Diskless Compute Nodeを使ったImmutable OpenStack

ビットアイル・エクイニクス株式会社
山下 祐生

自己紹介

自己紹介

- 名前：山下祐生
- 年齢：25歳
- 所属：ビットアイル・エクイニクス
- 職種：OpenStack Enginner
- 社会人歴：4年
- 経歴：某大手NIerに新卒入社し3年勤務したあと、
  2016年4月からビットアイル・エクイニクスにJoin
- 業務：OpenStackのコンサル、設計、構築、運用、RD的活動を幅広く行っています
- OpenStack歴：本格的に初めて2年

テーマ

- Diskless**にしようと思ったかのきっかけを知ってもらう**
  - Why Diskless、**前からある技術だけどなぜ今？**
- **インフラエンジニアのインフラ管理を楽にしたい**
  - **インフラってなに？インフラ管理をどう楽にするの？**
- DisklessComputeNode**を実現することでなにが嬉しいかを知ってもらう**
  - **なぜ、**OpenStack**に適用？**OpenStack**との親和性って？**

OpenStack運用の課題

悩ましい問題がいっぱい…

- クリティカルなバグが多い
  - パッチ適用、アップデートの機会が多い

- アップデート、パッチ適用問題
  - オープンスタックはスケールできる柔軟性があるのが売り
  - しかし、スケールしていけばアップデート台数が途方も無い規模に・・・
  - Controllerのアップデートには断時間が発生してしまう

- 大規模での運用コスト問題
  - OpenStackが大規模になればなるほど、ハードウェア故障から逃れられない
    - 特にHDDの故障が顕著・・・

悩ましい問題がいっぱい…

- クリティカルなバグが多い
  - パッチ適用、アップデートの機会が多い
- アップデート、パッチ適用問題
  - オープンスタックはスケールできる柔軟性があるのが売り・
  - Controllerのアップデートには断時間が発生してしまう
- 大
  - C
    -

この問題はなぜ難しいか
■パッケージのアップデートをした後の検証が必要となること
■止めるにしても停止時間をより短くする工夫が必要

悩ましい問題がいっぱい…

- クリティカルなバグが多い
  - パッチ適用、アップデートの機会が多い
- ア
  - ス
  - しかし、スケールしていけばアップデート台数が途方も無い規模に・・・
  - Controllerのアップデートには断時間が発生してしまう

この問題はなぜ難しいか
■大規模になるとアップデートを適用するノードがとてつもない台数になる
■大規模環境ではHDD故障の度に交換する運用は大変
だからといって交換しないことは難しい

- **大規模での運用コスト問題**
  - OpenStack**が大規模になればなるほど、ハードウェア故障から逃れられない**
    - **特にHDDの故障が顕著・・・**

この課題に対する解決を図る

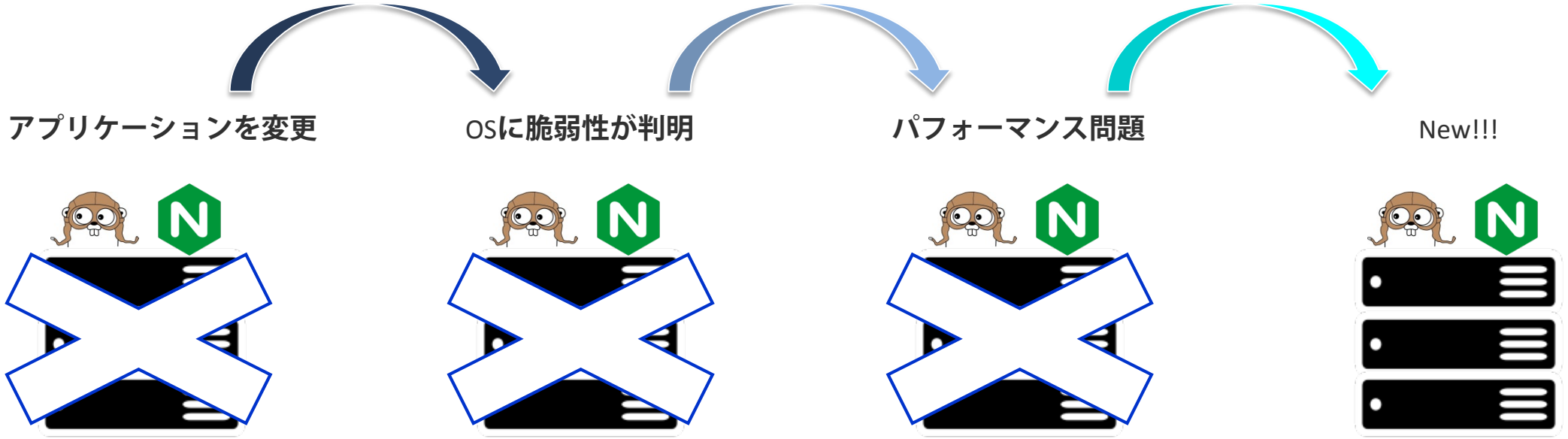

immutable infrastructureだ！

これからのOpenStack運用を考える
～ immutable infrastructureの適用～

immutableとは

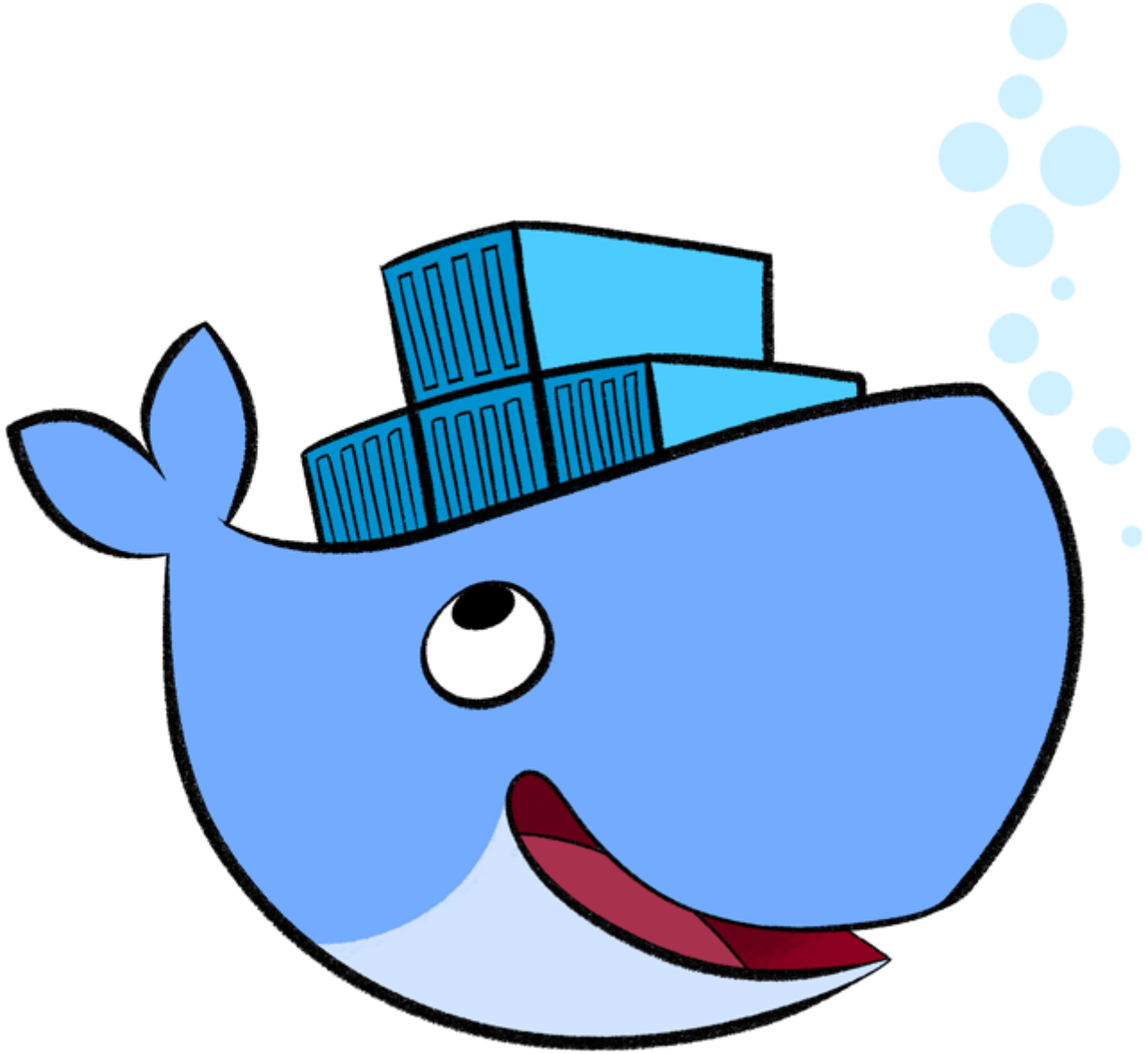
インフラでは、「設定変更などを行わない不変のままサーバを利用する」手法のことを表す

