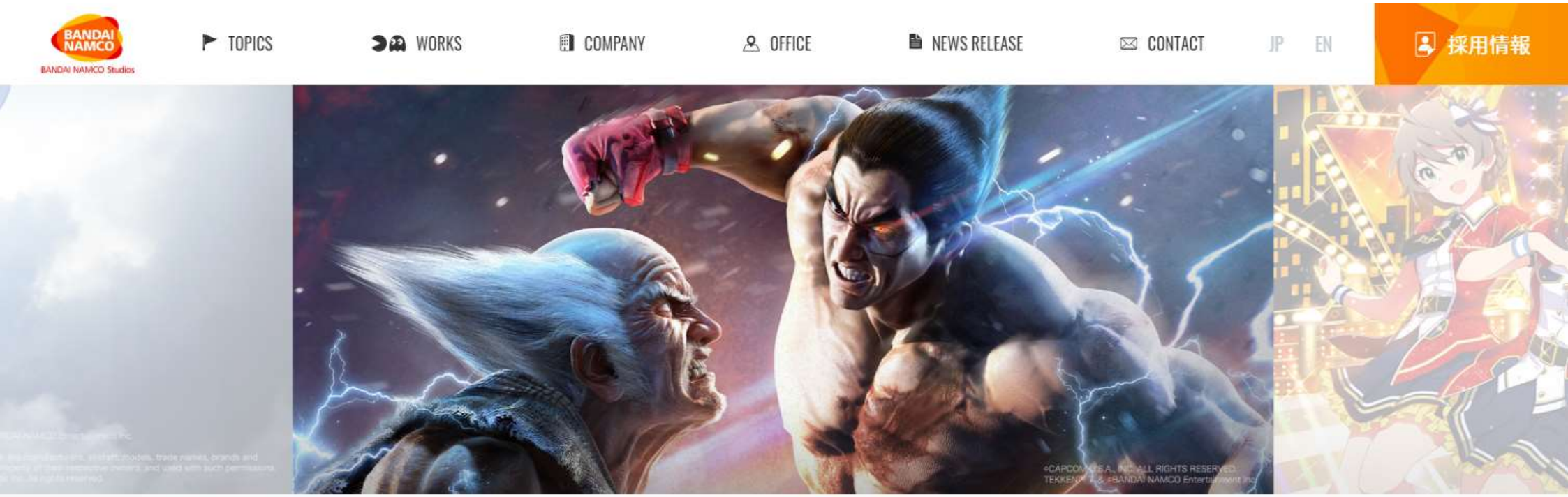
BANDAI NAMCO Studios

クラウドを使ったゲーム開発の効率化
（ビルドパイプラインのクラウド化）

株式会社バンダイナムコスタジオ
BANDAI NAMCO Studios Inc.

会社紹介

BANDAI NAMCO Studios



株式会社バンダイナムコスタジオ

・バンダイナムコグループの開発スタジオ
・家庭用ゲームソフト、モバイルコンテンツ、PCコンテンツなどの企画・開発・運営
・ミッション：**夢・遊び・感動**

自己紹介

株式会社バンダイナムコスタジオ
技術スタジオ 技術スタジオ付
エグゼクティブテクニカルディレクター
大井 隆義

業務
・2015年入社
・ゲーム開発（CS、PC、AC、Mobile…）
・BLUE PROTOCOLテクニカルディレクター

プライベート
・家族構成：妻
・趣味：旅行

株式会社バンダイナムコスタジオ
総務ITサービス部

吉田 卓哉

業務
・2011年入社
・社内開発支援ツールの構築・管理・運用
・ツール例：Perforce、Atlassian Confluence、JIRA、Redmine、Incredibuild、Alienbrain…

プライベート
・家族構成：妻1人、子供3人
・趣味：MTB(マウンテンバイク) 🚵

本日お話する事(アジェンダ)

1. BLUE PROTOCOLとは？
2. ゲーム開発でのビルドパイプライン解説
3. BLUE PROTOCOLのビルドパイプラインクラウド化した話
4. クラウド化した結果や成果、わかった事
5. 細かすぎて伝わらないAWS初心者のハマりどころ

アジェンダ

BANDAI NAMCO Studios

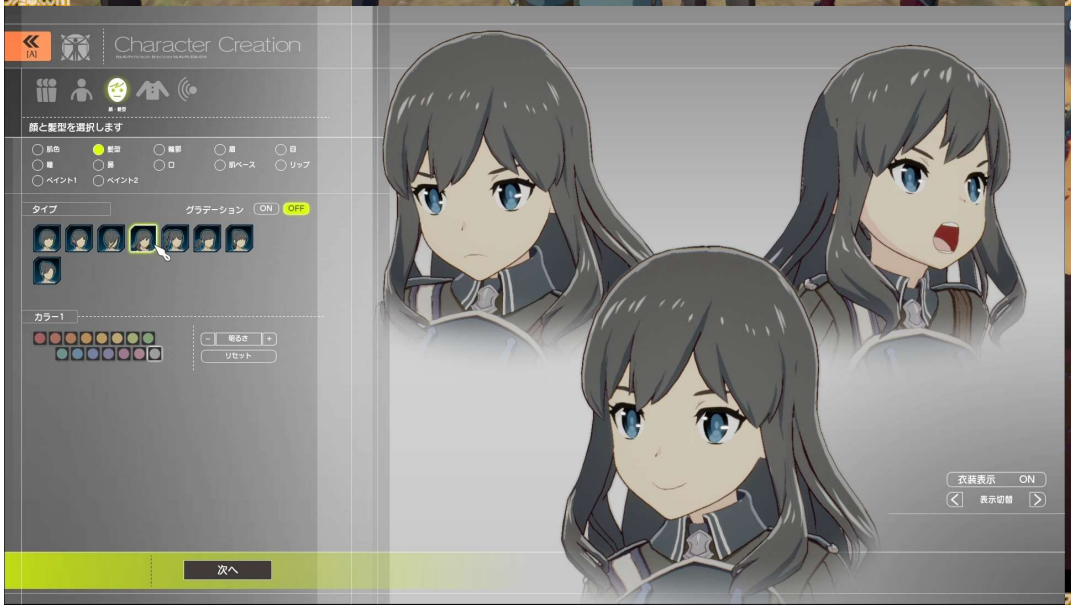
1. **BLUE PROTOCOLとは？**
2. ゲーム開発でのビルドパイプライン解説
3. BLUE PROTOCOLのビルドパイプラインクラウド化した話
4. クラウド化した結果や成果、わかった事
5. 細かすぎて伝わらないAWS初心者のハマりどころ

BLUE PROTOCOL(ブループロトコル)

ジャンル：国産オンラインアクションRPG
プラットフォーム：Windows
エンジン：UnrealEngine4

クローズβテストを4月に実施

BLUE PROTOCOL(ブループロトコル)



アジェンダ

BANDAI NAMCO Studios

1. BLUE PROTOCOLとは？

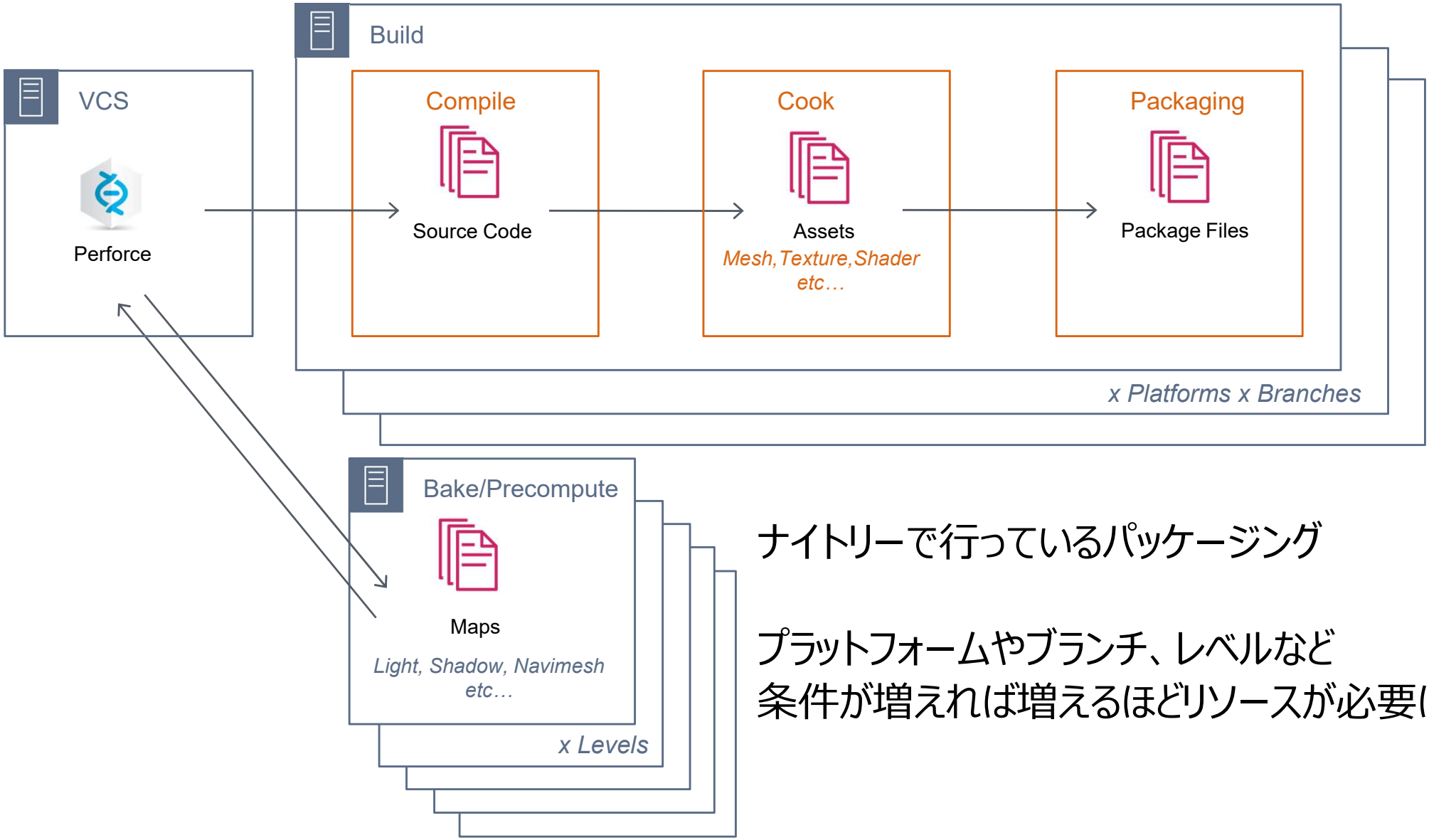
2. ゲーム開発でのビルドパイプライン解説

3. BLUE PROTOCOLのビルドパイプラインクラウド化した話

4. クラウド化した結果や成果、わかった事

5. 細かすぎて伝わらないAWS初心者のハマりどころ

ゲーム開発でのビルドパイプラインを解説(概要)



ナイトリーで行っているパッケージング

プラットフォームやブランチ、レベルなど
条件が増えれば増えるほどリソースが必要になる

Bake&Precompute

マップ（レベル）単位で事前に計算する

- ライティング＆影の焼き付け
  - 広大なマップに多くのライトを配置している
- ナビメッシュの計算
  - NPCが歩き回るために歩ける範囲を事前計算する

© BANDAI NAMCO Online Inc. © BANDAI NAMCO Studios Inc.

