IDPrime Virtual Evaluation Setup Guide

[Alexander Nicklas](mailto:alexander.nicklas@thalesgroup.com?subject=IDPrime%20Virtual%20Evaluation%20Setup%20Guide) / 2021-02-01

# Overview

This document should help you to circumvent the pitfalls during your first evaluation installation of **IDPrime Virtual**. The main challenge with this solution is the fact that it combines so many modules in one solution:

* **SafeNet Luna Network HSM** or **Data Protection on Demand (DPoD)**  
  🡪 currently DPoD is supported for test installations only
* **SafeNet Trusted Access (STA)** 🡪 MFA via OpenID Connect (OIDC)
* **CentOS Server**
  + **SafeNet IDPrime Virtual Server (IDPV Server)** 🡪 provided as Docker container
  + **MySQL** or **MariaDB** 🡪 IDPV Server configuration
* **Windows Client** with
  + **SafeNet Authentication Client (SAC)** or **SafeNet Minidriver**
  + **SafeNet IDPrime Virtual Client (IDPV Client)**

Sounds like a puzzle – feels (a bit) like a puzzle… ;-)

As a first step you might want to register on the [**IDPrime Virtual Demo**](https://idprimevirtual.demo.gemalto.com/) site that allows you to setup an IDPrime Virtual Client without needing to deploy your own server environment.

**NOTE:** This document should not replace the “IDPrime Virtual Solution Guide” that is part of the IDPV software package. However, it tries to provide a brief guidance concerning the steps required for a standard evaluation setup and might help to avoid general pitfalls.

## Versions

This document is based on the experience using the following software versions:

* CentOS 7.9
  + Docker 20.10.1
  + MariaDB 10.5.8
* IDPrime Virtual Server 2.1 ([KB0023000](https://supportportal.gemalto.com/csm?id=kb_article_view&sys_kb_id=62d11890dbff949080b23452399619e9&sysparm_article=KB0023000))  
  🡪 There is a Full version as well as a Trial version (with 50 licenses)
* IDPrime Virtual Client 2.0.1 (also under [KB0023000](https://supportportal.gemalto.com/csm?id=kb_article_view&sys_kb_id=62d11890dbff949080b23452399619e9&sysparm_article=KB0023000))
* SAC 10.8 R2 for Windows ([KB0022544](https://supportportal.gemalto.com/csm?id=kb_article_view&sys_kb_id=0e29e77ddb1ed01091a974233996196d&sysparm_article=KB0022544))

# Prerequisites

You have to prepare some components before you are able to install the **IDPrime Virtual Server**.

## Docker

IDPrime Virtual Server is provided as a Docker image. To install and run the latest release of the Docker software you can follow the documentation on the [Docker web site](https://docs.docker.com/engine/install/centos/). The cleanest way would be to add the official repo using the “yum-config-manager” which is part of the “yum-utils” and might have to be installed first:

|  |
| --- |
| # yum install -y yum-utils  # yum-config-manager --add-repo https://download.docker.com/linux/centos/ docker-ce.repo  # yum install docker-ce docker-ce-cli containerd.io |

After the installation of Docker you have to start the service:

|  |
| --- |
| # systemctl start docker  # docker info |

Use the following command to get further information on the Docker “bridge” network which will help you identify the container IPs later on:

|  |
| --- |
| # docker network inspect bridge |

## MariaDB

IDPrime Virtual Server stores all its configuration information in a database. MariaDB is a fork of the MySQL database under full GPL-2 license. In this scenario it is installed on the same CentOS server as the Docker environment.