immutable infrastructure

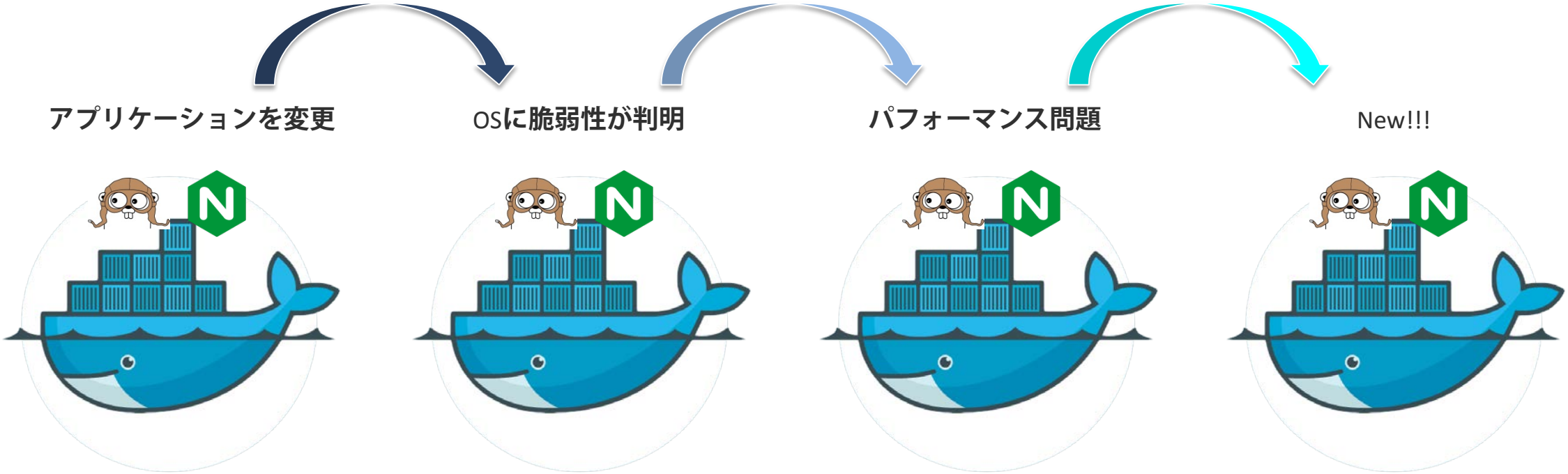


このように何か変更を加えるたび、サーバに変更を逐一入れていくのではなく、サーバを新しく作りなおして運用を行っていく

このアーキテクチャーって・・・



immutable infrastructure

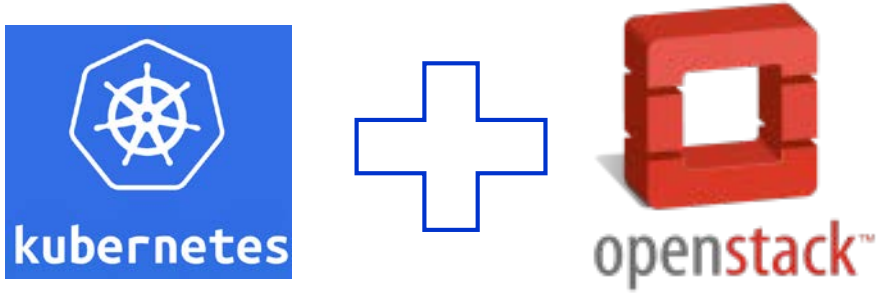


[朗報]Dockerで全て解決！！

コンテナ万歳！！


OpenStackもコンテナで
管理だ！

Stackanetes



OpenStack on k8s

- K8sのインフラ上にOpenStackコンポーネントがコンテナとして配置される
  - K8sの機能であるオートヒールなどの恩恵が受けられる
- OpenStackコンポーネント配置の自由度が向上
  - ホストのメンテナンス時にコンテナの退避などが非常に容易になる
- コンテナなのでスケールも容易
  - コンテナもk8s基盤上で増やすだけで可能に

この問題はなぜ難しいか　～再掲～

■パッケージのアップデートをした後の検証が必要となること

➡コンテナはスクラップアンドビルドに適しており、検証が容易

■停止時間をより短くする工夫が必要

➡コンテナの起動時間は非常に短い、ブルーグリーンのような切り替えが可能

■大規模になるとアップデートを適用するノードがとてつもない台数になる

➡アップデート適用済みコンテナを展開するだけでアップデートが完了する

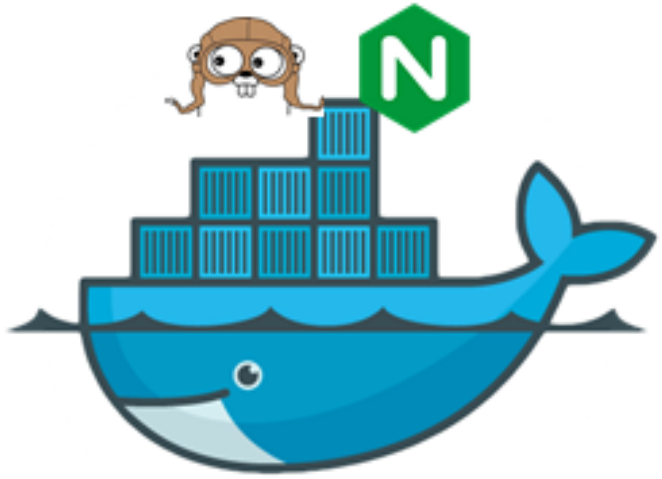
■大規模環境ではHDD故障の度に交換する運用は大変
だからといって交換しないことは難しい