Bake&Precompute

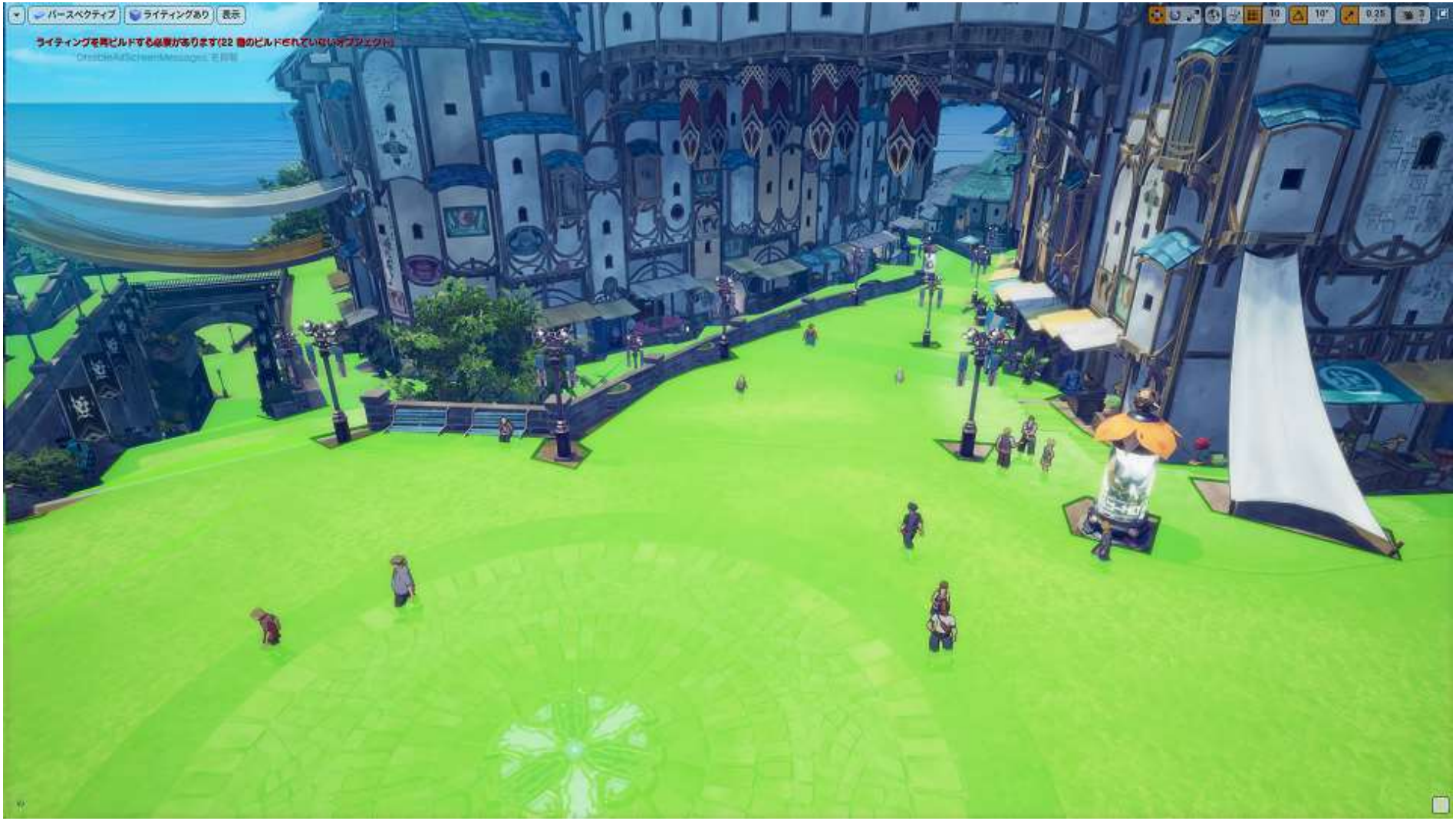
BANDAI NAMCO Studios

ライティング＆影

Bake&Precompute

BANDAI NAMCO Studios

ナビメッシュ、パス



アジェンダ

BANDAI NAMCO Studios

1. BLUE PROTOCOLとは？
2. ゲーム開発でのビルドパイプライン解説
3. **BLUE PROTOCOLのビルドパイプラインクラウド化の話**
4. クラウド化した結果や成果、わかった事
5. 細かすぎて伝わらないAWS初心者のハマりどころ

クラウドゲーム開発の知見はまだ少ないBNSの現状

BANDAI NAMCO Studios

ゲーム開発環境をクラウド化する際の課題

■ 社内LAN前提で構築されたゲーム開発環境
・ユーザー認証基盤, ファイルサーバ, VCS, インフラ,セキュリティ

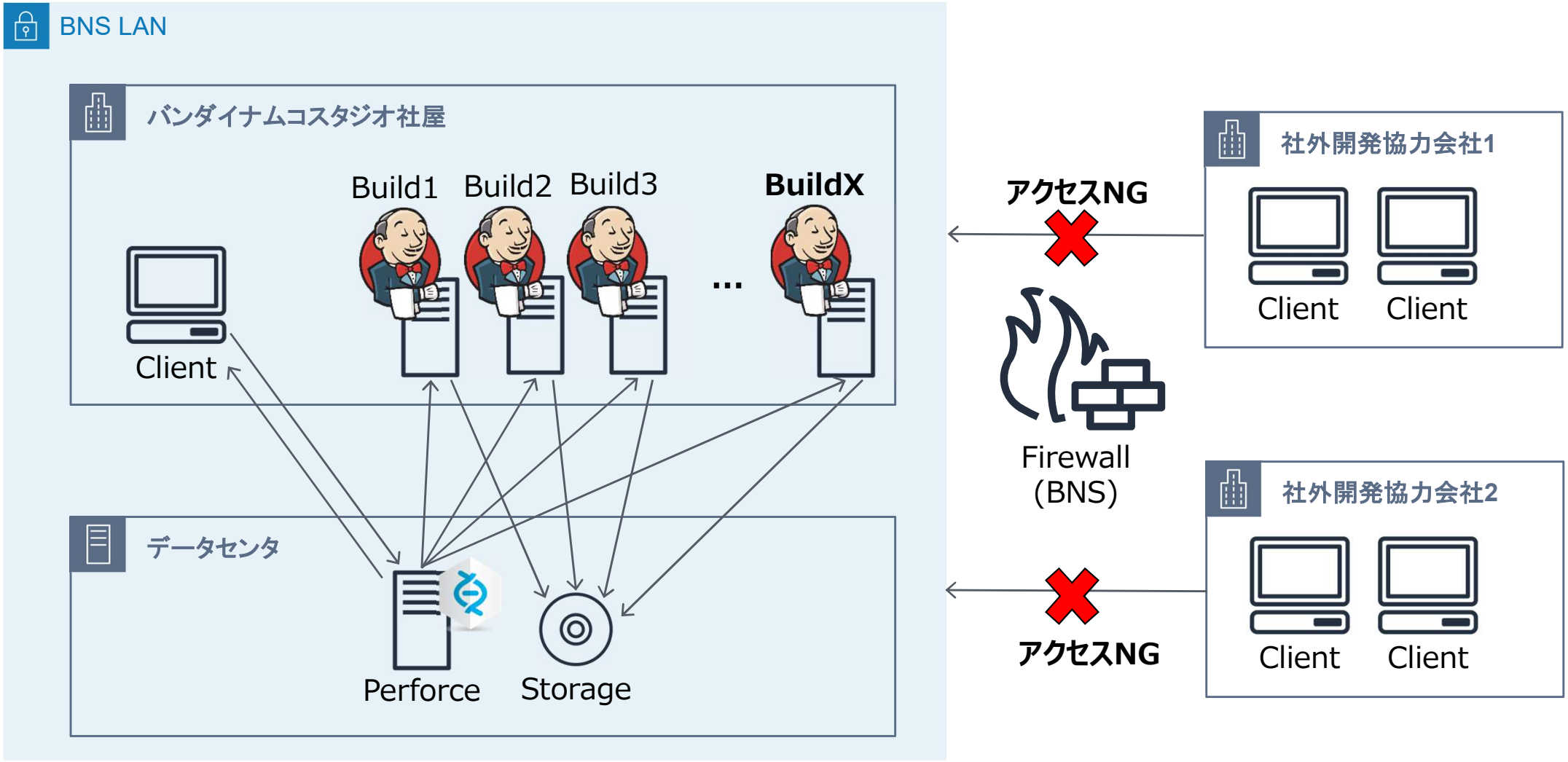
■ スピード・トラフィック
・大量・大容量アセットデータ → 1つのpsdファイルが1GB越え
・分散ビルド環境 → いつでも150コア並列分散ビルド
・描画性能(GPU)や操作レスポンス → 4k 60fps

