



SAP Integration Suite

Generated on: 2024-08-08 05:12:07 GMT+0000

SAP Integration Suite | Cloud

PUBLIC

Original content: https://help.sap.com/docs/SAP_INTEGRATION_SUITE/51ab953548be4459bfe8539ecaeee98d?locale=en-US&state=PRODUCTION&version=CLOUD

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

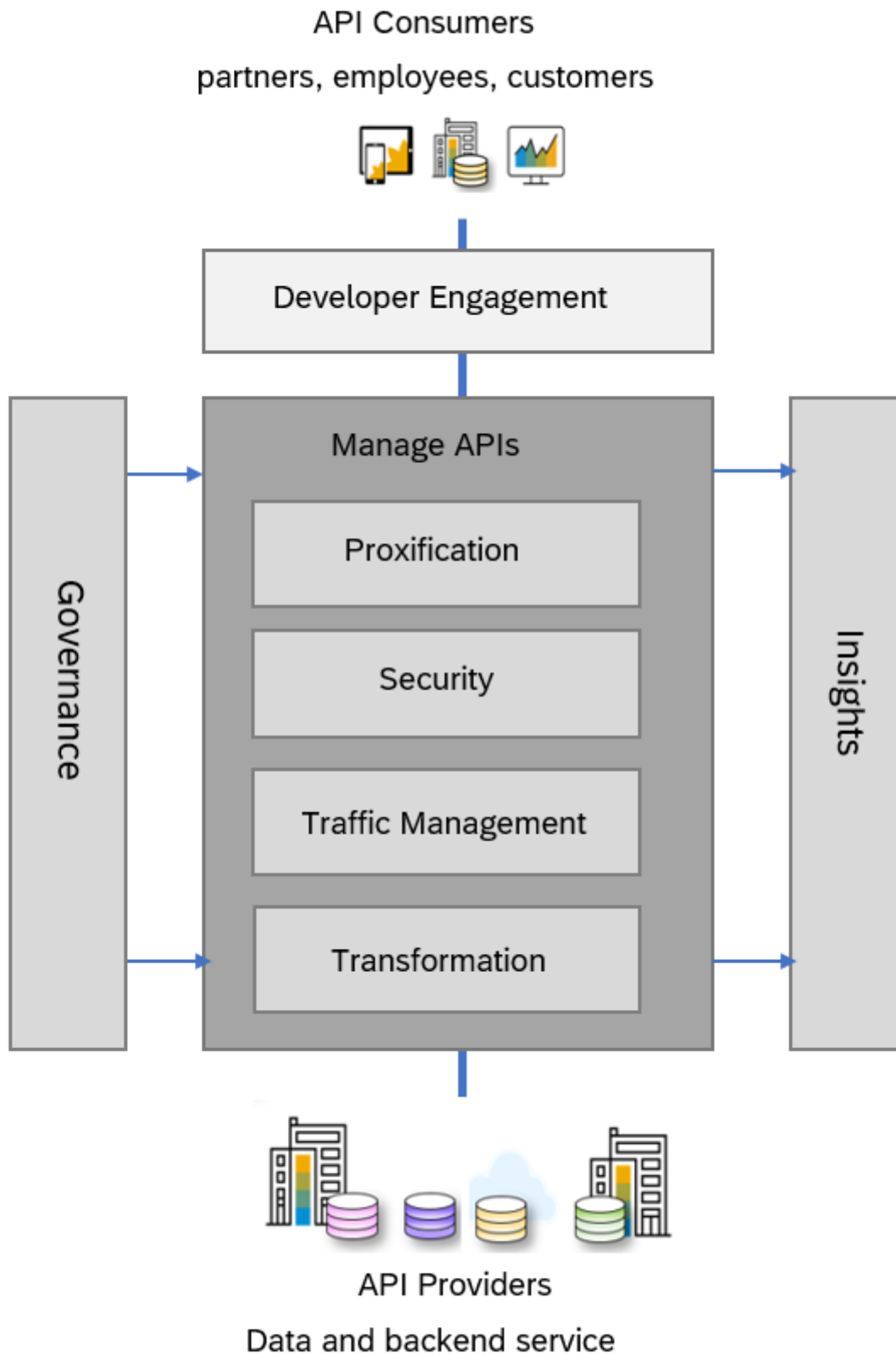
For more information, please visit the <https://help.sap.com/docs/disclaimer>.

API Management

The API Management capability in SAP Integration Suite is a complete solution, that addresses all enterprise requirements for API security and governance. It lets you publish, promote, and oversee APIs in a secure and scalable environment.

With API Management you can perform the following tasks:

- **Proxify your APIs:** Create your own unified and harmonised API presence, using your own domain.
- **Secure your APIs:** Secure your APIs against unauthorized access and threats. API management helps organizations define a standardized set of policies to protect APIs and the underlying backends.
- **Perform Traffic Management:** Configure cache, and control traffic quotas and spikes, using the traffic management policies.
- **Govern your APIs:** Discover and document all your APIs, manage the lifecycle of your APIs and govern them using the policies. Over 30 different policy types are available, ranging from traffic management and security policies.
- **Get Business Insights:** Monitor with usage analytics, logs, events and triggers; use business insights to monetize your APIs.
- **Transform your APIs:** Apply advance header and payload modifications.
- **Developer Engagements:** API business hub enterprise is a feature-rich, themed, and customizable portal designed specifically for application developers. It provides comprehensive API documentation, code snippets, and more. With API business hub enterprise, developers can easily engage with the platform, enabling them to discover, subscribe to, and consume APIs directly.



Features

Create omni-channel experiences

Use API Designer and Open APIs to create a omni-channel mobile experience across devices.

Secure your digital assets, interfaces

Help protect your data and digital assets in this hyper-connected world. Get deep insights on API usage.

Manage the end-to-end lifecycle of APIs

Scale billions of API calls to unlock new opportunities, new business potential and add additional value.

Engage developers and partners

API Business Hub Enterprise simplifies sharing managed APIs and collaborations with customers, partners, and developers.

Grow new revenue streams

Monetize your data and digital assets with help of API Portal. Upsell and cross-sell through your ecosystem.

Evolve B2B integrations

Extend solutions with additional SAP BTP capabilities for mobile, offline and integration.

Benefit from multitenancy support

Use this service in tenant-aware pplications.

Getting Started

You can provision the API Management capability from the Integration Suite launchpad. For the detailed steps, see [Activate and Configure API Management Capability](#).

Related Information

[API Lifecycle](#)

Working with API Management

Get an understanding of API Management within SAP Integration Suite and leverage its capabilities effectively.

The table below enumerates the concepts, capabilities, functions and services of API Management:

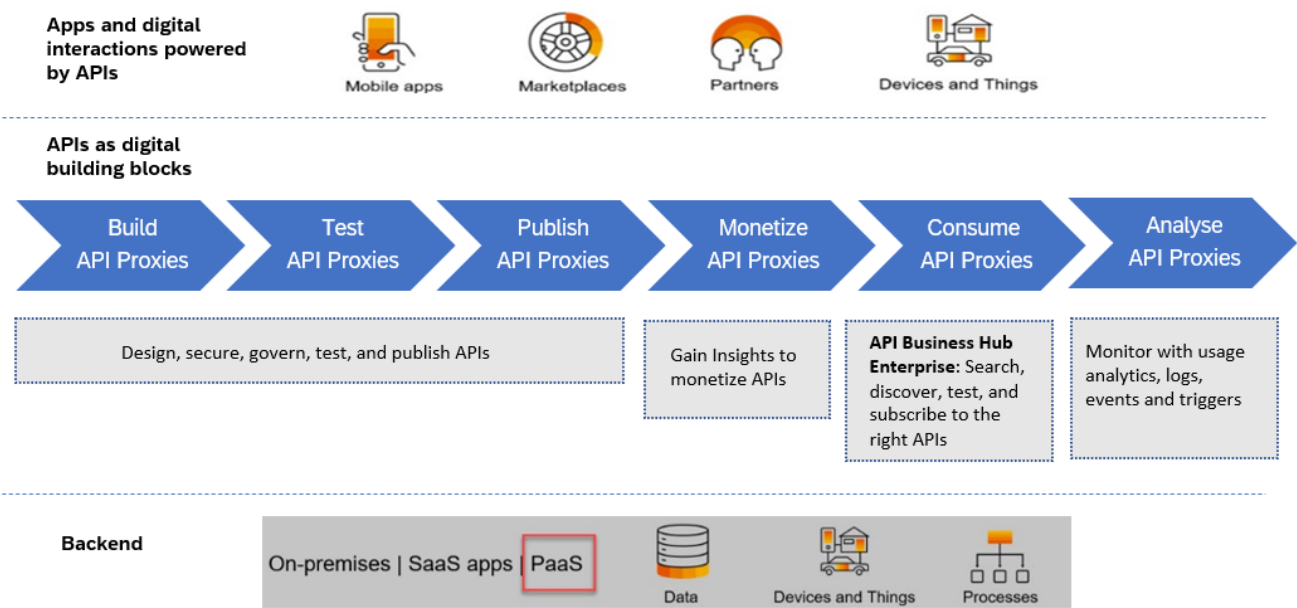
API Lifecycle	The API lifecycle encompasses several stages, including API planning, creation, and implementation by company developers, API creators, and composers. It extends to the consumption of these APIs by other employees, partners, or users of the company's products and services. For more information, see API Lifecycle .
Enable API Management Capability	Set up the API Management capability from Integration Suite. For more information, see Activate and Configure API Management Capability .
Components of API Management	The API Management components collaborate to offer a comprehensive solution, allowing organizations to securely expose, manage, and monetize their APIs. This empowers developers to effectively build applications that utilize these APIs. It comprises of five key components: API Management Runtime, API Management Design Time, API business hub enterprise, API Analytics and API Designer. For more information, see Components of API Management .
Important Concepts of API Management	The structure of the API Management capability within Integration Suite revolves around APIs, products, applications, users, developers, and accounts. Understanding this structure is crucial for effectively managing and organizing APIs within API Management. For more information, see Important Concepts of API Management .

Managing API Endpoints	<p>The primary purpose of API Management is to manage your API endpoints. Managing API endpoints involves two categories:</p> <ul style="list-style-type: none"> • Proxified API endpoints, which are the APIs that you explicitly want to "proxify" by routing API calls through an API proxy. • Unproxified API endpoints, also known as unmanaged or externally managed APIs, are endpoints that do not require an API proxy and are not part of the strictly managed APIs. <p>For more information, see Managing API Endpoints.</p>
Configure APIs	Create, update API proxies and register APIs. For more information, see Configure APIs .
API Services	A variety of APIs are offered as services in specific use cases and workflows. You can explore them and try it out in the SAP Business Accelerator Hub. For more information, see API Services .
API Management Service Plans	The functionality of the API Management capability is typically managed using the Integration Suite (UI) application. However, there are certain service plans that allows you to call API Management APIs, manage Cloud Foundry applications, and connect to an on-premise backend system. For more information, see API Management Service Plans .
Additional Configurations for API Management	In this section, you can find information about configuring additional virtual host, region-specific IP addresses for API Management, managing user roles in API Management, cancelling your API Management service subscription, and integrating API Management with SAP Cloud Identity Services. For more information, see Additional Configurations for API Management .
API Documentation	Gather additional instructions on how to effectively use and integrate with an API. For more information, see API Documentation .
Migration of API Management Content	You can now migrate the API Management content from Neo to Cloud Foundry or between different Cloud Foundry environments. For more information, see Migration of API Management Content .
Limits in API Management	There are certain boundary conditions to consider when building, managing, and reviewing APIs. API Management is designed to perform at its best when configured within these specified conditions. For more information, see Limits in API Management .

API Lifecycle

The API lifecycle, starts from API planning, creation and implementation by company developers and other API creators and composers, to the consumption of these APIs by other employees, partners or users of the company's products and services.

API Management in SAP Integration Suite is used to discover, shape, compose, integrate, manage, and secure APIs in the entire landscape. APIs are ultimately published in the form of a catalog (developer portal), and exposed for consumption by developers, who develop multi-experience applications



Related Information

- [Build API Proxies](#)
- [Test API Proxies](#)
- [Publish API Proxies](#)
- [Monetize APIs](#)
- [Consume API Proxies](#)
- [Analyze API Proxies](#)


Activate and Configure API Management Capability

Steps to activate and configure API Management capability in SAP Integration Suite.

Steps	Details
Subscribe to SAP Integration Suite	<p>To set up the API Management capability from Integration Suite, you should first have an Integration Suite subscription.</p> <p>Subscribe to the SAP Integration Suite in SAP BTP cockpit and assign the Integration_Provisioner role to gain access. For more information, see Initial Setup of SAP Integration Suite.</p> <p>i Note</p> <p>Please make sure that you do not have a starter plan instance created in the same subaccount where you intend to create an Integration Suite subscription. Additionally, please note that API Management capabilities from Integration Suite and API Management subscriptions using the stand-alone tile cannot coexist in the same subaccount.</p>
Activate the API Management capability	<p>Add and activate API Management capability in Integration Suite. For more information, see Activating and Managing Capabilities.</p>

Steps	Details
Configure the API Management capability	<p>Access the API Settings page, and complete the onboarding process. For more information, see Setting Up API Management Capability.</p> <p>To access the API Settings, the APIManagement.Selfservice.Administrator role must me assigned to you.</p> <p>i Note</p> <p>If you are using a trial account, your account will be automatically provisioned once you select the APIs option in the Settings menu. During the provisioning process, the system will assign a default virtual host for you. You will receive a notification when the provisioning is complete. At this point, the APIs option will no longer be visible in the side-navigation menu under Settings. You will be logged out of the Integration Suite automatically. To continue, simply log in again. After logging in, you will have the ability to create APIs, build API proxies as a service provider, and utilize other convenient services offered by the platform.</p>

i Note

For visual intructions on how to activate and configure API Management capability from Integration Suite, refer the tutorial [Set Up API Management from Integration Suite | Tutorials for SAP Developers](#).

Related Information

- [Assign User Roles in API Management](#)
- [Create an API Provider](#)
- [Different Methods of Creating an API Proxy](#)
- [Configuring a Custom Domain for a Virtual Host](#)
- [Configuring Mutual TLs for Virtual Host](#)
- [Cancel API Management Service Subscription](#)

Components of API Management

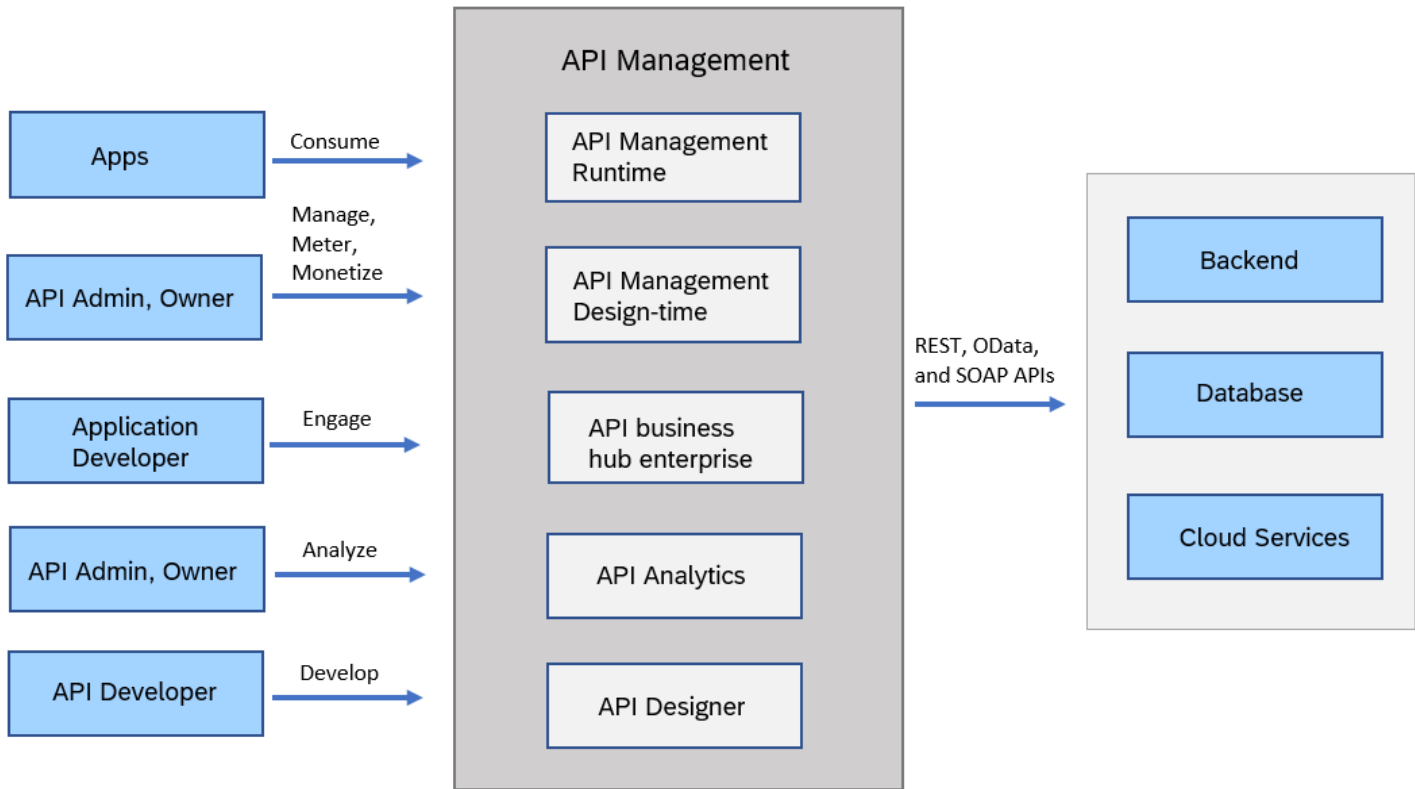
The API Management components work together to provide a comprehensive solution, enabling organizations to securely expose, manage, and monetize their APIs while empowering developers to build applications that leverage these APIs effectively.

The API Management infrastructure comprises of five key components:

- **API Management Runtime:** You can deploy and effectively utilize your APIs. Applications consume the API runtime, request API authentication, and gain access.
- **API Management Design Time:** Serves as a platform for easy API discovery, allowing API administrators to manage, meter, and secure their APIs. Additionally, it enables administrators to define and publish rate plans for their APIs. For more information, see[Build API Proxies](#), [Publish API Proxies](#) [Monetize APIs](#) [Analyze API Proxies](#).
- **API business hub enterprise:** It is a self-service portal for application developers. You can discover, browse, and explore APIs. Subscribe to a rate plan and build application. For more information, see [Consume API Proxies](#).

- **API Analytics:** Offers advanced analytical capabilities to track your API usage. Utilize API Analytics to gather data on URL usage, user IDs associated with API calls, latency information, and more. For more information, see [Analyze API Proxies](#).
- **API Designer:** API developers have the ability to define, implement, and document APIs. The API Designer offers support for open APIs and allows for the generation of various outputs. For more information, see [Create an API Proxy Using the API Designer](#).

The diagram below illustrates the interaction between various stakeholders and the different components of API Management. It also demonstrates how API Management integrates with both cloud and on-premise systems.

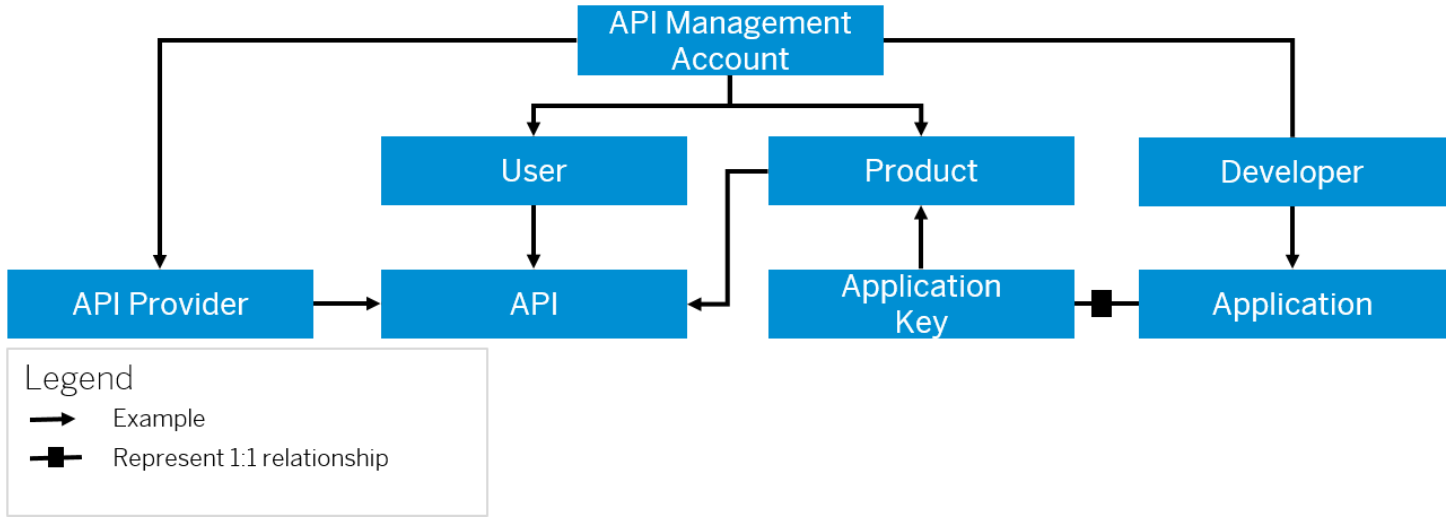


The API Management capability within Integration Suite allows you to utilize SAP's internal API providers as well as external APIs, regardless of whether they are SAP-related or not. By leveraging your existing investments in SAP solutions, the API Management capability facilitates integration with non-SAP solutions. This platform plays a crucial role in unlocking the value of your digital assets and empowers you to create and deliver content effectively. With this capability, applications can seamlessly access secure backend data, ensuring smooth operations. Additionally, it offers a unified experience for managing and monitoring APIs across diverse data platforms, accompanied by real-time analytics.

Important Concepts of API Management

The structure of the API Management capability within Integration Suite revolves around APIs, products, applications, users, developers, and accounts. APIs are grouped into products, which are accessed by applications created by developers.

Users interact with APIs through applications, and accounts are used to manage access and permissions. Understanding this structure is crucial for effectively managing and organizing APIs within API Management.



To effectively utilize the API portal within the Integration suite, it is important to grasp the following concepts:

Entity	Definition
API Management Account	An API Management account is the highest level of data hierarchy. An account is a representation of all components including APIs, products, applications, systems, users, and developers.
System	In API Management, System refers to the API provider systems where the actual backend services reside. System could either be an ABAP system, SAP Gateway system, Enterprise Services Repository, or systems that host generic REST services or third party provider systems. API Management allows you to add and manage an API provider system. After you have added a system, you can browse for the APIs in that system.
User	API Management can have multiple users. Different users have different roles and privileges assigned. For example, people who create APIs and products or analyze the metrics or the application consumer who can access the APIs provisioned by API Management.
API	APIs are Application Programming Interfaces. They comprise a set of routines, protocols, and tools for building software applications. APIs define sets of requirements that govern how applications communicate with one another. They facilitate interaction by selectively exposing certain functionalities, allowing different applications, websites, or devices to communicate effectively with each other. i Note API Management supports OData, REST, and SOAP services.
Product	A product is a bundle of APIs. It contains metadata specific to your business for monitoring or analytics. For example, all APIs related to CRM can be bundled as one CRM product. API Management collects data for analyzing the products.
Developer	One or more developers can create applications in the API Management account. A developer can consume the APIs, but cannot create APIs. To create an application, the developer must have registered the account. After having created an application, the developer uses the app (application) key to consume the APIs.

Entity	Definition
Application	Applications include the Web or mobile applications that consume the exposed APIs. When you create an application, you select the product to include in this application. For each application that you create, API Management generates an app key and secret. Use this key to gain access to multiple products. Developers create one or more applications using the APIs you expose.
App Key	Based on the authorization mechanism you define for your APIs, the application passes an app (application) key together with every request to your APIs. If that key is valid, the request is permitted. API Management supports different types of authentication, such as a simple API key, OAuth, and so on.

Managing API Endpoints

The primary purpose of API Management is to manage your API endpoints.

There are two categories of API endpoints:

- Those that you explicitly want to “proxify” by routing API calls through an API proxy (referred to as "proxified" endpoints)
- Those that should be registered in the list of governed APIs but not proxified (referred to as "unmanaged APIs" or "externally managed APIs")

Both proxified and unproxified API endpoints can be published to developers.

API Proxies

You proxy your APIs to add an extra layer of security by filtering and validating requests before they reach the API. They can also implement authentication and authorization mechanisms to control access to the API. For more information, see [API Proxy](#).

Policies

When it comes to API proxies, policies are a set of rules and configurations that are applied to API proxies to control their behavior. It is used to enforce security measures, perform data transformations, apply rate limiting, logging, caching, and more. They provide a way to customize and control the behavior of the API proxy. For more information, see [Policies](#).

API Provider

An API provider describes a specific host to whose API services you want to provide access. You specify the host details and any further details that are necessary to establish the connection, for example, proxy settings. This is particularly useful when create API proxies to on-premise (backend) OData based systems. See [Create an API Provider](#). For more information, see [API Providers](#).

Externally-managed APIs

Unmanaged or externally managed APIs, are endpoints that do not require an API proxy and are not part of the strictly managed APIs. These endpoints may already be managed by another API Gateway or API Management solution, such as one offered by a hyperscaler. Registering such an API into the catalog of managed APIs can be advantageous as it allows developers to easily access and utilize them. However, it's important to note that unproxified API endpoints are not managed or monitored, and they do not have rate plans assigned to them. For more information, see [Externally Managed APIs](#).

API Artifacts on Edge Integration Cell

The Edge Integration Cell introduces a new type of proxified integration flow, referred to as an API artifact. The API artifacts allows users to easily add and configure policies and integration flow steps in a unified manner using the Integration Flow Editor. This type of integration flow conveniently embeds the policies of an API proxy, which can be specified during the design of the integration flow. For more information, see [API Development](#).

Configure APIs

The **Configure APIs** menu allows you to perform two tasks: creating and updating API proxies, and registering APIs that are not currently proxified. The text is too short to provide a summary.

Registering an Externally-Managed API

Unproxified API endpoints, also known as unmanaged or externally managed APIs, are endpoints that do not require an API proxy and are not part of the strictly managed APIs. Registering an unproxified API into the catalog of managed APIs can have several benefits, as it allows for further exposure to developers. To register an unproxified API, an API specification is required. API specifications are based on the OpenAPI (also known as Swagger) standard and can be imported as a file or designed using the API Designer tool. For more information, see [Adding Externally Managed APIs](#).

Creating an API Proxy

API proxies are the fundamental components of the API Management feature in SAP Integration Suite. An API proxy acts as a protective layer in front of your service API (the target endpoint), offering consumers of your APIs a standardized and properly formatted URL. Additionally, it provides security measures and mediation policies to further safeguard the API. For more information, see [Build API Proxies](#).

There are two ways to create an API proxy for an existing system or service API:

- You can directly provide the complete service URL when creating the API proxy. For more information, see [Create an API Proxy by Providing a Direct Target Endpoint URL](#).
- Alternatively, you can first set up a system descriptor (host, port) as an API provider and then provide the service path when creating the API proxy. For more information, see [Create an API Proxy by Referring to an API Provider System](#)

You can create an API proxy based on an existing API proxy. By creating a new API proxy based on an existing one, you can add or modify features to meet your specific requirements. For more information, see [Create an API Proxy Based on an Existing API Proxy](#).

If you want to create API proxies for on-premise systems like SAP S/4HANA or ECC, you need to define an API provider first and use it when creating API proxies for its services.

For all other URLs, whether they are cloud-based or "Internet" URLs, API administrators have the option to directly provide the URL during the API creation process or define an API provider.

Creating an API Proxy by Providing a Cloud-based URL

Creating an API proxy using a cloud-based URL can enhance the security, scalability, performance, and monitoring capabilities of your API, providing a better experience for both developers and end-users. For more information, see [Create an API Proxy by Providing a Direct Target Endpoint URL](#).

Creating an API Proxy to an on-premise SAP (OData) System

The creation of an API Proxy for an on-premise system service involves three key steps:

1. Setting up a cloud connector to establish a connection between your account and the on-premise system.
2. Creating a service instance of the API Management connectivity plan to enable the necessary connectivity.
3. Creating an API provider of the on-premise type to facilitate the integration.

There are two different approaches to this process:

- Creating an API proxy that authenticates users with principal propagation.
- Creating an API proxy with transparent authentication.

These steps and variations allow for the effective implementation of an API proxy for an on-premise system service.

Creating an API Proxy with principal propagation to an on-premise SAP (OData) system

The ability to create an API proxy for an on-premise SAP endpoint and to propagate the identity of the caller (principal propagation) to this on-premise system is an important feature of API Management in Integration Suite. To address this challenge, the following steps need to be taken:

1. Set up a cloud connector from your account to the on-premise system, using either "X.509 certificate" or "certificate" as the principal type.
2. Establish a service instance of the API Management connectivity plan. For detailed instructions, see [Accessing On-Premise Systems through API Management](#).
3. Create an API provider of type **on-premise** and set the authentication to **Principal Propagation**. For more information, see [Create an API Provider](#).
4. When creating the API proxy, make sure to use the created API provider and specify the path to the API service.

Creating a transparent API proxy to an on-premise SAP (OData) system




Create an API proxy for an on-premise SAP endpoint without applying any authentication in the API proxy. This approach can be useful, for example, if the backend service expects the client to provide a user and password in the form of HTTP headers, which will be passed through the API proxy as-is.

To create an API proxy without authentication:

1. Set up a cloud connector from your account to the on-premise system, using either "X.509 certificate" or "certificate" as the principal type.
2. Establish a service instance of the API Management connectivity plan. For detailed instructions, see [Accessing On-Premise Systems through API Management](#).
3. Create an API provider of type **on-premise** and set the authentication to **None**. For more information, see [Create an API Provider](#).
4. When creating the API proxy, make sure to use the created API provider and specify the path to the API service.

API Services

From API Management, a variety of APIs are offered as services in specific use cases and workflows. You can explore them and try it out in the SAP Business Accelerator Hub in the following url: <http://api.sap.com>.

Services	Description
Integration Suite	You can browse through this API package for API admin services with the required resources.
API business hub enterprise	You can browse through this API package for application developer services that are offered.
Metering	You can now browse through this API package to view metering data for APIs, API Products, and applications in API Portal.
Client SDK	<p>A client software development kit (SDK) is available for developers through a non-commercial license on open source sites.</p> <p>In the API Portal, at the top-right corner, choose Navigation Links () and select Client SDK (). On selecting the client SDK, you are navigated to the maven repository, where you can download this package.</p> <p>For more information, see SAP API Management, 1.6.0 Client SDK.</p> <p>You can also download the package from https://int.repositories.cloud.sap/artifactory/deploy-releases/com/sap/apimgmt/client/sdk/apim-client-sdk/1.6.0/apim-client-sdk-1.6.0.zip . On navigating to this link, select the latest version and choose View All.</p>

API Management Service Plans

The functionality of the API Management capability is typically managed using the Integration Suite (UI) application, which is the primary focus of this user guide. However, you can also call API Management APIs, manage Cloud Foundry applications, and connect to an on-premise backend system by means of service plans.

The API Management services on Cloud Foundry provides different capabilities through the following plans:

[Accessing API Management APIs Programmatically](#)

The **apiportal-apiaccess** plan offers external applications the ability to access the public APIs of the Integration Suite API Management capability. These APIs are used by the external applications to perform CRUD operations on API Management features like API proxies or products. These APIs are built on REST and OData principles and are extensively documented on the SAP Business Accelerator Hub.

[Managing Cloud Foundry Microservices through API Management](#)

The **apim-as-route-service** plan helps you in managing Cloud Foundry applications by including policies such as rate limit, quota. The service instance you create through this plan allows you to bind to the route service and creates an API Proxy. This API Proxy serves in establishing a secure connection with your Cloud Foundry application and all the calls made to the Cloud Foundry application are routed via **API Management, API portal**.

[Accessing On-Premise Systems through API Management](#)

The **on-premise-connectivity** plan helps in achieving principal propagation while connecting to an on-premise backend system.

Accessing API Management APIs Programmatically

The **apiportal-apiaccess** plan offers external applications the ability to access the public APIs of the Integration Suite API Management capability. These APIs are used by the external applications to perform CRUD operations on API Management

features like API proxies or products. These APIs are built on REST and OData principles and are extensively documented on the SAP Business Accelerator Hub.

About the Plan

The **apiportal-apiaccess** plan allows you to programmatically import/export API proxies, create products, key value maps. It is especially useful when integrating API Management with a CI/CD process or when migrating from a Neo to Cloud Foundry environment using the migration tool.

The API Access plan allows you to generate a service key by creating a service instance. By creating a service instance, you can generate a service key that includes the application url, clientId, clientSecret, and tokenUrl is used to generate a bearer token with the help of a REST Console. This Bearer Token, along with the application url and API endpoint are used to trigger the API. Therefore, bearer token acts like a key to access the APIs.

Prerequisites

- You've enabled API Management capability using Integration suite. For more information, refer [Subscribing to Integration Suite](#) and [Activating Capabilities](#).

OR

You have subscribed to the standalone **API Management, API portal** tile in the Cloud Foundry environment. For more information, see [Set Up API Portal Application](#).

- As a subaccount administrator, you additionally need the role of (org member) and **space developer** in the Cloud Foundry space in which the Integration Suite is provisioned.

To enable API access for API Management, API portal execute the steps in the sections below:

Creating a Service Instance in the API Management, API portal

Create a service instance using API Access plan to generate a service key.

- In your web browser, open the **SAP BTP Cockpit** - <https://cockpit.btp.cloud.sap>.
- From your **Subaccount**, navigate to **Spaces** in your Cloud Foundry environment and choose **Services > Service Marketplace**.
- Choose **API Management, API portal > Instances > New Instance**.

i Note

If you are unable to view the **API Management, API Portal** tile, please check your entitlements. For more information, see [Managing Entitlements and Quotas Using the Cockpit](#).

- In the **Create Instance** dialog that opens, choose the **apiportal-apiaccess** plan.
- In the section **Specify parameters**, paste one of the following JSON codes, to assign a specific role.

The following roles are supported for the current scenario:

Assign *APIPortal.Administrator* role to access the API portal APIs and perform operations like create, update, delete on various API portal entities as specified in the SAP Business Accelerator Hub.

```
{
  "role": "APIPortal.Administrator"
}
```

Assign *APIPortal.Guest* role to access the API portal APIs in read-only mode. You can view the API portal entities as specified in the API Business Hub.

```
{
  "role": "APIPortal.Guest"
}
```

Assign *APIManagement.SelfService.Administrator* role to configure additional virtual hosts.

```
{
  "role": "APIManagement.SelfService.Administrator"
}
```

- 6. Click **Next** until you reach the **Confirm** section
- 7. In the section **Confirm**, enter a unique **Instance Name** and choose **Finish**.

The creation of service instance is successful.