やっぱり時代はコンテナ！

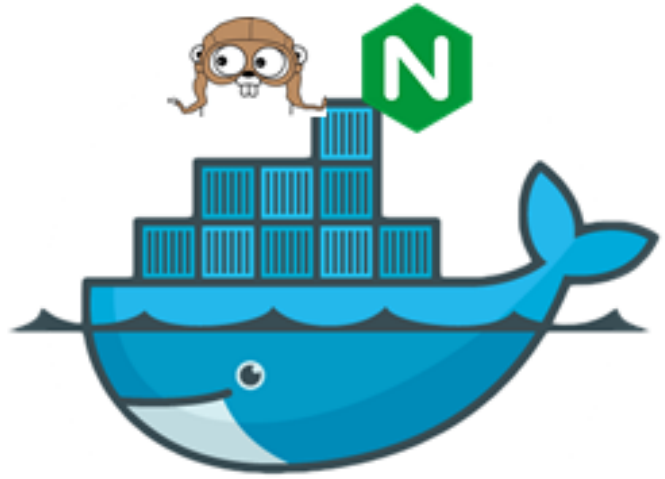
immutable infrastructure

アプリケーションを変更

OSに脆弱性が判明



パフォーマンス問題



[朗報]Dockerで全て解決！！

immutable infrastructure

[朗報]Dockerで全て解決！！

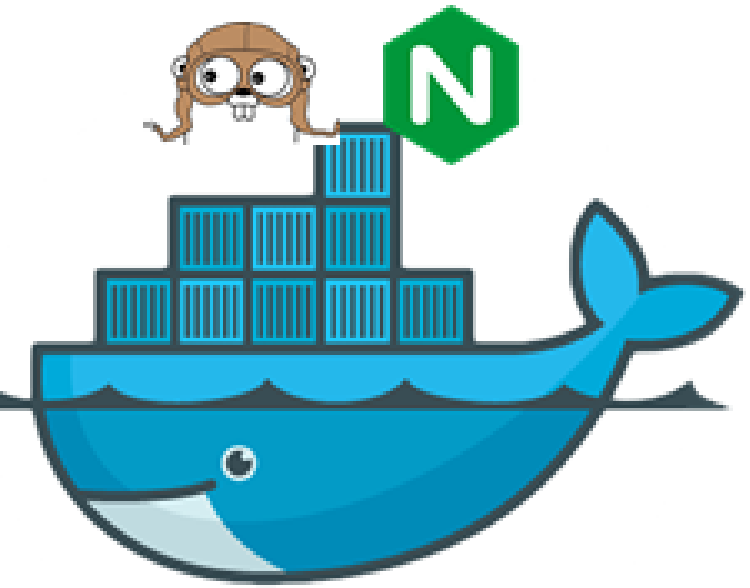
Immutable Infrastructure

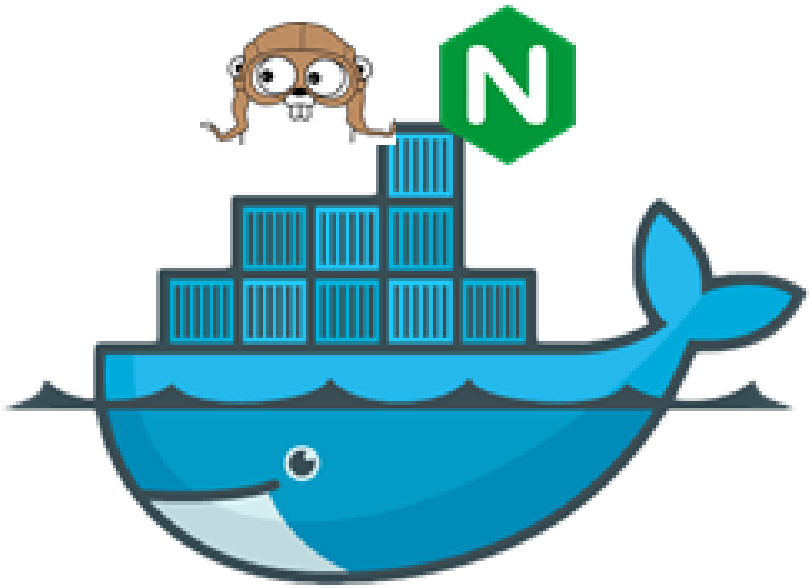
あれ？OSの脆弱性ってそもそも…？

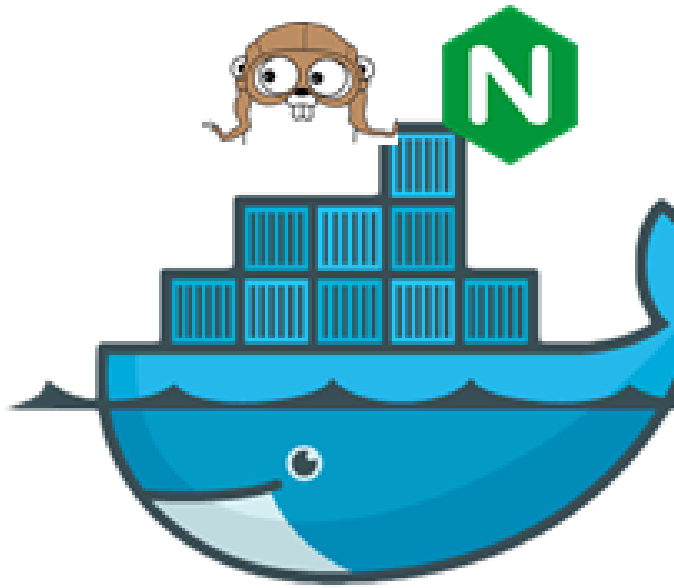
アプリケーションを変更

OSに脆弱性が判明

パフォーマンス問題







Let me tell you about Infra Engineer

- インフラエンジニアとは・・・
  - インフラの面倒を見る裏方
  - さっきの例であればOSをインストールして、Dockerを提供する側

- 物理、OSからは逃げられない
  - インフラエンジニアにとってのimmutable infrastructureは低レイヤーから意識されたものでなければならない

- スケールと運用
  - スケールしていけばスケールしていくほど、一台一台に対するアップグレードや、パッチ修正などが大変に・・・

k8sを使っても起こりうる課題

- K8sホストのメンテナンス
  - コンテナの退避を行い、メンテナンスを行う
  - しかし、台数が増えれば増えるほどコストが増加

- コンテナはkernel共有なので、KVMなどのノードはベアメタルで動かしたい
  - コンテナは便利！だけどなんでもコンテナというわけにはいかない

残りの課題をどう解決するか
■大規模環境ではHDD故障の度に交換する運用は大変
■k8sホストのメンテナンス、コンテナを使ってもホストOSのメンテが必要
←NEW
■kernelで動くアプリケーションはベアメタルで動かしたい ←NEW