■ 契約関連(NDA,EULA)
・ゲームエンジン、ミドルウェア等の開発ツール
・1stパーティ製SDK

全て解決してからのクラウド化は非現実的
出来るところ(ビルドパイプライン)からクラウド化してみる

ビルドパイプライン：Before

従来のビルドパイプライン構成

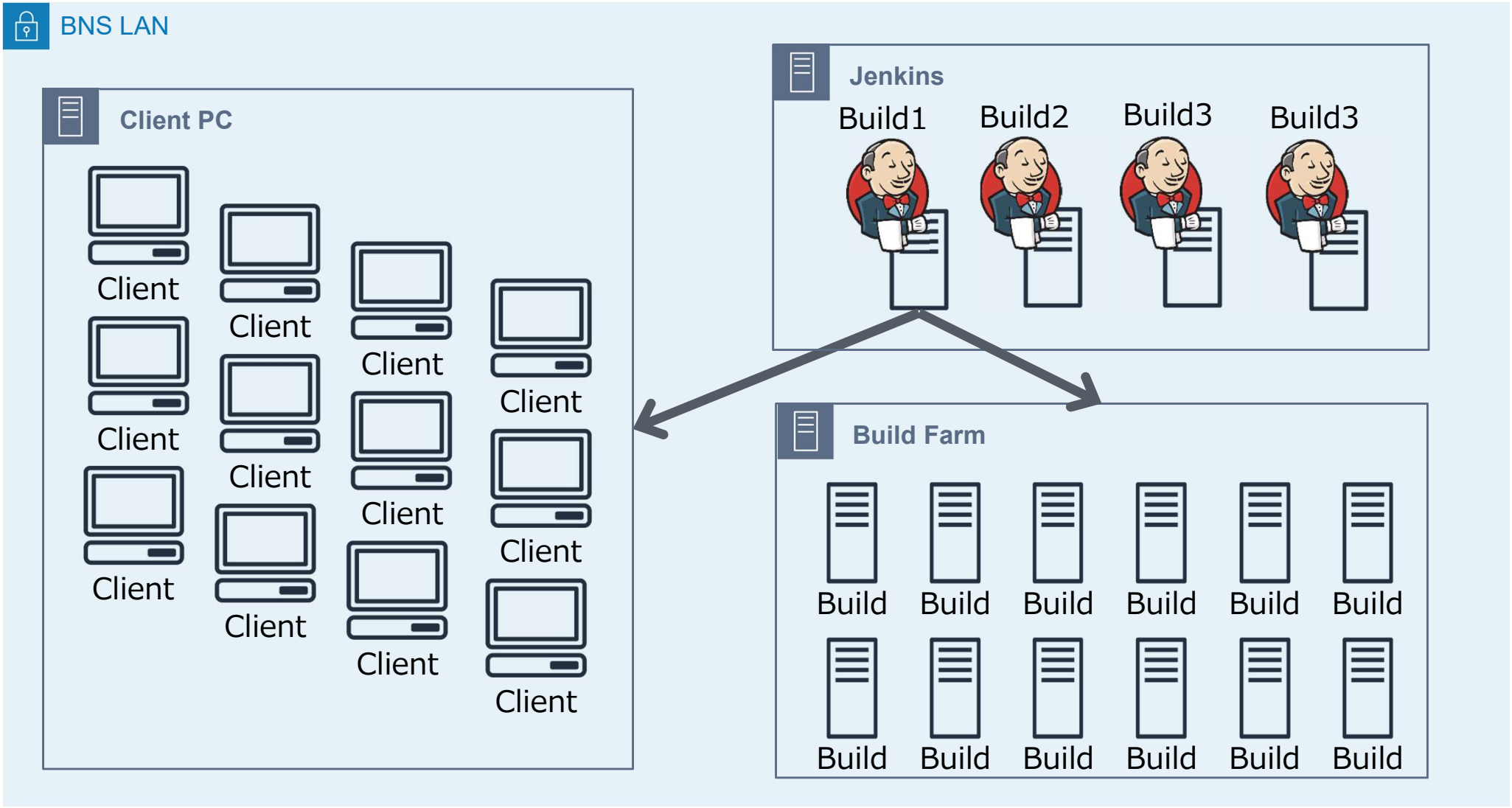


ビルドパイプライン：Before

BANDAI NAMCO Studios

150コア使った並列分散ビルド

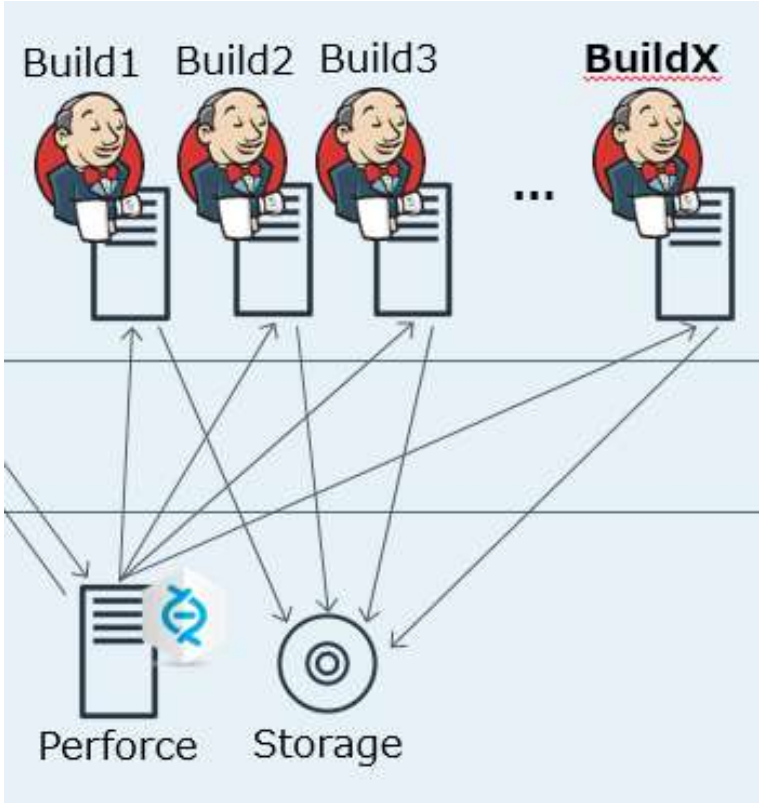
Incredibuildを使うことで開発者PC、ビルドファームの余剰CPUを使って分散ビルド



ビルドパイプライン：Before

BANDAI NAMCO Studios

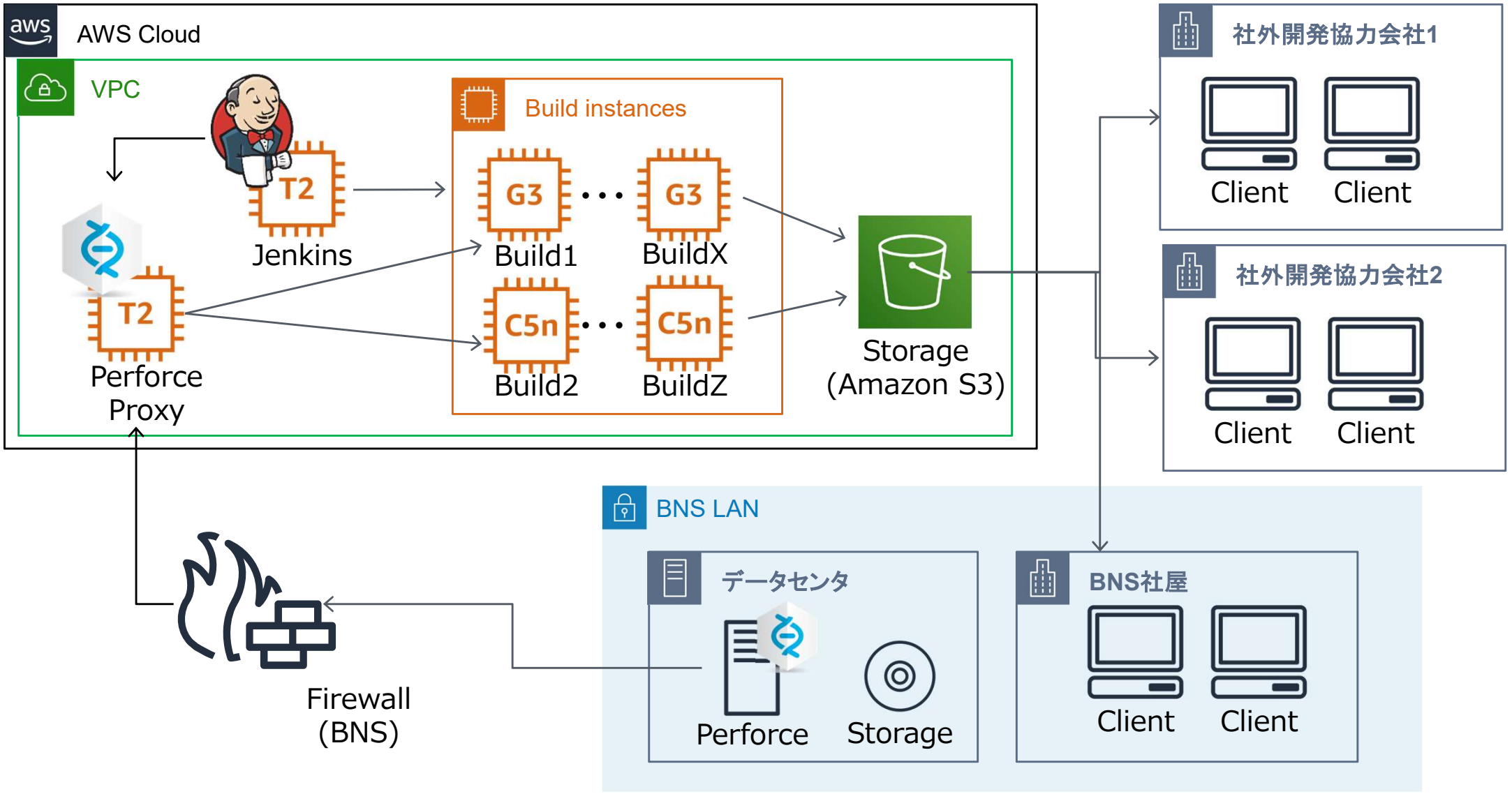
従来のビルドパイプラインの課題や悩み



- ビルドマシンの調達に時間がかかる
  - ・2週間程度
- ビルドマシンのセットアップが大変
  - ・各種ツールの初期インストール
  - ・Jenkinsのジョブ設定
- ビルドマシンの運用が大変
  - ・設置スペース・停電対応・故障対応
- 増えるJenkins
  - ・運用期間が長くなるほど故障時のトラブル対応のリスクが大きくなる→忙しい時に限って故障する

