Configure Secret Management Support

Last Modified on 12/04/2024 4:48 am EST

V10.0 and later

Prerequisites

- Basic understanding of Cora Orchestration configuration.
- Basic understanding of the external secret stores, AWS Secrets Manager, or Azure Key Vault, whichever you want to use.
- Knowledge of creating secret keys in the secret stores.

Overview

Cora Orchestration supports external secret stores to store and fetch values like user credentials, connection strings, database credentials, API keys, OAuth tokens, and other secrets for the configuration files at runtime. With this functionality you need not hardcode the sensitive information in plain text or encrypted text. With secret stores you can also control the permission-based access to the sensitive information.

Following are the three secret stores supported:

- AWS Secrets Manager
- Azure Key Vault
- System Environment Variables

Starting from V10.1, to choose a specific secret store for Cora Orchestration, and connect to the secret store, you need to run the Add-CoraOrchestrationSecretEnvironmentVariables PowerShell function.

For V10.0, configuring the secret store is done manually.
To choose the specific secret store, you need to add the `sequence:secrets:providerTypes` environment variable in your system environment variables, and to connect to the secret store, you need to configure the store specific environment variables in the system.

All the store specific environment variables are listed in the sections below.
For the list of secret keys required for Cora Orchestration, see the Secret keys section below.

> **NOTE**
> While upgrading from versions previous to V10.0, make sure to include the following secrets to the configs:
>
> For System Environment Variables store:
>
> - **Genpact.CoraSeQuence.Rijndael.Key**
> - **Genpact.CoraSeQuence.Rijndael.Salt**
>
> For Azure Key Vault store:
>
> - **Genpact0CoraSeQuence0Rijndael0Key** (with same value as Genpact.CoraSeQuence.Rijndael.Key)

AWS Secrets Manager environment variables

Environment variable	Description
sequence:secrets:providerTypes	The secret store type Value: **AWSSecretManager**
sequence:secrets:awsAccessKey	The access key
sequence:secrets:awsSecretKey	The secret key
sequence:secrets:awsRegion	The region for which secret store is being set
sequence:secrets:awsUseSecretNameAsKeyPrefix (Optional)	When True, will generate keys with secret name as prefix: "secretName:secretKey". When False, will generate keys without secret name as prefix: "secretKey".
sequence:secrets:awsKeyPrefixFilter (Optional)	The prefix that all keys must include.
sequence:secrets:awsAcceptedSecretArns (Optional)	The list of identifiers for the secrets that are to be retrieved. The secret ARN (full or partial) and secret name are supported. For example: MySecretFullARN-abcxyz;MySecretPartialARN;MySecretUniqueName
sequence:secrets:awsPollingInterval (Optional)	The waiting time before refreshing the secrets. If null, secrets will not be refreshed. For example, 00:15:00 for 15 minutes.
sequence:secrets:awsSecretNamesFilter (Optional)	The list of secret names that get passed to the client to filter the listed secrets before returning them. For example, secret1;secret2

Azure Key Vault environment variables

Environment variable	Description
sequence:secrets:providerTypes	The secret store type Value: **AzureKeyVault**
sequence:secrets:azureKeyVaultUri	The Azure Uniform Resource Identifier of the key vault.
sequence:secrets:azureKeyVaultTenantId	The ID of the tenant (directory) where the AD application is registered.
sequence:secrets:azureKeyVaultClientId	The ID of the application (client) that you created to read the secrets.
sequence:secrets:azureKeyVaultClientSecret	The secret for the Azure Active Directory application.

Environment variable	Description
sequence:secrets:azureKeyVaultSecretKeyPrefix (Optional)	The prefix for the names of the secrets in the vault.

System environment variables

If you don't want to use an external secret store, you can use your system environment variables to store secrets.

Environment variable	Value/Description
sequence:secrets:providerTypes	The secret store type. Value: **EnvironmentVariables**
sequence:services:executionType *(From V10.4 onwards)*	The execution type to execute ADD, BRS, and JES as **Console Application** under Docker in Kubernetes environments or **Windows Service** under Virtual Machine (VM) environments. Value: • For Kubernetes, predefined in HELM charts: **console** • For VM, created manually: **winsvc**

Secret keys

The following are the secret keys and their values you need to store in your secret store.

