scientiamobile

"The first step in a great mobile experience"



WURFL MICROSERVICE FOR GOOGLE CLOUD PLATFORM MARKETPLACE

Support

The ScientiaMobile Enterprise Support Portal is open to all WURFL users, both commercial license holders and evaluation users. It represents the combined knowledge base for the WURFL community. Commercial licensees are invited to post questions in the forum using the account to which their licenses are associated. This may mean faster handling of those posts by ScientiaMobile's personnel.

For commercial license holders, there are tiered support levels to address a variety of business support needs. After logging into your account, commercial licensees with support options can access the Enterprise Support portal to post tickets. These tickets will receive expedited attention.

To inquire about support plans, use our License Inquiry or our General Inquiry form.

Update Notifications

If you would like to be notified of our API updates, major data updates, and other technical changes, please subscribe to our ScientiaMobile Announcements list.

scientiamobile
www.scientiamobile.com
Tel +1.703.310.6650
E-mail: sales@scientiamobile.com

Copyright © 2026 ScientiaMobile, all rights reserved. WURFL Cloud, WURFL OnSite, WURFL and, InFuze WURFL InSight and respective logos are trademarks of ScientiaMobile. Apache is the trademark of the Apache Software Foundation. NGINX is the trademark of Nginx Software Inc. Varnish is the trademark of Varnish Software AB

WURFL Microservice for Google Cloud Platform Marketplace



Using WURFL Microservice for **Google Cloud Platform** (**Basic**, **Standard** or **Professional** Edition) will allow you to run your own WURFL-based device detection service in your hosting infrastructure by deploying familiar **Google Compute Engine (GCE)** images. If you are not familiar with Google Cloud Platform and the possibilities of the Google Cloud infrastructure, **here is a good place to get started**.

WURFL Microservice Client API (WM Client)

Given an HTTP Request and a device capability name, the WURFL Client API will return the corresponding property value.

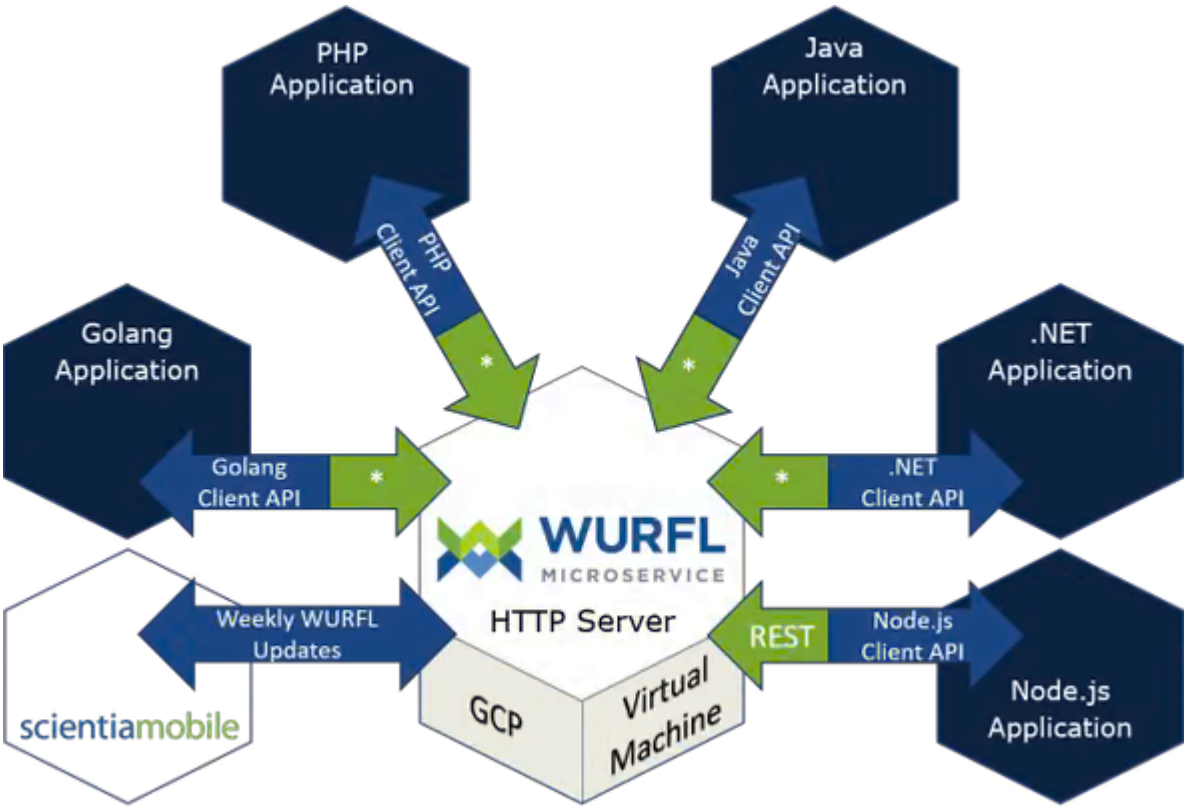
You can integrate the WURFL Microservice Standard, Basic or Pro Edition with major programming languages, including:

- .NET (C#)
- Java
- PHP
- Node.js
- Golang
- Python
- Scala
- Rust

WURFL Microservice Architecture

WURFL Microservice Server (**WM Server**) is a service that exposes its functions to the WURFL Microservice Client API (**WM Client**). The Client API depends on the availability of the WM server to work.