ビルドパイプライン：After

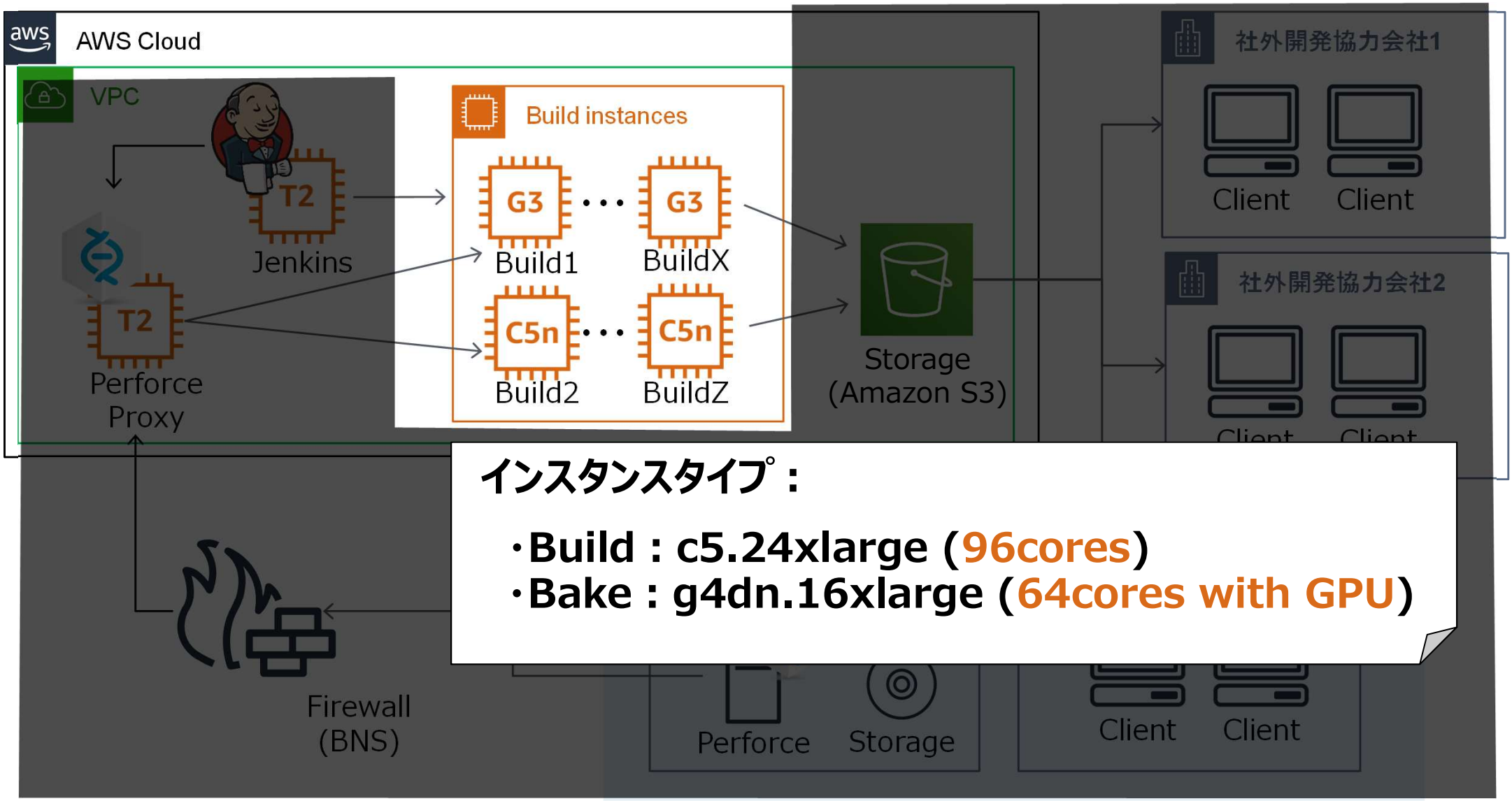
クラウド化したビルド構成



ビルドパイプライン：After

BANDAI NAMCO Studios

クラウド化したビルド構成



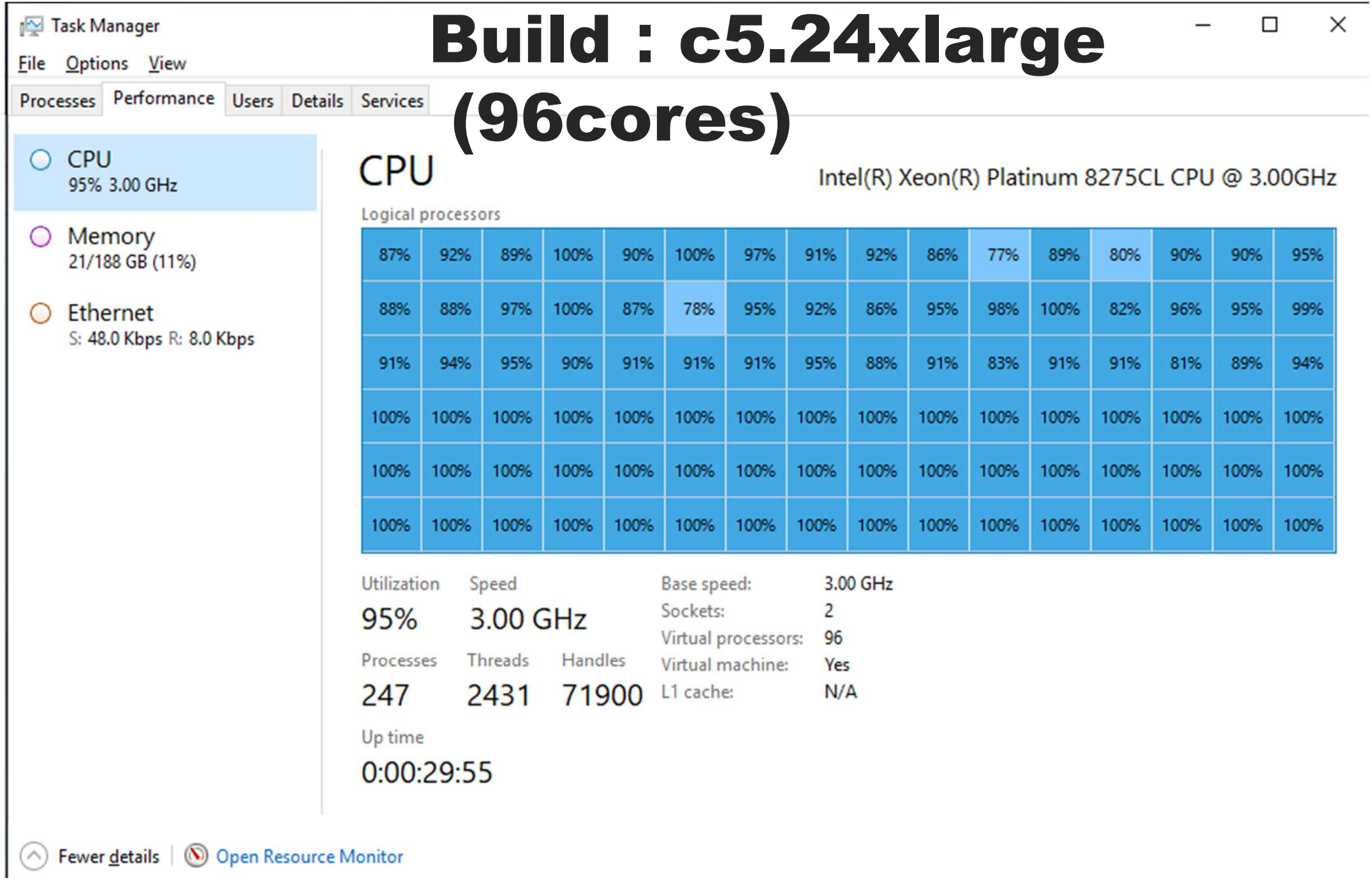
インスタンスタイプ：

- **Build：c5.24xlarge (96cores)**
- **Bake：g4dn.16xlarge (64cores with GPU)**

ビルドパイプライン：After

BANDAI NAMCO Studios

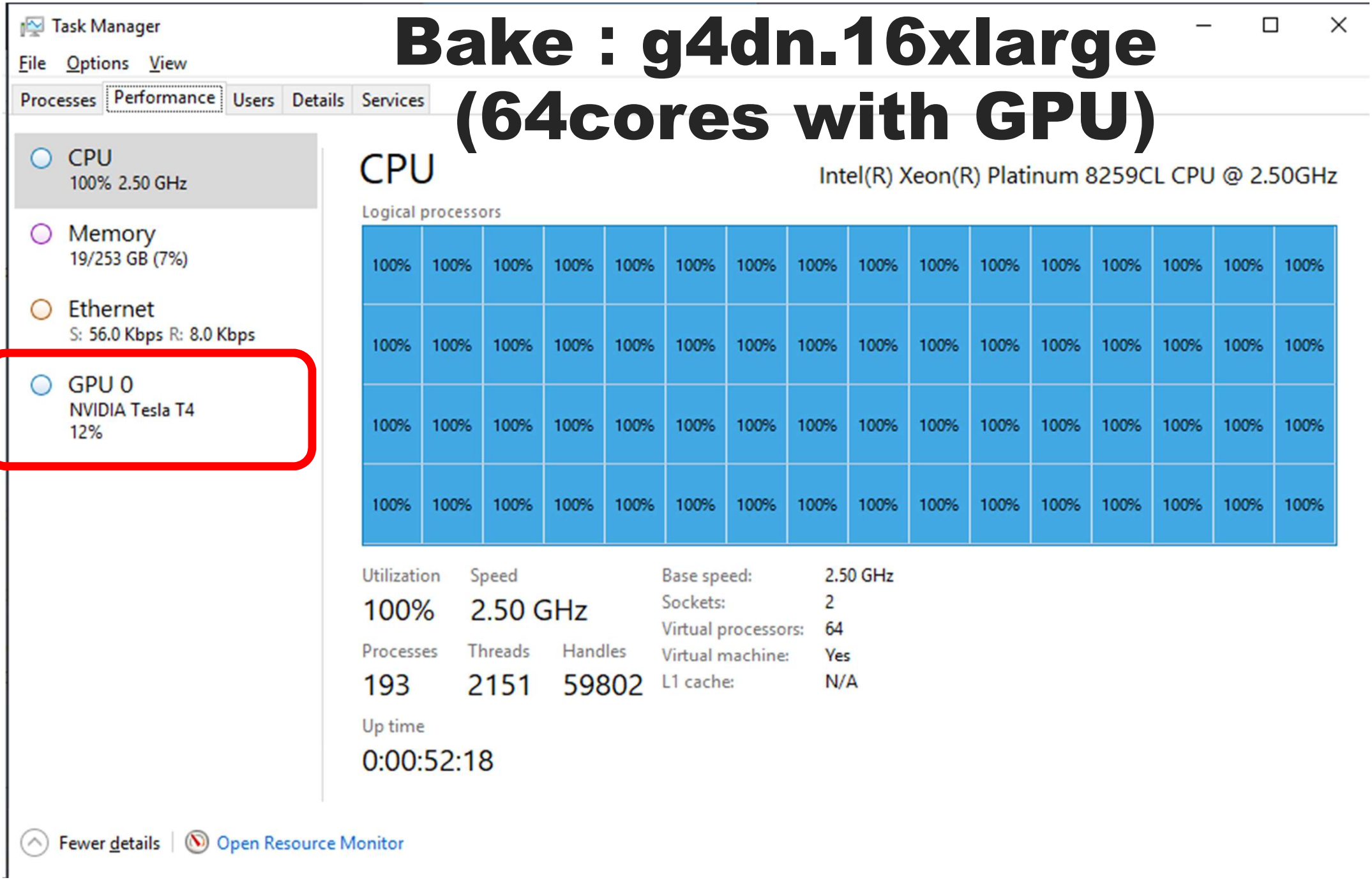
Build：c5.24xlarge (96cores)



ビルドパイプライン：After

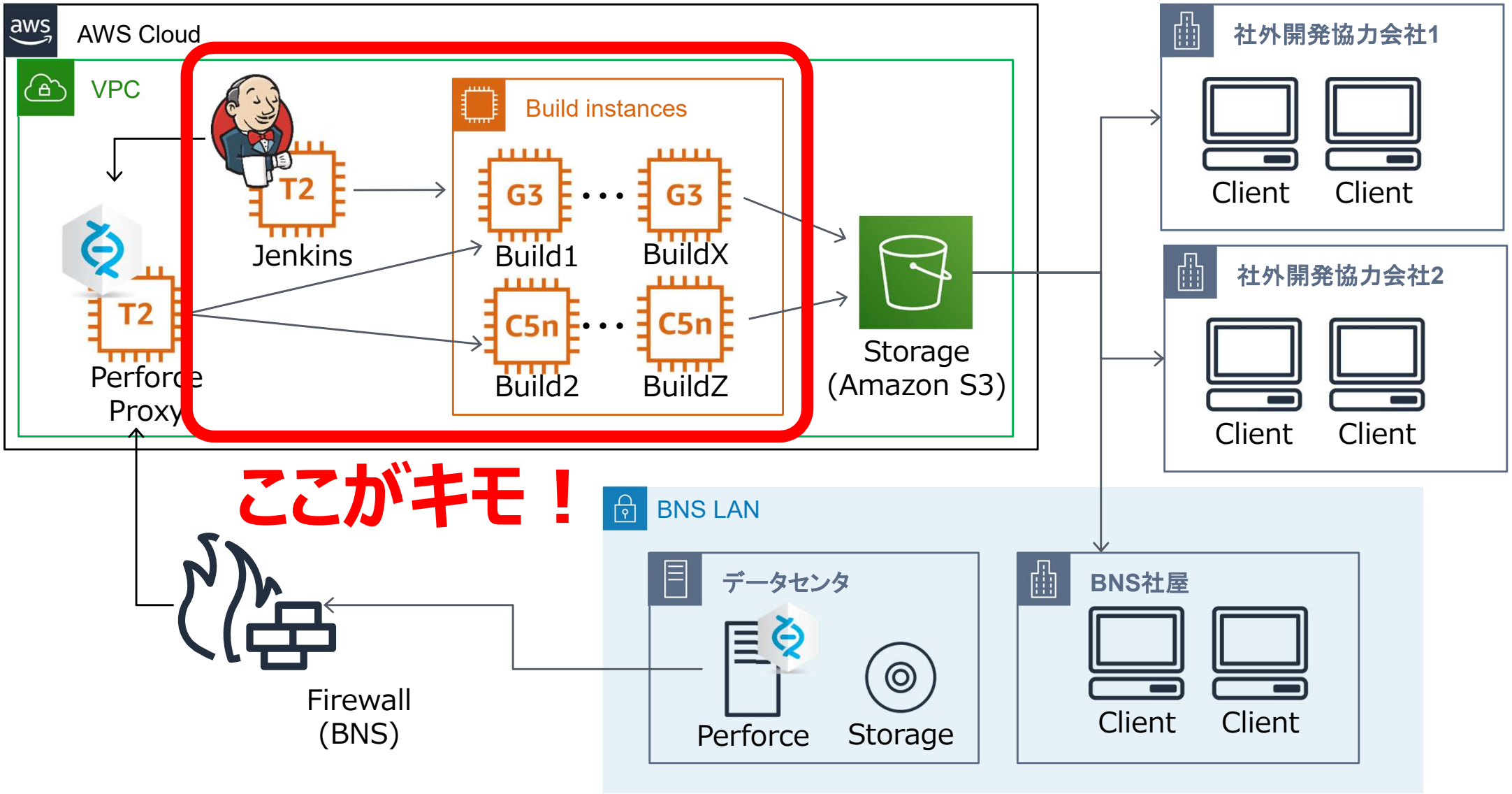
BANDAI NAMCO Studios

Bake：g4dn.16xlarge (64cores with GPU)



