**ELC - Remediation plan of 'ConfigMap with secrets' control**

**Submitted to**

Text, logo

Description automatically generated

**By**

A logo with colorful dots

Description automatically generated

**Wipro Technologies**

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date of Revision** | **Description of Change** | **Reason for Change** | **Reviewed By** |
| 1.1 | 02-Sep-24 | Initial Draft | NA | Mihirkumar Jani |
|  |  |  |  |  |
|  |  |  |  |  |

**Author/Reviewer/Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Designation** | **Responsibility** | **Date** |
| Mihirkumar Jani | Consultant | Author | 02-Sep-24 |
| Kannan Kuppusamy | Technical Lead | Reviewer | 13-Nov-24 |
| Felix Jebamani | Lead Consultant | Reviewer | 13-Nov-24 |
| Salama, Omar | Lead | Reviewer | 19-Dec-24  [CST-1252](https://jira.esteeonline.com/browse/CST-1252) |
| Jamshid Abedi | Executive Director | Approver |  |

**Table of Contents**

[Objective of the Document 3](#_Toc176451528)

[1.0 Purpose and Scope 3](#_Toc176451529)

[1.0 Observation of Aqua 'ConfigMap with secrets' control scan result. 4](#_Toc176451530)

[2.0 Remediation plan of 'ConfigMap with secrets' 5](#_Toc176451531)

[3. Azure Key Vault Provider for Secrets Store CSI Driver: 6](#_Toc176451532)

[4. Reference document links 7](#_Toc176451533)

# Objective of the Document

This document is related to the “Remediation plan of ConfigMap with secrets”

# 1.0 Purpose and Scope

The purpose of the remediation plan is to store secrets in External keystore - Azure Key Valt as per best practice and provide strong security to the secrets like Kubernetes ConfigMap.

This applies to all of the Estee Lauder Companies Inc. (the “Company”) Azure AKS clusters environment.

# 1.0 Observation of Aqua 'ConfigMap with secrets' control scan result.

As per the AquaSec ‘ConfigMap with secrets’ control scan results, we observed that ConfigMap secrets are exposed as High Risk. And below are the observations shown

Below patterns secrets should not be part of the Kubernetes ConfigMap and the secrets should be stored in Azure key vault.

patterns := [

"(?i)(password\\s\*(=|:))",

"(?i)(pw\\s\*(=|:))",

"(?i)(pass\\s\*(=|:))",

"(?i)(pword\\s\*(=|:))",

"(?i)(passphrase\\s\*(=|:))",

"(?i)(passwrd\\s\*(=|:))",

"(?i)(passwd\\s\*(=|:))",

"(?i)(secret\\s\*(=|:))",

"(?i)(secretkey\\s\*(=|:))",

"(?i)(appSecret\\s\*(=|:))",

"(?i)(clientSecret\\s\*(=|:))",

"(?i)(aws\_access\_key\_id\\s\*(=|:))",

"(?i)(pswrd\\s\*(=|:))",

"(?i)(token\\s\*(=|:))",

"(?i)(pwd\\s\*(=|:))",

]

As per the scan, Azure AKS clusters environment by Aqua ‘ConfigMap with secrets' control.

Observed below secrets as High Risk and scan result are present in ConfigMap in the AKS environment:

# 1.1 Secrets exposed as part of the ConfigMap with Secrets

*secret\_key, token, password, accessKeySecret, secretKeySecret, SONARQUBE\_JDBC\_PASSWORD, pathToPassword, SONAR\_JDBC\_PASSWORD, SECURITY\_CLIENT\_SECRET, DATABASE\_PASSWORD,EMAIL\_HOST\_PASSWORD, EMAIL\_HOST\_PASSWORD, RUNWAY\_features\_EnableAuthByPass, DB\_PASSWORD,REDIS\_PWD, nifi.security.keyPasswd, nifi.security.keystorePasswd, nifi.security.truststorePasswd, /52yg39sSofLjom395OMM4elSPw, use\_kiali\_token,login\_token, Os04aPGwzNIM9LiYMOhPtR+Zrpw, login\_token, oUsZslgtYo17+tGgxezJyEvo8pw, cGZWBkAZPNxEFBq/VE5z9nd0Rpw, FmOTMJ/Lbhp3TSsJaFy9Zf0DrPw, DB\_PASSWORD, spring\_security\_oauth2\_client\_provider\_forgerock\_token-uri, MAIL\_PASSWORD, DB\_PASSWORD, bearer\_token, event.tokenKey,event.tokenTimestamp, pathToPassword , couchbase-password.toml, elasticsearch-password.toml,password = '${CB\_ES\_CONNECTOR\_CB\_PASSWORD,password =, '${CB\_ES\_CONNECTOR\_ES\_PASSWORD,password = '${CB\_ES\_CONNECTOR\_TRUST\_PASSWORD,truststore-password.toml, security\_impl\_jwt\_token\_cache\_expire\_min,spring\_redis\_password, security\_impl\_jwt\_token\_cache\_expire\_min, security\_impl\_jwt\_token\_validation\_disable, spring\_redis\_password, security\_impl\_jwt\_token\_validation\_disable, spring\_redis\_password, FORGEROCK\_SSO\_CLIENT\_SECRET, DB\_PASSWORD,REDIS\_PWD, serviceAccountToken, serviceAccountToken, secret, serviceAccountToken, serviceAccountToken, REDIS\_PWD, redis.password, database.password, redis.password, cdn.ftp.password, database.password, redis.password, EMPLOYEE\_TOKEN\_CLIENT\_ID, EMPLOYEE\_TOKEN\_CLIENT\_SECRET, EMPLOYEE\_TOKEN\_GRANT\_TYPE, EMPLOYEE\_TOKEN\_SCOPE, EMPLOYEE\_TOKEN\_URL, EMAIL\_HOST\_PASSWORD, service\_notifications\_pw\_updated\_cron,service\_notifications\_pw\_updated\_period\_minutes, xpack.monitoring.elasticsearch.password*

