

EKS Version Rollback

「戻れる」はどう実装されたのか

2026-07-01 GA | 全リージョン・全パーティション | 追加料金なし



従来の回避策とそのコスト

年 3 回のマイナーリリース → 定期アップグレードは必須。しかし control plane のダウングレードは OSS Kubernetes の設計外



Blue/Green

クラスタを 2 面持ちして
段階的にトラフィックを移行

→ インフラ費が実質倍増



手動スナップショット

クラスタ状態を手動で保存し
問題発生時に復元を試みる

→ 工数と復元の不確実性



アップグレード先送り

リスクを承知で古い
バージョンに留まる

→ CVE 露出・延長サポート課金

使い方と Readiness Insights



EKS Version Rollback とは

アップグレード後 **7 日以内** なら、
control plane を **1 つ前のマイナーバージョン** に戻せる

- 例 : 1.34 → 1.33 に戻す
- etcd データ・ワークロード・PV は保持したまま
- Auto Mode ならデータプレーン（ノード）も自動で戻る
- 専用 API ではない — 既存の UpdateClusterVersion に古いバージョンを渡すだけ（アップグレードの逆再生）



**本番アップグレードの
セーフティネット**
DR 要件への対応にも



戻れるから早く上げられる
CVE 露出短縮・extended
support 費用回避

使い方 : CLI とコンソール

Bash

```
# 1.33 に上げたクラスタを 1.32 に戻す (アップグレードと同じ API)
aws eks update-cluster-version --name my-cluster --kubernetes-version 1.32

# 事前に Readiness を確認
aws eks list-insights --cluster-name my-cluster --filter category=ROLLBACK_READINESS
```

コンソール : Actions → 「Roll back cluster version」 → Rollback summary (現行 / 戻り先 / Insights 集計) → Initiate



Rollback Readiness Insights — 安全チェックの実体

アップグレード後 7 日間だけ自動生成 (category: ROLLBACK_READINESS)

- API 互換性 (新バージョン限定 API/ フィールドの使用)
- クラスタヘルス
- kubelet / kube-proxy のバージョンスキュー
- EKS マネージドアドオンの互換性

	PASSING	OK。そのまま進める
	WARNING	助言のみ。ブロックしない
	ERROR	ブロック
	UNKNOWN	ブロック (評価不能)

--force で bypass できるのは **insight の警告まで**。
後述の disruption budget は破れない

ノードが先、コントロールプレーンが後

スキューポリシー「**kubelet は API サーバーより新しくてはならない**」。CP を先に戻せない

1. ノードをロールバック
Karpenter drift 置換で N-1 に
(クラスタは ACTIVE のまま)

2. コントロールプレーンを
ロールバック
全ノード完了後に実行

製品判断

PDB / do-not-disrupt /
budget は
--force でも破らない

時間がかかる時の 3 択:

待つ

budget が緩めば数日で完了

budget を緩める

disruption を許容して加速

CancelUpdate

Auto Mode 専用。stop であって
元に戻すではない

ノードタイプ別のロールバック対応

ノードタイプ	ロールバック方法	補足
Auto Mode	EKS が自動でノード→CP の順に実行	budget を尊重しつつ進行
Managed Node Group	顧客が UpdateNodegroupVersion で先に戻す	その後 CP をロールバック
セルフマネージド / ハイブリッド	顧客が手動で対応	自前で kubelet を先に戻す必要あり
Fargate	非対応	kubelet を単独で戻せない構造

二段構えの安全チェック

同期チェック（API 受付時・即答）

- config の存在
- 1.28 以上
- バージョン整合
- 7 日以内
- サポート期限

非同期チェック（Cluster Insights）

- 監査ログとライブ照会で評価
- API / フィールド互換性
- バージョンスキュー
- アドオン互換性
- ERROR ならブロック

役割分担の考え方 : **顧客がアクションできるもの**は Insights で見せる、**どうにもならないもの**（7 日超過等）は API 側で即答する

制約・落とし穴・ベストプラクティス



Kubernetes バージョンロールバック

コントロールプレーンのアップグレードを 7 日以内に取り消せる | 2026年7月1日 GA | 全リージョン | 追加費用なし

仕組み

- お客様がクラスターをアップグレード（例: 1.33 → 1.34）
- 7 日以内に問題が出たら UpdateClusterVersion を 1.33 で呼ぶ
- EKS が自動 Insights でロールバック可否を評価
- チェック通過でコントロールプレーンが前バージョンに戻る
- etcd データ、ワークロード、PV はすべて保持される
- Auto Mode クラスターの場合:
 - EKS がコントロールプレーンより先にノードをロールバック
 - NodePool の disruption budget と PDB を尊重
 - タイムアウトは設定可能（既定 12 時間、最大 7 日）
 - CancelUpdate API でロールバックを中断可能
- MNG / セルフマネージド / ハイブリッドノードの場合:
 - 先にノードをロールバック（UpdateNodegroupVersion）
 - その後にコントロールプレーンをロールバック

