



Identity Cloud Guide

/ ForgeRock Identity Gateway 7

Latest update: 7.0.2

Joanne Henry

ForgeRock AS.
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Copyright © 2020-2021 ForgeRock AS.

Abstract

Instructions for configuring ForgeRock® Identity Gateway with the ForgeRock Identity Cloud.



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License.

To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

ForgeRock® and ForgeRock Identity Platform™ are trademarks of ForgeRock Inc. or its subsidiaries in the U.S. and in other countries. Trademarks are the property of their respective owners.

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

DejaVu Fonts

Bitstream Vera Fonts Copyright

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Arev Fonts Copyright

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the modifications to the Bitstream Vera Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: tavmjong @ free . fr.

FontAwesome Copyright

Copyright (c) 2017 by Dave Gandy, <https://fontawesome.com/>.

This Font Software is licensed under the SIL Open Font License, Version 1.1. See <https://opensource.org/licenses/OFL-1.1>.

Table of Contents

Preface	iv
About This Guide	iv
Example Installation for This Guide	iv
1. About Identity Gateway and the ForgeRock Identity Cloud	1
2. API Security With OAuth 2.0 and the ForgeRock Identity Cloud	2
3. Single Sign-On With OpenID Connect and the ForgeRock Identity Cloud	6
4. Cross-Domain Single Sign-On With the ForgeRock Identity Cloud	10

Preface

ForgeRock Identity Platform™ serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see <https://www.forgerock.com>.

About This Guide

This guide is for ForgeRock Identity Cloud evaluators, administrators, and architects. It provides examples of how to integrate your business application and APIs with ForgeRock Identity Cloud for Single Sign-On and API Security, with ForgeRock Identity Gateway.

Example Installation for This Guide

Unless otherwise stated, the examples in this guide assume the following installation:

- Identity Gateway installed on <http://openig.example.com:8080>, as described in "Downloading and Starting IG" in the *Getting Started Guide*.
- Sample application installed on <http://app.example.com:8081>, as described in "Downloading and Starting the Sample Application" in the *Getting Started Guide*.
- The ForgeRock Identity Cloud, with the default configuration, as described in the ForgeRock Identity Cloud Docs.

When you are using the ForgeRock Identity Cloud, you need to know the value of the following properties:

- The root URL of your ForgeRock Identity Cloud. For example, <https://myTenant.forgeblocks.com>.

The URL of the Access Management component of the ForgeRock Identity Cloud is the root URL of your ID Cloud followed by `/am`. For example, <https://myTenant.forgeblocks.com/am>.

- The realm where you work. The examples in this document use `alpha`.

Prefix each realm in the hierarchy with the `realms` keyword. For example, `/realms/root/realms/alpha`.

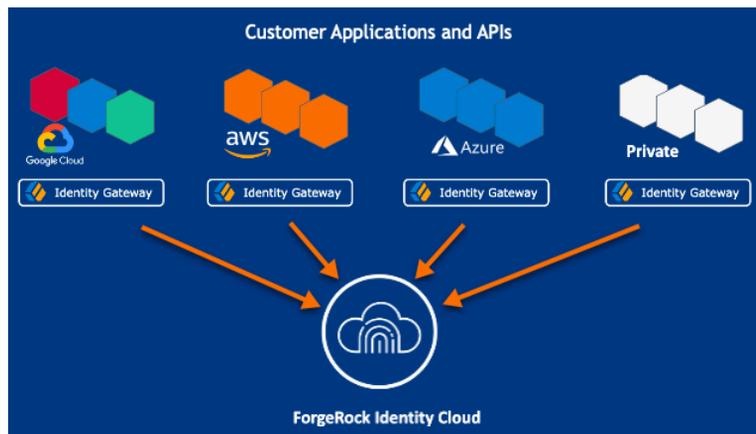
If you use a different configuration, substitute in the procedures accordingly.

Chapter 1

About Identity Gateway and the ForgeRock Identity Cloud

ForgeRock Identity Cloud simplifies the consumption of ForgeRock as an Identity Platform. However, many organizations have business web applications and APIs deployed across multiple clouds, or on-premise.

Identity Gateway facilitates non-intrusive integration of your web applications and APIs with the Identity Cloud, for SSO and API Security. The following image illustrates how Identity Gateway bridges your business to the ForgeRock Identity Cloud:



For information about the ForgeRock Identity Cloud, see the [ForgeRock Identity Cloud Docs](#).

Chapter 2

API Security With OAuth 2.0 and the ForgeRock Identity Cloud

This example sets up OAuth 2.0, using the standard introspection endpoint, where ForgeRock Identity Cloud is the authorization server, and Identity Gateway is the resource server.

