Acronis frends

Frends-Acronis Integration

Introduction

This document will describe how to use Frends iPaaS to integrate with Acronis Cyber Cloud.

Once you are set up you will be able to perform the following operations:

- Create a Customer Tenant
- Get a list of Customer Tenants
- Get a list of Applications available to the Customer Tenant
- Enable an Application for the Customer Tenant
- Create a User and activate them by password or email
- Assign Access Policies to the User
- Get a list of Users for a Customer Tenant

All this functionality is available from within the Frends iPaaS platform without any need to access the Acronis Cyber Protection web interfaces.

Glossary

Frends iPaaS – the Frends integration software
Frends Tenant – this is your instance of Frends iPaaS
Frends Agent – this is the remote integration runtime system
Frends Agent Group – a group of Frends Agents
Frends Processes and Subprocesses – these are the low-code components that perform the integrations.

Prerequisites

The Acronis Partner/Managed Service Provider needs to sign up for Frends iPaaS in order to have a Frends Tenant Provisioned for them.

Acronis accounts – T.B.D

How the integrations work

The integrations are built on the Frends iPaaS platform using BPMN 2.0 Frends Low-Code components which access the Acronis Cyber Protection APIs.

During provisioning of the Frends Tenant a customer will receive a ready-made set of Frends processes and subprocesses which will perform some of the more common operations within the Acronis Cyber Protection suite. There are some partner specific parameters which needs to be set up, i.e. Client Id, prior to be able to use these integrations.

Once these variables are set up the partner will be able to create an Acronis Customer Tenant, enable Acronis Applications for that Customer Tenant and create Users for the Customer Tenant and other operations.

A partner can easily build more processes utilising the prebuilt subprocesses or utilising the Acronis Cyber Protection APIs along with the existing solutions to use as an example.

The Available Prebuilt Integrations

Create an Acronis Customer Tenant

The ready-made process 1.1. Acronis Create Customer Tenant will perform this operation. The following parameters will need to be provided:

Parameters:

Name - name for the Tenant i.e. Frends Example A
Type - type for the Tenant i.e. customer, partner etc
InternalTag - tag for internal use
ContactAddress - Contact Address for the Tenant - Json format i.e

```
{
  "email": "user1@frends.com",
  "address1": "street 1",
  "address2": "street 2",
  "country": "Finland",
  "state": "Uusimaa",
  "city": "Espoo",
  "zipcode": "02700",
  "phone": "0+358407788990",
  "firstname": "Joe",
  "lastname": "Blogs",
  "title": "Mr",
  "website": "www.frends.com",
  "industry": "IT",
  "organization_size": "Large",
  "email_confirmed": true,
  "aan": null,
  "fax": null,
  "language": "en"
}
```

Language - prefered language of the Tenant

This process will perform the necessary Authentication based on the Client Id and Secret variables set up during the provision of the Frends Tenant. It will then call the Acronis Cyber Protection API to create the Customer Tenant based on the data provided in the parameters.

Enable Applications

The process available for this operation - 1.3. Acronis Enable Applications. The process takes one parameter:

CustomerTenantName - the Name of the Customer Tenant for which the Applications will be enabled

The process will perform Authentication, retrieve a list of the Applications available to the Customer Tenant and then enable each Application in turn.

This process will enable all the Applications for a Customer Tenant, but can easily be changed to enable just one Application, or a new process can be created based on this process to do one Application at a time.

Create A Customer Tenant User

The process to create a Customer Tenant User is 1.2. Acronis Create Tenant User, it takes the following parameters:

logIn - Login for the User
contact - Contact Address for the User- Json format i.e

```
{
        "email": "user1@frends.com",
        "address1": "street 1",
        "address2": "street 2",
        "country": "Finland",
        "state": "Uusimaa",
        "city": "Espoo",
        "zipcode": "02700",
        "phone": "0+358407788990",
        "firstname": "Joe",
        "lastname": "Blogs",
        "title": "Mr",
        "website": "www.frends.com",
        "industry": "IT",
        "organization_size": "Large",
        "email_confirmed": true,
        "aan": null,
        "fax": null,
        "language": "en"
}
```

enabled - Indicates whether the User is enabled or disabled
language - Preferred language for the User
businessType - An array of Business type for the User, i.e.

```
[
   "buyer"
]
```

notifications - An array of Notifications applicable to the User, i.e.

```
[  "maintenance",
   "quota",
   "reports",
   "backup_error",
   "backup_warning",
   "backup_info",
   "backup_daily_report",
   "backup_critical",
   "device_control_warning",
   "certificate_management_error",
   "certificate_management_warning",
   "certificate_management_info"
]
```

CustomerTenantName - Name of the Customer Tenant for which the User is to be created

Again Authentication is performed, then the user is created using the data provided in the parameters.

Activating Users

This can be done either by setting a password for the User or by sending an activation email to the User. There are two prebuilt processes for these operations - 1.7. Acronis Activate Users By Password and 1.8. Acronis Activate Users By Email. Both of these processes take CustomerTenantName as a parameter and will Activate all the user for the Customer Tenant.

Again this process can easily be changed to activate just one User, or a new process can be created based on this process to do one User at a time.

Assigning User Access Policies

This operation can be achieved with the process 1.4. Acronis Update User Roles. The following parameters are required:

userList - List of User logins and the roles for the Users, Json Format, i.e.

```
{
  "logins": [
        {
    "user": "user1",
               "roles": [
                       "backup_user"
                       ]
        },
        {
    "user": "user2",
               "roles": [
                       "pds_operator",
                       "pds_support"
                       ]
        },
        {
    "user": "user3",
               "roles": [
                       "backup_user"
                       ]
        }
        ]
}
```

CustomerTenantName - Name of the Customer Tenant that the User belongs to

The process will perform the necessary Authentication and then will assign the given roles for each User in the userList parameter.