Tenant Management Service

Contents

Copyright GE Digital

Tenant Management Service Overview

About the Tenant Management Service

In a multi-tenant environment, a tenant is an application or a group of users that share resources such as data, configuration, and user management. The tenants are logically isolated but physically integrated. That means even if the tenants use the same underlying resources, their data is isolated from each other. All users of a tenant have specific privileges to access the resources associated with that tenant. Each tenant can potentially use multiple service instances. The service instances are specific to the tenant. The Predix platform provides the Tenant Management service as a mechanism to provision service instances for a tenant.

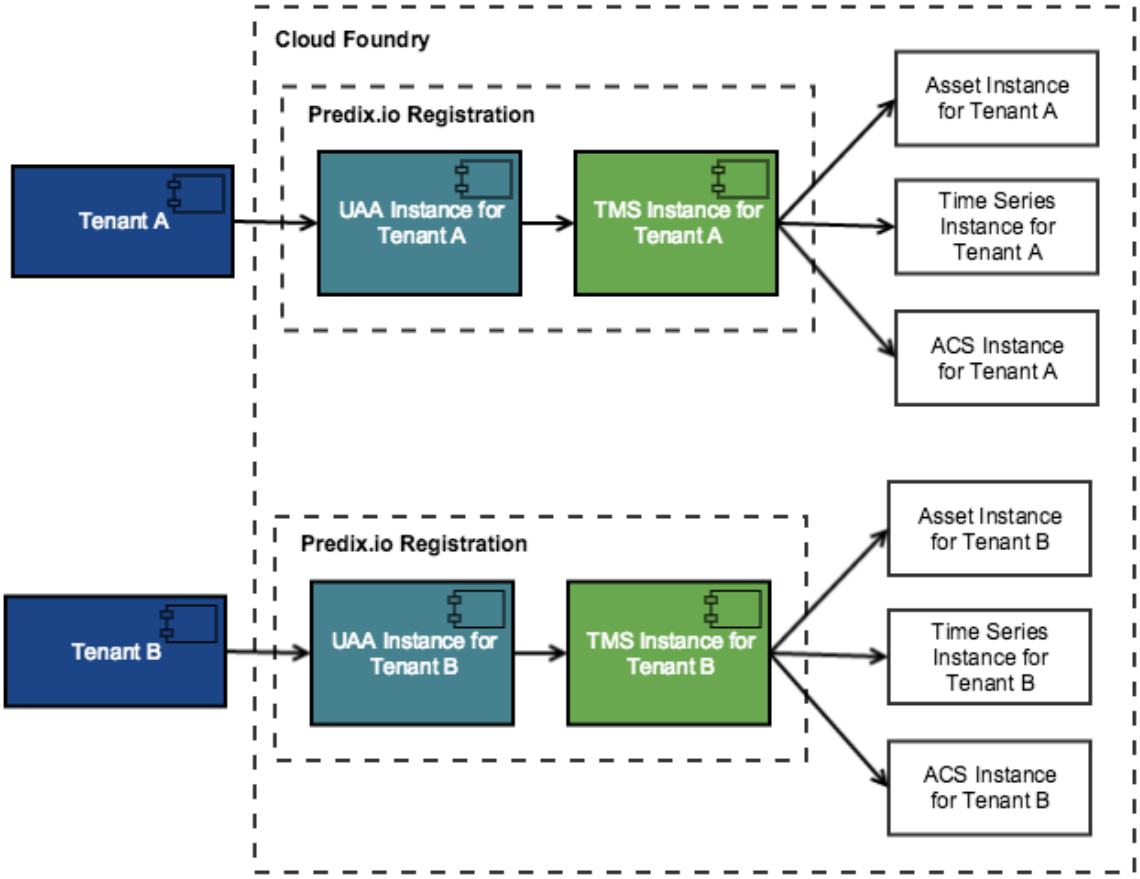
The Tenant Management service offers the following benefits:

- Provisioning of multiple service instances for a tenant. For example, if a tenant requires instances of the Access Control service, Asset service and Time Series service, you can use the Tenant Management service to provision these instances at the same time.
- Cleanup of instances if the tenant is deleted. If a tenant is no longer required, deleting the tenant also deletes the service instances related to the tenant.

