chrome enterprise

Ping Identity

Ping Device Trust Connector Integration with Chrome Setup Guide

September 2023

chrome enterprise

Ping Identity

Contents

chrome enterprise

Ping Identity

Device Trust Connector Overview

The Device Trust Connector is an integration between Chrome and a 3rd party IdP that provides attestation of the device identity and enables access to context aware signals.

Ping Identity can use the signals to implement Context Aware Access (CAA) for use in zero trust architectures. Signals are delivered to PingOne via a real-time HTTP header flow.

This document outlines the steps to enable and use the Device Trust Connector in PingOne DaVinci and PingFederate.

chrome enterprise

Ping Identity

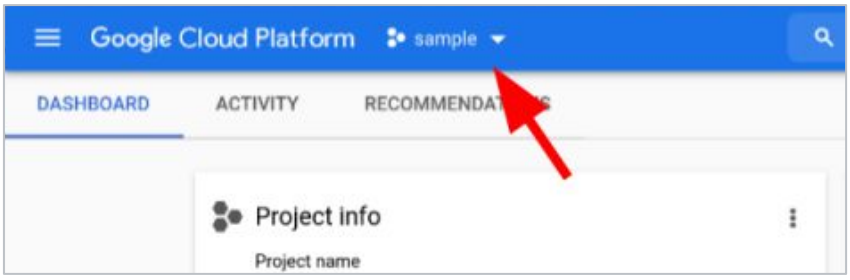
Part 1 - Create a Google Cloud Project

Create a Google Cloud Project.

All applications that make use of Google Platform APIs must create a Google Cloud Project for their application and enable the APIs that will be used.

1. You will need a google account with admin access in order to gain access to the cloud console as well as the Google Admin console in later steps. Either use an existing admin account or ask your super admin to give you access.

2. Navigate to the Google Cloud Console and sign in with your Google Admin account. Click on the drop down menu at top left of the window:

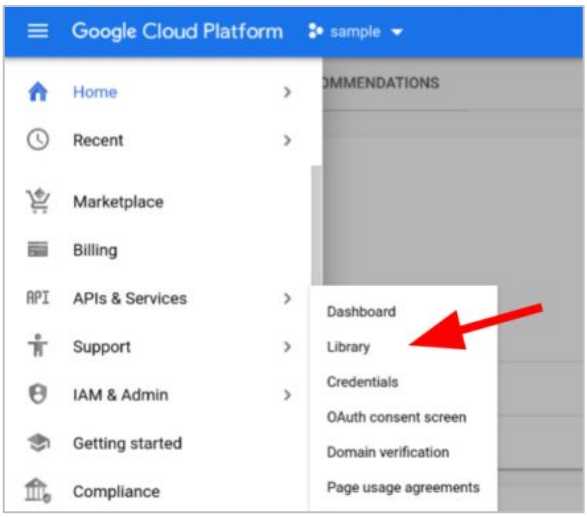


3. Open up the "**Select a project**" dialog by clicking the "**NEW PROJECT**" button at the top right of the dialog:

4. Give your project a name and click "**CREATE**".

Part 1 - Create a Google Cloud Project

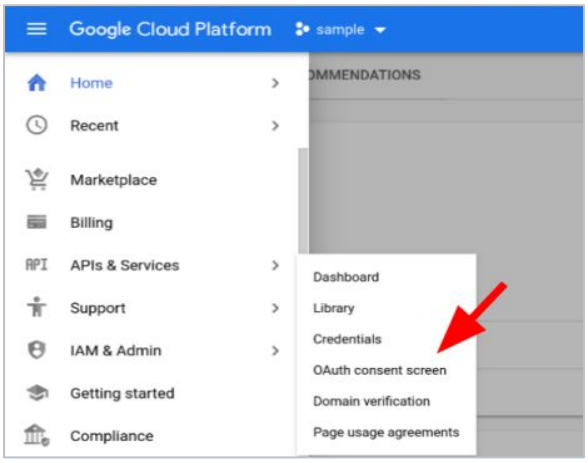
5 Add the “**Chrome Verified Access API**” — this is required in your project to support the Chrome Device Trust Connector. With your project open, select “**API & Services** > **Library**”.



6 In the “**Search for APIs & Services**” text box, enter “**Chrome Verified Access API**”. Click on the one result, and then click “**ENABLE**”.

7 Google Cloud requires setting up the OAuth consent screen as this is necessary to generate an API key and service account for the server backend.

With your project open, select “**API & Services** > **OAuth consent screen**”:



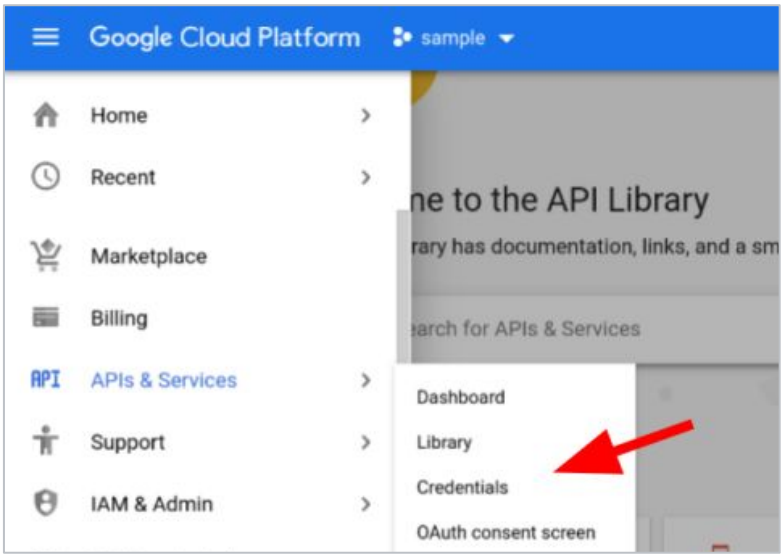
Part 1 - Create a Google Cloud Project

8. Select "**Internal**" for the user type and click "**Create**" to work through the wizard to give your project:
   - An app name
   - A support email address
   - Developer contact information

9. Then press "**Save and Continue**".

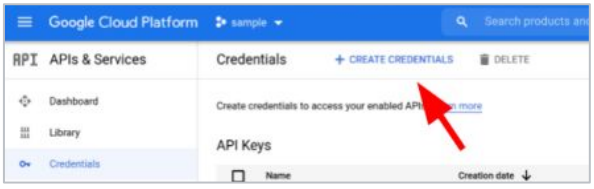
10. Click the **"add or remove scopes"** button and do a search for **"https://www.googleapis.com/auth/verifiedaccess"**. Select the checkbox by the scope and hit "**Update**".
    - NOTE: There is no need to add email addresses for testers since there is no need to sign in with a Google account. Press "**Back to dashboard**".