### Database Installation

To install the latest version on a CentOS 7 server you can follow the description on the [MariaDB website](https://mariadb.com/resources/blog/installing-mariadb-10-on-centos-7-rhel-7/) as the CentOS repository only contains MariaDB 5.5. You can use the following commands to install the latest 10.x version:

|  |
| --- |
| # wget https://downloads.mariadb.com/MariaDB/mariadb\_repo\_setup  # chmod +x mariadb\_repo\_setup  # ./mariadb\_repo\_setup  # yum install MariaDB-server |

### Create IDPV User

You have to **manually create the user account** for IDPrime Virtual Server. The following commands will create a user with the required access rights and network restrictions (given the default Docker “bridge” network is 172.17.0.0/16):

|  |
| --- |
| # mysql –p  MariaDB [(none)]> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, REFERENCES, INDEX, ALTER ON IDPrimeVirtualServer.\* TO 'idpvuser'@'172.17.%' IDENTIFIED BY '<db-password>';  MariaDB [(none)]> FLUSH PRIVILEGES; |

### INSTALLATION PARAMETERS (MariaDB)

For the installation of **IDPrime Virtual Server** you need the following information:

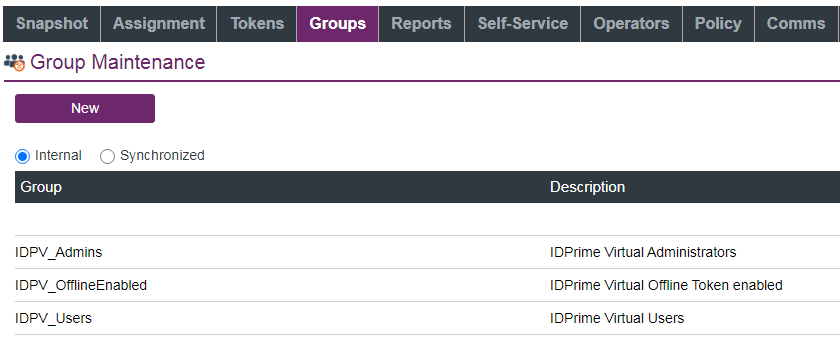
|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Notes** |
| **Database Server IP** | 172.17.0.1 | Standard Docker host address (on the “bridge” network) |
| **Database Server Port Number** | 3306 | Default port for MariaDB |
| **Database User** | idpvuser | Can be changed as required |
| **Database Password** | <db-password> | Set during database creation |

## STA

IDPrime Virtual relies on **multi-factor authentication via Open ID Connect (OIDC)**. For this you need to prepare a valid STA account to authenticate against. You also have to collect some parameters required later during the configuration process.

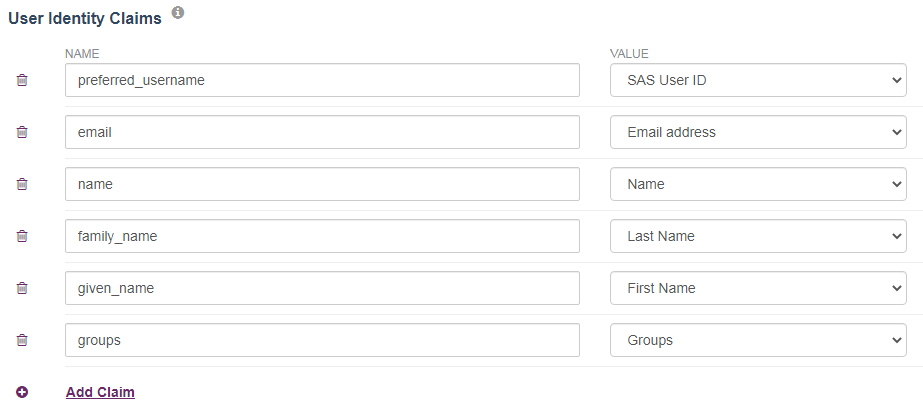
### Create IDPV Access Groups

Under **Groups 🡪 Group Maintenance** create three IDPV access groups:



### Create New OIDC Application

In the **STA Console** create a new application by following these steps:

