Amazon EKS における EC2 スポットインスタンスをもっと身近に

クラウド最適化コンテナ編 ～コンテナ x Graviton2 x スポットインスタンスによるコスト最適化～

アマゾン ウェブ サービス ジャパン株式会社
落水 恭介

2021.08.26

© 2021, Amazon Web Services, Inc. or its Affiliates.

自己紹介

落水 恭介 (Ochimizu Kyosuke)

コンテナスペシャリストソリューションアーキテクト

- SIer
- 教育業界ベンチャー
- Cloud Integrator
- サポートチーム / アマゾン ウェブ サービス ジャパン
- 現在のロール

好きな AWS サービス:

Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

本日お話すること

- Amazon EKS におけるコスト最適化
- EC2 スポットインスタンスの活用
  - スポット中断への対応
  - スポットのキャパシティ不足対応
  - スポット/オンデマンドの Pod 配置

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

Amazon EKS におけるコスト最適化

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

ECS / EKS におけるコストの最適化とは

ECS / EKS のコスト最適化 = データプレーンのコスト最適化

	Amazon ECS	Amazon EKS
コントロールプレーン (クラスター)	料金は発生しない	毎時 0.10USD
データプレーン (コンテナ実行環境)	• EC2 の利用料金 • Fargate の利用料金	• EC2 の利用料金 • Fargate の利用料金

※: 2021/08/26 時点での AWS の利用料金

© 2021, Amazon Web Services, Inc. or its Affiliates.

データプレーンのコスト最適化

オートスケーリング

必要な分だけ

EC2 インスタンス・コンテナを実行

適切なサイジング

コンテナに適切な

CPU やメモリを割り当て

購入オプション

スポットインスタンスや

Savings Plans の活用

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

データプレーンのコスト最適化

オー

EC2 インス

本日は購入オプションについてお話します

購入オプション

スポットインスタンスや

Savings Plans の活用

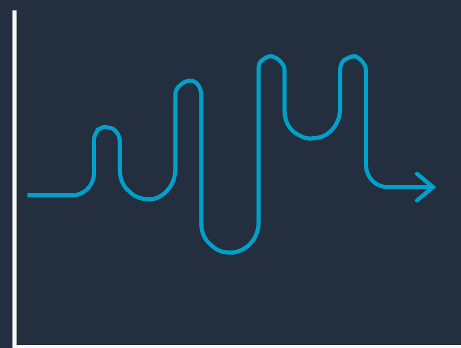
© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

Amazon EC2 の購入オプション

オンデマンドインスタンス

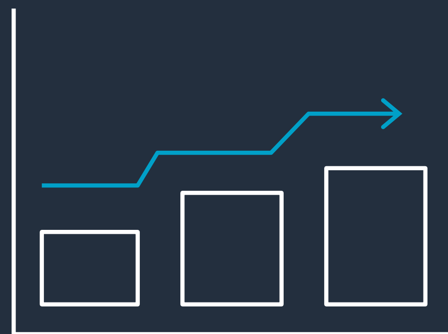
長期のコミット無し、使用分への支払い (秒単位/時間単位)。Amazon EC2 の定価



スパイクするようなワークロード

リザーブドインスタンス (RIs)

1年/3年の長期コミットに応じた大幅なディスカウント価格



一定の負荷の見通しがあるワークロード

Savings Plans

RI と同等のディスカウントに加え、さらなる柔軟性

ワークロードをまたがる柔軟性

スポットインスタンス

EC2 の空きキャパシティを活用し最大 90 % 引き。中断あり

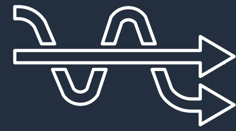
中断に強く、様々なインスタンスタイプを活用できるワークロード

EC2 インスタンスとしての性能に違いはない

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

AWS Fargate の購入オプション



- 柔軟な設定の選択肢 – 50 の CPU/メモリ設定から
- Fargate Spot は、通常の Fargate の価格と比較して 最大70% の割引
  - 現時点 (2021-08-26) では ECS でのみ利用可能
- Savings Plan は、最大 約 50% の割引 (3年コミット/全額一括前払い)、下限は 約 15 %

