**Access Management**

Anypoint Access Management enables you to create Anypoint Platform account or configure External Identity. It configures access and permissions within your organization and, depending on the access level, manages the users and their roles.  It provides administrative and organizational abilities that apply to the various entitlements in Anypoint Platform (API Manager, Runtime Manager, AnypointMQ,and more).

As the administrator of your organization or one of its business groups, you can enable, disable, or delete users. You can also invite new users and also manage existing users of your organization. A user with ‘Organization Admin’ can only configure any changes from the Access Management. A developer does not have access to Access Management by default.

**Features of Access Management:**

* Managing organizations, business groups,  users, roles, permissions, and environments
* Monitoring subscriptions and audit log
* Connect to existing (Idp) IDentity Management Solutions
  + - 1. **Organization**
* An Organization is an administrative collection of resources and users. When you create an Anypoint Platform account, a master (or root) organization is created, and you are assigned as the owner of the organization. Organization owners automatically inherit the Organization Administrators role.
* The organization name is the name you entered in the Company field in the initial Anypoint Platform signup form. You can change the name in the organization settings.
* An organization is an account where multiple users can share resources, including applications and environments. The level of access users have to various resources depends on their assigned roles and permissions. For example, one user might have permission to manage API alerts, while another user has permission only to view API alerts.
* When you sign into Anypoint Platform, the organization icon and name are displayed on the top right of the page. The master organization can contain multiple business groups. You can think of business groups as sub-organizations, or children of the master organization. Click the organization icon to navigate between business groups.

**Organization Owner**

The user who first signs up for an Anypoint Platform account is designated as the organization owner. This is not a role that is assigned, rather it is a super user identifier for this single user (creator of the Anypoint Platform account). The owner is assigned the Organization Administrators role by default. The organization administrator has every possible permission. OTher users can be assigned to the organization administrators role.

Every business group created within the organization hierarchy thereafter must have an owner assigned. Only users with the Organization Administrators role can be assigned as owners of business groups. Any organization administrator can assign and change owners of business groups. There cannot be more than one owner at once for a business group. The Organization Administrators role cannot be removed from organization owners.

An Anypoint Platform user who has Organization Administrator privileges can perform the following types of tasks:

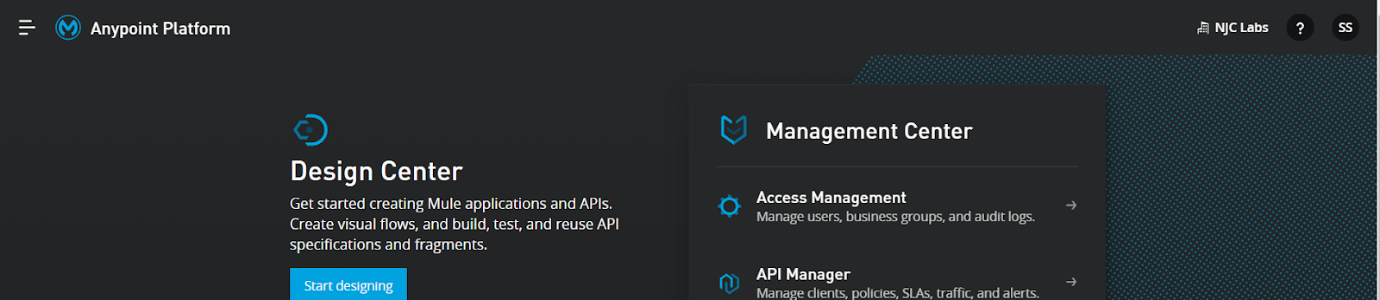
* Invite users to an organization
* Assign users to roles that define their permissions in the platform
* Edit and remove users from an organization
* Assign or change owners of business groups
* Configure organization settings
* View a client ID and client secret
* Access analytics for the APIs in your organization
* Create business groups to delegate management of the resources and define the scopes of roles and permissions
* Configure additional properties at the business group level.

**Organization Page**

* On the Organization page, you can:
* View a hierarchical tree of all of the organizations you have permissions to view. You can view and edit properties of an organization by clicking the name.
* Click the name of a group to view and edit its information. What you can view and edit depends on your role.
* Add and delete business groups (if enabled).

**Access the Organization page**

* Sign into your Anypoint Platform account as a user with the Organization Administrators role.
* Select the Access Management option from the main page, or click the menu button in the top left corner.



* The Organization page is displayed, where you can view organizations and business groups and add new business groups (if enabled).
* You can also click Organization in the left menu to view the Organization page.

**Client ID and Client Secret**

* Each master organization, environment, and business group within the master organization has its own associated unique client ID and client secret. These are used for authentication by users who are not organization administrators to access assets within an organization. The client ID and password are generated by Anypoint Platform for each environment you create, and they are globally unique.
* To deploy proxies or APIs to CloudHub, you must use these values to configure a customer-hosted Mule Runtime or legacy API Gateway.

**View** t**he Client ID and Secret for Organizations and Environments**

1. Sign into your Anypoint Platform account as a user with the Organization Administrators role.
2. Select the Access Management option from the main page, or click the menu button in the top left corner and click Access Management in the left navigation.
3. In the Access Management page, click Organization, or Environments in the left menu.
4. Click the name of the organization or environment for which to view the client ID and secret.

To change the client ID or client secret of an organization, you need to contact MuleSoft customer support representative.