  Note: The service instances are deleted only if they were created using the Tenant Management service for that tenant.
- Resolution of service instance credentials at runtime.
- The ability to store client credentials created by one tenant UAA, which clients can use to retrieve credentials (client ID and client secret) to access Predix services.

Tenant Management Service Architecture

The following figure shows the architecture of Tenant Management service:



When a tenant registers for an account on Predix.io, the registration process creates a UAA instance and an instance of Tenant Management service for the tenant. The tenant can then use the Tenant Management service to create other service instances.

Get Started With the Tenant Management Service

Creating a UAA Service Instance

You can create multiple instances of the UAA service in your space.

As a best practice, first delete any older unused instances before creating a new one.

1. Sign into your Predix account at https://www.predix.io.
2. Navigate to **Catalog** > **Services**, then click the **User Account and Authentication** tile.
3. Click **Subscribe** on the required plan.
4. Complete the fields on the **New Service Instance** page.

Field	Description
Org	Select your organization.
Space	Select the space for your application.
Service instance name	Enter a unique name for this UAA service instance.
Service plan	Select a plan.
Admin client secret	Enter a client secret (this is the admin password for this UAA instance). The client secret can be any alphanumeric string. **Note:** Record the client secret in a secure place for later use.
Subdomain	(Optional) Enter a subdomain you might need to use in addition to the domain created for UAA. You must not add special characters in the name of the subdomain. The value of subdomain is case-insensitive.

5. Click **Create Service**.

Your UAA instance is created with the following specifications:

- A client identifier (admin).

  Note: An admin client is required for bootstrap purposes. You can create additional clients to use with your application.
- A client secret (that you specified while creating the service).

To retrieve additional details of your instance, you can bind an application to your instance.

Using the Command Line to Create a UAA Service Instance

Optional procedure for using the command line instead of the graphical user interface to create a UAA service instance.

You can create up to 10 instances of UAA service in your space. If you need additional instances, you must delete an older unused instance and create a new one.

1. Use the Cloud Foundry CLI to log into Cloud Foundry.

```
cf login -a <API_Endpoint>
```

   Note: If you are a GE employee, you must use the cf login --sso command to log into Cloud Foundry. After you enter your SSO, you will receive a one-time passcode URL. Copy this URL and paste it in a browser to retrieve your one-time passcode. Use this code with the cf command to complete the CF login process.

   Depending on your Predix.io registration, the value of <API_Endpoint> is one of the following:

- Predix US-West
  https://api.system.aws-usw02-pr.ice.predix.io
- Predix US-East
  https://api.system.asv-pr.ice.predix.io
- Predix Europe
  https://api.system.aws-eu-central-1-pr.ice.predix.io

For example,

```
cf login -a https://api.system.aws-usw02-pr.ice.predix.io
```

2. List the services in the Cloud Foundry marketplace by entering the following command.

```
cf marketplace
```

The UAA service, predix-uaa, is listed as one of the available services.

3. Create a UAA instance by entering the following command.

```
cf create-service predix-uaa <plan> <my_uaa_instance> -c
'{"adminClientSecret":"<my_secret>","subdomain":"<my_subdomain>"}'
```

where:

- cf stands for the CLI command, cloud foundry
- cs stands for the CLI command create-service
- <plan> is the plan associated with a service. For example, you can use the tiered plan for the predix-uaa service.
- -c option is used to specify following additional parameters.
  - adminClientSecret specifies the client secret.
  - subdomain specifies a sub-domain you might need to use in addition to the domain created for UAA. This is an optional parameter. You must not add special characters in the name of the sub-domain. The value of sub-domain is case insensitive.

Note: Cloud Foundry CLI syntax can differ between Windows and Linux operating systems. See the Cloud Foundry help for the appropriate syntax for your operating system. For example, to see help for the create service command, run cf cs.

Your UAA instance is created with the following specification:

- A client identifier (admin).

  Note: An admin client is created for bootstrap purposes. You can create additional clients to use with your application.
- A client secret (that you specified while creating the service).

To retrieve additional details of your instance, you can bind an application to your instance.