	Fargate	**Savings Plan**	**Fargate Spot**
	料金	料金	料金
1 vCPU	$0.05056	$0.0394368 (22% Off)	$0.01584035 (69% Off)
1 GB メモリ	$0.00553	$0.0043134 (22% Off)	$0.00173254 (69% Off)
		期間 1 年、全額前払い	

※ 東京リージョンでの 2021-08-24 時点での料金

AWS Fargate の料金: https://aws.amazon.com/jp/fargate/pricing/

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

EKS における Pod 実行環境の選択肢

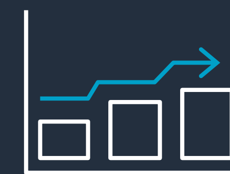
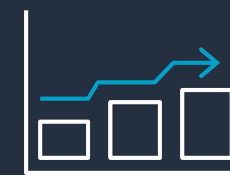
セルフマネージド型ノード	ユーザーが自身で EC2 インスタンス / Auto Scaling グループを管理
マネージド型ノードグループ	EKS がプロビジョニングやライフサイクルを管理
Fargate	サーバーレスのコンピューティングエンジン

https://docs.aws.amazon.com/eks/latest/userguide/eks-compute.html

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

Pod 実行環境と購入オプションの組み合わせ

	スポット インスタンス	Savings Plans	リザーブド インスタンス (RIs)
セルフマネージド型 ノード			
マネージド型 ノードグループ			
Fargate			

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

Pod 実行環境と購入オプションの組み合わせ

	スポット インスタンス	Savings Plans	リザーブド インスタンス (RIs)
セルフマネージド型 ノード			
マネージド型 ノードグループ			
Fargate			

本日はスポットインスタンスの
活用についてお話しします

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

EC2 スポットインスタンスの活用

© 2021, Amazon Web Services, Inc. or its Affiliates.

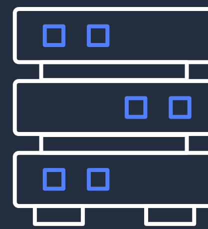
aws

スポットインスタンスの利用において考慮するポイント

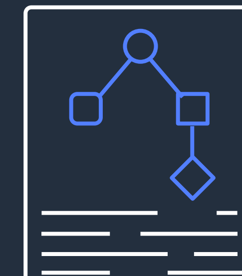
- スポットインスタンスを利用すること自体は簡単
  - ➢ [セルフマネージド型ノード]　スポットを利用する Auto Scaling グループを作成
  - ➢ [マネージド型ノードグループ]　キャパシティータイプに “スポット” を指定

- 実際の運用においては様々なポイントを考慮する必要がある

スポット中断への対応



スポットのキャパシティ不足対応



スポット/オンデマンドの Pod 配置

© 2021, Amazon Web Services, Inc. or its Affiliates.

スポット中断への対応

セルフマネージド型ノードの場合

Node Termination Handler (NTH) を利用する

- 以下のようなイベントを検知してノードの cordon/drain を行う
  - ➢ スポットインスタンスの中断
  - ➢ キャパシティリバランス通知
  - ➢ EC2 インスタンスの予定されたイベント
  - ➢ Auto Scaling グループ (ASG) のスケールイン
- インスタンスメタデータ (IMDS) / キューの 2 モードが利用可能

Feature	IMDS Processor	Queue Processor
K8s DaemonSet	✅	❌
K8s Deployment	❌	✅
Spot Instance Interruptions (ITN)	✅	✅
Scheduled Events	✅	✅
EC2 Instance Rebalance Recommendation	✅	✅
ASG Lifecycle Hooks	❌	✅
EC2 Status Changes	❌	✅
Setup Required	❌	✅