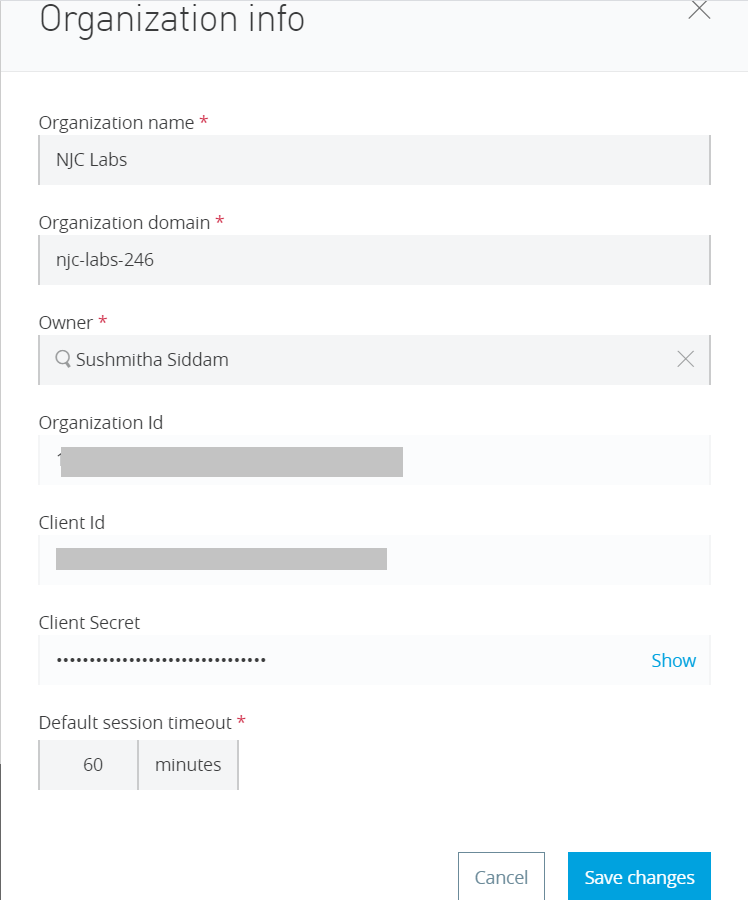
**Manage Master Organization Settings**

Only users with the Organization Administrators role can manage these settings.An organization administrator can modify the organization’s name, domain name, and session timeout for its users.

To access these settings:

1. In Access Management, click Organization in the left menu, then click the organization name.
2. Modify any of the following settings, then click Save changes.

* **Organization name**: This can be anything, for example, the name of the company.
* **Domain name**: Although multiple organizations can be created by different users using the same company name, each organization must have a unique domain name.
* **Owner**: The identifier of the organization owner.
* **Default session timeout**: Set the amount of time (in minutes) a user is inactive before they are automatically signed out of Anypoint Platform. The default is 60 minutes, the minimum is 15 minutes, and the maximum is 180 minutes.



You can also view the organization ID, client ID, and client secret. These values apply to the master organization and grant permissions for all of the business groups contained within.

**Sharing Resources in an Organization**

All API versions and CloudHub environments that you create in an organization are accessible only to users within the organization.

If you want to share resources with a user, you have to invite the user to join your organization, and the user must create a new account under the organization you sent the invitation from.

If your organization is configured to use an external federated identity system, you do not need to invite users, as they are authenticated by the external identity provider.

After a user joins your organization, they have access to the resources associated with the role(s) assigned to them. You can assign roles to grant users access to different resources within the organization. A best practice is to assign roles to the user at the time you invite them to join your organization so the roles are in effect when the user signs in for the first time.

If your organization contains business groups, you can give users access to multiple business groups by granting them roles within each group.

**Business Group**

Business groups are self-contained resource groups that contain Anypoint Platform resources such as APIs and applications. Business groups provide a way to separate and control access to Anypoint Platform resources, as users have access only to the business groups in which they have a role.

Business groups reside within the master organization and are organized in a hierarchical tree where the top-level business group is the root. Each business group you create has one direct parent and can have multiple children. The maximum number of business groups that can be created for an Anypoint Account is 100 including sub-business groups.

If you are an organization administrator and your organization is enabled for business group support, you can create business groups within your organization. The organization administrator is also the administrator of any business groups within the organization.

Since business groups are hierarchical, the owner of a parent business group always retains administrator permissions for any child business group within that parent. The owner of a business group can create child business groups and assign another user (who has the Organization Administrators role) as an owner of those child business groups. The owner of the parent group remains an Organization Administrator within the child business group.

Conversely, owners of child business groups cannot:

* Access or modify the parent business group or master organization settings
* View the parent business group’s client ID and client secret

Refer [Exercise](https://docs.google.com/document/d/1mcHZwHk5POskkvd57ri7m4PWsIRyqX8zPtxYe9Y4VIU/edit#heading=h.93uo0voap11t) for steps to Create a Business Group.

**Do Not Delete Business Groups**

It is not suggested to Delete Business Groups because once deleted it is not easy to recover it. If a business group is deleted, all the users, roles, permissions and environments created in the business group will be deleted.

However, if deemed necessary to delete a business group, only an organization administrator that belongs to a business group can delete it. The top-level (root) business group can’t be deleted, even by an organization administrator.

* In Anypoint Platform Access Management, in the left menu, click Organization.
* Click the name of the group to delete.
* In the Business Group Info dialog, click Delete Business Group.
* In the confirmation dialog, enter the name of the business, then click Delete.

**Resource Isolation**

Business groups provide more fine grained control over access to resources. It lets you delegate management of your Anypoint Platform resources including APIs, Runtime Manager applications, other business groups, users and roles. Each Business Group has its own set of permissions and roles. It provides complete isolation of resources and leads to multi-tenant use cases within an Anypoint Platform account.

When you create a business group under another business group, only the redistributable entitlements, such as VPCs and load balancers, that were assigned to the parent business group can be allocated to the child business group.

Resources like vCores are assigned to business groups are available only to that particular group and unavailable even to the parent organization.

Anypoint Master Organizations owns the resources for all its environments.

Parent Business Organizations owns resources for all of its environments.