ステートレスなノードは全
てDisklessにしてしまえ！

Diskless ComputeNode

Diskless

Diskless Boot

- Diskless Boot**とは・・・**
  - ***ディスクドライブを使わずに、ネットワークブートを利用してサーバーからオペレーティングシステムをロードするワークステーション、またはパーソナルコンピュータのことである。***
  - **具体的な技術としては**ISCSI boot ,PXE+NFSroot,gPXE+ISCSI**など**

- **元々は会社の検証用保守切れサーバの再利用で活躍**
  - **保守切れのサーバはどんどん**HDD**がやられていく・・・**
  - OS**のデータを保持しているストレージだけ保護レベルを上げて、他のディスクはシングルでバックアップを取得し運用**

ComputeNode

OpenStackにおけるComputeNode

- OpenStackのコンピュートリソースを提供するノード
  - 具体的には仮想マシンが動くノード、一般的にはKVMとOVSが同居している
  - 最近はHypervisorとしてLXDやdockerが使われたりすることも

- ステートレスなノード
  - コンピュートノードには消えてはいけないデータはなく、増減したりする
  - ステートフルなデータはDB、MQ、ストレージが持っている

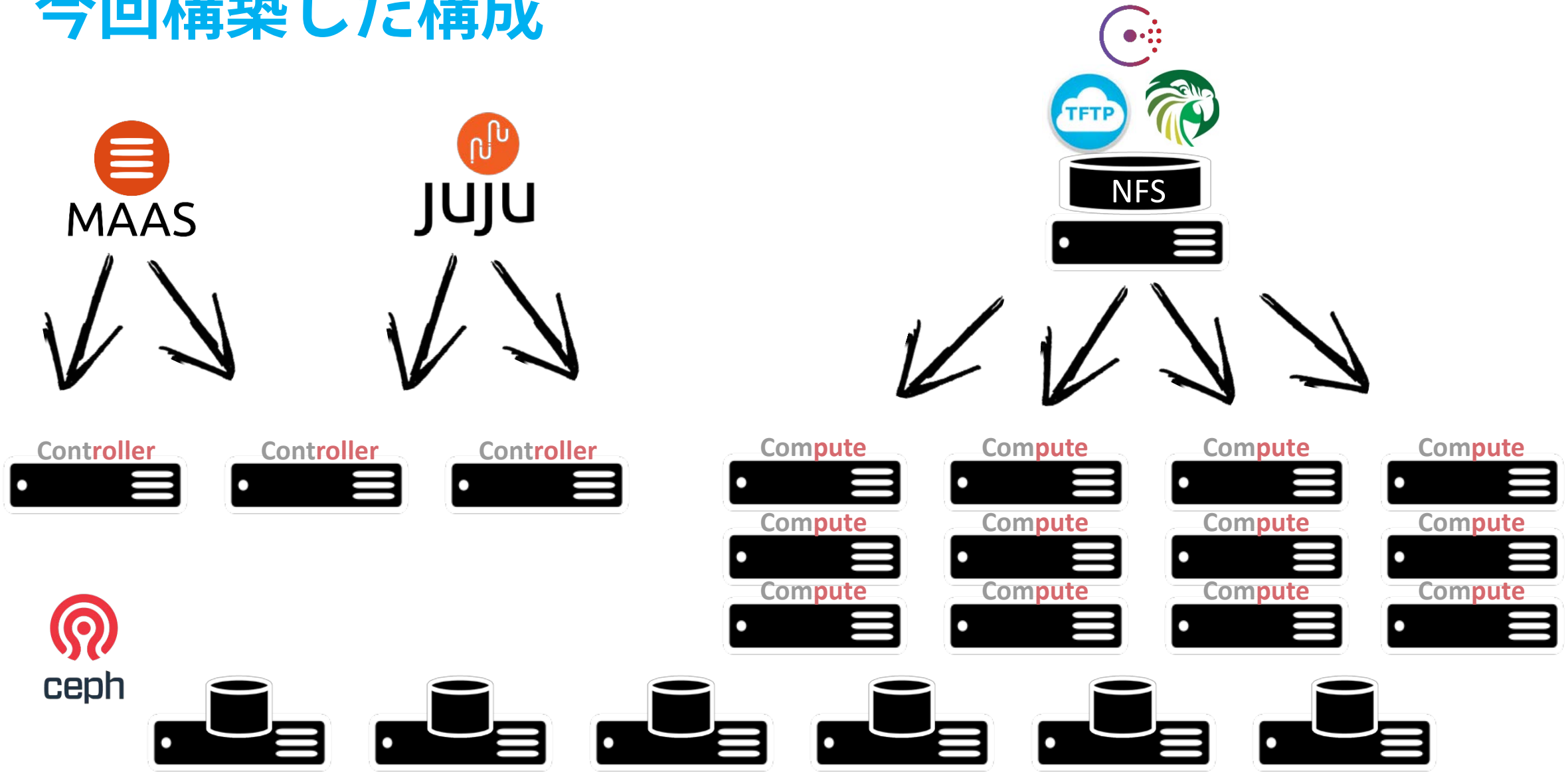
- OpenStackというシステムの中で一番台数を占めるコンポーネント
  - OpenStackの中でスケールと言われればNova Compute

Diskless**を実際にやってみた**

Diskless ComputeNodeってなに？

- 今回後述するOpenStack運用上の課題を解決するために考えたアーキテクチャ
- 技術的にはその名の通り、Diskless BootするCompute Node
  - 工夫している点として、NFSをROマウントしてオンメモリに移しBoot するようになっている
- 増設、廃棄、アップデートをより簡単にコンテナのような使い勝手をベアメタルで目指してみた
- ComputeNodeにはDiskを搭載しないで作ったほうがメリットがでるのではという問いかけ

