



# GRAFANA DOCUMENTATION

**Manan Tandon**

@4d616e616e

# HOW TO SIGN A CUSTOM PLUGIN (PRIVATELY)?

## PUBLISH A PLUGIN (PRIVATE)

### STEP 1: BUILDING YOUR PLUGIN

"cd" change directory to your plugin folder

**syntax:**

```
[sh-3.2# cd m4d616e616e-testplugin-panel/]
```

execute this command "**yarn install --pure-lockfile**" this command will make a new **yarn.lock** file

**syntax:**

```
[sh-3.2# yarn install --pure-lockfile]
```

**yarn.lock** ensures that when another developer runs **yarn install**, they will get the exact same version of each dependency, avoiding any potential incompatibilities or bugs that can arise when different versions of the same package are installed.

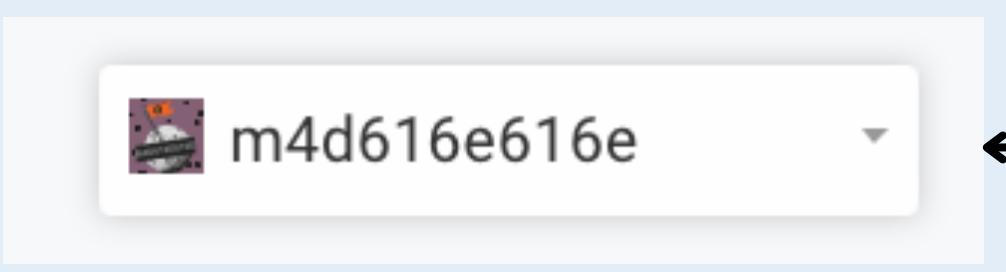
after executing the above command build your project with the following command "**yarn build**". this command ensures that you have installed all the required packages and will also check for errors in your code. if your code has any error or any dependency is not installed properly then you will get prompted with the error message in the CLI.

**syntax:**

```
[sh-3.2# yarn build]
```

## STEP 2: Sign the Plugin

- initially you need to make an [Grafana Cloud](#) account and make sure that the first part of your **PluginId** matches you handle name in Grafana Account.



← account name in grafana

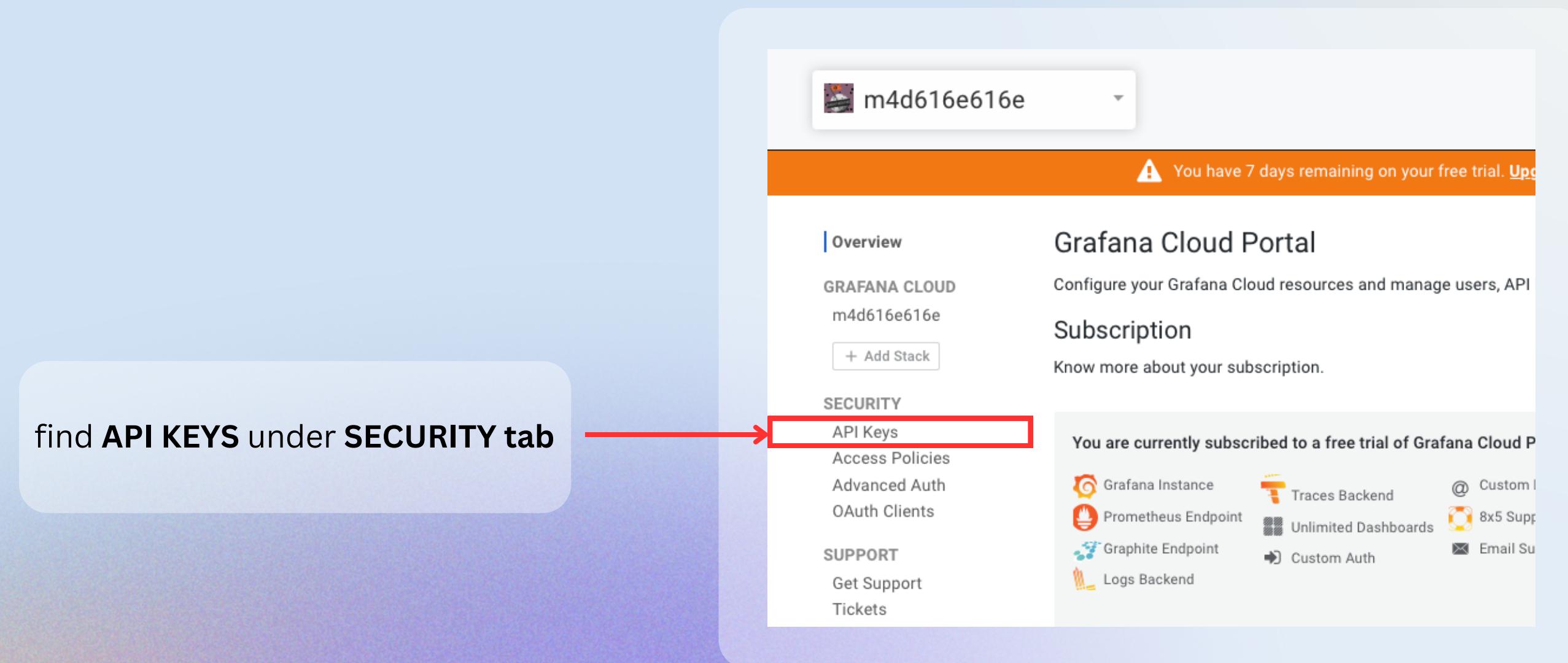
- you can find your Plugin's **PluginId** in the **plugin.json** inside the **src** folder your project.

```
{  
  "$schema": "https://raw.githubusercontent.com/grafana/grafana/master/docs/sources/developers/plugins/plugin-schema.json",  
  "type": "panel",  
  "name": "Testplugin",  
  "id": "m4d616e616e-testplugin-panel",
```