11. Create an API Key. With your project open, select "**API & Services > Credentials**".



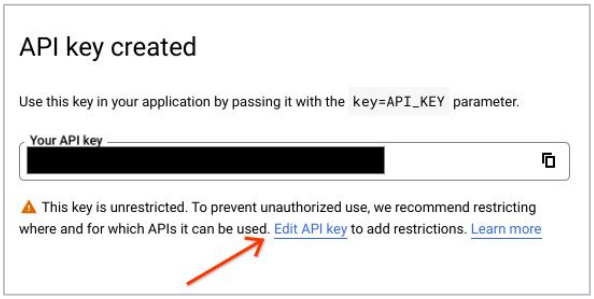
Part 1 - Create a Google Cloud Project

12 Click "+ **CREATE CREDENTIALS**".

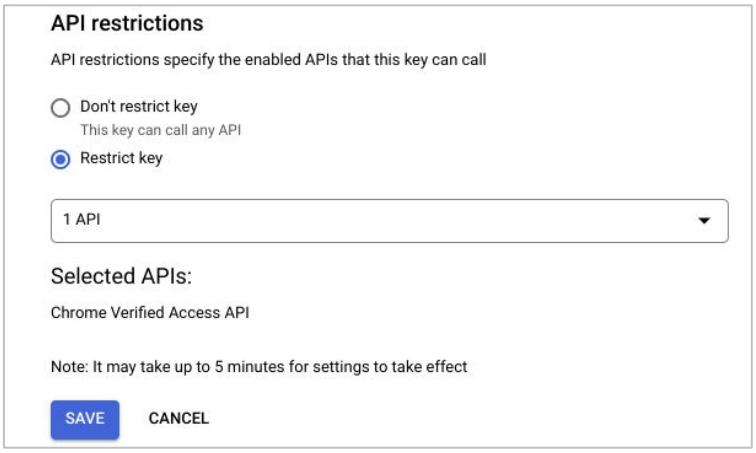


13 Choose "**API key**" from the drop down menu.

14 Click the "**Edit API key**" link.

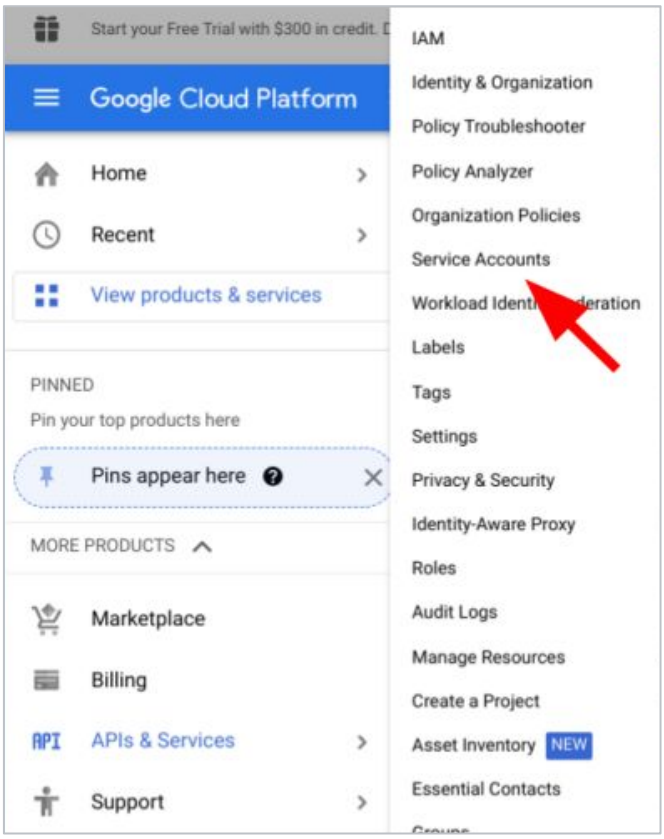


15 Under "**API restrictions**" select "**Restrict Key**". Then in the dropdown, find and select the "**Chrome Verified Access API**" and click "**OK**", and then click "**SAVE**".

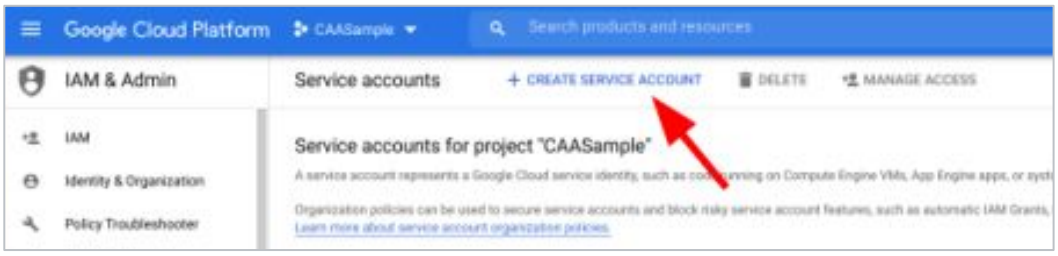


Part 2 - Create a Service Account + Key

1 With your project open, select "**IAM & Admin > Service Accounts**":



2 Click "+ **CREATE SERVICE ACCOUNT**":



Part 2 - Create a Service Account + Key

3. Enter a service account name and click "**Create and Continue**".

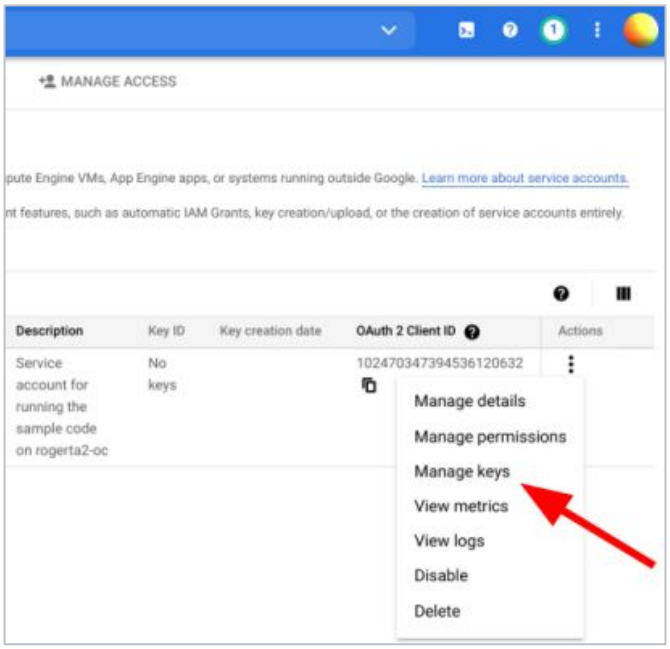
   There is no need to grant this service account access to the project.

4. Click "**Continue**".

   There is no need to grant users access to this service account as owners and editors already have access to it.

5. Click "**Done**".

6. Click on the three vertical dots to the right of the service account and choose "**Manage keys**":



7. Click on "**ADD KEY**", then "**Create new key**". Select "**JSON**" as the key type, and finally click "**CREATE**". This downloads a .json file that is used for authentication.