ビルドパイプライン：After

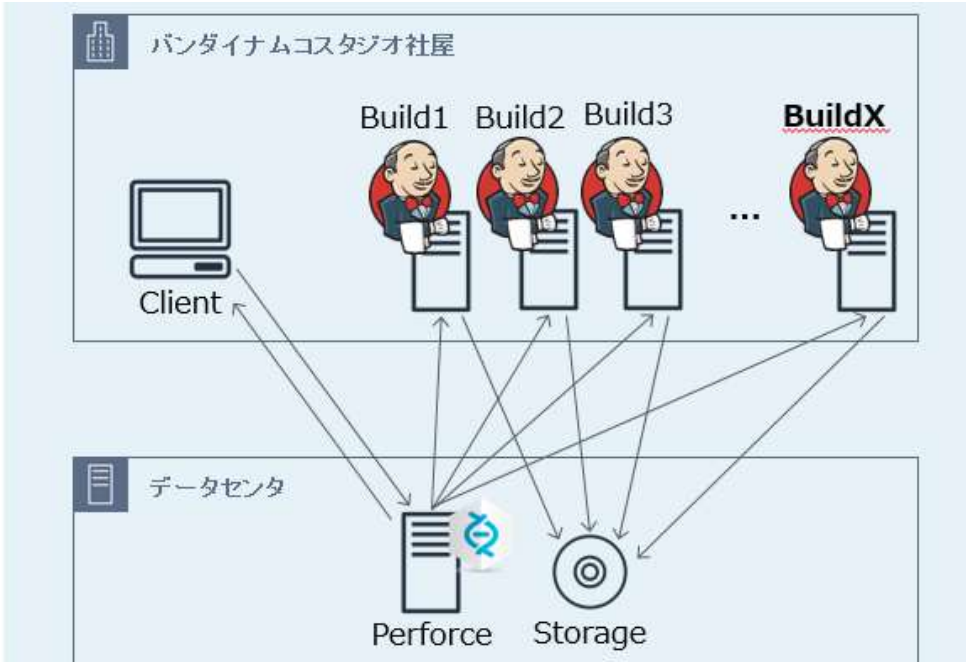
クラウド化したビルド構成



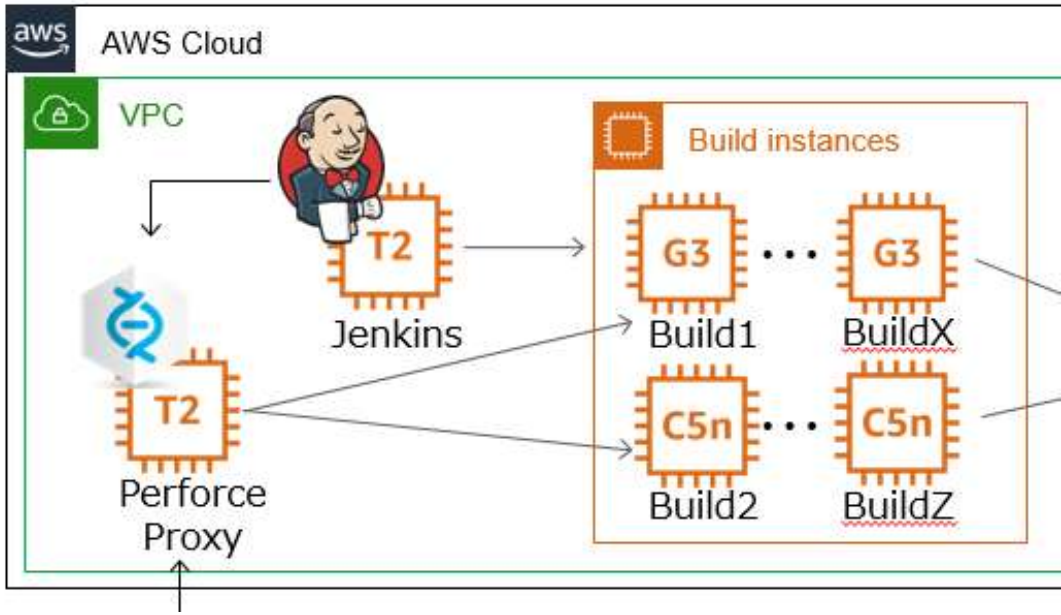
Jenkinsのジョブ解説

ポイントは**Jenkins1台運用**です

社内ビルド環境



クラウドビルド環境



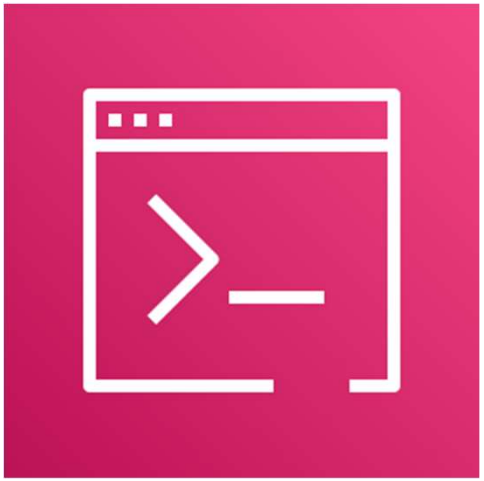
理由1. ビルド実行中以外はインスタンスを停止したい(**お金**)
理由2. 複数Jenkinsマシン管理から脱却したい(**手間削減**)

BANDAI NAMCO Studios

利用するツール

Jenkins

ビルドインスタンスの「起動・ビルドリモート実行・ステータスチェック・停止」を管理。Jenkinsインスタンス本体ではビルド実行しない。



AWS CLI

AWSのサービスをコマンドラインから操作・管理するツール。AWSコンソール操作をコマンドラインで操作できる。

AWS Systems Manager

AWSやオンプレミスのリソースを管理するツール。AWS CLIのssmコマンド経由でビルドスクリプトをリモート実行、ステータスや実行結果取得が可能。マネージドインスタンスとして登録する必要がある。

BANDAI NAMCO Studios

インスタンス起動とビルド実行

インスタンス起動

aws ec2 start-instances

--instance-ids インスタンスID
--region リージョン名

スクリプトのリモート実行

aws ssm send-command

--document-name AWS-RunPowerShellScript
--instance-ids インスタンスID
--region リージョン名
--parameters commands=スクリプトのパス executionTimeout=制限時間
--query Command.CommandId

デフォルト実行時間は1時間なので、ビルド実行時間が長い場合、”executionTimeout”を指定する必要がある(ハマった)

AWS System Managerのコマンドが返ってくる

コマンドID

Jenkinsのジョブ解説

BANDAI NAMCO Studios

```
:STEP1
REM インスタンス起動
for /f "usebackq" %%A in (`aws ec2 describe-instances --instance-id
%instance%  --region ap-northeast-1 --query
Reservations[].Instances[].State.Name --output text`) do set
InstanceState=%%A
if %InstanceState% == stopping aws ec2 wait instance-stopped --
instance-id %instance%  --region ap-northeast-1 --query
Reservations[].Instances[].State.Name --output text
aws ec2 start-instances --instance-ids %instance% --region ap-
northeast-1
aws ec2 wait instance-running --instance-ids %instance% --region ap-
northeast-1
ping -n 30 127.0.0.1
```

Jenkinsのジョブ解説

ビルド状況を確認

ビルド中のステータスを定期的に確認

aws ssm get-command-invocation

--command-id コマンドID
--instance-id インスタンスID
--region リージョン名
--output text
--query Status

Inprogress

コマンドIDのステータスが返ってくる

コマンドステータス

ビルド成功/失敗確認へ

Jenkinsのジョブ解説

BANDAI NAMCO Studios

```
:STEP2
REM ビルドコマンドを実行。終了までステータスチェックを繰り返す
aws ssm send-command --document-name AWS-RunPowerShellScript --
instance-ids %instance% --region ap-northeast-1 --parameters
commands=%command% executionTimeout=18000 --query Command.CommandId --
output text > tmp.txt
type tmp.txt
for /f "usebackq" %%A in (`type tmp.txt`) do set CommandId=%%A
del tmp.txt

:DoCmdStatChk
for /f "usebackq" %%A in (`aws ssm get-command-invocation --command-id
%CommandId% --instance-id %instance% --region ap-northeast-1 --output
text --query Status`) do set CommandStatus=%%A
ping -n 1 127.0.0.1
if %CommandStatus% == InProgress goto DoCmdStatChk
```

BANDAI NAMCO Studios

ビルド成功/失敗の判定とインスタンス停止

ビルドスクリプトの標準エラー出力確認

aws ssm get-command-invocation

--command-id **コマンドID**
--instance-id **インスタンスID**
--region **リージョン名**
--output text
--query **StandardErrorContent**

標準エラー出力結果が返ってくる

StandardErrorContent

インスタンス停止

aws ec2 stopt-instances

--instance-ids **インスタンスID**
--region **リージョン名**

Next ビルド結果に応じた処理を進める(e.g. Slackへ通知)