↑  
first part same as account name

- now we have to generate **API key** with the role of a **PluginPublisher**



Overview

GRAFANA CLOUD  
m4d616e616e  
+ Add Stack

API Keys

first api  
PluginPublisher role | Issued on March 22, 2023

new key  
PluginPublisher role | Issued on March 23, 2023

test api  
PluginPublisher role | Issued on March 28, 2023

+ Add API Key

A tooltip on the '+ Add API Key' button says: click on **Add API KEY**

Create an API Key

Create an API Key to use your Grafana Cloud Stack resources.

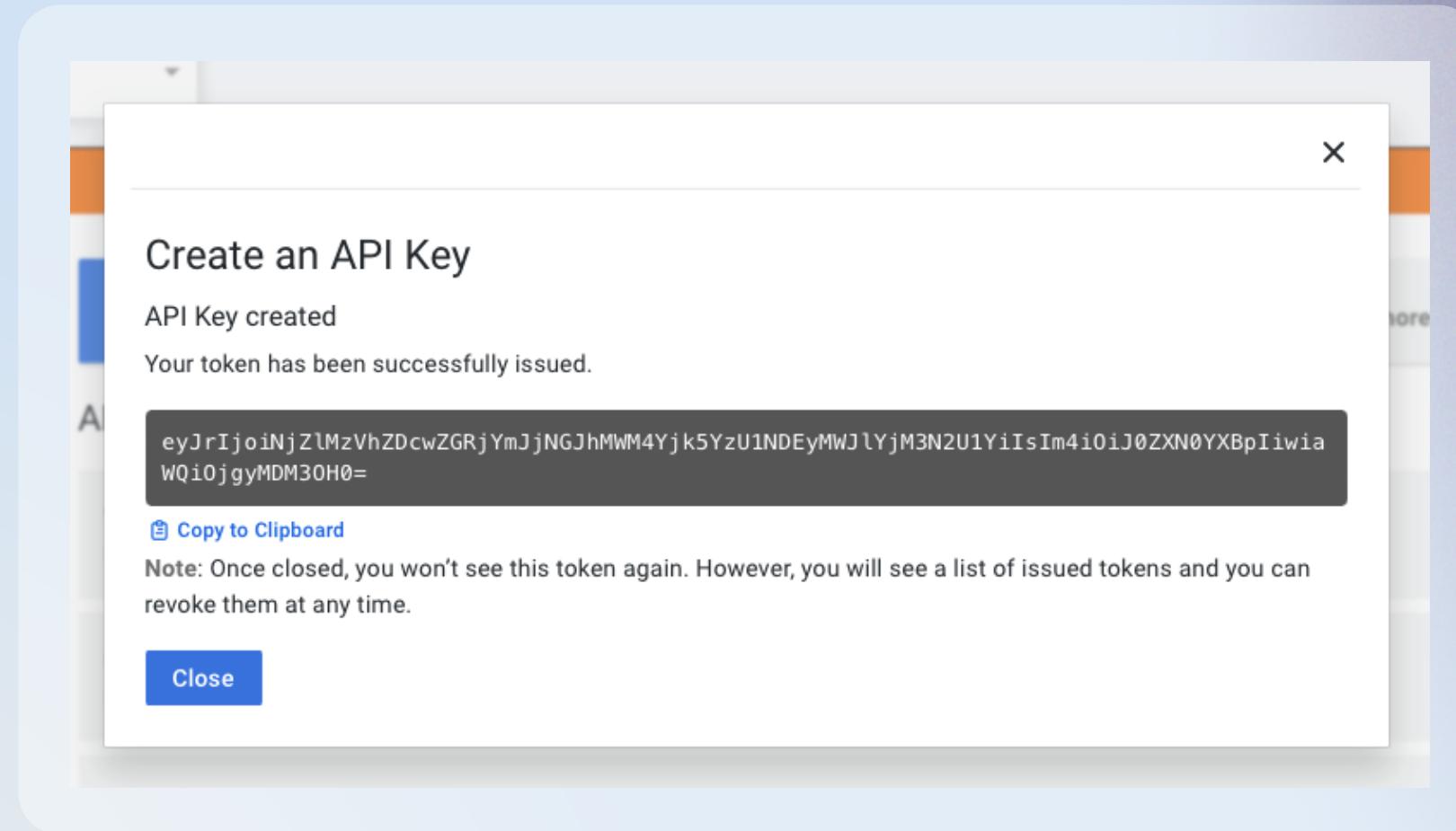
API Key Name  
testapi

Role  
PluginPublisher

Create API Key

give a **name** to your API and select the role of **PluginPublisher** and click on **Create Api Key**

- you will get prompted with an **API KEY** copy and save it somewhere.



- now **Sign** the plugin with the created **API KEY**. Signing will create a **MANIFEST.TXT** inside the **dist** directory of your project file which will have all the license information of your **Custom Plugin**.

execute the following command inside you project directory to sign the plugin:

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

# for MAC:

paste the copied **API KEY** in the **GRAFANA\_API\_KEY** environment variable  
and set **rootUrls** to **http://localhost:3000/**

## syntax:

```
[sh-3.2# GRAFANA_API_KEY=eyJrIjoINTQ0YzgyY2MwNDQ5YjgwMDJmOTI5ZjA5YjlkNjRkM2NkMzMzdMMiIsIm4iOiJuZXcg2V5IiwiaWQiOjgyMDM3OH0= npx @grafana/sign-plugin@latest --rootUrls http://localhost:3000/
```