* Go to the **Applications** tab by clicking 
* Click the  and search for the “**Generic Template**”
* Add this template and rename it as “**IDPrime Virtual**”
* Choose “OIDC” as **Integration Protocol**
* Leave the **Access Type** as “Confidential”
* Under **STA Setup** set the following parameters:
  + **ALLOWED FLOW TYPE:** “Authorization Code Flow”
  + **VALID REDIRECT URL:** <http://my.idpv.com> (any HTTP URL)
* Under **User Identity Claims** use “Add claim” to create a new one called “groups” and set the following mappings:  
  
* **Assign** the new IDPV application to the relevant user groups

### INSTALLATION PARAMETERS (STA)

**Part 1:** From the **Generic Template Setup** section of the STA application settings get the following information which is required later during the further configuration:

|  |  |
| --- | --- |
| **Parameter** | **Examples** |
| **CLIENT ID** | 18a6bb02-d311-4f32-a6d0-c65391acc13d |
| **CLIENT SECRET** | e340dae4-bcab-4fa5-b90a-525b188c79f9 |
| **VALID REDIRECT URL** | https://my.idpv.com |
| **AUTHORIZATION END POINT URL** | https://idp.eu.safenetid.com/auth/realms/D92SU4EJGP-STA/ protocol/openid-connect/auth |
| **ISSUER URL** (previous URL without the “/protocol/…” part) | https://idp.eu.safenetid.com/auth/realms/D92SU4EJGP-STA |

**Part 2:** Copy above “AUTHORIZATION END POINT URL”, paste it into your web browser, replace “…/auth” with “…/certs” and press enter. This will display a web page in JSON format.

**NOTE:** The JSON output is hardly readable so you can copy and paste the web page content into a JSON formatter, e.g. <https://jsonformatter.org/>. To keep the data local you could also use a plugin available for Notepad++ (e.g. JSTool).

From this page you need the following **STA public key parameters** that will be required later:

|  |  |
| --- | --- |
| **Parameter** | **Example** |
| **“n”**  (Key Modulus) | y1nA5wvYoTlIPyPxjO62soODuJms96CrNp9UqJIcr0ebY6seW1lGY1zcZ3qdHUtoCFWS0gD7RBdbWjRkHzQEH8s5dkPrTZrjmeQ6yhhKZ3pxwIhkosZBZvsImgExzc0Z1u0ziJwbMEpIH2jOiOh8-zBtb0xSmqpQ\_g0P3uctfXHptEIUEhri4tt6sPg6-LOfVXIEyN dWozjprXSWQ3iWjwO2dP5JSblucrta-ZnPLomQszalrb-Emzxcs8RKdIzq5jh9ZCne joe1fET8bwhPZx60BJiBs8Obdjs3OX4raGg04z\_2B61T\_vMZKIVyYVuO3m-wWt58 |
| **“e”**  (Key Exponent) | AQAB |
| **“kid”**  (Key ID) | ohB2F9\_d-4xAaQeKtBxmayRuC4PtkDthWliCrLrKJ-Q |

## Luna HSM / DPoD

You can either use **Data Protection on Demand (DPoD)** or a **SafeNet Luna Network HSM** partition as “Root of Trust” for the IDPrime Virtual Server.

**NOTE:** Currently **DPoD is only supported in test and evaluation scenarios** due to the fact that it doesn’t support Key Export mode. Therefore the size of the partition limits the number of virtual smart card keys that can be stored.

However, there will be a **new DPoD service tile with Key Export functionality** in Q1/2021.

