Migrating to a new IBM® Key Protect service instance

IBM® Key Protect service instances provisioned before 15 December 2017 are running on a legacy infrastructure that is based on Cloud Foundry. To enable fine-grained access control with Cloud IAM and other service improvements, we recommend that teams migrate their Key Protect keys into a newly provisioned instance of Key Protect.

Use this document to help you migrate keys to a new Key Protect service instance. Choose from two migration options: Migrating keys using a migration client, or manually moving keys using the Key Protect API.

Contents

Before you begin

> **Important:** Before you begin the migration process, back up your existing encryption keys to a secure location to ensure you maintain access to your data.

To work with Key Protect keys that are stored in a Cloud Foundry space:

- You must have access to the IBM Cloud account where your Key Protect service instance was initially provisioned.
- You must be assigned the appropriate Cloud Foundry access role to view and retrieve Key Protect resources within your IBM Cloud account. For example, if you are assigned a Developer access role, you can retrieve the Key Protect keys that are stored in a Cloud Foundry space. To learn more about viewing your existing Cloud Foundry access policy, see Cloud Foundry access.

To move keys into a new instance of Key Protect:

- You must have a new Key Protect service instance provisioned within your IBM Cloud account. To learn more about creating a new Key Protect service instance, see Provisioning the service.
- New instances of Key Protect use Cloud Identity and Access Management (IAM) for access control. You must be assigned the appropriate Cloud IAM access role to view and create resources within the new Key Protect service instance. If you are assigned a Manager or Writer Cloud IAM role, you can view and create keys in your new Key Protect service instance. To learn more about viewing your existing Cloud IAM access policy, see Working with users.
- New instances of Key Protect support two key types, root keys and standard keys. Legacy service instances support only standard keys. When you import keys into a new instance, you can designate the keys as either root keys or standard keys if you choose to import the keys manually using the Key Protect API.

Migrating keys using the migration client

You can use the migration client to migrate your existing encryption keys into a new Key Protect service instance, so that you may take advantage of the latest IBM Cloud platform functionalities, enhanced security, and expanded availability of our service.

> **Note:** The migration client requires the IBM Cloud CLI and a local environment that can run Bash shell scripts. To learn more about downloading the IBM Cloud CLI for your operating system, see Installing the stand-alone IBM Cloud CLI.

How it works

This utility looks for any Key Protect keys that are stored within the specified Cloud Foundry space and organization in your IBM Cloud account. When you run the client, the utility copies each encryption key into a new Key Protect service instance, where you can continue to manage the lifecycle of the keys and leverage new service capabilities.

Keep in mind the following updates:

- **The identifying information for each key, such as the key metadata and the key ID, will be different after the key is migrated into the new Key Protect service instance.** The client migrates only the key material (the `payload` value) for each encryption key. To run the migrated keys on your existing applications, you must update any references to the old key IDs so that they reflect the new key ID values.
- **You must update your applications to handle base64 encoded key payloads.** This client handles base64 encoding on your behalf as part of the migration process. If you want to store more keys in the new service instance, you must update your applications to handle the base64 encoding requirement.
- **You must use Cloud IAM access tokens to access a new Key Protect service instance.** Cloud Foundry (UAA) tokens are now deprecated for older service instances, and they can no longer be used to access Key Protect. If you are already using Cloud IAM access tokens in your legacy service instance, ensure that your IAM access policies are updated to access the new Key Protect instance.
- **You must update your applications to use a regional endpoint.** Other than the Cloud IAM and base64 encoding changes mentioned above, the Key Protect API remains compatible between legacy and new instances. However, you must access a new Key Protect service instance by using an updated regional endpoint. Generally, this will be a change to the host portion of the URL in configuration or application code. To learn more, see Connecting to the new regional endpoint.

After the migration completes, the client populates your new Key Protect service instance with your migrated encryption keys and creates a `migration.csv` file that shows how the old key IDs map to the migrated keys for easy identification.

Setting up the migration client

Step 1. Download the latest client

1. Download the latest release of the migration client.
2. Extract the release, and then change into a newly created directory to begin working with the migration client.