# 2.0 Remediation plan of 'ConfigMap with secrets'

Storing secrets data in a ConfigMap in Kubernetes is not recommended because ConfigMaps are not designed for storing the secrets information. ConfigMaps stores data in plain text.

Azure Key Vault service is more secure way to store the secrets information.

Secrets like password, token, passphrase, secretkey, clientSecret, ect. should be stored in Azure key vault.

We can use Azure key vault to avoid security violation.

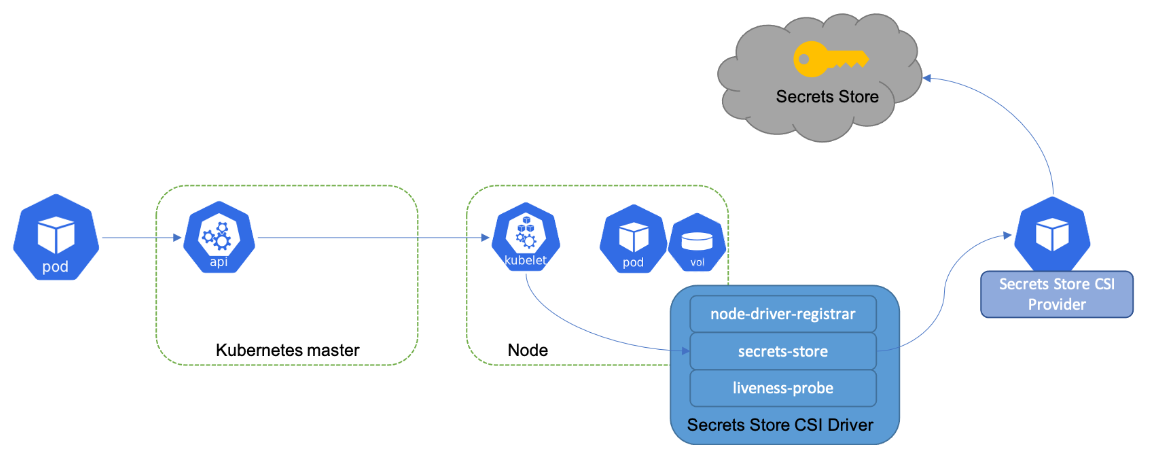
* Secrets, keys, passwords, certificates are stored in the Azure key vault.
* Owner of these keys can manage it, others can access these keys programmatically.
* Others can't see the details of these keys/secrets with nicked eyes.
* As per the security norms, owner can rotate keys/secrets.

To safely use Secrets, following steps need to be followed as a best practice. It can be reviewed through the reference document shown below

* [Enable Encryption at Rest](https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/) for Secrets.
* [Enable or configure RBAC rules](https://kubernetes.io/docs/reference/access-authn-authz/authorization/) with least-privilege access to Secrets.
* Restrict Secret access to specific containers.
* [Consider using external Secret store providers](https://secrets-store-csi-driver.sigs.k8s.io/concepts.html#provider-for-the-secrets-store-csi-driver).

Reference Link: <https://kubernetes.io/docs/concepts/configuration/secret/>

The diagram below illustrates how Secrets Store CSI volume works:



Similar to Kubernetes secrets, on pod start and restart, the Secrets Store CSI (Cloud Storage Interface) driver communicates with the provider using gRPC to retrieve the secret content from the external Secrets Store specified in the SecretProviderClass custom resource. Then the volume is mounted in the pod as tmpfs and the secret contents are written to the volume.