Create a predix-uaa service instance with client secret as admin and sub-domain as ge-digital:

```
cf cs predix-uaa tiered test-1 -c '{"adminClientSecret":"admin","subdomain":"ge-digital"}'
```

This is how it appears in VCAP SERVICES when using the cf env <app_name> command:


```
"VCAP_SERVICES": {
"predix-uaa": [
  {
   "credentials": {
    "dashboardUrl": "https://uaa-dashboard.run.asv-pr.ice.predix.io/#/login/04187eb1-e0cf-4874-8218-9fb77a8b4ed9",
    "issuerId": "https://04187eb1-e0cf-4874-8218-9fb77a8b4ed9.predix-uaa.run.asv-pr.ice.predix.io/oauth/token",
    "subdomain": "04187eb1-e0cf-4874-8218-9fb77a8b4ed9",
    "uri": "https://04187eb1-e0cf-4874-8218-9fb77a8b4ed9.predix-uaa.run.asv-pr.ice.predix.io",
    "zone": {
     "http-header-name": "X-Identity-Zone-Id",
     "http-header-value": "04187eb1-e0cf-4874-8218-9fb77a8b4ed9"
    }
   },
   "label": "predix-uaa",
   "name": "testuaa",
   "plan": "Tiered",
   "provider": null,
   "syslog_drain_url": null,
   "tags": [],
   "volume_mounts": []
  }
 ],
```


Creating a Tenant Management Service Instance

1. Sign into your Predix account at https://www.predix.io.
2. Navigate to **Catalog** > **Security**, and click the **Tenant Management** tile.
3. Choose the plan, and click **Subscribe**.
4. On the New Service Instance page, enter:
5. Click **Create Service**.

Binding an Application to the Tenant Management Service Instance

You must bind your application to the Tenant Management service instance to provision its connection details in the VCAP_SERVICES environment variable. Cloud Foundry runtime uses then VCAP_SERVICES environment variable to communicate with a deployed application about its environment.

You can retrieve the following Tenant Management service instance details from the VCAP_SERVICES environment variable:

- A tenant_management_service_instance_uri for your instance.
- HTTP header information to access your Tenant Management service instance:

  | http-header-name as Predix-Zone-Id

| http-header-value

- An oauth-scope for your instance. The scope is required in the end-user token to access a specific Tenant Management service instance.

Note: The following steps are performed using the Cloud Foundry CLI. To complete the steps in a web browser, follow the instructions on the service page in the Predix Catalog.

1. Bind your application to the new Tenant Management service instance.

   ```
   cf bind-service <your_app_name> <my_tenant_management_instance>
   ```

   The <my_tenant_management_instance> instance is bound to your application, and the following message is returned:

   ```
   Binding service <my_tenant_management_instance> to app <your_app_name> in org predix-platform / space predix as userx@ge.com...
   OK
   TIP: Use 'cf restage' to ensure your env variable changes take effect
   ```

2. Verify the binding:

   ```
   cf env <your_app_name>
   ```

Creating an OAuth2 Client

You can create OAuth2 clients with specific permissions for your application to work with Predix Platform services. Often this is the first step after creating an instance of a service.

When you create an instance of UAA, the UAA Dashboard is available for configuring that instance of UAA. You can use the Client Management tab in the UAA Dashboard to create the OAuth2 clients.

If you are prefer using the UAA command-line interface (UAAC) instead of UAA Dashboard to create an OAuth2 client, see Using UAAC to Create an OAuth2 Client

1. In the Predix.io Console view, select the Space where your services are located.
2. In the Services Instances page, select the UAA instance to configure.
3. Select the **Configure Service Instance** option.
4. In the UAA Dashboard login page, specify your admin client secret and click **Login**.
5. In UAA Dashboard, select the **Client Management** tab.

   The Client Management tab has two views, **Clients** and **Services** . The **Services** view displays the service instances that you have created for your services.

   Note: The service instances displayed in the Services view were created while using the UAA that you are trying to configure. Service instances that you created using other UAA instances are not displayed on this page.
6. Click **Create Client** to open the **Create Client** form.
7. Complete the **Create Client** form.

Field	Description
Client ID	Specify a name for the OAuth2 client you are creating.
Authorized Grant Types	Choose one or more of the following grant types: • **authorization_code** When you use the authorization code grant type, the client directs the resource owner to UAA, which in turn directs the resource owner back to the client with the authorization code. • **client_credentials** When you use the client credentials grant type, the OAuth2 endpoint in UAA accepts the client ID and client secret and provides Access Tokens. • **password** When you use the resource owner password credentials grant type, the OAuth2 endpoint in UAA accepts the username and password and provides Access Tokens. • **refresh_token** The refresh tokens are credentials used to obtain access tokens. You can choose this option to obtain refresh token from UAA. You can then use the refresh token to obtain a new access token from UAA when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope. • **implicit** When you use the implicit grant type, UAA directly issues an Access Token to the client without authenticating the client. This reduces the number of round trips required to obtain an access token. For more information on grant types, see RFC 6749.
Client Secret	Specify the password. It is important that you keep a note of this password. If lost, this password cannot be retrieved.
Confirm Client Secret	Reenter the client secret.