Now, with the help of the created service instance, generate a service key from the steps given below:

Creating a Service Key

- 1. Choose the created service instance link from the visible list.
- 2. In the left-hand pane, navigate to **Service Keys** **Create Service Key**.
- 3. In the **Create Service Key** dialog that opens, provide a **Name** and **Description** (optional).
- 4. In the text box enter one of the following payloads as per your requirement:

To create service key of credential type...	Use payload...	Level of Security	Important Notes	Sample of generated creden
"instance-secret" (without payload)		Low	For instance-secret, the clientSecret generated is same for all the keys.	For admin role: <pre>{ "url": "https://a "tokenUrl": "http "clientId": " "clientSecret</pre>
"instance-secret" (with payload)	<pre>{ "xsuaa": { "credential-type": "instance-secret" } }</pre>	Low	For instance-secret, the clientSecret generated is same for all the keys.	For admin role: <pre>{ "url": "https://a "tokenUrl": "http "clientId": " "clientSecret</pre>

To create service key of credential type...	Use payload...	Level of Security	Important Notes	Sample of generated creden
"binding-secret"	<pre>{ "xsuaa": { "credential-type": "binding-secret" } }</pre>	Medium	For binding-secret, the clientSecret generated for every key is unique.	For admin role: <pre>{ "url": "https "tokenUrl": " "clientId": " "clientSecret</pre>
"x509" (certificate based)	<pre>{ "xsuaa": { "credential-type": "x509", "x509": { "key-length": 2048, "validity": 65, "validity-type": "DAYS" } } }</pre>	High	For X509, ensure that the credential rotation is done based on the validity provided in the payload. For example, delete and create a new service key every 65 days.	For admin role: <pre>{ "url": "https://x "certificate": "x "certurl": "https "clientId": "xxxx "privateKey": "xx "tokenUrl": "http</pre>

5. Click **Save**. The client credentials like **url**, **clientId**, **clientSecret**, and **tokenUrl** details appear for the given instance name.

The application **url** is used to make API calls.

The **clientId** and **clientSecret** are necessary credentials required to fetch the Bearer Token.

The **tokenUrl** is used to fetch the bearer token.

Example:

```
url --cert client.crt --key client.key --location --request POST '<URL from the response>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: No Auth \
--data '{"client_id":"clientId"}' 'grant_type=client_credentials'
```

The generated credentials (tokenUrl, clientId and clientSecret) can then be used by an application developer to authenticate the client (or user) via OAuth, and access the APIs exposed by the service.

Copy the client credentials in a notepad.

i Note

Once your client is setup you can use it to authenticate against the x509 endpoint by providing the client certificate and key.

Create the certificate.cer and certificate.key files based on the public and private keys obtained from the x509 credentials respectively.

For certificate.cer file:

- a. Copy the "certificate" value starting from -----BEGIN CERTIFICATE----- all the way to -----END CERTIFICATE-----\n (the certificate value might contain multiple certificates). Make sure you copy all the certificates.
- b. Paste it in a text editor. Find and replace all the occurrences of \\n by \n.
- c. Save the file as certificate.cer.

For certificate.key file:

- a. Copy the "certificate" value starting from -----BEGIN RSA PRIVATE KEY-----\n... all the way to-----END RSA PRIVATE KEY-----\n
- b. Paste it in a text editor. Find and replace all the occurrences of \\n by \n.
- c. Save the file as certificate.key.

Open a command prompt/terminal in the folder where you have saved the certificate files and execute the following curl command to get the response in the my-oauth-response.json file in the same folder. From this file, you can fetch the bearer token from the value of "access_token".

```
curl --cert certificate.cer --key certificate.key --location --request POST '<certurl from t
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=<clientId from the servicekey x509 credentials>' \
--cert certificate.cer \
--key certificate.key \
> my-oauth-response.json
```

i Note

You can update an already provisioned service instance of an API access plan by performing the following steps:

Prerequisite: You must have the Cloud Foundry CLI installed.

1. Log in to the Cloud Foundry CLI by running the *cf login* command.
2. Select [Org](#).
3. Run the following command to update your service instance. *cf update-service <service-instance-name> -c <empty-json-file>.json*.

Sample Code

Sample json: {}

Next Steps: Obtaining a Bearer Token

In the REST Console:

1. Paste the copied [tokenUrl](#). Append ?grant_type=client_credentials to the [tokenUrl](#).
2. Choose *Basic Auth* as the Authorization type.
3. Similarly, paste the [clientId](#) and [clientSecret](#) in the place of Username and Password.

4. Make a POST Call.

5. Obtain the Bearer Token from the output and copy it in a notepad.

- Now, to trigger an API, in the same REST Console, append the API endpoint (obtained from the API portal APIs that are located in the SAP API Management package in SAP Business Accelerator Hub) to the [url](#).

i Note

Currently, the [apiportal-apiaccess](#) plan allows you to access only the API Management APIs from the SAP API Management package in SAP Business Accelerator Hub.

- Choose *Bearer Token* as the Authorization type and paste the copied Bearer Token in the specified space.
- Include payloads, if needed.

❖ Example

This example summarizes the steps that we have executed so far.

To fetch all the available API proxies in your API portal, you can make a call to the URL that appears in the endpoint<url>/apiportal/api/1.0/Management.svc/APIProxies.

To successfully trigger this API, you need to enable the 'API portal API Access Plan'. Once the plan is enabled and suitable roles are assigned, you can generate a service key using the created service instance. The service key should include your URL, client ID, client secret, and token URL. Copy the URL and append `/apiportal/api/1.0/Management.svc/APIProxies` to it, as shown here: <url>/apiportal/api/1.0/Management.svc/APIProxies

In addition, to establish a connection with this API, you will need to generate a bearer token. Take the token URL and append the grant type to it: `https://token-endpoint-url/oauth/token?grant_type=client_credentials`

Now, execute a GET operation on this API with the Client ID and Client secret as your Basic auth credentials. The bearer token will be embedded in the response code. Next, append the bearer token to the URL as shown here:
<url>/apiportal/api/1.0/Management.svc/APIProxies/<bearer token>

By calling this URL, you will be able to list all the API proxies available in your API portal.

Deleting an API Management, API Portal Service Instance

If you are no longer using the service or if it is not serving its intended purpose, deleting the instance can free up resources and reduce maintenance overhead.


Prerequisites

- You have the *space developer* role assigned to you.
- You have created a service instance under [API Management, API portal](#).

Context

Before deleting an API Management or API Portal instance, it is important to carefully consider the implications and ensure that any necessary backups or data transfers are performed to avoid data loss or service disruption.

Procedure

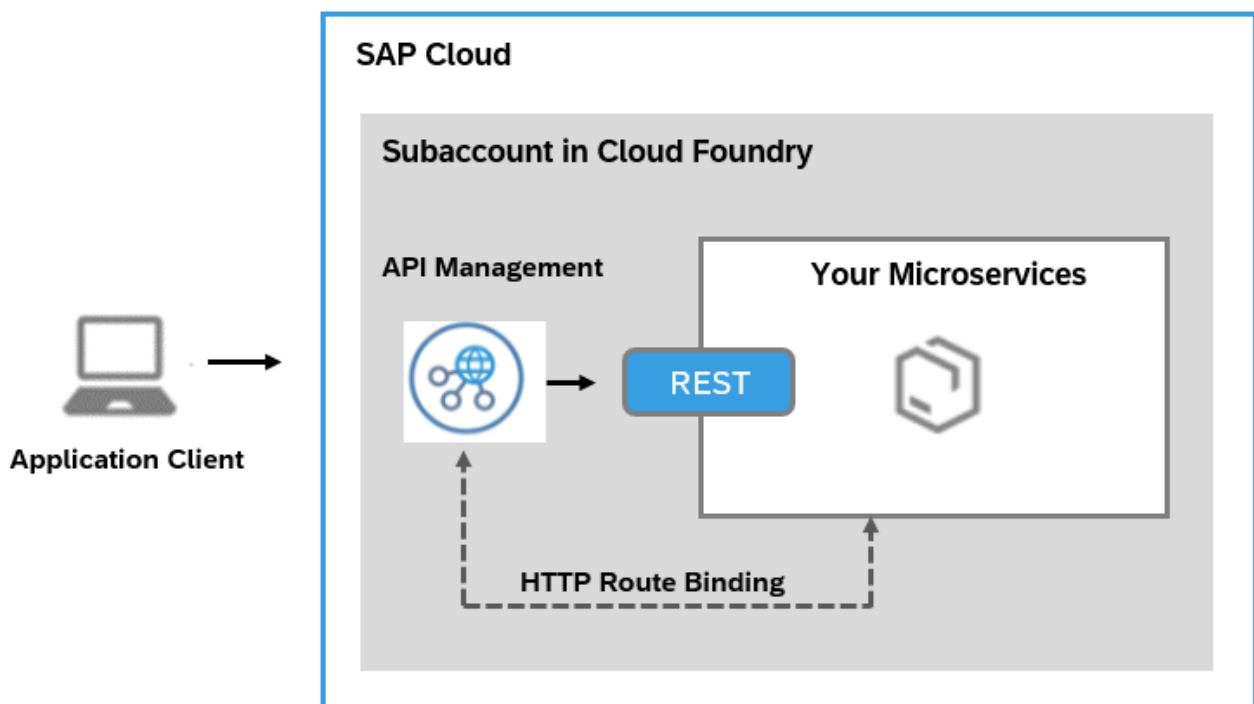
1. In your Web browser, open the [SAP BTP Cockpit](#).
2. In the provider account, choose [Services](#) [Service Marketplace](#) [Instances](#).
3. Select the [API Management, API portal](#) tile.
4. Choose [Instances](#) from the left-hand pane.
5. From the list of instances visible, select the instance that you want to delete and choose .
6. Choose [OK](#).

Managing Cloud Foundry Microservices through API Management

The [apim-as-route-service](#) plan helps you in managing Cloud Foundry applications by including policies such as rate limit, quota. The service instance you create through this plan allows you to bind to the route service and creates an API Proxy. This API Proxy serves in establishing a secure connection with your Cloud Foundry application and all the calls made to the Cloud Foundry application are routed via [API Management, API portal](#).

About the Plan

This service plan is necessary if you have a microservice built and deployed on the SAP Cloud Foundry environment and you want to manage it using API Management. With this plan, even when calling the actual microservice URL (not the API proxified URL), the Cloud Foundry router will ensure that API Management is invoked first. This means that regardless of whether the API proxified URL or the actual microservice URL is called, API Management policies will always be enforced. This functionality is only available for APIs developed and deployed on the Cloud Foundry environment.



The service instance you create through this plan allows you to bind to the route service and creates an API Proxy. This API Proxy serves in establishing a secure connection with your Cloud Foundry application and all the calls made to the Cloud Foundry application are routed via [API Management, API portal](#).

Prerequisites

- You've enabled API Management capability using Integration suite and have completed the setup. For more information, refer [Subscribing to Integration Suite](#) and [Activating Capabilities](#).
- You have the *space developer* role assigned to you.

Creating an API Management, API portal Service Instance

Create a service instance in **API Management, API portal** to start managing your Cloud Foundry applications.

Follow the below procedure to create a service instance on Cloud Foundry:

1. In your web browser, open the **SAP BTP Cockpit** - <https://cockpit.btp.cloud.sap>.
2. From your **Subaccount**, navigate to **Spaces** in your Cloud Foundry environment and choose **Services > Service Marketplace**.
3. Choose **API Management, API portal > Instances > New Instance**.
4. In the **Create Instance** dialog, choose **apim-as-route-service** plan.
5. Choose **Next** until you reach the **Confirm** section.
6. In the section **Confirm**, enter a unique **Instance Name** and choose **Finish**.

i Note

The **apim-as-route-service** plan allows you to create multiple service instances and connect to many Cloud Foundry applications using the same Subaccount.

Binding a Multi-Cloud Foundation Application to an API Management, API Portal Service Instance

Create a service instance and bind the Cloud Foundry application to **API management, API portal** service. When you bind an application, an API proxy is created and a new route is added to the application. The route initially redirects all calls to the proxy URL and then to the application.

Prerequisites

- You have the *space developer* role assigned to you.
- You have created a service instance under **API Management, API portal**.

Open the command-line interface for Cloud Foundry and enter the following command:

Sample Code

```
cf bind-route-service sap-cf-domain.com apim-service-instance-name --hostname my-app -c '{"api_ni

<-- Example
//Without parameters
cf bind-route-service cfapps.sap.hana.ondemand.com apim-prod-instance --hostname taxapp

//Cloud foundry URL for the above example is https://taxapp.cfapps.sap.hana.ondemand.com

//With parameters for Linux/MAC system
cf bind-route-service sap-cf-domain.com apim-service-instance-name --hostname my-app -c '{"api_ni
```

```
//With parameters for Windows system
cf bind-route-service sap-cf-domain.com apim-service-instance-name --hostname my-app -c "{\"api_

-->
```

i Note

API Management, API portal supports only English alpha numerics, hyphens (-) and underscores (_) characters for "api_name".

You can bind an application to a service only from the command-line interface and not from SAP BTP Cockpit.

Providing a value for the parameter during binding is optional. If you provide a value for api_name, then the API proxy created in **API Management, API portal** for current binding gets the given name. Also, if an API with the same name exists in the API portal, then the same API proxy is used for the binding. That is, the API proxy end point is registered as the route service URL for the current binding.

Unbinding a Multi-Cloud Foundation Application from an API Management, API portal Service Instance

When you unbind a multi-cloud foundation application from an API Management, API portal service instance, an API proxy is undeployed and the connection between the route service and the multi-cloud foundation application is removed.

Prerequisites

- You have logged on as a space developer.
- You have bound an application to service instance under **API Management, API portal**.

Procedure

In order to unbind, open the command prompt and enter the following command:

```
cf unbind-route-service sap-cf-domain.com apim-service-instance-name --hostname my-app

<-- Example
cf unbind-route-service cfapps.sap.hana.ondemand.com apim-prod-instance --hostname taxapp
-->
```

i Note

You can unbind an application from a service only from the command-line interface and not from SAP BTP Cockpit.

Accessing On-Premise Systems through API Management

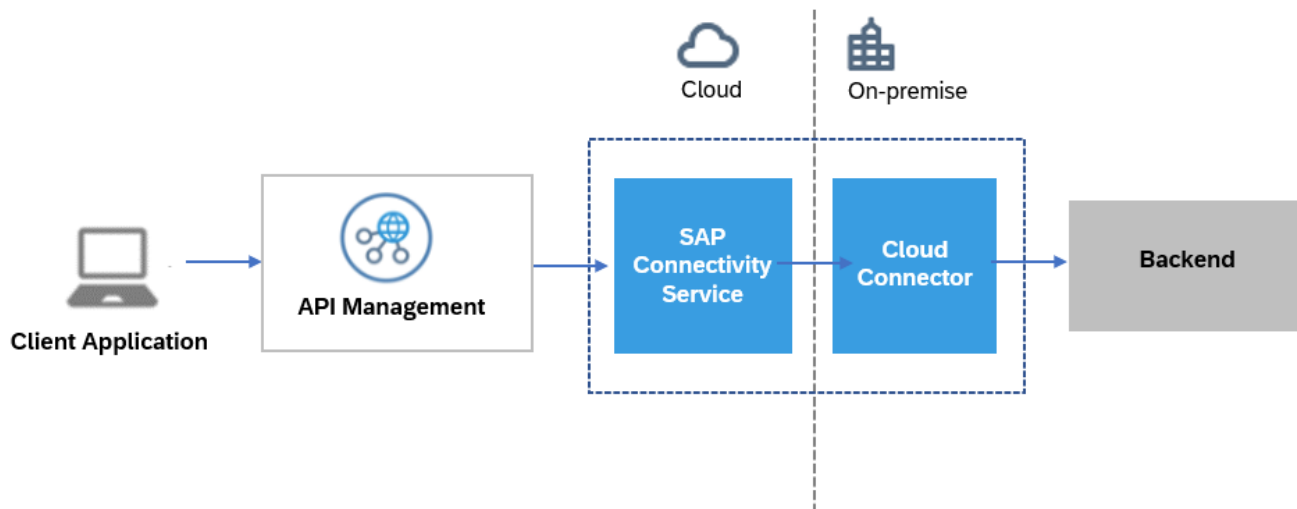
The **on-premise-connectivity** plan helps in achieving principal propagation while connecting to an on-premise backend system.

About the Plan

Let us consider an use case where you want to pass the identity and security context of the logged-in user in the client application (known as the principal) from client application to on-premise backend. It ensures that the downstream services have the necessary information to authenticate the client without requiring the client to re-authenticate for each service. When a client makes a request to an API gateway, the gateway authenticates the user. It then propagates the principal information, such as the user's identity, to the backend services that the client's request needs to access. This allows the downstream services to make authorization decisions based on the user's details.

i Note

The API Management platform incorporates the circuit breaker pattern to enhance the resilience of the back-end. For more information, see [Circuit Breaker for On-Premise Provider](#).



To accomplish principal propagation, you require a service key. This plan allows you to obtain the token by creating a service instance and generating a service key.

This topic explains how to obtain a service key in order to enable principal propagation using **API Management** in the Cloud Foundry Environment.

Prerequisites

- You've enabled API Management capability using Integration suite. For more information, refer [Subscribing to Integration Suite](#) and [Activating Capabilities](#).
- You have created an API Provider of type **On Premise** and chosen **Principal Propagation** as a mode of authentication to connect to an on-premise system. For more information, see [Create an API Provider](#).
- You have the *space developer* role assigned to you.

Creating a Service Instance on API Management, API Portal

Create a service instance to generate a service key that is used to enable the principal propagation.

1. In your web browser, open the **SAP BTP Cockpit** - <https://cockpit.btp.cloud.sap>.
2. From your **Subaccount**, navigate to **Spaces** in your Cloud Foundry environment and choose **Services** > **Service Marketplace**.
3. Choose **API Management, API portal** > **Instances** > **New Instance**.
4. In the **Create Instance** dialog, choose **on-premise-connectivity** plan.
5. Click **Next** until you reach the **Confirm** section.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

6. In the section **Confirm**, enter a unique **Instance Name**, and choose **Finish**.

Service Key Generation

After you have created a service instance, proceed with:

- 1. Choose the created service instance link from the visible list.
- 2. In the left-hand pane, navigate to **Service Keys** **Create Service Key**.
- 3. In the **Create Service** dialog, provide a **Name** and **Description** (optional).
- 4. In the text box enter one of the following payloads as per your requirement:

To create service key of credential type...	Use payload...	Level of Security	Important Notes	Sample of generated creden
"instance-secret" (without payload)		Low	For instance-secret, the clientSecret generated is same for all the keys.	{ "url": "https "clientId": " "clientSecret "orgId": "xxxxxxx "tokenUrl": "http }
"instance-secret" (with payload)	{ "xsuua":{ "credential-type":"instance-secret" } }	Low	For instance-secret, the clientSecret generated is same for all the keys.	{ "url": "https "clientId": " "clientSecret "orgId": "xxxxxxx "tokenUrl": "http }
"binding-secret"	{ "xsuua":{ "credential-type":"binding-secret" } }	Medium	For binding-secret, the clientSecret generated for every key is unique.	{ "url": "https "clientId": " "clientSecret "orgId": "xxxxxxx "tokenUrl": "http }

To create service key of credential type...	Use payload...	Level of Security	Important Notes	Sample of generated creden
"x509" (certificate based)	<pre>{ "xsuaa": { "credential-type": "x509", "x509": { "key-length": 2048, "validity": 65, "validity-type": "DAYS" } } }</pre>	High	For X509, ensure that the credential rotation is done based on the validity provided in the payload. For example, delete and create a new service key every 65 days.	For admin role: <pre>{ "url": "https://x "certificate": "x "certurl": "https "clientId": "xxxx "privateKey": "xx "orgId": "xxxxxxx "tokenUrl": "http }</pre>

5. Choose **Save**. The client credentials like **url**, **clientId**, **clientSecret**, and **tokenUrl** details appear for the given instance name.

The **tokenUrl** is used to fetch the bearer token.

Example:

```
{
  "url": "https://<apiportal application name>.cfapps.sap.hana.ondemand.com",
  "tokenUrl": "https://<Space name>.authentication.sap.hana.ondemand.com/oauth/token",
  "clientId": "sb-opproxy-xsuaa-clonexxxxxxxxxx!xxxxxx|opproxy-xsuaa!xxxxxx",
  "clientSecret": "xxxxxxxxxxxxxxxxxxxxxxxxxx="
}
```

Copy the client credentials in a notepad.

You can now navigate back to Principal Propagation from Neo to the Cloud Foundry environment or Principal Propagation from the same Cloud Foundry subaccount as you have the required client credentials.

You can use the credentials to establish:

- [Principal Propagation from the Neo to the Cloud Foundry Environment](#): Enable an application in your subaccount in the Neo environment to access an API Management proxy created on a Cloud Foundry based **API Management, API portal** without a user login. For this scenario to work, the Neo subaccount needs to be trusted by the Cloud Foundry subaccount where **API Management, API portal** is enabled. Now, the application on Neo can call API Management proxy using OAuth2SAMLBearer destination.
- [Principal Propagation from the Same Cloud Foundry Subaccount](#): Enable an application in your subaccount in the Cloud Foundry environment to access an API Management proxy created on the same Cloud Foundry based API Management, API portal without a user login. The JWT user token in your application can be exchanged with the API Management token using the service key credentials created for **API Management, API portal**.

Additional Configurations for API Management

In this section, you'll find information about different plans for managing API Management, API portal and API business hub enterprise.

You'll also find information on how to request for additional virtual host, custom domain for a virtual host, and two-way SSL certificate.

[Configuring Additional Virtual Host in Cloud Foundry Environment](#)

A virtual host allows you to host multiple domain names on the API Management capability within Integration Suite.

[Region-Specific IP Addresses Available for API Management Cloud Foundry Environment](#)

API Management protects your backend services. However, API Management needs to establish connectivity to your backend services during an API call execution.

[User Roles in API Management](#)

Use role collections to group together different roles that can be assigned to API Portal and API business hub enterprise users.

[Cancel API Management Service Subscription](#)

You can deactivate your API Management capability from Integration Suite to disable your account from the API Management service.

[Setting Up API Management with SAP Cloud Identity Services](#)

SAP Cloud Platform allows customers to connect their SAP Cloud Identity Services with the BTP offerings.

Configuring Additional Virtual Host in Cloud Foundry Environment

A virtual host allows you to host multiple domain names on the API Management capability within Integration Suite.

Create a new virtual host with default domain or custom domain and update alias, keystore, keyalias, and truststore for an existing virtual host in the Cloud Foundry environment.

[Configuring a Custom Domain for a Virtual Host](#)

The API Management capability enables you to personalize the virtual host URL by configuring a custom domain of your choice. This means that you can have all your APIs displayed as "https://api.bestrun.com/..." if desired. Additionally, you have the option to set up multiple virtual hosts using the same custom domain, such as "https://api1.bestrun.com," "https://api2.bestrun.com," and so on.

[Configuring a Default Domain for a Virtual Host](#)

After successful onboarding, API proxies are assigned a default virtual host URL. Currently, this URL uses the domain "ondemand.com," which is the common domain for the Business Technology Platform. It's prefixed with the subdomain consisting of the subaccount name and the data center where the Integration Suite tenant is onboarded. For example, the default host alias could be https://myaccount....eu10.hana.ondemand.com.

[Configuring Mutual TLs for Virtual Host](#)

You can configure mutual TLs for a virtual host, which validates the identities of both the web server and the web client.

Parent topic: [Additional Configurations for API Management](#)

Related Information

[Region-Specific IP Addresses Available for API Management Cloud Foundry Environment](#)

[User Roles in API Management](#)

[Cancel API Management Service Subscription](#)

[Setting Up API Management with SAP Cloud Identity Services](#)

Configuring a Custom Domain for a Virtual Host

The API Management capability enables you to personalize the virtual host URL by configuring a custom domain of your choice. This means that you can have all your APIs displayed as "https://api.bestrun.com/..." if desired. Additionally, you have the option to set up multiple virtual hosts using the same custom domain, such as "https://api1.bestrun.com," "https://api2.bestrun.com," and so on.

Prerequisites

- You must have a service key for the `APIManagement.SelfService.Administrator` role.

To know more about creating a service key for accessing APIs in the API portal, see the **Creating a Service Key** section in [Accessing API Management APIs Programmatically](#).

- You have fetched a valid bearer token. To know more about obtaining a bearer token, see the **Obtaining a Bearer Token** section in [Accessing API Management APIs Programmatically](#).

Context

The advantage of establishing your own custom domain with a virtual host is that it protects you from future changes, such as those that may occur when rehosting your APIs.

In addition, you have the option to secure your custom domain with a default one-way TLS or a mutual TLS. A one-way TLS validates only the secure identity of the API Proxy, providing encryption and authentication for the server. On the other hand, a mutual TLS validates the identities of both the API Proxy and the client, providing mutual authentication.

If you want to configure a mutual TLS, see [Configuring Mutual TLS for Virtual Host](#). This process enables you to establish a more secure connection between your API Proxy and the client, ensuring that both parties can trust each other's identities.

To request a custom domain with one-way TLS, perform the following steps:

Procedure

- Run the services using a standard REST console.
- Service to create a new virtual host:
 - Service URL: `https://<url-from-service-key>/apiportal/operations/1.0/Configuration.svc/VirtualHostRequests`
 - Method: POST
 - Request Header: Authorization: Bearer <token>
 - Content Type: application/json
 - Accept: application/json
 - Request Body:

Sample Code

```
{
  "accountId" : "subdomain of your subaccount",
  "virtualHostUrl": "apis.customdomain.com",
  "isDefaultVirtualHostRequest" : false,
  "isForCustomDomain": true,
  "keyStoreName": "ref://<reference_name>" or "<key_store_name>",
  "keyStoreAlias": "key_store_alias",
  "operation" : "CREATE"
}
```

i Note

- accountId: This is the subdomain of your subaccount.
- virtualHostUrl - This is your virtual host URL which is used to access the deployed API proxies, e.g prod-apis.customdomain.com, testapi.customdomain.com.
- isDefaultVirtualHostRequest -If you want the new virtual host to be the default virtual host, set the value to "true", else set it to "false".
- isForCustomDomain - Ensure that the value is set to "true".
- keyStoreName: The keyStoreName parameter refers to the name of the keystore that should contain the custom domain's public and private key, or the name of the certificate store reference pointing to the keystore. To learn how to create a keystore and upload certificates, see [Manage Certificates](#). Alternatively, you can use a certificate store reference name that points to the keystore containing the custom domain's public and private keys. For more information, see [Working with References](#).
- keyStoreAlias: The keyStoreAlias parameter refers to the name of the keystore certificate containing the custom domain's public and private key. To learn how to create a keystore certificate and upload certificates, see [Manage Certificates](#).

i Note

To enable client authentication (mutual TLS) while configuring the virtual host with custom domain, see [Configuring Mutual TLS for Virtual Host](#).

- o Response: 201

Sample Code

```
{
  "d": {
    "__metadata": {
      "id": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/VirtualH",
      "uri": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/Virtual",
      "type": "apimgmtconfiguration.VirtualHostRequest"
    },
    "accountId": "subdomain of your subaccount",
    "allocatedPort": 443,
    "allocationStatus": "COMPLETE",
    "clusterName": "",
    "id": "1F02AD6A-A53C-43F4-BF95-F053A8A1469A",
    "isClientAuthEnabled": false,
    "isDefaultVirtualHostRequest": false,
    "isForCustomDomain": true,
    "isForNonSni": false,
    "isTLS": false,
    "keyStoreAlias": "key_store_alias",
    "keyStoreName": "ref://<reference_name>" or "<key_store_name>",
    "life_cycle": {
      "__metadata": {
        "type": "apimgmtconfiguration.History"
      },
      "changed_at": "/Date(1707380514905)/",
      "changed_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864",
      "created_at": "/Date(1707380504072)/",
      "created_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864"
    },
    "operation": "CREATE",
    "trustStore": null,
    "virtualHostId": "c269915f-7adc-4f78-bdd0-dd39ffcb079f",
    "virtualHostUrl": "apis.customdomain.com",
    "lbHost": "lbHost url"
  }
}
```

}

The "lbHost" field contains the host URL which is required for the custom domain DNS mapping.

3. Service to update a virtual host:

You can update the virtualHostUrl, keyStoreName, keyStoreAlias, trustStore, and the isDefaultVirtualHostRequest flag

- Service URL: `https://<url-from-service-key>/apiportal/operations/1.0/Configuration.svc/VirtualHostRequests`
- Method: POST
- Request Header: Authentication Bearer <token>
- Content Type: application/json
- Accept: application/json
- Request Body:

Sample Code

```
{
  "accountId" : "subdomain of your subaccount",
  "virtualHostUrl": "apisupdated.customdomain.com",
  "isDefaultVirtualHostRequest" : false,
  "isForCustomDomain": true,
  "keyStoreName": "ref://<reference_name>" or "<key_store_name>",
  "keyStoreAlias": "key_store_alias",
  "operation" : "UPDATE",
  "virtualHostId": "1F02AD6A-A53C-43F4-BF95-F053A8A1469A"
}
```

i Note

- accountId: This is the subdomain of your subaccount.
- virtualHostUrl - This is your virtual host URL which is used to access the deployed API proxies, e.g prod-apis.customdomain.com, testapi.customdomain.com.
- isDefaultVirtualHostRequest -If you want the new virtual host to be the default virtual host, set the value to "true", else set it to "false".
- isForCustomDomain - Ensure that the value is set to "true".
- keyStoreName: The keyStoreName parameter refers to the name of the keystore that should contain the custom domain's public and private key, or the name of the certificate store reference pointing to the keystore. To learn how to create a keystore and upload certificates, see [Manage Certificates](#). Alternatively, you can use a certificate store reference name that points to the keystore containing the custom domain's public and private keys. For more information, see [Working with References](#).
- keyStoreAlias: The keyStoreAlias parameter refers to the name of the keystore certificate containing the custom domain's public and private key. To learn how to create a keystore certificate and upload certificates, see [Manage Certificates](#).
- virtualHostId: This is the unique ID of the virtual host you are trying to update.

i Note

To enable client authentication (mutual TLS) while configuring the virtual host with custom domain, see [Configuring Mutual TLS for Virtual Host](#).

- Response: 201

Sample Code

```
{
  "d": {
    "__metadata": {
      "id": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/VirtualH",
      "uri": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/Virtual",
      "type": "apimgmtconfiguration.VirtualHostRequest"
    },
    "accountId": "subdomain of your subaccount",
    "allocatedPort": 443,
    "allocationStatus": "COMPLETE",
    "clusterName": "",
    "id": "1F02AD6A-A53C-43F4-BF95-F053A8A1469A",
    "isClientAuthEnabled": false,
    "isDefaultVirtualHostRequest": false,
    "isForCustomDomain": true,
    "isForNonSni": false,
    "isTLS": false,
    "keyStoreAlias": "key_store_alias",
    "keyStoreName": "ref://<reference_name>" or "<key_store_name>",
    "life_cycle": {
      "__metadata": {
        "type": "apimgmtconfiguration.History"
      },
      "changed_at": "/Date(1707380514905)/",
      "changed_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864",
      "created_at": "/Date(1707380504072)/",
      "created_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864"
    },
    "operation": "UPDATE",
    "trustStore": null,
    "virtualHostId": "c269915f-7adc-4f78-bdd0-dd39ffcb079f",
    "virtualHostUrl": "apisupdated",
    "lbHost": "lbHost url"
  }
}
```

i Note

The "lbHost" field contains the host URL that is required for the custom domain DNS mapping. If the "lbHost" field does not display any value, please raise a support ticket through the [SAP Support Portal](#) using the component OPU-API-OD-OPS.

i Note

After the virtual host is updated, APIs associated to a product using the updated virtual host must be redeployed and republished.

Task overview: [Configuring Additional Virtual Host in Cloud Foundry Environment](#)

Related Information

[Configuring a Default Domain for a Virtual Host](#)

[Configuring Mutual TLs for Virtual Host](#)

[Configuring Additional Virtual Host in Cloud Foundry Environment](#)

Configuring a Default Domain for a Virtual Host

After successful onboarding, API proxies are assigned a default virtual host URL. Currently, this URL uses the domain "ondemand.com," which is the common domain for the Business Technology Platform. It's prefixed with the subdomain consisting of the subaccount name and the data center where the Integration Suite tenant is onboarded. For example, the default host alias could be `https://myaccount....eu10.hana.ondemand.com`.

Prerequisites

- You must have a service key for the `APIManagement.SelfService.Administrator` role.

To know more about creating a service key for accessing APIs in the API portal, see the **Creating a Service Key** section in [Accessing API Management APIs Programmatically](#).

- You have fetched a valid bearer token. To know more about obtaining a bearer token, see the **Obtaining a Bearer Token** section in [Accessing API Management APIs Programmatically](#).

Context

If you want to configure a mutual TLS, see [Configuring Mutual TLS for Virtual Host](#). This process enables you to establish a more secure connection between your API Proxy and the client, ensuring that both parties can trust each other's identities.

To request a custom domain with one-way TLS, perform the following steps:

Procedure

- Run the services using a standard REST console.
- Service to create a new virtual host:
 - Service URL: `https://<url-from-service-key>/apiportal/operations/1.0/Configuration.svc/VirtualHostRequests`
 - Method: POST
 - Request Header: Authorization: Bearer <token>
 - Content Type: application/json
 - Accept: application/json
 - Request Body:

Sample Code

```
{
  "accountId" : "subdomain of your subaccount",
  "virtualHostUrl": "prod-apis",
  "isDefaultVirtualHostRequest" : false,
  "operation" : "CREATE"
}
```

i Note

- `accountId`: This is the subdomain of your subaccount.
- `virtualHostUrl` - This is your virtual host alias, for example, `prod-apis`, `testapi`.

i Note

The virtual host alias allows a maximum of 63 characters.

- `isDefaultVirtualHostRequest` -if you want the new virtual host to be the default virtual host, set the value to "true", else set it to "false".

i Note

To enable client authentication (mutual TLS) while configuring the virtual host with default domain, see [Configuring Mutual TLS for Virtual Host](#).

- o Response: 201

Sample Code

```
{
  "d": {
    "__metadata": {
      "id": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/VirtualH",
      "uri": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/Virtual",
      "type": "apimgmtconfiguration.VirtualHostRequest"
    },
    "accountId": "subdomain of your subaccount",
    "allocatedPort": 443,
    "allocationStatus": "COMPLETE",
    "clusterName": "",
    "id": "1F02AD6A-A53C-43F4-BF95-F053A8A1469A",
    "isClientAuthEnabled": false,
    "isDefaultVirtualHostRequest": false,
    "isForCustomDomain": false,
    "isForNonSni": false,
    "isTLS": false,
    "keyStoreAlias": null,
    "keyStoreName": null,
    "life_cycle": {
      "__metadata": {
        "type": "apimgmtconfiguration.History"
      },
      "changed_at": "/Date(1707380514905)/",
      "changed_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864",
      "created_at": "/Date(1707380504072)/",
      "created_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864"
    },
    "operation": "CREATE",
    "trustStore": null,
    "virtualHostId": "c269915f-7adc-4f78-bdd0-dd39ffcb079f",
    "virtualHostUrl": "prod-apis.sapdefaultdomain",
    "lbHost": null
  }
}
```

3. Service to update a virtual host:

You can update the virtualHostUrl, trustStore, and the isDefaultVirtualHostRequest flag.

You can also convert your default domain virtual host to custom domain by referring to the update section (Step 3) of the [Configuring a Custom Domain for a Virtual Host](#) topic.

- o Service URL: `https://<url-from-service-key>/apiportal/operations/1.0/Configuration.svc/VirtualHostRequests`
- o Method: POST
- o Request Header: Authentication Bearer <token>
- o Content Type: application/json
- o Request Body:

Sample Code

```
{
  "accountId" : "subdomain of your subaccount",
  "virtualHostUrl": "prod-apis-updated",
```

```

    "isDefaultVirtualHostRequest" : false,
    "operation" : "UPDATE",
    "virtualHostId": "c269915f-7adc-4f78-bdd0-dd39ffcb079f"
  }
<!--
-->

```

i Note

- accountId: This is the subdomain of your subaccount.
- virtualHostUrl - This is your virtual host alias, for example, prod-apis, testapi.

i Note

The virtual host alias allows a maximum of 63 characters.

- isDefaultVirtualHostRequest -if you want the new virtual host to be the default virtual host, set the value to "true", else set it to "false".
- virtualHostId: This is the unique ID of the virtual host you are trying to update.

i Note

To enable client authentication (mutual TLS) while configuring the virtual host with default domain, see [Configuring Mutual TLS for Virtual Host](#).