   Keep the credentials of this service account in a safe place. Anyone who has access to the credentials can act as this service account.

   This service account's email address will be used to enable the device trust connector in the Google Admin console. See details in Part 3.

Part 3 - Enabling Device Trust Connector in the Google Admin console

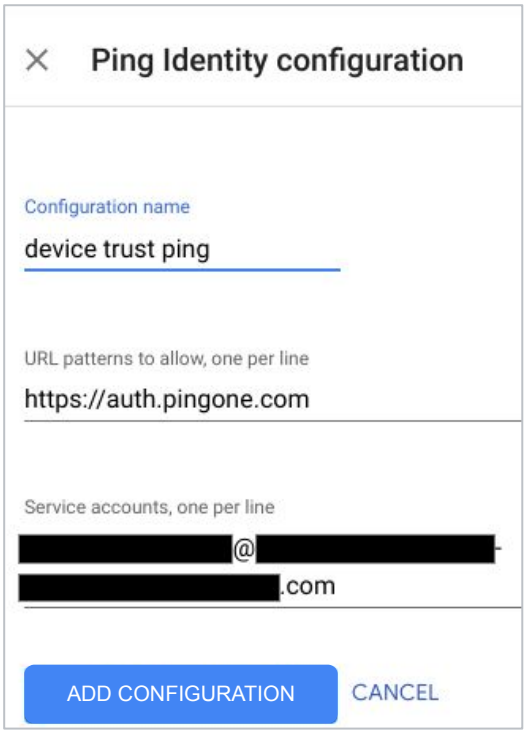
1. Go to the Google Admin console.
2. Go to "**Devices > Chrome > Connectors**".
3. (If applicable) Accept the license agreement for using Connectors.
4. Click the "**+ New Provider Configuration**" button.
5. Choose the **Ping Identity** device trust connector provider and click "**SET UP**".
6. Provide a unique name for your configuration under "**configuration name**".
7. Input the DaVinci tenant callback URL (Most likely https://auth.pingone.com) under "**URL patterns to allow, one per line**"
8. Input the service account created in Part 2 - Step 7 under "**Service accounts, one per line**".

chrome enterprise

Ping Identity

Part 3 - Enabling Device Trust Connector in the Google Admin console

9 Click “Add Configuration”.



10 Now you can apply this provider configuration to your desired organizational unit.

a Choose your desired organizational unit on the tree UI widget to the left.

b Scroll down to “**Device trust connectors**”, use the radio buttons in this section to apply the appropriate configuration.

c Click “**Save**”.

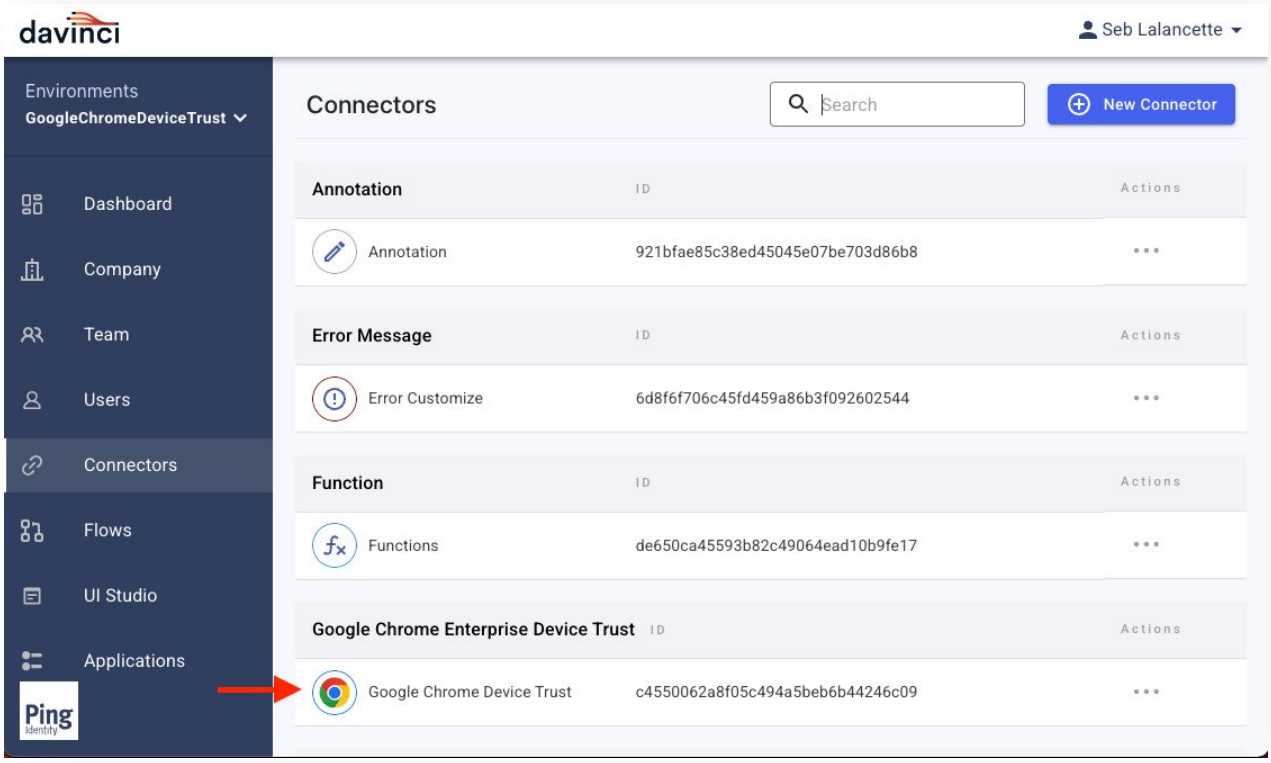
To continue set up in **PingOne** Davinci or **Ping**Federate go to:

For PingOne Davinci

For Ping Federate

Part 4a - Add Chrome Device Trust Connector in PingOne DaVinci

1. Log into your **PingOne DaVinci** account and click on the "**New Connector**" button (top-right)
2. Search/find the "**Google Chrome Enterprise Device Trust**" entry,
3. Click the plus sign + and give your connection a name
4. Once created, the connection will appear on the page next to the existing one.

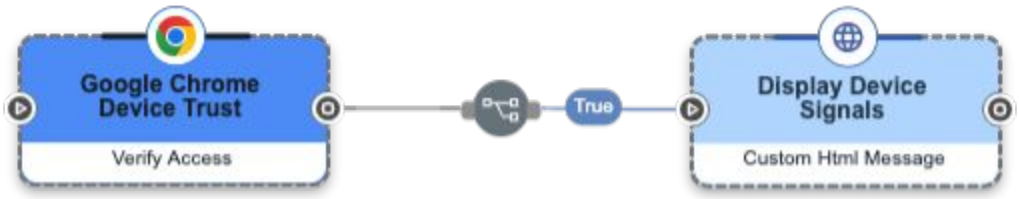


5. Browse to the "**Flows**" section.

Part 4a - Add Chrome Device Trust Connector in PingOne DaVinci

6. A sample flow including the PingOne DaVinci Chrome Device Trust connector is available for download here: https://raw.githubusercontent.com/pingone-davinci/flows/main/third-party/Chrome_Device_Trust_Flow.json or you can create your own flow.