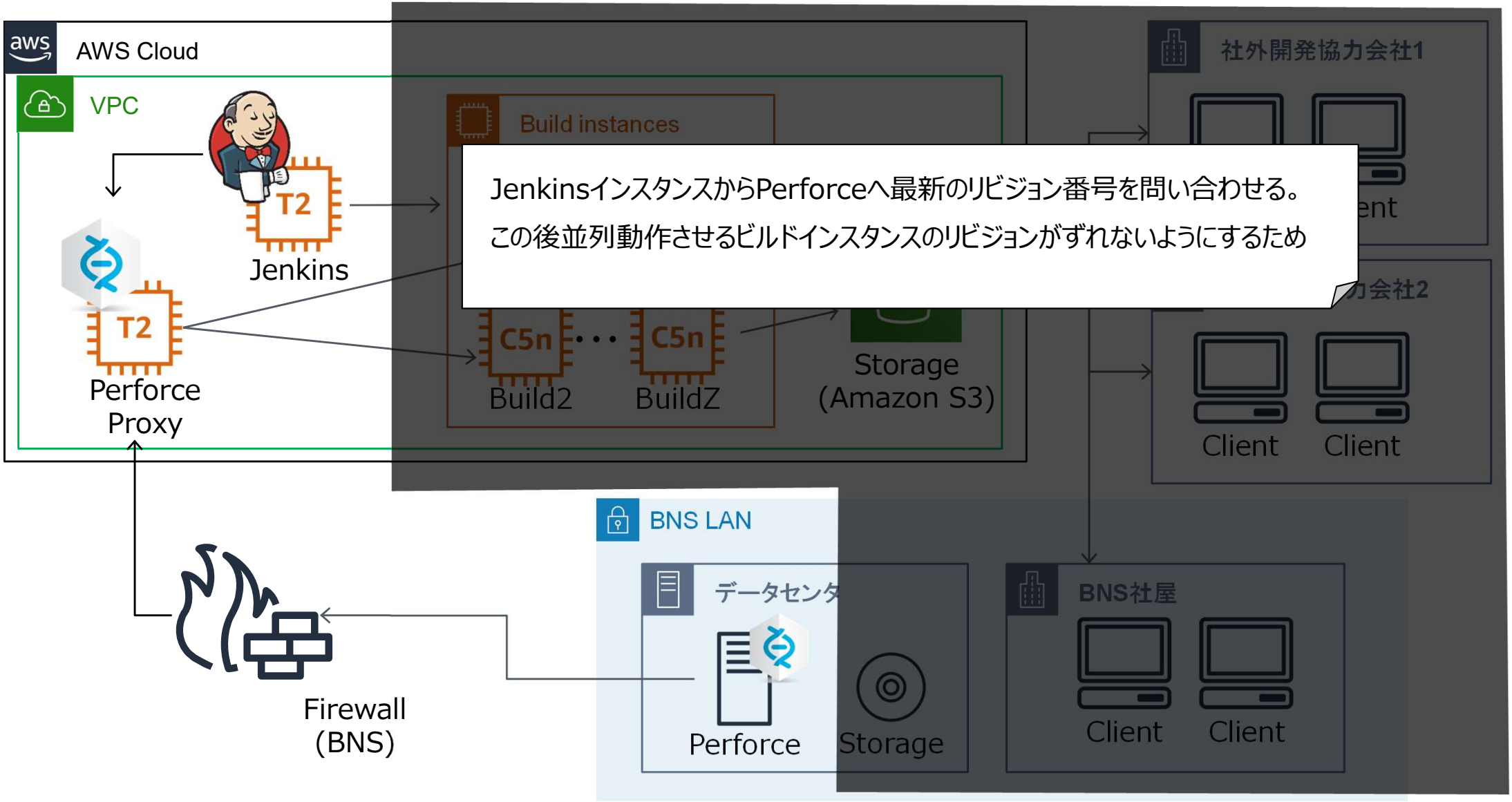
Jenkinsのジョブ解説

BANDAI NAMCO Studios

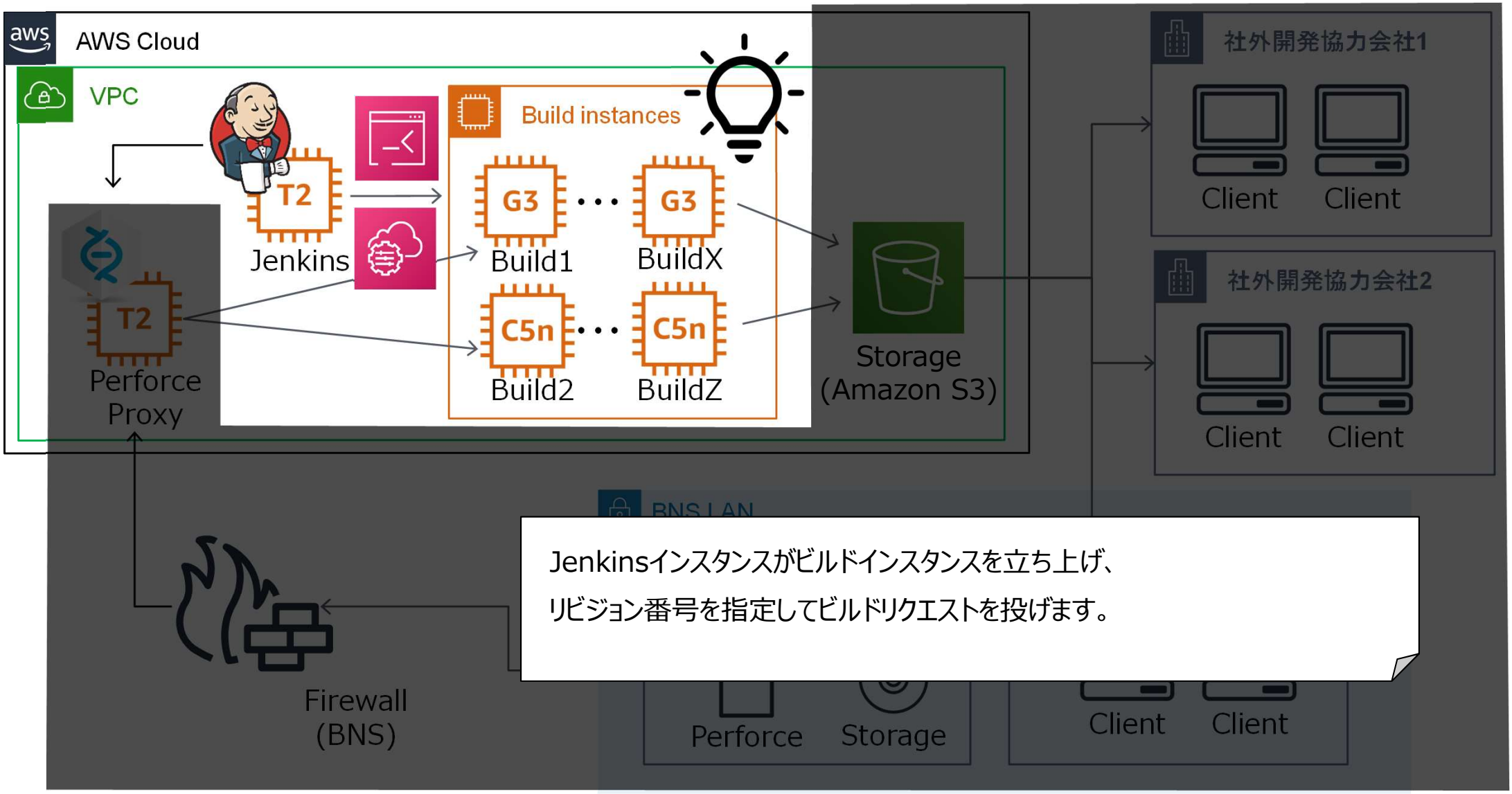
```
:STEP3
REM コマンド実行結果からビルド結果を判定。インスタンス終了
aws ssm get-command-invocation --command-id %CommandId% --instance-id
%instance% --region ap-northeast-1 --output text --query StandardOutputContent
> CmdSOC.log
aws ssm get-command-invocation --command-id %CommandId% --instance-id
%instance% --region ap-northeast-1 --output text --query StandardErrorContent >
CmdSEC.log
for /f "usebackq" %%A in (`findstr "." CmdSOC.log`) do set CmdSOC=%%A
for /f "usebackq" %%A in (`findstr "." CmdSEC.log`) do set CmdSEC=%%A

if %CmdSEC%_nul == _nul (
  goto JobSuccess
 ) else (
   goto JobFailed
 )
:JobSuccess
echo BUILD SUCCESS
aws ec2 stop-instances --instance-ids %instance% --region ap-northeast-1
exit 0
:JobFailed
echo BUILD FAILED
aws ec2 stop-instances --instance-ids %instance% --region ap-northeast-1
exit 1
```

