+++ title = "Sign a plugin" +++

Sign a plugin

Signing a plugin allows Grafana to verify the authenticity of the plugin with [signature verification]({{< relref "../../plugins/plugin-signatures.md" >}}). This gives users a way to make sure plugins haven't been tampered with. All Grafana Labs-authored backend plugins, including Enterprise plugins, are signed.

> ***Important:*** *Future versions of Grafana will require all plugins to be signed.*

Before you can sign your plugin, you need to decide whether you want to sign it as a *public* or a *private* plugin.

If you want to make your plugin publicly available outside of your organization, you need to sign your plugin under a *community* or *commercial* signature level. Public plugins are available from grafana.com/plugins and can be installed by anyone.

For more information on how to install public plugin, refer to [Install Grafana plugins]({{< relref "../../plugins/installation.md" >}}).

If you intend to only use the plugin within your organization, you can to sign it under a *private* signature level.

Generate an API key

To verify ownership of your plugin, you need to generate an API key that you'll use every time you need to sign a new version of your plugin.

1. Create a Grafana Cloud account.
2. Make sure that the first part of the plugin ID matches the slug of your Grafana Cloud account.

   You can find the plugin ID in the `plugin.json` file inside your plugin directory. For example, if your account slug is `acmecorp`, you need to prefix the plugin ID with `acmecorp-`.
3. Create a Grafana Cloud API key with the **PluginPublisher** role.

Sign a public plugin

Public plugins need to be reviewed by the Grafana team before you can sign them.

1. Submit your plugin for [review]({{< relref "package-a-plugin.md#publishing-your-plugin-for-the-first-time" >}})
2. When your plugin is approved, you're granted a plugin signature level. **Without a plugin signature level, you won't be able to sign your plugin**.
3. In your plugin directory, sign the plugin with the API key you just created. Grafana Toolkit creates a MANIFEST.txt file in the `dist` directory of your plugin.

```
export GRAFANA_API_KEY=<YOUR_API_KEY>
npx @grafana/toolkit plugin:sign
```

> ***Note:*** *If running NPM 7+ the* `npx` *commands mentioned in this article may hang. The workaround is to use* `npx --legacy-peer-deps <command to run>`.

Sign a private plugin

1. In your plugin directory, sign the plugin with the API key you just created. Grafana Toolkit creates a MANIFEST.txt file in the `dist` directory of your plugin.

   The `rootUrls` flag accepts a comma-separated list of URLs to the Grafana instances where you intend to install the plugin.

   ```
   export GRAFANA_API_KEY=<YOUR_API_KEY>
   npx @grafana/toolkit plugin:sign --rootUrls https://example.com/grafana
   ```

Plugin signature levels

To sign a plugin, you need to decide the *signature level* you want to sign it under. The signature level of your plugin determines how you can distribute it.

You can sign your plugin under three different *signature levels*.

Plugin Level	Paid Subscription Required?	Description
Private	No; Free of charge	You can create and sign a Private Plugin for any technology at no charge. Private Plugins are for use on your own Grafana. They may not be distributed to the Grafana community, and are not published in the Grafana catalog.
Community	No; Free of charge	You can create, sign and distribute plugins at no charge, provided that all dependent technologies are open source and not for profit. Community Plugins are published in the official Grafana catalog, and are available to the Grafana community.
Commercial	Yes; Commercial Plugin Subscription required	You can create, sign and distribute plugins with dependent technologies that are closed source or commercially backed, by entering into a Commercial Plugin Subscription with Grafana Labs. Commercial Plugins are published on the official Grafana catalog, and are available to the Grafana community.

For instructions on how to sign a plugin under the Community and Commercial signature level, refer to Sign a public plugin.

For instructions on how to sign a plugin under the Private signature level, refer to Sign a private plugin.

Plugin manifest

For Grafana to verify the digital signature of a plugin, the plugin must include a signed manifest file, *MANIFEST.txt*. The signed manifest file contains two sections:

- **Signed message -** The signed message contains plugin metadata and plugin files with their respective checksums (SHA256).
- **Digital signature -** The digital signature is created by encrypting the signed message using a private key. Grafana has a public key built-in that can be used to verify that the digital signature have been encrypted using expected private key.

Example manifest file:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

{
  "manifestVersion": "2.0.0",
  "signatureType": "community",
  "signedByOrg": "myorgid",
  "signedByOrgName": "My Org",
  "plugin": "myorgid-simple-panel",
  "version": "1.0.0",
  "time": 1602753404133,
  "keyId": "7e4d0c6a708866e7",
  "files": {
    "LICENSE": "12ab7a0961275f5ce7a428e662279cf49bab887d12b2ff7bfde738346178c28c",
    "module.js.LICENSE.txt":
"0d8f66cd4afb566cb5b7e1540c68f43b939d3eba12ace290f18abc4f4cb53ed0",
    "module.js.map":
"8a4ede5b5847dec1c6c30008d07bef8a049408d2b1e862841e30357f82e0fa19",
    "plugin.json":
"13be5f2fd55bee787c5413b5ba6a1fae2dfe8d2df6c867dadc4657b98f821f90",
    "README.md": "2d90145b28f22348d4f50a81695e888c68ebd4f8baec731fdf2d79c8b187a27f",
    "module.js": "b4b6945bbf3332b08e5e1cb214a5b85c82557b292577eb58c8eb1703bc8e4577"
  }
}
-----BEGIN PGP SIGNATURE-----
Version: OpenPGP.js v4.10.1
Comment: https://openpgpjs.org

wqEEARMKAAYFAl+IE3wACgkQfk0ManCIZudpdwIHTCqjVzfm7DechTa7BTbd
+dNIQtwh8Tv2Q9HksgN6c6M9nbQTP0xNHwxSxHOI8EL3euz/OagzWoiIWulG
7AQo7FYCCQGucaLPPK3tsWaeFqVKy+JtQhrJJui23DAZLSYQYZlKQ+nFqc9x
T6scfmuhWC/TOcm83EVoCzIV3R5dOTKHqkjIUg==
=GdNq
-----END PGP SIGNATURE-----
```

Troubleshooting issues while signing your plugin

Why am I getting a "Modified signature" in Grafana?

Due to an issue when signing the plugin on Windows, grafana-toolkit generates an invalid MANIFEST.txt. You can fix this by replacing all double backslashes, `\\`, with a forward slash, `/` in the MANIFEST.txt file. You need to do this every time you sign your plugin.

Error signing manifest: Field is required: rootUrls

If you're trying to sign a **public** plugin, this means that your plugin doesn't have a plugin signature level assigned to it yet. A Grafana team member will assign a signature level to your plugin once it has been reviewed and approved. For more information, refer to Sign a public plugin.

If you're trying to sign a **private** plugin, this means that you need to add a `rootUrls` flag to the `plugin:sign` command. The `rootUrls` must match the [root_url]({{< relref "../../administration/configuration.md#root_url" >}}) configuration. For more information, refer to Sign a private plugin.

If you still get this error, make sure that the API key was generated by a Grafana Cloud account that matches the first part of the plugin ID.