7. To import the sample flow directly into your DaVinci tenant, start by clicking on "**Flows>Add Flow>Import from JSON**" and select the "**downloaded sample flow**" file. The imported flow is displayed below as such:

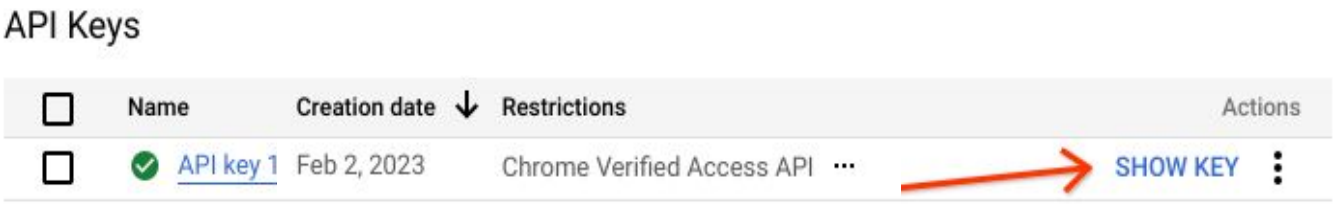


8. Once it is imported, click on the "**Google Chrome Device Trust**" node and click "**configure**".

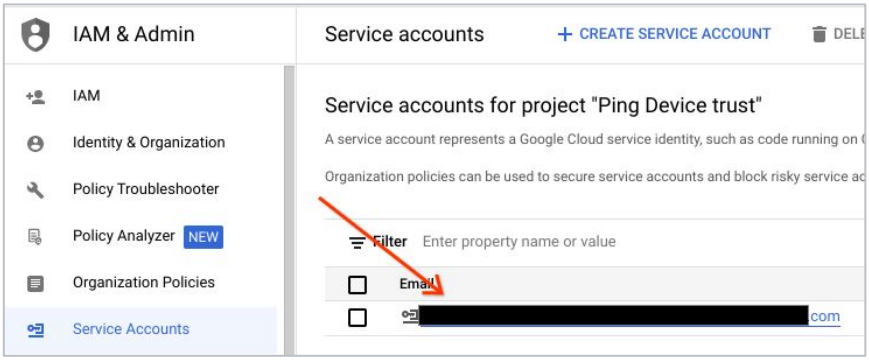
Part 4a - Add Chrome Device Trust Connector in PingOne DaVinci

9. Enter the following (from console.cloud.google.com) into the DaVinci Google Chrome Device Trust connector:

   a. **API key:** (Found in "**APIs & Services>Credentials>Click Show Key**" on the API that you created in Part 1.)



   b. **Credentials Client email:** Service account id (Found in "**IAM & Admin>service accounts>Email**")

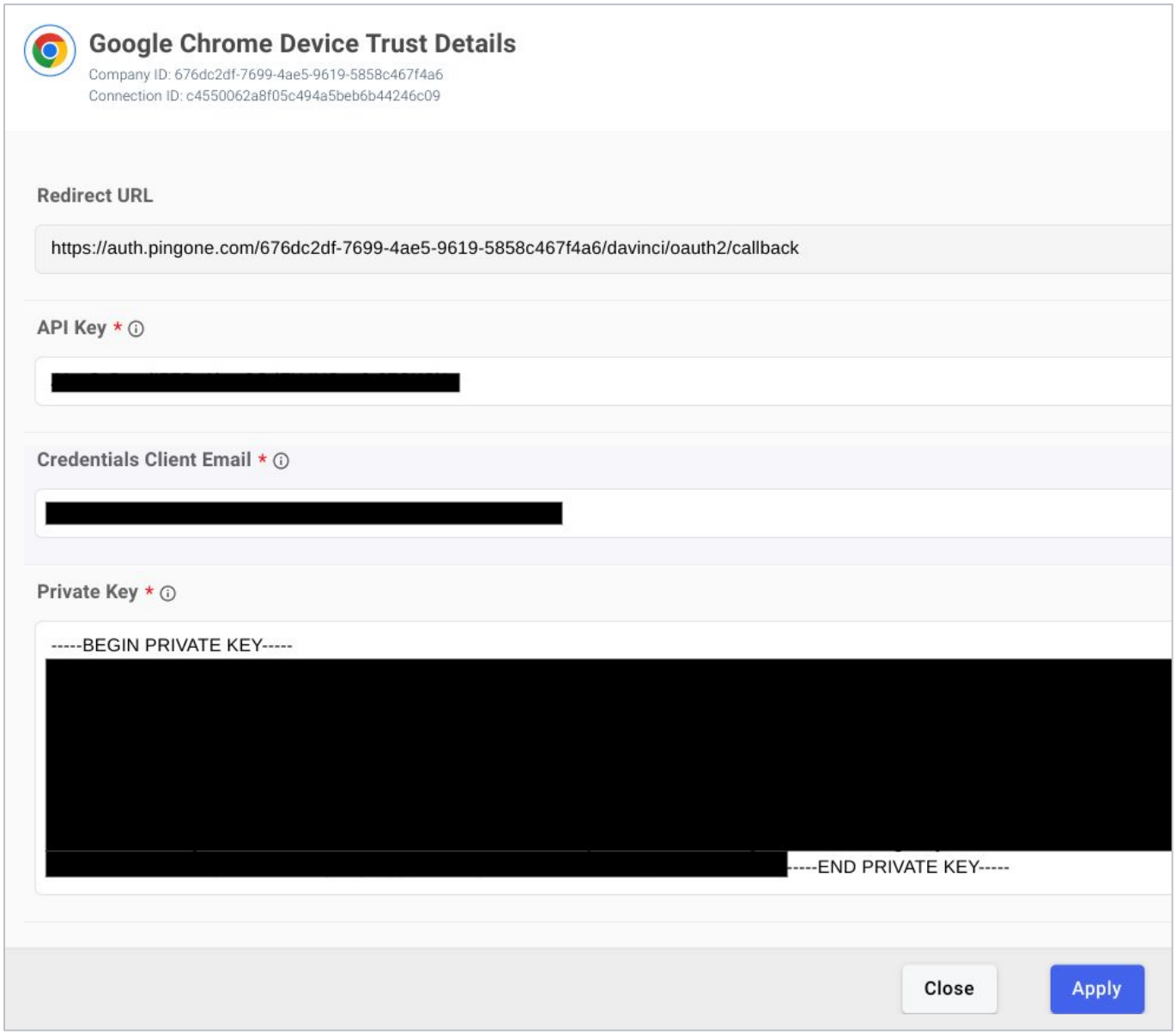


   c. **Private key**: (The contents of the private key JSON that was downloaded in Part 2. Copy everything from -----BEGIN PRIVATE KEY----- to -----END PRIVATE KEY-----)

   d. Click "**Apply**" and your new connection can now be used in a flow.