For more information about Identity Gateway as an OAuth 2.0 resource server, see "Validating Access_Tokens Through the Introspection Endpoint" in the *Gateway Guide*.

Before you start, prepare Identity Cloud and Identity Gateway as described in "Example Installation for This Guide".

1. Set up Identity Cloud:
 - a. Log in to the ForgeRock Identity Cloud as an administrator.
 - b. In the platform console, go to Identities > Manage > Alpha realm - Users, and add a new user with the following values:
 - Username: `demo`
 - First name: `demo`
 - Last name: `user`
 - Email Address: `demo@example.com`
 - Password: `Ch4ng3!t`
 - c. Make sure that you are managing the `alpha` realm. If not, click the current realm at the top of the screen, and switch realm.
 - d. Add a web application:
 - i. In the platform console, click  Applications >  Add Application > Web, and add a web application with the following values:
 - Client ID: `oauth2-client`
 - Client Secret: `password`
 - ii. On the application page, add the following general settings:

- Grant Types: `Resource owner Password Credentials`
 - Scopes: `mail`
- e. Add an Identity Gateway agent:
- Click Gateways & Agents, and add an agent profile with the following values:
 - ID: `ig_agent`
 - Password: `password`

By default, the agent can introspect OAuth 2.0 tokens issued to any client, in the realm and subrealm where it is created. To change the introspection, click Native Consoles > Access Management, and update the agent in the AM console.

2. Set up Identity Gateway:

- a. Set an environment variable for the IG agent password, and then restart IG:

```
$ export AGENT_SECRET_ID='cGFzc3dvcmQ='
```

The password is retrieved by a `SystemAndEnvSecretStore`, and must be base64-encoded.

- b. Add the following route to IG, to serve `.css` and other static resources for the sample application:

Linux

```
$HOME/.openig/config/routes/static-resources.json
```

Windows

```
%appdata%\OpenIG\config\routes\static-resources.json
```

```
{
  "name" : "sampleapp_resources",
  "baseURI" : "http://app.example.com:8081",
  "condition" : "${matches(request.uri.path, '^/css')}",
  "handler": "ReverseProxyHandler"
}
```

- c. Add the following route to Identity Gateway, replacing the value for the property `amInstanceUrl`:

Linux

```
$HOME/.openig/config/routes/oauth2rs-idc.json
```

Windows

```
%appdata%\OpenIG\config\routes\oauth2rs-idc.json
```

```
{
  "name": "oauth2rs-idc",
```

```

"baseURI": "http://app.example.com:8081",
"condition": "${matches(request.uri.path, '^/oauth2rs-idx')}",
"properties": {
  "amInstanceUrl": "<myIdentityCloudUrl/am>"
},
"heap": [
  {
    "name": "SystemAndEnvSecretStore-1",
    "type": "SystemAndEnvSecretStore"
  },
  {
    "name": "AmService-1",
    "type": "AmService",
    "config": {
      "url": "&{amInstanceUrl}",
      "realm": "/alpha",
      "version": "7.1",
      "agent": {
        "username": "ig_agent",
        "passwordSecretId": "agent.secret.id"
      },
      "secretsProvider": "SystemAndEnvSecretStore-1"
    }
  }
],
"handler": {
  "type": "Chain",
  "config": {
    "filters": [
      {
        "name": "OAuth2ResourceServerFilter-1",
        "type": "OAuth2ResourceServerFilter",
        "config": {
          "scopes": [
            "mail"
          ],
          "requireHttps": false,
          "realm": "OpenIG",
          "accessTokenResolver": {
            "name": "TokenIntrospectionAccessTokenResolver-1",
            "type": "TokenIntrospectionAccessTokenResolver",
            "config": {
              "amService": "AmService-1",
              "providerHandler": {
                "type": "Chain",
                "config": {
                  "filters": [
                    {
                      "type": "HttpBasicAuthenticationClientFilter",
                      "config": {
                        "username": "ig_agent",
                        "passwordSecretId": "agent.secret.id",
                        "secretsProvider": "SystemAndEnvSecretStore-1"
                      }
                    }
                  ]
                }
              }
            }
          }
        }
      },
      {
        "name": "ForgeRockClientHandler",
        "type": "ForgeRockClientHandler"
      }
    ]
  }
}

```


Chapter 3

Single Sign-On With OpenID Connect and the ForgeRock Identity Cloud

This example sets up ForgeRock Identity Cloud as an OpenID Connect identity provider, and Identity Gateway as a relying party.

