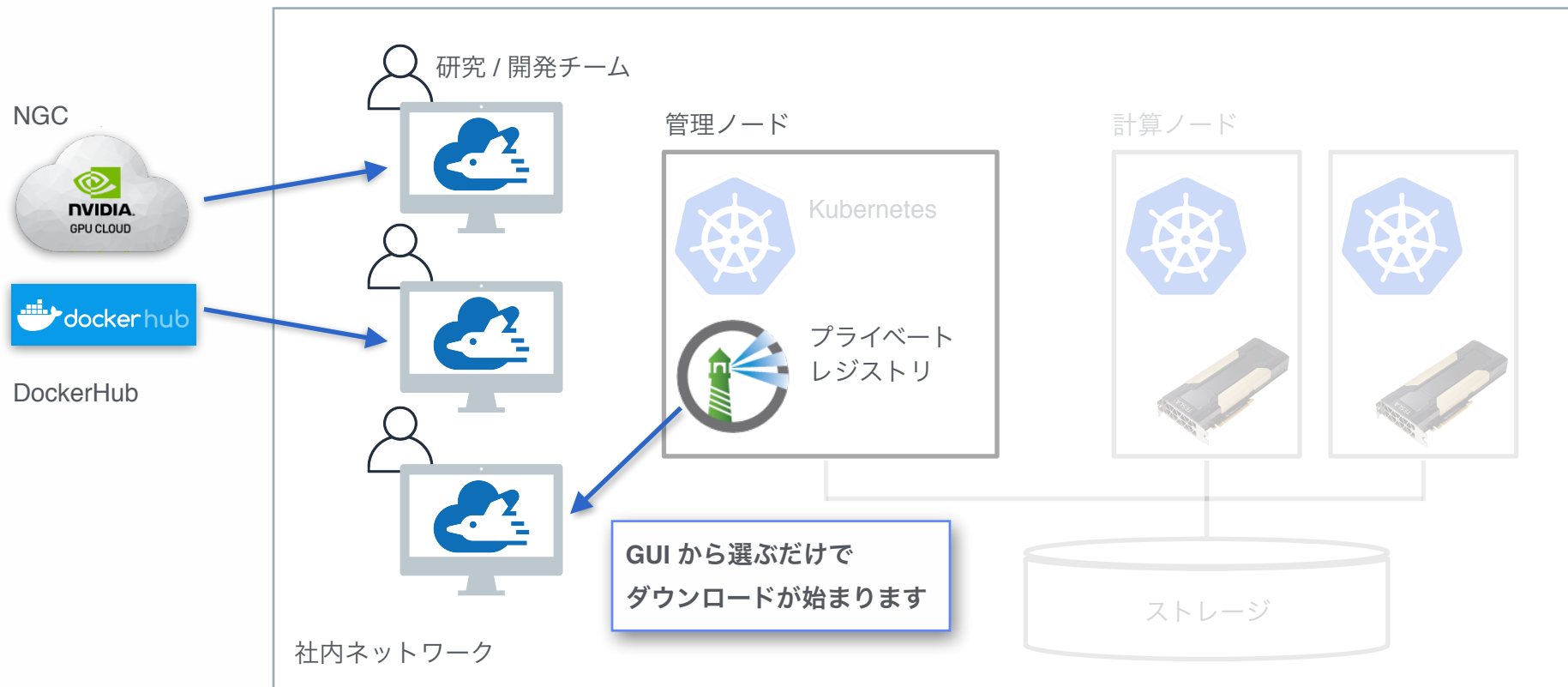


ストレージ



社内ネットワーク

# 1. 機械学習ソフトウェアの選択

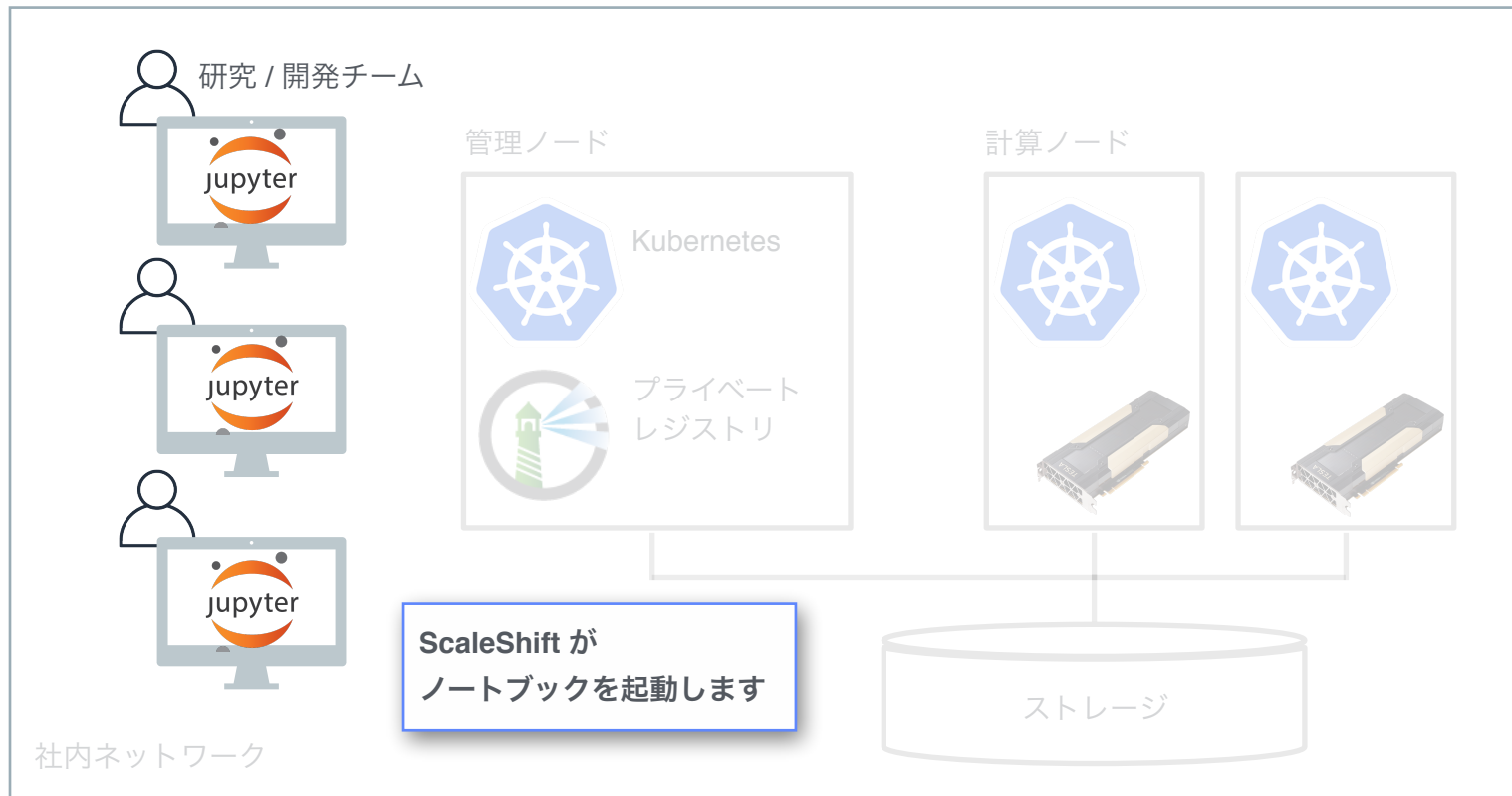


## 2. モデル構築

NGC



DockerHub

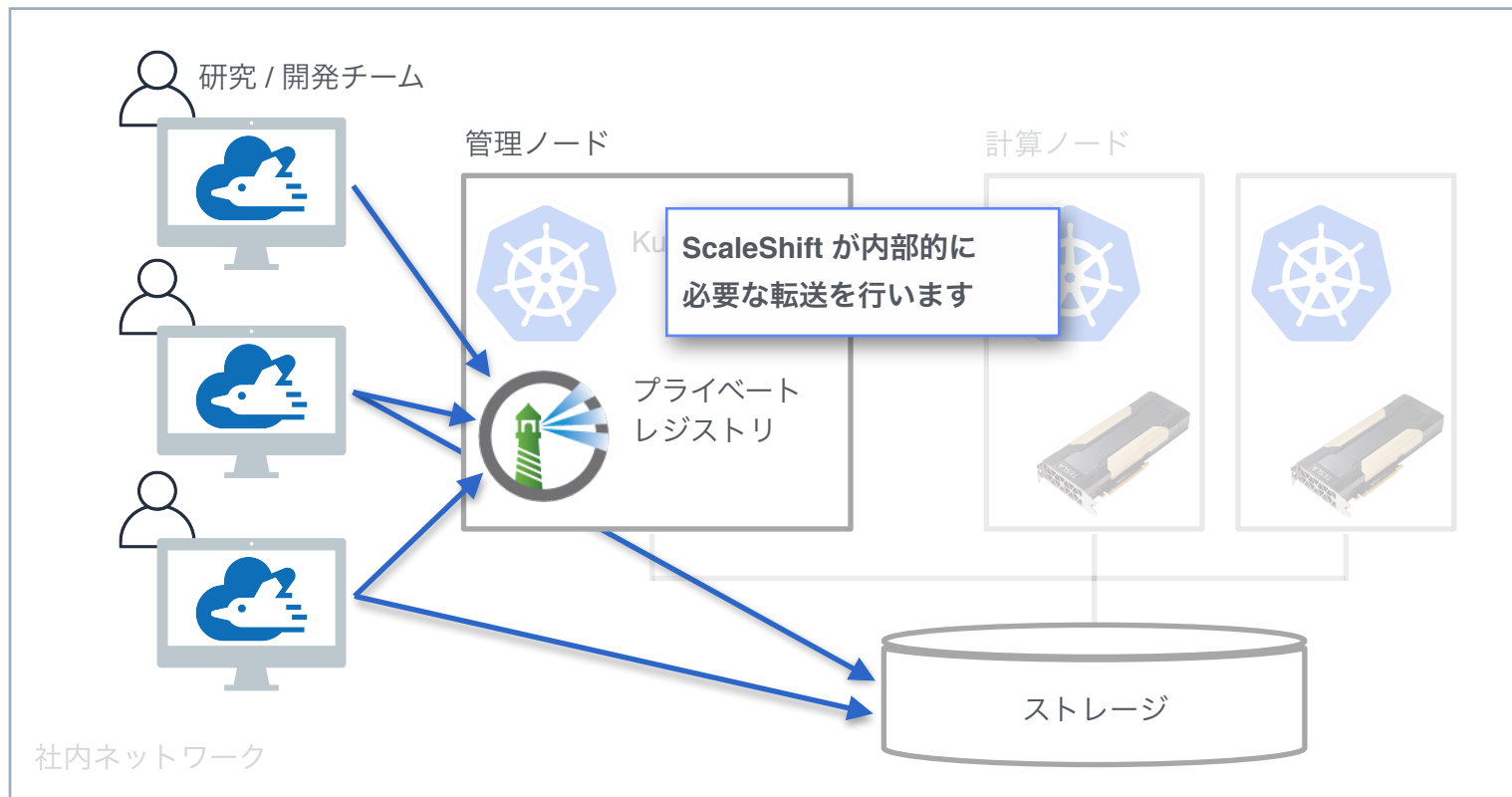


### 3. 実行環境・入力データの転送

NGC



DockerHub

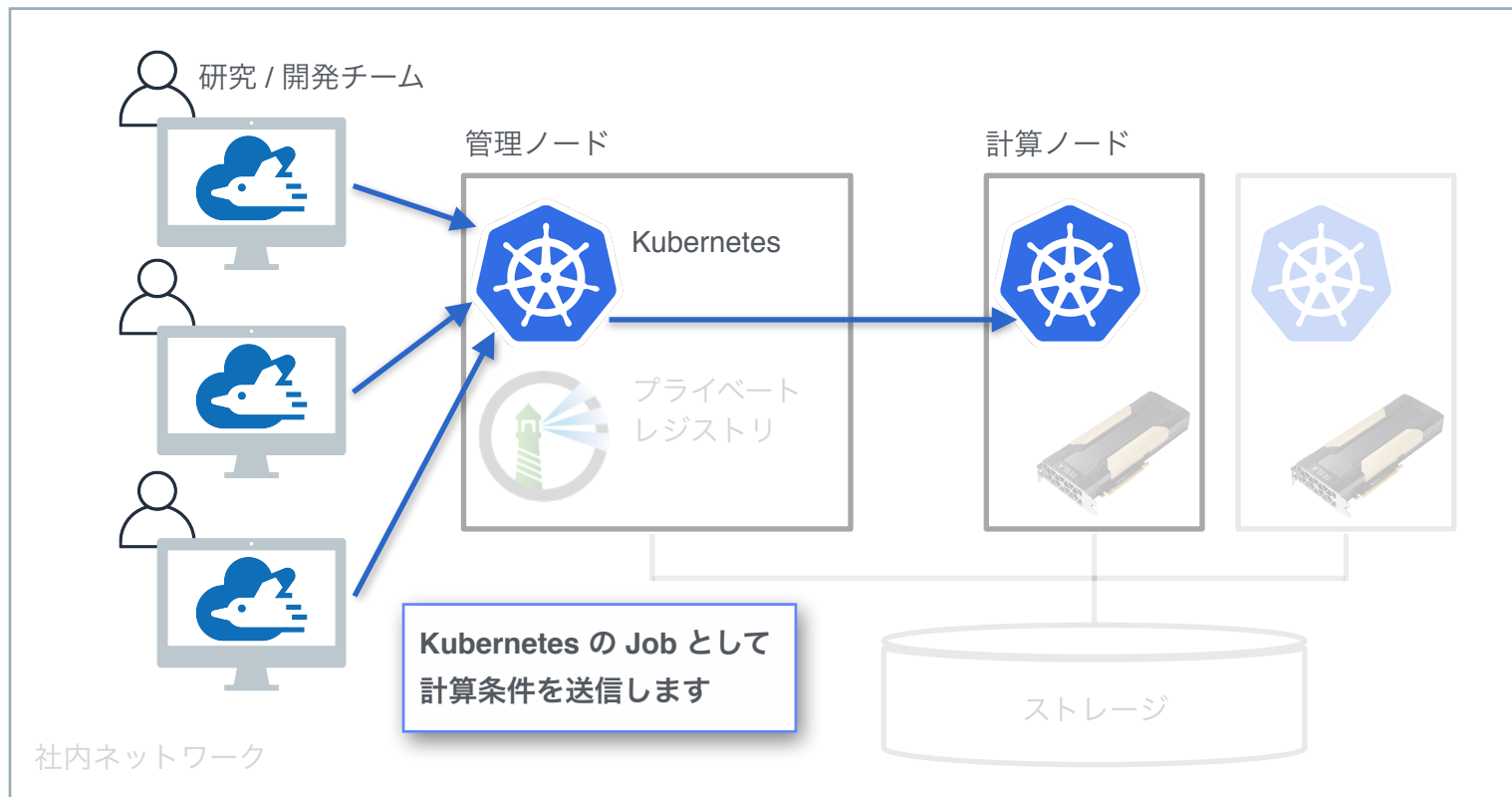


## 4. 大規模計算の実行を指示

NGC



DockerHub

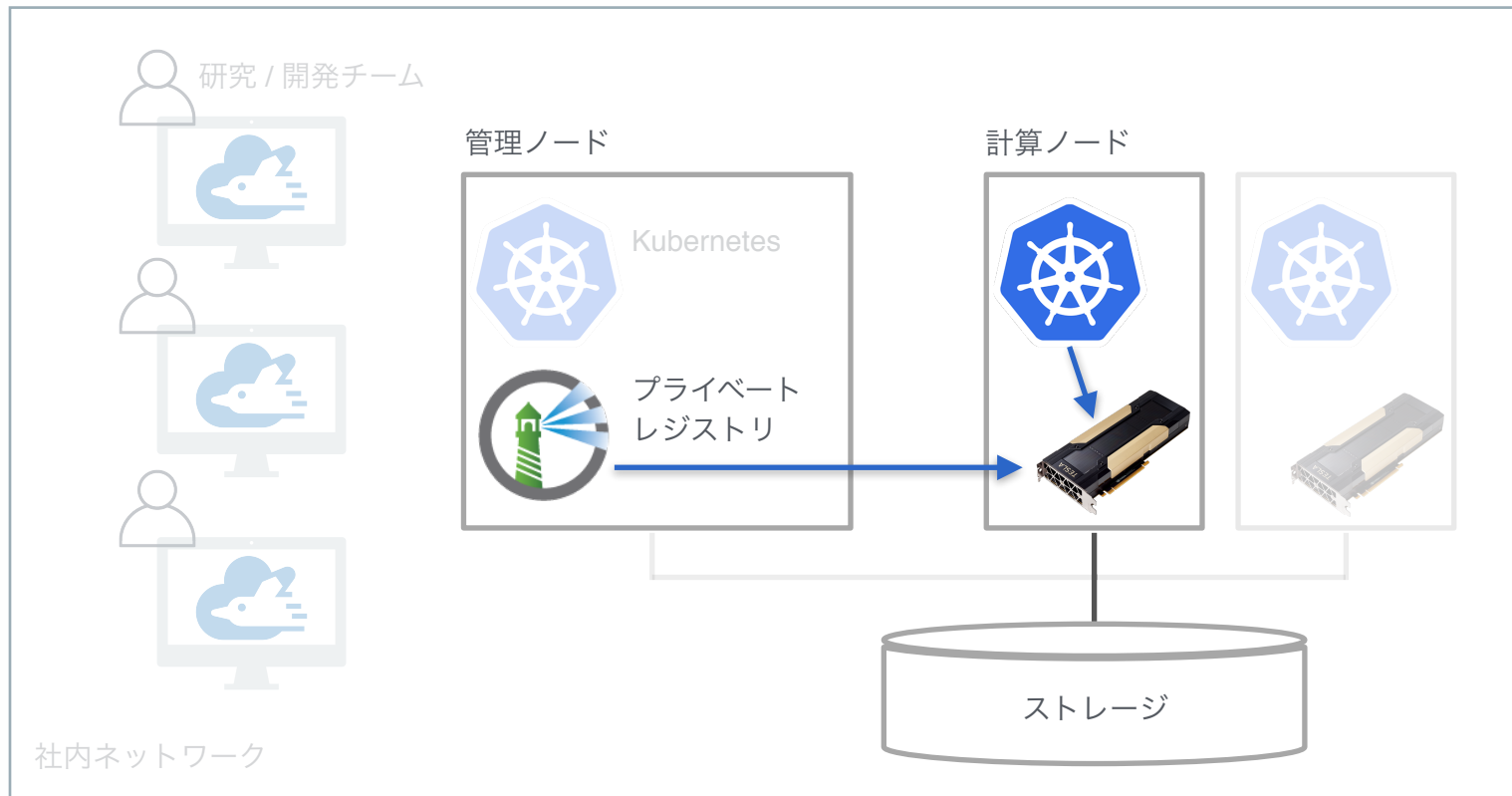