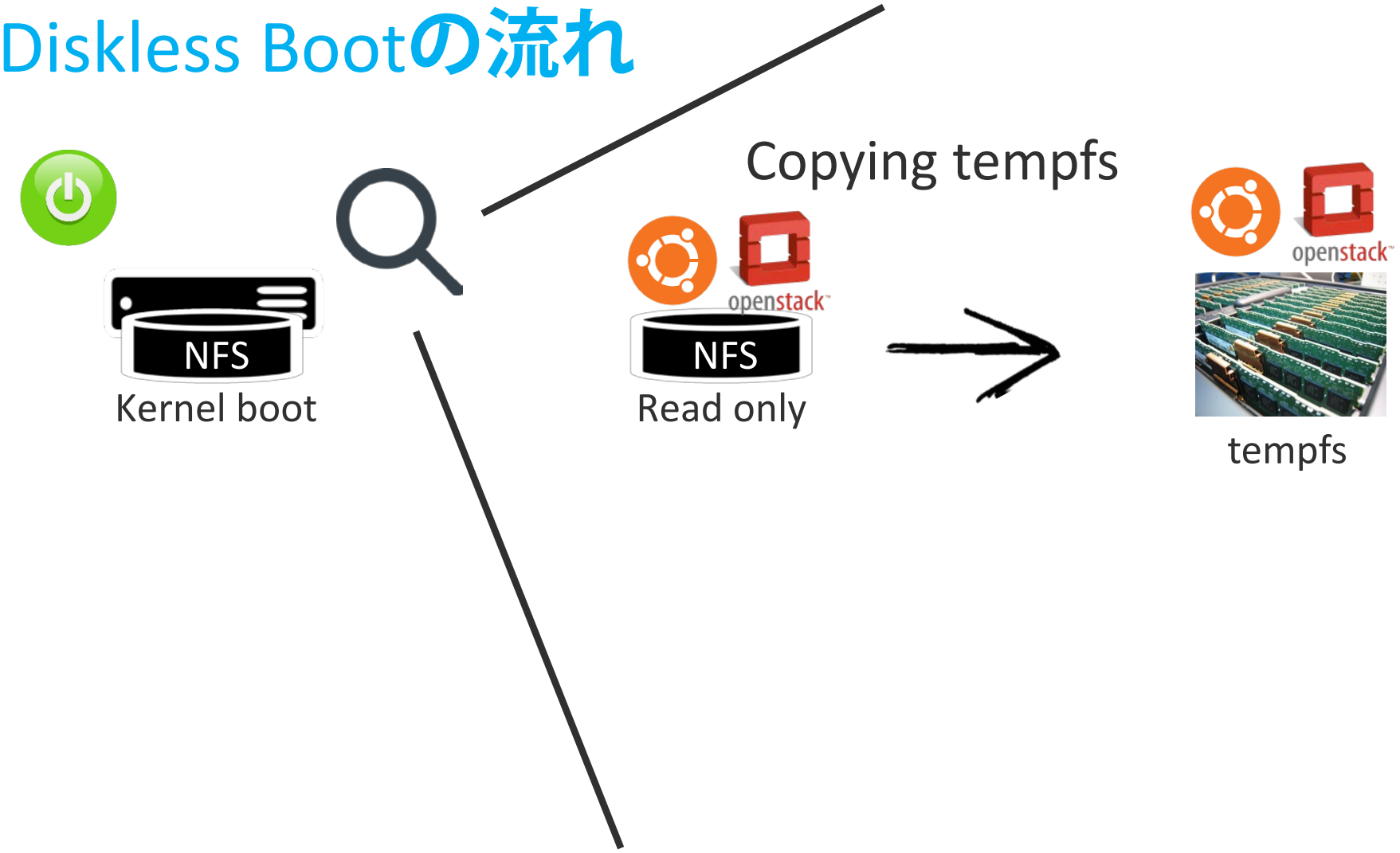
今回構築した構成



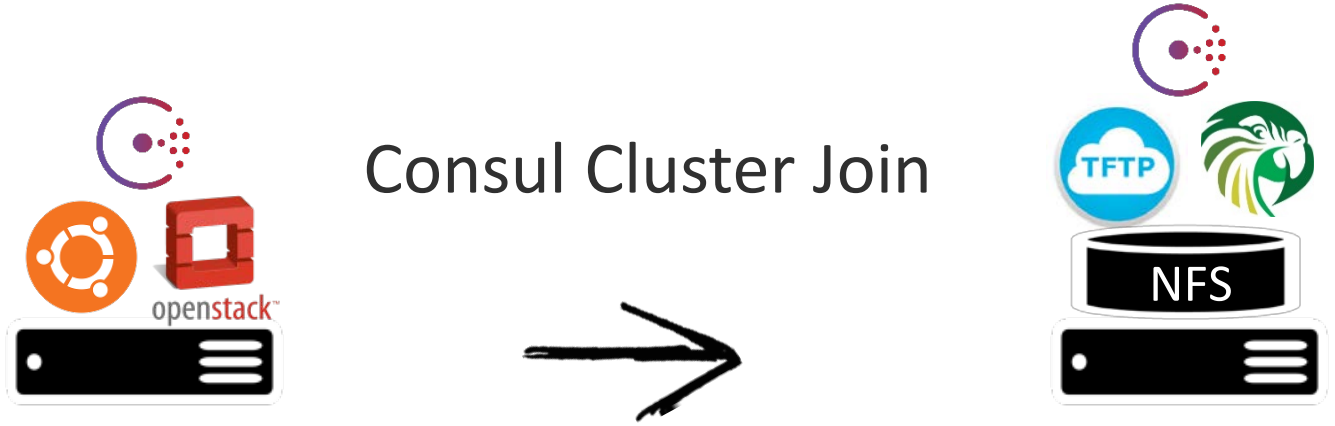
Diskless Bootの流れ

Diskless Bootの流れ

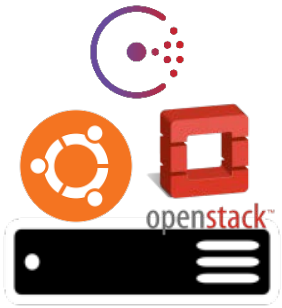
Diskless Bootの流れ

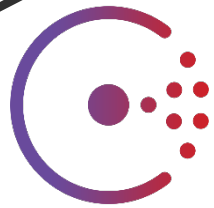


Diskless Bootの流れ



Diskless Bootの流れ

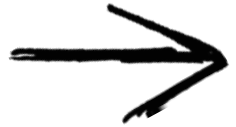




consul template

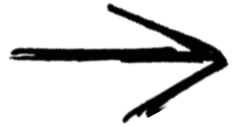
setting configure file

hosts.tmpl



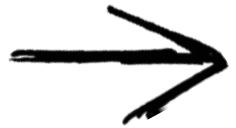