- o Response: 201

Sample Code

```

{
  "d": {
    "__metadata": {
      "id": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/VirtualH
      "uri": "https://apiportalurl:443/apiportal/api/1.0/Management.svc/Virtual
      "type": "apimgmtconfiguration.VirtualHostRequest"
    },
    "accountId": "subdomain of your subaccount",
    "allocatedPort": 443,
    "allocationStatus": "COMPLETE",
    "clusterName": "",
    "id": "2F02AD6A-A53C-43F4-BF95-F053A8A1469B",
    "isClientAuthEnabled": false,
    "isDefaultVirtualHostRequest": false,
    "isForCustomDomain": false,
    "isForNonSni": false,
    "isTLS": false,
    "keyStoreAlias": null,
    "keyStoreName": null,
    "life_cycle": {
      "__metadata": {
        "type": "apimgmtconfiguration.History"
      },
      "changed_at": "/Date(1707380514905)/",
      "changed_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864",
      "created_at": "/Date(1707380504072)/",
      "created_by": "sb-apiaccess_1705298659201!b109482|api-portal-xsuaa!b11864"
    },
    "operation": "UPDATE",
    "trustStore": null,
    "virtualHostId": "c269915f-7adc-4f78-bdd0-dd39ffcb079f",
    "virtualHostUrl": "prod-apis-updated.sapdefaultdomain",
    "lbHost": null
  }
}

```


}

- Response: 201

i Note

After the virtual host is updated, APIs associated to a product using the updated virtual host must be redeployed and republished.

Task overview: [Configuring Additional Virtual Host in Cloud Foundry Environment](#)

Related Information

[Configuring a Custom Domain for a Virtual Host](#)

[Configuring Mutual TLs for Virtual Host](#)

Configuring Mutual TLs for Virtual Host

You can configure mutual TLs for a virtual host, which validates the identities of both the web server and the web client.

Context

i Note

Since client certificate chains are used in the authentication process to establish the identity of clients accessing the API Management service, it is important to ensure that these chains have sufficient security measures in place. Weak client certificate chains lack the necessary security measures and are therefore vulnerable to attacks. As a result, weak client certificate chains have been deprecated. For more detailed information, please [3418201 - Deprecation of Weak Client Certificate Chains in API Management \(sap.corp\)](#).[🔒]

Procedure

1. Run the services using a standard REST console.
2. Service to configure client authentication for a new or existing virtual host:
 - Service URL: `https://<url-from-service-key>/apiportal/operations/1.0/Configuration.svc/VirtualHostRequests`
 - Method: POST
 - Request Header: Authentication Bearer <token>
 - Content Type: application/json
 - Accept: application/json
 - Request Body:

Create

≡ Sample Code

```
{
  "accountId" : "subdomain of your subaccount",
  "virtualHostUrl": "prod-apis",
  "isDefaultVirtualHostRequest" : false,
```

```

    "isClientAuthEnabled": true,
    "trustStore": "ref://<reference_name>" or "<trust_store_name>",
    "operation" : "CREATE"
  }

```

Update

Sample Code

```

{
  "accountId" : "subdomain of your subaccount",
  "virtualHostUrl": "prod-apis",
  "isDefaultVirtualHostRequest" : false,
  "isClientAuthEnabled": true,
  "trustStore": "ref://<reference_name>" or "<trust_store_name>",
  "virtualHostId": "c269915f-7adc-4f78-bdd0-dd39ffcb079f",
  "operation" : "UPDATE"
}

```

Note

- isClientAuthEnabled: This field must be set to "true" to enable mutual TLS.
- trustStore: This refers to the name of the truststore that holds the client certificate, or name of the certificate store reference that points to the trust store. To learn how to create a truststore and upload certificates, see [Manage Certificates](#). Alternatively, you can use a certificate store reference name instead of the truststore name. This reference name points to the truststore that contains the client certificate. For detailed instructions, see [Working with References](#).

Note

For enabling client authentication (mutual TLS) for custom domain virtual host, append isClientAuthEnabled and trustStore fields to the create/update virtual host request.

Note

After the virtual host is updated, APIs associated to a product using the updated virtual host must be redeployed and republished.

Task overview: [Configuring Additional Virtual Host in Cloud Foundry Environment](#)

Related Information

[Configuring a Custom Domain for a Virtual Host](#)

[Configuring a Default Domain for a Virtual Host](#)

Region-Specific IP Addresses Available for API Management Cloud Foundry Environment

API Management protects your backend services. However, API Management needs to establish connectivity to your backend services during an API call execution.

In case your backend service is restricting access to certain IPs as part of security measures, you need to add API Management NAT IPs to the list of allowed IPs in your backend services.

i Note

- NAT (Network Address Translation) IPs are egress IPs for requests from API Management.
- In the Cloud Foundry environment, IPs are controlled by the respective IaaS provider (AWS, Azure, Google Cloud, Alibaba Cloud). IPs may change due to network updates on the provider side. Any planned changes will be announced at least four weeks before they take effect.

To get region-specific egress (outbound) NAT IP addresses for API Management, see the following table:

Regions for Enterprise Accounts

IaaS Provider	Region	Region Name	Technical Key	Technical Key of IaaS Provider	NAT IPs (egress, for outgoing requests)
Microsoft Azure	eu20	Europe (Netherlands)	cf-eu20	West Europe	51.105.226.79, 20.4.205.181, 20.31.245.126
Microsoft Azure	ap20	Australia (Sydney)	cf-ap20	Australia East	20.53.178.190
Microsoft Azure	ap21	Singapore	cf-ap21	Southeast Asia	20.43.177.113
Microsoft Azure	us20	US West (WA)	cf-us20	West US 2	51.143.126.237
Microsoft Azure	jp20	Japan (Tokyo)	cf-jp20	Japan East	52.155.117.53
Microsoft Azure	us21	US East (VA)	cf-us21	East US	20.42.28.32
Amazon Web Services	br10	Brazil (São Paulo)	cf-br10	sa-east-1	18.229.180.216, 18.230.68.32, 18.229.200.51
Amazon Web Services	jp10	Japan (Tokyo)	cf-jp10	ap-northeast-1	52.69.140.122, 18.181.69.241, 18.182.245.202
Amazon Web Services	ap10	Australia (Sydney)	cf-ap10	ap-southeast-2	3.105.155.212, 13.211.74.25, 13.55.87.26
Amazon Web Services	ap11	Asia Pacific (Singapore)	cf-ap11	ap-southeast-1	54.254.127.94, 54.179.36.212, 54.151.195.2
Amazon Web Services	ap12	Asia Pacific (Seoul)	cf-ap12	ap-northeast-2	3.35.108.250, 54.180.45.228, 3.36.176.209
Amazon Web Services	ca10	Canada (Montreal)	cf-ca10	ca-central-1	3.96.232.61, 3.96.230.37, 15.222.204.174
Amazon Web Services	eu10	Europe (Frankfurt)	cf-eu10	eu-central-1	52.29.48.148, 3.120.95.10, 18.194.144.165, 18.196.191.48, 52.59.78.206, 18.195.138.5, 3.73.160.117, 52.57.130.124, 3.72.189.179
Amazon Web Services	eu11	Europe (Frankfurt)	cf-eu11	eu-central-1	18.156.85.8, 3.65.144.116, 3.121.107.212

IaaS Provider	Region	Region Name	Technical Key	Technical Key of IaaS Provider	NAT IPs (egress, for outgoing requests)
Amazon Web Services	us10	US East (VA)	cf-us10	us-east-1	3.213.79.219, 3.209.244.202, 3.213.81.148, 54.87.110.53, 54.208.172.140, 54.86.163.159
Google Cloud	us30	US Central (IA)	cf-us30	us-central-1	35.223.165.172, 34.133.13.45, 34.72.36.170
Alibaba Cloud	cn40	China (Shanghai)	cf-cn40	cn-shanghai	101.132.190.155, 106.14.165.33, 106.14.184.113
Microsoft Azure	ch20	Switzerland (Zurich)	cf-ch20	Switzerland North	20.250.216.117, 20.250.176.24
Google Cloud	in30	India (Mumbai) GCP	cf-in30	asia-south1	34.100.176.82, 34.93.181.26, 35.200.251.32, 34.100.182.244, 34.93.184.71, 34.100.176.113
Google Cloud	eu30	Europe (Frankfurt) GCP	cf-eu30	europa-west3	34.159.208.178, 34.159.31.39, 34.141.20.163, 34.107.78.67, 34.159.45.83, 35.246.220.63

i Note

In case any discrepancies are observed in the IPs, please create a support ticket on the **OPU-API-OD-OPS** component.

Regions for Trial Accounts

IaaS Provider	Region	Region Name	Technical Key	Technical Key of IaaS Provider	NAT IPs (egress, for outgoing requests)
Microsoft Azure	ap21	Singapore	cf-ap21	Southeast Asia	20.195.52.254
Amazon Web Services	eu10	Europe (Frankfurt)	cf-eu10	eu-central-1	18.157.223.60, 18.157.143.164, 52.28.147.96
Amazon Web Services	us10	US East (VA)	cf-us10	us-east-1	54.172.191.202, 52.1.75.180, 52.207.193.169

i Note

If customers are implementing allow or deny listing based on IPs from their clients to API Management design time applications - API Portal, API Business Hub Enterprise, or other platform services in Cloud Foundry, such as SAP Authorization and Trust Management Service (XSUAA) for fetching tokens, they will need to refer to the documentation for SAP Business Technology Platform Cloud Foundry IP addresses, which can be found at the following link: [Regions and API Endpoints Available for the Cloud Foundry Environment | SAP Help Portal](#).

Parent topic: [Additional Configurations for API Management](#)

Related Information

- [Configuring Additional Virtual Host in Cloud Foundry Environment](#)
- [User Roles in API Management](#)
- [Cancel API Management Service Subscription](#)
- [Setting Up API Management with SAP Cloud Identity Services](#)

User Roles in API Management

Use role collections to group together different roles that can be assigned to API Portal and API business hub enterprise users.

API Portal Roles

Role Collection	Description
<code>APIPortal.Administrator</code>	Use this role to access the API portal user interface (UI) and services, manage the API proxies by adding additional policies, and create products. You can also use this role to manage APIs using the API Designer.
<code>APIPortal.Service.CatalogIntegration</code>	You need this role assigned to you because the client credentials, which are necessary for establishing a connection between the Integration Suite API Management tenant and API business hub enterprise, are generated for this role.
<code>APIManagement.Selfservice.Administrator</code>	Use this role during the onboarding of API Portal and to get access to its settings page.
<code>APIPortal.Guest</code>	Use this role to access the API portal in read-only mode. You can view all APIs, policies, API providers, and analytics, but can't edit them.

API business hub enterprise Roles

Role Collection	Description
<code>AuthGroup.SelfService.Admin</code>	Use this role during the onboarding of API business hub enterprise and to get access to it.
<code>AuthGroup.API.Admin</code>	<p>Use this role to:</p> <ul style="list-style-type: none"> • Manage an application developer's access to the portal by either accepting or rejecting an application developer's request. In addition, you can revoke the access of an existing application developer. • Manage roles for a user by adding new roles or removing existing roles. • On-behalf of an application developer, admin can also perform the following tasks: <ul style="list-style-type: none"> ◦ Create, update, and delete applications. ◦ Create custom attributes for applications. ◦ Provide app key and secret, while creating or updating an application.
<code>AuthGroup.ContentAuthor</code>	<p>Use this role to:</p> <ul style="list-style-type: none"> • Publish content to the API business hub enterprise. • Establish a connection from the API portal to the API business hub enterprise.

Role Collection	Description
AuthGroup.API.ApplicationDeveloper	<p>Use this role to:</p> <ul style="list-style-type: none"> • Access the API business hub enterprise. • Create, update, and delete applications. • View analytics information on application usage, performance, and error count. • View and download bills for subscribed applications.
AuthGroup.Content.Admin	<p>Use this role to:</p> <ul style="list-style-type: none"> • Create and update categories.
AuthGroup.Site.Admin	<p>Use this role to:</p> <ul style="list-style-type: none"> • Configure updates. • Perform portal changes like uploading logo, changing the name and description, and changing the footer links.
AuthGroup.APIPortalRegistration	<p>This role is necessary for creating a connection request between the Integration Suite API Management tenant and the API business hub enterprise. It's also used to update the connection request credentials.</p>

Parent topic: [Additional Configurations for API Management](#)

Related Information

[Configuring Additional Virtual Host in Cloud Foundry Environment](#)

[Region-Specific IP Addresses Available for API Management Cloud Foundry Environment](#)

[Cancel API Management Service Subscription](#)

[Setting Up API Management with SAP Cloud Identity Services](#)

Assigning Role Collections to Users

Role collections enable you to group together the roles you create. The role collections you define can be assigned to users logged on to SAP ID service.

Procedure

1. In your web browser, open **SAP BTP Cockpit** and choose the relevant subaccount.
2. In the left-hand pane, choose **Security > Role Collections**.
3. Choose the role collection to which you want to assign users.
4. Go to the **Users** section and choose **Edit**.
5. Enter the user ID of the user that you want to assign to the role collection. If the user only exists in a connected identity provider, you must choose the identity provider and type in the e-mail address.
6. (Optional) To add more users, choose **+** (Add a user).
7. Save your changes.

Related Information

[Set Up API Portal Application](#)

[Set Up API business hub enterprise Application Using the Standalone Tile](#)

[Shadow Users](#)

Creating a Custom Role

Create a custom role for API products in API Management.

Context


You can restrict access to an API product in API Management using a custom role. That is, only an authorized user can discover and subscribe to API Products that are tagged to a custom role.

i Note

If you're using Integration Suite, refer [Create Roles for Applications Using Existing Role Templates](#) to create a custom role for API Products.

To create a custom role for API Product, use the **ApplicationDeveloper** role template. Also, ensure that for the **CustomRole** attribute, you choose the value of **Source** as **Static**, and in the **Values**, specify the attribute values and press enter. This value is later used to assign permission while creating an API Product.

Procedure

1. Go to your **Subaccount** in **SAP BTP Cockpit** for Cloud Foundry environment.
2. Choose **Service Marketplace** in the left-hand pane.
3. In order to create a custom role, choose the **API Management, API Business Hub Enterprise** tile.
4. Under **Application Plans** for API BusinessHub Enterprise, choose the  **Action** icon and select **Manage Roles**.
5. To add a new custom role, choose  **Add a role**.
6. In the **Create Role** dialog, fill the details for:
 - Role Name
 - Description
 - Role Template: Choose **ApplicationDeveloper**.
 - Attributes: For the **CustomRole** attribute, keep the value of **Source** as **static**. Under **Values** specify the attribute values and press enter.

i Note

Use only alphanumeric characters for the attribute values. Also, the attribute values shouldn't contain any spaces or special characters except for the hyphen '-' and underscore '_'.

It is recommended that you use CamelCase for better readability.

The **Next** option on the **Create Role** dialogue will remain greyed out until you press enter after typing the attribute values in the text box.

Destination Subacco Destination Subaccount Trust

Create Role

1 - 2 Configure Attributes 3 Select Role Collections 4 Review

Configure Attributes

Select the available type and enter the value of your role attributes.

Attributes	Source	Values
CustomRole	Static	ReadOnlyRole X

Previous Next Cancel

This value is later used to assign permission while creating an API product.

A new role is created and added to the Roles list.

7. **Add the created role to Role Collection:** Adding a custom role to the role collection ensures that you choose a specific application and role template.
 - a. Navigate to your ► **Subaccount** ► **Security** ► **Role Collections** ►. Choose ☐ *Create New Role Collection* and provide a **Name** and **Description** to the new role collection.
 - b. Choose the newly created **Role Collection** and choose **Edit**.
 - c. To select the custom role, choose the ☐ icon under the **Roles** tab.
 - d. On the **Select: Role** dialog, choose the custom role from the **Role Name** dropdown, select the checkbox under **Roles**, and choose **Add**.
 - e. Choose **Save**.
8. **Assign role collection to the user:** To assign the created role collection to your authorized email ID:
 - a. Go to the **Users** section and choose **Edit**.
 - b. Enter the user ID of the user that you want to assign to the role collection. If the user only exists in a connected identity provider, you must choose the identity provider and type in the email address.
 - c. (Optional) To add more users, choose ☐ (Add a user).
 - d. Save your changes.

→ Remember

Application Developers who are already onboarded in the API business hub enterprise should have the custom role. If any user has been assigned a custom role but hasn't been onboarded as an application developer in the API business hub enterprise, the application creation fails. In this case, Authgroup.API.Admin can onboard the user as an Application Developer in the portal.

Next Steps

After completing the above steps, assign permissions while creating a Product in API Portal application. For more information on the same refer, [here](#).

Related Information

[Create a Product](#)

[Assign User Roles in API Management](#)

Cancel API Management Service Subscription

You can deactivate your API Management capability from Integration Suite to disable your account from the API Management service.

Deactivate the API Management Capability from Integration Suite

Prerequisite: The [Integration_Provisioner](#) role must be assigned to you.

If you've enabled the API Management capability via Integration Suite, perform the following steps to cancel the API portal and the API business hub enterprise subscriptions:

1. Log on to SAP BTP Cockpit and navigate to your subaccount.
2. Navigate to **Services > Instances and Subscriptions**.
3. Select **Integration Suite** under **Subscriptions**, choose **Actions**, and choose **Go To Application**.

You're navigated to the **Integration Suite** home page where the **Design, Develop, and Manage APIs** tile is displayed.

4. On the **Design, Develop, and Manage APIs** tile, choose **Actions**, and choose **Manage Capability**.

You're navigated to the **Integration Suite Provisioning** page.

5. To cancel the API Management subscription, choose **Deactivate**.

The **Deactivate API Management** confirmation dialog appears.

6. Once you choose **Deactivate**, the **API Management, API Portal** and **API Management, API Business Hub Enterprise** applications are deactivated.

Parent topic: [Additional Configurations for API Management](#)

Related Information

[Configuring Additional Virtual Host in Cloud Foundry Environment](#)

[Region-Specific IP Addresses Available for API Management Cloud Foundry Environment](#)

[User Roles in API Management](#)

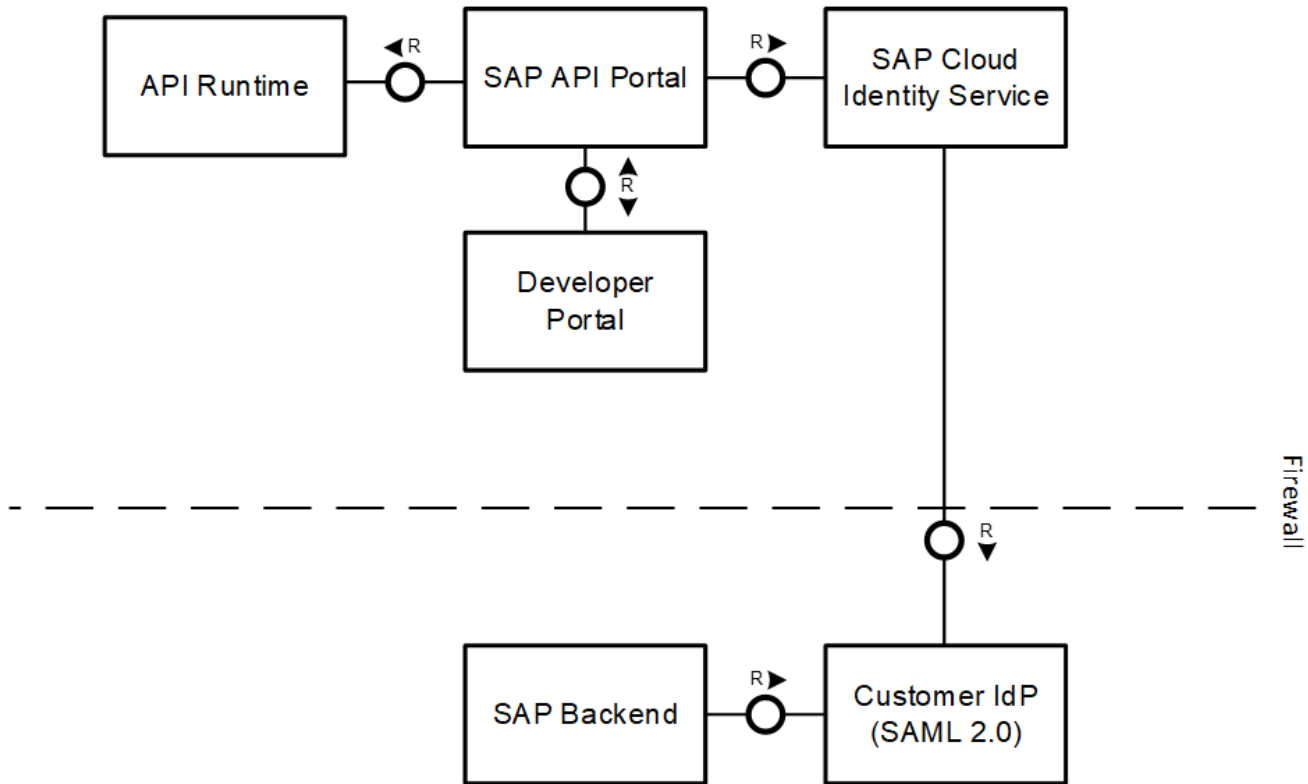
[Setting Up API Management with SAP Cloud Identity Services](#)

Setting Up API Management with SAP Cloud Identity Services

SAP Cloud Platform allows customers to connect their SAP Cloud Identity Services with the BTP offerings.

SAP Cloud Identity (SCI) service supports SAML 2.0 for identity federations. Using SAML, it can federate and connect with any Custom IDP that supports SAML 2.0.

The high-level view for SAP API Management and the SAP Cloud Identity is captured in the diagram below:



[Establishing Trust between SAP Cloud Identity Account and BTP Offerings](#)

Connect SAP Cloud Identity services with the BTP offerings.

[Establishing Trust between SAP HANA Cloud Platform Account and SAP Cloud Identity](#)

Connect SAP HANA Cloud Platform account with the SAP Cloud Identity services.

Parent topic: [Additional Configurations for API Management](#)

Related Information

[Configuring Additional Virtual Host in Cloud Foundry Environment](#)

[Region-Specific IP Addresses Available for API Management Cloud Foundry Environment](#)

[User Roles in API Management](#)

[Cancel API Management Service Subscription](#)

Establishing Trust between SAP Cloud Identity Account and BTP Offerings

Connect SAP Cloud Identity services with the BTP offerings.

Context

To configure and establish trust between SAP Cloud Identity and BTP offerings, follow the steps below:

Procedure

1. Log on to the SAP BTP Cockpit.
2. Navigate to **Trust**. In the **Local Service Provider** tab:

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

- a. Select **Custom** in the **Configuration Type** field.
 - b. Click **Generate Key Pair**.
 - c. Click **Save**.
 - d. Click **Get Metadata** to generate the metadata. This metadata will need to be imported into SAP Cloud Identity.
3. Log on to the SAP Cloud Identity Admin Cockpit (<https://<yourscitenant>.ondemand.com/admin/>).
 4. From the left navigation pane, choose **Applications**, scroll down to the bottom of the page and choose **Add**.
 5. On the **Add Application** dialog, provide an application name, (for example: SAP APIM), and then choose **Save**.
 6. In the newly created application, under the **Trust** tab, choose the **SAML 2.0 Configuration** option.
 7. In the **Define from Metadata** tab, click **Browse** and upload the SAML metadata downloaded from Step 2, and then choose **Save**.
 8. Navigate to **Assertion Attributes** of the SAML 2.0 configurations, and then provide the mapping between the SAML Assertion values. The developer needs these configurations for onboarding on the API business hub enterprise.

The following table contains the mapping between the user attribute and the assertion attribute fields:

User Attribute	Assertion Attribute
Last Name	last_name
Display Name	display_name
E-Mail	mail
First Name	first_name

Task overview: [Setting Up API Management with SAP Cloud Identity Services](#)

Related Information

[Establishing Trust between SAP HANA Cloud Platform Account and SAP Cloud Identity](#)

Establishing Trust between SAP HANA Cloud Platform Account and SAP Cloud Identity

Connect SAP HANA Cloud Platform account with the SAP Cloud Identity services.

Context

To configure and establish trust between SAP HANA Cloud Platform and SAP Cloud Identity, follow the steps below:

Procedure

1. Download the SAML metadata from the **SAP Cloud Identity** by navigating to **Tenant Settings** > **SAML 2.0 Configurations**.
2. In the **SAML 2.0 Configurations** window, scroll down to the bottom of the page and choose **Download Metadata File**.
3. Log on to the SAP HANA Cloud Platform Cockpit.
4. Navigate to **Trust** > **Trusted Identity Provider**, and choose **Add Trusted Identity Provider**.
5. In the **Trusted Identity Provider** window, for the **Metadata File** field, click **Browse** and upload the SAML metadata that was downloaded from Steps 1 and 2.

6. In the **Trusted Identity Provider** window, navigate to the **Attributes** tab, and then add the attributes value.

The following table contains the mapping between the assertion attribute and the principal attribute:

Assertion Attribute	Principal Attribute
first_name	firstname
last_name	lastname
mail	email

7. Now, you can navigate to the API Portal or API business hub enterprise, and login using your SAP Cloud IDP users.


Task overview: [Setting Up API Management with SAP Cloud Identity Services](#)

Related Information

[Establishing Trust between SAP Cloud Identity Account and BTP Offerings](#)

API Documentation

This section contains additional instructions on how to effectively use and integrate with an API.

The standard documentation for API Management APIs is already available on the [SAP Business Accelerator Hub](#).

It provides you with a concise reference manual containing additional information required to work with various entities in the API portal. For example, when to expect a \$batch call, the format of a \$batch call, information about expected headers, payload format, how to create/update single and multiple records, mandatory and optional fields, and, so on.

Limits in API Management

This topic describes the product configuration and the naming conventions for API Management.

Consider the boundary conditions mentioned in the following tables for building, managing, and reviewing the APIs. API Management is designed to perform at its maximum, when it is configured within the specified conditions. Exceeding these values leads to:


- High API latency
- Low API throughput
- Failing API calls.

Currently, certain configurations are automatically enforced for optimal performance.

Product Configurations

Feature	Specified Configuration Values	Automatically Enforced
API Proxies		
Port	Virtual host URL can only be configured only on the secured port 443 over https.	Yes

Feature	Specified Configuration Values	Automatically Enforced
API proxy resource file size (such as XSL, JavaScript or Python).	15 MB	Yes
Maximum number of resources that can be attached to an API proxy	<p>You can attached up to 100 resources to an API proxy. However, it is recommended that you do not add more than 100 resources to an API proxy as it might lead to a timeout while updating or deploying an API proxy.</p> <p>In case you have a business requirement to attach more than 100 resources to an API proxy, please contact the SAP API Management support team by creating a ticket with component OPU-API-OD-DT.</p>	Yes
Virtual Host		
Additional virtual host in Cloud Foundry	By default, only 3 virtual hosts can be configured per tenant. In case you have a business requirement to have more than 3 virtual hosts within a tenant, please raise a support ticket through the SAP Support Portal using the component OPU-API-OD-OPS.	No
Cache and KVM		
Caches	100 MB	No
Cache key size	2 KB	Yes
Cache value size	512 KB	Yes
Cache expiration	>=180 seconds, <= 30 days	No
Key Value Map (KVM) key size	2 KB	Yes
Key Value Map (KVM) value size	10 KB	No
Keys, Developers, Apps, Products		
API key size	2 KB	Yes
Custom attribute name size	255 characters	Yes
Custom attribute value size	1024 characters	Yes
Number of custom attributes permitted	18	Yes
OAuth		
OAuth access token expiration	>= 180 seconds, <= 30 days	No
OAuth refresh token expiration	>= 1 day, <= 90 days	No
OAuth access and refresh token size	2 KB	Yes
System		
API proxy request URL size	7 KB	Yes
Request header size	18 KB	Yes
Response header size	25 KB	Yes
Request size (for non-streamed HTTP requests)	10 MB	Yes

Feature	Specified Configuration Values	Automatically Enforced
Request size (for streamed HTTP requests)	<500 MB. However, if the connection is terminated unexpectedly, you must reinitiate the connection.	No
Response size (for non-streamed HTTP requests)	10 MB	Yes
Timeout (Applicable for all API Proxies and the Backend Server is expected to respond within the value set)	55 Seconds	Yes
Security Protocol	TLSv 1.2 only (on Hyperscalers)	Yes
	TLSv 1.2 & TLSv1.1 (on SAP DC)	Yes
	TLSv1.1 (deprecated, planned to be removed by end of 2019)	No
	TLSv1.0 (Unsupported)	Yes
SNI (Server Name Indication) Enforcement	<p>In API Management, the client is expected to pass a SNI extension (server_name) with target endpoint hostname as part of the initial TLS handshake. Based on the server_name SNI extension the APIM servers will determine the certificate that would be served to the client.</p> <p>For Client which doesn't support SNI extension, APIM would send a default certificate which is provided by SAP.</p> <p>For Reference on SNI : https://en.wikipedia.org/wiki/Server_Name_Indication </p>	Yes
Security Protocol applicable for API Management Runtime	TLSv 1.2 only (on Hyperscalers & SAP DC)	Yes
	TLSv1.0 & TLS 1.1 (Unsupported)	Yes

Naming Conventions

Table below describes the naming constraints for API Management.

Name	Maximum Characters	Permitted Characters
API product		Alphanumeric, space, and the following: _ - . # \$ %
Cache name	255	Alphanumeric
Developer app		Alphanumeric, space, and the following: _ - . # \$ %
E-mail ID		Valid e-mail address syntax
Policy name	255	Alphanumeric and underscore.
Resource file names.	255	Alphanumeric, space, and the following: : / \ ! @ # \$ % ^ & { } [] () _ + - = , . ~ '
Revision name	5	Numeric

Migration of API Management Content

You can choose to clone the API Management content from Neo to Cloud Foundry or between different Cloud Foundry environments.

This table summarizes the migration strategy that we currently support:

Migration Strategy	Types of Migration	Use Cases	More Details
Migration of API Management content from Neo to Cloud Foundry	Standard Migration		Migrating API Management from Neo to the Multi-Cloud Foundation
	Starter Plan Migration (Runtime Reuse Design time Only Migration)	Migration of API Management content within the same Cloud Foundry subaccount	Migrating API Management Subscription Created Using the Starter Plan Service Instance
		Migration of API Management content between different Cloud Foundry subaccounts	Migrating API Management Subscription Created Using the Starter Plan Service Instance to Different Subaccounts
Migration of API Management content from one Cloud Foundry environment to another Cloud Foundry environment	Standard Migration		Migration of API Management Content between Cloud Foundry Environments

Migrating API Management from Neo to the Multi-Cloud Foundation

Migrate the API Management content in Neo environment to a public cloud infrastructure (hyperscalers) within a multi-cloud foundation.

→ **Remember**

SAP Business Technology Platform, Neo environment will sunset on **December 31, 2028**, subject to terms of customer or partner contracts.

For more information, see SAP Note [3351844](#)🔗.

Migration Assistant for asset migration includes the tools and utilities that enable migration of design time assets nondisruptively from the Neo to multi-cloud foundation.

Your source system is the system that contains your API Management content in the Neo environment.

Your target system refers to the infrastructure that hosts your API Management content on a hyperscaler-managed infrastructure within a multi-cloud foundation. This multi-cloud foundation can either be your native standalone API Management subscription or the API Management capability within the Integration Suite.

For the migration assistance, you must have an Integration Suite subscription with API Management capability enabled within Integration Suite.

After completing the prerequisites mentioned in the steps below, you can clone your API Management artifacts nondisruptively from the source to the target system. Post cloning, you must complete some user actions and validate your target system.

i **Note**

The developer portal is renamed to API business hub enterprise within the multi-cloud foundation. In this document API business hub enterprise is referred to as developer portal even within the multi-cloud foundation.

The steps assisting the migration of your API Management from your source system to a target system are:

- 1. [Prerequisites](#)
- 2. [Clone API Management Content](#)
- 3. [Post Cloning Tasks](#)

Prerequisites

Checks to be completed before you start migrating your API Management content nondisruptively from your source system to a target system.

- Your source system is the system that has your API Management subscription in the Neo environment.
- Your target system is the system that has your API Management content on the hyperscalers-managed infrastructure within the multi-cloud foundation.

Prerequisites for the source system

- You must have a valid API Management system (API portal and Developer Portal) running in the Neo environment.
- The source system must support basic authentication for API access on Integration Suite and API business hub enterprise(which is the developer portal).
- Make a note of the Integration Suite and API business hub enterprise(developer portal) URLs of the source system and keep it handy.
- You must have identified a user with the following roles assigned in your source systems:
 - APIPortal.Administrator
 - AuthGroup.API.Admin role

Keep the credentials of this user handy. These credentials are used while filling in the details of the apim-tct-input.json file before running the Tenant Cloning Tool. See [Clone API Management Content](#).

Prerequisites for the target system

- If API Management is not already enabled on your target system, complete the set-up. See [Initial Setup](#) and [Enable API Management Capability](#).

Check whether the API Management service broker service instance is created with the Starter Plan in the same subaccount.

Service Instance for Starter Plan in the Subaccount	Instruction
Already present	<p>You cannot use this subaccount for migration. Create a new subaccount in the hyperscalers-managed infrastructure within the cloud foundry environment and enable API Management on that subaccount for it to act as your target system.</p> <p>Additionally, if you want to reuse the existing runtime then follow the steps mentioned in the Migrating API Management Subscription Created Using the Starter Plan Service Instance.</p>

Service Instance for Starter Plan in the Subaccount	Instruction
Not present	You can choose to reuse this account as your target system for migration, or create a new subaccount.

- If you have already enabled API Management on your target system, and want to reuse the same for migration, it's recommended that you do not have any pre-existing entities such as API proxies or products on this system.

i Note

Any entity, if pre-existing in your target API Management capability, can be over-written during the cloning process.

- If your target system is connected to a custom IDP, ensure that your IDP is configured correctly, and mapping for the details like your first name, last name, email ID, and user ID is done.

i Note

Please ensure that the application developer's attributes, like first name, last name, email ID, and user ID, are identical in both source and target identity providers. In API Management, the application developer's attributes are case-sensitive.

Consider the following example: During cloning, the email address `john.smith@abc.com` in the source becomes `John.Smith@abc.com` in target due to the change in configurations in Custom IDP. This mismatch might lead to data discrepancy during application creation and metering in the target after cloning.

- Ensure that API access is enabled for the Integration Suite and the API business hub enterprise(developer portal) for the following roles:
 - APIPortal.Administrator
 - AuthGroup.API.Admin

For Integration Suite, see [Accessing API Management APIs Programmatically](#).

For API business hub enterprise, execute the following mandatory steps:

- Make a note of the service keys (`url`, `tokenurl`, `clientId`, and `clientSecret`) for the given roles, and keep handy.
- Create a service instance under the **Authorization and Trust Management** tile.
- Create a destination of type **OAuth2Credentials** to the XSUAA APIs by using the credentials you derived from creating the service key.
- Create a service instance with the **AuthGroup.API.Admin** role to access the API business hub enterprise APIs.

To perform the above steps, see [Accessing API business hub enterprise APIs Programmatically](#).

- When you have API products protected by the custom roles permission in the source Neo system, ensure that custom roles creation and assignments are done in the target system within the multi-cloud foundation before starting the migration.

Once you complete these checks, you can start cloning your API Management content from the source to the target system. See [Clone API Management Content](#).

Clone API Management Content

Clone the API Management content using the Tenant Cloning tool.

Once you have your source and target system ready, you can clone your API Management content to the target system by running the Tenant Cloning Tool that you downloaded from [here](#) ➔ .

Prerequisites

- You must have downloaded the Tenant Cloning Tool () from the link provided above. APIM-TCT-<version>.zip
- APIM-TCT-You must have extracted the contents of the APIM-TCT-<version>.zip file into a folder (example name apim-tct).

This extracted folder must contain:

- a java apim-tct-client-<version>.jar file
- a sample apim-tct-input.json file
- a lib folder (this folder and its contents must not be modified)
- a README.md file
- Script files to download open-source libraries that are required to run the apim-tct-client-<version>.jar file:
 - download_dependencies.ps1 for Windows systems
 - download_dependencies.sh for Mac and Linux systems

i Note

Download the dependencies as described in the **Downloading the Dependencies** section.

i Note

If you are using the version of the Tenant Cloning Tool prior to 1.5.2, make sure that you update to the latest version 1.5.2 or above. This is done to handle the critical vulnerability CVE-2021-44228 and CVE-2021-45046, which was detected in the open-source library log4j2.