Secret key	Description	Value
sequence:persistence:database:provider	Database provider name	Microsoft.Data.SqlClient
sequence:persistence:database:credentials	Database credentials	user id=sa;password=sa;
sequence:persistence:database:connectionString	Database connection string	For example, MultipleActiveResultSets=true;initial catalog=DBName;persist security info=True;data source=DBserverName;packet size=4096;

Secret key	Description	Value
sequence:persistence:database:timeout *(From V10.7 onwards)*	To update the command timeout in the config file.	For example: The result in the config file: • <database ... commandTimeout="1200" /> • <database ... commandTimeout="%%sequence:persistence:database:timeout%%" />
sequence:messageBus:connections:defaultConnectionName	Connection name for the Default type message bus activity set for communication infra for the activity.	Any label that is defined as "name" in the section <messageBus> <connections>. • SqlServiceBroker • ActiveMQ • AzureServiceBus *(From V10.5 onwards)*
sequence:messageBus:connections:notificationsConnectionName *(From V10.5 onwards)*	Connection name for the Notifications type message bus activity set for communication between Cora applications.	
sequence:messageBus:connections:redundancyConnectionName *(From V10.5 onwards)*	Connection name for the Redundancy type message bus activity set for communication between JES applications.	
sequence:messageBus:connections:azureServiceBus:connectionString *(From V10.5 onwards)*	Azure Service Bus connection string, if you have set AzureServiceBus as default message bus connection name.	Endpoint=sb://xxxx.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=xxxxxxxxxxxxxxxxxx
sequence:messageBus:connections:activeMQ:credentials	ActiveMQ credentials, if you have added ActiveMQ as default message bus connection name	user id=mb1;password=sd;
sequence:messageBus:connections:activeMQ:connectionString	ActiveMQ connection string, if you have added ActiveMQ as default message bus connection name	For example, Server=failover:(tcp://192.168.xx.x:00000);Username=usr1;Password=pswd1;

Secret key	Description	Value
sequence:cryptography:sha256:salt	The sha256 salt to prevent identical passwords **NOTE** When you upgrade, this value should not be changed.	Base64string
sequence:cryptography:rijndael:key *(Obsolete from V10.4 onwards)*	The Rijndael key. **NOTE** When you upgrade, this value should not change.	Base64string
sequence:cryptography:rijndael:salt *(Obsolete from V10.4 onwards)*	The Rijndael salt to prevent identical passwords. **NOTE** When you upgrade, this value should not change.	Base64string
sequence:cryptography:aes:key *(From V10.4 onwards)*	The AES key. **NOTE** When you upgrade, this value should not change.	Base64string
sequence:cryptography:aes:salt *(From V10.4 onwards)*	The AES salt to prevent identical passwords. **NOTE** When you upgrade, this value should not change.	Base64string

Since the Rijndael is obsolete from .NET6 onwards, we have upgraded to use AES (Advanced Encryption Standard) algorithm for enhanced security. Cora Orchestration V10.4 onwards uses AES ECB mode for encryptions.
For details, see this article.

For Azure Key Vault, once the secrets are created and finalized, you need to alter the existing legacy credentials and connections strings created before V10.0 to use the new key with the following SQL:

For credentials:

```
update tblCredentials set
fldPassword = REPLACE(fldPassword,'bnBhY3QuQ29yYVNlUXVlbmNlLlJpam5kYWVsLktle','bnBhY3QwQ29yYVNlUXVlbmNlMFJpam5kYWVsMEtle'),
fldCredentialString = REPLACE(fldCredentialString,'bnBhY3QuQ29yYVNlUXVlbmNlLlJpam5kYWVsLktle','bnBhY3QwQ29yYVNlUXVlbmNlMFJpam5kYWVsMEtle')
where fldName='SMTP'
```

For connection strings:

```
update tblConnectionString set
fldConnectionString =REPLACE(fldConnectionString,'bnBhY3QuQ29yYVNlUXVlbmNlLlJpam5kYWVsLktle','bnBhY3QwQ29yYVNlUXVlbmNlMFJpam5kYWVsMEtle')
where fldName='ConnectionString Name'
```

NOTE

You can update the *where* clause based on your needs.