## 5. 大規模計算の実行

NGC



DockerHub



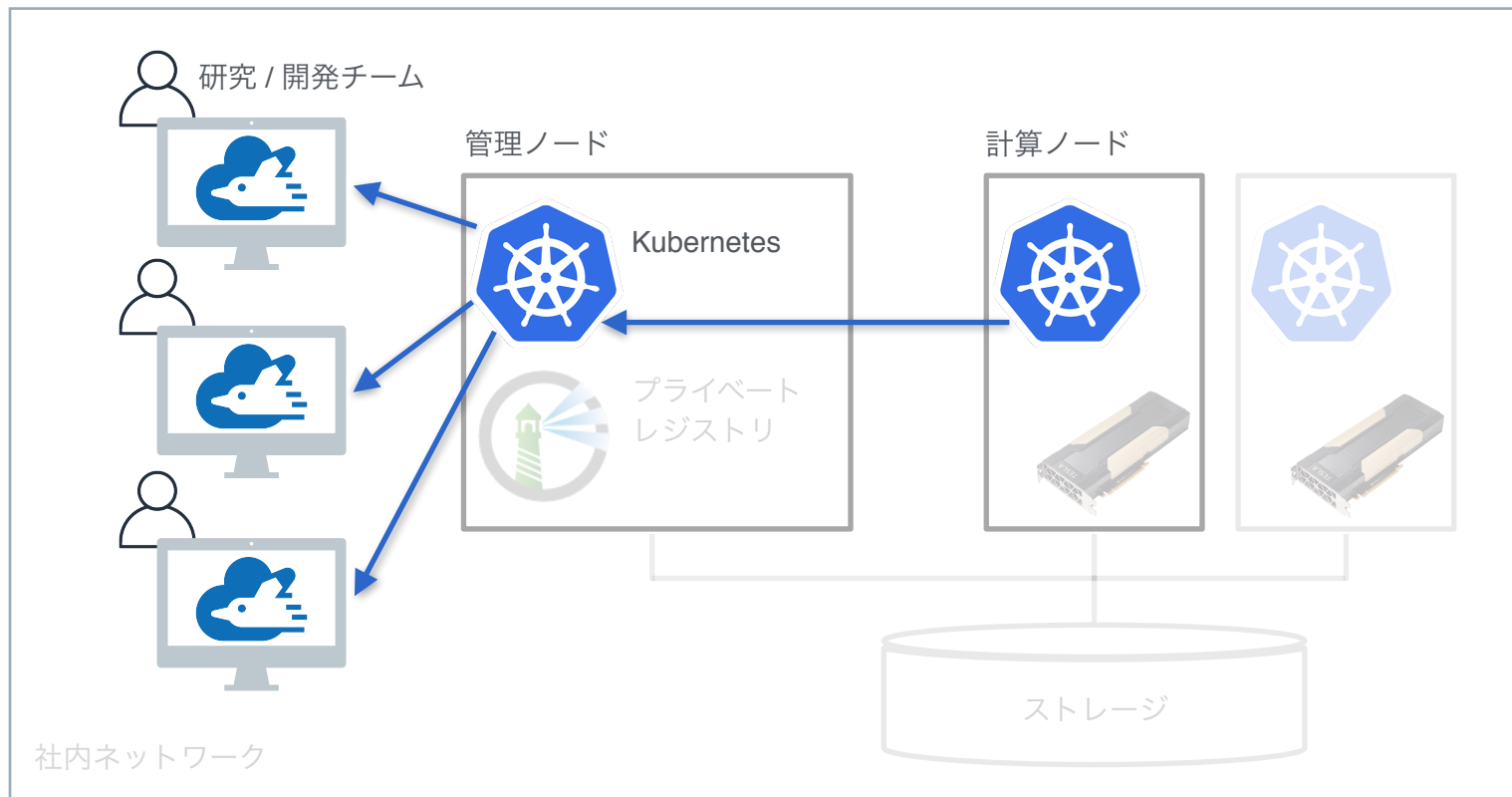


## 6. 計算結果の確認


NGC



DockerHub



# Kubernetes 設定 / タスク実行画面

 ScaleShift

Account  
admin

Navigation

Local Images

Notebooks

Workspaces

## Kubernetes

kubeconfig

```
apiVersion: v1
kind: Config
clusters:
- cluster:
    api-version: v1
    server: "https://foobar.com"
    name: "Default"