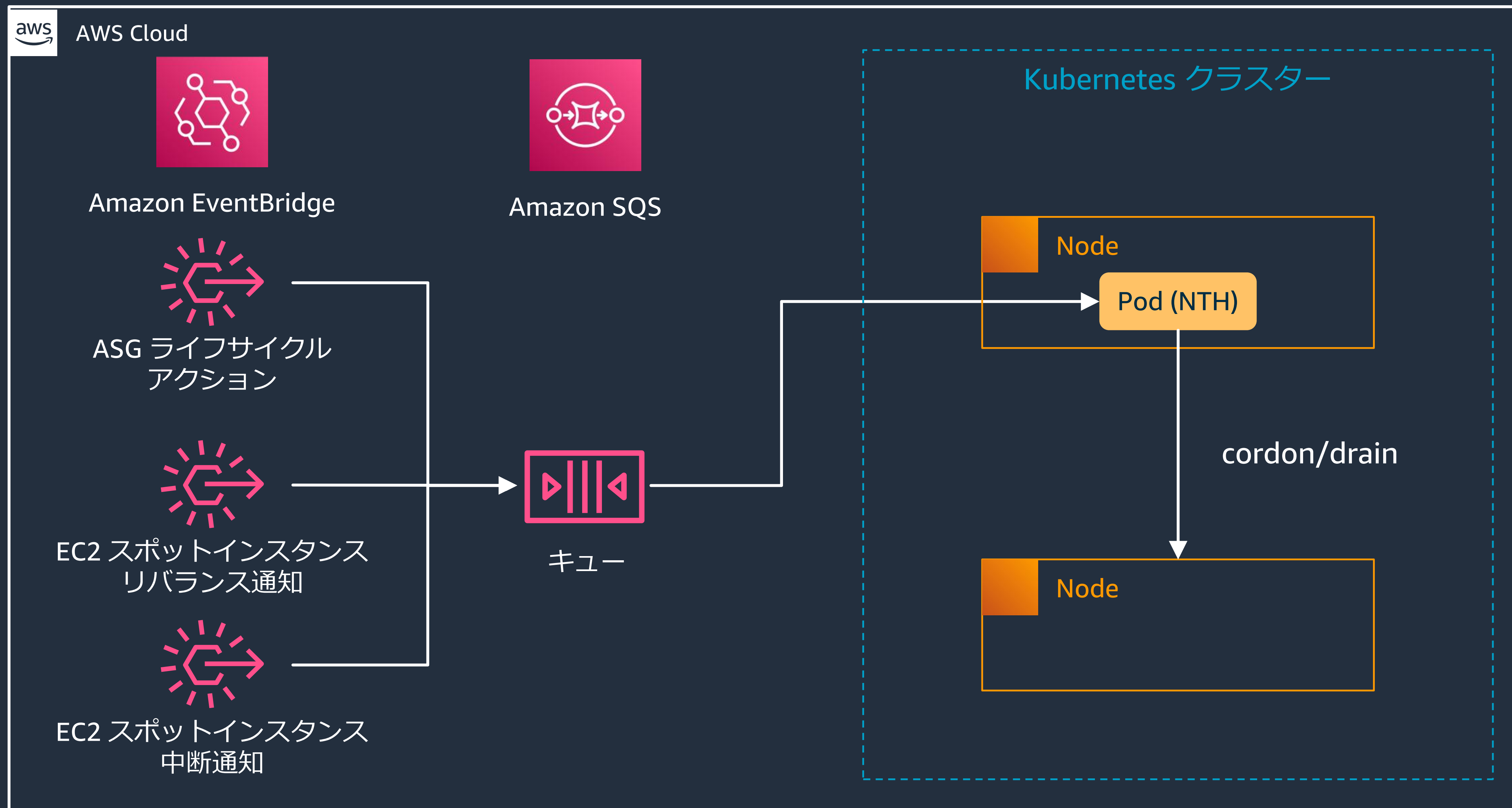
https://github.com/aws/aws-node-termination-handler

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

Node Termination Handler (キュープロセッサーモード)