ロールバック準備 Insights（自動チェック）

- API 利用の互換性（フィールド単位の変更も含む）
- クラスターのヘルスに関する問題
- kubelet のバージョンスキュー（ノードと CP）
- kube-proxy のバージョン互換性
- EKS マネージドアドオンのバージョン互換性
- Auto Mode: NodePool budget、PDB、do-not-disrupt

Insight ステータスの挙動

- PASSING = そのまま実行して問題なし
- WARNING = 参考情報のみ、ブロックしない
- ERROR = ロールバックをブロック（解消するか --force）
- UNKNOWN = ロールバックをブロック（評価不能）
- --force は Insight チェックのみ回避（budget は回避不可）



Kubernetes バージョンロールバック

推奨すべきケースと IaC の考慮事項

ロールバックを推奨すべきお客様

- 破壊的変更を懸念してアップグレードを先送りしている
 - -> ロールバックにより先送りする理由がなくなる
- 安全性のためブルー/グリーンでアップグレードしている
 - -> インプレース+ロールバックがより簡単・低コスト
- 大規模フリート（100 クラスター以上）を運用している
 - -> ロールバック+bake でフリート展開を高速化
- 延長サポート期間のさらなる延長を要望している
 - -> 早期に自信を持ってアップグレードできる
 - -> 延長サポート料金を削減できる
- ベストプラクティス: 先に CP をアップグレードし 1 週間 bake、その後ノードを更新。ロールバック期間は維持される。

IaC の制約（CloudFormation / Terraform）

- CFN のタイムアウト: リソースあたり 36 時間
- Terraform Enterprise/Cloud: 約 24 時間でタイムアウト
- IaC タイムアウトを超えると CFN は no-op として扱う
 - -> クラスターがドリフト状態に（テンプレートと実際が不一致）
 - -> ロールバックは必ず明示的に開始する必要がある
- 推奨: rollbackConfig の timeoutMinutes を使用する
- CFN スタックのロールバック（テンプレートの復元）では、クラスターのバージョンは戻らない。明示的な実行が必要。

押さえておくべき主な制約

- 期間は 7 日間のみ（アップグレード完了後から）
- 1 バージョンのみ（N → N-1、複数スキップ不可）
- Fargate: 非対応（先に Pod を削除するか --force）
- アドオン: 自動ロールバックされない（個別に管理）
- アップグレード済みクラスターのみ（新規作成クラスターは不可）
- EKS 機能の非互換: --force では回避できない



制約と落とし穴 早見表

やりたいこと	可否	理由
2 バージョン戻す	×	3 世代離れた API サーバー間の互換保証がない
新規作成したクラスタを戻す	×	戻る場所の記録がない
1.27 以前のクラスタ	×	サポート対象は 1.28 以上
--force で PDB 無視	×	force は insight のみ。 budget は常に尊重
Fargate	×	kubelet を単独で戻せない
アドオンも一緒に戻る	×	UpdateAddon で個別に戻す
Terraform/CFN と併用	△	タイムアウト衝突に注意 → timeoutMinutes で調整。 CFN テンプレの巻き戻しはロールバックをトリガーしない（常に明示操作）
extended support 版に戻る	○	可能だが extended 課金が発生

rollback 窓を活かすアップグレード手順

- 1 アドオンを N-1/N/N+1 クロス互換バージョンに揃える
- 2 コントロールプレーンだけアップグレード
- 3 **最大 7 日 bake**— Insights を監視、新バージョン限定 API の採用は控える
- 4 問題なければノードをアップグレード

問題発生時 : データプレーンだけの問題ならノードのみ戻す / 両方ならノード→CP の順で戻す

rollback はセーフティネットであり、定常ワークフローではない。テストを置き換えるものではない



まとめ

- 既存 API の逆再生という最小の顧客体験で、8 年分の「一方向前提」を壊した機能
- ルールはすべて「3 フィールドの記録 + 二段構えのチェック」で説明できる
- 想定顧客課題
 - アップグレードを先送りしている
 - blue-green のコストに悩んでいる
 - extended support 延長を繰り返している

Q&A