For **Luna HSM 7** you need to know the following:

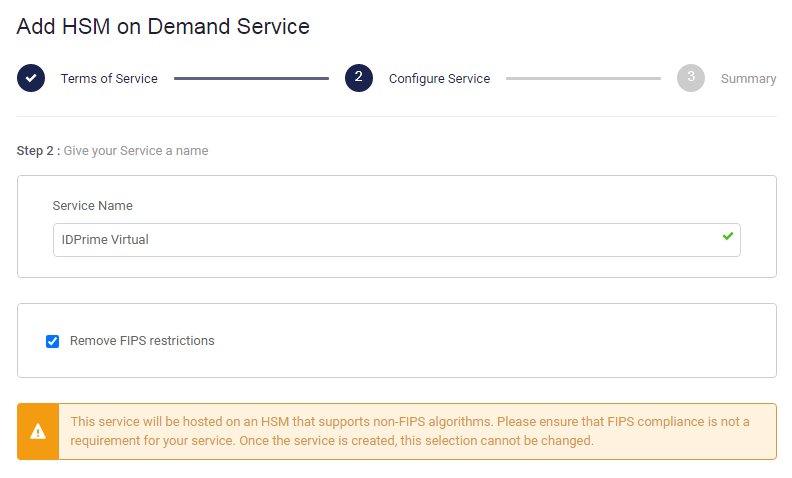
* You have to configure your IDPV partition in **Key Export mode**. To allow using this feature you need to be on firmware version 7.1 or higher.
* If the Luna is in **FIPS mode** you have to set the "RSAKeyGenMechRemap = 1" in "/var/thales/hsm/Chrystoki.conf"

### Luna HSM

[to be done]

### DPoD

Create a new “HSM on Demand” service with FIPS deactivated and download the client package:



**NOTE:** You have to make sure that you **select** **the “Remove FIPS restrictions” option** as FIPS mode is currently not supported with DPoD due to a known limitation in the service.

First you have to **unpack the DPoD client** in your “dpod” folder and **deploy a copy** of these files in a new folder “/var/thales/hsm” which will later be used by your Docker container:

|  |
| --- |
| # unzip setup-dpod.zip  # tar xf cvclient-min.tar  # chmod a+x setenv  # mkdir –p /var/thales/hsm  # cp –r \* /var/thales/hsm/ |

**NOTE:** Keeping a copy of the DPoD files outside the “/var/thales/hsm” folder allows running the HSM tools independently from the Docker instance (due to different configuration paths in the “Chrystoki.conf” instances).

To **initialize the DPoD instance** you have to perform the following steps:

* **Set the environment variables** from within the “dpod” folder (NOT in the “/var/thales/hsm” folder):

|  |
| --- |
| # source ./setenv |

* Start “dpod/bin/64/lunacm” and **initialize the partition** as well as the Partition Security Officer (“po”) and Crypto Officer (“co”) roles:

|  |
| --- |
| lunacm:> partition init -label IDPrimeVirtual  lunacm:> role login -n po  lunacm:> role init -n co  lunacm:> role logout  lunacm:> role login -n co  lunacm:> role changepw -n co  lunacm:> role logout |

The full documentation of DPoD can be found on the [Thales Documentation Hub](https://thalesdocs.com/dpod/).

### INSTALLATION PARAMETERS (DPoD)

For the installation of **IDPrime Virtual Server** you need this information of your DPoD environment:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Example** | **Notes** |
| **Token Serial Number** | 1334054146809 | Listed after “lunacm” start |
| **Crypto Officer Password** | <CO-Password> | Password set in last step of partition init |

# IDPrime Virtual Server

The following sections describe the installation and configuration of the **IDPrime Virtual Server**.

## Overview

There are two **different versions** of IDPrime Virtual Server available:

* **Trial Version** – allows test and evaluation installations without additional licenses. However, this version is limited to 50 virtual smart cards
* **Full Version** – requires a dedicated license

The current version is available from the [Thales Support Portal](https://supportportal.gemalto.com/csm?id=csm_product&sys_id=16e029a94f0c3b48102400818110c725) (requires valid service account).

## Installation

Unzip IDPrime Virtual Server package and load the **Docker image** using this command:

|  |
| --- |
| # docker load –i virtual\_idprime\_server.tar.gz |

You can verify that the image was imported correctly:

|  |
| --- |
| # docker images |

## Configuration

There are several configuration files that have to be provided in your “/var/thales/config” folder on your Docker host. The configuration templates can be found in your “idpv/config” folder.