- The system running the API Management Tenant Cloning Tool must have Java Runtime Environment 8 or above supported.
- Microsoft Excel File Reader

Downloading the Dependencies

For Windows Systems:

- Open the PowerShell terminal.
- Go to the apim-tct folder in the terminal.
- Run the `.\download_dependencies.ps1` command.

The required libraries are downloaded to the lib folder.

For Mac and Linux Systems:

- Open the default terminal from your system.
- Go to the apim-tct folder in the terminal.
- Run the `chmod +x download_dependencies.sh` command to make the file executable.

- Run the `.\download_dependencies.sh` command.

The required libraries are downloaded to the `lib` folder.

i Note

If you encounter an error while running these commands, then you can download the dependencies manually from the link provided in the script file and place them into the `lib` folder.

Context

To migrate all API Management entities, you need to complete the `apim-tct-input.json` file in the tenant cloning tool by providing all the necessary details.

In case you want to migrate selected API proxies from the source API Management tenant to the target API Management tenant, make the following configurations in the `apim-tct-input.json` file:

- Set `selectiveEntityMigration` to `true`
- Provide the names of the API proxies in `selectiveEntities`, separated by commas.

For more information, see [selectiveEntityMigration](#) and [selectiveEntities](#).

By enabling this feature, you can explicitly clone the API proxies mentioned in the configuration file from the source to the target tenant. The cloning process will occur in the following sequence:

- Certificate stores
- Key value maps entries
- API providers
- API proxies

i Note

If `selectiveEntityMigration` is set to `true`, only the certificate stores, key-value maps, API providers, and API proxies will be migrated. Other entities such as products and applications will not be migrated. If it is set to `false` or not available in the `apim-tct-input.json` file, all entities will be considered for migration.

The `selectiveEntityMigration` parameter is optional.

i Note

We recommend migrating all API artifacts during the migration activity. While it is possible to selectively migrate API proxies, this should not be the preferred method for migrating API artifacts. It should only be used with careful consideration of dependencies.

If you need to regularly move or migrate API Management artifacts between Integration Suite tenants, it is recommended to use the transport capability instead. For more information, see [Transport APIs and Its Related Artifacts](#).

Procedure

1. Fill in the `apim-tct-input.json`

Ensure that you don't modify the name of the `apim-tct-input.json` file.

For more information on how to create the service key, refer the [Accessing API Management APIs Programmatically](#) and [Accessing API business hub enterprise APIs Programmatically](#).

Structure of the apim-tct-i

Input Field			Credentials Type	Data Type	Supported Values
source	apiportal	url		String	
		username	Basic	String	
		password		String	
	devportal	url		String	
		username	Basic	String	
		password		String	
	cfSubaccountTenantID			String	Supported values: "guic

Input Field			Credentials Type	Data Type	Supported Values
target	apiportal i Note Choose the relevant fields based on the credential type you've configured for the API access plan. For example, if you've used Client Secret as the credential type, do not select the fields from X509 mTLS.	Url		String	
		tokenUrl	Client Secret	String	
		clientId		String	
		clientSecret		String	
		certurl	X509 mTLS	String	
		certificate		String	
		clientid		String	
		privatekey		String	
	apiportalSelfServiceAdmin i Note Choose the relevant fields based on the credential type you've configured for the API access plan. For example, if you've used Client Secret as the credential type, do not select the fields from X509 mTLS.	Url	Client Secret	String	
		tokenUrl		String	
		clientId		String	
		clientSecret		String	
		certurl	X509 mTLS	String	
		certificate		String	
		clientid		String	
		privatekey		String	
	devportal i Note	url	Client Secret	String	

Input Field			Credentials Type	Data Type	Supported Values
	Choose the relevant fields based on the credential type you've configured for the API access plan. For example, if you've used Client Secret as the credential type, do not select the fields from X509 mTLS.	tokenUrl		String	
		clientId		String	
		clientSecret		String	
		certurl	X509 mTLS	String	
		certificate		String	
		clientid		String	
		privatekey		String	
skipApplicationKeySecretCloning				Boolean	Supported values: true/false
targetDestinationRefreshOnSwitchOver				Boolean	Supported values: true/false
clone	skip-apiportal			Boolean	Supported values: true/false
	skip-devportal			Boolean	Supported values: true/false
stage				string	Supported values: "DEFAULT" "SWITCHOV

Input Field			Credentials Type	Data Type	Supported Values
selectiveEntityMigration				Boolean	Supported values: true/false
selectiveEntities	API proxies			Enter the list of API proxy names in a comma-separated manner as shown below.	

*** apiportalSelfServiceAdmin This input field is mandatory for Starter Plan migration.

*** API portal credentials for source and target for all scenarios are mandatory.

→ Remember

For the clone input attribute:

- Both skip-apiportal and skip-devportal are set to false by default, so, API portal entities are cloned first, followed by Developer Portal entities.
- If both skip-apiportal and skip-devportal are set to true, no cloning takes place.
- If skip-apiportal is set to false, but skip-devportal is set to true, then only the API portal entities are cloned.
- If skip-apiportal is set to true, but skip-devportal to false, then only Developer Portal entities are cloned and cloning for entities (like applications) may fail, pertaining to nonavailability of dependent entity (like API Product) in Developer Portal.

Sample configuration:

```
{
  "source": {
    "apiportal": {
      "url": "<URL of Source (Neo based) API Portal>",
      "username": "<user id having APIPortal.Administrator role in above subscription>",
      "password": "<password of the above user>"
    },
    "devportal": {
      "url": "<URL of Source (Neo based) Developer Portal>",
      "username": "<user id having AuthGroup.API.Admin role in above subscription>",
      "password": "<password of the above user>"
    }
  },

  "target": {
    "apiportal": {
      "url": "<url received during service key creation for API Portal's API Access for
```

```

    "tokenUrl": "<token url received during service key creation for API Portal's API",
    "clientId": "<clientId received during service key creation for API Portal's API",
    "clientSecret": "<clientSecret received during service key creation for API Portal",
  },
  "apiportalSelfServiceAdmin": {
    "url": "<url received during service key creation for API Portal's API Access for",
    "tokenUrl": "<token url received during service key creation for API Portal's API",
    "clientId": "<clientId received during service key creation for API Portal's API",
    "clientSecret": "<clientSecret received during service key creation for API Portal",
  },

  "devportal": {
    "url": "<url received during service key creation for Developer Portal's API Access",
    "tokenUrl": "<token url received during service key creation for Developer Portal",
    "clientId": "<clientId received during service key creation for Developer Portal's",
    "clientSecret": "<clientSecret received during service key creation for Developer",
  }
},

"skipApplicationKeySecretCloning" : <false|true>,

"clone": {
  "skip-apiportal": <false|true> ,
  "skip-devportal": <false|true>
},
"stage": <"DEFAULT" | "SWITCHOVER">
"selectiveEntityMigration": <false|true>, //If you are setting the 'selectiveEntityMigration'
"selectiveEntities": {
  "APIProxies": ["Proxy1", "Proxy2", "Proxy3"]
}

}
```

2. Run the following commands from your Java command-line interface to verify the setup and check the version of the tool. This is an optional step.

- To verify the setup:
`java -jar apim-tct-client-<version>.jar verifyExample:`
- To check the version of the tenant cloning tool you're using:
`java -jar apim-tct-client-<version>.jar version`

3. To begin the cloning process, run the following command from your Java command-line interface:

```
java -jar apim-tct-client-<version>.jar
```

Result

Your API Management entities are now cloned to your target system.

Example:An excel file named `apimtct-output.xlsx` and a log file named `apimtct-logs.log` are generated in the same folder where the `.jar` file is present.

The status of each cloned entity is stored in a separate worksheet within the output excel file.

Structure of a Worksheet Within `apimtct-output.xlsx` File

Column	Description
--------	-------------

Column	Description
ID	Entity ID
Name	Entity name
Type	Entity type
Script Execution Timestamp (UTC)	Script execution time in UTC
Artifact's Last Modified Timestamp (UTC)	Last modified time of the entity in the source API Management system (UTC)
STATUS	Migration Status: <ul style="list-style-type: none">◦ SUCCESS (Entity successfully cloned)◦ FAILURE (Entity failed to clone)◦ SKIPPED (Cloning of Entity skipped)


You can view the status of the cloned content in the `apimtct-output.xlsx` file or in the `apimtct-logs.log` file.

i Note

- Ensure that the `apimtct-output.xlsx` file isn't open while you run the script.
- It's recommended that you don't modify the `apimtct-output.xlsx` file.

Troubleshooting During Cloning:

- If the Tenant Cloning Tool shuts down unexpectedly, restart and try again.

If the tool throws an error repeatedly while running, you can report the incident or error on the component OPU-API-OD-DT through the [SAP Support Portal](#) .

Next Steps

After the cloning process completes, you must perform the tasks mentioned in the `User Actions` worksheet within the output excel file `apimtct-output.xlsx`.

To know more about what actions you must take, see the **User Actions** section in [Post Cloning Tasks](#).

To know more about the entities that are cloned and the entities that aren't cloned, see [Cloned and Uncloned Entities](#).

Cloned and Uncloned Entities

Refer this section for the entities that are cloned and entities that aren't cloned during the migration process.

Entities That Are Cloned

i Note

Currently, when a custom role is assigned to a product, the application creation using the tenant cloning tool is not supported.

As a work-around, before initiating the cloning process, remove the custom role assigned to the product in the source system and proceed with the cloning process.

After the cloning process is completed, reassign the custom roles to the product in the source system. Also, ensure that the custom roles are assigned to the product in the target system.

In case the custom roles aren't appearing in the [Permission](#) tab, as mentioned in the **Prerequisite** section, ensure that the custom roles are created and assigned to the developers in the target multi-cloud foundation.

i Note

If you have made any customizations to the HelloWorld sample proxy, and you want to migrate this proxy to the target, while cloning you might get the following error: "Unable to import API Proxy from zip file; xml content invalid" To address this, execute the following steps:

1. Export the HelloWorld API.
2. Open the zip file and edit the metadata.xml file to add the created_by field as shown below:

Sample Code

```
<life_cycle>
    <changed_by>yourUserId</changed_by>
    <created_by>yourUserID</created_by>

</life_cycle>
```

Please note that your userId is as per your **Identity Service** configuration. You can find your userId when you open any proxies in the API portal.

3. Save the zip file.
4. Delete the existing HelloWorld proxy from the API portal.
5. Import this edited zip file.

With this the created_by will reflect in the API proxy.

The following list displays the API Management entities that can be cloned:

- Certificates and Certificate Store
- Rate Plans
- Key Value Maps
- API Providers
- Policy Templates
- API Proxies
- API Products
- Measure Codes for Custom Measures
- Dimension Codes for Custom Dimensions
- Application
- Application Developer

- Access Control Permissions for API Product
- Custom Metrics and Charts
- Cache Resources
- CertificateStoreReferences

Entities That Are Not Cloned

The following list displays the API Management entities that aren't cloned, including sensitive data like your certificates and credentials.

- **Sensitive Data**
 - Certificates
 - Encrypted Key Value Maps
 - API Provider Passwords
 - Monetization Bills

To know about the actions that you must perform for the uncloned certificates, encrypted key value maps, and API provider passwords, see the **User Actions** section in [Post Cloning Tasks](#).

- **Runtime data**
 - Quota Counters
 - OAuth Tokens for API Proxy runtime calls
 - Runtime states of any API Management entity
- **Configurations**
 - Cloud Connector Setup
 - Custom Role creation and its assignments
 - Default role assignment to users
 - Principal Propagation setup for OpProxy
 - Any configurations created at the subaccount level
 - Any integrations with other systems (like SAP Web IDE)
 - Custom IDP Setup (if any)
 - Existing Route Bindings (if any)

To know about the actions that you must perform for these uncloned entities, see the **Actions required on Configurations** section in [Post Cloning Tasks](#).

Tenant Cloning Tool Behavior

This topic describes the behavior of the Tenant Cloning Tool with respect to cloning some of the entities from your source system.

- If you add or modify an entity in your source system, it is always cloned to the target system in your subsequent run of the Tenant Cloning Tool.
- If you add a new entity to your target system at any point, it is retained in the target system after the subsequent run of the Tenant Cloning Tool, irrespective of whether the entity is present in your source system or not.
- Newer state of an existing entity present in your source system is always migrated to the target system after the subsequent run of the Tenant Cloning Tool, and overwrites any older state of the entity in the target.
- During the cloning of the Developer Portal entity Application Developer, the app developer receives email notifications while being onboarded to the target Developer Portal.

We recommend that you inform your developers about the impending migration and email notifications that they might receive during the process.

- Custom Charts are cloned to the target Integration Suite as many times as you run the Tenant Cloning Tool.
- All the API proxies are cloned onto the default virtual host.
- Post cloning, the API proxies on the target system are in active and deployed state. You must reapply the desired states to the proxies.

To know more about API proxy states, see [API Proxy States](#).

i Note

If the Tenant Cloning Tool is used to clone an API proxy or a product with more than 100 resources attached to it, you might notice data inconsistency in the target system (API business hub enterprise or Integration Suite). It is recommended that you do not add more than 100 resources per proxy or product. For more information, see [Limits in API Management](#).

.

- Cloning of custom chart is now supported for migrating API Management content created using the Starter Plan service instance.

Post Cloning Tasks

Post the completion of the cloning process, you must perform some actions, checks, and validations.

The following sections outline the tasks that need to be completed after the cloning of your API Management content from Neo to the multi-cloud foundation.

User Actions

You can view the status of the cloned artifacts in the apim-tct-output.xlsx excel file or in the apim-tct.log file, generated in the same folder where the .jar file is present.

Perform the tasks mentioned in the User Actions worksheet within the apim-tct-output.xlsx excel file.

The following table describes the actions required for each cloned entity:

Cloned Entity	User Action
---------------	-------------

Cloned Entity	User Action
Certificates	<p>All the certificates that are cloned to the target system are dummy certificates.</p> <p>Perform the following steps:</p> <ol style="list-style-type: none"> 1. From your source system, note down the certificate names and the corresponding certificate store name. 2. From your target system, delete the dummy certificates that were cloned: <ol style="list-style-type: none"> a. In your target API Management, API portal, navigate to ► Configure ► Certificates ►. b. Select the cloned dummy certificate that you want to delete. c. Click the delete icon under the Actions column. 3. In your target API Management, API portal, upload the relevant certificates, providing the same names and under same certificate store as present in your source system. <ol style="list-style-type: none"> a. In your target API Management, API portal, navigate to ► Configure ► Certificates ►. b. Click Create. c. In the Create Certificate window, provide the details and upload the certificate.
Key Value Maps	<p>Fill in the values for the keys of the encrypted Key Value Maps.</p> <ol style="list-style-type: none"> 1. In your target API Management, API portal, navigate to ► Configure ► Key Value Maps ►. 2. Click on the encrypted key value map. 3. In the Edit Key Value Map window, provide the details and click Save.
API Provider Credentials	<p>Provide the Basic Auth password (if present in your source system) for an API Provider.</p> <ol style="list-style-type: none"> 1. In your target API Management, API portal, navigate to ► Configure ► API Providers ►. 2. Click on the desired API provider. 3. In the View API Provider window, click ► Catalog Service Settings ► Edit ►. 4. Provide the basic auth password for the API provider and click Save. <p>Update open connector credentials:</p> <ol style="list-style-type: none"> 1. In your target API Management, API portal, navigate to ► Configure ► API Providers ►. 2. Click on the desired API provider. 3. In the View API Provider window, click ► Catalog Service Settings ► Edit ►. 4. Provide the Org ID and User Secret from your corresponding Open Connector subscription.

Cloned Entity	User Action
Proxy Scoped Key Value Map	<p>Provide instance token value in the proxy scoped Key Value Map.</p> <ol style="list-style-type: none"> 1. In your target API Management, API portal, navigate to Develop > APIs. 2. Click on the desired API proxy. 3. In the View API page, scroll down to Key Value Map Associated and choose the Key Value Map. 4. On the Edit Key Value Map page, update the Value with the instance token value for the corresponding open connector instance. 5. Provide the Org ID and User Secret from your corresponding Open Connector subscription.
<p>Custom Domain based Virtual Host</p> <p>i Note This is only relevant for starter plan migration.</p>	<p>If you have custom domain based virtual host in the source system, then perform the following checks to verify whether the custom domain based virtual hosts are cloned on the target systems:</p> <p>If the URL of virtual hosts looks different in target, which means if the sub domain of the URL is different than the source, revert in the same migration ticket asking the Operations team to set the correct URL for this virtual host.</p> <p>Please ensure that you provide the following virtual host details (from the source) to the Operations team:</p> <ul style="list-style-type: none"> • Custom domain virtual host URL • Virtual host ID <p>The Operations team will use these details to update the virtual host domain in the target system so that it matches with the source.</p> <p>i Note This step has to be completed before starting the Switch Over stage.</p> <p>If you have multiple customain based virtual host, perform the same procedure for each virtual host.</p>

Actions Required on Configurations

Depending on the configurations you have on your source system, you must configure the following in your target system:

- Custom IDP Setup (if any)
- Default role assignment to users
- Custom Role creation and its assignments
- Cloud Connector setup
- Principal Propagation setup at the subaccount level
- Changes to Principal Propagation policy for on-premise connectivity

- Migration of route service bindings. For more information, see [Migrating Route Service Binding](#)
- Any integrations with other systems (like SAP Web IDE)
- Any other configurations that you created for API Management at the subaccount level of your source system

To know more about the entities that are cloned and the entities that aren't cloned, see [Cloned and Uncloned Entities](#).

i Note

For the on-premise APIs, the URL of the target.basepath changes while migrating from Neo to multi-cloud foundation. If you've customized any of the policies, where the target.basepath is being used, then make sure that you update the content of the policy accordingly in the target multi-cloud foundation system. For example, after migration the target basepath URL in multi-cloud foundation might have an additional segment. You need to verify if this additional segment adversely affects the policy execution in target multi-cloud foundation system.

Migrating Route Service Binding

If you've used the [Managing Cloud Foundry Microservices through API Management](#) to manage your multi-cloud foundation applications, you can now migrate the existing route service binding, from the API Management instance on Neo to the new API Management instance on multi-cloud foundation.

Prerequisites

- A route service binding exists between your application on multi-cloud foundation and the API Management service instance in the Neo environment.
- You have enabled API Management on your multi-cloud foundation subaccount.
- You have the space developer role assigned to you.

Depending upon the location of your application, and your API Management service instance, the steps to migrate the route service binding vary.

Multi-Cloud Foundation Application and API Management capability on the same subaccount

If your cloud foundry application and the API Management capability are on the same sub account, then use the following steps to migrate the route service binding:

1. Create an API Management, API portal service instance using the service plan, apim-as-route-service. For more information, see [Creating an API Management, API portal Service Instance](#)
2. Unbind your application from the API Management service instance on Neo. For more information, see [Unbinding a Multi-Cloud Foundation Application from an API Management, API portal Service Instance](#)
3. Bind your application to the API Management service instance on multi-cloud foundation. For more information, see [Binding a Multi-Cloud Foundation Application to an API Management, API portal Service Instance](#)

Multi-Cloud Foundation Application and API Management capability on different sub accounts

If your multi-cloud foundation application and the API Management capability are on different sub accounts, then use the following steps to migrate the route service binding:

1. Create a **User Provided Service** in the subaccount where your multi-cloud foundation application is present, using the proxy URL from the sub account in which your API Management instance is present. In order to create this **User Provided Service**, open the command prompt and use the following command

Sample Code

```
cf create-user-provided-service apim-route-service -r https://apiproxy.url.from.source.  
OK
```

For more information, see [User Provided Service](#) ➦

- 2. Unbind your application from the API Management service instance on Neo. For more information, see [Unbinding a Multi-Cloud Foundation Application from an API Management, API portal Service Instance](#)
- 3. Bind the User Provided Service created in the first step to the multi-cloud foundation application. For this binding, use the following command:

☰ **Sample Code**

```
cf bind-route-service cfapps.eu10.hana.ondemand.com --hostname <your-app-host> apim-route-
```

Validate Your Target API Management System

Validate that all your API Management artifacts have been cloned to the target system and that all your artifacts and route bindings are in working condition.

Switch Over from Source to Target System

You can choose to switch over completely from your source to target system after you've successfully cloned all the entities, performed the post-cloning tasks, and validated that your target system is working correctly.

This section explains the various scenarios for a switch-over:

Switching Over Runtime Proxy URLs

Scenario		Actions Required for Switchover
If you want to retain the same proxy URL as that of your source system	If the proxy URL of your source system is on a domain managed by SAP	There's no option to retain the old proxy URL. You must adopt the new proxy URL that is generated for your target system.
	If the proxy URL of your source system is on a custom domain	1. Update the virtual host of the target system to that of the source system. See Configuring Additional Virtual Host in Cloud Foundry Environment . 2. Perform a DNS change from the old cluster to a new cluster.
If you have multiple virtual hosts configured on your source system subscription, and want to retain those on your target system		1. Create multiple virtual hosts on your target system. See Configuring Additional Virtual Host in Cloud Foundry Environment . 2. Bind each API proxy to the desired virtual host on your target system.

Switching Over Design time URLs of API portal and Developer portals

- Domains managed by SAP can't be switched over.
- To switch over a custom domain, create an incident on the component OPU-API-OD-OPS through the [SAP Support Portal](#) ➦.

Applicable Only During Starter Plan Migration

During the switchover, the Tenant Cloning Tool has a wait time of five minutes for the design-time to connect to the runtime on the target API portal. You can enable this feature by setting the "targetDestinationRefreshOnSwitchOver" parameter to true.

Once this parameter is set to true, the wait time is prompted on the console. Use this time to create a test API proxy on the target API portal and try to deploy the same on the virtual host, which is cloned from the source API portal. Once done, key in **Yes** on the Tenant Cloning Tool console.

i Note

When you try to deploy the test proxy on the target, you might encounter an API proxy deployment error because at this point, the connection between the design time and the runtime is still being refreshed. We recommend that you keep trying until the proxy gets deployed successfully. Without a successful deployment, do not proceed with the next steps.

⚠ Caution

Do not attempt the Stage Switchover of the Tenant Cloning Tool unless it is a "Runtime Reuse Design time Only Migration" scenario.

Recommendations

This topic lists the recommendations that you must consider for migration.

- After cloning the API Management content from the source to the target system for the first time, you must maintain both the systems, until you switch over from the source system to your target system completely.
- It is recommended that you always add or modify your entities in the source system, and clone it to the target system by rerunning the tool as and when required, instead of adding them directly to the target system.

Security Features of the Tenant Cloning Tool

The security features of the Tenant Cloning Tool is described in this section.

Auditing and Logging

- The Tenant Cloning tool calls the APIs provided by Integration Suite and API business hub enterprise(developer portal). Hence, there are no security-related events available in the tool.
- All application logs generated from the cloning tool are stored in "APIM-Tenant-Cloning-Tool.log", an autogenerated log file.

Data Protection and Privacy

- The tool doesn't persist any data on its own, nor is there a persistence layer.
- The tool logs the cloning status in the log file and in the output excel file named "apim-tct-output.xlsx".
 - The log and output excel file contain e-mail IDs of application developers, needed for troubleshooting and migration reporting, which are being cloned from source to target system.
 - The tool doesn't store any personal data (except e-mail IDs of application developers) in the log and output excel file.
 - We recommend storing the log file and output excel file securely, if further processing is needed; else these files must be deleted.

- The tool doesn't read any sensitive personal data.
- The tool doesn't change any personal data.

Identity and Access Management

- No specific identity and access management configuration is needed to run the tool.
- Application developer's details are copied from source to target system as is. If some of the developer information isn't valid in the IDP configured in the target tenant, it must be corrected.

Network and Communication Security

The tool uses standard HTTPS communication to make API calls, as provided by the Integration Suite and API business hub enterprise(developer portal).

Migrating API Management Subscription Created Using the Starter Plan Service Instance

You can choose to migrate the design-time components that you have in the Neo environment, which was previously set up using Starter Plan instance, to the multi-cloud foundation, keeping the runtime components as is.

Context

You can also enable the new API Management design time subscription on the same multi-cloud foundation subaccount, where you have created the starter plan service instance.

i Note

You must subscribe to the API portal and the developer portal in the same multi-cloud foundation subaccount where the starter plan instance is created.

Tenant type (for example, production and test) of the newly onboarded API Management on the multi-cloud foundation must be same as that of the source API Management on the Neo environment.

⚠ Caution

The migration of the Starter Plan Service Instance might involve downtime of the API runtime calls.

Procedure

1. Raise a ticket through the [SAP Support Portal](#). For more information, see [Product Support](#).

Use the following component for your incident:

Component Name	Component Description
OPU-API-OD-OPS	SAP API Management Operations - On Demand

When submitting the incident, include the following information:

- Incident title: Starter Plan Migration

- Description: State that you want to migrate API Management subscription created using the Starter Plan.
- Provide the Neo account details where API Management is enabled.
- Provide the multi-cloud foundation account details where starter plan service instance is created.

i Note

Once you receive a confirmation from SAP on the ticket, you can resume the migration process from step 2.

2. Prepare the target system by enabling the API Management subscription on the multi-cloud foundation subaccount where your starter plan instance was created.

To complete the checks, before you start migrating your API Management artifacts nondisruptively from your source system to a target system, see [Prerequisites](#).

3. Run the Tenant Cloning Tool in the DEFAULT stage. See [Clone API Management Content](#) for more information.

For the list of cloned and uncloned entities, see [Cloned and Uncloned Entities](#). For understanding the behavior of the Tenant Cloning Tool with respect to cloning some of the entities from your source system, see [Tenant Cloning Tool Behavior](#).

i Note

Since this is starter plan migration scenario, only the API portal artifacts at this stage get cloned.

4. After completing the cloning process, you must perform some actions, checks, and validations. For the task details, see [Post Cloning Tasks](#).

For recommendations for migration, refer the following topic: [Recommendations](#)

To know more about the security features of the tenant cloning tool, see [Security Features of the Tenant Cloning Tool](#).

5. Run the Tenant Cloning Tool in the SWITCHOVER stage. For more information, see [Clone API Management Content](#).

i Note

If applicable, API business hub enterprise (developer portal) entities are cloned in this step.

6. After the SWITCHOVER, if you have any API Provider of the type onpremise, provide the Basic Auth password in the target system. For more information, see "API Provider Credentials" under [User Actions](#) in [Post Cloning Tasks](#).

i Note

If you skip this step, the test connection on the particular on-prem provider will fail. Also, the discovery of the on-prem providers will fail. Therefore, please ensure that you complete this step before proceeding.

7. Inform SAP that migration is complete by updating the same ticket.

i Note

If you encounter any issues during the migration process, you can report and track the updates in the same ticket. Therefore, we recommend that you keep the ticket open until you reach step 6.

Results

Migration of API Management subscription created using the Starter Plan service instance is complete. There can be downtime for certain API proxies (having policies that are specific to Neo/ multi-cloud foundation) created out of on-premise providers.

Migrating API Management Subscription Created Using the Starter Plan Service Instance to Different Subaccounts

Migrate the design-time components from the Neo environment, which was previously set up using Starter Plan instance, to the multi-cloud foundation, keeping the runtime components as is.

Context

With the Integration Suite premium edition license available in a different subaccount, you can migrate the API Management design time subscription to this subaccount as well.

i Note

Make a note of the following:

- Analytics data can't be retained, as Advanced Analytics gets newly configured in the different subaccount. However, if you come across any analytics data from the previous subaccount, you must ignore the data and consider the analytics data after the migration task is completed.
- Subscribe to the API portal and the API business hub enterprise in the other multi-cloud foundation subaccount. This is the subaccount with the Integration Suite premium edition license.
- Tenant type (for example, production and test) of the newly onboarded API Management on the multi-cloud foundation must be same as that of the source API Management on the Neo environment.
- Ensure that both the source and the target subaccounts are in the same data center for migrating the API Management subscription to a different subaccount.
- To migrate to a different subaccount, you must provide the input `cfSubaccountTenantID` in the `apim-tct-input.json` file. For more information, see [Clone API Management Content](#).

⚠ Caution

The migration of the Starter Plan Service Instance may involve downtime of the API runtime calls.

Procedure

1. Raise a ticket through the [SAP Support Portal](#)🔗. For more information, see [Product Support](#)🔗.

Use the following component for your incident:

Component Name	Component Description
OPU-API-OD-OPS	SAP API Management Operations - On Demand

When submitting the incident, include the following information:

- Incident title: Starter Plan Migration to Different Subaccounts
- Description: State that you want to migrate API Management subscription created using the Starter Plan to different subaccounts.
- Provide the Neo account details where API Management is enabled.
- Provide the multi-cloud foundation account details where starter plan service instance is created.

- o Provide the details of the target Integration Suite subscription account for migrating the starter plan subscription to a different subaccount. Since you already have the Integration Suite premium license available, you can create the API Management subscription before creating the ticket.

i Note

Once you receive a confirmation from SAP on the ticket, you can resume the migration process from step 2.

2. To avoid any disruption, complete the checks before you start migrating your API Management artifacts from your source system to your target system. For details, see the [Prerequisites](#) section.
3. Run the Tenant Cloning Tool in the DEFAULT stage. See [Clone API Management Content](#) for more information.

For the list of cloned and uncloned entities, see [Cloned and Uncloned Entities](#). For understanding the behavior of the Tenant Cloning Tool with respect to cloning some of the entities from your source system, see [Tenant Cloning Tool Behavior](#).

i Note

Since this is starter plan migration scenario, only the API portal artifacts at this stage get cloned.

4. After completing the cloning process, you must perform some actions, checks, and validations. For the task details, see [Post Cloning Tasks](#).

For recommendations for migration, refer the following topic: [Recommendations](#)

To know more about the security features of the tenant cloning tool, see [Security Features of the Tenant Cloning Tool](#).

5. Run the Tenant Cloning Tool in the SWITCHOVER stage. For more information, see [Clone API Management Content](#).

i Note

If applicable, API business hub enterprise entities are cloned in this step.

6. After the SWITCHOVER, if you have any API Provider of the type onpremise, provide the Basic Auth password in the target system. For more information, see "API Provider Credentials" under [User Actions](#) in [Post Cloning Tasks](#).

i Note

If you skip this step, the test connection on the particular on-prem provider will fail. Also, the discovery of the on-prem providers will fail. Therefore, please ensure that you complete this step before proceeding.

7. Inform SAP that migration is complete by updating the same ticket.

i Note

If you encounter any issues during the migration process, you can report and track the updates in the same ticket. Therefore, we recommend that you keep the ticket open until you reach step 6.

Results

Migration of API Management subscription (created using the Starter Plan service instance) to a different subaccount is complete. There can be downtime for certain API proxies (having policies that are specific to Neo/ multi-cloud foundation) created out of on-premise providers.

Migration of API Management Content between Cloud Foundry Environments

You have the option to migrate your API Management content from one Cloud Foundry environment to another. This migration is possible between tenants within the same data center or between tenants located in different data centers.

i Note

Runtime re-use of the tenant is not yet supported.

At the end of the Cloud Foundry to Cloud Foundry migration, your content from source Cloud Foundry is cloned to the target Cloud Foundry.

Migration Assistant for asset migration includes the tools and utilities that enable migration of design time assets non-disruptively from the Cloud Foundry to the Cloud Foundry environment.

Your source system is the system that has your API Management content in the Cloud Foundry environment.

Your target system is the system that has your API Management content within the Cloud Foundry environment. Here Cloud Foundry environment can be your native standalone API Management subscription or API Management capability within Integration Suite.

Possible Migration Paths

Source Subscription	Target Subscription	Allowed
Standalone	Standalone	Yes
Standalone	Integration Suite	Yes
Integration Suite	Standalone	No
Integration Suite	Integration Suite	Yes

The target system may or may not retain the application key/secret based on the target system runtime cluster. This can be pre-checked with OPS via ticket during initial steps.

If the target can't retain the application key/secret then the applications will be created with a new application key/secret in the target system. You must guide your end users to adopt to the change in application key/secret.

Source and target subscriptions must be in same environment. Both the source and the target subscription must be in production environment or both must be in non-production environment. Cross environment migration is not supported.

You can clone your API Management content non-disruptively from the source to the target system only after completing the steps in the prerequisites section. Post cloning, you must complete some user actions and validate your target system.

i Note

The developer portal is renamed to API business hub enterprise in Cloud Foundry environment. In this document API business hub enterprise is referred to as developer portal even in Cloud Foundry environment.

The steps assisting the migration of your API Management from your source system to a target system are:

- 1. Raise a ticket through the [SAP Support Portal](#). For more information, see [Product Support](#).

Use the following component for your incident:

Component Name	Component Description
OPU-API-OD-OPS	SAP API Management Operations - On Demand

When submitting the incident, include the following information:

- Incident title: API Management Migration from one Cloud Foundry to another Cloud Foundry environment
- Description: State that you want to migrate API Management subscription from one Cloud Foundry to another Cloud Foundry environment .
- Provide the Cloud Foundry account details where API Management is enabled.
- Provide the Cloud Foundry account details where you want to move the data .

i Note

Once you receive a confirmation from SAP on the ticket, you can resume the migration process from step 2.

2. Complete all the steps in the **Prerequisite** section.
3. Clone the API Management content using the Tenant Cloning tool.
4. Complete the post cloning tasks.

Prerequisites for the Source and the Target System

Checks to be completed before you start migrating your API Management content nondisruptively from your source system to a target system.

Both the source and the target system are the system that has your API Management content on the hyperscalers-managed infrastructure within the Cloud Foundry environment.

Prerequisites for the source system

- You must have a valid API Management system (Integration Suite and API business hub enterprise) running in the Cloud Foundry environment.
- The source system must support OAuth client credentials for Cloud Foundry. You need the auth token url and key secret to access the Integration Suite and API business hub enterprise. For more information, refer [Accessing API Management APIs Programmatically](#).
- Make a note of the API portal and API business hub enterprise URLs of the source system and keep handy.
- Ensure that API access is enabled for the Integration Suite and API business hub enterprise systems for the following roles:
 - APIPortal.Administrator
 - AuthGroup.API.Admin

The client id, client secret are used while filling in the details of the `apim-tct-input.json` file before running the Tenant Cloning Tool. See [Clone API Management Content](#).

i Note

During Cloud Foundry to Cloud Foundry migration, the default `input.json` shipped with Tenant Cloning Tool maven zip bundle has username and password as the source field. Please ensure that you change this to token URL, clientId, and client secret as mentioned in the sample configuration in [Clone API Management Content between Cloud Foundry Environments](#).

Prerequisites for the target system

- If API Management is not already enabled on your target system, complete the set-up. For more information, see [Initial Setup](#) and [Enable API Management Capability](#).

Check whether the API Management service broker service instance is created with the Starter Plan in the same subaccount.

Service Instance for Starter Plan in the Subaccount	Instruction
Already present	<p>You can't use this subaccount for migration. Create a new subaccount in the hyperscalers-managed infrastructure within the cloud foundry environment and enable API Management on that subaccount for it to act as your target system.</p> <p>Additionally, if you want to reuse the existing runtime then follow the steps mentioned in the Migrating API Management Subscription Created Using the Starter Plan Service Instance.</p>
Not present	<p>You can choose to reuse this account as your target system for migration, or create a new subaccount.</p>

You can also consider the "Possible Migration Paths" table in [Migration of API Management Content between Cloud Foundry Environments](#) to choose the target subscription type.

If you have already enabled API Management on your target system, and want to reuse the same for migration, you can refer the following recommendations:

- It's recommended that you don't have any pre-existing entities such as API proxies or products on this system.

i Note

Any entity, if pre-existing in your target API Management capability , can be over-written during the cloning process.