The following diagram explains the overall architecture of WURFL Microservice as deployed through the Google Cloud Platform Marketplace. The GCE will start an instance of the WM Server. In addition to supporting the REST interface for WM Clients, the service will periodically query ScientiaMobile for updates to the WURFL Device Description Repository to include the profiles of newly released devices and updated API logic.



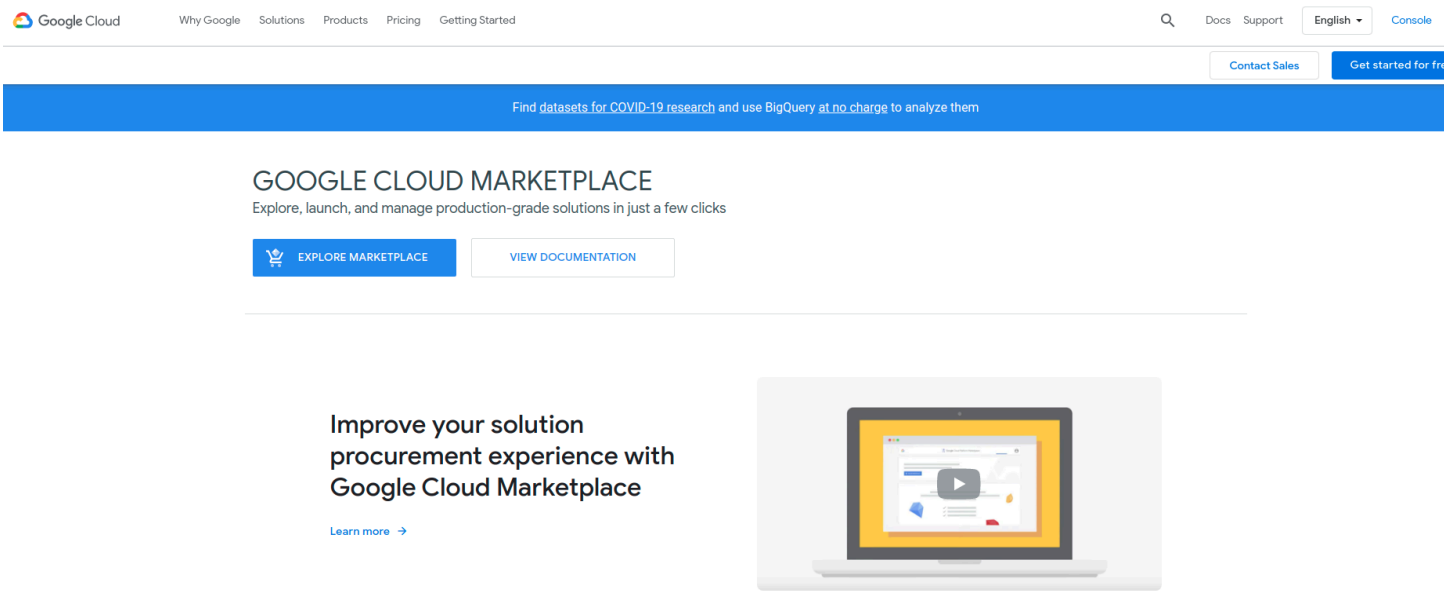
Note 1: *While the WM Client feels like a "standalone API" for most practical purposes, under the hood it requires interaction with the WURFL Microservice HTTP server (running in Google Cloud Platform) which introduces some latency (hugely mitigated by the built-in caching layer in the WM Client itself). For this reason, ScientiaMobile does not refer to the WURFL Microservice Client API (WM Client) as a "WURFL API" (that name is reserved for the WURFL OnSite APIs).*

Note 2: *The WM Server needs to have access to ScientiaMobile servers to obtain data and updates to the API logic. In high-security scenarios, access to outside internet-based services may be blocked. In those cases, your firewall may need to be configured to allow traffic between the WM Server and ScientiaMobile: outbound internet access on TCP port* `443` *and DNS is required.*