### appsetting.yml

The main **configuration parameters for IDPrime Virtual Server** are defined in this file. There are two different templates for HTTP and HTTPS:

|  |
| --- |
| # This is a yml file. Values are in Key: Value format. Values are not required to be put in qoutes single' or double "  **DatabaseConfig:**  **DatabaseProvider: MariaDB** # (Mandatory) Database provider name. List of supported databases are 'MySQL, MariaDB and MSSQL'  **ConnectionString: server=172.17.0.1;port=3306; User=idpvuser; Password=<db-password>; Database=IDPrimeVirtualServer;** # (Mandatory) Database connection string  **HSMConfig:**  **HSMProvider: Dpod** # (Mandatory) HSM provider name. Supported providers are 'Luna, Dpod ,KeySecure' . Note- Dpod and KeySecure do not support offline virtual token.  **TokenSerial: <token-serial>** # (Mandatory) HSM partition serial number. #Leave it as blank in case of Key Secure  **TokenPin: <co-password>** # (Mandatory) HSM crypto officer (co) pin. OR #In case of KeySecure the value must be in format user:password  **UserGroup:** #This is the user group name of Key Secure user mentioned in above parameter. i.e. TokenPin  **TokenPasscode:** # (Optional) This value is recommended for enhanced security. If you don't want to change pin then remove this example value. Please note that it can be used only when server is configured with single partition only.  # Additional passcode string value (any new value). This value will be used to change the above TokenPin by the IDPV server to take complete ownership on hsm partition.  # Once this value is set then the above hsm crypto officer pin will be changed and the hsm partition can be accessed by IDPV server only.  # Caution !! This is one time configuration value. Any modification or changes on this value is not allowed which may lead to lock the hsm partition.  **WebServerConfig:**  **ServerPublicUrl: http://<ip-or-hostname>** # (Mandatory) It is mandatory to provide IDPV server url (public/intranet) which is being accessible from client machines.  **TlsCertificateThumbprint:** # (Optional) Thumbprint is not required in case of HTTP url. However it is recommended to host IDPV server on Https url and to provide thumbprint value of Server TLS certificate.  **Kestrel:** # It is recommended to configure Https settings.  **EndPoints:**  **Http:**  **Url: http://\*:5000** |

### idp-configuration.json

The **IDP connection parameters** collected in the “STA” section of this document are defined in this configuration file:

|  |
| --- |
| {  **"IdpPublicKeyModulus":**"y1nA5wvYoTlIPyPxjO62soODuJms96CrNp9UqJIcr0ebY6seW1lGY1zcZ3qdHUtoCFWS0gD7RBdbWjRkHzQEH8s5dkPrTZrjmeQ6yhhKZ3pxwIhkosZBZvsImgExzc0Z1u0ziJwbMEpIH2jOiOh8-zBtb0xSmqpQ\_g0P3uctfXHptEIUE hri4tt6sPg6-LOfVXIEyNdWozjprXSWQ3iW6jwO2dP5JSblucrta-ZnPLomQszalrb-Emzxcs8RKdIzq5jh9ZCnejoe1fET8bgH aTwhPZxMD6Oi0BJiBs8Obdjs3OX4raGg04z\_2B61T\_vMZKIVyYVuO3m-wWt58",  **"IdpPublicKeyExponent":**"AQAB",  **"IdpKeyId":**"ohB2F9\_d-4xAaQeKtBxmayRuC4PtkDthWliCrLrKJ-Q",  **"IdpClientId":**"18a6bb02-d311-4f32-a6d0-c65391acc13d",  **"IdpIssuerUrl":**"https://idp.eu.safenetid.com/auth/realms/D92SU4EJGP-STA",  **"IdpRedirectUrl":**"http://my.idpv.com",  **"JwtExpiration":**"0000001e",  **"JwtGroupClaim":**"groups",  **"JwtUserClaim":**"preferred\_username",  **"JwtAdminWhiteList":**"",  **"IDPrimeVirtualAdmin":**"IDPV\_Admins",  **"IDPrimeVirtualUser":**"IDPV\_Users",  **"OfflineTokenEnabledGroup":**"IDPV\_OfflineEnabled"  } |