各種イベントを SQS キューに集約して Node Termination Handler が処理を行う

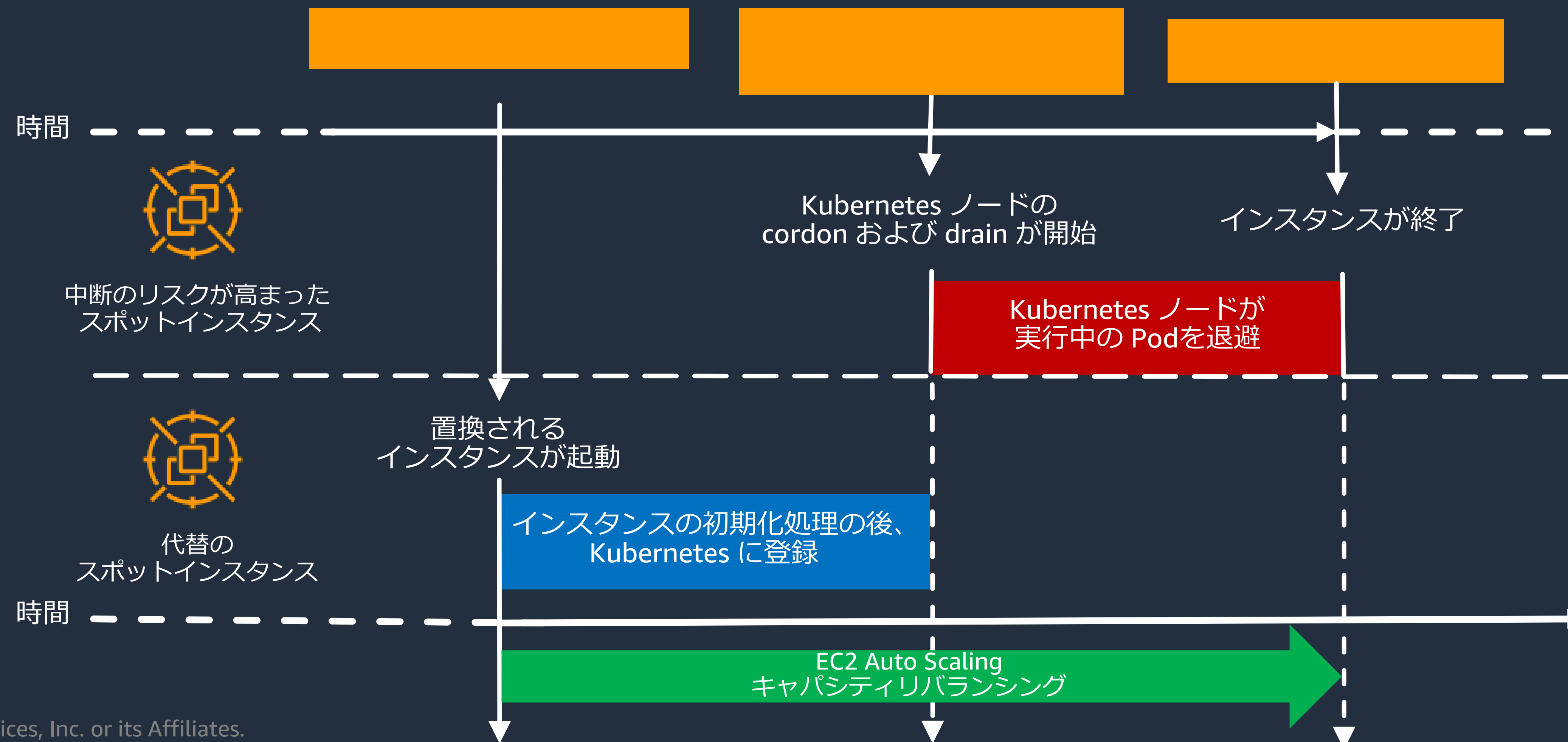


© 2021, Amazon Web Services, Inc. or its Affiliates.

スポット中断への対応

マネージド型ノードグループの場合

- Node Termination Handler の導入など、お客様のアクションは不要
- リバランス通知を受け取ると、新規ノードの起動および通知対象のノードの drain を行う



© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

[補足] コンテナアプリケーション側での対応

- これらの仕組みは、あくまで Pod に SIGTERM を送るまでの仕組み
- SIGTERM を受けとった後の graceful shutdown や checkpoint の書き出しなどはコンテナアプリケーション側でシグナルハンドリングを行う必要がある

```python
import signal, time, os

def shutdown(signum, frame):
    print('Caught SIGTERM, shutting down')
    # Finish any outstanding requests, then...
    exit(0)

if __name__ == '__main__':
    # Register handler
    signal.signal(signal.SIGTERM, shutdown)
    # Main logic goes here
```

```javascript
process.on('SIGTERM', () => {
  console.log('The service is about to shut down!');

  // Finish any outstanding requests, then...
  process.exit(0);
});
```