クラウド化したビルドの流れ

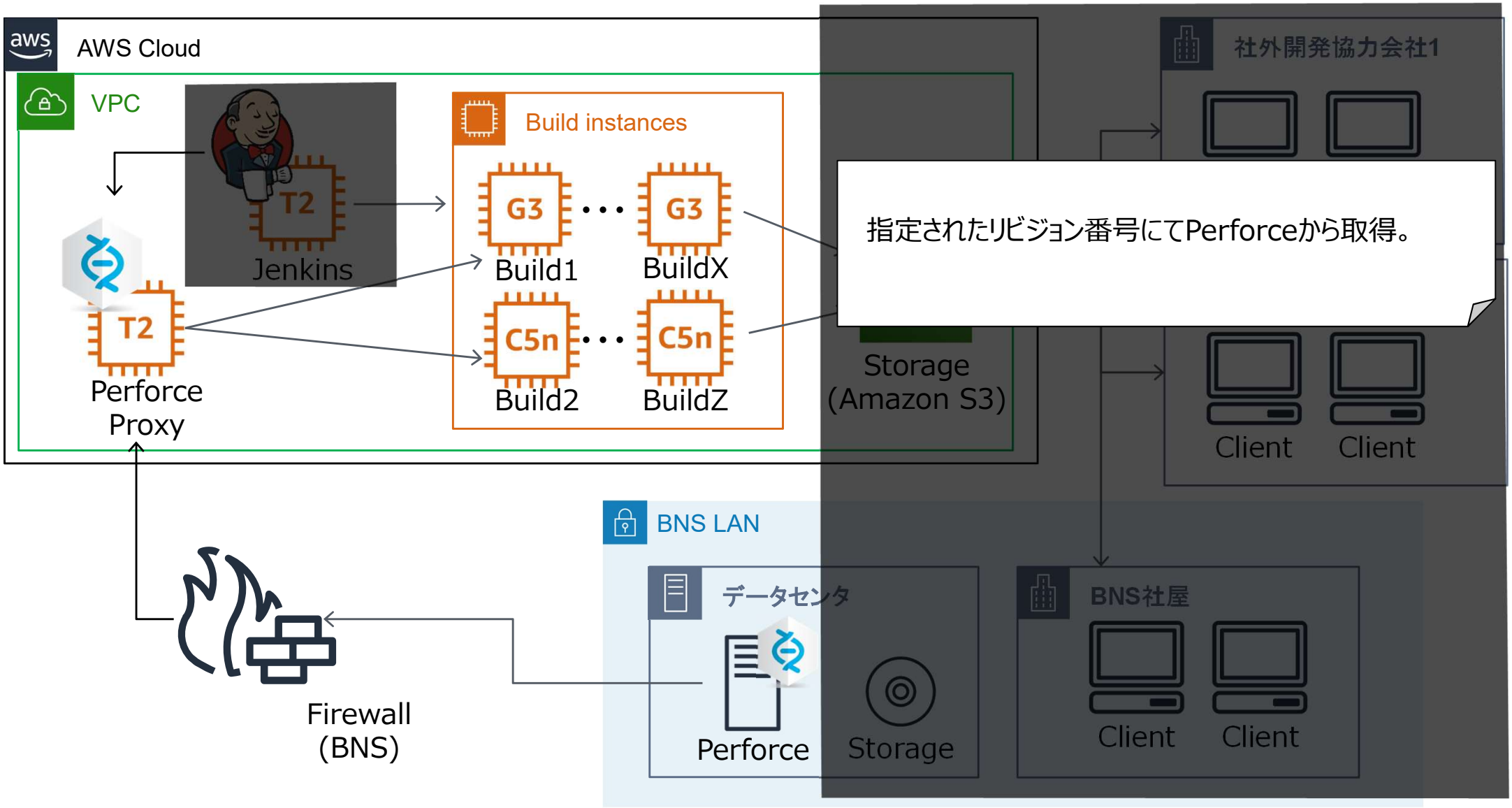


クラウド化したビルドの流れ

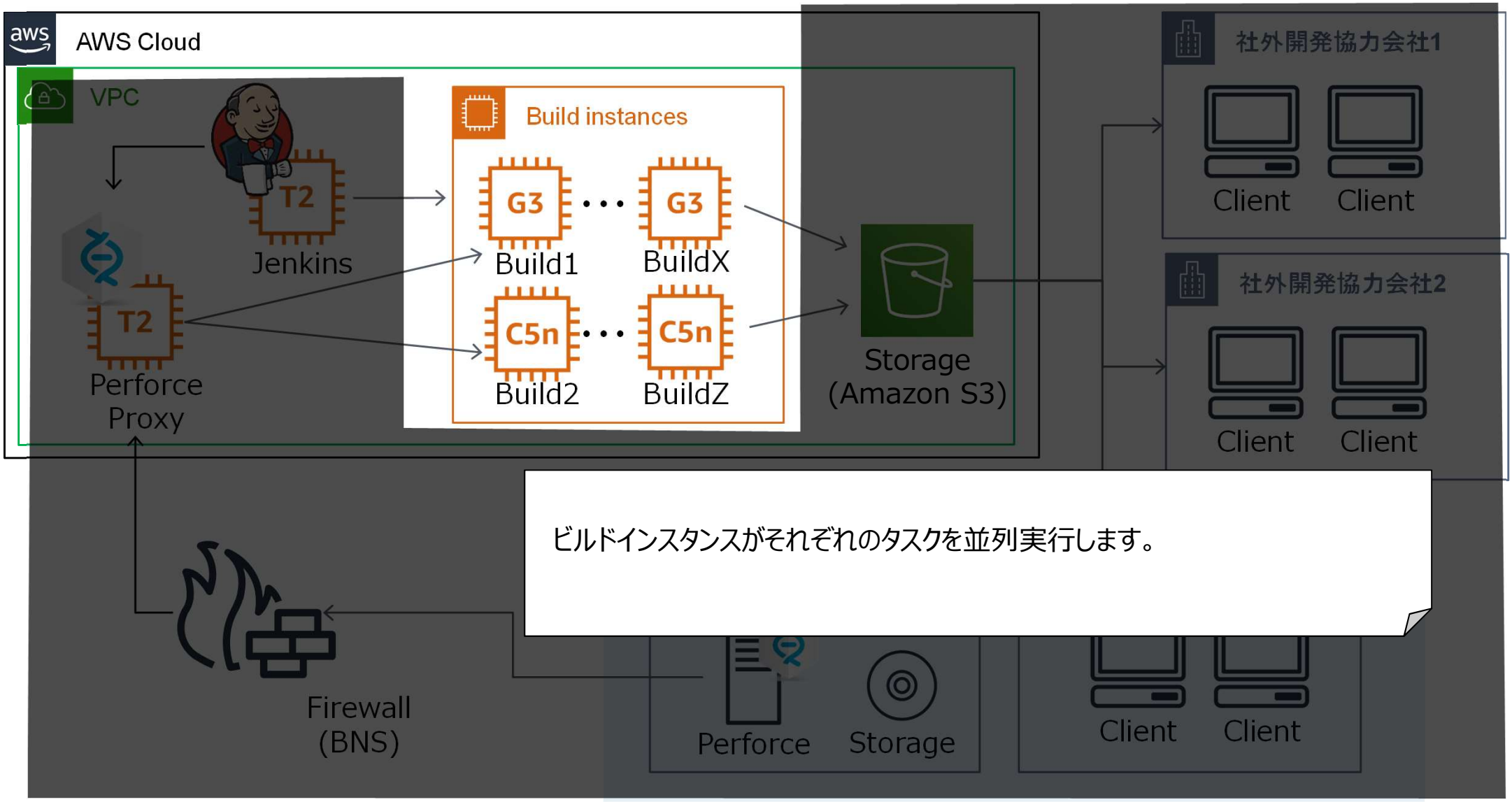


ビルドパイプライン：After

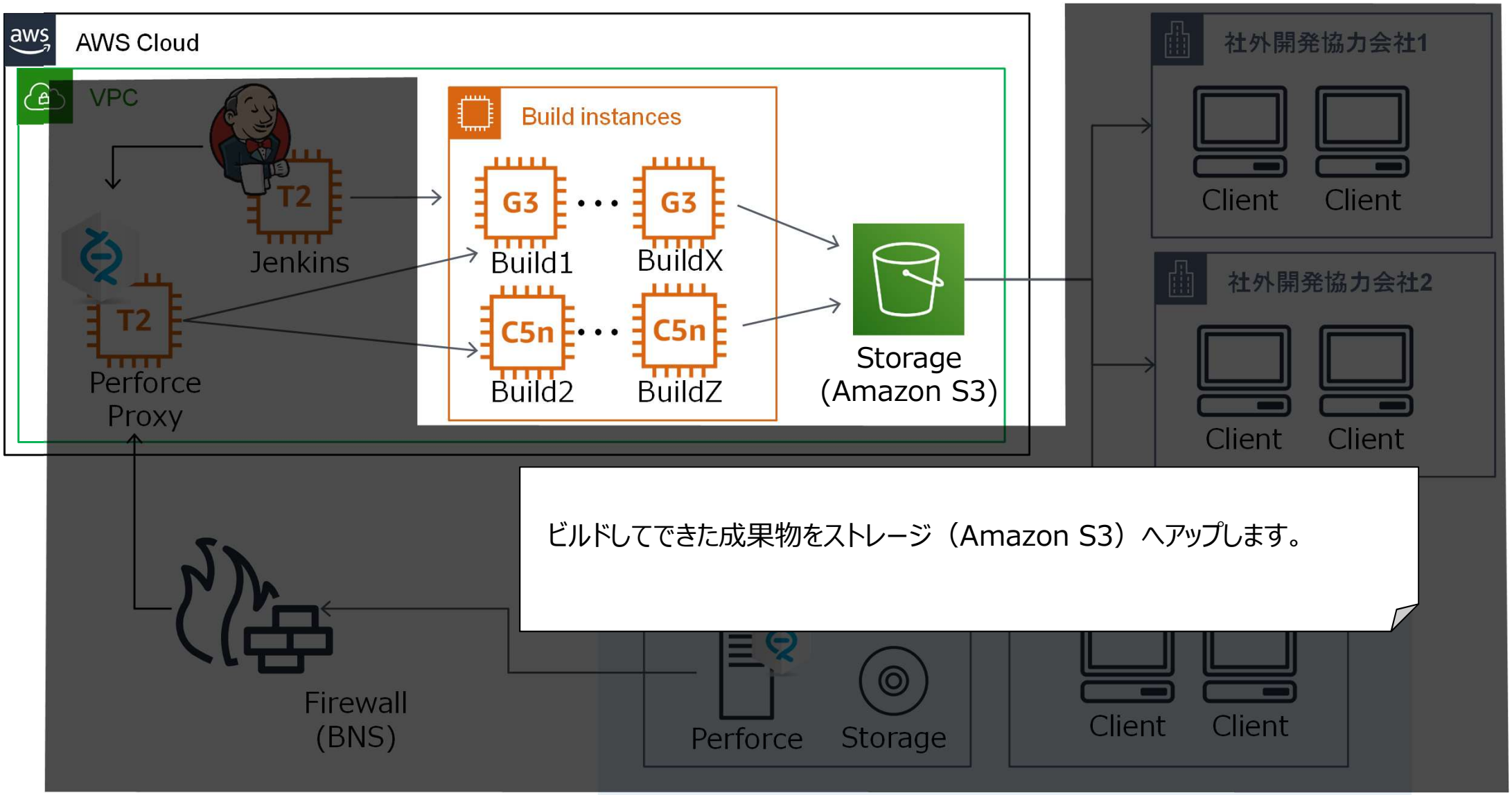
クラウド化したビルドの流れ



クラウド化したビルドの流れ

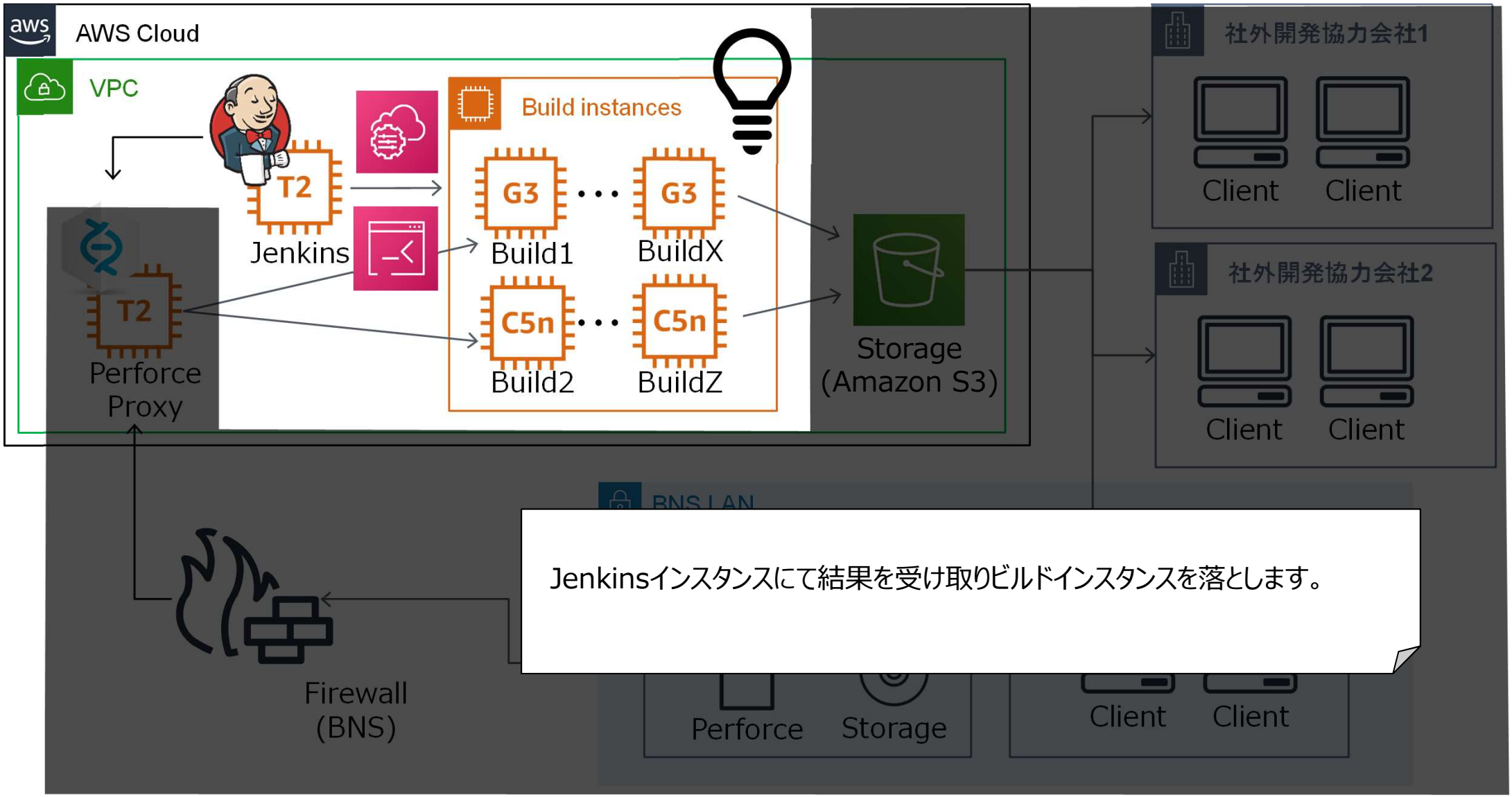


クラウド化したビルドの流れ



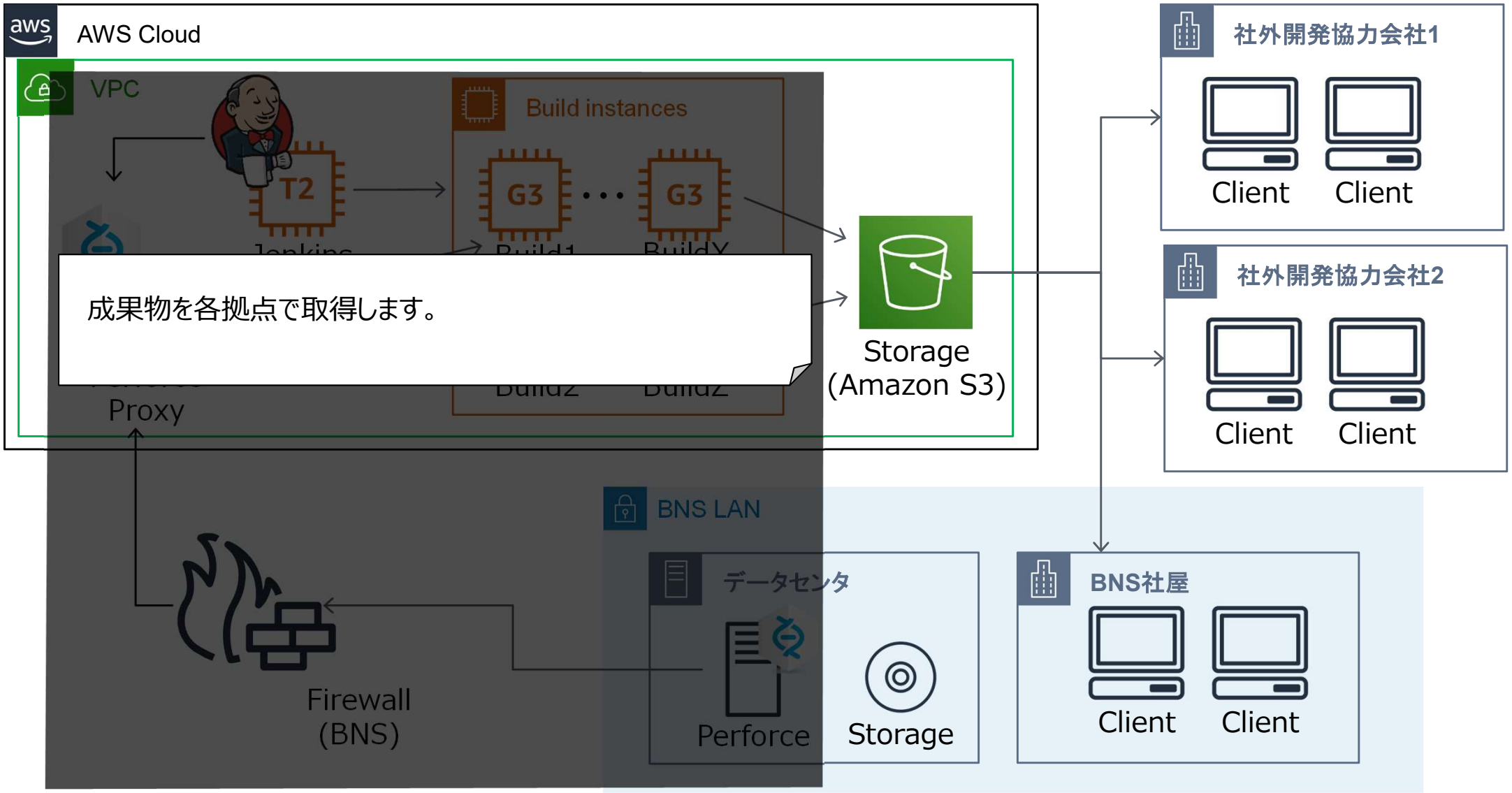
ビルドパイプライン：After

クラウド化したビルドの流れ



ビルドパイプライン：After

クラウド化したビルドの流れ



アジェンダ

BANDAI NAMCO Studios

1. BLUE PROTOCOLとは？
2. ゲーム開発でのビルドパイプライン解説
3. BLUE PROTOCOLのビルドパイプラインクラウド化した話
4. **クラウド化した結果や成果、わかった事**
5. 細かすぎて伝わらないAWS初心者のハマりどころ

クラウド化した結果や成果、わかった事

BANDAI NAMCO Studios

結果・成果・わかった事

150コア！

	ローカルPC	ローカルPC並列化 (IncrediBuild)	クラウド (c5.24xlarge)
Compile	約96分	約5分	約18分
Cook	約80分	約80分	約83分
Packaging	約62分	約66分	約25分
合計	約3.9時間	約2.5時間	約2.1時間

キャッシュ次第なので、
あまり参考にならない

コア数が多いほうが早い

IncrediBuildがなくてもクラウドで遜色ないくらいの速度は出る

クラウド化した結果や成果、わかった事

BANDAI NAMCO Studios

結果・成果・わかった事

	ローカルPC	ローカルPC並列化 (Unreal Swarm)	クラウド (g4dn.16xlarge)
CPU	Intel Core i7 4core	Intel Core i7 AMD Ryzen Threadripper x 2 4core + 32core x 2	Intel Xeon 64core
Bake Lighting	約4時間17分	約47分	約41分

ひとつのマップでこんなにも時間がかかってしまうため、
多くのマップを捌くためにはたくさんのインフラが必要になる。

クラウド化した結果や成果、わかった事

BANDAI NAMCO Studios

ランニングコスト

サービス	項目	インスタンスタイプ	単価/時(USD)	起動時間/日(h)	金額/30日(USD)
EC2	Jenkins (Windows)	t2.micro	0.0198	24	14.26
EC2	Build1(C5 Windows)	c5d.24xLarge	10.272	2	616.32
EC2	Build2(C5 Windows)	c5d.24xLarge	10.272	1.5	462.24
EC2	Build3(G4 Windows)	g4dn.16xlarge	8.819	1	264.57
EC2	P4P(Linux)	t2.small	0.0396	24	28.51

サービス	項目	ストレージタイプ	容量(GB)	利用時間/日	金額/30日
EBS	Build1のストレージ	gp2	1000	2	9.86
EBS	Build2のストレージ	gp2	1000	1.5	7.40
EBS	Build3のストレージ	gp2	1000	1	4.93
EBS	P4Pのストレージ	gp2	2000	24	236.71
EBS	Jenkinsのストレージ	gp2	30	24	3.55

サービス	項目	一回のサイズ(GB)	POSTリクエスト回数/日	GETリクエスト数/日	金額/30日
S3	成果物のアップロード先	10	1	10	7.50

合計：
1655.85 USD/月

構成：
- P4Proxy x1
- Jenkins x1
- Build x2
- Build(Bake) x1
- Strage(S3) x1

リージョン：
・アジアパシフィック(東京) ap-northeast-1

クラウド化した結果や成果、わかった事

BANDAI NAMCO Studios

検証から導入までのスケジュール感

2019年
- 11月　検証スタート
- 12月　AWS環境構築・ Jenkins無しでP4P経由でビルド速度計測

2020年
- 1月　ビルドパイプライン構成を決定
- 2月　Jenkinsジョブの作りこみ、ビルドスクリプトアップデート
- 3月　進捗なし(COVID-19関連)
- 4月　進捗なし(COVID-19関連)
- 5月　CloudFormationによるBLUE PROTOCOL用AWS環境構築
- 6月　プロジェクトでの実運用開始

クラウド化した結果や成果、わかった事

BANDAI NAMCO Studios

AWS良かった事・悪かった事（構築してみた所感）