For more information about Identity Gateway and OpenID Connect, see "*Acting As an OpenID Connect Relying Party*" in the *Gateway Guide*.

Before you start, prepare Identity Cloud, Identity Gateway, and the sample application as described in "Example Installation for This Guide".

1. Set up Identity Cloud:
 - a. Log in to the ForgeRock Identity Cloud as an administrator.
 - b. In the platform console, go to Identities > Manage > Alpha realm - Users, and add a new user with the following values:
 - Username: `demo`
 - First name: `demo`
 - Last name: `user`
 - Email Address: `demo@example.com`
 - Password: `Ch4ng3!t`
 - c. Make sure that you are managing the `alpha` realm. If not, click the current realm at the top of the screen, and switch realm.
 - d. Add a web application:
 - i. In the platform console, click  Applications >  Add Application > Web, and add a web application with the following values:
 - Client ID: `oidc-client`
 - Client Secret: `password`
 - ii. In General Settings on the application page, add the following values:

- Sign-in URLs: `http://openig.example.com:8080/home/id_token/callback`
- Grant Types: `Authorization Code`, `Resource owner Password Credentials`
- Scopes: `openid`, `profile`, `mail`

iii. Click Show advanced settings > Authentication, and click Implied Consent:

The resource owner is not asked for consent during authorization flows.

2. Set up Identity Gateway:

a. Set an environment variable for the `oidc-client` password, and then restart IG:

```
$ export CLIENT_SECRET_ID='cGFzc3dvcmQ='
```

b. Add the following route to IG, to serve `.css` and other static resources for the sample application:

Linux

```
$HOME/.openig/config/routes/static-resources.json
```

Windows

```
%appdata%\OpenIG\config\routes\static-resources.json
```

```
{
  "name": "sampleapp_resources",
  "baseURI": "http://app.example.com:8081",
  "condition": "${matches(request.uri.path, '^/css')}",
  "handler": "ReverseProxyHandler"
}
```

c. Add the following route to Identity Gateway, replacing the value for the property `amInstanceUrl`:

Linux

```
$HOME/.openig/config/routes/oidc-idc.json
```

Windows

```
%appdata%\OpenIG\config\routes\oidc-idc.json
```

```
{
  "name": "oidc-idc",
  "baseURI": "http://app.example.com:8081",
  "condition": "${matches(request.uri.path, '^/home/id_token')}",
  "properties": {
    "amInstanceUrl": "<myIdentityCloudUrl/am>"
  },
  "heap": [
    {
      "name": "SystemAndEnvSecretStore-1",
      "type": "SystemAndEnvSecretStore"
    }
  ]
}
```

```

    }
  ],
  "handler": {
    "type": "Chain",
    "config": {
      "filters": [
        {
          "name": "OAuth2ClientFilter-1",
          "type": "OAuth2ClientFilter",
          "config": {
            "clientEndpoint": "/home/id_token",
            "failureHandler": {
              "type": "StaticResponseHandler",
              "config": {
                "status": 500,
                "headers": {
                  "Content-Type": [
                    "text/plain"
                  ]
                }
              },
              "entity": "Error in OAuth 2.0 setup."
            }
          }
        },
        {
          "name": "oauth2-client",
          "type": "ClientRegistration",
          "config": {
            "clientId": "oidc-client",
            "clientSecretId": "client.secret.id",
            "issuer": {
              "name": "Issuer",
              "type": "Issuer",
              "config": {
                "wellKnownEndpoint": "&{amInstanceUrl}/oauth2/realms/alpha/.well-known/
openid-configuration"
              }
            },
            "scopes": [
              "openid",
              "profile",
              "mail"
            ],
            "secretsProvider": "SystemAndEnvSecretStore-1",
            "tokenEndpointAuthMethod": "client_secret_basic"
          }
        }
      ],
      "requireHttps": false,
      "cacheExpiration": "disabled"
    }
  },
  "handler": "ReverseProxyHandler"
}
}
}
}

```

Notice the following features of the route compared to `07-openid.json` in "Use AM As a Single OpenID Connect Provider" in the *Gateway Guide*, where Access Management is running locally:

- The ClientRegistration `wellKnownEndpoint` points to the Identity Cloud.

3. Test the setup:

- a. Go to `http://openig.example.com:8080/home/id_token`. The Identity Cloud login page is displayed.
- b. Log in to Identity Cloud as user `demo`, password `Ch4ng3!t`. The home page of the sample application is displayed.

Chapter 4

Cross-Domain Single Sign-On With the ForgeRock Identity Cloud

For organizations relying on AM's session and policy services with SSO, consider cross-Domain Single Sign-On (CDSSO) as an alternative to SSO through OpenID Connect.