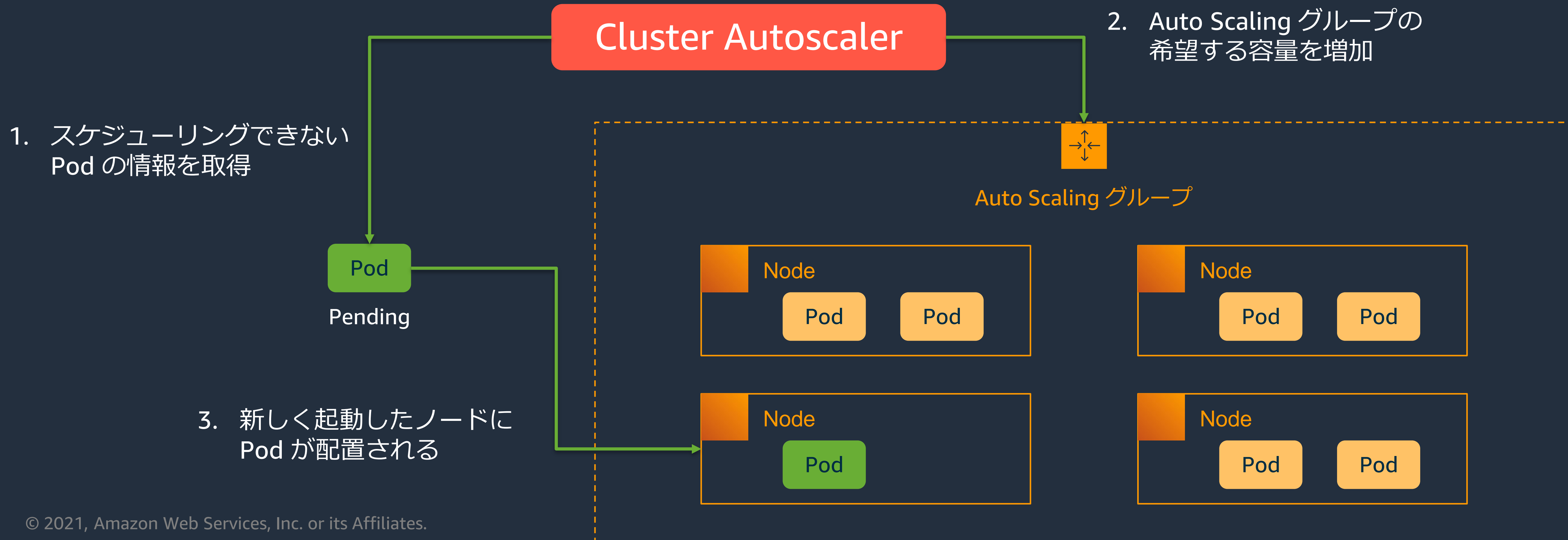
© 2021, Amazon Web Services, Inc. or its Affiliates.