**Each business group have their own user, roles and business group attributes(static IPs, vCores) and are available only in that particular business group.**

**Users**

Users are Anypoint Platform accounts with access to an organization.  You must have the Organization Administrators role to manage users within an Anypoint Platform organization. To manage user permissions for your API version, you must have API Version Owner permissions.

As an administrator of your organization, or of one of its business groups, you can enable, disable, or delete users. Access this menu by first making sure that the correct business group is selected in the top-right of the screen next to your user name, then clicking the gear icon next to it, and then picking the Users link in the left menu.

By ticking the checkbox next to a user, additional options related to that user are displayed:

**Enable**: Enables the user in the selected Organization / Business Group.

**Disable**: Disables the user in the selected Organization / Business Group. They will no longer be able to login.

**Delete**: When deleting from the top level Organization, all of the user’s roles/permissions are removed and the user becomes unattached. When deleting from a business group, all of the user’s roles/permissions for only the selected business group are removed.

To grant permissions or rules to users, click a username to access more information about that user, add roles and permissions to it, or reset its password.

By default, in every new organization and business group, you get three options:

* Assign API Permissions: Write the name of the API you want to give access to, then pick a version and permission.
* Assign Runtime Manager Permissions: Write the name of the Runtime Manager environment to give access to, then pick a permission.
* Assign Roles: Write the name of the role to grant. Check the roles section for a description of the default roles within an organization and business group.

Roles and permissions are grouped under organizations (and also, optionally, under business groups). This means that you can only assign roles and permissions that are related to resources that exist in the organization and/or business group that you are selecting.

If necessary, you can also remove user permissions, but keep in mind that if users have been granted their permission through an assigned role, you cannot view or remove those role-based permissions here

* + - * 1. Internal Users

To add users to an organization, as an organization administrator, or of one of its business groups, you can invite new users and manage existing users for your organization on the Access Management Administration page, which you can reach by selecting the Users option in your Access Management section.

* + - * 1. External Users

When you make an API portal public, users from any other Anypoint Platform organization can register client applications to call your API. When these users sign into your public developer portal, they are considered external users because they are outside of your organization.

When a user logs into Anypoint Platform for the first time, they are automatically added to the External Users tab. From the External Users tab in Access management, you can view a list of all external users. You can also enable or disable each of these external users from this screen. When you disable an external user, they can no longer sign into your public portal.

You cannot search for external users in other parts of Anypoint Platform because these users do not belong to your organization and do not have additional permissions. To grant users permission to perform tasks like deploying an application, you must invite the user to join your organization.

**External Identity Manager**

As the Anypoint Platform organization administrator, you can configure identity management in Anypoint Platform to set up users for single sign-on (SSO). Configure identity management using one of the following single sign-on standards:

* OpenID Connect: End user identity verification by an authorization server including SSO
* SAML 2.0: Web-based authorization including cross-domain SSO

**Roles**

To manage roles and permissions within Anypoint Platform, you must have the Organization Administrators role. To manage user permissions for an API version, you must have the API Versions Owner role.

A role is a set of pre-defined permissions for each product, or feature, within Anypoint Platform. Depending on the product, you can use pre-defined roles with standard permissions, or you can specify your own permissions for each role.

Anypoint Platform provides two types of roles:

* **Default roles**: Roles that are created automatically when an organization or business group is created. These roles provide permissions and access to core functionality of Anypoint Platform. You can assign users to default roles, but you can’t delete default roles.
* **Custom roles**: You can create and delete custom roles. You can assign users and add permissions to a custom role, as well as associate a custom role with specific Anypoint Platform products. For example, you can create a custom role called "Application Designer" and then grant Design Center permissions to the role so users assigned the Application Designer role can access Design Center.

Roles are business group specific, so ensure that you are in the correct business group for which to manage roles by clicking the menu next to your username on the top-right of the screen.

To access the Roles menu in Access Management:

1. Sign into your Anypoint Platform account as a user with the Organization Administrators role.
2. Select the Access Management option from the main page, or click the menu button in the top-left corner.
3. In the Access Management page, click Roles.

**Client Management**

Anypoint Platform acts as a client provider by default, but you can also configure external client providers to authorize client applications. As an API owner, you can apply an OAuth 2.0 policy to authorize client applications that try to access your API. You need an OAuth 2.0 provider to use an OAuth 2.0 policy.

**Auth2 policies:**

OAuth 2 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, GitHub, and DigitalOcean. It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account.

**Audit Logs**

Changes made by users within Anypoint Platform organizations are logged through an audit logging service. You can access the data logs through the audit logging query API or through the audit logging UI.

The audit logging service provides a queryable history of actions performed within the Anypoint Platform. It keeps track of all users who have interacted with objects in the system, and timestamps those actions. It also provides mechanisms for querying the set of users who have performed actions, the set of objects that had actions performed on them, and other endpoints that enable the querying of log entries.

Audit Logs can be accessed from UI of anypoint platform or from Audit logging Query API provided by Mulesoft. Audit logs can be accessed from Audit Logs in Access Management.

Audit Log Contents:

Each log entry has a set of properties that provides information about the activity:

o   **Time**: The timestamp when the activity occurred.

o   **Product**: The product where the object resides, for example, Access Management

o   **Type**: The type of the object on which the action is performed, for example, Organization

o   **Action**: The action associated with the object, for example, Create

o   **Object**: The name of the Object, for example, foo

o   **User**: The User who performed the action, for example, johndoe

o   **Environment**: Environment names for events from API Manager, Runtime Manager, CloudHub, Partner Manager, and MQ.

o  **Parent**: (Optional) The parent of the object (if any) on which the action is performed. Mainly relevant to APIs.

o   **Payload**: (Optional) More information about the log properties. For example, if an Organization was created, then the payload would contain information about the organization and the owner, such as IDs.