Field	Description
Redirect URI	Specify a redirect URI to redirect the client after login or logout (for example, http://example-app.com/callback). Use this URI when you start using UAA as the service provider for your external Identity provider. UAA uses the value of Redirect URI for /oauth/authorize and /logout endpoints. You must specify a Redirect URI value if you use the Authorization Code or Implicit authorization grant type. When you use the Authorization Code grant type, the Redirect URI is your application's endpoint or callback that expects user authorization code. When you use the Implicit grant type, the Redirect URI is the end point where UAA sends the bearer token. Unique Resource Identifier consists of: • Access Protocol, http or https • Domain or IP address • Access Port such as 80 or 443 • Path If you have a specific URL for your application callback, you can use that to set the Redirect URI value for the related client. For example, https://your-app-domain.run.aws-usw02-pr.ice.predix.io/path1/path2/callback. You can specify multiple values for Redirect URI as a list of allowed destinations that UAA server can redirect the users. For example, https://yourappdomain1.run.aws-usw02-pr.ice.predix.io/path1/path2/callback, https://yourappdomain2.run.aws-usw02-pr.ice.predix.io/path1/path2/callback. If the subdomain of your application is dynamic, you can set the value of Redirect URI using wilcards. For example, https://*.your-app-domain.run.aws-usw02-pr.ice.predix.io/path1/path2/callback. **Note:** You must only use '*' for a domain that is exclusive to your application (Such as your-app-domain in example above). This prevents the redirect to be routed to an application that you do not own. You cannot use * in the top domain and sub domain (such as predix.io in the example above).
Scopes	Scopes are permissions associated with an OAuth Client to determine user access to a resource through an application. The user permissions are for authorization grant types authorization_code, password and implicit. By default, the admin client is assigned all required scopes. For a new client, an administrator can select the scopes to be added based on client requirements. For a list of available scopes, see Scopes Authorized by the UAA. To use an OAuth2 client for your Predix Platform service instance, you must update your OAuth2 client to add scopes that are specific to each service after adding the client to the service instance.
Authorities	Authorities are permissions associated with the OAuth Client when an application or API is acting on its own behalf to access a resource with its own credentials, without user involvement. The permissions are for the client_credentials authorization grant type. By default, the admin client is assigned all required authorities. For a new client, an administrator can select the authorities to be added based on client requirements. The list of authorities matches the list of scopes. For a list of available UAA scopes, see Scopes Authorized by the UAA. To use an OAuth2 client for your Predix Platform service instance, you must update your OAuth2 client to add authorities that are specific to each service after adding the client to the service instance. **Note:** An admin client is not assigned the default authority to change the user password. To change the user password, you must add the uaa.admin authority to your admin client.

Field	Description
Auto Approved Scopes	Specify scopes that can be approved automatically for the client without explicit approval from a resource owner.
Allowed Providers	Specifies the names of the external identity providers, if any. This field is required if you are using external identity providers with UAA as a service provider.
Access Token Validity	Specifies the access-token expiration time in ms.
Refresh Token Validity	Specifies the refresh-token expiration time in ms.

Updating the OAuth2 Client for Services on page 9 for your service specific information.

Updating the OAuth2 Client for Services

To use an OAuth2 client for secure access to your Predix Platform service instance from your application, you must update your OAuth2 client to add additional authorities or scopes that are specific to each service.

To enable your application to access a platform service, your JSON Web Token (JWT) must contain the scopes required for a platform service. For example, some of the scope required for Access Control service are acs.policies.read acs.policies.write.

The OAuth2 client uses an authorization grant to request an access token. Based on the type of authorization grant that you have used, you must update your OAuth2 client to generate the required JWT. For more information on how the OAuth2 client is created, see Creating OAuth2 client.

If you use the UAA Dashboard to create additional clients, the client is created for the default client_credentials grant type. Some required authorities and scopes are automatically added to the client. You must add additional authorities or scopes that are specific to each service.