/etc/hosts

nova.conf.templ

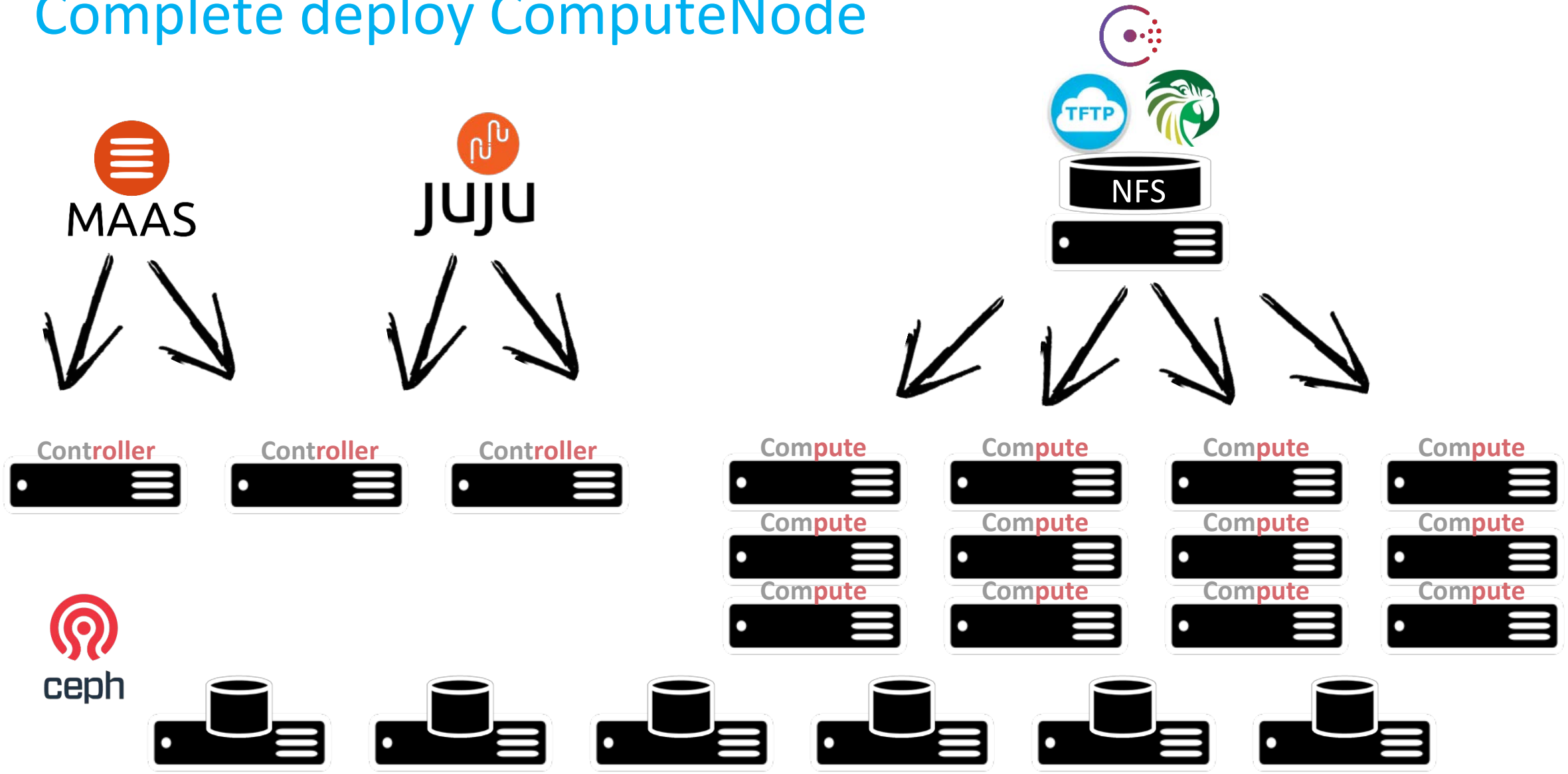


/etc/nova/nova.conf

Etc…



Complete deploy ComputeNode

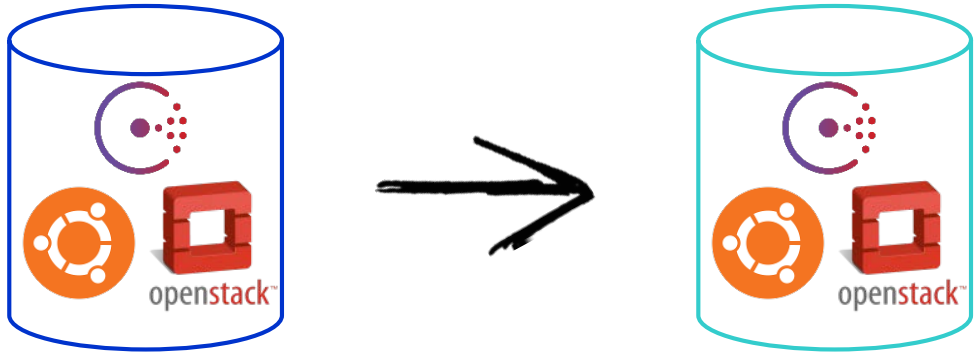


何が嬉しいの？

- ComputeNodeの増設が非常に簡単になる
  - 従来の仕組みではOSをインストールして、そこからアプリケーションをデプロイすると言う手順を踏んでいた
  - Deployにかかる時間が大幅短縮可能に！
- HDD障害が起きない
  - 一番故障確率の高いHDDを搭載しないで済むならそれに越したことはない
- オンメモリで動くので再起動すると変更されたデータが消える
  - 仮想マシンのスナップショットのような使い方が可能
  - 検証が容易にできる