**Secure Configuration Properties**

Properties in Mule 4 can be encrypted to keep our sensitive data like ClientID and Client Password. Secure property module provided by mulesoft. We can be used to encrypt a .yaml or .properties file. Generally, in properties files, we store information like Client ID, Secret, UserId, UserPassword, Splunk Tokens, Oauth Token, AWS Keys, etc.

We need to encrypt the data inside any property files to restrict unauthorized access and to protect the data.

The first task in securing configuration properties is to create a YAML configuration file (.yaml) or a Spring-formatted Properties file (.properties), where you define the properties. The Mule Secure Configuration Properties Extension module enables you to configure YAML or Properties file types.

You can create secure configuration properties files either in src/main/resources in your Mule project, or by using absolute paths.

Supported Configuration file types are: **.yam and .properties** file

Supported Algorithms: AES(default), DES, Blowfish, RS2, RCA etc

Supported Modes: CBC, CFB, ECB, OFB

## *Creating Secure Properties Is Done in Three Steps*

Step 1: Create a configuration properties file.

Step 2: We can encrypt the whole file or encrypt individual property. For individual property, we can define secure properties in the file by enclosing the encrypted values between the sequence![value].

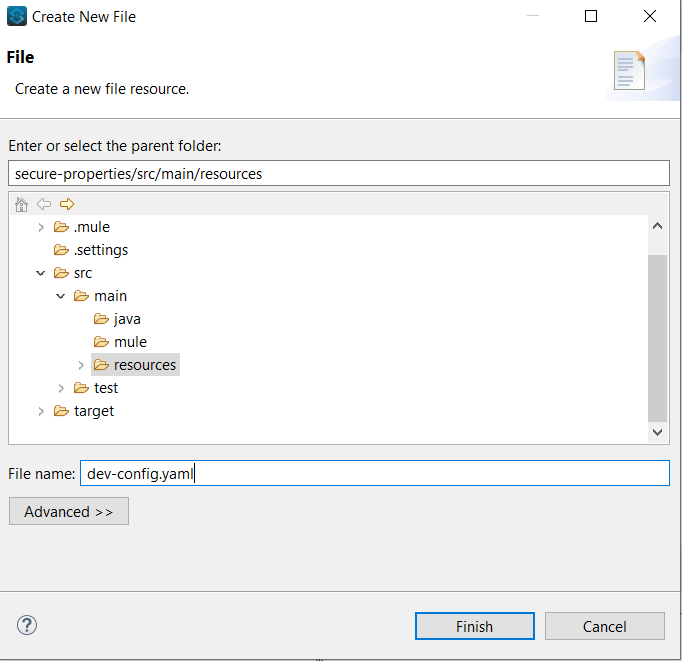
Step 3: Configure the file in the project with the Mule Secure Configuration Properties Extension module dependency. The file must point to or include the decryption key.

### *Create a Configuration Properties File*

The first task in securing configuration properties is to create a YAML configuration file (.yaml) or a Spring-formatted Properties file (.properties), where you define the properties in src/main/resources in your Mule project. The Mule Secure Configuration Properties extension module enables you to configure YAML or Properties file types.

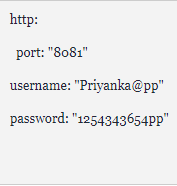
Refer to the below artifacts for more information.

Open Anypoint Studio -> Go to Project Folder -> src/main/resources ->Select Create New file(File extension can be either .yaml or .properties)

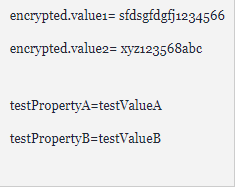


The following test.yaml, dev.properties  files contains non encrypted configuration properties values:

test.yaml (sample yaml file)



dev.properties (sample properties file)



***How to Define Secure Configuration Properties in The File***

**1. Adding the Premium Security Connector in AnyPoint Studio:**

\*\*\*Note: We cannot encrypt the YAML file this way as after the encryption process all the property key alignment will be rearranged. \*\*\*

#### 2. Encrypt Properties Using the Secure Properties Tool(jar)

Use the Secure Properties Tool to encrypt or decrypt text strings, values inside a properties file, or all the contents of a properties file.

We can run it in the command line as follows

–          Java command to encrypt text strings

Java -jar secure-properties-tool.jar <method> <operation> <algorithm> <mode> <key> <value>

–          Java command to encrypt all the content of a file

Java -jar secure-properties-tool.jar <method> <operation> <algorithm> <mode> <key> <input file> <output file>

Parameters Reference:

 <Method>

 – Sets the encryption or decryption method. Accepted values are string and file.

– String: Encrypts a single text string provided as parameter i.e. <value>

– File: Encrypts the value of each property set in the properties file provided as       parameter i.e. <input file>

<Operation>

–          Sets the operation to perform. Accepted values are

–          Encrypt: Sets the operation to encryption.

–          Decrypt: Sets the operation to decryption

<Algorithm>

–           Specifies the algorithm to use during the operation.

–          Supported Algorithms are AES (default), Blowfish, DES, DESede, RC2, and RCA

<Mode>

–           Specifies the mode to use during the operation.

–          Supported Modes are CBC (default), CFB, ECB, and OFB

<Key>

–          Specifies the word used to lock or unlock the file or value during the operation.

<Value>

–          Specifies the string value that is encrypted or decrypted during the string operation.

<Input file>

–          Specifies the properties file that is encrypted or decrypted during the file operation.

<Output file>

–          Specifies the encrypted or decrypted properties file that is created after the file or operation

Encrypt/ Decrypt Text Strings:

 Encrypt:

 Java -jar secure-properties-tool.jar string encrypt Blowfish CBC myKey password

Encrypted value is:  BJtXmtDAhpDskScZpbveIg==

Method: String

Operation: encrypt

Algorithm: Blowfish

Mode: CBC

Key: myKey

Value: password

Decrypt:

Java -jar secure-properties-tool.jar string decrypt Blowfish CBC myKey BJtXmtDAhpDskScZpbveIg==

Decrypted value is: password

Note: Here same key needs to be used for encryption and decryption operation.