### policy-configuration.json

This file defines some **policy settings** for IDPrime Virtual:

|  |
| --- |
| {  **"UserPinPolicy":** {  "MaxRetries": 5,  "IsMustChange": false  },  **"AdminPinPolicy":** {  "MaxRetries": 5,  "IsMustChange": false  },  **"OfflineTokenPolicy":** {  "ValidityDurationInHours": 120,  "PrivateKeyExportLevel": "All"  }  } |

### log4net.config

This configuration file allows setting the **log levels** for different modules to “ERROR”, “WARN”, “INFO” or “DEBUG”.

## Running the Server

To **run the Docker instance** you have to execute the following command:

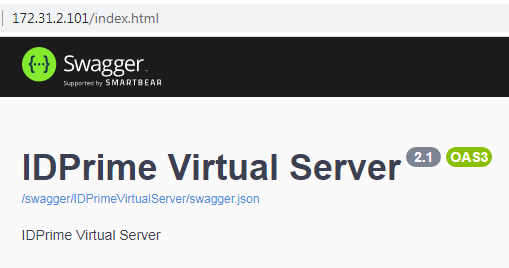
|  |
| --- |
| # docker run [-d] [--rm] --name idpv -it -v /var/thales/config/:/publish/Config/ -v /var/thales/hsm:/usr/local/hsm/ -p 80:5000 -p 443:5001 idprimevirtual\_server:2.1.0.132 |

The following **“docker run” command switches** might be helpful to understand:

* **-d** – This will “detach” the container from the bash console to run it in the background
  + Use “docker logs idpv” to check the console output of a detached container
  + Otherwise, the run-command will remain open to display messages on the console which might be helpful when running it the first time to immediately see if the server starts without errors
* **--rm** – This parameter will “remove” the container as soon as it is stopped or when it exits. Otherwise processes remain in the list of stopped containers (see "docker ps -a") where they may have to be deleted manually before being able to run a new instance.

See the [Appendix](#_Docker_Commands) for further comments on Docker commands and the Docker web site for the full documentation of “[docker run](https://docs.docker.com/engine/reference/run/)”.

To check if the server is running properly you can invoke the swagger interface from your web browser using the IP or hostname of your Docker host:



## Tenant Creation

After the initial configuration of the IDPrime Virtual Server you have to create your first tenant on the server instance. IDPV Server supports multiple tenants. Therefore you have to create separate IDP and policy configuration files for each tenant.

To start the “SetupTenant” script you have to open a “bash” shell within the container:

|  |
| --- |
| # docker exec -it idpv bash  # setuptenant/Thales.IDPrimeVirtual.SetupTenant create -i Config/idp-configuration.json -p Config/policy-configuration.json -a "<sta-client-secret>" |

There are further optional parameter for the “SetupTenant create” command:

* **-k [true | false]** – Defines if “Key Export” is allowed which is necessary for offline usage of a virtual card. Defaults depend on the “HSMProvider” set in the “appsettings.yml”:
  + “false” for “Dpod” and “KeySecure”
  + “true” for “Luna”

After successful execution of the script it will **display the tenant information** generated from the configuration files. You will find this information also in a file with the name “<TenantId>.txt” in the folder “/publish/Tenant/”.