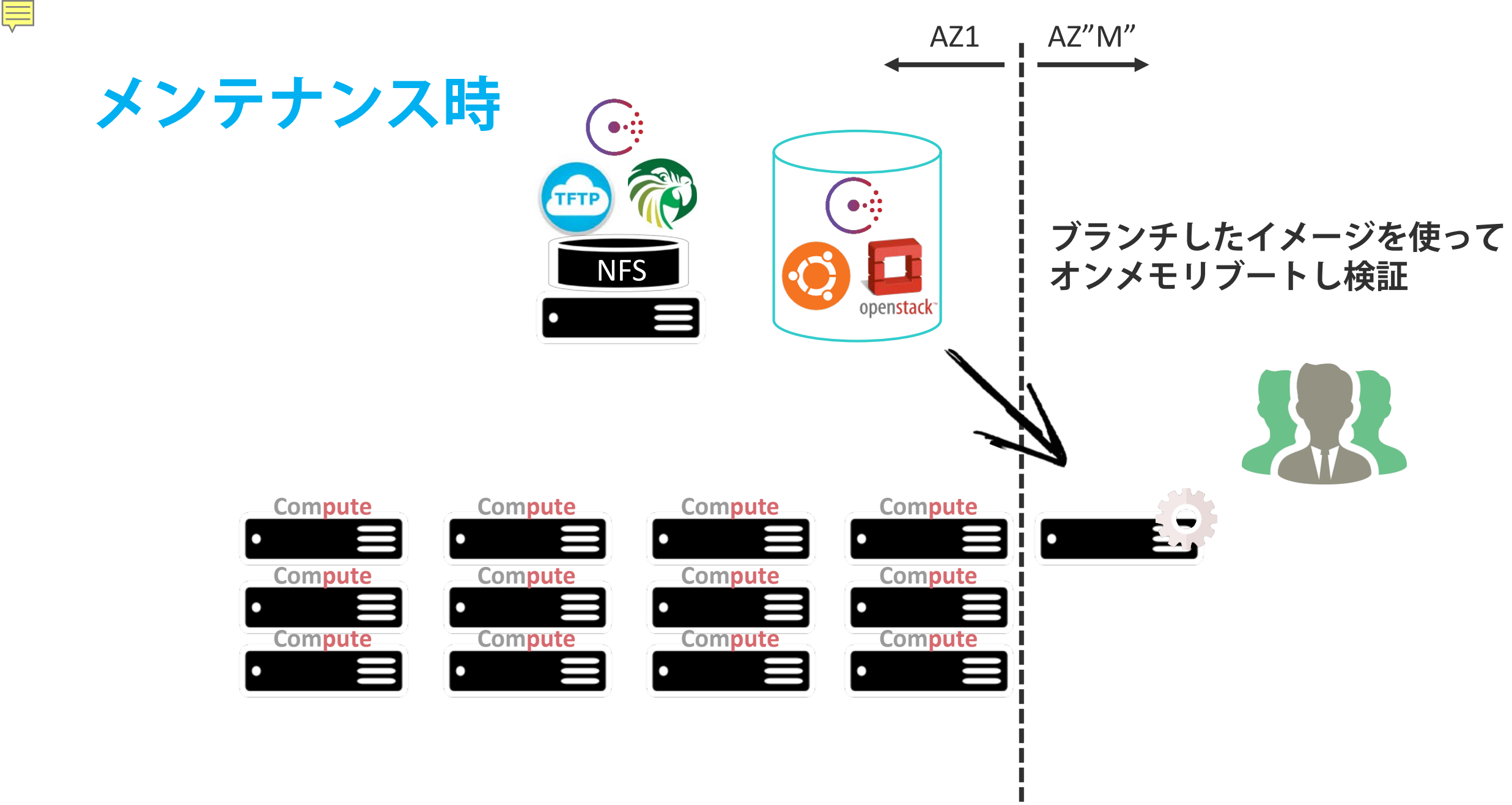
メンテナンス時

現用イメージからコピーを作成

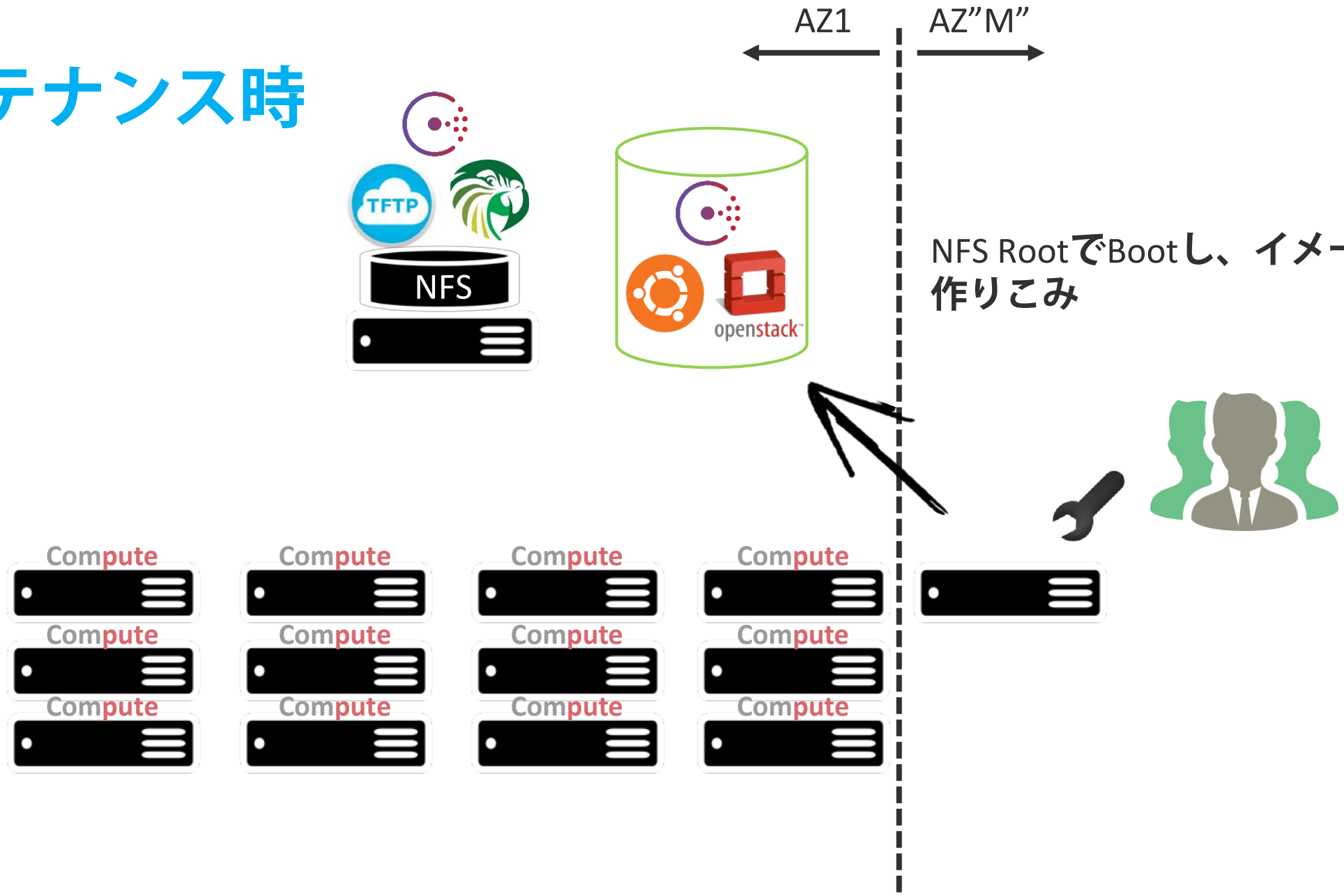


Gitのブランチを切るイメージ

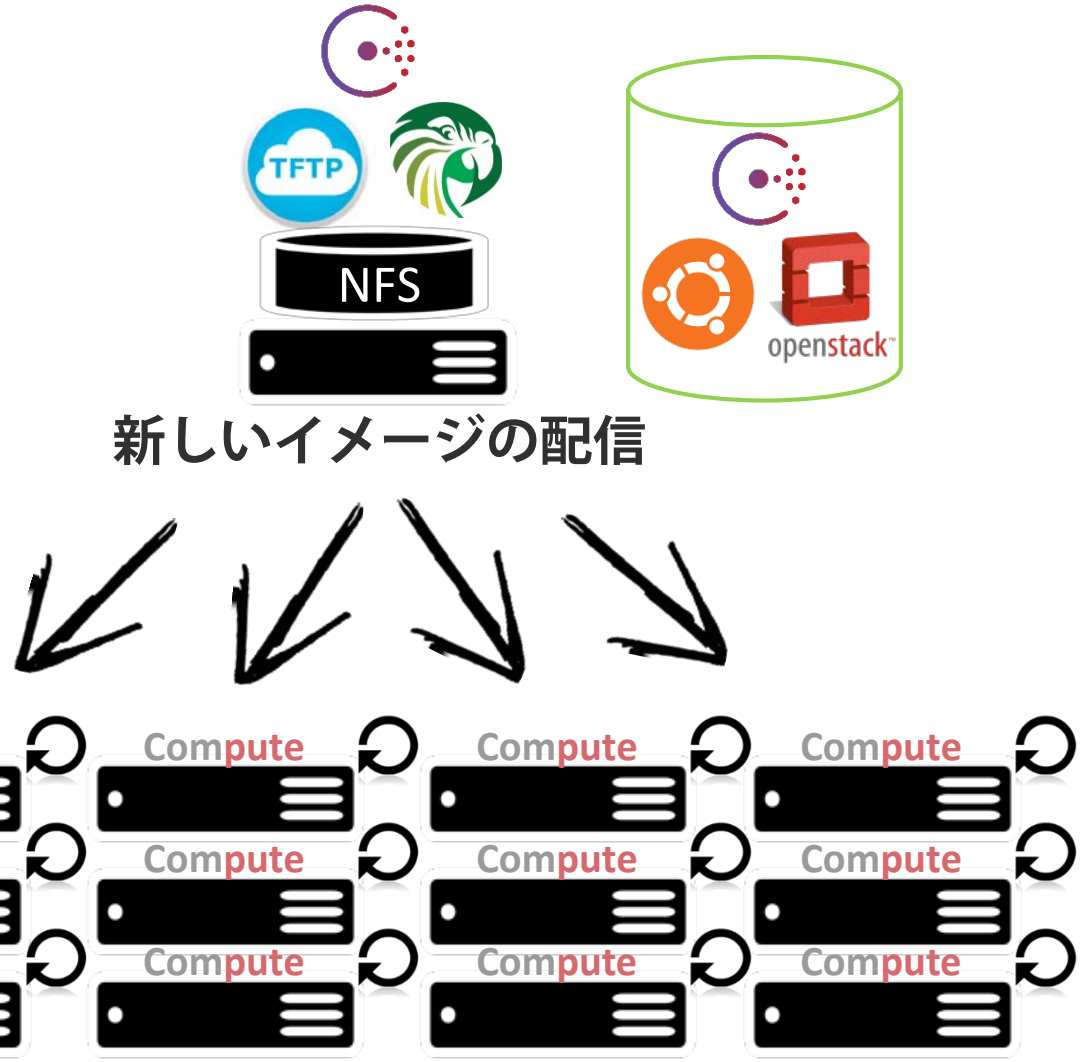
メンテナンス時



メンテナンス時



メンテナンス時



残りの課題をどう解決するか

■大規模環境ではHDD故障の度に交換する運用は大変

➡Disklessにすることで解決

■k8sホストのメンテナンス、コンテナを使ってもホストOSのメンテが必要

➡k8sもコンテナのような使い勝手のDiskless化

■kernelで動くアプリケーションはベアメタルで動かしたい

➡Disklessでベアメタルデプロイできる

ImmutableをOpenStackに

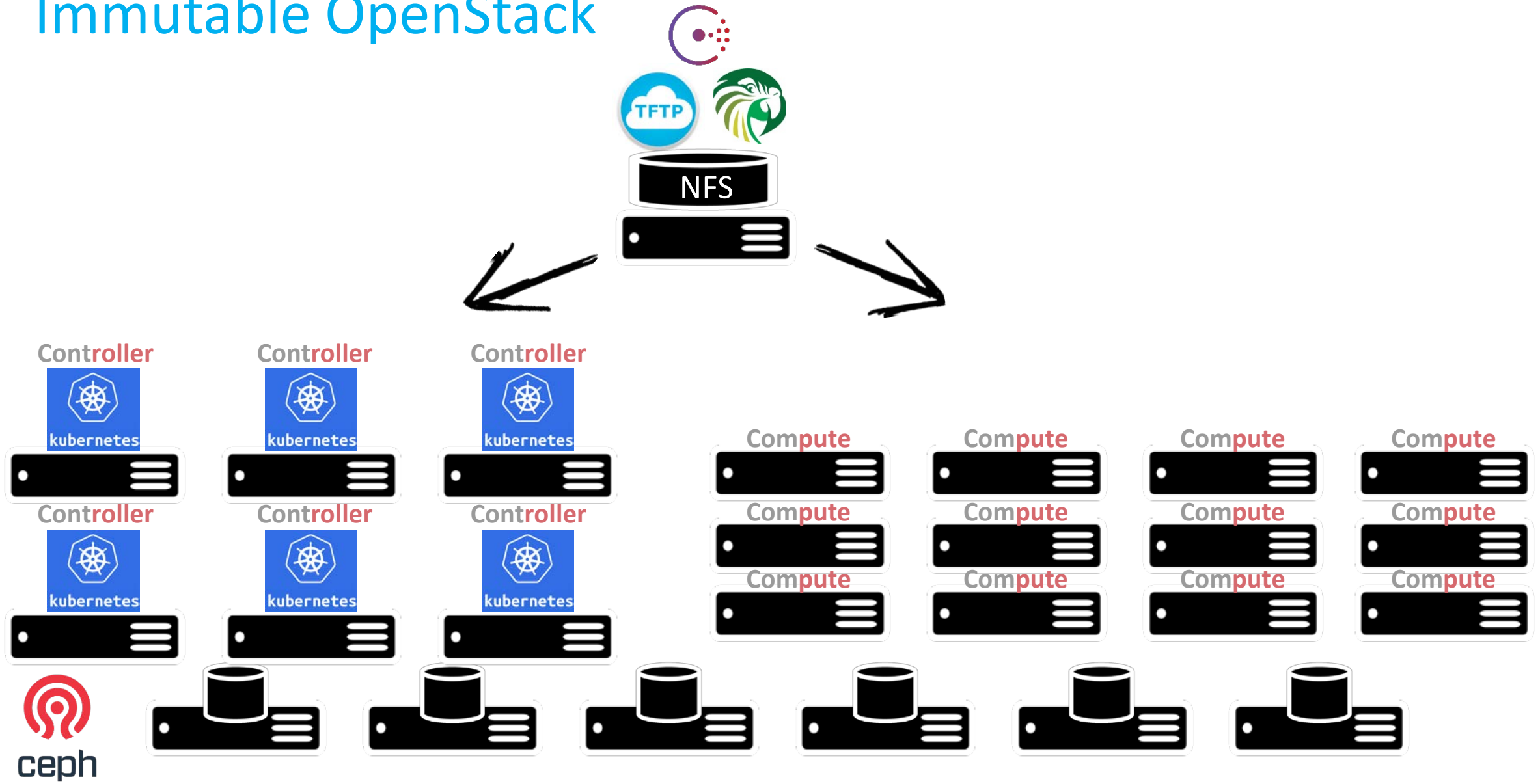
immutable infrastructureとOpenStack

- もともとOpenStackは疎結合なアーキテクチャ
  - Immutable infrastructureとの親和性が高い

- StackanetesとDisklessでOpenStackにおけるimmutableを適用可能に
  - コントローラノード群はkubernetesで管理し、コンピュート郡は今回のアーキテクチャを適用