- If your target system is connected to a custom IDP, ensure that your IDP is configured correctly, and mapping for the details like your first name, last name, email ID, and user ID is done.

i Note

Please ensure that the application developer's attributes, like first name, last name, email ID, and user ID, are identical in both source and target identity providers. In API Management, the application developer's attributes are case-sensitive.

Consider the following example: During cloning, the email address john.smith@abc.com in the source becomes John.Smith@abc.com in target due to the change in configurations in Custom IDP. This mismatch might lead to data discrepancy during application creation and metering in the target after cloning.

- Ensure that API access is enabled for the Integration Suite and the API business hub enterprise for the following roles:
 - APIPortal.Administrator
 - AuthGroup.API.Admin

For Integration Suite, see [Accessing API Management APIs Programmatically](#).

For API business hub enterprise, execute the following mandatory steps:

- Make a note of the service keys (url, tokenurl, clientId, and clientSecret) for the given roles, and keep handy.
- Create a service instance under the [Authorization and Trust Management](#) tile.

- Create a destination of type **OAuth2Credentials** to the XSUAA APIs by using the credentials you derived from creating the service key.
- Create a service instance with the **AuthGroup.API.Admin** role to access the API business hub enterprise APIs.

To perform the above steps, see [Accessing API business hub enterprise APIs Programmatically](#).

- When you have API products protected by the custom roles permission in the source Cloud Foundry system, ensure that custom roles creation and assignments are done in the target Cloud Foundry environment before starting the migration.

Once you complete these checks, you can start cloning the API Management content from the source to the target system. See [Clone API Management Content between Cloud Foundry Environments](#).

Clone API Management Content between Cloud Foundry Environments

Clone the API Management content using the Tenant Cloning Tool.

Once you have your source and target system ready, you can clone your API Management content to the target system by running the Tenant Cloning Tool that you downloaded from [here](#) 🖱️ .

Prerequisites

- You must have downloaded the Tenant Cloning Tool () from the link provided above. `APIM-TCT-<version>.zip`
- APIM-TCT-You must have extracted the contents of the `APIM-TCT-<version>.zip` file into a folder (example name `apim-tct`).

This extracted folder must contain:

- a `java apim-tct-client-<version>.jar` file
- a sample `apim-tct-input.json` file
- a `lib` folder (this folder and its contents must not be modified)
- a `README.md` file
- Script files to download open-source libraries that are required to run the `apim-tct-client-<version>.jar` file:
 - `download_dependencies.ps1` for Windows systems
 - `download_dependencies.sh` for Mac and Linux systems

i Note

Download the dependencies as described in the **Downloading the Dependencies** section.

i Note

If you are using the version of the Tenant Cloning Tool prior to 1.5.2, make sure that you update to the latest version 1.5.2 or above. This is done to handle the critical vulnerability CVE-2021-44228 and CVE-2021-45046, which was detected in the open-source library log4j2.

- The system running the API Management Tenant Cloning Tool must have Java Runtime Environment 8 or above supported.
- Microsoft Excel File Reader

Downloading the Dependencies

For Windows Systems:

- Open the PowerShell terminal.
- Go to the `apim-tct` folder in the terminal.
- Run the `.\download_dependencies.ps1` command.

The required libraries are downloaded to the `lib` folder.

For Mac and Linux Systems:

- Open the default terminal from your system.
- Go to the `apim-tct` folder in the terminal.
- Run the `chmod +x download_dependencies.sh`
- Run the `.\download_dependencies.sh` command.

The required libraries are downloaded to the `lib` folder.

i Note

If you encounter an error while running these commands, then you can download the dependencies manually from the link provided in the script file and place them into the `lib` folder.

Context

To migrate all API Management entities, you need to complete the `apim-tct-input.json` file in the tenant cloning tool by providing all the necessary details.

In case you want to migrate selected API proxies from the source API Management tenant to the target API Management tenant, make the following configurations in the `apim-tct-input.json` file:

- Set `selectiveEntityMigration` to `true`
- Provide the names of the API proxies in `selectiveEntities`, separated by commas.

For more information, see [selectiveEntityMigration](#) and [selectiveEntities](#).

By enabling this feature, you can explicitly clone the API proxies mentioned in the configuration file from the source to the target. The cloning process will occur in the following sequence:

- Certificate stores
- Key value maps
- API providers
- API proxies

i Note

If `selectiveEntityMigration` is set to `true`, only the above entities will be migrated. Other entities such as products and applications will not be migrated. If it is set to `false` or not available in the `apim-tct-input.json` file, all entities will be considered for migration.

The selectiveEntityMigration parameter is optional.

i Note

We recommend migrating all API artifacts during the migration activity. While it is possible to selectively migrate API proxies, this should not be the preferred method for migrating API artifacts. It should only be used with careful consideration of dependencies.

If you need to regularly move or migrate API Management artifacts between Integration Suite tenants, it is recommended to use the transport capability instead. For more information, see [Transport APIs and Its Related Artifacts](#).

Procedure

- 1. Fill in the apim-tct-input.json file by providing details such as the URLs of your source and target systems, access token URLs, client id, and client secret to your source and target systems.

Ensure that you don't modify the name of the apim-tct-input.json file.

For more information on how to create the service key, refer the [Accessing API Management APIs Programmatically](#) and [Accessing API business hub enterprise APIs Programmatically](#).

Structure of the apim-tct-input

Input Field			Type of Credentials	Data Type	Supported Values	Req
source	apiportal i Note Choose the relevant fields based on the credential type you've configured for the API access plan. For example, if you've used Client Secret as the credential type, do not select the fields from X509 mTLS.	url		String		Req
		tokenUrl	Client Secret	String		Opti
		clientId		String		Opti
		clientSecret		String		Opti
		certurl	X509 mTLS	String		Req
		certificate		String		Req
		clientid		String		Req
		privatekey		String		Req
	devportal i Note	url		String		Req

Input Field			Type of Credentials	Data Type	Supported Values	Req
	Choose the relevant fields based on the credential type you've configured for the API access plan. For example, if you've used Client Secret as the credential type, do not select the fields from X509 mTLS.	tokenUrl	Client Secret	String		Opti
		clientId		String		Opti
		clientSecret		String		Opti
		certurl	X509 mTLS	String		Opti
		certificate		String		Opti
		clientid		String		Opti
		privatekey		String		Opti
	cfSubaccountTenantID			String	Supported values: "guid"	Not
target	apiportal i Note Choose the relevant fields based on the credential type you've configured for the API access plan. For example, if you've used Client Secret as the credential type, do not select the fields from X509 mTLS.	Url		String		Req
		tokenUrl	Client Secret	String		Req
		clientId		String		Opti
		clientSecret		String		Opti
		certurl	X509 mTLS	String		Opti
		certificate		String		Opti
		clientid		String		Opti
	privatekey	String			Opti	

Input Field			Type of Credentials	Data Type	Supported Values	Req
	<div>devportal</div> <div>i Note</div> <div>Choose the relevant fields based on the credential type you've configured for the API access plan. For example, if you've used Client Secret as the credential type, do not select the fields from X509 mTLS.</div>	url		String		Req
		tokenUrl	Client Secret	String		Req
		clientId		String		Opti
		clientSecret		String		Opti
		certurl	X509 mTLS	String		Opti
		certificate		String		Opti
		clientid		String		Opti
		privatekey		String		Opti
skipApplicationKeySecretCloning				Boolean	Supported values: true/false	Opti
clone	skip-apiportal			Boolean	Supported values: true/false	Opti
	skip-devportal			Boolean	Supported values: true/false	Opti
stage				string	Supported values: "DEFAULT" "SWITCHOVER	Opti

Input Field			Type of Credentials	Data Type	Supported Values	Req
selectiveEntityMigration				Boolean	Supported values: true/false	Opti
selectiveEntities	API proxies			Enter the list of API proxy names in a comma-separated manner as shown below.		Opti

*** API portal credentials for source and target for all scenarios are mandatory.

→ Remember

For the clone input attribute:

- Both skip-apiportal and skip-devportal are set to false by default, so, API portal entities are cloned first, followed by API business hub enterprise entities.
- If both skip-apiportal and skip-devportal are set to true, no cloning takes place.
- If skip-apiportal is set to false, but skip-devportal is set to true, then only the API portal entities are cloned.
- If skip-apiportal is set to true, but skip-devportal to false, then only API business hub enterprise entities are cloned and cloning for entities (like applications) may fail, pertaining to nonavailability of dependent entity (like API Product) in API business hub enterprise.

Sample configuration:

```
{
  "source": {
    "apiportal": {
      "url": "<URL of Source (Cloud Foundry based) API Portal>",
      "tokenUrl": "<token url received during service key creation for API Portal's API",
      "clientId": "<clientId received during service key creation for API Portal's",
      "clientSecret": "<clientSecret received during service key creation for API P
    },
    "devportal": {
      "url": "<URL of Source (Cloud Foundry based) API business hub enterprise>",
      "tokenUrl": "<token url received during service key creation for API business hub",
      "clientId": "<clientId received during service key creation for API business I",
      "clientSecret": "<clientSecret received during service key creation for API bi
    }
  },
  "target": {
    "apiportal": {
```

```

    "url": "<URL of Source (Cloud Foundry based) API Portal>",
    "tokenUrl": "<token url received during service key creation for API Portal's API",
    "clientId": "<clientId received during service key creation for API Portal's API",
    "clientSecret": "<clientSecret received during service key creation for API Portal",
  },

  "devportal": {
    "url": "<URL of Source (Cloud Foundry based) API business hub enterprise>",
    "tokenUrl": "<token url received during service key creation for API business hub",
    "clientId": "<clientId received during service key creation for API business hub",
    "clientSecret": "<clientSecret received during service key creation for API busin",
  }
},

"skipApplicationKeySecretCloning" : <false|true>,

"clone": {
  "skip-apiportal": <false|true> ,
  "skip-devportal": <false|true>
},
"stage": <"DEFAULT">,
"selectiveEntityMigration": <false|true>, //If you are setting the 'selectiveEntityMigration'
"selectiveEntities": {
  "APIProxies": ["Proxy1", "Proxy2", "Proxy3"]
}

}
```

2. Run the following commands from your Java command-line interface to verify the setup and check the version of the tool. This is an optional step.

- To verify the setup:
`java -jar apim-tct-client-<version>.jar verify`
- To check the version of the tenant cloning tool you're using:
`java -jar apim-tct-client-<version>.jar version`

3. To begin the cloning process, run the following command from your Java command-line interface:

```
java -jar apim-tct-client-<version>.jar
```

Result

Your API Management entities are now cloned to your target system.

An excel file named `apimtct-output.xlsx` and a log file named `apimtct-logs.log` are generated in the same folder where the `.jar` file is present.

The status of each cloned entity is stored in a separate worksheet within the output excel file.

Structure of a Worksheet Within `apimtct-output.xlsx` File

Column	Description
ID	Entity ID
Name	Entity name
Type	Entity type

Column	Description
Script Execution Timestamp (UTC)	Script execution time in UTC
Artifact's Last Modified Timestamp (UTC)	Last modified time of the entity in the source API Management system (UTC)
STATUS	Migration Status: <ul style="list-style-type: none"> ◦ SUCCESS (Entity successfully cloned) ◦ FAILURE (Entity failed to clone) ◦ SKIPPED (Cloning of Entity skipped)


You can view the status of the cloned content in the `apimtct-output.xlsx` file or in the `apimtct-logs.log` file.

i Note

- Ensure that the `apimtct-output.xlsx` file isn't open while you run the script.
- It's recommended that you don't modify the `apimtct-output.xlsx` file.

Troubleshooting During Cloning:

- If the Tenant Cloning Tool shuts down unexpectedly, restart and try again.

If the tool throws an error repeatedly while running, you can report the incident or error on the component OPU-API-OD-DT through the [SAP Support Portal](#) .

Next Steps

After the cloning process completes, you must perform the tasks mentioned in the **User Actions** worksheet within the output excel file `apimtct-output.xlsx`.

To know more about what actions you must take, see the **User Actions** section in [Post Cloning Tasks](#).

To know more about the entities that are cloned and the entities that aren't cloned, see [Cloned and Uncloned Entities](#).

Cloned and Uncloned Entities

Refer this section for the entities that are cloned and entities that aren't cloned during the migration process.

Entities That Are Cloned

i Note

Currently, when a custom role is assigned to a product, the application creation using the tenant cloning tool is not supported.

As a work-around, before initiating the cloning process, remove the custom role assigned to the product in the source system and proceed with the cloning process.

After the cloning process is completed, reassign the custom roles to the product in the source system. Also, ensure that the custom roles are assigned to the product in the target system.

In case the custom roles aren't appearing in the [Permission](#) tab, as mentioned in the prerequisite section, ensure that the custom roles are created and assigned to the developers in the target Cloud Foundry environment.

i Note

If you have made any customizations to the HelloWorld sample proxy, and you want to migrate this proxy to the target, while cloning you might get the following error: "Unable to import API Proxy from zip file; xml content invalid". To address this, execute the following steps:

1. Export the HelloWorld API.
2. Open the zip file and edit the metadata.xml file to add the created_by field as shown below:

Sample Code

```
<life_cycle>
    <changed_by>yourUserId</changed_by>
    <created_by>yourUserID</created_by>

</life_cycle>
```

Please note that your userId is as per your Identity Service configuration. You can find your userId when you open any proxies in the API portal.

3. Save the zip file.
4. Delete the existing HelloWorld proxy from Integration Suite.
5. Import this edited zip file.

With this the created_by will reflect in the API proxy.

The following list displays the API Management entities that can be cloned:

- Certificates and Certificate Store
- Rate Plans
- Key Value Maps
- API Providers
- Policy Templates
- API Proxies
- API Products
- Application
- Application Developer
- Access Control Permissions for API Product
- Cache Resources
- CertificateStoreReferences

Content That Are Not Cloned

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

The following list displays the API Management content that aren't cloned, including sensitive data like your certificates and credentials.

- **Analytics Data**

- Measure Codes for Custom Measures
- Dimension Codes for Custom Dimensions
- Custom Metrics and Charts

i Note

These entities are not cloned via the cloning tool. You have to create them manually in Integration Suite target system under **Analytics**. For more information, see [Advanced API Analytics](#).

- **Sensitive Data**

- Certificates
- Encrypted Key Value Maps
- API Provider Passwords
- Monetization Bills

To know about the actions that you must perform for the uncloned certificates, encrypted key value maps, and API provider passwords, see the **User Actions** section in [Post Cloning Tasks](#).

- **Runtime data**

- Quota Counters
- OAuth Tokens for API Proxy runtime calls
- Runtime states of any API Management entity

- **Configurations**

- Cloud Connector Setup
- Custom Role creation and its assignments
- Default role assignment to users
- Principal Propagation setup for OpProxy
- Any configurations created at the subaccount level
- Any integrations with other systems (like SAP Web IDE)
- Custom IDP Setup (if any)
- Existing Route Bindings (if any)

To know about the actions that you must perform for these uncloned content, see the **Actions required on Configurations** section in [Post Cloning Tasks](#).

Tenant Cloning Tool Behavior

This topic describes the behavior of the Tenant Cloning Tool with respect to cloning some of the entities from your source system.

- If you add or modify an entity in your source system, it is always cloned to the target system in your subsequent run of the Tenant Cloning Tool.
- If you add a new entity to your target system at any point, it is retained in the target system after the subsequent run of the Tenant Cloning Tool, irrespective of whether the entity is present in your source system or not.
- Newer state of an existing entity present in your source system is always migrated to the target system after the subsequent run of the Tenant Cloning Tool, and overwrites any older state of the entity in the target.
- During the cloning of the Developer Portal entity Application Developer, the app developer receives email notifications while being onboarded to the target Developer Portal.

We recommend that you inform your developers about the impending migration and email notifications that they might receive during the process.

- Custom Charts are cloned to the target Integration Suite as many times as you run the Tenant Cloning Tool.
- All the API proxies are cloned onto the default virtual host.
- Post cloning, the API proxies on the target system are in active and deployed state. You must reapply the desired states to the proxies.

To know more about API proxy states, see [API Proxy States](#).

i Note

If the Tenant Cloning Tool is used to clone an API proxy or a product with more than 100 resources attached to it, you might notice data inconsistency in the target system (API business hub enterprise or Integration Suite). It is recommended that you do not add more than 100 resources per proxy or product. For more information, see [Limits in API Management](#).

.

- Cloning of custom chart is now supported for migrating API Management content created using the Starter Plan service instance.

Post Cloning Tasks

Post the completion of the cloning process, you must perform some actions, checks, and validations.

The following sections explain the tasks that you must perform after the cloning of your API Management artifacts from the Cloud Foundry to the Cloud Foundry environment is complete.

User Actions

You can view the status of the cloned artifacts in the `apim-tct-output.xlsx` excel file or in the `apim-tct.log` file, generated in the same folder where the `.jar` file is present.

Perform the tasks mentioned in the User Actions worksheet within the `apim-tct-output.xlsx` excel file.

The following table describes the actions required for each cloned entity:

Cloned Entity	User Action
---------------	-------------

Cloned Entity	User Action
Certificates	<p>All the certificates that are cloned to the target system are dummy certificates.</p> <p>Perform the following steps:</p> <ol style="list-style-type: none"> 1. From your source system, note down the certificate names and the corresponding certificate store name. 2. From your target system, delete the dummy certificates that were cloned: <ol style="list-style-type: none"> a. In your target API Management, API portal, navigate to ► Configure ► Certificates ►. b. Select the cloned dummy certificate that you want to delete. c. Click the delete icon under the Actions column. 3. In your target API Management, API portal, upload the relevant certificates, providing the same names and under same certificate store as present in your source system. <ol style="list-style-type: none"> a. In your target API Management, API portal, navigate to ► Configure ► Certificates ►. b. Click Create. c. In the Create Certificate window, provide the details and upload the certificate.
Key Value Maps	<p>Fill in the values for the keys of the encrypted Key Value Maps.</p> <ol style="list-style-type: none"> 1. In your target API Management, API portal, navigate to ► Configure ► Key Value Maps ►. 2. Click on the encrypted key value map. 3. In the Edit Key Value Map window, provide the details and click Save.
API Provider Credentials	<p>Provide the Basic Auth password (if present in your source system) for an API Provider.</p> <ol style="list-style-type: none"> 1. In your target API Management, API portal, navigate to ► Configure ► API Providers ►. 2. Click on the desired API provider. 3. In the View API Provider window, click ► Catalog Service Settings ► Edit ►. 4. Provide the basic auth password for the API provider and click Save. <p>Update open connector credentials:</p> <ol style="list-style-type: none"> 1. In your target API Management, API portal, navigate to ► Configure ► API Providers ►. 2. Click on the desired API provider. 3. In the View API Provider window, click ► Catalog Service Settings ► Edit ►. 4. Provide the Org ID and User Secret from your corresponding Open Connector subscription.

Cloned Entity	User Action
Proxy Scoped Key Value Map	<p>Provide instance token value in the proxy scoped Key Value Map.</p> <ol style="list-style-type: none">1. In your target API Management, API portal, navigate to Develop > APIs.2. Click on the desired API proxy.3. In the View API page, scroll down to Key Value Map Associated and choose the Key Value Map.4. On the Edit Key Value Map page, update the Value with the instance token value for the corresponding open connector instance.5. Provide the Org ID and User Secret from your corresponding Open Connector subscription.

Actions Required on Configurations

Depending on the configurations you have on your source system, you must configure the following in your target system:

- Custom IDP Setup (if any)
- Default role assignment to users
- Custom Role creation and its assignments
- Cloud Connector setup
- Principal Propagation setup at the subaccount level
- Changes to Principal Propagation policy for on-premise connectivity
- Migration of route service bindings. For more information, see [Migrating Route Service Binding](#)
- Any integrations with other systems (like SAP Web IDE)
- Any other configurations that you created for API Management at the subaccount level of your source system

To know more about the entities that are cloned and the entities that aren't cloned, see [Cloned and Uncloned Entities](#).

Migrating Route Service Binding

If you've used the [Managing Cloud Foundry Microservices through API Management](#) to manage your Cloud Foundry applications, you can now migrate the existing route service binding, from the API Management instance on Cloud Foundry to the new API Management instance on Cloud Foundry.

Prerequisites

- A route service binding exists between your application on Cloud Foundry and the API Management service instance in the Cloud Foundry environment.
- You have enabled API Management on your Cloud Foundry sub account
- You have the space developer role assigned to you.

Depending upon the location of your application, and your API Management service instance, the steps to migrate the route service binding vary.

Cloud Foundry Application and API Management capability on the same subaccount

If your cloud foundry application and the API Management capability are on the same sub account, then use the following steps to migrate the route service binding:

- 1. Create an API Management, API portal service instance using the service plan, apim-as-route-service. For more information, see [Creating an API Management, API portal Service Instance](#)
- 2. Unbind your application from the API Management service instance on Cloud Foundry. For more information, see [Unbinding a Multi-Cloud Foundation Application from an API Management, API portal Service Instance](#)
- 3. Bind your application to the API Management service instance on Cloud Foundry. For more information, see [Binding a Cloud Foundry Application to an API Management, API portal Service Instance](#)

Cloud Foundry Application and API Management capability on different sub accounts

If your Cloud Foundry application and the API Management capability are on different sub accounts, then use the following steps to migrate the route service binding:

- 1. Create a User Provided Service in the sub account where your Cloud Foundry application is present, using the proxy URL from the sub account in which your API Management instance is present. In order to create this User Provided Service, open the command prompt and use the following command

≡ Sample Code

```
cf create-user-provided-service apim-route-service -r https://apiproxy.url.from.source.  
OK
```

For more information, see [User Provided Service](#)

- 2. Unbind your application from the API Management service instance on Cloud Foundry. For more information, see [Unbinding a Multi-Cloud Foundation Application from an API Management, API portal Service Instance](#)
- 3. Bind the User Provided Service created in the first step to the Cloud Foundry Application. For this binding, use the following command:

≡ Sample Code

```
cf bind-route-service cfapps.eu10.hana.ondemand.com --hostname <your-app-host> apim-route-
```

Validate Your Target API Management System

Validate that all your API Management artifacts have been cloned to the target system and that all your artifacts and route bindings are in working condition.

Switch Over from Source to Target System

You can choose to switch over completely from your source to target system after you've successfully cloned all the entities, performed the post-cloning tasks, and validated that your target system is working correctly.

This section explains the various scenarios for a switch-over:

Switching Over Runtime Proxy URLs

Scenario	Actions Required for Switchover
----------	---------------------------------


Scenario		Actions Required for Switchover
If you want to retain the same proxy URL as that of your source system	If the proxy URL of your source system is on a domain managed by SAP	There's no option to retain the old proxy URL. You must adopt the new proxy URL that is generated for your target system.
	If the proxy URL of your source system is on a custom domain	1. Update the virtual host of the target system to that of the source system. See Configuring Additional Virtual Host in Cloud Foundry Environment . 2. Perform a DNS change from the old cluster to a new cluster.
If you have multiple virtual hosts configured on your source system subscription, and want to retain those on your target system		1. Create multiple virtual hosts on your target system. See Configuring Additional Virtual Host in Cloud Foundry Environment . 2. Bind each API proxy to the desired virtual host on your target system.

i Note

If your source and target belongs to the same data center and your source has a custom domain virtual host, and if you are planning to carry forward the same custom domain virtual host to target, please ensure that the following aspects are considered:

1. Since custom domain virtual host URL and port should be unique in a data center accross tenants. It is not possible to have the same virtual host URL in both source and target at the same time. Therefore, delete the custom domain virtual host from source and then create the same custom domain virtual host in the target. To do this, you must create an incident on the component OPU-API-OD-OPS through the SAP Support Portal. For details, refer [Configuring Additional Virtual Host in Cloud Foundry Environment](#).
2. When virtual host gets deleted in the source tenant, there will be downtime for all the APIs in the source account. The downtime will continue until the virtual host configuration gets completed. This configuration activity will require manual intervention by the API Management Operations team and also your DNS service provider for DNS cutover. We recommend that you plan this activity during your planned maintenance window.

Switching Over Design time URLs of API portal and API business hub enterprise portals

- Domains managed by SAP can't be switched over.
- To switch over a custom domain, create an incident on the component OPU-API-OD-OPS through the [SAP Support Portal](#) .

Recommendations

This topic lists the recommendations that you must consider for migration.

- After cloning the API Management content from the source to the target system for the first time, you must maintain both the systems, until you switch over from the source system to your target system completely.

- It is recommended that you always add or modify your entities in the source system, and clone it to the target system by rerunning the tool as and when required, instead of adding them directly to the target system.

Security Features of the Tenant Cloning Tool

The security features of the Tenant Cloning Tool is described in this section.

Auditing and Logging

- The Tenant Cloning tool calls the APIs provided by Integration Suite and API business hub enterprise(developer portal). Hence, there are no security-related events available in the tool.
- All application logs generated from the cloning tool are stored in "APIM-Tenant-Cloning-Tool.log", an autogenerated log file.

Data Protection and Privacy

- The tool doesn't persist any data on its own, nor is there a persistence layer.
- The tool logs the cloning status in the log file and in the output excel file named "apim-tct-output.xlsx".
 - The log and output excel file contain e-mail IDs of application developers, needed for troubleshooting and migration reporting, which are being cloned from source to target system.
 - The tool doesn't store any personal data (except e-mail IDs of application developers) in the log and output excel file.
 - We recommend storing the log file and output excel file securely, if further processing is needed; else these files must be deleted.
- The tool doesn't read any sensitive personal data.
- The tool doesn't change any personal data.

Identity and Access Management

- No specific identity and access management configuration is needed to run the tool.
- Application developer's details are copied from source to target system as is. If some of the developer information isn't valid in the IDP configured in the target tenant, it must be corrected.

Network and Communication Security

The tool uses standard HTTPS communication to make API calls, as provided by the Integration Suite and API business hub enterprise(developer portal).

API Business Hub Enterprise

The API business hub enterprise is a web-based platform designed for developers to discover, explore, and utilize APIs offered by an organization.

To enable external application developers to consume APIs from different business systems, it is essential to publish these APIs. Publishing involves presenting the API proxies in a structured manner, essentially treating them as products. As an API administrator, you can accomplish this by creating a product in Integration Suite and then publishing it on the API business hub enterprise. This allows you to expose one or more API proxies to application developers. Furthermore, content administrators on the API business hub enterprise also have the ability to publish their APIs, complete with relevant documentation, to a catalog that developers can access.

Through this interface, developers can easily browse through the available APIs, access comprehensive documentation, and gain a clear understanding of how to effectively utilize them. They can also use the API testing console to make test calls to APIs and observe the corresponding responses.

In addition to its documentation and testing capabilities, the API business hub enterprise incorporates features like self-registration, providing developers with the ability to create accounts and obtain API keys for accessing protected APIs. Furthermore, it offers functionalities for application management, including analytics and usage statistics.



There are two subscription plans available for API business hub enterprise:

Subscription Plans	Plan Details	Reference Links
Developer Plan	By subscribing to the Developer Edition, you can build a catalog by creating products that consume APIs from different business systems.	
Standard/Premium Plan	On the other hand, the Standard/Premium edition provides additional features and capabilities beyond the developer edition. It is designed for organizations that want to expose their APIs to external developers and partners, enabling them to consume and integrate with the APIs. This plan includes features such as publishing API proxies as products from Integration Suite, subscribing to these products on API business hub enterprise to create applications, access control, analytics, and monetization options.	

Activating API Business Hub Enterprise

Steps to activate API business hub enterprise in SAP Integration Suite. API business hub enterprise is one of the the sub-capabilities of API Management with SAP Integration Suite

Steps	Details
Subscribe to SAP Integration Suite	<p>To set up the API Management capability from Integration Suite, you should first have an Integration Suite subscription.</p> <p>Subscribe to the SAP Integration Suite in SAP BTP cockpit and assign the Integration_Provisioner role to gain access. For more information, see Initial Setup of SAP Integration Suite.</p> <p>i Note</p> <p>Please make sure that you do not have a starter plan instance created in the same subaccount where you intend to create an Integration Suite subscription. Additionally, please note that API Management capabilities from Integration Suite and API Management subscriptions using the stand-alone tile cannot coexist in the same subaccount.</p>

Steps	Details
Activate API business hub enterprise	<p>Add and activate API business hub enterprise in Integration Suite. For more information, see Activating and Managing Capabilities.</p> <p>i Note</p> <p>The API business hub enterprise checkbox is selected by default if you've subscribed to the premium/standard service plan and have activated API Management.</p>
Access API business hub enterprise	<p>Click on the  <i>product switcher</i> icon and select API business hub enterprise.</p> <p>i Note</p> <p>If you have an SAP Build subscription, the API business hub enterprise tile appears on the home page under Quick Links section. Select the tile to navigate to the API business hub enterprise web page. Alternatively, you can click on the  <i>product switcher</i> icon on the page header and select API business hub enterprise.</p> <p>i Note</p> <p>To onboard API business hub enterprise web page, the AuthGroup.SelfService.Admin role must me assigned to you. This is an one time activity.</p> <p>Please be aware that the Authgroup.API.Admin role is required for onboarding into the API business hub enterprise. This role will be automatically assigned to your scope once you have been assigned the AuthGroup.SelfService.Admin role. After the onboarding process is completed, it is necessary for an admin to assign the Content Administrator role to a user in order to access and discover the APIs from different business systems in theAPI business hub enterprise.</p>

Accessing API business hub enterprise APIs Programmatically

The [devportal-apiaccess](#) plan allows you to access the API business hub enterprise APIs to programmatically onboard developers, create applications, and more.

About the Plan

The service key, consisting of url (application url), clientId, clientSecret, and tokenUrl is used to generate a bearer token with the help of a REST client. This bearer token, along with the application url and API endpoint, is used to trigger the APIs.

This topic explains how to enable API access for API business hub enterprise.

Prerequisites

- If you've enabled API Management capability using Integration suite, ensure that you've also enabled API business hub enterprise in Integration suite. For more information, refer [Subscribing to Integration Suite](#) and [Activating Capabilities](#).

To access API business hub enterprise from Integration Suite, select API business hub enterprise from the **Navigation Links** on the header.

i Note

Please ensure that you can access API business hub enterprise before creating an instance.

- You have the space `developer` role assigned to you.
- You have created a service instance under the **Authorization and Trust Management** tile.
 1. In your web browser, open the **SAP BTP Cockpit** - <https://cockpit.btp.cloud.sap>.
 2. From your **Subaccount**, navigate to **Spaces** in your Cloud Foundry environment and choose **Services > Service Marketplace**.
 3. Choose **Authorization and Trust Management > Instances > New Instance**.
 4. In the **Create Instance** dialog that opens, choose the **apiaccess** plan.
 5. Click **Next** until you reach the **Confirm** section.
 6. In the section **Confirm**, enter a unique **Instance Name** and choose **Finish**.
- You have created a service key for the service instance above.
 1. Choose the service instance that you created above.
 2. In the left-hand pane, navigate to **Service Keys > Create Service Key**.
 3. In the **Create Service Key** dialog that opens, provide a name.
 4. Click **Save**.

The client credentials like url, clientId, and clientSecret details appear for the given service key.

- You have created a destination of type `OAuth2Credentials` to the XSUAA APIs by using the credentials you derived from creating the service key. This is required to access the XSUAA APIs for authorization and trust management services.
 1. From your **Subaccount**, navigate to **Connectivity > Destinations > New Destination**.
 2. Choose the service instance that you created above.
 3. In the **Destination Configuration** window, provide the details.

i Note

You must enter the details exactly as mentioned below:

```
Name: apimgmt-platform-access
Type: HTTP
Description:
URL: https://yourxsuaa.authentication.sap.hana.ondemand.com (Provide the value of the ur
Proxy Type: Internet
Authentication: OAuth2ClientCredentials
Client ID: apiaccess-client_id (Provide the value of the "clientId" field from the servi
Client Secret: xxxxxxxxxxxxxxxxxxxxxxxxx (Provide the value of the "clientsecret" fiel
Token Service URL: https://yourxsuaa.authentication.sap.hana.ondemand.com (Provide the v
Token Service User:
Token Service Password:
```

- For URL, provide the value of the `url` field from the service key you created above.
- For Client ID, provide the value of the `clientId` field from the service key you created above.
- For Client Secret, provide the value of the `clientsecret` field from the service key you created above.
- For the Token Service URL, provide the value of the `url` field from the service key you created above.

4. Click **Save**.

Creating a Service Instance in the API Management, API business hub enterprise

Create a service instance using **devportal-apiaccess** plan.

1. In your web browser, open the **SAP BTP Cockpit** - <https://account.hana.ondemand.com/cockpit>.
2. From your **Subaccount**, navigate to **Spaces** in your Cloud Foundry environment and choose **Services** > **Service Marketplace**.
3. Choose **API Management, API Business Hub Enterprise** > **Instances** > **New Instance**.
4. In the **Create Instance** dialog that opens, choose **devportal-apiaccess**.
5. Click **Next**.
6. In the section **Specify parameters**, provide the details as mentioned below, based on the role you require.

The roles that support API access in the API business hub enterprise are *AuthGroup.API.Admin*, *AuthGroup.Content.Admin*, and *AuthGroup.API.ApplicationDeveloper*.

Create a service instance with the *AuthGroup.API.Admin* role to access the API business hub enterprise APIs (applications and attributes, API packages, API proxies and products, app developer and metering), and perform operations like create, update, and delete on various API business hub enterprise entities as specified in the [Business Accelerator Hub](#).

```
{
  "role": "AuthGroup.API.Admin"
}
```

Create a service instance with the *AuthGroup.Content.Admin* role to manage the domain categories in API business hub enterprise and add the related products into relevant categories.

```
{
  "role": "AuthGroup.Content.Admin"
}
```

Create a service instance with the *AuthGroup.API.ApplicationDeveloper* role to access the API business hub enterprise APIs (applications, API packages, and API proxies and products), and perform operations like create, update, and delete on various API business hub enterprise entities as specified in the [Business Accelerator Hub](#).

```
{
  "role": "AuthGroup.API.ApplicationDeveloper"
  "developerId": "developerId"
}
```

i Note

What is developerId:

Providing an invalid or an empty developerId throws an error in the service instance creation process.

To successfully create an application via the API business hub enterprise, you must provide a valid developerId. This means that you must have already registered as an application developer to the API Management, API business hub enterprise service or you must have been onboarded by your administrator.

- If you have registered to the API Management, API business hub enterprise application, provide your developerId.

See the section below to know how to obtain your developerId.

- If you have not registered to the API Management, API business hub enterprise application, follow the steps in [Register on API business hub enterprise](#) and try again.

- If you are not registered to the API Management, API business hub enterprise application, and require your admin to onboard you, contact your admin. See [Onboard an Application Developer](#).

How to obtain the developerId:

- If you are a registered developer in the API business hub enterprise, access the following URL in your browser to obtain your developerId:

```
https://devportal-url/api/1.0/user
#Response
[{"Name": "",
  "FirstName": "",
  "LastName": "",
  "LoggedOut": false,
  "Email": ""}]
```

The Name field in the response is your developerId.

- If you are an admin and are obtaining the developerId for a developer you have already onboarded, pick the userId that you provided during the developer onboarding.

To view a list of the registered developers, access the following URL in your browser. The userId field in the response is the developerId.

```
https://devportal-url/api/1.0/registrations?type=registered
#Response
autoReLogin: false
country: ""
emailId: ""
firstName: ""
lastName: ""
rolesAccess: [{status: "registered", role: "API_ApplicationDeveloper"}]
0: {status: "registered", role: "API_ApplicationDeveloper"}
userId: ""
```

Limitation: Self-service onboarding request is not supported for a developer. So, the POST operation under the [API Business Hub Enterprise - Registering Users](#) tile in the [API Business Hub](#) cannot be made by the application developer service key. As an alternative, you can invoke this API using the admin service key.

7. In the section **Confirm**, enter a unique **Instance Name**, and choose **Finish**.

The service instance is successfully created and listed in the **Instances** window.