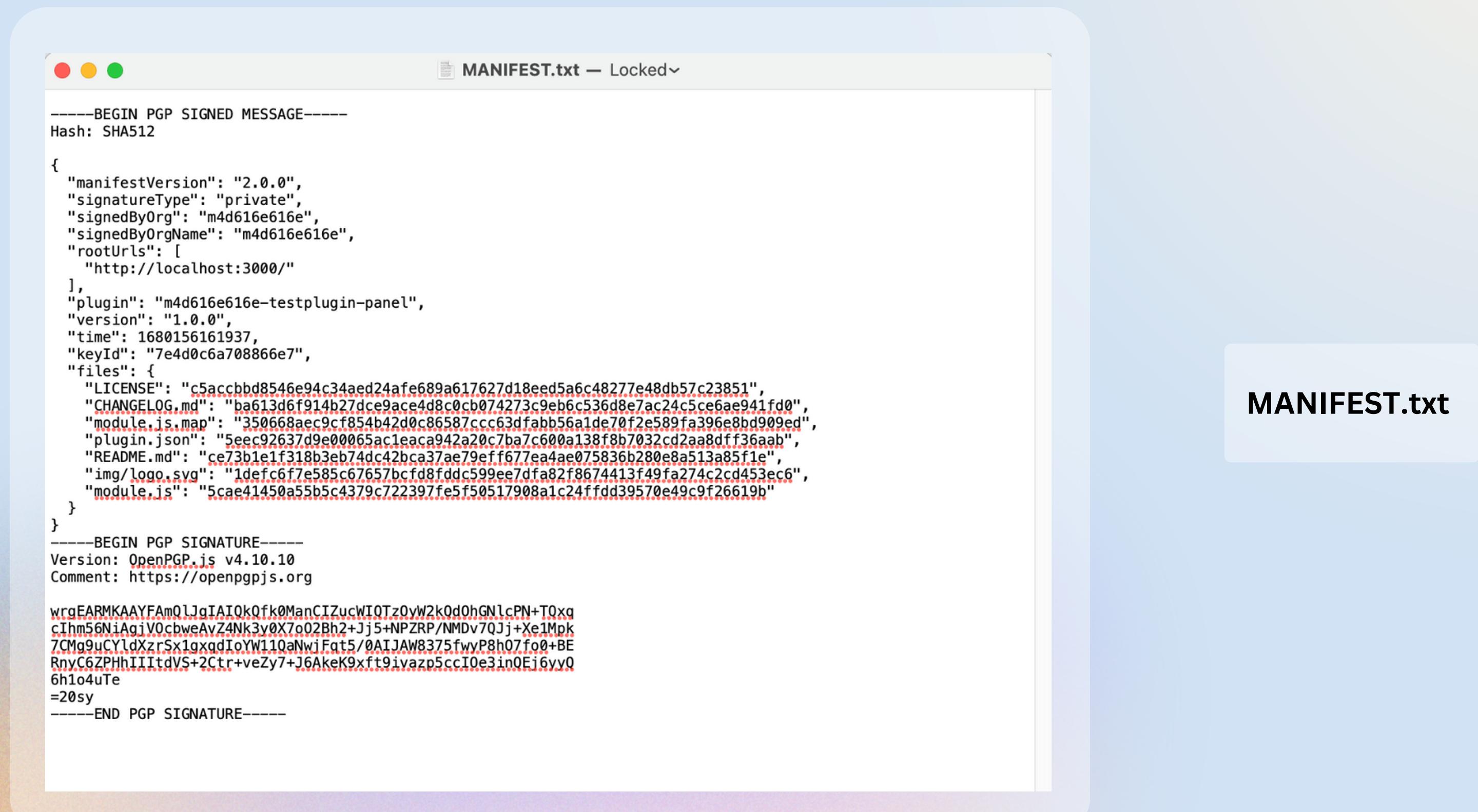
## for Windows:

change the 'export' command to 'set' and paste the copied **API KEY** in the **GRAFANA\_API\_KEY** environment variable and set **rootUrls** to **http://localhost:3000/**

## syntax:

```
updatefields-panel>set GRAFANA_API_KEY=eyJrIjoi0WE3MjVkJY2NmNGIyMDFiZjFlZWlwc40DgwYWUxMDViNDY3ZWY3NyIsIm4iOiJ0aW1lIHNLcmllcyIsImlkIjo4MjAzNzh9  
updatefields-panel>npx @grafana/sign-plugin@latest --rootUrls http://localhost:3000/
```

- after your Plugin is Signed a **MANIFEST.txt** file will be created inside **dist** folder in your project.



MANIFEST.txt — Locked

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

{
  "manifestVersion": "2.0.0",
  "signatureType": "private",
  "signedByOrg": "m4d616e616e",
  "signedByOrgName": "m4d616e616e",
  "rootUrls": [
    "http://localhost:3000/"