Encrypt properties inside a File:

Consider the file config\_in.yaml with the following content:

User:

   Username: “MaxMule”

   Password: “Max@#@Mule”

If we run command:

Java -jar secure-properties-tool.jar file encrypt Blowfish CBC mulesoft config\_in.yaml config\_out.yaml

The tool generates the file config\_out.yaml with the following content:

User:

   Username: “![xHvoKxS+Ito=]”

   Password: “![4AOqzs6uHB5s7cPCej3b6w==]”

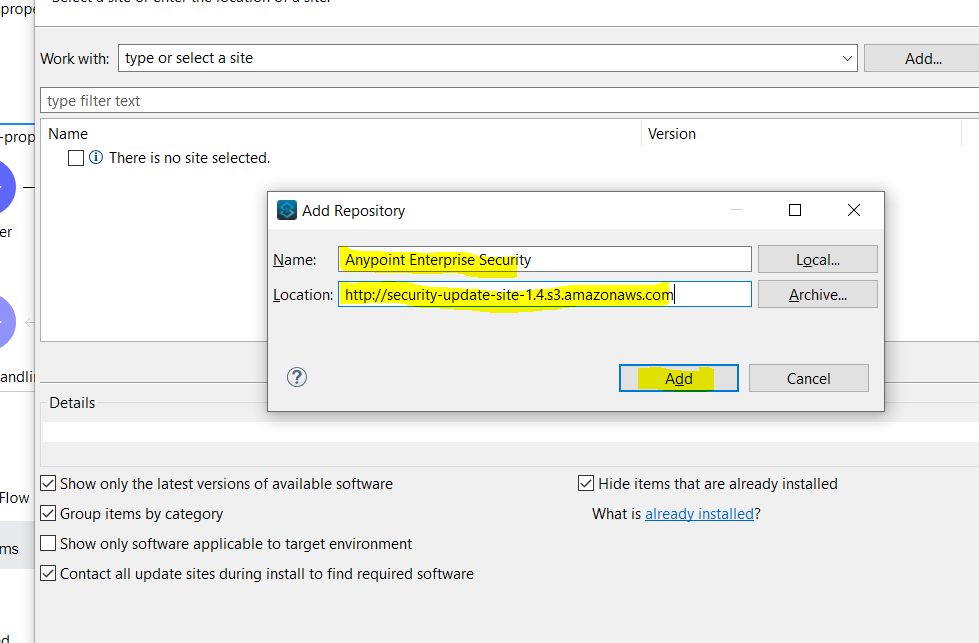
CloudHub - encryption\_key as a hidden value using mule-artifact.json

Steps to secure your properties and deploying to cloudhub with masking enabled

1. Initially start building an app with studio and decide the fields or file that you want to encrypt. We can encrypt our contents in 2 ways as suggested by MuleSoft and we can use configuration files like **.properties or .yaml** files.

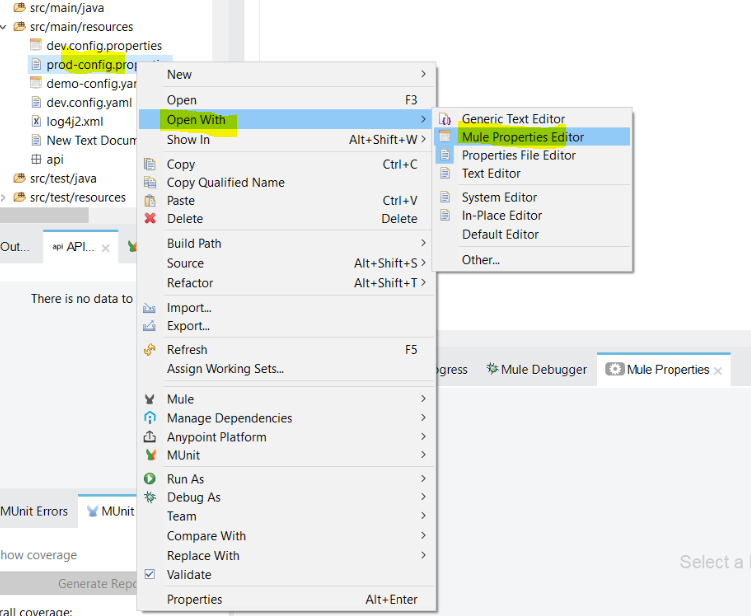
* By installing anypoint security suite
* By using the secure properties tool(JAR)

1. **By installing anypoint security suite**

**2.1 .**

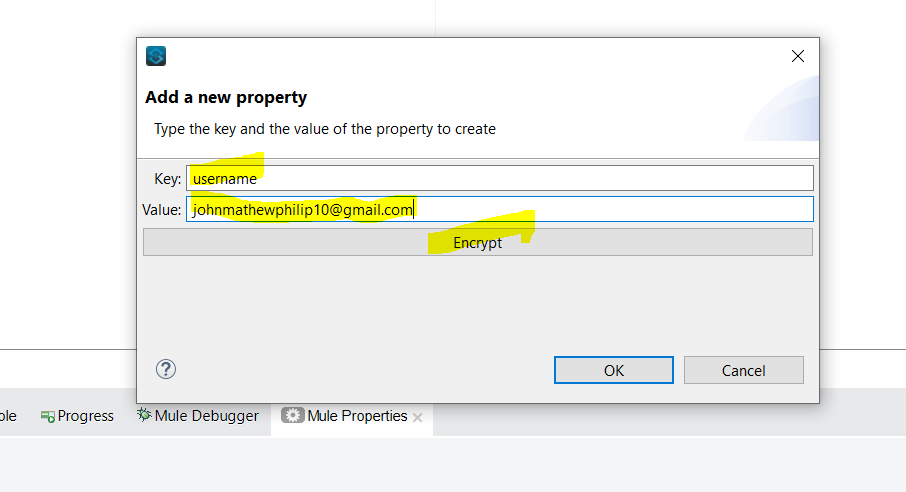
Anypoint security suite is not available by def initially in studio so we will have to manually download it. Go to help -> install new software and fill the fields as mentioned in the above diagram. Continue with the options and give consent for downloading and restarting studio.