This example sets up ForgeRock Identity Cloud as an SSO authentication server for requests processed by Identity Gateway. For more information about Identity Gateway and CDSSO, see "Authenticating With CDSSO" in the *Gateway Guide*.

Before you start, prepare Identity Cloud, Identity Gateway, and the sample application as described in "Example Installation for This Guide" in the *Gateway Guide*.

1. Set up Identity Cloud:
 - a. Log in to the ForgeRock Identity Cloud as an administrator.
 - b. In the platform console, go to Identities > Manage > Alpha realm - Users, and add a new user with the following values:
 - Username: `demo`
 - First name: `demo`
 - Last name: `user`
 - Email Address: `demo@example.com`
 - Password: `Ch4ng3!t`
 - c. Make sure that you are managing the `alpha` realm. If not, click the current realm at the top of the screen, and switch realm.
 - d. Add an Identity Gateway agent:
 - Click Gateways & Agents, and add an agent profile with the following values:
 - ID: `ig_agent_cdssso`
 - Password: `password`
 - Redirect URLs: `http://openig.ext.com:8080/home/cdssso/redirect`

By default, the agent can introspect OAuth 2.0 tokens issued to any client, in the realm and subrealm where it is created. To change the introspection, click Native Consoles > Access Management, and update the agent in the AM console.

2. Set up Identity Gateway:

- a. Set an environment variable for the IG agent password, and then restart IG:

```
$ export AGENT_SECRET_ID='cGFzc3dvcmQ='
```

The password is retrieved by a SystemAndEnvSecretStore, and must be base64-encoded.

- b. Add the following route to IG, to serve .css and other static resources for the sample application:

Linux

```
$HOME/.openig/config/routes/static-resources.json
```

Windows

```
%appdata%\OpenIG\config\routes\static-resources.json
```

```
{
  "name" : "sampleapp_resources",
  "baseURI" : "http://app.example.com:8081",
  "condition": "${matches(request.uri.path, '^/css')}",
  "handler": "ReverseProxyHandler"
}
```

- c. Add the following route to Identity Gateway, replacing the value for the property `amInstanceUrl`:

Linux

```
$HOME/.openig/config/routes/cdsso-idc.json
```

Windows

```
%appdata%\OpenIG\config\routes\cdsso-idc.json
```

```
{
  "name": "cdsso-idc",
  "baseURI": "http://app.example.com:8081",
  "condition": "${matches(request.uri.path, '^/home/cdsso')}",
  "properties": {
    "amInstanceUrl": "<myIdentityCloudUrl/am>"
  },
  "heap": [
    {
      "name": "SystemAndEnvSecretStore-1",
      "type": "SystemAndEnvSecretStore"
    },
    {
      "name": "AmService-1",
      "type": "AmService",
    }
  ]
}
```

```

"config": {
  "url": "&{amInstanceUrl}",
  "realm": "/alpha",
  "version": "7",
  "agent": {
    "username": "ig_agent_cdsso",
    "passwordSecretId": "agent.secret.id"
  },
  "secretsProvider": "SystemAndEnvSecretStore-1",
  "sessionCache": {
    "enabled": false
  }
}
},
"handler": {
  "type": "Chain",
  "config": {
    "filters": [
      {
        "name": "CrossDomainSingleSignOnFilter-1",
        "type": "CrossDomainSingleSignOnFilter",
        "config": {
          "redirectEndpoint": "/home/cdsso/redirect",
          "authCookie": {
            "path": "/home",
            "name": "ig-token-cookie"
          },
          "amService": "AmService-1",
          "verificationSecretId": "verify",
          "secretsProvider": {
            "type": "JwkSetSecretStore",
            "config": {
              "jwkUrl": "&{amInstanceUrl}/oauth2/realms/alpha/connect/jwk_uri"
            }
          }
        }
      }
    ]
  },
  "handler": "ReverseProxyHandler"
}
}
}

```

Notice the following features of the route compared to `cdsso.json` in "Set Up CDSSO" in the *Gateway Guide*, where Access Management is running locally:

- The AmService `URL` points to Access Management in the Identity Cloud.
- The AmService `realm` points to the realm where you configure your IG agent.

3. Test the setup:

- Go to `http://openig.ext.com:8080/home/cdsso`. The Identity Cloud login page is displayed.
- Log in to Identity Cloud as user `demo`, password `Ch4ng3!t`.

Access Management calls `/home/cdssso/redirect`, and includes the CDSSO token. The `CrossDomainSingleSignOnFilter` passes the request to sample app.