■ GOOD

- マシン調達の待ち時間　**数週間→数分**
- ビルド性能　**社内の分散環境とほぼ同等**
- アクセスコントロール　**取引先とのデータ授受等も柔軟**
- ネット上のナレッジ　**公式はもちろんユーザー記事も多い**
- 手厚いサポート　**ソリューションアーキテクト様に感謝**

■ BAD

- 従量課金制　**稼働時間が長いと金額が怖い**
- Visual Studio　**使えない(ライセンス問題)**
- キャッチアップ大変　**サービスの追加更新が早い(うれしい悲鳴)**
- コスト試算　**複数サービス使うと試算が大変**

アジェンダ

BANDAI NAMCO Studios

1. BLUE PROTOCOLとは？
2. ゲーム開発でのビルドパイプライン解説
3. BLUE PROTOCOLのビルドパイプラインクラウド化した話
4. クラウド化した結果や成果、わかった事
5. **細かすぎて伝わらないAWS初心者のハマりどころ**

細かすぎて伝わらないAWSハマりどころ

BANDAI NAMCO Studios

細かいけど解決に地味に時間がかかった事を紹介

■ **AWSアカウント作成どうしよう？**
→BNSでCloudpackを包括契約していたので利用

■ **クラウドビルドパイプラインの構成で悩む**
→AWSサービスの選択肢が多いのでソリューションアーキテクトに相談

■ **WindowsインスタンスにRDP(リモートデスクトップ)できない…**
→会社のファイアーウォールで deny されていた

■ **インスタンス作成から数か月…突然RDPできなくなった…**
→Windowsのパスワード更新期限過ぎていた。無期限にしよう

細かすぎて伝わらないAWSハマりどころ

BANDAI NAMCO Studios

細かいけど解決に地味に時間がかかった事を紹介

■ **金額の試算が難しい**(そして偉い人からはまず費用感を聞かれる)
- →1回の「ビルド時間」「やり取りするデータ量」「通信する回数」がわかれば大体のサービスは試算できた
- ご参考：1日1回ビルド：1,743.54 USD/月
  1日2回ビルド：3,116.36 USD/月

■ **リージョンはどこがよい？**
- スピード重視→ アジアパシフィック (東京) ap-northeast-1
- 安さ重視→ 米国西部 (オレゴン) us-west-2
  - 東京より約1割安く、転送速度は約2倍遅い
- 新機能→米国東部 (バージニア北部)us-east-1
  - 障害件数が多い傾向？

細かすぎて伝わらないAWSハマりどころ

BANDAI NAMCO Studios

細かいけど解決に地味に時間がかかった事を紹介

■ **ビルド開始後1時間で"TimeOut"終了してしまう**
→ssmコマンドで"executionTimeout=<起動時間>"を付ける

■ **万が一ビルドインスタンスが停止せずに高額課金になる事を防ぎたい**
→Lambda + Cloud Watch で長時間稼働インスタンスを停止

■ **EBSのIOPSは足りている？**
→Cloud Watchでスループットを確認するのが正攻法。ざっくり調べるなら「バーストバランスをチェックしてクレジットが100から減っていないか」を確認すればOK。

将来の展望・次のアクション(大井)

BANDAI NAMCO Studios

パッケージする前にアセットのベリファイを行う

・パッケージエラーの事前検知
・後ろの工程ほどエラーが発生すると辛い

完成したパッケージの自動テスト

・AIなども活用出来たらカッコいい

クラウドでBotを大量投入

・大規模オンラインゲームの為同時接続をテストするのが大変

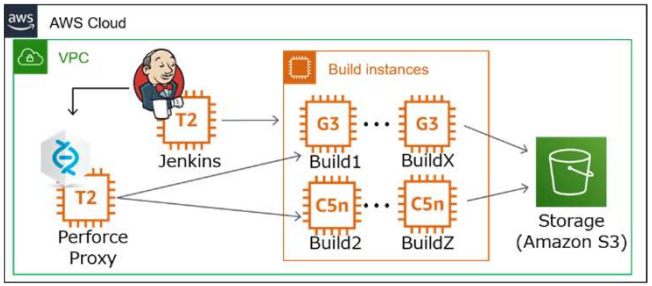
将来の展望・次のアクション(吉田)

クラウドビルドパイプラインを社内のプロジェクトへ共有・展開

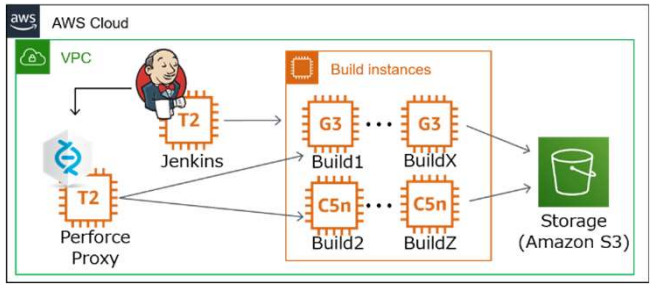
・CloudFormationを利用してP4P,Jenkins,ビルドマシンがすぐに使える状況のサブセットを提供。

在宅勤務に合わせたクラウド共存型の開発環境

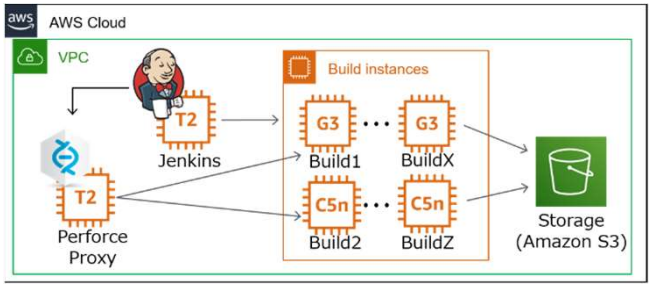
・オフィス・自宅・社外(協力会社)等、どこからでもクラウド上の開発環境にアクセス



Project A



Project B



Project C

まとめ

BANDAI NAMCO Studios

社内LANに閉じてた開発を(一部)クラウドに移行することができた

- 社内分散ビルドとほぼ変わらないビルド時間
- Perforceの取得はP4P(キャッシュ)を使えば変わらない
- 今まで通りJenkinsとWindowsを使えればOK

クラウドのメリット

- インスタンス追加の手軽さ
- 物理Jenkins管理からの手放
- アクセスコントロール

宣伝

株式会社バンダイナムコスタジオでは
キャリア採用を積極的に行っています