# 3. Azure Key Vault Provider for Secrets Store CSI Driver:

Azure Key Vault provider for [Secrets Store CSI Driver](https://github.com/kubernetes-sigs/secrets-store-csi-driver) allows you to get secret contents stored in an [Azure Key Vault](https://docs.microsoft.com/en-us/azure/key-vault/general/overview) instance and use the Secrets Store CSI driver interface to mount them into Kubernetes pods.

| **Azure Key Vault Provider** | **Compatible Kubernetes** | **secrets-store.csi.x-k8s.io Versions** |
| --- | --- | --- |
| [v1.5.2](https://github.com/Azure/secrets-store-csi-driver-provider-azure/releases/tag/v1.5.2) | 1.21+ | v1, v1alpha1 [DEPRECATED] |
| [v1.4.1](https://github.com/Azure/secrets-store-csi-driver-provider-azure/releases/tag/v1.4.1) | 1.21+ | v1, v1alpha1 [DEPRECATED] |

Azure Key Vault provider for Secrets Store CSI Driver is available as a managed add-on in:

* Azure Kubernetes Service (AKS). For more information, see [Use the Azure Key Vault Provider for Secrets Store CSI Driver in an AKS cluster](https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-driver).
* Azure Arc enabled Kubernetes. For more information, see [Use the Azure Key Vault Secrets Provider extension to fetch secrets into Azure Arc-enabled Kubernetes clusters](https://learn.microsoft.com/en-us/azure/azure-arc/kubernetes/tutorial-akv-secrets-provider).

Use the Azure Key Vault provider for Secrets Store CSI Driver in an Azure Kubernetes Service (AKS) cluster

<https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-driver>

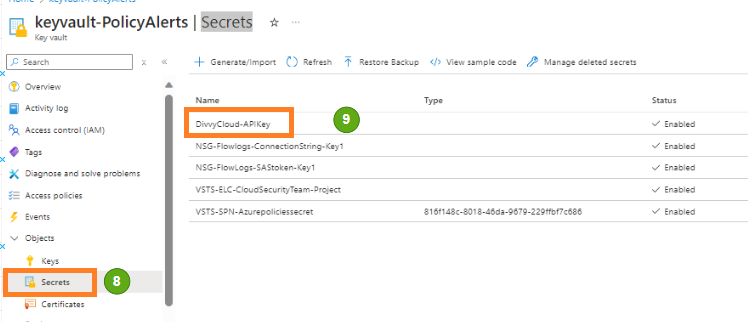
Use the Azure Key Vault Secrets Provider extension to fetch secrets into Azure Arc-enabled Kubernetes clusters

<https://learn.microsoft.com/en-us/azure/azure-arc/kubernetes/tutorial-akv-secrets-provider>

Steps to create secrete in Azure Key Vault:

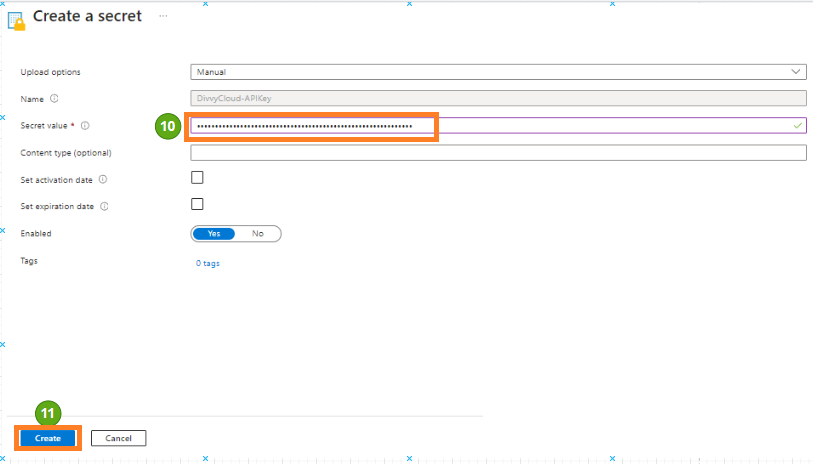
Step 8 – Login into Azure Portal navigate to respective Key Vault Name (keyvault-PolicyAlerts) and select the Secrets.

Step 9 – select the correct secret name from the list.



Step 10 – Click On New Version and enter the Secret Value.

Step 11 – and Create



# 4. Reference document links

<https://kubernetes.io/docs/concepts/configuration/secret/>

<https://secrets-store-csi-driver.sigs.k8s.io/concepts.html#provider-for-the-secrets-store-csi-driver>

<https://azure.github.io/secrets-store-csi-driver-provider-azure/docs/>

<https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-driver>

<https://learn.microsoft.com/en-us/azure/azure-arc/kubernetes/tutorial-akv-secrets-provider>

<https://akv2k8s.io/tutorials/sync/6-secret-to-configmap/>

<https://medium.com/kpmg-uk-engineering/integrate-azure-key-vault-with-aks-3f93f51ea635>