2.1.2 Right click on the properties file (assuming that it is empty) that you want to add in the encrypted value along with their keys. Navigate to these options as given below.



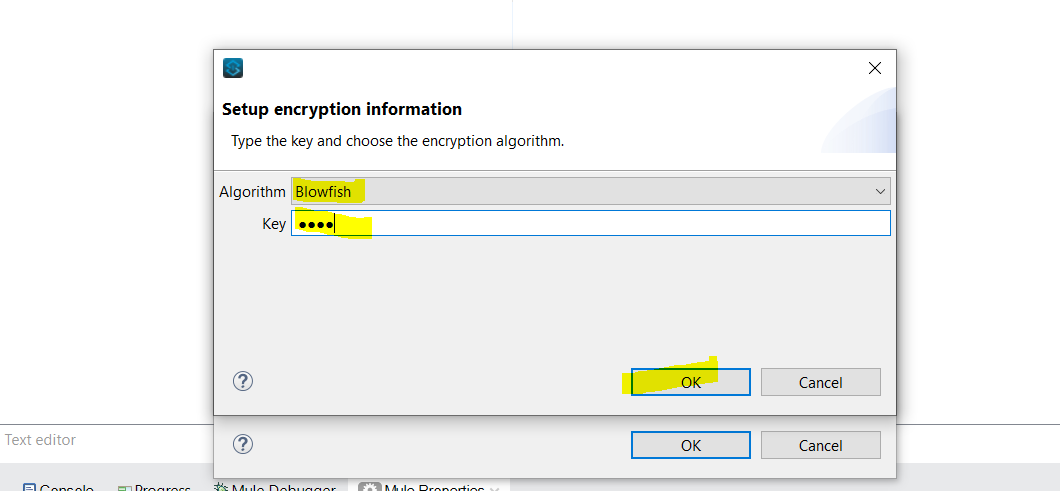
2.1.3  A new window will be opened. Look for a green “+” symbol on the top taskbar. If it is not green clink anywhere on the canvas to make it green. Click on this green cross button.

2.1.4. Start entering the keys and value that we want to encrypt in the popped up window and click on encrypt as given below.

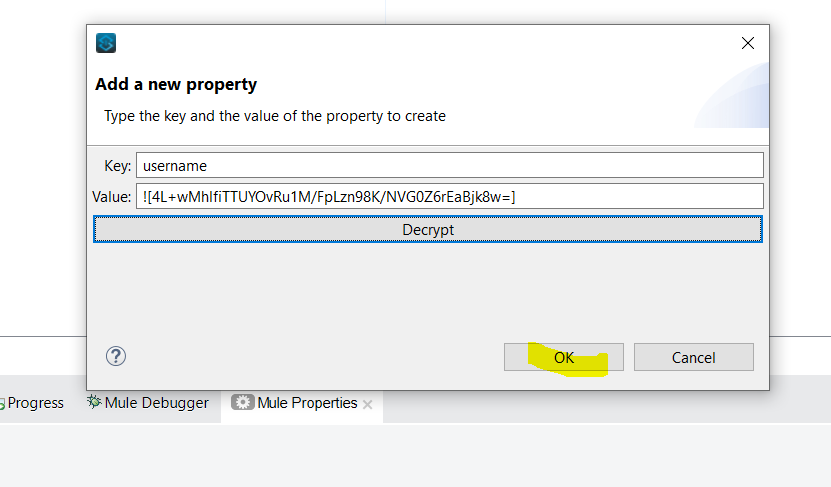


2.1.5 A new window will be popped asking for algorithm and key for encryption. Select the algorithm from the dropdown and give any desired key for encryption as given below.

note: The algorithm for encryption is AES by default and it requires a key size of minimum 16 characters.