スポットのキャパシティ不足対応

セルフマネージド型ノード / マネージド型ノードグループ共通

Cluster Autoscaler (priority expander) を利用

例) Cluster Autoscaler によるスケールアウト



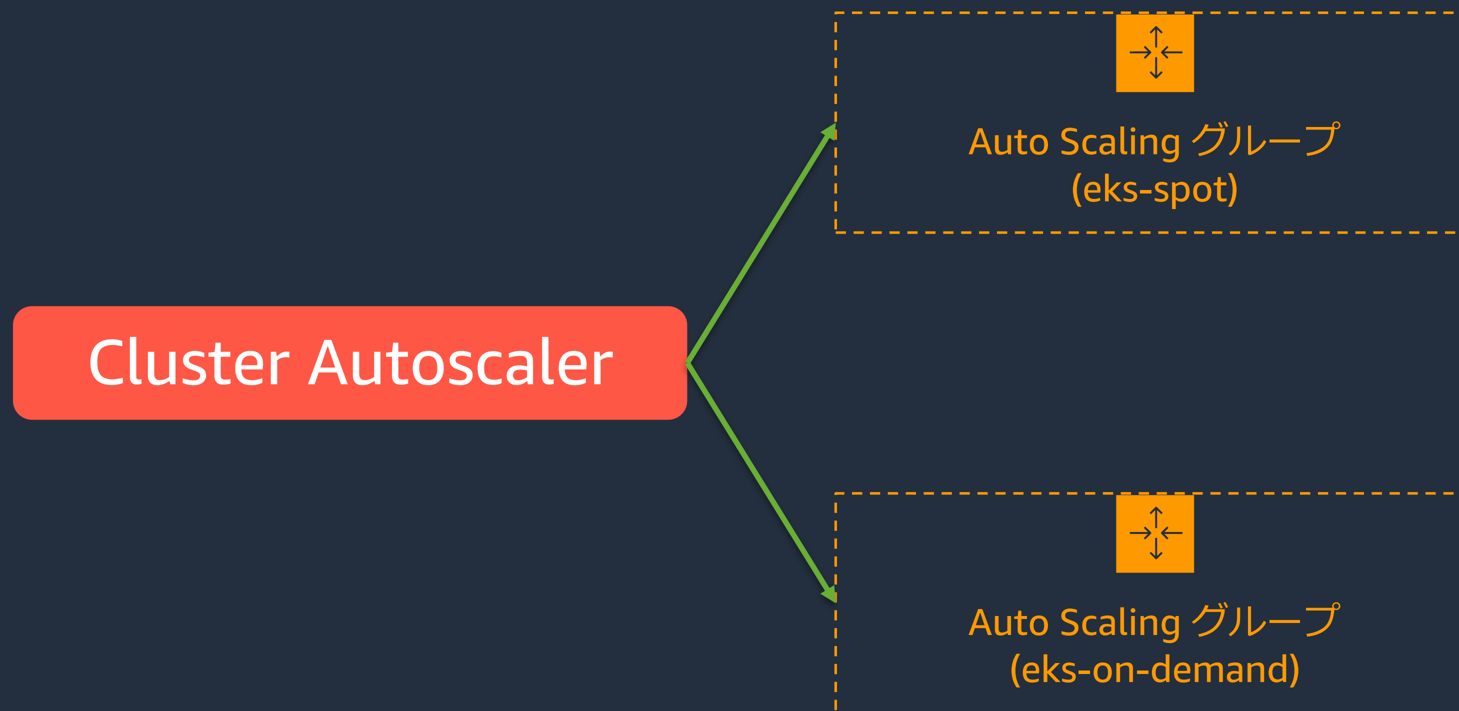
© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

Cluster Autoscaler の priority expander

ユーザーが定義した優先度に基づきスケールアウトする Auto Scaling グループを選択

1. ASG ”eks-spot“ のスケールアウトを試みる



2. 一定時間内にスケールアウトが完了しない場合、
   ASG “eks-on-demand” のスケールアウトを試みる

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: cluster-autoscaler-priority-expander
  namespace: kube-system
data:
  priorities: |-
    20:
      - eks-spot.*
    10:
      - eks-on-demand.*
```

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

スポット活用における Cluster Autoscaler の考慮事項 (1)

Auto Scaling グループでは、同じサイズの vCPU とメモリリソースを持つインスタンスタイプを柔軟に組み合わせての使用を推奨

EC2 instance selector

```
$ ec2-instance-selector ¥
> --memory 4 ¥
> --vcpus 2 ¥
> --cpu-architecture x86_64 ¥
> -r ap-northeast-1
c5.large
c5a.large
c5d.large
t2.medium
t3.medium
t3a.medium
```

eksctl での利用例

```yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: cluster
  region: us-west-2

nodeGroups:
- name: ng
  instanceSelector:
    vCPUs: 2
    memory: “4” # 4 GiB, unit defaults to GiB