- コンテナなのでスケールも容易
  - コンテナもk8s基盤上で増やすだけで可能に

Immutable OpenStack



現状での課題

- Diskを全く使わないため、メモリが枯渇するとノードがダウンする
  - Diskにデータを蓄積していくようなアプリケーションは載せられない

→実際に利用するとなると、オンメモリブートのAZとトラディショナルな構成のAZを分けて運用をしていくことになる

- ディスクイメージを管理する仕組みが必要
  - イメージファイルのディレクトリを管理できる「git」のような仕組みが欲しい

- PXEブートのラベル生成を自動化したい
  - イメージのブランチを作ってcommitしたらラベルが生成されるような仕組みがあると便利

まとめ

まとめ

- OpenStackを運用をより良くしていくために
  - いろいろな考え方を吸収して反映していかなければならない
  - Disklessの発想も一例

- OpenStackのもっと得意なことを伸ばす設計、アーキテクチャを考える
  - 守りに徹するとOpenStackの良い所が失われ、もったいない
  - 何かが起こらないシステムより、何かが起こっても即時対応できるシステムを目指す
  - ステートレスなノードとステートフルなノードで運用を分けるなど

×

ビットアイル・エクイニクス株式会社

TEL 03-5805-8154　　FAX 03-3474-5538　　URL http://www.bit-isle.jp/