AWS KMS Encryption

The Aspire AWS KMS encryption provider uses keys created in KMS to encrypt and decrypt the data. The secrets will be encrypted with KMS encryption mechanisms, and stored in the Aspire Configuration indexes once encrypted. When Aspire needs the secret value, it calls KMS decrypt mechanism to obtain the value back.

To enable AWS KMS Encryption, you must change your Aspire Settings file on the **encryptionProvider** section to point to the KMS encryption provider jar:

```
"encryptionProvider": {
   "_comments_implementation": [
     "Maven coordinates of the encryption provider",
     "default is: com.accenture.aspire:aspire-encryption-provider"
   ],
   "implementation": "com.accenture.aspire:aspire-aws-kms-encryption-provider"
 },
```

There are two ways of configuring the encryption provider through Properties or Settings File (click each link to see more details)

Regardless of which way it is used to configure the provider, the following parameters will be used:

Parameter	Required	Default	Description
roleARN	no	null	(Optional) If the KMS service must be accessed through the assumption of an IAM role, specify the role ARN. Role Assumption is recommended so the base account won't have direct access to the resources. If not specified, the base account will be used to execute the encryption/decryption calls directly.
keyARN	yes	N/A	The KMS key ARN.
region	yes	N/A	The AWS region on which the KMS service will be used
accessKey	no	null	(Optional) Specify the access key if static credentials must be used for the base account. If this is not specified the Default Credential Provider Chain will be used.
secretKey	no	null	(Optional) Specify the secret key if static credentials must be used for the base account. If this is not specified the Default Credential Provider Chain will be used.

How to create a KMS Key suitable for Aspire?

When creating a KMS key for Aspire, make sure to include the following properties:

- **Symmetric key**: allows aspire to encrypt and decrypt secrets using the key
- Permissions: The user or role to be used by Aspire should be granted the **kms:Encrypt**, **kms:Decrypt** and **kms:DescribeKey** permissions.

Key Policy:

You can add or remove permissions to this policy if needed, but make sure it still have the **Encrypt, Decrypt** and **DescribeKey** ones for the user or role that Aspire will use.

```
{
    "Version": "2012-10-17",
    "Id": "key-consolepolicy-3",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[account_id]:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::[account_id]:[role/user]/[role_id/user_id]"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Create kms key with aws cli

Save the policy specified above into a file called **policy.json**, fill in the *[account_id]* , *[role/user]* and *[role_id/user_id]* details and execute (inside the same folder where the policy file was created):

```
aws kms create-key --policy file://policy.json --description "Aspire Encryption key" > newKey
```

on the file newKey you will see a json with the details of your new key. Copy the Key ARN and configure it as Aspire Properties

Optionally, you can create an alias for your key to help AWS administrators to know what this kms key is for

```
aws kms create-alias --target-key-id [key_id_taken_from_newKey_file] --alias-name alias/aspire5-encryption-key
```