EKS Cluster Upgrades for Substrate

Document Level Classification

100

Introduction

Because Substrate accounts/environments come built with AWS EKS Kubernetes clusters, keeping these clusters updated is important. This document provides information on what to expect as a Substrate customer when your Kubernetes cluster needs to be upgraded and how to handle situations where manual intervention may be needed before a cluster can be upgraded.

CLM and CLM Upgrade Scheduler

CLM (*Cluster Lifecycle Manager*) is a unified API service capable of performing all Substrate cluster-related activities (for details see Cluster Lifecycle Manager). One key activity is to perform Substrate cluster version upgrades; however, before any upgrade, clusters are monitored and checked to ensure they are in an upgrade-eligible state. Addressing the issues raised by these checks are a shared responsibility: some are managed by us (Cloud Hosting) whereas others must be fixed by the team who owns it (or rather the stewards - probably you!).

CLM Upgrade Scheduler is a service that extends *CLM* to manage the scheduling of Substrate cluster upgrades and monitoring of clusters' Upgrade Ready state (for details see CLM - Scheduler). *CLM Upgrade Scheduler* identifies *CLM* health checks that typically require cluster stewards to intervene before a cluster can be safely upgraded and automatically sends out notifications.

Manual Intervention Notifications via Slack

If you are a Substrate customer and also a cluster steward, you may have received a Manual Intervention Notification via Slack from the *CLM Upgrade Scheduler*, which has identified one or more cluster checks that are failing and need to be fixed to allow the cluster to be upgraded by us. **Your cluster is not failing** but some things need you attention before it can be marked as upgrade-eligible and scheduled for upgrade.

This article provides the high-level steps you should go through to identify and fix the issues highlighted by the *CLM Upgrade Scheduler*. Additionally, it will point you toward getting support if needed.

Handling Kubernetes Cluster Upgrade Manual Intervention Notifications

The notification messages asking for manual intervention will be based on one or more health checks monitored by CLM and will be listed within the message body. However, providing more details regarding the health checks is not practical in Slack messages so a link is provided to a web page providing a more detailed view.

Important Considerations

Addressing the issues related to your `Pod`s depend on how the affected resources are deployed in your cluster:

- In the case of deployment through a Helm chart, **it is imperative to update the impacted resources within the Helm chart and proceed with deployment**. Note that if you choose to modify the resources using tools like `kubectl`, the Scheduler will persistently register the health check failure until a permanent resolution is applied.
- **Orphan/Standalone Pods** will be permanently terminated during upgrade and can't be recreated (as they don't belong to, for example, a `Deployment`). In edge cases where a standalone pod is necessary, **please contact** `#cloud-ops-support-ext` with a justified reason so that the affected cluster can be handled in a different way.
    - However, it is important to remind that standalone pods are an anti-pattern in itself when using Kubernetes and therefore it's highly encouraged to not use them for any workload that cannot be immediately and permanently terminated at any time.
- For standalone deployments i.e. without a Helm chart (for example, through `kubectl apply -f ...`), besides applying the remediation, **ensure the removal of the `kubectl.kubernetes.io/last-applied-configuration` annotation from your resource** or otherwise the health check failure will remain. A quick way to remove

the annotation is by running the following command (note the dash "-" at the end!):

```
kubectl annotate <RESOURCE_TYPE> <RESOURCE_NAME> -n <NAMESPACE> kubectl
```

Health check failure(s) will determine the actions required. The following provides an overview of each health check and either the steps to take to resolve or a link to a separate runbook:

Error	**Description**	**Remediation**
pod(s) with not "running" status detected	One or more Pods in the cluster that are scheduled but aren't running as expected.	Pod(s) need to be removed or fixed to run normally.
not allowed mounts discovered	One or more Pods mounting non-allowed paths from the host system. The specific unallowed path is described in the error.	Removing the unallowed paths.
ingress(es) with invalid rules or wide open security groups discovered	One or more Ingresses have associated AWS security groups with too wide open / invalid rules.	Adjusting the Security Group rules.

Error	Description	Remediation
resources with deprecated APIs discovered	One or more K8s resources are using old API versions that are going to disappear in the upcoming K8s versions.	API version(s) for the affected resource(s) are updated (for example, `autoscaling/v2beta2` would become `autoscaling/v2` or the latest available API version). In many cases, updating your epic-app version may fix all these problems if the outdated APIs originate from this chart.
PDB will disrupt node draining	One or more PDBs aren't correctly configured and will not allow proper node draining.	The affected pdb needs to be adjusted. Run the following command to check current configuration: `kubectl describe pdb -n <namespace> <pdb-name>` • Ensure *Allowed Disruptions* is not set to 0.

Planned Upgrades

Substrate clusters are all AWS EKS (Elastic Kubernetes Service) clusters, and AWS limits the Kubernetes versions it supports on EKS clusters and so clusters need to be regularly upgraded to maintain support and security.

AWS only support a set of Kubernetes versions (normally the latest five major versions) at any time, and so to ensure clusters stay functional and

supported by AWS we much upgrade in line with AWS's upgrade schedule.

The earlier we get upgrades done, the better for everyone.

Expected Downtime

To perform a cluster upgrade, nodes need to be rotated one or multiple times, depending on the gap between the current cluster version and the target version. This process involves migrating workloads from older nodes to newer ones. In most cases, downtime is avoided as Pods are moved gradually, allowing applications to get ready again without interruption.

However, certain applications, like certain Flink clusters, may experience downtime due to their specific requirements. In such cases, special handling is necessary and coordinated between Cloud Hosting and the team's stewards. If your cluster includes workloads susceptible to this issue, please contact `#cloud-ops-support-ext` to ensure appropriate attention is given to your cluster.

Additional Slack Notifications

Notifications are sent out as early as possible during each Substrate wide upgrade cycle, managed by the Cloud Hosting team, to give cluster stewards time to get things fixed and avoid delays to upgrades.

However, notifications will be sent multiple times (with a delay between messages) until all required fixes are performed. The aim is to remind cluster stewards of the actions needed, and we try no not to spam anyone.

If you are working on rectifying any checks highlighted by a notification, then you can ignore any other messages. As soon as all checks are rectified, *CLM Upgrade Scheduler* will stop sending out any more notification messages and instead will send a final one with the upgrade date for you to acknowledge.

Support

For support with your Substrate cluster regarding the manual intervention actions, support queries should be directed to the support Slack channel `#cloud-ops-support-ext`. Please note that your own steward channel is not monitored!