   ```
   unzip migration-client-<your_OS>.zip -d migration-client
   cd migration-client
   ```

Step 2. Generate authentication credentials

To generate authentication credentials for your legacy Key Protect service instance:

1. Log in to the IBM Cloud console.
2. From your user profile, select the account that contains the Cloud Foundry org and space where your legacy Key Protect service instance resides.
3. From the Resource list, navigate to **Cloud Foundry Services**, and then select the Key Protect service instance that contains the encryption keys that you want to migrate.

   Note the **Org** and **Space** names that are associated with the legacy Key Protect service. You'll need to set these names as environment variables in a later step.

To generate authentication credentials for your new Key Protect service instance:

1. In the IBM Cloud console, select the account and resource group where your new Key Protect service instance resides.
2. From the Resource list, navigate to **Services**, and then select the Key Protect service instance where you want to migrate your existing encryption keys.

   Note the name that is associated with your Key Protect service instance. You'll need to set this name as an environment variable in a later step.

Step 3. Set your environment variables

1. Open the `envs` file that is located in the `migration-client` directory.
2. Set the following environment variables to authenticate to your Key Protect service instances.

   ```
   ## Legacy account variables ##
   export CF_ORG="<organization_name>"
   export CF_SPACE="<space_name>"

   # New Key Protect account variables ##
   export KP_SERVICE_INSTANCE_NAME="<instance_name>"

   # Optional. Set if your new Key Protect service instance is in a different IBM Cloud account.
   # export KP_ACCOUNT_ID="<account_ID>"
   ```

Replace `<organization_name>`, `<space_name>`, and `<instance_name>` with the values that you retrieved in the previous step.

3. Save the `envs` file and continue to the next step.

Migrating your keys

1. Run the `client-wrapper.sh` script to start migrating keys from your legacy Key Protect service instance.

   ```
   ./client-wrapper.sh
   ```

   The client logs into IBM Cloud by using the IBM Cloud CLI plug-in, and then authenticates to each of your Key Protect service instances.

Success! Your existing keys are now migrated into a new Key Protect service instance. You can view how the old key IDs map to the migrated keys by inspecting the `migration.csv` file that is generated after the migration completes. The following table shows an example `migration.csv` file:

Old key ID	New key ID
ef9eb687-b508-45f0-8a3e-1def949bc9f8	e9ab551c-46fe-448a-8a3c-e0f23dfff362

The Key Protect keys that are stored in your Cloud Foundry org and space remain in the legacy Key Protect service instance until you're ready to permanently delete the keys, and then delete the legacy Key Protect service instance.

> **Note:** If migration fails in the middle of moving keys, check the `migration.csv` file to view the keys that were successfully migrated. To resume the migration process, be sure to save the `migration.csv` file, otherwise the client will move the keys again and create duplicate keys in the new instance. If you encounter more errors, check the `migration-client.log` file to understand how to proceed.

Migrating keys using the Key Protect API

You might want to migrate your keys manually if you have only a few keys stored in your legacy Key Protect service instance. If you prefer this approach, you can retrieve and then import your keys by using the Key Protect API.

Step 1. Generate your authentication credentials

To generate authentication credentials for your legacy Key Protect service instance:

1. Log in to IBM Cloud through the IBM Cloud CLI.

   ```
   ibmcloud login
   ```

   If the login fails, run the `ibmcloud login --sso` command to try again. The `--sso` parameter is required when you log in with a federated ID. If this option is used, go to the link listed in the CLI output to generate a one-time passcode.

2. Select the IBM Cloud org and space that contain your Key Protect service instance.

3. Retrieve your IBM Cloud org and space GUIDs.

   ```
   ibmcloud iam org <organization_name> --guid
   ibmcloud iam space <space_name> --guid
   ```

   Replace `<organization_name>` and `<space_name>` with the unique aliases that you assigned to your organization and space.

4. Retrieve your IBM Cloud access token.

   ```
   ibmcloud iam oauth-tokens
   ```

Step 2. Retrieve your existing keys

After you generate your authentication credentials, call the Key Protect API to list and retrieve the keys that are available in your legacy Key Protect service instance.

1. List the keys that are stored in your service instance by running the following command.

   ```
   curl -X GET \
      https://ibm-key-protect.edge.bluemix.net/api/v2/keys \
      -H 'accept: application/vnd.ibm.collection+json' \
      -H 'authorization: <access_token>' \
      -H 'bluemix-org: <organization_GUID>' \
      -H 'bluemix-space: <space_GUID>'
   ```

Replace the variables in the example request according to the following table.