  ],
  "plugin": "m4d616e616e-testplugin-panel",
  "version": "1.0.0",
  "time": 1680156161937,
  "keyId": "7e4d0c6a708866e7",
  "files": {
    "LICENSE": "c5acchbd8546e94c34aed24afe689a617627d18eed5a6c48277e48db57c23851",
    "CHANGELOG.md": "ba613d6f914b27dce9ace4d8c0cb074273c9eb6c536d8e7ac24c5ce6ae941fd0",
    "module.js.map": "350668aec9cf854b42d0c86587ccc63dfabb56a1de70f2e589fa396e8bd909ed",
    "plugin.json": "5eec92637d9e00065ac1eaca942a20c7ba7c600a138f8b7032cd2aa8dff36aab",
    "README.md": "ce73b1e1f318b3eb74dc42bca37ae79eff677ea4ae075836b280e8a513a85f1e",
    "img/logo.svg": "1defc6f7e585c67657bcfd8fddc599ee7dfa82f8674413f49fa274c2cd453ec6",
    "module.js": "5cae41450a55b5c4379c722397fe5f50517908a1c24ffdd39570e49c9f26619b"
  }
}
-----BEGIN PGP SIGNATURE-----
Version: OpenPGP.js v4.10.10
Comment: https://openpgpjs.org