In addition, the admin client is not assigned the default authority to change the user password. To change the user password, you must add the uaa.admin authority to your admin client.

Use the following procedure to update the OAuth2 client.

1. In the Console view, select the Space where your services are located.
2. In the Services Instances page, select the UAA instance to configure.
3. Select the **Configure Service Instance** option.
4. In the UAA Dashboard login page, specify your admin client secret and click **Login**.
5. In UAA Dashboard, select the **Client Management** tab.

   The Client Management tab has two views, **Clients** and **Services**. The **Services** view displays the service instances that you have created for your services.

   Note: The service instances displayed in the **Services** view are the instances that you created using the UAA that you are trying to configure. The service instances that you created using some other UAA instance are not displayed on this page.
6. Select the **Switch to Services View** option.
7. In the **Services** view, select the service that you need to update.
8. Choose an existing client or choose the **Create a new client** option. If you chose to create a new client, follow the steps in Creating an OAuth2 Client on page 6.
9. Click **Submit**.
10. Click on the **Switch to Clients View** option.
11. In the **Clients** view, click the edit icon corresponding to the client added in the previous step.
12. Complete the **Edit Client** form.

Field	Description
Authorized Grant Types	Choose one or more of the following grant types: • **authorization_code** When you use the authorization code grant type, the client directs the resource owner to UAA, which in turn directs the resource owner back to the client with the authorization code. • **client_credentials** When you use the client credentials grant type, the OAuth2 endpoint in UAA accepts the client ID and client secret and provides Access Tokens. • **password** When you use the resource owner password credentials grant type, the OAuth2 endpoint in UAA accepts the username and password and provides Access Tokens. • **refresh_token** The refresh tokens are credentials used to obtain access tokens. You can choose this option to obtain refresh token from UAA. You can then use the refresh token to obtain a new access token from UAA when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope. • **implicit** When you use the implicit grant type, UAA directly issues an Access Token to the client without authenticating the client. This reduces the number of round trips required to obtain an access token. For more information on grant types, see RFC 6749.
Redirect URI	Specify a redirect URI to redirect the client after login (for example, http://example-app.com/welcome). This URI is used when you start using UAA as service provider for your external Identify provider.
Scopes	By default, the client is assigned a few required scopes. For a new client, an administrator can select the scopes to be added based on the selected grant type. If you select the authorization_code, password and implicit grant type, you must update the scopes with service specific scopes. For a complete list of required scopes, see Authorities or Scopes Required for Platform Services. For a list of available UAA scopes, see Scopes Authorized by the UAA.
Authorities	By default, the client is assigned a few required authorities. For a new client, an administrator can select the authorities to be added based on the selected grant type. If you select the client_credentials grant type, you must update the authorities with service specific authorities. For a complete list of scopes to be added for each service, see Authorities or Scopes Required for Platform Services. For a list of available UAA authorities, see Scopes Authorized by the UAA.
Auto Approved Scopes	Specify scopes that can be approved automatically for the client without explicit approval from the resource owner.
Allowed Providers	Specify the names of the external identity providers, if any. This field is required if you are using external identity providers with UAA as a service provider.
Access Token Validity	Specifies the access token expiration time in ms.
Refresh Token Validity	Specifies the refresh token expiration time in ms.

You can complete the following additional tasks in UAA Dashboard:

- If you are using authorization grant type as Authorization Code, Implicit, or Resource Owner Password, you can manage users in UAA.
- You can create password policies for user passwords.

- You can set up external identity provider or use UAA as an identity provider. See Managing Identity Providers.

If you have completed your OAuth2 client setup, you can bind your application to your service instance.

Authorities or Scopes Required for Tenant Management Service

To enable applications to access the Tenant Management service, your JSON Web Token (JWT) must contain the following scopes:

- tms.tenant.read
  This is required for using the Tenant management service read APIs.
- tms.tenant.write
  This is required for using the Tenant Management service write APIs.
- predix-tms.zones.<tms_instance_guid>.user
  This value is generated in the VCAP_SERVICES environment variable as oauth-scope when you bind your application to your Tenant Management service instance.
- tms.tenant.credentials.admin
  This is required for using the Tenant Management service API for storing, updating, and deleting client credentials.
- tms.tenant.credentials.read
  This is required for authentication when retrieving client credentials.