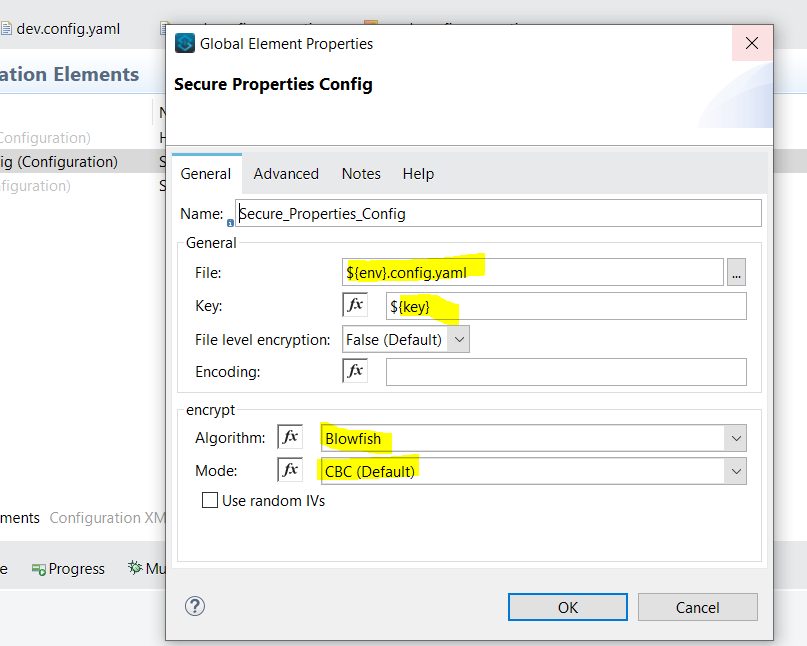


2.1.6.  This will give us the decrypted value and click on OK to add it to the already defined property file.

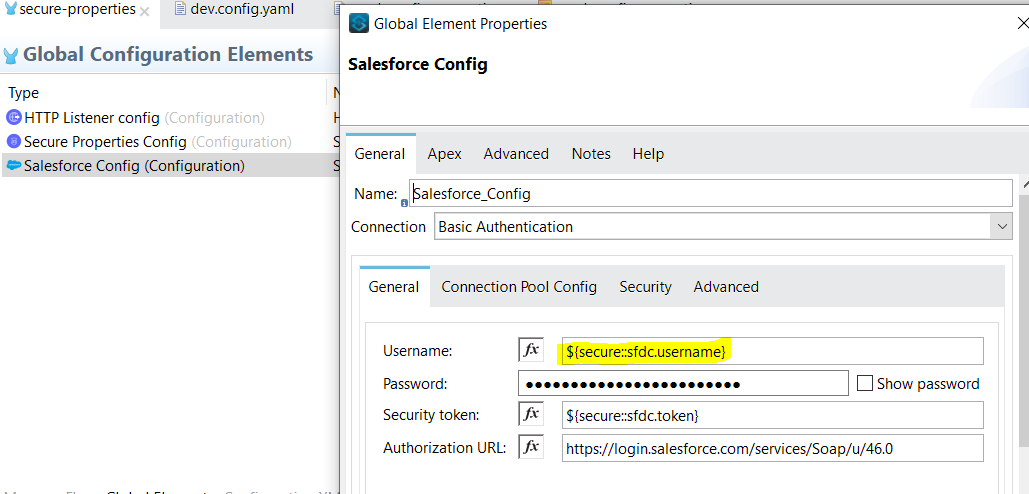


Likewise any no of keys and values can be encrypted in the same way using this method.

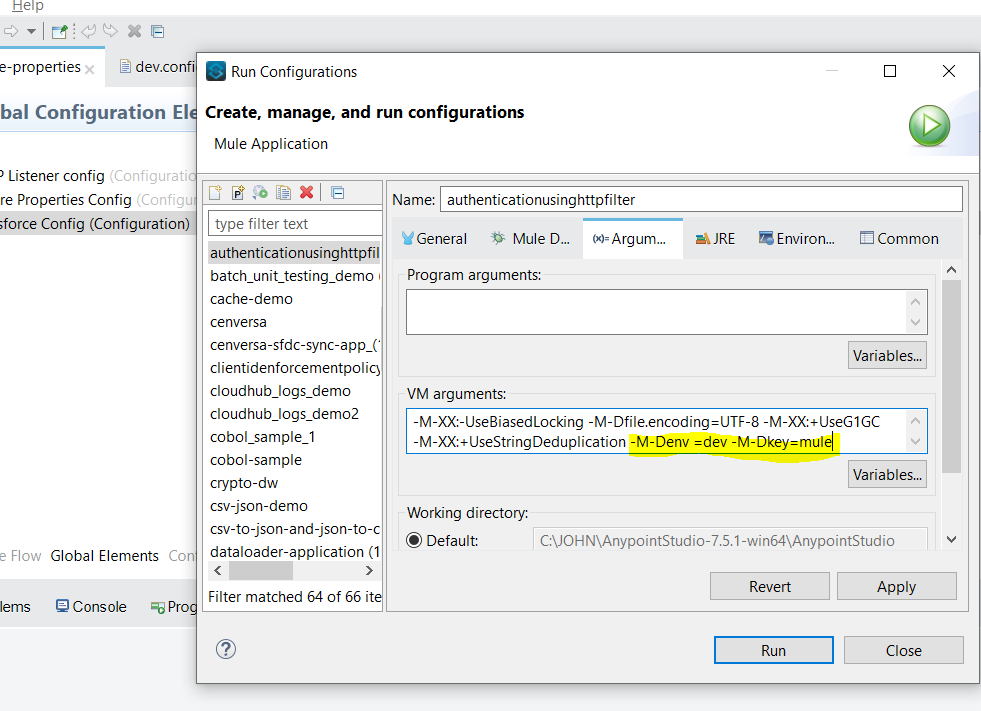
The next step is to create a secure configuration property. By default this won’t be available and we will have to download it from exchange and then configure again. Here we have to specify the file name, a good practice is to always externalise values using placeholders so that they become dynamic. The value of env can be passed from run configurations as an argument during runtime. For key, we have to specify the key we mentioned earlier during encryption. On thing to seriously note is that anyone who has access to this key can easily decrypt our credentials with ease. So it is always recommended to externalise the key too and pass during runtime as mentioned above. Also we have to set the algorithm that was mentioned earlier by us and any mode from the drop down for supporting the algorithm.