contexts:
- context:
    cluster: "Default"
    user: "Default"
    name: "Default"
current-context: "Default"
```

## Rescale

Platform

<https://platform.rescale.com>

API Key

SAVE

## Task Definition

Platform	Kubernetes
Job type	Run your iPython notebook
Notebook	tf.ipynb
Commands	python <converted-notebook.py>
CPU	1
GPU	0

It takes 5 minutes at least to start a job on cloud.

START CANCEL

# ScaleShift の設定

Configurations

# 外部連携

	連携機能	設定値
NVIDIA GPU CLOUD	<ul style="list-style-type: none"><li>• NVIDIA 社の管理する機械学習 Docker イメージの一覧 / 詳細情報取得、イメージのダウンロード</li></ul>	API キー & ユーザ設定
プライベートレジストリ	<ul style="list-style-type: none"><li>• 自社で管理する機械学習 Docker イメージの一覧情報取得、イメージのダウンロード</li></ul>	接続先 & ユーザ設定
AWS	<ul style="list-style-type: none"><li>• 機械学習 Docker イメージのダウンロード</li><li>• ローカルファイルシステムと S3 間のデータ連携</li></ul>	(実装予定)
Kubernetes	<ul style="list-style-type: none"><li>• 社内クラスタ / クラウドでの大規模計算実行</li></ul>	kubecfg
Rescale	<ul style="list-style-type: none"><li>• Rescale プラットフォームでの大規模計算実行</li></ul>	地域指定 & API キー

# 起動オプション（抜粋）

[docker-compose.yml](#) に設定を記載、起動できます

	設定概要	初期値
SS_JUPYTER_MINIMUM_PORT	コンテナへの接続ポート動的割当開始番号	30000
SS_LOG_LEVEL	アプリケーションのログ出力レベル	warn
SS_WORKSPACE_HOST_DIR	ホスト側の作業データ保存領域	なし（指定必須）
SS_NGC_REGISTRY_ENDPOINT	NGC 接続先	<a href="https://registry.nvidia.com">https://registry.nvidia.com</a>
SS_NGC_REGISTRY_USER_NAME	NGC ユーザー名	\$oauthtoken
SS_RESCALE_SINGULARITY_VERSION	Rescale での Singularity ランタイムバージョン	3.2.0
SS_RESCALE_JOB_WALLTIME	Rescale でのタスク実行最大時間	3600