wrgEARMKAAYFAmQ1JgIAI0k0fk0ManCIZucWIOTz0yW2k0d0hGNlcPN+T0xg
cIhm56NiAgjV0cbweAvZ4Nk3v0X7o02Bh2+Jj5+NPZRP/NMDv7QJj+Xe1Mpk
7CMg9uCYldXzrSx1gxqdIoYW11QaNwjFqt5/0AIJAW8375fwyP8h07fo0+BE
RnyC6ZPHhIIItdVS+2Ctr+veZy7+J6AkeK9xft9ivazp5ccI0e3in0Ej6yy0
6h1o4uTe
=20sy
-----END PGP SIGNATURE-----
```

MANIFEST.txt

## RUN the grafana-server web

After the signing process is done change the **app\_mode** to production in **defaults.ini** file and restart the **grafana-server web**.

```
1 ##### Grafana Configuration Defaults #####
2 #
3 # Do not modify this file in grafana installs
4 #
5
6 # possible values : production, development
7 app_mode = production
```

```
|sh-3.2# ./bin/grafana-server web
Grafana server is running with elevated privileges. This is not recommended
INFO [03-30|12:14:33] Starting Grafana
INFO [03-30|12:14:33] Config loaded from
[INFO [03-30|12:14:33] Path Home
[INFO [03-30|12:14:33] Path Data
[INFO [03-30|12:14:33] Path Logs
[INFO [03-30|12:14:33] Path Plugins
[INFO [03-30|12:14:33] Path Provisioning
[INFO [03-30|12:14:33] App mode production
[INFO [03-30|12:14:33] Connecting to DB
[INFO [03-30|12:14:33] Starting DB migrations
INFO [03-30|12:14:33] migrations completed
[INFO [03-30|12:14:33] Validated license token
INFO [03-30|12:14:33] Plugin registered
INFO [03-30|12:14:34] Plugin registered
INFO [03-30|12:14:34] Plugin registered
INFO [03-30|12:14:34] Profiler
WARN [03-30|12:14:34] Plugin process is running with elevated privileges. This is not recommended logger=plugin.grafana-googlesheets-datasource
INFO [03-30|12:14:34] Plugin registered
INFO [03-30|12:14:34] Plugin registered
INFO [03-30|12:14:34] Envelope encryption state
INFO [03-30|12:14:34] Query Service initialization
INFO [03-30|12:14:34] Live Push Gateway initialization
INFO [03-30|12:14:34] registering usage stat providers
INFO [03-30|12:14:34] starting to provision alerting
logger=settings version=9.3.6 commit=978237e7cb branch=HEAD compiled=2023-01-2
6T20:08:58+05:30
logger=settings file=/Users/manantandon/grafana-9.3.6/conf/defaults.ini
logger=settings path=/Users/manantandon/grafana-9.3.6
logger=settings path=/Users/manantandon/grafana-9.3.6/data
logger=settings path=/Users/manantandon/grafana-9.3.6/data/log
logger=settings path=/Users/manantandon/grafana-9.3.6/data/plugins
logger=settings path=/Users/manantandon/grafana-9.3.6/conf/provisioning
logger=settings
logger=sqlstore dbtype=sqlite3
logger=migrator
logger=migrator performed=0 skipped=551 duration=427.708µs
logger=licensing appURL=http://localhost:3000/ source=disk status=NotFound
logger=plugin.loader pluginID=input
logger=plugin.loader pluginID=m4d616e616e-testplugin-panel
logger=plugin.loader pluginID=grafana-googlesheets-datasource
logger=plugin.grafana-googlesheets-datasource enabled=false
logger=plugin.loader pluginID=m4d616e616e-smstimeseries-panel
logger=plugin.loader pluginID=m4d616e616e-temperatureseries-panel
logger=secrets enabled=true current provider=secretKey.v1
logger=query_data
logger=live.push_http
logger=infra.usagestats.collector usageStatsProvidersLen=2
logger=provisioning.alerting
```

your custom plugin gets registered