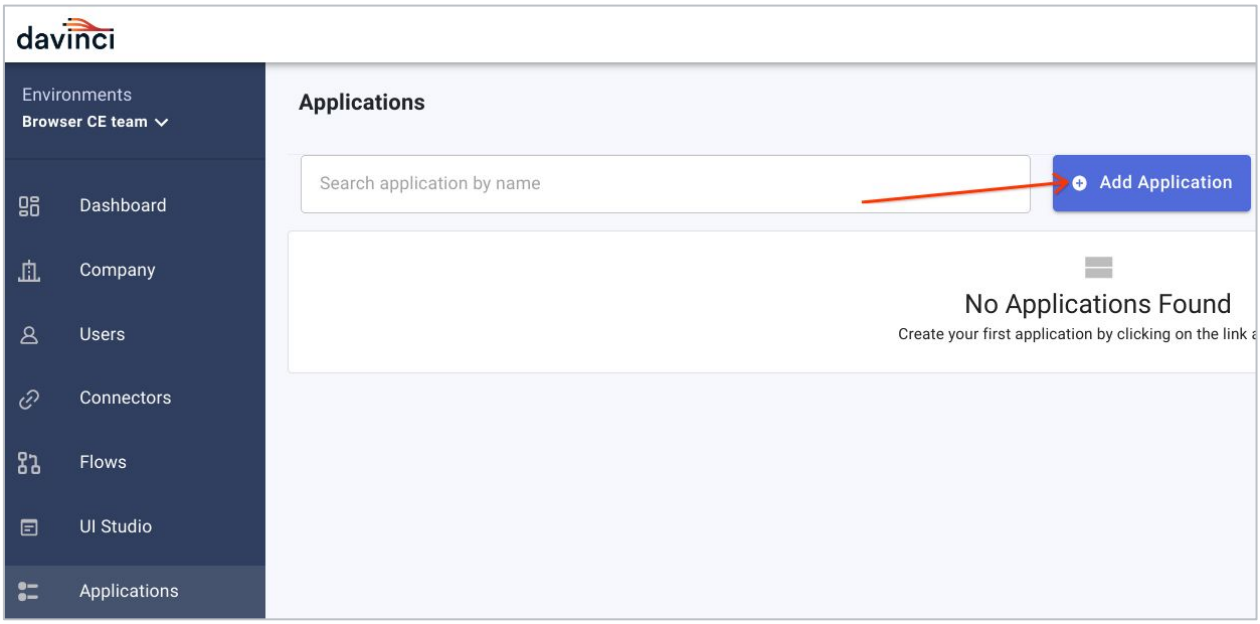
Part 4a - Add Chrome Device Trust Connector in PingOne DaVinci

Here is an example of of what it looks like within PingOne DaVinci



Part 4a - Add Chrome Device Trust Connector in PingOne DaVinci

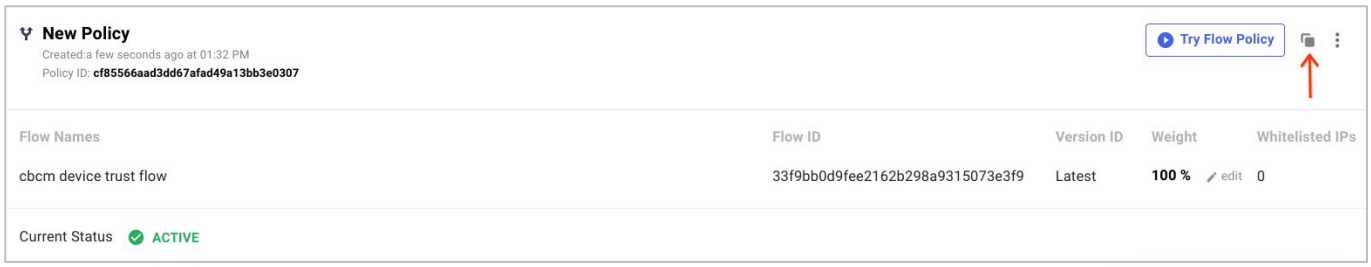
10 In Da Vinci go to Applications and click on the Add Application button and give it a name and hit the create button.



11 Click the edit button and select the "**Flow policy**" tab and click "**Add Flow Policy**" and select the Flow that you created or imported from the previous steps and select latest version.

12 Give it a weight of 100% and hit "**save Flow Policy**".

13 Hit the **copy icon** next to "Try Flow Policy" to save a link that you can run on your enrolled machine to test the connection.

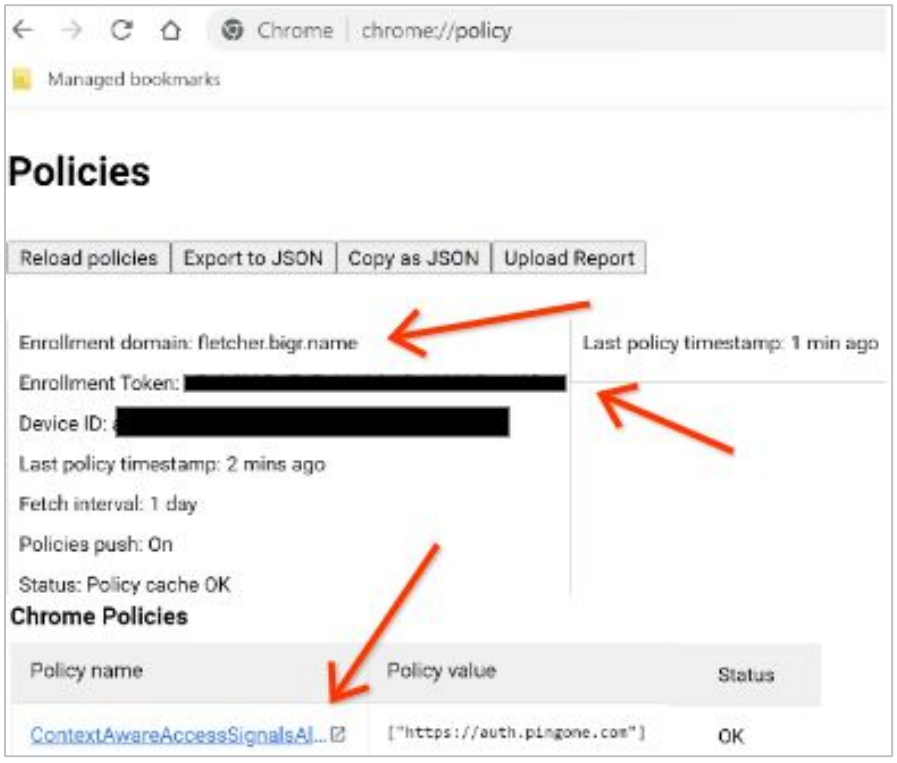


Part 5a - Verify that test device is configured correctly

On the machine that you have enrolled in Chrome Browser Cloud management (CBCM) go to "**Chrome://policy**" and confirm the following:

- Enrollment domain matches the domain that you have set in the Google Admin console
- Enrollment token value matches the one corresponding to the Organization Unit that you have the Ping integration activated on.
- In CBCM*, there is a policy set called "**BrowserContextAwareAccessSignalsAllowlist**" with a value equal to the value that you entered in the URL pattern space in the Connectors setup in the Google Admin console

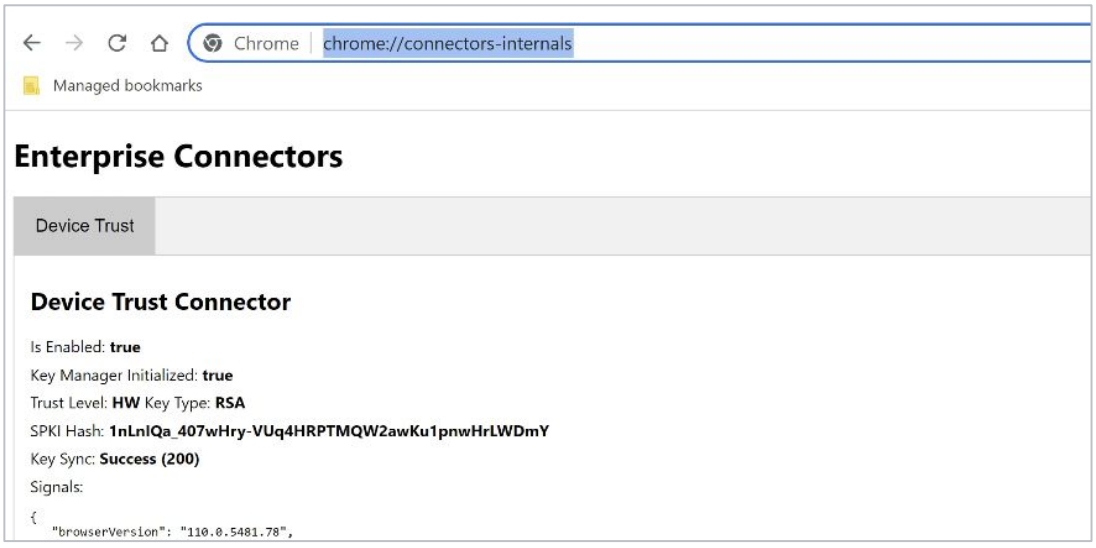
* In ChromeOS, the policy will be called either "**DeviceLoginScreenContextAwareAccessSignalsAllowlist**" for DTC enabled for the managed device, or "**UserContextAwareAccessSignalsAllowlist**" for DTC enabled for the managed user.



Part 5a - Verify that test device is configured correctly

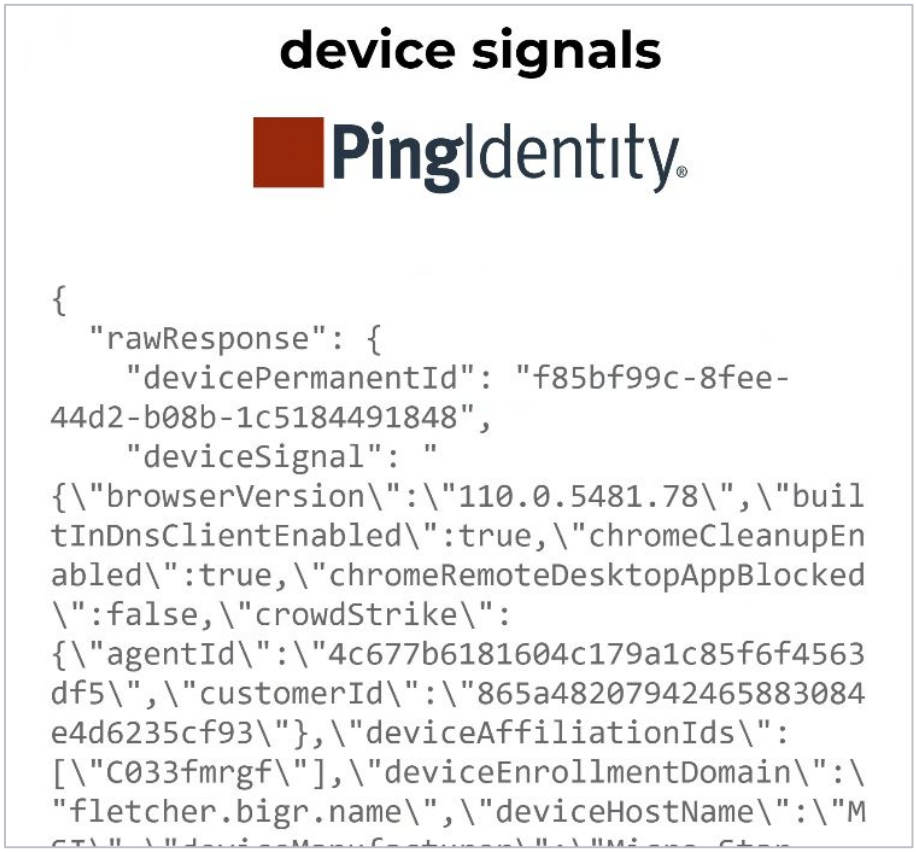
You can also verify that the integration is active and the key has been created by going to **chrome://connectors-internals** on the enrolled machine. Here is a bit more information about what each field means

- **Is enabled:** verifies that the policy is enabled on the machine
- **Key Manager initialized (non-CrOS):** Chrome has loaded the key or created a key if no key was created already
- **Trust Level (non-CrOS):** Could be HW or SW.
  - HW (hardware) means that the key is stored in the machine's hardware (like on Mac with Secure Enclave or Windows when a TPM is present).
  - SW (software) means that the key is stored at the OS level (for example in a file, like on Linux)
- **SPKI Hash (non-CrOS):** A hash of the public key
- **Key Sync (non-CrOS):** The response status + code from the latest attempted key upload. Chrome tries to re-upload the key every time it starts
- **Signals:** An overview of the signals that can be sent from the machine



Part 5a - Verify that test device is configured correctly

- To test that the flow that you created in Davinci is working correctly, in Chrome, paste the flow policy link that you copied from the Application setup in the Step 13 of Part 4a.
- Go through the flows and if you see a result of the raw responses (that is the raw data of the device signals) you are setup correctly.



- Once these settings are confirmed to be in place, attempt to access a resource that is gated by Ping to confirm that the integration is working.