You can also call the script with the “list” parameter to get all existing tenants:

|  |
| --- |
| # setuptenant/Thales.IDPrimeVirtual.SetupTenant list |

### INSTALLATION PARAMETERS (IDPV)

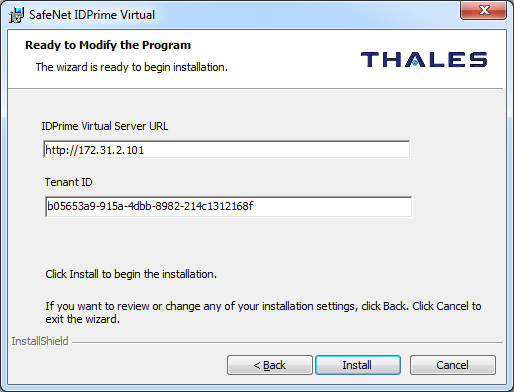
These parameters will be required for the following installation of the **IDPrime Virtual Client**:

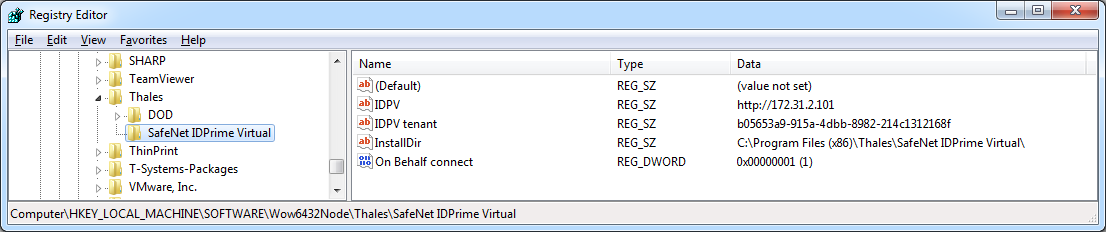
|  |  |  |
| --- | --- | --- |
| **Parameter** | **Example** | **Notes** |
| **TenantId** | e99e9003-bd9c-45ef-9097-88b7a417c7d4 | Random unique ID created by script |
| **IDPV Server URL** | http://172.31.2.101/ | External URL of IDPV container (i.e. the Docker host) |

# IDPrime Virtual Client

IDPrime Virtual is currently only working on Windows. On the client side you need two components:

* **SafeNet Authentication Client (SAC)**SAC is used to manage the content of the card as you would with any other regular smart card. IDPrime Virtual is supported by SAC 10.7 and later.
* ALTERNATIVELY: **SafeNet Minidriver**  
  It might be sufficient to have SafeNet Minidriver 10.7 (Post GA) or later installed on the client.
* **IDPrime Virtual Client (IDPV Client)**  
  This client is visible as a tray icon and allows you to connect to and disconnect from the IDPrime Virtual Server to make your virtual smart card visible in your operating system.

For the IDPV Client installation the **URL of IDPrime Virtual Server** and the **Tenant ID** are required:  


After the installation of the client you can find the **configuration settings** under the following Registry Key “HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Thales\SafeNet IDPrime Virtual”:

# WORKSHEET

This worksheets helps you to collect all the relevant parameters for your IDPrime Virtual installation.

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| **IDPrime Virtual Server** | |
| **MariaDB / MySQL** | |
| **Database Server IP** |  |
| **Database Server Port** | 3306 |
| **Database User (+Pwd)** | idpvuser |
| **STA** | |
| **CLIENT ID** |  |
| **CLIENT SECRET** |  |
| **VALID REDIRECT URL** |  |
| **AUTHORIZATION END POINT URL** |  |
| **ISSUER URL** |  |
| **“n”** (Key Modulus) |  |
| **“e”** (Key Exponent) | AQAB |
| **“kid”** (Key ID) |  |
| **IDPV Admin Group** | IDPV\_Admins |
| **IDPV User Group** | IDPV\_Users |
| **IDPV OfflineEnabled Group** | IDPV\_OfflineEnabled |
| **DPoD** | |
| **Token Serial Number** |  |
| **Crypto Officer Password** |  |
| **IDPrime Virtual Client** | |
| **TenantId** |  |
| **IDPV Server URL** |  |

# Appendix

Here are some further tips and tricks related to Linux and other topics.