These values can then be used in their respective places as mentioned below, a keyword secure along side double colon has to be used for access:

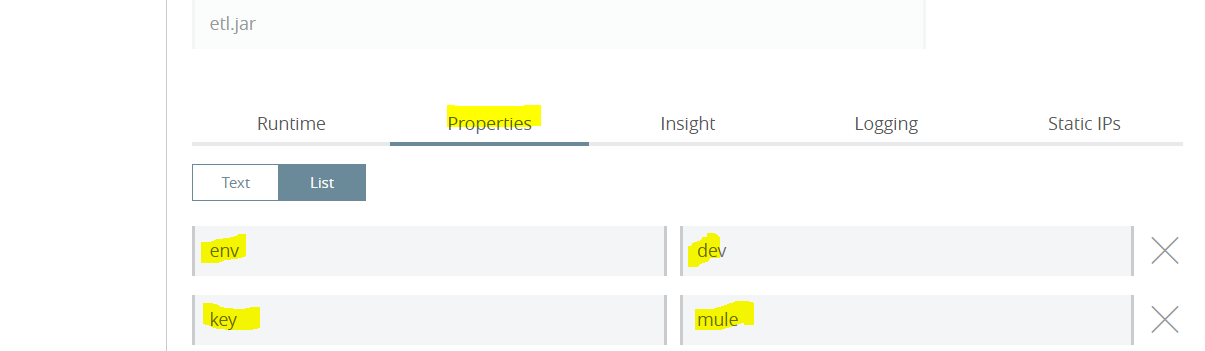


After this we are good to go. We can now run our application locally by passing the externalised fields via run configurations as arguments as mentioned below:



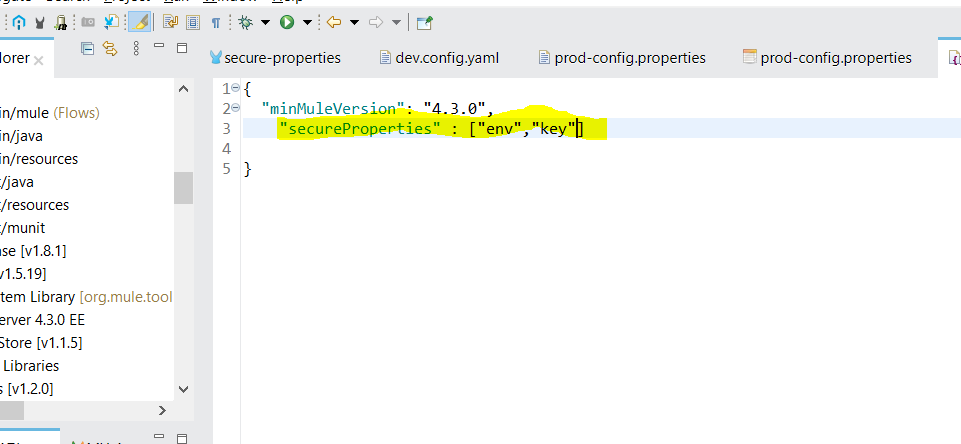
The mule runtime fetches these values alongside the key passed and internally decrypts and performs the designed operations.

2.1.12 Lets deploy these to cloudhub now. Right click the project and deploy it to cloudhub. In the popped up window go to properties and set up the arguments that we passed locally via run configurations. These values will be required as we are deploying it to cloudhub now unlike we were running it locally earlier.



2.1.13 Deploy it to cloudhub.

2.1.14 One thing to seriously note is that after successful deployment of our application on cloudhub if we goto runtime manager -> our deployed application -> settings -> properties. We can still see the key or whatever value we passed as an argument as an exposed open value. This should be again avoided to protect the resources. A simple tweak to do this is to go to our project’s artifact.json file and give commands and values as given below:



Give a key as secureProperties and give all the desired argument’s key that we need to hide in runtime enclosed in [] brackets.

2.1.15 Redeploy the application. Now we can see that the values that were earlier exposed is masked. (\*\*\*\*\*\*\*)

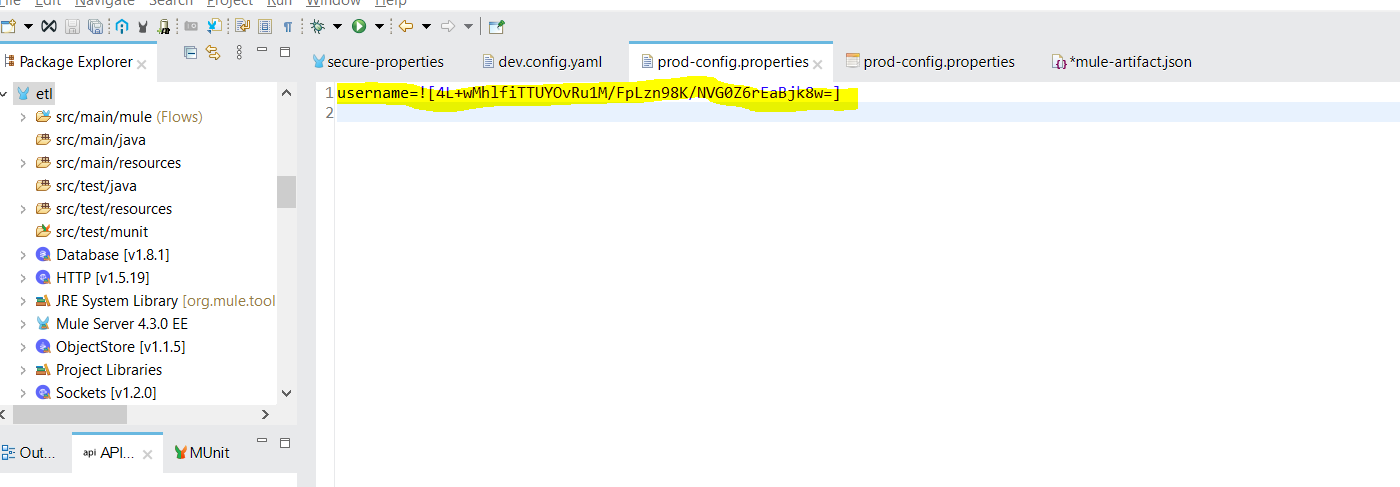
**Disadvantages** of this method of encryption:

* cannot be really used with .yaml files
* cannot encrypt a whole configuration file if one intend to do that

**Advantages**

Can encrypt the values by remaining in studio itself compared to the latter method(explained further down)

* Easy to use



Advantages:

* Can encrypt a whole configuration fie.

Disadvantages:

* Cannot use from within the studio.
* Little bit of manual effort is required.