Create a Service Key

Generate a service key for the service instance that you created above:

1. From the **Instances** window, choose the service instance that you created above.
2. In the left-hand pane, navigate to **Service Keys** > **Create Service Key**.
3. In the **Create Service Key** dialog that opens, provide a name.
4. In the text box enter one of the following payloads as per your requirement:

To create service key of credential type...	Use payload...	Level of Security	Important Notes	Sample of generated creden
"instance-secret" (without payload)		Low	For instance-secret, the clientSecret generated is same for all the keys.	For admin role: { "url": "https "tokenUrl": " "clientId": " "clientSecret } For developer role: { "url": "https "tokenUrl": " "developerId" "clientId": " "clientSecret }
"instance-secret" (with payload)	{ "xsuaa": { "credential-type": "instance-secret" } }	Low	For instance-secret, the clientSecret generated is same for all the keys.	For admin role: { "url": "https "tokenUrl": " "clientId": " "clientSecret } For developer role: { "url": "https "tokenUrl": " "developerId" "clientId": " "clientSecret }

To create service key of credential type...	Use payload...	Level of Security	Important Notes	Sample of generated creden
"binding-secret"	<pre>{ "xsuaa": { "credential-type": "binding-secret" } }</pre>	Medium	For binding-secret, the clientSecret generated for every key is unique.	<p>For admin role:</p> <pre>{ "url": "https "tokenUrl": " "clientId": " "clientSecret" }</pre> <p>For developer role:</p> <pre>{ "url": "https "tokenUrl": " "developerId" "clientId": " "clientSecret" }</pre>
"x509" (certificate based)	<pre>{ "xsuaa": { "credential-type": "x509", "x509": { "key-length": 2048, "validity": 65, "validity-type": "DAYS" } } }</pre>	High	For X509, ensure that the credential rotation is done based on the validity provided in the payload. For example, delete and create a new service key every 65 days.	<p>For admin role:</p> <pre>{ "url": "https://x "certificate": "x "certurl": "https "clientId": "xxxx "privateKey": "xx "tokenUrl": "http }</pre> <p>For developer role:</p> <pre>{ "url": "https://x "certificate": "x "certurl": "https "clientId": "xxxx "developerId": "x "privateKey": "xx "tokenUrl": "http }</pre>

5. Click [Save](#).

The credentials like url, tokenUrl, developerId (for developer role), clientId, and clientSecret details are displayed for the given service key.

- The application url is used to make API calls.
- The clientId and clientSecret are necessary credentials required to fetch the Bearer Token.
- The tokenUrl is used to fetch the Bearer Token.

Make a note of these credentials as you will need them in the next steps to obtain a bearer token, in order to access the API business hub enterprise APIs.

i Note

Once your client is setup you can use it to authenticate against the x509 endpoint by providing the client certificate and key.

Create the certificate.cer and certificate.key files based on the public and private keys obtained from the x509 credentials respectively.

For certificate.cer file:

1. Copy the "certificate" value starting from -----BEGIN CERTIFICATE----- all the way to -----END CERTIFICATE-----\n (the certificate value might contain multiple certificates). Make sure you copy all the certificates.
2. Paste it in a text editor. Find and replace all the occurrences of \n by \n.
3. Save the file as certificate.cer.

For certificate.key file:

1. Copy the "certificate" value starting from -----BEGIN RSA PRIVATE KEY-----\n... all the way to-----END RSA PRIVATE KEY-----\n
2. Paste it in a text editor. Find and replace all the occurrences of \n by \n.
3. Save the file as certificate.key.

Open a command prompt/terminal in the folder where you have saved the certificate files and execute the following curl command to get the response in the my-oauth-response.json file in the same folder. From this file, you can fetch the bearer token from the value of "access_token".

```
curl --cert certificate.cer --key certificate.key --location --request POST '<certurl from the sei
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=<clientId from the servicekey x509 credentials>' \
--cert certificate.cer \
--key certificate.key \
> my-oauth-response.json
```

Updating a Service Instance in the API Management, API business hub enterprise

You can update an already provisioned service instance of an API access plan by performing the following steps:

Prerequisite:

You must have the Cloud Foundry CLI installed.

1. Log in to the Cloud Foundry CLI by running the *cf login* command.
2. Select [Org](#).
3. Run the following command to update your service instance. *cf update-service <service-instance-name> -c <empty-json-file>.json*.

☰ **Sample Code**

Sample json: {}

Next Steps

Obtaining a Bearer Token

In the REST client:

1. Paste the copied **tokenUrl**. Append `?grant_type=client_credentials` to the **tokenUrl**.
2. Choose *Basic Auth* as the Authorization type.
3. Similarly, paste the **clientId** and **clientSecret** in the place of Username and Password.
4. Make a POST Call.
5. Obtain the Bearer Token from the output and copy it in a notepad.
 - Now, to trigger an API, in the same REST client, append the API endpoint (obtained from the API business hub enterprise APIs that are located in the SAP API Management package of API Business Hub) to the **url**.
 - Choose *Bearer Token* as the Authorization type and paste the copied Bearer Token in the specified space.
 - Include payloads, if needed.
 - Make an API call.

Centralized API business hub enterprise

The API business hub enterprise is a central API catalog, allowing application developers to consume APIs and other assets, from a common platform.

If you have enabled API business hub enterprise and API Management tenant in the same Integration Suite sub-account, they will automatically connect to each other.

i Note

Once this connection is established, you will not be able to connect the API Management tenant to any other API business hub enterprise enabled in a different sub-account. However, if you have not enabled API business hub enterprise in the same Integration Suite sub-account where you have enabled API Management tenant, you can connect this API Management tenant to an API business hub enterprise enabled in another sub-account and designate it as a centralized API business hub enterprise.

This centralized API business hub enterprise can be used to establish connections with multiple API Management tenants and can receive API proxies, API products, and other assets from each connected API Management tenants. It is important to ensure that all assets published to the centralized API business hub enterprise are unique.

→ Remember

You can configure multiple Integration Suite API Management tenants to cater to different stages of the API lifecycle. For example, you can have separate instances for development, testing, and production. However, connecting these API Management tenants having such a relationship to the same API business hub enterprise will violate the uniqueness of the assets.

Once the application developers register with the centralized API business hub enterprise, they can easily search, explore, and test APIs. They can also create and subscribe to specific types of applications available from the API business hub enterprise.

The API business hub enterprise admin identifies which existing or new API business hub enterprise application can accept content from multiple Integration Suite API Management tenants.

Create a Connection Request for the Centralized API business hub enterprise

Create a request to connect the Integration Suite API Management tenant to the API business hub enterprise. You need to establish this connection to publish the content of the Integration Suite API Management tenant on the API business hub enterprise.

Prerequisites

- To establish connections between the API business hub enterprise and Integration Suite API Management tenants, a Cloud Foundry space should be created in the sub-account from where the API business hub enterprise is hosted.
- To establish a connection between an Integration Suite API Management tenant and the centralised API business hub enterprise which is available in a different sub-account, you must ensure that the API business hub enterprise capability is not enabled in the same sub-account as that of the API portal.
- The following role collections should be assigned to you:
 - AuthGroup.API.Admin
 - APIPortal.Administrator: To generate the access credentials from the Integration Suite API Management tenant, you must have the [APIPortal. Administrator](#) role assigned to you.
 - AuthGroup.APIPortalRegistration: You need this role to create a connection request and update the connection request credentials.
 - APIPortal Service.CatalogIntegration: You need to have this role assigned to you as the client credentials is generated for this role.
- Generate the access credentials to establish the connection.
 1. Log in to the Integration Suite.
 2. Choose the navigation icon on the left and choose **Settings > APIs**.
 3. Choose the **Connection** tab.
 4. Follow the onscreen instructions under **Connect the API Portal to the centralized API Business Hub Enterprise** to generate the Integration Suite API Management tenant access credentials.

i Note

The client credentials get generated for the [APIPortal .Service.CatalogIntegration](#) role.

Context

The API business hub enterprise administrator identifies which existing or new API business hub enterprise application can accept content from multiple Integration Suite API Management tenants.

i Note

Only new Integration Suite subscriptions with API Management capability enabled with the Integration Suite are allowed to set up a connection with the centralized API business hub enterprise.

i Note

You can connect a maximum number of three Integration Suite API portals to the centralized API business hub enterprise.

Create a new subaccount in Cloud Foundry and set up only the Integration Suite API Management tenant.

For the newly set up Integration Suite API Management tenant, you can request for the API business hub enterprise connection to be established.

i Note

The option to disconnect an Integration Suite API Management tenant from an existing API business hub enterprise isn't supported currently.

i Note

Once this connection is set up, you can't place a request to severe this connection and establish a new connection with any other centralized API business hub enterprise.

To create a request to connect the Integration Suite API Management tenant to the centralized API business hub enterprise.

Procedure

- 1. Log on to the [API Business Hub Enterprise](#).
- 2. Navigate to the [Enterprise Manager](#) [Manage Connections](#) and choose [Approved Requests](#).
- 3. Choose [Add New Connection](#).
- 4. Fill in the following details on the [Submit Connection Request](#) page.

Parameters	Values
API Portal Alias Name	Enter the Integration Suite API Management tenant name that gets displayed on the API Business Hub Enterprise. This name is used to distinguish products that are published from the API portal and likewise for applications created for the product.
API Portal Access Credentials	<div>Enter the Integration Suite API Management tenant access credentials that you generated earlier. These credentials are used by the API business hub enterprise to establish the connection.</div> <div>Sample credentials:</div> <div><pre>{ "url": "https://<application name>.cfapps.sap.hana.ondemand.com", "tokenurl": "https://<name>.authentication.sap.hana.ondemand.com/oauth/token", "certurl": "https://xxxxxx.authentication.cert.sap.hana.ondemand.com", "certificate": "xxxxxxxxxxxxxxxxxxxxxx", "key": "xxxxxxxxxxxxxxxxxxxxxx" }</pre></div> <div>i Note</div> <div>These credentials will remain valid for a period of 65 days. Please make sure to regenerate them and reestablish the connection within this timeframe.</div>
Comment	Provide the details to the approver about the need for the connection request.

Parameters	Values
Once this connection is set up, you can't place a request to sever this connection and establish a new connection with any other centralized API business hub enterprise.	Select the checkbox to confirm.

5. Choose **Submit**.

Results

You've submitted the connection request to the API business hub enterprise administrator. Once the connection request is approved by the administrator, you can start publishing the Integration Suite API Management tenant content to the API business hub enterprise.

i Note

You can log on to the Integration Suite API Management tenant and check the connection status. Navigate to **Settings > APIs** and choose **Connection**.

You can also choose **Test Connection** to get the details about the connectivity status once your connection request is approved. You will get a connection error, if the destination is deleted or configured incorrectly. In case of an error, retry after revalidating the destination configuration.

Updating the Connection Request Credentials for a Pending Request

Update the credentials you've used to establish a connection between the Integration Suite API Management tenant and the API business hub enterprise.

Prerequisites

- Only users who have submitted a connection request and have the AuthGroup.APIPortalRegistration role assigned to them can edit the credentials.
- To update the Integration Suite API Management tenant access credentials, you must first generate it. To generate the credentials from the Integration Suite API Management tenant, you must have the **APIPortal. Administrator** role assigned to you.
 - Log in to the Integration Suite.
 - Choose the navigation icon on the left and choose **Settings > APIs**.
 - Choose the **Connection** tab.
 - Choose **Generate Credentials** under **Connect the API Portal to the centralized API Business Hub Enterprise** and **Copy** the access credentials.

i Note

The client credentials get generated for the [APIPortal .Service.CatalogIntegration](#) role.

Context

The credentials required to access the Integration Suite API Management tenant are shared during the connection request process.

If you encounter one of the following situations when your connection request is in the pending request state, you have to update the credentials:

- You have submitted incorrect credentials while raising a connection request, and your request is in pending approval state.
- You've deleted the service instance, or the service key, after the connection request was submitted. In this case, the credentials you used before deleting the service instance or the service key becomes invalid.

Procedure

1. Log on to the [API Business Hub Enterprise](#).
2. Navigate to the [Enterprise Manager](#) [Manage Connections](#) and choose [Pending Requests](#).
3. Go to the [Actions](#) column of the connection request that you want to edit and choose [Edit Credentials](#).
4. On the [Edit Credentials for <API Portal Alias Name >](#) popup, enter the mandatory [*API Portal Access Credentials](#) that you copied earlier from the Integration Suite API Management tenant.

Sample credentials:

```
{
  "url": "https://<application name>.cfapps.sap.hana.ondemand.com",
  "tokenurl": "https://<name>.authentication.sap.hana.ondemand.com/oauth/token",
  "certurl": "https://xxxxxx.authentication.cert.sap.hana.ondemand.com",
  "certificate": "xxxxxxxxxxxxxxxxxxxxxx",
  "key": "xxxxxxxxxxxxxxxxxx"
}
```

i Note

The credentials required to establish the connection will be valid for 365 days. Please remember to regenerate them and reestablish the connection within this timeframe. However, any credentials generated prior to February 2024 with a validity of 65 days will remain valid for that specific duration. The 365-day timeframe will apply to all newly generated credentials.

5. Choose [Save](#).

Results

You've successfully updated the credentials of the connection request that is pending.

Approve the Pending Connection Requests

As an API business hub enterprise administrator, you must approve or reject the connection request after you receive them.

Prerequisites

The following roles must be assigned to you:

- AuthGroup.API.Admin
- AuthGroup.APIPortalRegistration

Procedure

1. Log on to the [API Business Hub Enterprise](#).
2. Navigate to the [Enterprise Manager > API Management Connections](#) and choose [Pending Requests](#).
The connection requests that are pending for approval are listed on the [Pending Requests](#) page.
3. Choose [View](#) to read the comments from the requester before approving or rejecting a connection request.
4. Select the connection that you want to approve, choose the [Actions](#) icon and select [Approve](#).

Results

The connection has been set up between the Integration Suite API Management tenant and the API business hub enterprise.

Updating the Connection Request Credentials for an Approved Request

There can be instances where you have to update the credentials once the connection request is approved by the API business hub enterprise admin.

Prerequisites

To update the API portal access credentials, you must first generate it. To generate the credentials from the Integration Suite API Management tenant , you must have the APIPortal.Administrator role assigned to you.

1. Log in to the Integration Suite.
2. Choose the navigation icon on the left and choose [Settings > APIs](#).
3. Choose the [Connection](#) tab.
4. Choose [Regenerate Credentials](#) and [Copy](#) the access credentials.

i Note

The client credentials get generated for [APIPortal .Service.CatalogIntegration](#) role.

Context

To establish the connection between the Integration Suite API Management tenant and the API business hub enterprise, the client Id and client secret created for the Integration Suite API Management tenant is shared during the connection request process.

If you encounter one of the following situations after the connection request has already been approved by the API business hub enterprise admin, you have to update the credentials:

- The service instance, or the service key gets deleted after the connection between the Integration Suite API Management tenant and the API business hub enterprise was established. In this case, the credentials you were using before the service instance or the service key got deleted becomes invalid.
- Similarly, if the destination that fetches the API content from the Integration Suite API Management tenant workspace gets deleted, the credentials you were using before the destination got deleted becomes invalid.

Procedure

1. Log on to the [API Business Hub Enterprise](#).
2. Navigate to the **Enterprise Manager** > **API Management Connections** and choose **Approved Requests**.
The connection requests that are pending for approval are listed on the **Approved Requests** page.
3. Go to the **Actions** column and select the approved connection request that you want to edit and choose **Re-establish Connection**.
4. On the **Re-establish Connection** page, enter the access credentials that you copied earlier from the Integration Suite API Management tenant in the ***API Portal Access Credentials** : text box.

Sample credentials:

```
{
  "url": "https://<application name>.cfapps.sap.hana.ondemand.com",
  "tokenurl": "https://<name>.authentication.sap.hana.ondemand.com/oauth/token",
  "certurl": "https://xxxxxx.authentication.cert.sap.hana.ondemand.com",
  "certificate": "xxxxxxxxxxxxxxxxxxxxxx",
  "key": "xxxxxxxxxxxxxxxxxx"
}
```

i Note

The credentials required to establish the connection will be valid for 365 days. Please remember to regenerate them and reestablish the connection within this timeframe. However, any credentials generated prior to February 2024 with a validity of 65 days will remain valid for that specific duration. The 365-day timeframe will apply to all newly generated credentials.

5. Choose **Submit**.

Results

You've updated the Integration Suite API Management tenant access credentials successfully.

Consume API Proxies

Consume API proxies via the API business hub enterprise. In the API business hub enterprise, an application developer registers, explores the API exposed by customers, creates applications, and tests API proxies.

⚠ Caution

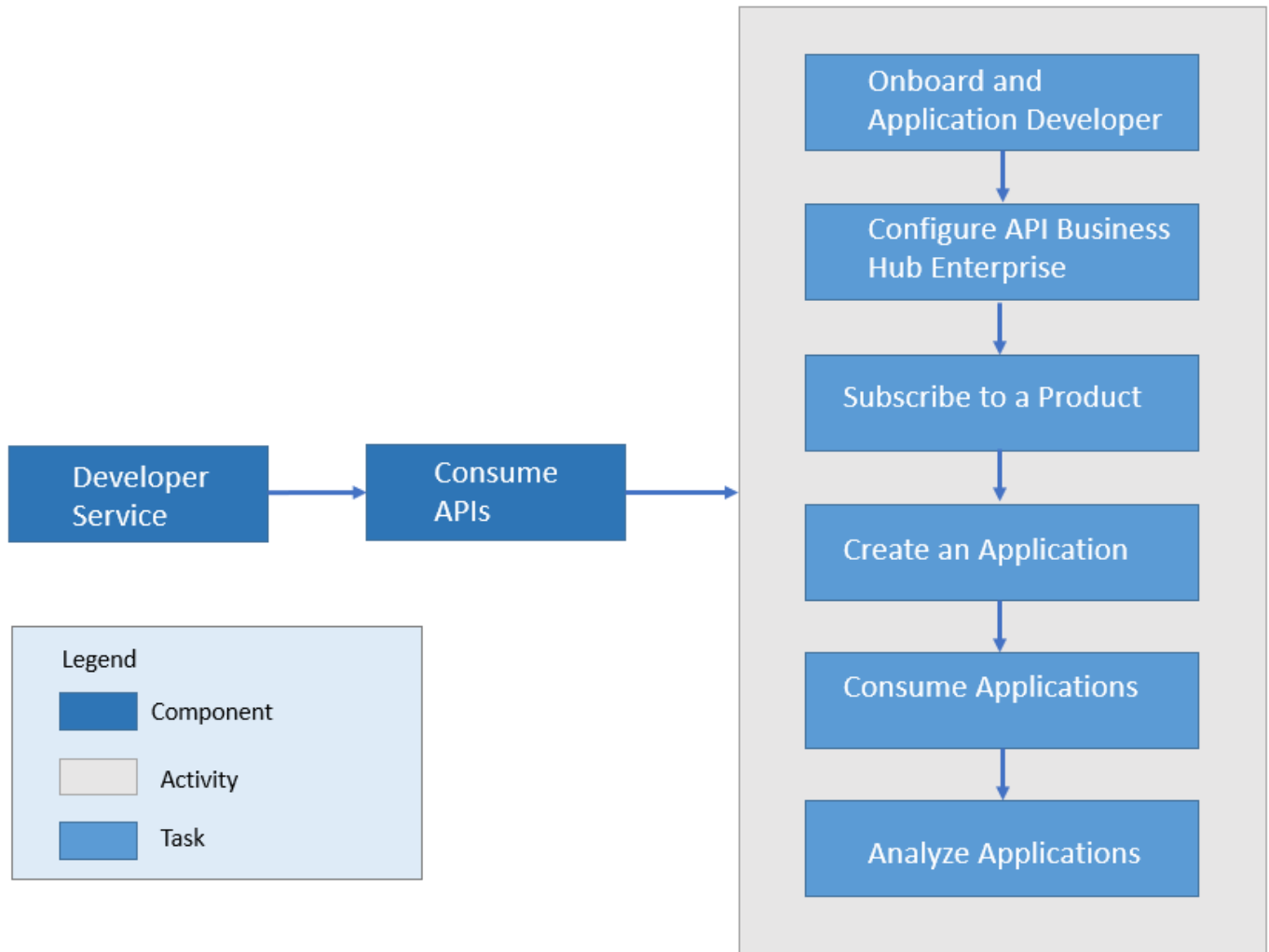
Effective June 2024, the classic design of the API business hub enterprise will be deprecated and will no longer be accessible. The new design of the API business hub enterprise will be set as your default design from March 2024. For more information, see [Configure API business hub enterprise](#).

If you've added API business hub enterprise as a capability with Integration suite, or if you've subscribed to the API business hub enterprise as part of the standalone API Management subscription, you have the option to experience the new design of the API. This is custom documentation. For more information, please visit the [SAP Help Portal](#)

business hub enterprise user interface along with the classic design.

i Note

By default, the Site Administrator has an option to switch from classic to new design and set the new design as the default UI using the **Site Editor**. The Site Administrator has the right to enable the configuration to let all the other users switch between the old and the new design. For more information, see [Customize the Visual Format of the API business hub enterprise](#).



API business hub enterprise is an application that provides a common platform for Application developers to consume API proxies. Every API Management customer is provided with their own API business hub enterprise application on cloud. The API business hub enterprise offers capabilities to onboard application developers, explore and test API proxies, create and subscribe to Applications.

The API business hub enterprise supports the following features:

- **Onboard an Application developer**- To explore the API proxies and subscribe to an Application, an Application developer must be registered to the API business hub enterprise. On registering, the Application developer is provided access to the API business hub enterprise.
- **Browse Catalog**- Explore the Products (assembled APIs) available in the Catalog store, navigate to individual API proxies, read the API Documentation, and view the resources attached to the API proxies.

i Note

A limitation within the open-source Swagger library, on which the API business hub enterprise relies, causes slow, improper, or no rendering of API schemas that contain circular references on deeply nested models on the platform.

- **Create Applications** – An Application developer can create one or more applications to consume API proxies. To consume the API proxies, an Application developer must subscribe to an Application (assembled Products). It is by subscribing to an Application that you return to the developer the key required to access the API proxies.
- **Download JSON**- You can download the open API specification for the APIs that are part of the API business hub enterprise in JSON format. This enables the developer to use the metadata of the APIs for various aspects such as code/SDK generation for developing applications.
- **Download SDK**- You can also download the client software development kit (SDK) for developers through a non-commercial license on open source sites. You can use this SDK for developing applications.
- **Test API Proxies** - You can test the API proxies and understand the runtime behavior of the API proxies better. Use the Test Console to explore the resources associated with an API and execute the operations.

Related Information

[Configure API business hub enterprise](#)

Onboard an Application Developer

Explains how API administrators can onboard application developers so they can access the API business hub enterprise.

Context

A user must be onboarded to API business hub enterprise only via Self-registration or **Add User** flow.

To provide application developers with access to the API business hub enterprise, the API Administrator first has to onboard them. The steps to onboard an application developer are as follows:

Procedure

1. The application developers log on to the API business hub enterprise application with their IDP user credentials, and register to the API business hub enterprise. For more information, see [Register on API business hub enterprise](#).
2. The API administrator approves or rejects the request to access the API business hub enterprise. For more information, see [Managing the Access Request of the Users \[New Design\]](#).

If you haven't enabled the automatic creation of shadow users, and you've not explicitly created shadow users for your developers, then they're unable to log on to the application, and they're asked to contact the administrator. For more information, see [Shadow Users](#)

Register on API business hub enterprise

Procedure to register as an application developer on the API business hub enterprise to view the products available in the catalog store. The API business hub enterprise also enables you to explore the APIs, read the associated API documentation, and view resources.

Prerequisites

- As a developer you're trying to self-register:
 - You're already a valid Application IDP user.
 - The admin has already added your email ID in the subaccount.

i Note

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

If the [AuthGroup.API.ApplicationDeveloper](#) role is already assigned to you by the SAP BTP admin or via the IDP Role Collection mapping, you will get automatically registered as an application developer in API business hub enterprise when you logon for the first time.

If you don't have the [AuthGroup.API.ApplicationDeveloper](#) role assigned to you in SAP BTP cockpit, complete the self-registration process to access all the functionalities and features of API business hub enterprise.

Please note that the [AuthGroup.API.ApplicationDeveloper](#) role that has been assigned to you will only take effect if you login to API business hub enterprise. Only relevant for New Design of the API business hub enterprise.

- As an Admin you're trying to onboard multiple users:
 - The admin has already added the email IDs of the users in the subaccount.
 - The admin has assigned the [AuthGroup.API.Admin](#) role to all the users.

i Note

While onboarding multiple users, it is recommended that you don't assign the [AuthGroup.API.Admin](#) role to all the users as this will enable the developers to take on the admin role. Instead you can automate the process of onboarding multiple users by using the API "[API Business Hub Enterprise - Registering Users\(CF\)](#)".

In this case, admin approval is not required. When the user logs in and chooses the [Register](#) button, they get auto registered as developers.

i Note

Consider the following behavior for auto-registration:

- **Use Case 1: User is no longer in the organization:**

If the [AuthGroup.API.ApplicationDeveloper](#) role is removed from either the SAP BTP cockpit or the IDP Role Collection mapping, as an admin, you should also ensure that the [AuthGroup.API.ApplicationDeveloper](#) role is removed from the API business hub enterprise, or vice versa. Failing to do so may lead to confusion and discrepancies.

- **Use Case 2: User is still in the organization:**

If a user is still part of the organization, the role removal in BTP will take effect in the API business hub enterprise once the user logs in. If this user wishes to access the API business hub enterprise, they must either follow the self-registration process or have this role assigned to them from the SAP BTP Cockpit.

Context

The procedure below describes the sequence of steps when as a developer you're trying to self-register:

Procedure

1. Log on to the API business hub enterprise application with your IDP user credentials.
2. To register to the API business hub enterprise as an Application developer, choose [Register](#).

A dialog box with the prepopulated data such as, your first name, last name, and e-mail address appears.

3. Enter the country/region and reason for requesting access to the API business hub enterprise.
4. Choose [OK](#).

The request is sent to the administrator with the [AuthGroup.API.Admin](#) role.

- If the administrator approves your request, you'll receive an e-mail notification. You can log in to the API business hub enterprise via the link provided in the e-mail.
- If the administrator rejects the request, you'll receive an e-mail notification with the reason for the rejection. When you log on to the application, you'll see the reason for request rejection on the display page.

i Note

Application Developers can now email to the administrator by replying to the email notification they receive for any queries regarding their access request to the API business hub enterprise application.

Managing the Access Request of the Users [New Design]

As an API administrator, you can approve or reject the access request made by an application developer to use the API business hub enterprise.

Prerequisites

You're assigned the **AuthGroup.API.Admin** role.

i Note

This document describes the new design of the API business hub enterprise. To view the documentation for the classic design, see [Managing the Access Request of the Users \[Classic Design\]](#).

Procedure

1. Log on to the API business hub enterprise.

Use the **Manage Users** page to approve or reject the developer's registration requests and manage the roles of the registered users. For assigning roles to the users, use the SAP BTP Cockpit.

2. Choose **Enterprise Manager > Manage Users > E-mail Configuration** and add the administrator's email in the **E-mail Configuration** textbox.

The administrator receives email notification of the pending developer registration requests on this email id. Also, the e-mail notifications to the developers or users are sent from this e-mail id.

i Note

Only one administrator's email address can be entered in the **E-mail Configuration** textbox.

3. To view the pending requests, navigate to **Manage Users > New Requests**.
4. Look for the request and choose **Accept Request** from the **Actions** column. The application developer can now access the API business hub enterprise.

If you don't wish to provide access to the user, choose **Decline Request** from the **Actions** column.

On accepting the request, an approval email is sent to the requester. On rejecting a request, you need to provide a reason for rejection; an email notification is sent to the requester.

You can also view the registered users by choosing **Manage Connections > Registered Users**.

On the **Registered Users** page, you can:

- Register a new user by choosing **Add User**.

i Note

A user must be onboarded to API business hub enterprise only via Self-registration or **Add User** flow.

In the **Add User** dialog:

- a. Enter the **User ID** and the user details like **First Name**, **Last Name** and **Email ID**.
- b. Under **Assigned Roles**, identify the needed scope and select the roles accordingly:

Roles	Description
Administrator	Manages user registration. Create and delete applications on behalf of application developers. In addition, create custom attributes and import the app key.
Developer	Create applications and check billing and metering data. Test APIs and view analytics data.
Site Administrator	Configure updates, and perform portal changes like uploading the logo, changing the name and the description, and changing the footer links for the site.
Content Administrator	Manages content categories.

- o Edit an existing user to add or remove the **Assigned Roles**.

Revoke Access [New Design]

Revoke the access of an application developer.

Prerequisites

You are an API administrator and the role **AuthGroup.API.Admin** is assigned to your user.


Context

As an API administrator, you use this procedure to revoke an application developer's access for using the API business hub enterprise.

i Note

This document describes the new design of the API business hub enterprise. To view the documentation for the classic design, see [Revoke Access \[Classic Design\]](#).

Procedure

1. Log on to the API business hub enterprise.
2. Choose **Enterprise Manager > Manage Users > Registered Users**.
3. From the list of application developers, select the application developer whose access you want to revoke and choose the  *Revoke User* icon under the **Actions** column.
4. In the **Revoke** window, provide a reason for revoking the access.

i **Note**

By revoking roles, users lose all the roles assigned to them. However, user account will be retained.

Results

You have revoked the access of the user from using API business hub enterprise successfully.

Delete Data of Unregistered Users

SAP API Management stores the data of users who have logged on to the developer portal but have not registered. This topic describes the service used to delete the data of such users.

Prerequisites

You are assigned the AuthGroup.API.Admin role.

Context

Procedure

Run the following service using the standard REST console:

- Service URL: `https://<Dev-Portal-URL>/api/1.0/offboarding/{userId}`
- Method: POST
- Request Header: x-csrf-token: fetch
- Content Type: application/json
- Response: 201

The user data is deleted.

Configure API business hub enterprise

You can configure and customize the API business hub enterprise to suit your organization's needs.

Key Actions for Admins in API business hub enterprise

As...	You can use the...	For more information, see...
An API Admin , you already have the AuthGroup.API.Admin role assigned to you.	Home Page to browse and search through the various categories, APIs, and products available.	Register on API business hub enterprise

As...	You can use the...	For more information, see...
	<p>My Workspace to perform the following actions on behalf of an application developer:</p> <ul style="list-style-type: none"> • Create, update, and delete applications • Create custom attributes for applications • Provide app key and secret, while creating or updating an application • View and access all the applications created in API business hub enterprise • Monitor costs • Analyze reports 	Creating an Application with API business hub enterprise Administrator Role
	<p>► Enterprise Manager ► Manage External Content ► to adjust the visibility of the Graph navigator on the API business hub enterprise.</p>	Manage External Content
	<p>► Enterprise Manager ► Manage Access ► to control the level of access for your users, allowing them to search, discover, and access the content available on the API business hub enterprise.</p>	Manage Developer Access
	<p>► Enterprise Manager ► Manage Users ► to add and revoke user access to the API business hub enterprise.</p>	Managing the Access Request of the Users [New Design] Revoke Access [New Design]
	<p>► Enterprise Manager ► Manage API Management Connections ► to approve and reject the pending connection requests and update the API portal access credentials.</p> <p>i Note Additionally, the AuthGroup.APIPortalRegistration role must be assigned to you to perform the above actions.</p>	Approve the Pending Connection Requests
A Site Admin you already have the AuthGroup.Site.Admin role assigned to you.	<p>Site Editor to customize the visual layout of the API business hub enterprise.</p>	Customize the Visual Format of the API business hub enterprise
	<p>► Enterprise Manager ► Manage Notifications ► to configure notifications to keep the end users of the API business hub enterprise informed about website updates and news items.</p>	Manage Notifications

As...	You can use the...	For more information, see...
A Content Admin you already have the AuthGroup.Content.Admin role assigned to you.	Enterprise Manager > Manage Domain Categories to create domain categories and add the related products into relevant categories. i Note Additinally, the AuthGroup.API.Admin role must be assigned to you to perform the above actions.	Manage Domain Categories

Key Actions for Application Developers inAPI business hub enterprise

As...	You can use...	For more information, see...
An Application Developer you already have the AuthGroup.API.ApplicationDeveloper role assigned to you. i Note The AuthGroup.API.ApplicationDeveloper role is assigned by default to a user who onboards to the API business hub enterprise using the Self-registration process or via Add User flow.	My Workspace to create applications, view your applications, monitor costs, and analyze reports.	Creating an Application with Application Developer Role
	Test Environment to test the runtime behaviour of APIs.	Test Runtime Behavior of APIs

Customize the Visual Format of the API business hub enterprise

As a Site Administrator, you can customize the visual layout of the API business hub enterprise using the Site Editor. The customizations you make using the Site Editor appear to the other users in the system.

Prerequisites

You already have the **Site Administrator** role assigned to you.

Site Editor

You can use the **Site Editor** to:

- Edit site name, logo, and description
- Modify header design
- Set default design for the site
- Change banner image
- Customize background settings
- Customize footer

Edit Site Name and Logo

To modify the site name and logo,

1. Choose the [Edit Site Name and Logo](#) tab.
2. Enter the name in the [Site Name](#) field.
3. Upload or drag and drop an image for the logo in the [Site Logo](#) field.

Edit Header Design

To customize the header design,

1. Choose the [Edit Header Design](#) tab.
2. In the [Edit Header Design](#) popup, choose colors for the header bar, text, and the selection bar.

Set Default Design

To set a default design,

1. Choose the [Set Default Design](#) tab.
2. Select the [Set current design as default design](#) checkbox.
3. To allow users to toggle between the old and the new interface, select the [Allow users to toggle between old design and new design](#) checkbox.

i Note

Once the new design is activated, the changes are permanent, and you can't revert to the classic design.

Change Banner Image


To change the banner image,

1. Choose the [Change Banner Image](#) tab.
2. You can upload the banner image in the [Upload Image](#) field or select from the predefined library available.

Edit Description

Choose the [Edit Description](#) tab to modify the [Title](#) and [Subtitle](#) of the home page.

i Note

As a part of banner description, you can use the markdown language to add links and email addresses to the [Title](#) and [Subtitle](#) fields within the [Edit Description](#) dialog. For example, [SAP](www.sap.com ) and <mailto:john.doe@example.com>.

You can also add multiple lines in the sub-titles using markdown. If you want to move to a new line, end the previous line with a backslash (\).

Edit Banner Settings

To modify the banner settings,

1. Choose the [Edit Banner Settings](#) tab.
2. You can adjust the height of the banner image by using the [Minimum Height of the Background Image](#) field.
3. Colors for the text and search box can be selected using the [Text Color](#) and [Search Box Color](#) fields.

Customize Footer

1. To add reference links to the footer, choose the [Edit Footer](#) tab.
 - Choose [Add a Link](#) to create a new reference link.
 - To bundle relevant links into a group, choose [Group links](#).
2. Choose the [Edit Footer Design](#) tab to customize the footer bar color, text color, and hover link text color.

Publish Changes

The customizations you've made to the API business hub enterprise layout will be available to the users once you publish the changes.

Manage Domain Categories

Domain categories are displayed on the API business hub enterprise home page.

Prerequisites

You need the following roles to create and update categories:

- [AuthGroup.Content.Admin](#)

To assign the role, see [Managing the Access Request of the Users \[New Design\]](#).

- [AuthGroup.API.Admin](#)

To assign the role, see [Assign User Roles in API Management](#) .

Context

Content administrators can use [Manage Content](#) to create domain categories and add the related products into relevant categories. They can also configure the order in which these categories and the contained products get displayed in the home page.

i Note

If you've configured the API business hub enterprise to connect to multiple API portals then you can add products from different API portals under one category. Whereas in classic design, you can only add products from one API portal under a category.

Use the following procedure to configure navigation categories.

Procedure

- 1. Log on to the API business hub enterprise.
- 2. Choose ►Enterprise Manager ► Manage Domain Categories ► from the top navigation bar.
- 3. To add a category, choose Add New Domain Category.
- 4. Enter the following details in the Add Domain Category dialog and choose Save:

Name	Description
Category Name	Provide a name for the category.
Category Title	Provide a title for the category. Categories are identified by their title on the home screen.
Description	Provide a description for the category.