## IDPV Troubleshooting

[TBD]

## Automatic Restart of IDPV Server

To make sure that IDPrime Virtual Server is working when the host is powered up you have to make sure that MariaDB and Docker services as well as the IDPV container are started automatically.

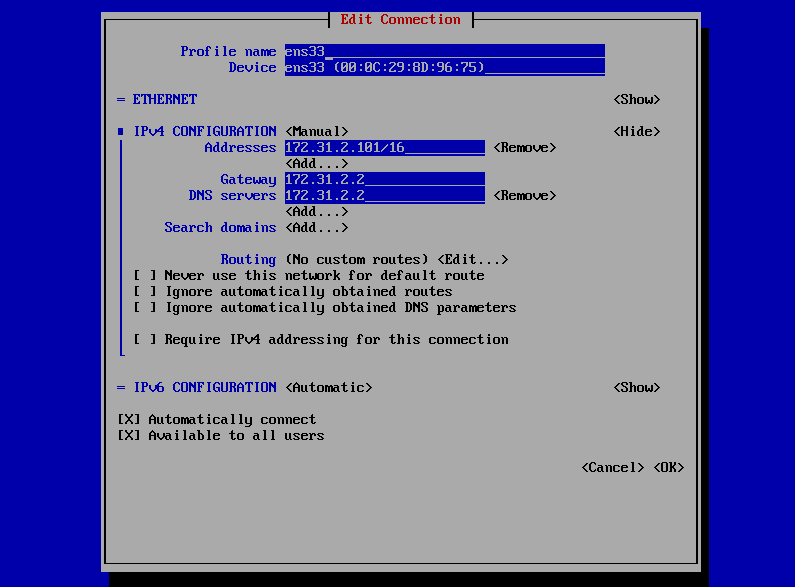
|  |
| --- |
| # systemctl enable MariaDB  # systemctl enable docker  # docker run --restart unless-stopped […] |

The option “**--restart unless-stopped**” will restart the container whenever the Docker service is started (e.g. after a reboot of the host machine) unless the container was intentionally stopped by using "docker stop idpv".

## CentOS Network Config

To review the current network settings you can use the “ip” and “nmcli” commands:

|  |
| --- |
| # ip address  # ip route  # nmcli |

The easiest way to configure the network interface in CentOS is to use the graphical tool “nmtui”:

|  |
| --- |
| # nmtui |

This will open up the graphical interface to configure all network related settings (see screenshot).

## MariaDB Commands

|  |  |
| --- | --- |
| List database users | SELECT user,host FROM mysql.user; |
| List user rights | SHOW GRANTS FOR idpvuser@127.0.0.1; |
| List databases | SHOW DATABASES; |
| Change user password | ALTER USER 'idpvuser'@'127.0.0.1' IDENTIFIED BY '<New\_Password>'; |
| Delete user | DROP USER 'idpvuser'@'127.0.0.1'; |
| Delete data base | DROP DATABASE IDPrimeVirtualServer; |
| Get current user | SELECT current\_user() ; |
| Get MariaDB version | > mysql -p |

## Docker Commands

|  |  |
| --- | --- |
| List stopped containers | > docker ps -a |
| Remove container | > docker rm <Container-ID> |
| Remove all stopped containers | > docker rm $(docker ps -aq) |
| Automatically restart the Docker container | > docker run --restart unless-stopped […] |

## Linux Commands

|  |  |  |
| --- | --- | --- |
| **Install Useful Tools** | | |
| Install “telnet” | > yum install telnet | Quickly check port connectivity |
| Install “unzip” | > yum install unzip | Unpack ZIP files |
| Install “nslookup” and “dig” | > yum install bind-utils | DNS queries |
| **Get Version Information** | | |
| Get CentOS version | > cat /etc/centos-release |  |