The Oauth2 client uses an authorization grant to request an access token. OAuth2 defines four grant types. Based on the type of authorization grant that you have used, you must update your Oauth2 client to generate the required JWT. For more information on how the OAuth2 client is created, see Creating a UAA Service Instance on page 3.

Using Tenant Management Service

Creating a Tenant

Tenancy Management Service Provides REST APIs for creating a tenant and provisioning the services required for that tenant. For a tenant, you can either create new instances of the required services or bind the tenant to existing service instances.

To create new instances of the services, you must first obtain the required trusted issuer IDs. For more information on creating trusted issuers, see Creating a UAA Service Instance on page 3.

Use the following REST API to create a tenant.

```
HTTP POST
/tenant
```

For more information about this API, see the API Documentation.

For example, to create and bind to new instances of the ACS and Asset service, specify the following parameters in the REST API:

```
...
"services": [
  {
    "parameters": {},
    "seq": 0,
    "serviceName": "predix-acs",
    "servicePlan": "free",
    "trustedIssuerIds": [
      "https://13fa0384-9e2a-48e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io/oauth/
token"
    ]
  }

  {
    "parameters": {},
    "seq": 1,
    "serviceName": "predix-asset",
    "servicePlan": "free",
    "trustedIssuerIds": [
      "https://13fa0384-9e2a-48e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io/oauth/
token"
    ]
  }
]
...
```

To bind to existing instances of ACS and Asset service, specify the following parameters in the REST API:

```
...
"serviceInstances": [
  {
    "serviceInstanceName": "my-acs-instance",
    "serviceName": "predix-acs"
  }

  {
```

```
    "serviceInstanceName": "my-asset-instance",
    "serviceName": "predix-asset"
  }
]
...
```

Retrieving Service Instance Details

When you use the Tenant Management service to create and bind to service instances, you can use the REST APIs to retrieve the service instance information for each tenant service.

For more information about this API, see the API Documentation.

1. Use the following REST API to retrieve the details of all service instances for a tenant:

   ```
   HTTP GET
   /v1/tenant/{tenantName}
   ```

2. Use the following REST API to retrieve details of a specific service instance for a specific tenant:

   ```
   HTTP GET
   /v1/tenant/{tenantName}/service/{serviceName}
   ```

   Note:

   If you require the credentials of a specific service instance, use the following REST API:

   ```
   HTTP GET
   /v1/tenant/{tenantName}/service/{serviceName}/credentials
   ```

Updating a Tenant

You can update a tenant for adding additional services.

Use the following REST API to update a tenant:

```
HTTP PUT
/v1/tenant
```

For more information about this API, see the API Documentation.

For example, to create and bind to a new instances of the ACS service, specify the following parameters in the REST API:

```
...
    "services": [
      {
        "parameters": {},
        "seq": 0,
        "serviceInstanceName": "my_acs_instance",
        "serviceName": "predix-acs",
        "servicePlan": "basic",
        "trustedIssuerIds": [
                "https://13fa0384-9e2a-48e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io/
oauth/token"
              ]
```

```
      }

    ]
```

To bind to an existing instance of the ACS service, specify the following parameters in the REST API:

```
...
  "name": "string",
  "templateData": {
    "serviceInstances": [
      {
        "serviceInstanceName": "my_new_acs_instance",
        "serviceName": "predix-acs"
      }
    ],
}
```

Updating Services Provisioned Using Tenant Management Service

You can update services that you provisioned using Tenant Management service.

Use the following REST API to update a service:

```
HTTP PUT
/v1/tenant/{tenantName}/service/{instanceName}
```

For more information about this API, see the API Documentation.

Deleting a Tenant

You can use the Tenant Management service REST API to delete a tenant. When you delete a tenant, all service instances associated with that tenant are also deleted if they are not being used by another tenant.

Use the following REST API to delete a specific tenant:

```
HTTP DELETE
/v1/tenant/{tenantName}
```

Tenant Management Service Release Notes

Tenant Management Service

Q4 2016

New Features

Client Credentials Store

Tenant Management service provides REST APIs for storing, updating, retrieving, and deleting client credentials.

Q2 2016

- Added ability to update a tenant
  You can now update a tenant to add additional services. You can bind to existing instance of services or create and bind to new instances.
- Added ability to update the services that you provisioned using Tenant Management service
  A new API is now available to update the services that you provisioned using Tenant Management service.