- 5. To add products, choose □ Add Products from the top-right corner of the newly created category pane. In the Add Products dialog, select the products that you want to add to this category and choose Save.
- 6. Save the changes.

Newly configured category is visible on the Manage Content page. For individual categories, you can perform the following:

- Reorder the categories by using the □ Move Up and □ Move Down action icons.
- Edit a category by choosing the □ edit icon.
- Delete a category by choosing the □ icon.

Manage Notifications

As a site administrator you can configure notifications for providing information to the API business hub enterprise end users on any website updates, events or news items.

Prerequisites

You're assigned the AuthGroup.Site.Admin role. To assign the role, see Managing the Access Request of the Users [New Design].

Procedure


- 1. Log on to the API business hub enterprise.
- 2. Choose ►Enterprise Manager ► Notifications ► from the top navigation bar.
- 3. Choose Add Notification.
- 4. Provide the following details on the Add Notification dialog:

Topic Name	Enter a name for the notification entity. Example: Experience the new design of the API business hub enterprise!
Description	Enter a description for the notification entity. Example: If you've subscribed to API business hub enterprise as part of Integration Suite subscription , we now have a new design of the user interface for you to experience.

- If you want to provide more information through a link, for example, a link to a blog post, or a help document, choose [Add Link](#) and enter the link text and the URL.

i Note

The **Topic Name** and the **Display Name** (link text) support a maximum of 250 and 512 characters, respectively. The **Description** and the **URL** fields support 2048 characters each.

- Choose [Add](#) to publish the notification.
- After adding the notifications, enable the notification feature using the toggle switch on the [Manage Notifications](#) page. The notifications will be visible to the users under the  *notification bell* icon in the header only if the site administrator has explicitly enabled the notification.

Results

The notification appears on the [Manage Notifications](#) page.

i Note

You can also, edit notifications, change the order of the notifications, and delete the notifications per your requirement.

Manage External Content

On this page, you have the option to adjust the visibility of the Graph navigator on the API business hub enterprise.

Prerequisites

- Assign the [AuthGroup.API.Admin](#) role.

To assign the role, see [Assign User Roles in API Management](#) .



- Enable **Graph** in Integration Suite. For more information, see [Activating and Managing Capabilities](#) and [Configuring User Access](#).

Context

If the graph feature is activated in the Integration Suite, the **Configure Graph** setting will be enabled by default in [Manage External Content](#), and the **Graph** option will be displayed in the header. If you want to disable it, you can do so from this page.

If the graph feature is deactivated in the Integration Suite, the **Configure Graph** setting in [Manage External Content](#) will be disabled.

Procedure

- Log on to the API business hub enterprise.
- Choose  **Enterprise Manager**  from the top navigation bar.
- Use the slider button to enable/disable the graph feature.

Choose **Yes/No** in the confirmation dialog.

Manage Developer Access

As an API business hub enterprise admin, you have the authority to control the level of access for your users, allowing them to search, discover, and access the content available on the API business hub enterprise.

Prerequisites

You need the following role to configure the access control checks:

- [AuthGroup.API.Admin](#)

To assign this role, see [Assigning Role Collections to Users](#).

i Note

The **Manage Access** feature is available only in the new design of the API business hub enterprise on the Cloud Foundry environment.

Context

In API business hub enterprise, managing access for different users is important for several reasons like improving user productivity, resource optimization, privacy, and security.

User productivity is enhanced by granting users the appropriate access to carry out their tasks efficiently, without being overwhelmed by unnecessary information or resources. In this context, the **All Visitors** option allows anyone, whether logged in or not, to utilize the APIs without requiring authentication. However, the ability to consume the APIs still depends on obtaining the necessary developer role.

Moreover, by managing access, you can provide access to **Authenticated Users** who do not have a designated role, allowing them to access different pages of the API business hub enterprise based on their specific needs. This facilitates broader exploration and enables users to familiarize themselves with the available resources. Nevertheless, the ability to consume the APIs still relies on obtaining the necessary developer role.

To maintain privacy and security, you can grant access to **Authorized Users** who are logged in and possess the required developer role. This ensures that only authorized individuals can seamlessly access and consume the APIs while upholding privacy and security measures.

i Note

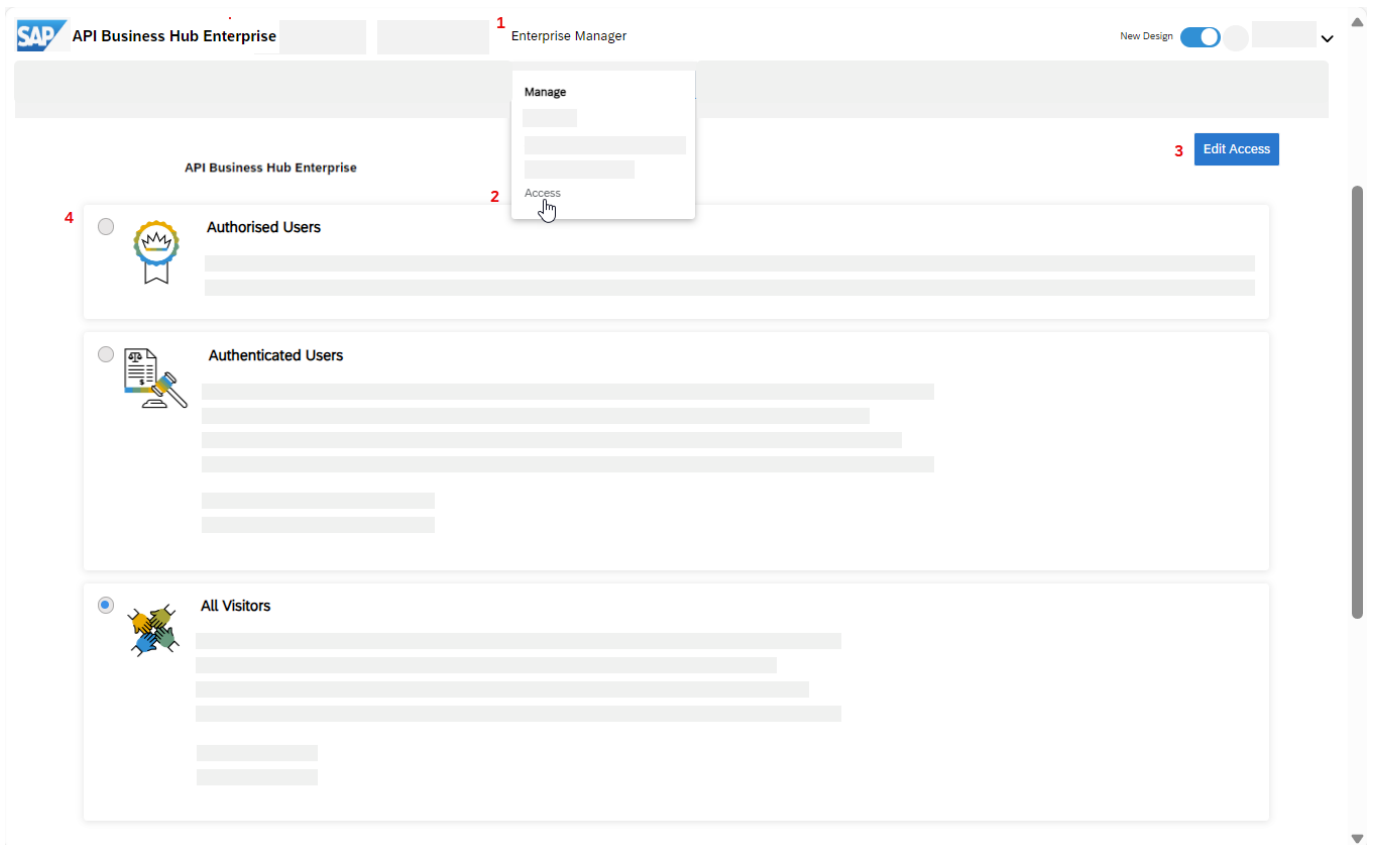
Access to the API business hub enterprise content using the API access plan is not affected by these permissions.

i Note

As an administrator of API business hub enterprise, please note that when you update these permissions, it may take up to 5 minutes for the changes to be applied for other users of the API business hub enterprise.

Procedure

1. Log on to the API business hub enterprise.
2. Choose **Enterprise Manager** > **Manage Access** from the top navigation bar.



3. Choose **Edit Access** to manage the level of access for different kinds of users.

Select the appropriate permission from the following list:

- o **Authorised Users:** These are the users who are registered as application developers or have the privilege of the application developer assigned through the SAP BTP cockpit. They can search, discover, and consume the contents of this API business hub enterprise application.

i Note

By default, the option for **Authorised Users** is selected. However, you have the flexibility to change it to either **All Visitors** or **Authenticated Users**, depending on your specific requirements.

- o **Authenticated Users:** These users are not registered as application developers but are successfully authenticated by the API business hub enterprise application and can access the following pages:

- Home
- Search results
- Product Details

Optionally, you can select the **API Details** checkbox to provide access to the **API Details** page as well.

- o **All Visitors:** These are the users who can visit the following pages even before they are authenticated by the API business hub enterprise application, that is without logging in to the API business hub enterprise application:

- Home
- Search results
- Product Details

Optionally, you can select the **API Details** checkbox to provide access to the **API Details** page as well.

4. Choose **Save** after providing the required permission.

Subscribe to a Product

You can subscribe to a product and add it to an existing application or create a new application.

Context

On the homepage, you can find the list of products under various categories. You can also view the various resources that each product has to offer.

Procedure

1. Log on to the API business hub enterprise.
2. You can either look for the product under various categories, or use the search bar to search for the product.
3. In the product details screen, choose **Subscribe**. You can subscribe to:
 - **Add to Existing Application**: The list of applications appears. Choose the required application.
 - **Create New Application**: Create an application by entering the name and title. The selected Product is added to the application by default.
4. Choose **Save**.

View Applications, Costs, and Analyze Reports

From the **My Workspace** page you can view your applications, costs, and analyze reports.

Applications created by you or any other application developers are displayed under the **Applications** section. For a created application, you can view the total number of calls made in the current month. From this page, you can create a new application or create an application on behalf of a user, mostly application developers.

By default, the **Cost** section displays the cost incurred in the last 6 months and the cost incurred in the current month. However, you can choose a month to view the cost incurred for that month.

You can analyze the performance of all your applications and get an overview of the application usage from the **Performance Analytics** section. In this section the runtime data is gathered, analyzed, and displayed as charts, headers, and key performance indicators (KPIs). For more information, see [Analyze Applications](#).

You can also navigate to the **Error Analytics** tab to view the error-related charts and KPIs for all the applications for the selected time period.

Create an Application

Create an Application to consume the required APIs.

An application is a discrete representation of the actual developer's application. It provides the developer with an API key to pass-in with every request to the API.

In API Management, similar APIs are bundled together to form products, which are published in the catalog. An application developer enters necessary details to register to the API business hub enterprise. After successful registration, the application developer can explore the required products and APIs to create an application. Once the application has been created successfully, the system generates an application key and application secret. If APIs in the application you created are protected via **Verify API Key** policy, then to access those APIs, you must pass the generated application key. Whereas, if APIs are protected

via **OAuth** policy, then to access those APIs, you must pass an OAuth token that can be obtained by using the combination of generated application key and application secret.

A user must be onboarded to API business hub enterprise only via Self-registration or **Add User** flow. For more information on registering in API business hub enterprise, see [Register on API business hub enterprise](#). In the **Add User** flow, the API business hub enterprise admin adds a user who wants to be onboarded to API business hub enterprise. However, the user who is requesting to be onboarded must ensure that the user details provided to the admin matches the user details obtained from the response of <developer portal url>/api/1.0/users.

Creating an Application with Application Developer Role

As an application developer you can create an application, and view the existing application details, and its associated products, custom attributes and analytics data.

Prerequisites

- You have the **AuthGroup.API.ApplicationDeveloper** role assigned to you. For more information on roles, see [Assign User Roles in API Management](#).

i Note

The **AuthGroup.API.ApplicationDeveloper** role must not be assigned manually to a user from the SAP BTP Cockpit. Also, this role must not be a part of any user group assignment.

The **AuthGroup.API.ApplicationDeveloper** role is assigned by default to a user who onboards to the API business hub enterprise using the self-registration process or via **Add User** flow. For more information on registering in API business hub enterprise, see [Register on API business hub enterprise](#).

In the Add User flow, the API business hub enterprise admin adds a user who wants to be onboarded to API business hub enterprise. However, the user who is requesting to be onboarded must ensure that the user details provided to the admin matches the user details obtained from the response of <developer portal url>/api/1.0/users.

Context

You are about to create an application and add products to your application. You can also select an existing application and view its details under the **Application Details** tab. Navigate to the **Products** tab to view the products and the rate plan associated with this application. You can add custom attributes to your applications, and manage them from the **Custom Attributes** tab. For more information, see [Add Custom Attributes to an Application](#). Navigate to the **Analytics** tab to analyze application usage, performance, and error count. For more information, see [Analyze Applications](#).

Procedure

1. Log on to the API business hub enterprise and navigate to **My Workspace**.

The applications you created earlier, are displayed under the **Applications** tab. For an existing application, you can view the total number of calls made in the current month on the **Applications** page.

2. To create an application, choose **Create New Application** in the **Applications** section.

3. In the **Create an Application** dialog, enter a **Title**, a **Description** (optional), and a **Callback URL** (optional) for the application.

You can also choose the options **Create this application on behalf of someone else** or **Already have Application Key & Secret**.

i Note

While creating the application, if you've selected the [Take me to this new application now](#) checkbox, you're directly navigated to the newly created application.

4. To add products to this application, choose [Add Products](#).
5. In the [Add Products](#) dialog, select the products that you want to associate with the application.

i Note

You can select multiple products published from the same portal but you can't select products published from different portals. For example, you can select products P1 and P2 from API portal A1. But you can't choose products, P3 and P4 from API portal A2 if you've already selected P1 and P2 from A1.

6. Choose [Create](#).

You can find the details of the application you just created under the [Application Details](#) tab.

i Note

The system generates an [Application Key](#) automatically when an application is created. You should use this application key value to access the API. At any point in time, you can regenerate the application key using [Regenerate Key](#) option. When you regenerate the key, both the application key and the secret key are changed. When you trigger API using the old key, then the response is negative. The old application key becomes invalid on regeneration.

7. To add a custom attribute, choose [Custom Attributes](#) > [Add Custom Attributes](#).
8. In the [Add Custom Attribute](#) dialog, enter a name and a value for your custom attribute and choose [Add](#)

i Note

You can create a maximum of 18 custom attributes per application. You cannot modify the name of a created custom attribute. However, you can modify its value whenever required. You can delete a custom attribute if it is no longer needed.

For more information on the usage of custom attributes in an application, see [Example: Accessing the Custom Attributes of an Application](#).

Creating an Application with API business hub enterprise Administrator Role

With the API business hub enterprise administrator role you can create an application on behalf of a user (application developer), and view the existing application details, and its associated products, custom attributes, and analytics data.

Prerequisites

You should have the [AuthGroup.API.Admin](#) role assigned to you. For more information on roles, see [Assign User Roles in API Management](#).

Context

An API business hub enterprise administrator can perform the following tasks:

- Create an application on behalf of a user (Application Developer) and handover the application key and secret to that user.

- Create new applications in different landscapes(example: production, nonproduction) by maintaining the same application key and secret.
- Create custom attributes at application level and regulate the API call logic.

Procedure

1. Log on to the API business hub enterprise and navigate to [My Workspace](#).

If you or other application developers have created applications earlier, they're displayed under the Applications section. For a created application, you can view the total number of calls made in the current month.

i Note

For API business hub enterprise administrators, analytics data is unavailable for those applications that they created on behalf of other users or application developers.

2. To create an application, choose [Create New Application](#) in the [Applications](#) section.
3. In the [Create an Application](#) dialog, enter a [Title](#), a [Description](#) (optional), and a [Callback URL](#) (optional) for the application.

As an administrator, you have the option to create an application on behalf of a user (application developer). To achieve this task, select the [Create this application on behalf of someone else](#) checkbox, and enter the [User ID](#) of the user on behalf of whom you are creating the application. If you already possess an application key and secret, then select the [Already have Application Key and Secret](#) checkbox and enter the [Application Key](#) and [Application Secret](#).

i Note

Application key and secret of an existing org can't be used in a new application in a new org. This implies that you'll not be able to use the same application key and secret in multiple orgs within the same data center and region.

i Note

While creating the application, if you've selected the [Take me to this new application now](#) checkbox, you're directly navigated to the newly created application.

4. To add products to this application, choose [Add Products](#).
5. In the [Add Products](#) dialog, select the products that you want to associate with the application.

i Note

You can select multiple products published from the same portal but you can't select products published from different portals. For example, you can select products P1 and P2 from API portal A1. But you can't choose products, P3 and P4 from API portal A2 if you've already selected P1 and P2 from A1.

6. Choose [Create](#).

You can find the details of the application you just created under the [Application Details](#) tab.

7. To add a custom attribute, choose [Custom Attributes](#) [Add Custom Attributes](#).
8. In the [Add Custom Attribute](#) dialog, enter a name and a value for your custom attribute and choose [Add](#)

i Note

You can create a maximum of 18 custom attributes per application. You cannot modify the name of a created custom attribute. However, you can modify its value whenever required. You can delete a custom attribute if it is no longer needed.

For more information on the usage of custom attributes in an application, see [Example: Accessing the Custom Attributes of an Application](#).

Example: Accessing the Custom Attributes of an Application

Let's say as a Developer Portal Administrator, you would want to restrict the number of calls to an application based on Application Key. To achieve this result, you create two applications `Application_1` and `Application_2`.

`Application_1` contains two products namely `Prod_1` and `Prod_2`.

`Application_2` contains two products namely `Prod_3` and `Prod_4`.

`Prod_1` and `Prod_2` contain two common APIs namely `API_1` and `API_2`.

For `Application_1`, add the following custom attributes and its corresponding values:

- `app_time_unit = minute`
- `app_quota_interval = 1`
- `app_quota_count = 9`

For `Application_2`, add the following custom attributes and its corresponding values:

- `app_time_unit = minute`
- `app_quota_interval = 1`
- `app_quota_count = 5`

To leverage these custom attributes in your API proxy execution, you must:

- add a verify API Key policy to the APIs that are part of your application.
- add a Quota policy to APIs that are part of your application.

For `API_1` and `API_2`, add the following sample policy payloads:

Sample payload for Verify API Key policy:

Sample Code

```
<!--Specify in the APIKey element where to look for the variable containing the api key-->
<VerifyAPIKey async='true' continueOnError='false' enabled='true'
xmlns='http://www.sap.com/apimgmt'>
    <APIKey ref='request.queryparam.apikey'/>
</VerifyAPIKey>
```

Sample payload for Quota policy:

Sample Code

```
<!-- can be used to configure the number of request messages that an app is allowed to submit to
<Quota async="false" continueOnError="false" enabled="true" type="calendar" xmlns="http://www.sap
    <Identifier ref="verifyapikey.Verify-api-key.developer.id"/>
    <!-- specifies the number of requests allowed for the API Proxy -->
        <Allow countRef="verifyapikey.Verify-api-key.app_quota_count"/>
    <!-- the interval of time for which the quota should be applied -->
    <Interval ref="verifyapikey.Verify-api-key.app_quota_interval"/>
```

```

<!-- used to specify if a central counter should be maintained and continuously synchron:
<Distributed>true</Distributed>
<!-- Use to specify the date and time when the quota counter will begin counting,
      regardless of whether any requests have been received from any apps -->
<StartTime>2015-2-11 12:00:00</StartTime>
<!-- if set to true, the distributed quota counter is updated synchronously. This means t
      the update to the counter will be made at the same time the API call is quota-ch
<Synchronous>true</Synchronous>
<!-- Use to specify the unit of time applicable to the quota. Can be second, minute, hour
<TimeUnit ref="verifyapikey.Verify-api-key.app_time_unit"/>

</Quota>

```

i Note

The attribute names `app_quota_interval`, `app_quota_count`, and `app_time_unit` must be the same attributes that you have added while creating the application.

To verify if the custom attributes are used in runtime, make an API call with `<appKey_1>` passed as a query parameter. For example, `https://<API_proxy_URL>?apikey=<appKey_1>`.

Call the same URL repeatedly and after 9 successive calls, your API proxy must return a Quota violation message.

Similarly, make an API call with `<appKey_2>` passed as a query parameter. For example, `https://<API_proxy_URL>?apikey=<appKey_2>`.

Call the same URL repeatedly and after 6 successive calls, your API proxy must return a Quota violation message.

Consume Applications

Once you create an Application, you can then consume the APIs based on your business requirements.

On subscribing to an application, the application developer receives an API key that the application must pass on every request to the API. API keys provide a simple mechanism for authenticating applications.

API Management generates API keys for applications, and enables you to add API key-based authentication to your APIs using policies. However, enforcement of the key is performed at the API proxy level, not by the API product itself. Therefore, you must ensure that all API proxies, and the corresponding resources defined by those API proxies, implement some form of key validation.

Before you use the API keys, ensure you are aware of the policies that support API keys and their functionality. There are two popular ways how APIKeys are provisioned. They are provided either as part of a Simple APIKey verification or as part of OAuth verification.

VerifyAPIKey Validation

If you define an API proxy to perform key validation by using the VerifyAPIKey policy, provide the API Key details to gain access to the applications.

OAuth 2.0 Validation

API Management supports standard OAuth flow. Currently SAP supports only Client credentials grant_type in OAuth.

Before you start, make a note of the Application key and secret for the required application.

A request is made using the following:

- URL: <URL of OAuth token>
- Method: POST
- Custom Header: Authorization value: Basic <Application key>:<Application secret>(base64 encoded)
- Payload: grant_type=client_credentials

This call returns a json payload with the OAuth validation response. On successful validation, it contains the access token. Note down the access token.

As default, the expiry time is configured to 3600 secs (1 hr). You can also configure the expiration time and the details that have to be displayed as part of the response.

You can now use this access token to fetch OAuth enabled services while making the actual business API call:

- URL: http[s]://<host>:<port>/<service_path>
- Custom Header: Authorization value: Bearer <access_token>

If the access token is valid, a valid service response is returned.

Analyze Applications

Use analytics capabilities to analyze application usage, performance, and error count.

API Management provides comprehensive analytics capabilities to understand application consumption. The runtime data is gathered, analyzed, and displayed as charts, headers, and key performance indicators (KPIs).

As an application developer, navigate to [My Workspace](#) to view the analytics information. By default, the analytics section displays the data for all the applications subscribed by you. All charts are displayed based on the Application Developer context.

The analytics information can be viewed as [Performance Analytics](#) and [Error Analytics](#).

- **Performance Analytics:** Displays the performance-related charts and KPIs for the selected time period. Following table describes the charts used to analyze the performance of all applications:

Performance Analytics

Chart Name	Description
Traffic Across all APIs	This chart displays total API calls made across all applications.
Slowest APIs	This chart displays the slowest APIs based on the API response time.
Top APIs	This chart displays most frequently used APIs.
Top Products	This chart displays most frequently used products based on the number of calls made to the APIs associated with the product.
Top Applications	This chart displays most frequently used applications based on the number of calls made to the APIs associated with the application.

- **Error Analytics:** Displays the error-related charts and KPIs for the selected time period. Following table describes the charts used to view error analytics of all the applications:

Error Analytics	
Chart Name	Description
Total Errors	This chart displays total errors.
Error Prone APIs	This chart displays number of errors per API.
Error Prone Applications	This chart displays number of errors per API associated with the application.

To view analytics for a specific application, navigate to the application details screen by selecting the required application. However, in the application details screen the analytics information is available only for the following KPIs:

- Traffic Across all APIs
- Slowest APIs
- Error Prone APIs

Consume API Proxies Using SAP Business Application Studio

The service center in SAP Business Application Studio provides a central entry point to explore products and services from the API business hub enterprise.

You can use this service center to develop your applications based on the OData Services available as a part of products published in API business hub enterprise. For more information, see [API Business Hub Enterprise Service Provider](#).

Test Runtime Behavior of APIs

Use the API Test Environment to test the runtime behavior of APIs.

The **Test Environment** enables you to test your APIs. Testing an API is essential to understand the runtime behavior of the APIs. It allows you to explore the resources associated with an API and execute the operations. It also allows you to test OData and REST-based services.

i Note

This document describes the new design of the API business hub enterprise. To view the documentation for the classic design, see [Test API Proxies](#).

Pre-requisite

The **Test Environment** tab will be visible to you only if you have the **AuthGroup.API.ApplicationDeveloper**role

Procedure

1. Log on to the API business hub enterprise.
2. Navigate to the **Test Environment**.

3. A list of APIs appears on the left.
4. Select the required API.

The URL for the selected API is populated automatically in the **API Test Console**. For the selected API, the URLs of the supported resources appear in the dropdown list. One resource is selected by default.

5. If you want to choose a different collection, use the dropdown list to select the required collection
6. If you have the URL of the service that contains the API, enter the service URL.
7. Choose **Authentication** to select the required type of authentication. You can choose from the following options:
 - a. **None**: No authentication required.
 - b. **Basic Authentication**: Provide a user name and password.
8. Enable the required method:
 - o **GET**: Reads an entity
 - o **POST**: Creates an entity
 - o **PUT**: Updates an entity
 - o **DELETE**: Deletes an entity

i Note

You can enable only the methods supported by the service.

9. Enter the **Request Body** for PUT and POST methods.

10. Choose **Header** to add a header.

i Note

If you want to add multiple headers, choose **Add Request Headers**.

11. Choose **Url Params** to enter the query parameter and value.

i Note

If you want to add multiple query parameters, choose the button **Add URL Params**. Test Console supports passing of custom headers such as X-sap-apimgt-proxy-host:-proxy-trai and X-sap-apimgt-proxy-port:- 8080

12. Choose **Send**.

The response appears in the tabs:

- o **Body**: View the formatted response.
- o **Body (Raw)**: View the unformatted response.
- o **Headers**: View the headers.
- o **Cookies**: View the cookies.

13. If you want to use the response body as an input request, choose **Use as Request** on the **Body (Raw)** tab.
14. To view the transactions based on the testing activity that you did, choose **Launch API Viewer**. For more information on tracing API proxy, see [Debug an API Proxy](#).

Graph

Extending traditional API Management, Graph enables you to expose all your business data in the form of a semantically connected data graph, accessed via a single unified and powerful API.

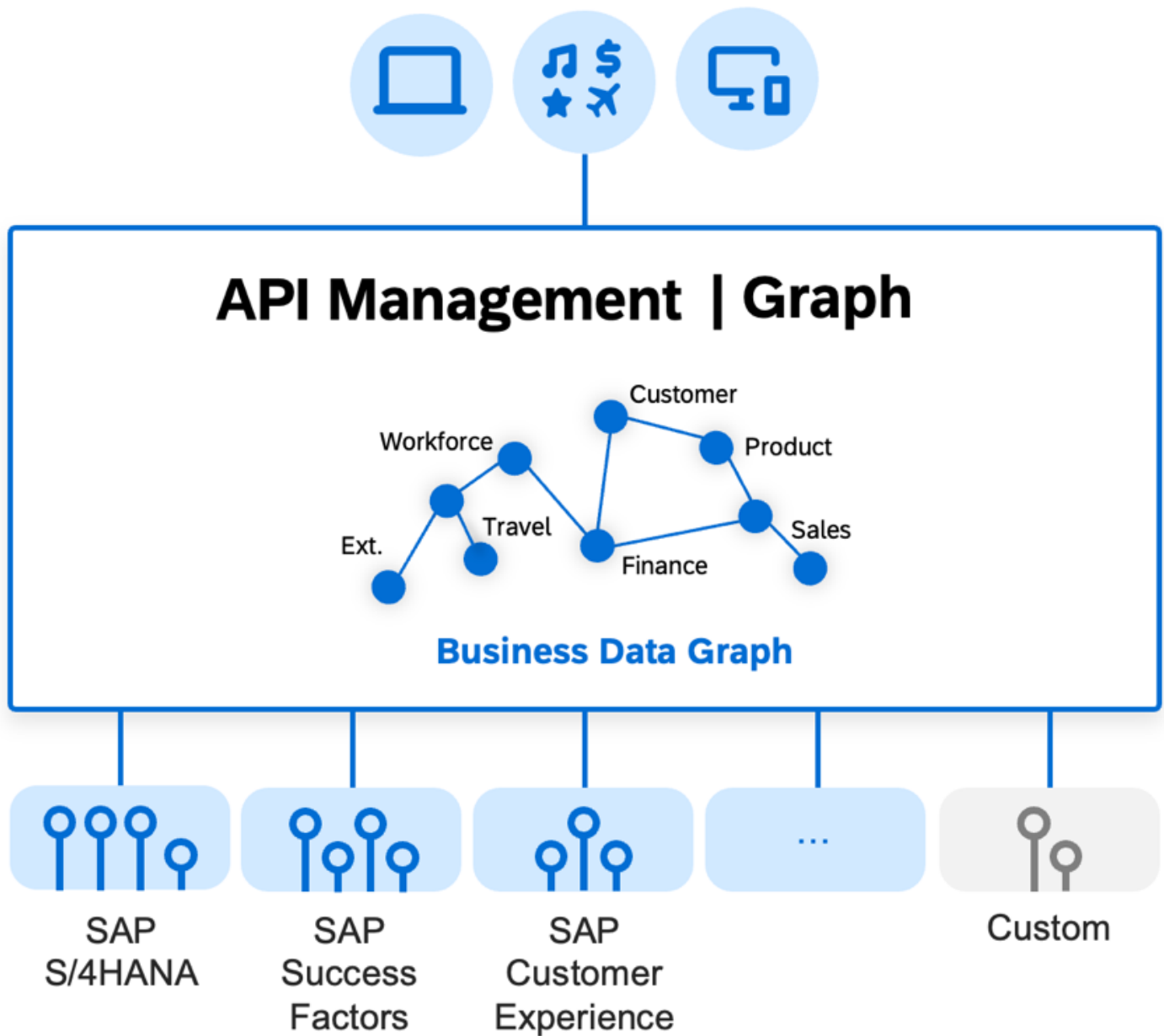
What Is Graph?

Graph is a capability of API Management within SAP Integration Suite. With Graph, developers access your business data as a single semantically connected data graph, spanning the suite of SAP products and beyond. Targeting SAP's ecosystem of partner and customer developers, Graph's powerful API reduces the cost and complexity of creating and deploying reusable extensions and other client applications.

Enterprise landscapes continue to expand in scale and complexity. Each additional system, SaaS, or microservice introduces new protocols, data models, connectivity, and security conventions. Real-world problems often span multiple lines of business, services, and APIs. Consequently, even the most experienced developers struggle to understand all of the technologies and interfaces involved. Developing new business-extending client applications requires an ever-growing range of expertise and skills. The phenomenal adoption of low-code tools by nonprofessional developers further increases the gap.

Enterprises use API Management to partially address this gap: APIs can be renamed, authentication can be streamlined, APIs can be protected against unauthorized access or threats. But this doesn't address the deeper problem: separate, disconnected APIs from different data sources and systems.

Extension Apps



Enterprise API Landscape

Graph is a solution to unify your business APIs in the form of a semantically connected data graph, accessed via a single powerful API. Out of the box, it provides developers a single connected and unified view of your SAP-managed business data. Graph consolidates thousands of data entities from SAP systems like SAP S/4HANA, SAP Sales Cloud, and SAP SuccessFactors, into one curated, semantically connected, data model. We call this connected graph a *Business Data Graph*.

The out-of-the-box data graph of SAP-managed data is the baseline, the starting point to your own data graph. You can expand it by adding your own data sources and your own data models, projections, and compositions, creating a unique data model of your business.

The business data graph is ultimately an abstraction of the data in your landscape, for developers. It establishes a separation of concerns, by exposing the data graph through a single unified API, entirely hiding the complexities of the landscape itself.

Developers use standard and powerful data graph query languages (OData V4 or GraphQL) to efficiently navigate the data, without being exposed to the complexity of data sources, URLs, connections, replications, VPNs, or underlying security

concerns. All the data, through one API. Often a single powerful graph-navigating query replaces the complex programming logic that would have been required to issue repeated queries to separate systems or APIs.

Graph technically acts as a scalable and stateless multitenant service, accepting navigation queries from applications, breaking up those queries, and accessing the APIs of the actual data sources on their behalf. Graph doesn't maintain or cache data; all data requests are semantically routed to the data source systems.

Because of the decoupling of the system landscape from applications, enterprises can deploy Graph-based applications more easily, across more landscapes, and at a lower cost.

As part of SAP Integration Suite, Graph is compatible with SAP's Cloud Application Programming model (CAP) and with the range of SAP Build development solutions.

Related Information

[Configure](#)

[Model](#)

[Develop](#)

Business Data Graph

The business data graph is a connected graph of all your business data.

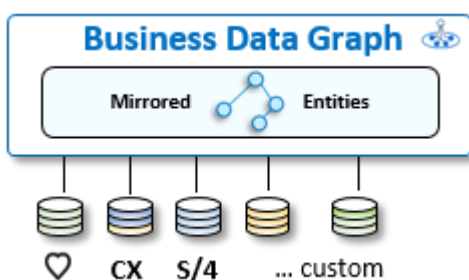
Business data is by nature richly connected. For example, a sales order has many items, and each item references a product. Still, traditional APIs (based on SOAP, or plain REST) access data as individual endpoints. Developers don't only use separate API calls to access different types of data, but they must also learn and remember how to reconstruct the connection between different types of data in their code. This leads to error-prone code and fragile, non-portable applications.

More recent modeling methodologies, such as SAP's Cloud Application Programming model (CAP), introduce semantic, graph-like, relationships between different data types. In the past few years, modern open-standard query protocols emerged that can take advantage of these graph-like data models. OData, based on REST principles, was one of the first such protocols, and GraphQL, developed by Facebook, is growing in popularity.

With Graph, developers use a single API, and the most up-to-date OData V4 and GraphQL protocols, to efficiently access all business data in a landscape of data sources.

Graph's data graph is constructed as a projection on these data sources. The nodes of the graph represent entities. Entities are composed of attributes – the data fields. Entity-connecting attributes, the edges of the graph, are referred to as associations. This graph is effectively an abstract data model, whose entities are defined as projections on entities from actual data sources (back-end applications, systems, and microservices) with one or more APIs. The data graph is constructed at design time in three steps:

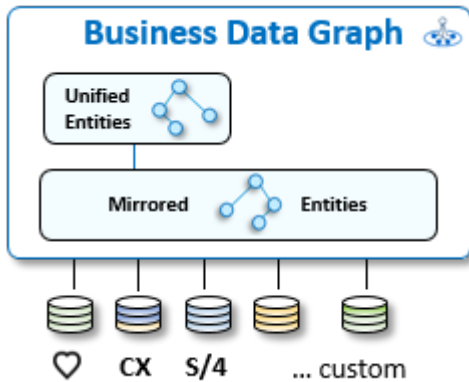
1. [Mirrored Entities](#)



Graph automatically creates projections from each of the entities of the discovered data sources in the landscape. We call these automatic data projections *mirrored entities*.

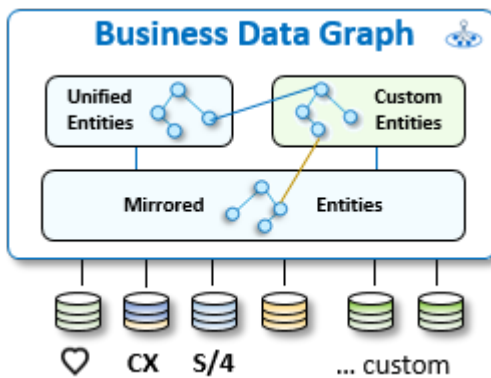
Graph distinguishes known data source types (for example, SAP S/4HANA, SAP SuccessFactors, and SAP C4C) and unknown data sources. The mirrored entities of supported SAP data sources are added to the data graph under a reserved SAP namespace (`sap.s4`, `sap.c4c`, and `sap.hcm`) and then connected to each other by potentially hundreds of additional semantic associations. Entities from unsupported data sources are mirrored under custom namespaces.