managedNodeGroups:
- name: mng
  instanceSelector:
    vCPUs: 2
    memory: 2GiB
    cpuArchitecture: x86_64 # default value
```

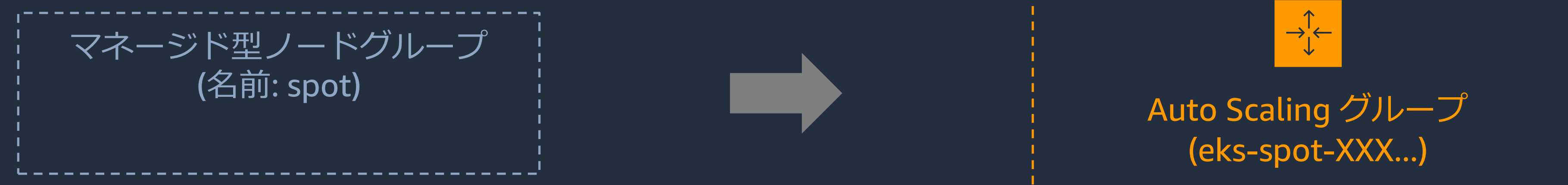
© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

スポット活用における Cluster Autoscaler の考慮事項 (2)

- Kubernetes バージョン 1.20 以前の EKS クラスターでは、マネージド型ノードグループの作成する Auto Scaling グループ名がランダムなため priority expander の指定が難しい
- Kubernetes バージョン 1.21 の EKS クラスターより Auto Scaling グループの命名規則が変更され、 priority expander の指定が容易に

Kubernetes バージョン	命名規則
1.20 以前	`eks-<uuid>`
1.21	`eks-<managed-node-group-name>-<uuid>`



© 2021, Amazon Web Services, Inc. or its Affiliates.

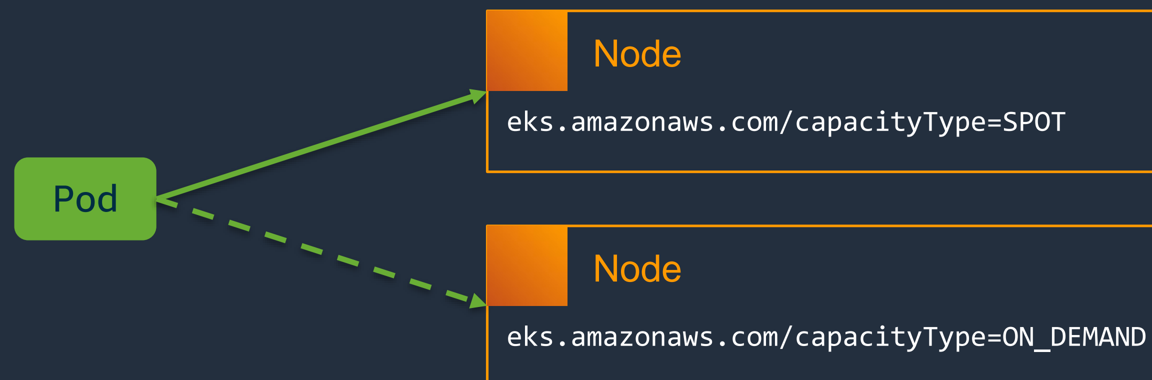
aws

スポット/オンデマンドの Pod 配置

Taint や Affinity を利用して、Pod のスケジューリングを制御する

例) 優先的にスポットインスタンスへ配置

```
affinity:
  nodeAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 1
      preference:
        matchExpressions:
        - key: eks.amazonaws.com/capacityType
          operator: In
          values:
          - SPOT
```



例) 必ずオンデマンドインスタンスへ配置

```
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: eks.amazonaws.com/capacityType
          operator: In
          values:
          - ON_DEMAND
```

© 2021, Amazon Web Services, Inc. or its Affiliates.

[補足] マネージド型ノードグループ x スポットインスタンス

マネージド型ノードグループのキャパシティータイプに “スポット” を選択した場合、以下のベストプラクティスが適用された状態で作成される

- eks.amazonaws.com/capacityType: SPOT という Kubernetes ラベルが付与
- 配分戦略に capacity-optimized を使用
- キャパシティリバランスがオプトイン済み
- リバランス通知を受け取ると、新規ノードの起動および通知対象のノードの drain を実施

https://aws.amazon.com/jp/blogs/news/amazon-eks-now-supports-provisioning-and-managing-ec2-spot-instances-in-managed-node-groups-jp/

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

まとめ

- Amazon EKS におけるコスト最適化
    - ➢ データプレーンの「オートスケーリング」「適切なサイジング」「購入オプション」
- EC2 スポットインスタンスの活用
    - スポット中断への対応
        - ➢ [セルフマネージド型ノード] Node Termination Handler
        - ➢ [マネージド型ノードグループ] 対応不要
    - スポットのキャパシティ不足対応
        - ➢ Cluster Autoscaler の Priority Expander
    - スポット/オンデマンドの Pod 配置
        - ➢ Taint や Affinity の利用

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws

Thank you!

© 2021, Amazon Web Services, Inc. or its Affiliates.

aws