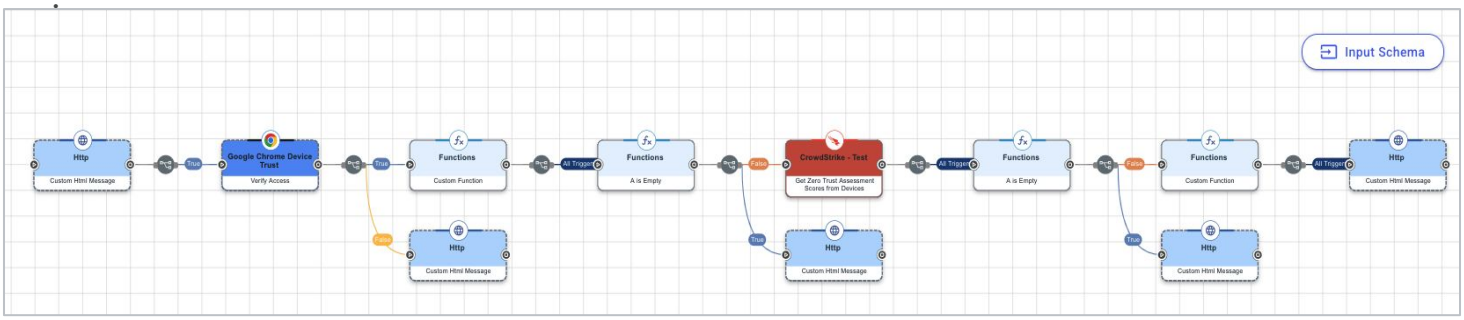
chrome enterprise

Ping Identity

Part 6a - Create a Chrome Device Trust Connector Flow in PingOne DaVinci.

Reach out to your account representative at Ping Identity if you need assistance creating the context aware rules using the Chrome Device Trust Connector in PingOne DaVinci.

In order to set up the Chrome Device Trust Connector flow in PingOne DaVinci, refer to this guide for further instructions.



Set up in PingOne DaVinci complete.

For FAQ go to page 26.

PingOne Davinci

Part 4b - Add Chrome Device Trust Adapter to PingFederate

To get started with the integration, deploy the Google Chrome Device Trust Integration Kit files to your PingFederate directory.

Note: If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps:

1. Download the Google Chrome Integration Kit .zip archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, back up your customizations and delete previous versions of the integration files.
4. Delete "**pf-google-chrome-device-trust-adapter-<version>.jar**" from "**<pf_install>/pingfederate/server/default/deploy**".
5. If you backed up any customized files, modify the new files with your customizations.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-6 for each engine node.

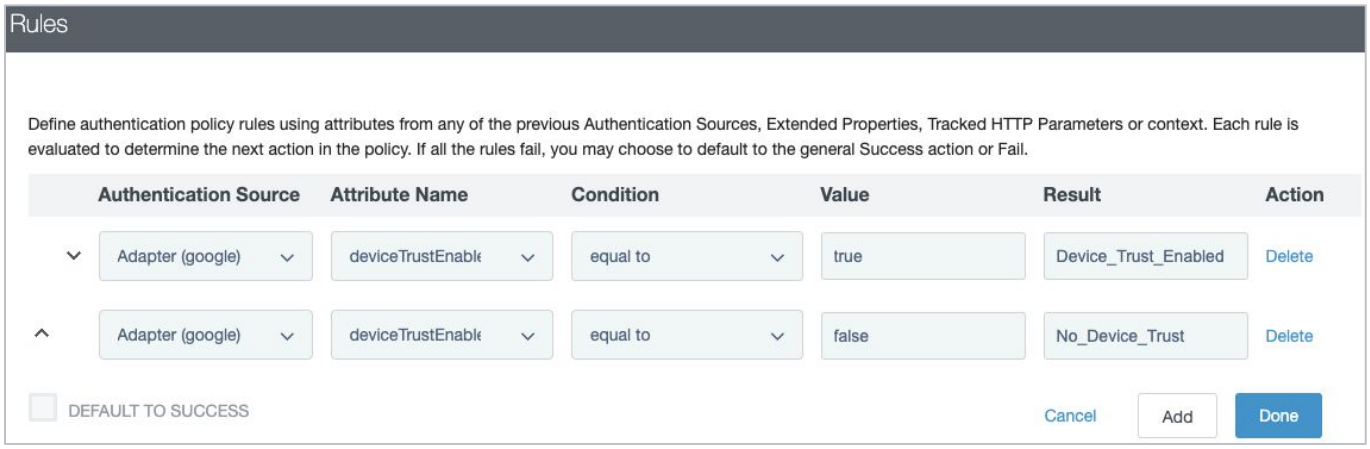
Part 4b - Add Chrome Device Trust Adapter to PingFederate

8. Login to the PingFederate Administrator console
9. Navigate to the "**IdP Adapters**" section
10. Click "**Create New Instance**"
    a. Fill in the "**instance name and id**"
    b. For "**type**", select "**Google Chrome Device Trust Adapter**"
    c. Click "**Next**"
11. Enter the API key (Found in "**APIs & Services>Credentials>Click Show Key**" on the API that you created in Part 1.)
12. Select the "**GCP Private Key File**" downloaded in Part 2 - Step 7.
13. Click "**Next**"
14. For the "**Extended Contract**" page, click "**Next**" again.
    Note, all the attributes returned by the adapter are listed on this page.
15. Next, select the "**devicePermanentId**" as the "**Pseudonym**" and click "**Save**".

Now, the Chrome Device Trust adapter is ready to use within the PingFederate authentication policy.

Part 4b - Add Chrome Device Trust Adapter to PingFederate

Leverage the PingFederate authentication policy Rule for IdP adapters to create the required authentication logic based off attributes from the Google Chrome Enterprise Device Trust integration such as shown below:

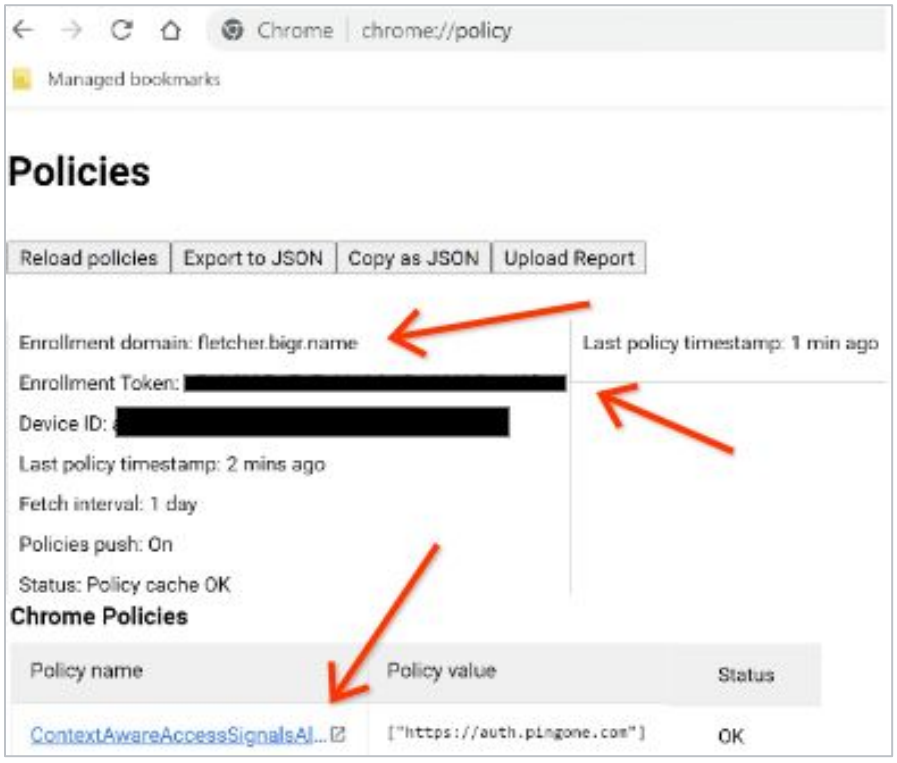


Part 5b - Verify that test device is configured correctly

On the machine that you have enrolled in Chrome Browser Cloud management (CBCM) go to "**Chrome://policy**" and confirm the following:

- Enrollment domain matches the domain that you have set in the Google Admin console
- Enrollment token value matches the one corresponding to the Organization Unit that you have the Ping integration activated on.
- In CBCM*, there is a policy set called "**BrowserContextAwareAccessSignalsAllowlist**" with a value equal to the value that you entered in the URL pattern space in the Connectors setup in the Google Admin console

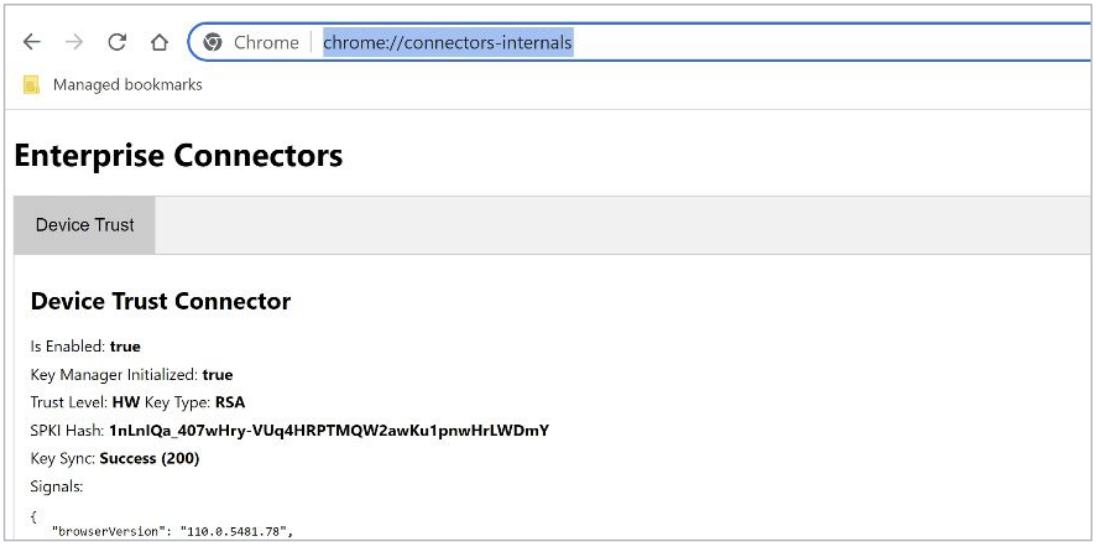
* In ChromeOS, the policy will be called either "**DeviceLoginScreenContextAwareAccessSignalsAllowlist**" for DTC enabled for the managed device, or "**UserContextAwareAccessSignalsAllowlist**" for DTC enabled for the managed user.



Part 5b - Verify that test device is configured correctly

You can also verify that the integration is active and the key has been created by going to **chrome://connectors-internals** on the enrolled machine. Here is a bit more information about what each field means

- **Is enabled:** verifies that the policy is enabled on the machine
- **Key Manager initialized (non-CrOS):** Chrome has loaded the key or created a key if no key was created already
- **Trust Level (non-CrOS):** Could be HW or SW.
  - HW (hardware) means that the key is stored in the machine's hardware (like on Mac with Secure Enclave or Windows when a TPM is present).
  - SW (software) means that the key is stored at the OS level (for example in a file, like on Linux)
- **SPKI Hash (non-CrOS):** A hash of the public key
- **Key Sync (non-CrOS):** The response status + code from the latest attempted key upload. Chrome tries to re-upload the key every time it starts
- **Signals:** An overview of the signals that can be sent from the machine



Once these settings are confirmed to be in place, attempt to access a resource that is gated by Ping to confirm that the integration is working.

FAQ

How are managed browsers trusted?

The Chrome servers establish trust with managed browsers based on the Trust On First Use mechanism. When it detects that the Device Trust connector is enabled, a managed browser will create an asymmetric key pair and upload the public key to be stored along with the browser's record in the Google Admin console. That public key will subsequently be used to validate signatures and establish trust with regards to the origin of a payload.

Some Notes on Keys

- Keys are only used on Windows and Mac. The ChromeOS integration instead establishes trust using enterprise certificates stored on managed devices.
- The "Clear key" operation can be useful for admins who are trying to unblock their users who, somehow, managed to lose their initial key.

FAQ

How can I clear a trusted key?

Admins with access to the Google Admin console can clear a trusted public key for a specific browser. This troubleshooting step can prove useful if a user is experiencing access issues which have the symptoms of a managed browser no longer having access to the trusted key pair.

The **"Clear Key"** action will simply delete the public key stored on the server for the corresponding browser. This will allow the user to restart the browser and have it upload its current public key to establish trust once again.

Key Revocation

- Windows
- Mac
- CrOS (N/A)

Clearing a Trusted Key

To clear a key visit Cloud Browser Cloud Management and follow the steps:

1. Go to **"Devices > Chrome > Managed browsers"**.
2. Select the Organizational Unit where the browser(s) is located.
3. Select the browser with the key to be cleared.
4. Underneath the Managed Browser details box on the left hand side click **"Configure Key"**.
5. Select **"CLEAR KEY"**.

If the "Configure Key" is not clickable it is most likely because the key does not exist on the server.

chrome enterprise

Ping Identity

FAQ

How do I unenroll a device?

To unenroll a managed device from Chrome Browser Cloud Management navigate to this page for more information. To unenroll a Chrome OS device follow these steps.

What platforms is Device Trust Connector supported on?

- Windows
- ChromeOS*
- Mac

*Currently not available on ChromeOS Flex

Additional Resources

- Chrome Browser Cloud Management
- Chrome Device Management
- Learn More at Chrome Enterprise Help Center
- Learn More at Ping Support Center