Variable	Description
access_token	Your IBM Cloud access token. Include the full contents of the token, including the Bearer value, in the cURL request.
organization_GUID	The unique identifier that is assigned to your IBM Cloud organization.
space_GUID	The unique identifier that is assigned to your IBM Cloud space.

A successful request returns a collection of keys available in your Key Protect service instance.

2. Copy the returned ID value for each key. The ID is a unique identifier that is assigned to your key and is used for subsequent calls to the Key Protect API.

3. Retrieve a key that is stored in your service instance by running the following command. Repeat the command for each key.

```
curl -X GET \
   https://ibm-key-protect.edge.bluemix.net/api/v2/keys/<key_ID> \
   -H 'accept: application/vnd.ibm.collection+json' \
   -H 'authorization: <access_token>' \
   -H 'bluemix-org: <organization_GUID>' \
   -H 'bluemix-space: <space_GUID>'
```

Replace the variables in the example request according to the following table.

Variable	Description
key_ID	The identifier for the key that you retrieved in step 1.
access_token	Your IBM Cloud access token. Include the full contents of the token, including the Bearer value, in the cURL request.
organization_GUID	The unique identifier that is assigned to your IBM Cloud organization.
space_GUID	The unique identifier that is assigned to your IBM Cloud space.

A successful response returns details about your key and the key material.

4. Copy the returned `payload` value for each key, and save the information to a secure location. The payload represents the key material that is associated with your key.

Step 3. Import the keys to a new service instance

To import the keys into your new service instance:

1. Base64 encode the `payload` values that you retrieved in the previous step. Repeat the command for each key.

```
echo <payload> | base64
```

2. Import the base64 encoded keys into your new Key Protect service instance.

   To learn about adding keys, check out the documentation for Importing standard keys.

Updating your applications

To start using the new Key Protect service instance, update your applications so that they reference the new key IDs and point to the latest Key Protect API endpoint.

Connecting to the new regional endpoint

Key Protect service instances that exist within a Cloud Foundry org or space use the legacy `https://ibm-key-protect.edge.bluemix.net` endpoint to interact with the Key Protect API. To interact with your new service instance, you must update any references to this endpoint to `https://<region>.kms.cloud.ibm.com`.

For example, if you created your new service instance in the US South region, use the following endpoint and API headers to browse keys in your service:

```
curl -X GET \
   https://us-south.kms.cloud.ibm.com/api/v2/keys \
   -H 'accept: application/vnd.ibm.collection+json' \
   -H 'authorization: <IAM_token>' \
   -H 'bluemix-instance: <instance_ID>'
```

For more information, see the Key Protect API reference.

> **Note:** On 2018 November 30, IBM Cloud rolled out updates to the platform, and `bluemix.net` was replaced with `cloud.ibm.com`. As part of this change, Key Protect updated its API endpoints from `https://keyprotect.<region>.bluemix.net` to `https://<region>.kms.cloud.ibm.com`. Currently the service supports both URLs.

Handling the base64 encoding requirement

Because new Key Protect service instances allow only base64 encoded key material (the `payload` value in the JSON body) for keys, you must base64 decode keys on retrieval to get the same payload data that you expected previously.

There are many libraries in the various languages that are available for this task. If you want to check your keys by hand (or if you use shell), you can use the base64 utility to decode the retrieved payload.

For example, if you want to decode the base64 encoded payload after you retrieve it from Key Protect, run the following shell command:

```
echo <base64_encoded_payload> | base64 -D
```

If you plan to use your new Key Protect service instance to import encryption keys in the future, ensure that you provide key material that is base64 encoded before you upload it to the service.

```
echo <payload> | base64
```

Testing your migration

To ensure that your apps continue to work with the new changes, perform a regression test on your associated applications to complete the migration process.

After your migration and testing is complete, please notify the Key Protect team by sending an email to the Key Protect offering manager at mosbaugh@us.ibm.com.

Getting help

If you encounter a problem during a migration or in the regression tests of your applications, you can reach out to the IBM Key Protect team for help. Connect with the Key Protect development team by sending an e-mail to Terry Mosbaugh at mosbaugh@us.ibm.com.

To find out more about the latest Key Protect service features, check out the Key Protect service documentation.