2. [Unified Entities](#)



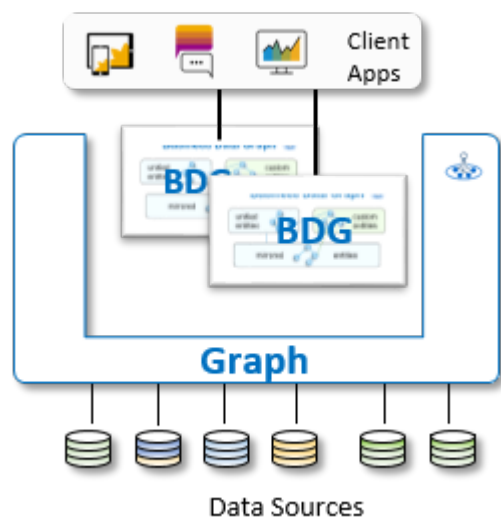
Graph then adds additional projections on top of the mirrored entities from supported SAP systems, which we call *unified entities*. Unlike the mirrored entities, these are thoughtfully designed and constructed projections, created by SAP experts under the reserved namespace `sap.graph`. Unified entities follow the SAP One Domain Model compatibility guidelines and are designed to bridge and connect semantically common business concepts from multiple data sources (for example, Business Partner, and Product). This allows client apps to get started with cross-system queries.

3. [Custom Entities](#)



Graph adds the final set of projections, *custom entities*. Custom entities are created by a skilled customer modeler to extend the data graph, through the design and addition of their own projections as a collection of attribute mappings from available SAP and non-SAP data source entities. The modeler can submit a set of custom entity definition files, developed with any text editor.

4. To complete the process, customer administrators create a *Business Data Graph*: a runtime manifestation of the data graph based on a concrete configuration and version, accessible as a unified API by activating the data graph. The business data graph provides a connected 360° view of the data in the landscape.



Graph functions as a runtime mediation layer. From the perspective of the API consumer, the nodes of a business data graph look like data entities. Client applications use the business data graph API and graph protocols, such as OData V4 or GraphQL, to issue powerful cross-entity requests like "get a list of products sold to customer C in August, ordered by value" or "what is the address of the top supplier of product P?". Graph interprets and deconstructs complex access requests into simpler query components that typically refer to a single data item (such as product, and supplier). It consults the business data graph configuration to determine which data sources hold the pertinent data for these simpler queries, in order to execute the access request and return the response. Here, Graph must select the appropriate native API that is exposed by the data source, implement its technical API protocol, and work around the specific limitations and capabilities of the native API.

Related Information

- [Data Locating Policy](#)
- [Modeling Guide](#)

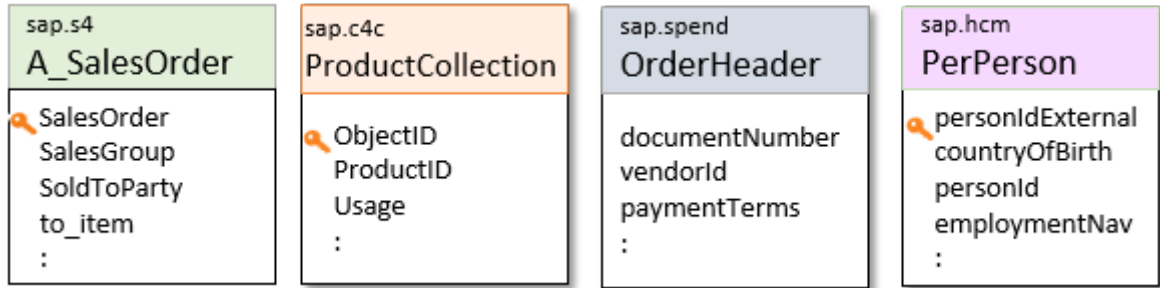
Mirrored Entities

Developers familiar with existing SAP product data models can continue to consume the resources of these models with Graph. Rather than accessing them separately via different system APIs, they're accessed from the business data graph, using the Graph API. The mirrored entities of supported SAP data sources are added to the data graph under a reserved SAP namespace. Entities from unsupported data sources are mirrored under custom namespaces.

The following SAP system-specific namespaces are supported:

Namespace	Description
<i>sap.s4</i>	Mirrored entities from the data model of SAP S/4HANA.
<i>sap.c4c</i>	Mirrored entities from the data model of SAP Sales Cloud, which is part of SAP CX's suite of products.
<i>sap.hcm</i>	Mirrored entities from the data model of SAP Human Capital Management (HCM, also known as SAP SuccessFactors).

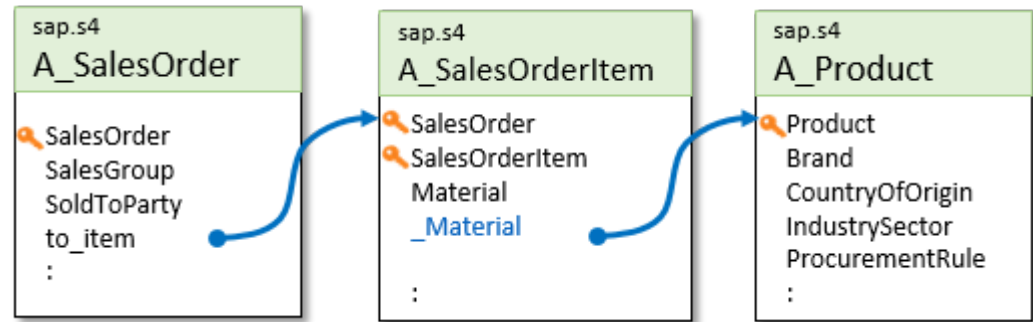
When referencing a mirrored entity, the namespace is simply prepended. For example: *sap.s4/A_SalesOrder* references sales order information from an SAP S/4HANA data source, and *sap.hcm/PerPerson* references person information from an SAP SuccessFactors data source.



Developers can easily combine all these resources in one application, focusing on the data, without having to know where the specific data sources are located or how to connect to them.

Additional Connections

In addition to consolidating the different system models, Graph introduces hundreds of additional connections between related entities in the business data graph, in the form of associations that are recognized by OData or GraphQL. A common example is the relation from an order item to an ordered product:



The additional connections are usually named by prefixing an underscore to the name of an attribute of type *String*, which represents the reference value. In the illustrated example, the original attribute *Material* is a string that represents a foreign key, and *_Material* is a relation.

Such relationships improve the semantic intent of the business data graph and lead to simpler, more intuitive, and more efficient navigational queries. For example, a developer could follow the illustrated relationships to access *sap.s4/A_SalesOrder(15)/to_Item(10)/_Material/Brand* in a single OData query, answering the question: *show the brand of the product ordered in item 10 of the sales order with key 15*.

Similarly, the business data graph represents hierarchical entities as compositions, which clearly expose the structural boundaries of the model, simplifying the interaction and reducing developer errors. An example of a composition is a book with chapters – you need to access the book to read the chapters. In the diagram, the relationship between *A_SalesOrder* and *A_SalesOrderItem* is modeled as a composition via *to_Item*, ensuring that developers always access a sales order item by going through the root of the composition (the root entity).

To complete the consolidation of the system models, Graph introduces hundreds of additional connections between related root entities in the business data graph, in the form of associations that are recognized by OData or GraphQL. A common example is the relationship from an order item to an ordered product.

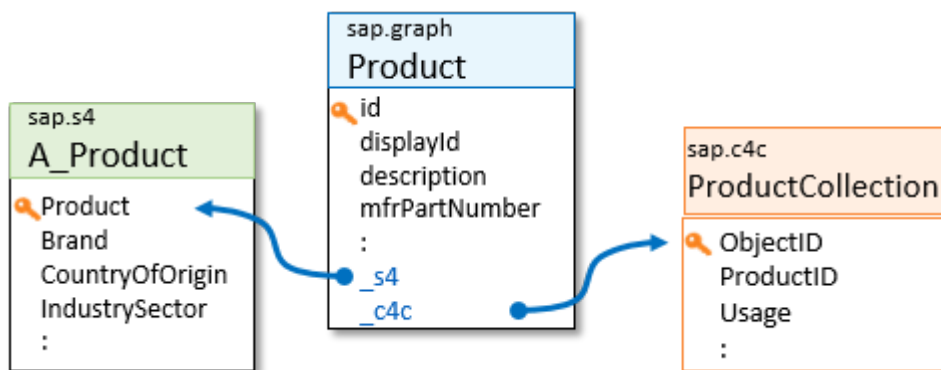
Using a more traditional API, this small example would have required at least three different round-trip API calls, plus the necessary expertise to develop the business logic to extract the keys to match the requirements of the different entity instances.

Unified Entities

Certain business objects (primarily master data, such as customer or product descriptions) are commonly replicated in multiple SAP systems, sometimes under different names. What one system calls *Product*, another may refer to as *ProductCollection*, *Material*, or even *supplierPart*. They all represent the same product object instance, with common attributes like its name and description, but then each SAP system manages additional, system-specific aspects: SAP S/4HANA maintains details of the manufacture and inventory of products, SAP Sales Cloud is concerned with the conditions of selling or using the product (for example, the skills required by a sales team), and SAP Ariba manages elaborate buyer-supplier pricing. Enterprises must synchronize the different representations of the same object, which often have different keys in the different systems, leading to high complexity for application developers as well.

Developers often only need the common attributes of such business objects and are mystified by the different system representations and key sequences of the same data. To address this, Graph introduces unified entities. Unified entities define the common and most widely used attributes of a business object, using a consistent and easier to understand structure and naming convention. Unified attributes are accessed under the *sap.graph* namespace.

Developers of extension apps use these common attributes, regardless of where this data resides. Under the hood, Graph maps these attributes to one of the data sources in the landscape, but this doesn't concern the developer. Consequently, the use of unified entities results in SAP-extending apps that are portable and reusable across a wide range of customer landscapes.



Unified entities have association attributes that connect them to the system-specific representations of the same object (*_s4* and *_c4c*). These associations effectively provide developers with a consolidated and navigable 360° view of all the attributes of these objects in SAP. To access an attribute such as *Brand*, the app simply issues a *sap.graph/Product(123)/_s4/Brand* request. Of course, SAP S/4HANA system-specific attributes are only available if such a system is part of the underlying enterprise landscape. Graph handles key mapping complexities under the hood. For more information, see [Data Locating Policy](#).

Whether or not an entity is read-only depends on various parameters of your landscape. The metadata of your business data graph tells you which entities are writable.

Citizen developers use low-code tools to access the unified entities. Advanced developers with more complex requirements can follow the edges of the graph to use detailed system-specific attributes.

In summary, unified entities play two roles:

1. They provide consistent and simplified access to the **common** attributes of a multi-sourced business object. This simplified and common information provides sufficient detail for many extension applications. It is written without worrying about the differences and more complex variations of the system-specific models, making these applications portable over a broad range of landscape configurations.
2. They "connect" the system-specific entities via explicit associations. This provides developers of extension applications a comprehensive 360° perspective of how objects are managed in their enterprise and supports powerful cross-system queries of system-specific attributes.

SAP is gradually introducing new unified entities, along with the extended support of Graph for more SAP systems. The initial release of Graph supports SAP S/4HANA, SAP Sales Cloud, and SAP SuccessFactors.

To all Graph developers, the business data graph looks and behaves like a single, giant, consistent, navigable SAP system, accessible via a single API and access protocol, ignoring the physical landscape of data source system instances.

Custom Entities

Custom entities are created by a skilled customer modeler to extend the data graph, by designing and adding their own projections as a collection of attribute mappings from available SAP and non-SAP data source entities. The modeler can submit a set of custom entity definition files, developed with a text editor of their choice.

The ability to extend the SAP data graph with custom entities is a powerful capability – customers can essentially design their own corporate data model. Here are some of the advantages:

Use the Same Protocols

By mediating your data sources as a data graph, developers enjoy the use of a single data endpoint, a simplification of access and security, and the consistent use of the same query language and protocol. Investments in client-side SDKs, frameworks, and data access abstractions apply to all of the data sources.

Create Your Own API Shape

From simple renaming, to more powerful transformations, custom entities allow you to control how the data is perceived by app developers. You can:

- Replace an existing entity with one that is simpler to understand, by filtering out unnecessary or undesired attributes.
- Rename entities and attributes to match a consistent corporate or SAP data naming convention.
- Design your interface to precisely match the API expectations of application developers, or of existing applications.

Hide Landscape Inconsistencies

Hide incompatibilities between data-source variations or versions as an abstraction. This is useful while preparing for major system upgrades or migrations involving API incompatibilities. Custom entities can hide such API changes, providing you with more control over how to implement the migration, while not breaking dependent applications.

Add or Change Semantics

- Replace a hard-to-understand normalized data representation (often representing the way that the data is stored) into a denormalized view that is easier to consume by client applications.
- Connect separate entities (with string-type foreign keys) into a navigable graph of entities, by introducing associations and compositions into the graph.
- Turn associations into compositions (to many) or a structured type (of one), making the semantics of the relationship more obvious to understand for consumers.
- Combine attributes from separate source entities into one virtual, composed entity. Combine data attributes into compositions that hide underlying implementation technicalities. A common example is a side-car extension, such as a CAP-created application that extends an SAP S/4HANA data model, such as a BusinessPartner. Using custom entities, you can present a new natural entity with attributes from both.

i Note

Certain restrictions apply, such as the inability to guarantee atomic data modification, when writing back to two separate data sources.

Introduce More Control and Security

In many cases, IT administrators can use Graph to avoid the need to create data copies, replications, and complex ETLs (Extract Transform Load) to serve the need for simpler data APIs for certain application developers and use cases. Administrators can do the following:

- Secure their data, by only exposing data that is safe to use.
- Using a data-filter, custom entities can be used to systematically access only a subset of data.
- Projections can support finer-grain authorization: a custom entity can be read-only, for example, while the underlying entities are not.

Initial Setup

As a Subaccount or Tenant Administrator, you need to add Graph as a capability of API Management within SAP Integration Suite.

Prerequisites

1. Your SAP BTP Global Account administrator has already created a subaccount. For more information, see [Create a Subaccount](#).
2. An SAP Integration Suite entitlement has been created for your subaccount. For more information, see [Configure Entitlements](#).
3. You have subscribed to the SAP Integration Suite and assigned the **Integration_Provisioner** role collection to yourself. You must assign this role to add capabilities, such as API Management and Graph, on the Integration Suite home page.

For more information, see [Set Up API Management from Integration Suite](#) (Steps 1–3).

4. You have activated Graph on the Integration Suite home page and assigned yourself to the following Graph role collection:

- **Graph.KeyUser**

When you assign yourself to this role collection, you are automatically assigned to the following roles:

- **Graph_Key_User**
- **Graph_Navigator_Viewer**

1. Configure Entitlement for Graph

Your global account administrator is responsible for configuring the Graph entitlement as follows:

1. Go to **Entitlements** in the SAP BTP cockpit.
2. Choose **Configure Entitlements** > **Add Service Plans**.
3. Search for and choose **SAP Graph** and do the following:
 - a. Choose one or more of the following plans:

- [api](#) for business data graph consumption
- [configuration](#) to configure business data graphs using the Graph Configuration API.

b. Choose [Add Service Plan](#).

c. Choose [Save](#).

2. Configure Graph

1. Go to the [Integration Suite](#) home page, and under [Capabilities](#), choose [Add Capabilities](#).
2. On the [Activate Capabilities](#) dialog, under [Select Capabilities](#), choose [Design, Develop, and Manage APIs](#) and choose [Next](#).
3. To activate [Graph](#), select the following, and choose [Next](#):
 - a. [Enable SAP API Business Hub Enterprise](#)
 - b. [Graph](#)

i Note

If you have already set up API Management, choose [Edit](#), and select [Graph](#). You must select [SAP API Business Hub Enterprise](#) to activate [Graph](#).

4. Choose [Activate](#) and once the status on the [Summary](#) changes from [In Progress](#) to [Active](#), choose [OK](#).

3. Define Users for Graph

Graph doesn't manage user identities and it can't directly authenticate clients. Instead, it relies on XSUAA, the SAP Authorization and Trust Management Service.

There are two ways to define users:

- Add users, one-by-one, for the purpose of bootstraps, demos, and proofs of concepts (PoCs) to the SAP BTP subaccount (for example, via the SAP BTP cockpit).
- Use your own user base and Identity Provider (IdP). For this, you must establish a trust relationship with a SAML 2.0 IdP in your subaccount.

For more information, see [Trust and Federation with Identity Providers](#).

Once you enable Graph, you see both Graph role collections in your list.

Enable Graph Users

- Graph Key User

The creation and activation of business data graphs is managed by a user with a special role, the [Graph_Key_User](#) authorization role. You must assign this role to one or more users so that they can create and activate business data graphs for a landscape.

To assign the [Graph.KeyUser](#) role collection to a user, do the following:

1. In the SAP BTP cockpit, go to [Security > Users](#).
2. Select the relevant user. Under [Role Collections](#), choose [Assign Role Collection](#).
3. Search for [Graph.KeyUser](#), select the role collection, and choose [Assign Role Collection](#).
4. To apply the role, go back to the Integration Suite home page.

- Graph Navigator Viewer

Graph Navigator is a tool in SAP API business hub enterprise that developers use to inspect business data graphs. For more information, see [Graph Navigator in SAP API Business Hub Enterprise](#).

To assign the **GraphNavigator.Viewer** role collection to a user with the **Graph_Navigator_Viewer** role, do the following:

1. In the SAP BTP cockpit, go to **Security > Users**.
2. Select the relevant user. Under **Role Collections**, choose **Assign Role Collection**.
3. Search for **GraphNavigator.Viewer**, select the role collection, and choose **Assign Role Collection**.
4. To apply the role, go back to the Integration Suite home page.

i Note

Reselect your subaccount to access **Users** after assigning the role collection.

For more information, see:

- [Define a Role Collection](#)
- [Add Roles to a Role Collection](#)
- [Assign Users to Role Collections](#)

Connect to Your Business Systems

As an administrator, you need to set up the data sources you want to use in your business data graphs, setup the connectivity between Graph and your business systems, and create destinations.

Graph supports data sources based on SAP S/4HANA (all editions, cloud and on-premise), SAP SuccessFactors, and SAP Sales Cloud.

i Note

Graph supports SAP ERP Central Component (ECC) content which has been prepared and exposed as an OData service. To create the service interface, you can use Gateway Builder starting from SAP Gateway version GW_FND 7.52 and the latest SP (refer to [2217489 - Maintenance and Update Strategy for SAP Fiori Front-End Server](#)) or OData Provisioning.

Set Up Data Sources

To create a business data graph, you must have data sources (business systems). Adding data sources to your subaccount consists of two steps for each data source:

1. Establish trust between the business systems and the SAP BTP subaccount. You can include various SAP systems into a formation and thus combine diverse SAP solutions into an extended business scenario.

For more information, see [Including SAP Systems in a Formation](#).

2. Create a destination for each service of these data sources that is exposed to client applications.

Custom SAP BTP Destination Annotations

Graph supports custom annotations for SAP BTP destinations.

In the SAP BTP cockpit, you can add **Additional Properties** to destinations. The properties provide additional configuration options. You can add predefined properties forSAP BTP or add freestyle properties using any name and value.

The following table provides an overview of the custom annotations that Graph supports for SAP BTP destinations:

Name	Value	Description
Graph . Ignore	true	A destination is ignored during Graph configuration, generation, and activation.

Related Information

[Create a Business Data Graph in Integration Suite](#)
[Actions and Functions](#)

SAP S/4HANA Cloud

As an administrator of an SAP S/4HANA Cloud configured IAS tenant you need to configure the communication between Graph and SAP S/4HANA Cloud.

To do this, you need to set up the connectivity on the SAP S/4HANA Cloud tenant and on the SAP BTP tenant for SAP Cloud Identity Access Governance (IAG), and create destinations.

Set Up Connectivity

Prerequisites

The SAP S/4HANA Cloud administrator must have the following business role assignments:

Business Role ID	Area
SAP_BCR_CORE_COM	Communication Management
SAP_BCR_CORE_IAM	Identity and Access Management
SAP_BCR_CORE_EXT	Extensibility

Procedure

- 1. Configure the single-sign on (SSO) Identity Authentication service.
- 2. Upload your key-pair keystore in SAP BTP cockpit. Otherwise, use the default key-pair keystore.
- 3. Set up the SAP S/4HANA Cloud side.
- 4. Set up the SAP BTP side.

For a detailed description of each of these steps, see [Using SAML Bearer Assertion Authentication](#).

Handover Information to the SAP BTP Administrator

- URL of your SAP S/4HANA Cloud account
- Name of the communication user in the SAP S/4HANA Cloud tenant
- Password for the communication user
- Token Service URL from the OAuth 2.0 details in the communication arrangement

- Provider name for the communication system in SAP S/4HANA Cloud

Create Destinations

Create a destination to enable the communication between your business system and Graph.

You can define destinations to data sources using two authentication models: identity propagation (preferred) and based on a technical user. Both options are described here for each supported business system.

i Note

Graph caches destination settings. If you want to change these settings after the business data graph is created, you need to edit the business data graph configuration and update it. For more information, see [Modify Your Business Data Graph](#).

As the SAP BTP administrator, you must create and configure an HTTP destination that either supports principal propagation or is based on a technical user.

i Note

An HTTP destination can be generated automatically by extending SAP S/4HANA Cloud in the Cloud Foundry and Kyma environments. For more information, follow the steps in:

- [Extending SAP Solutions Using Automated Configurations](#)
- [Extending SAP S/4HANA Cloud in the Cloud Foundry and Kyma Environment](#).

Destination for Identity Propagation

Prerequisites

The SAP S/4HANA Cloud Administrator configured the communication between Graph and SAP S/4HANA Cloud. You know the following:

- URL of your SAP S/4HANA Cloud account
- Name of the communication user in the SAP S/4HANA Cloud tenant
- Password for the communication user
- Token Service URL from the OAuth 2.0 details in the communication arrangement
- Provider name for the communication system in SAP S/4HANA Cloud

For more information, see [Principal Propagation Scenario: Cloud to Cloud](#).

Procedure

Create an **OAuth2SAMLBearerAssertion** HTTP destination and configure its settings as follows:

1. Go to SAP BTP cockpit ► **Connectivity** ► **Destinations** ► and choose **New Destination**. Enter the following:

Parameter	Value
Name	Enter the destination name.
Type	HTTP

Parameter	Value
Description	Optional
URL	The service URL from the communication arrangement. i Note Make sure you use the HTTPS protocol.
Proxy Type	Internet
Authentication	OAuth2SAMLBearerAssertion
Key Store Location	In the dropdown list, select the key-pair keystore file you uploaded in Upload Your Key-Pair Keystore in SAP BTP .
Key Store Password	The password for the keystore. i Note The password for the keystore must be the same as the one for the key-pair entry in the keystore file.
Audience	The URL of your SAP S/4HANA Cloud account. To get it, log on to your SAP S/4HANA Cloud account. Select the profile picture, choose Settings , and copy the value from the Server field. Add <code>https://</code> to the beginning of the URL.
Client Key	The name of the communication user you have in the SAP S/4HANA Cloud tenant.
Token Service URL	The token service URL from the OAuth 2.0 details in the communication arrangement.
Token Service URL Type	Dedicated
Token Service User	The name of the communication user in the SAP S/4HANA Cloud tenant.
Token Service Password	The password for the communication user.
System User	Leave the field empty.

2. Configure the required additional properties. To do so, in the **Additional Properties** panel, choose **New Property**, and enter the following parameters:

Parameter	Value
authnContextClassRef	urn:oasis:names:tc:SAML:2.0:ac:classes:X509
nameIdFormat	urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
scope	For example: API_BUSINESS_PARTNER_0001
userIdSource	email

3. Select the **Use default JDK truststore** checkbox.

4. Click **Save**.

Destination with a Technical User

Prerequisites

The SAP S/4HANA Cloud administrator configured the communication between Graph and SAP S/4HANA Cloud. You know the following:

- URL of your SAP S/4HANA Cloud account
- Name of the communication (technical) user in the SAP S/4HANA Cloud tenant
- Password for the communication user

Procedure

Create a **BasicAuthentication** HTTP destination and configure its settings as follows:

1. Go to SAP BTP cockpit ► **Connectivity** ► **Destinations** ► and choose **New Destination**. Enter the following:

Parameter	Value
Name	Enter a destination name.
Type	HTTP
Description	Optional
URL	The service URL from the communication arrangement. i Note Make sure you use the HTTPS protocol.
Proxy Type	Internet
Authentication	BasicAuthentication
User	The name of the communication user in the SAP S/4HANA Cloud tenant.
Password	The password for the communication user.

2. Click **Save**.

SAP S/4HANA

As an administrator for SAP S/4HANA, you need to configure the communication between Graph and SAP S/4HANA by installing a cloud connector and creating destinations.

Setup Connectivity

As the Cloud Connector administrator, you must install a cloud connector on your business system.

Prerequisites

Your SAP BTP Administrator has provided the following:

- Region
- Subaccount
- Assigned a Cloud Connector administrator to a role or role collections . For more information, see [Initial Configuration](#).

Procedure

1. Install and configure the cloud connector. For more information, see [Cloud Connector](#).
2. Add the SAP BTP subaccount to the cloud connector. For more information, see [Manage Subaccounts](#).

Establish Trust

1. Configure the single-sign on (SSO) Identity Authentication Service
2. Upload your key-pair keystore in SAP BTP cockpit. Otherwise, use the default key-pair keystore.
3. Set up the SAP S/4HANA side.
4. Set up the SAP BTP side.

For a detailed description of each of these steps, see [Authenticating Users against On-Premise Systems](#).

Create Destinations

As the SAP BTP administrator, you must create and configure an HTTP destination that either supports principal propagation or is based on a technical user.

Destination for Identity Propagation

Prerequisites

- Cloud Connector administrator has added the SAP BTP subaccount to the cloud connector.
- To configure the HTTP destination, you need to know the following:
 - Host and port of the system exposed by the cloud connector
 - Authentication type
 - Location ID (if configured in the cloud connector)
 - SAP S/4HANA tenant ID

Procedure

Create a [PrincipalPropagation](#) HTTP destination and configure its settings as follows:

1. Go to SAP BTP cockpit ► [Connectivity](#) ► [Destinations](#) , and choose [New Destination](#). Enter the following:

Parameter	Value
Name	The destination name.
Type	HTTP
Description	Optional
URL	Virtual URL of the protected on-premise application.

Parameter	Value
Proxy Type	OnPremise
Authentication	PrincipalPropagation

2. Click **Save**.
- To create a destination that uses the host exposed by the cloud connector, make sure that:
- **Proxy Type** is set to **OnPremise**.
 - **Authentication Type** is supported by your cloud connector configuration. For more information, see [Configuring Principal Propagation](#).
 - The SAP S/4HANA tenant ID is configured as an **Additional Property**:

Parameter	Value
sap-client	SAP S/4HANA tenant ID

For more information about how to create the destination, see [Create HTTP Destinations](#).

Destination with a Technical User

Prerequisites

The SAP S/4HANA administrator configured the communication between Graph and SAP S/4HANA. You know the following:

- URL of your SAP S/4HANA account
- Name of the communication (technical) user in the SAP S/4HANA tenant
- Password for the communication user

Procedure

Create a **BasicAuthentication** HTTP destination and configure its settings as follows:

1. Go to SAP BTP cockpit▶**Connectivity**▶ **Destinations**▶ and choose **New Destination**. Enter the following:

Parameter	Value
Name	Enter the destination name.
Type	HTTP
Description	Optional
URL	The service URL from the communication arrangement. i Note Make sure you use the HTTPS protocol.
Proxy Type	Internet
Authentication	BasicAuthentication

Parameter	Value
User	The name of the communication user in the SAP S/4HANA tenant.
Password	The password for the communication user.

2. Click **Save**.

Related Information

[Configure Systems in Cloud Connector](#)

SAP Sales Cloud

As an administrator for SAP Sales Cloud, you need to establish trust with the SAP BTP subaccount and create destinations..

You can use the SAML Bearer assertion flow for consuming OAuth-protected resources. Users are authenticated by using SAML against the configured trusted identity providers. The SAML assertion is then used to request an access token from an OAuth authorization server.

This access token must be added as an Authorization header with the value Bearer <access - token> in all HTTP requests to the OAuth-protected resources.

Configure the OAuth Identity Provider in SAP Sales Cloud

As the SAP Sales Cloud administrator, you must add the SAP BTP service provider as a trusted OAuth identity provider.

For more information, see [Configure OAuth Identity Provider in SAP Sales Cloud](#).

Prerequisites from SAP BTP administrator

- Trust certificate
- Region host
- Subaccount ID

Procedure

Create an **OAuth2SAMLBearerAssertion** HTTP destination and configure its settings as follows:

1. Go to SAP BTP cockpit▶**Connectivity**▶**Destinations**▶ and choose **New Destination**. Enter the following:

Parameter	Value
Name	Enter the destination name
Type	HTTP
Description	Optional
URL	The service URL from the communication arrangement. i Note Make sure you use the HTTPS protocol.

Parameter	Value
Proxy Type	Internet
Authentication	OAuth2SAMLBearerAssertion
Key Store Location	In the dropdown list, select the key-pair keystore file you uploaded in Upload Your Key-Pair Keystore in SAP BTP .
Key Store Password	The password for the keystore. i Note The password for the keystore must be the same as the one for the key pair entry in the keystore file.
Audience	The URL of your SAP Sales Cloud account. To get the URL, log on to your SAP Sales Cloud account. Select the profile picture and then choose Settings and copy the value from the Server field. Add <i>https://</i> to the beginning of the URL.
Client Key	The name of your communication user in the SAP Sales Cloud tenant.
Token Service URL	This is the Token service URL from the OAuth 2.0 Details in the communication arrangement.
Token Service URL Type	Dedicated
Token Service User	The name of the communication user in the SAP Sales Cloud tenant.
Token Service Password	The password for the communication user.
System User	This parameter is not used, leave the field empty.

2. Configure the required additional properties. To do so, in the **Additional Properties** panel, choose **New Property**, and enter the following:

Parameter	Value
authnContextClassRef	urn:oasis:names:tc:SAML:2.0:ac:classes:X509
nameIdFormat	urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
scope	For example: API_BUSINESS_PARTNER_0001
userIdSource	email

- 3. Select the **Use default JDK truststore** checkbox.
- 4. Click **Save**.

Configure the OAuth Client for OData Access

As the SAP Sales Cloud administrator, you must configure the OAuth client for OData access to SAP Sales Cloud OData APIs.

For more information, see [Configure the OAuth Client for OData Access](#).

Prerequisites

- You know the name of the identity provider created in [Configure OAuth Identity Provider in SAP Sales Cloud](#).

Handover Information to the SAP BTP Administrator

- Client ID
- Client secret
- Scope

Create Destinations

As the SAP BTP administrator, you must create and configure an HTTP destination that either supports principal propagation or is based on a technical user.

Destination for Identity Propagation

Prerequisites

The SAP Sales Cloud administrator has configured the communication between Graph and SAP Sales Cloud. You know the following:

- Client ID
- Client secret
- Scope

Procedure

As the SAP BTP administrator, you must create and configure an HTTP destination that supports principal propagation. For more information, see [Create and Configure the HTTP Destination](#).

Destination with a Technical User

Procedure

Create a **BasicAuthentication** HTTP destination and configure its settings as follows:

1. Go to SAP BTP cockpit▶**Connectivity** ▶ **Destinations** ▶ and choose **New Destination**. Enter the following:

Parameter	Value
Name	Enter the destination name.
Type	HTTP
Description	Optional
URL	The service URL from the communication arrangement. i Note Make sure you use the HTTPS protocol.
Proxy Type	Internet
Authentication	BasicAuthentication
User	The name of the communication user in the SAP Sales Cloud tenant.

Parameter	Value
Password	The password for the communication user.

2. Click [Save](#).

SAP SuccessFactors

As an SAP SuccessFactors and SAP BTP administrator, you need to set up the connectivity to Graph and create destinations.:

1. Download the certificate from the [Destination](#) service of your SAP BTP account.
2. Create an OAuth client in SAP SuccessFactors.
3. Create the [Destination](#) on your SAP BTP Destination service.

Download the Trust Certificate from the Destination Service

As the SAP BTP subaccount administrator, download the trust certificate from your account's destination service.

Procedure

1. In the SAP BTP cockpit, go to your subaccount in the SAP BTP, Cloud Foundry environment.
2. Choose [Connectivity > Destinations](#).
3. Choose [Download Trust](#) to get the certificate for this subaccount and save it to your local file system.

Create an OAuth Client on SAP SuccessFactors

As the SAP SuccessFactors administrator, you need to create an OAuth client that will be used to configure the trust between SAP SuccessFactors and Graph.

Procedure

1. In the SAP SuccessFactors system, go to the [Admin Center](#) and search for [OAuth](#). Choose [Manage OAuth2 Client Applications](#) from the search results.
2. Choose [Register Client Application](#).
3. In the [Application Name](#), choose a descriptive name for the client of your choice.
4. In the [Application URL](#) field, enter the URL of yourGraph tenant.
5. In the [X.509 Certificate](#) field, open the certificate you downloaded before from the Destination service using any text editor, then copy the content between -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----, and paste it into the field.
6. Choose [Register](#) to save the OAuth client.

Create Destinations

As the SAP BTP administrator, you need to create an HTTP destination to be able to make calls to the SAP SuccessFactors HCM Suite OData APIs using SAML 2.0 Bearer Assertion authentication.

Procedure

1. In the SAP BTP cockpit, go to your subaccount in the SAP BTP, Cloud Foundry environment.
2. Choose [Connectivity > Destinations](#).

3. Choose **New Destination** and fill in the following parameters:

Parameter	Value
Name	Enter the destination name
Type	HTTP
Description	Optional
URL	Enter the URL of the SAP SuccessFactors OData API that you want to consume. For a list of the API Endpoint URLs for the SAP SuccessFactors environments, see About HXM Suite OData APIs
Proxy Type	Internet
Authentication	OAuth2SAMLBearerAssertion
Key Store Password	The password for the keystore. i Note The password for the keystore must be the same as the one for the key-pair entry in the keystore file.
Audience	www.successfactors.com
AuthnContextClassRef	urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
Client Key	Enter the API Key of the OAuth client that you created in SAP SuccessFactors.
Token Service URL	Enter the API Endpoint URL for the SAP SuccessFactors instance followed by /oauth/token. For example, https://apisalesdemo2.successfactors.eu/oauth/token . For a list of the API Endpoint URLs for the SAP SuccessFactors environments, see About HXM Suite OData APIs .
Token Service URL Type	Dedicated

4. Configure the required additional properties. To do so, in the **Additional Properties** panel, choose **New Property**, and enter the following parameters:

Parameter	Value
apiKey	Enter the API Key of the OAuth client that you created in SAP SuccessFactors.
companyId	The ID of your SAP SuccessFactors company.
nameIdFormat	urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
userIdSource	email

5. Select the **Use default JDK truststore** checkbox.

6. Click **Save**.

Custom OData Services

In addition to connecting to SAP business systems, you can set up a custom datasource that supports OData protocol to enable Graph to communicate with your API service.

As an administrator of your OData API service, you need to configure the communication between Graph and your business system.

Setup Connectivity

Make sure your OData API service is publicly accessible. All the necessary setup should be considered for your custom API service, such as secure communication.

Create Destinations

Create a destination to enable the communication between your business system and Graph. As the SAP BTP administrator, you need to create an HTTP destination to be able to make calls to OData APIs from Graph. The authentication type is based on your setup for that specific business system.

You can create destinations to your API service using BasicAuthentication, or even using NoAuthentication for public services (mainly for testing purposes and not recommended for production).

Destinations to services that require special HTTP headers can be defined with additional URL.headers. Use XXX properties for this purpose. For more information, see [HTTP Destinations](#).