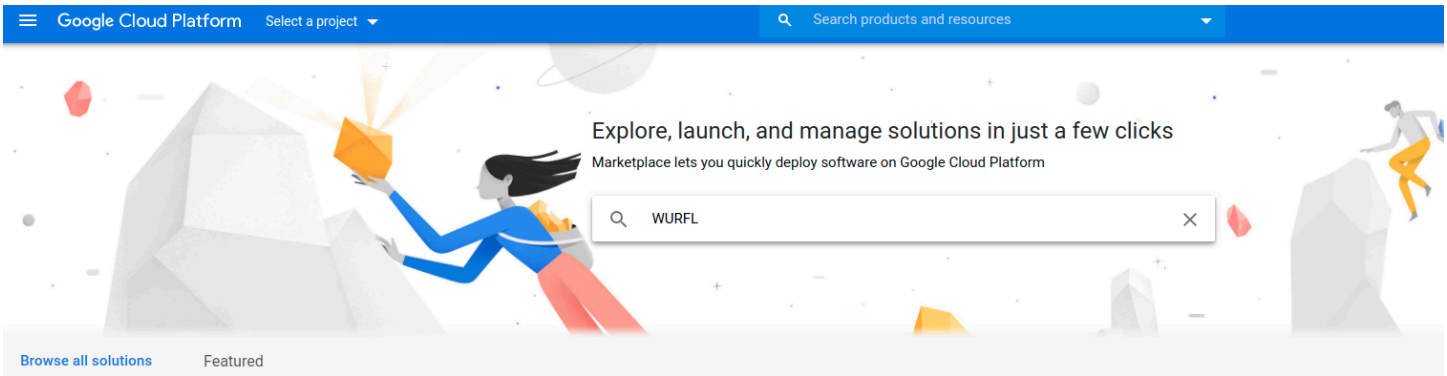
***Note 3**: communication between the WM Client API and the WM Server happens through a REST API, but this is totally transparent to users of WURFL Microservice and ScientiaMobile acknowledges this aspect of the WURFL Microservice product for sake of transparency. Unless you have a very specific use-case(s), we strongly discourage you from utilizing the REST Interface directly. Not only does the WM Client provide a caching layer (which delivers much greater performance), but ScientiaMobile reserves the right to modify the internal REST API or even to switch to different non-REST protocols in future versions of the product without notice.*

Deploying the GCE Image and Setting up the WM Client APIs

1. Go to the Google Cloud Platform Marketplace

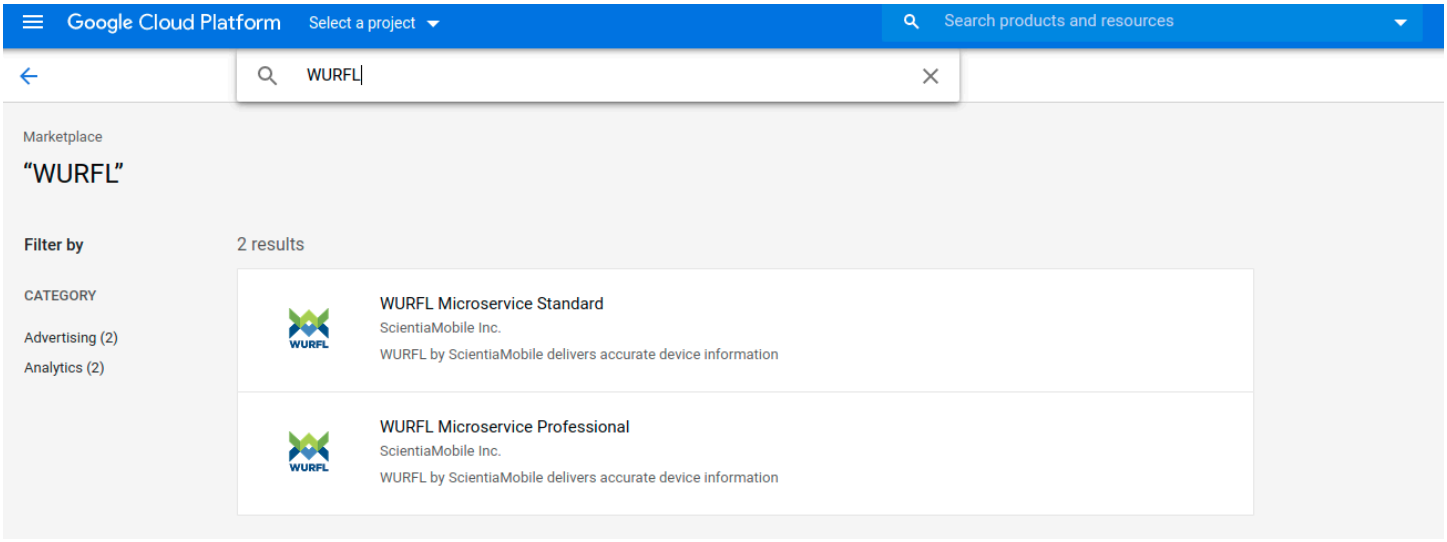


Click on the "Explore marketplace" button. You'll land in a search page.



Searching for WURFL in the searchbox will bring up the three ScientiaMobile listings. Direct links are provided below for your convenience.

Select the right Google Cloud Engine Image for your needs. The three GCEs only vary based on the capability set each one of them supports (Capabilities = Device Properties):



The three GCE images were designed with different use-cases in mind. Choose the WURFL Microservice version that fits your needs. You can pick Basic, Standard or Professional.

Basic

WURFL Microservice for GO,JAVA,dotNET,NODE.JS,PHP - Basic Edition

Standard

WURFL Microservice for GO,JAVA,dotNET,NODE.JS,PHP - Standard Edition

Professional

WURFL Microservice for GO,JAVA,dotNET,NODE.JS,PHP - Professional Edition

2. Select the Virtual machine instance you want to run your Google Cloud Image on.

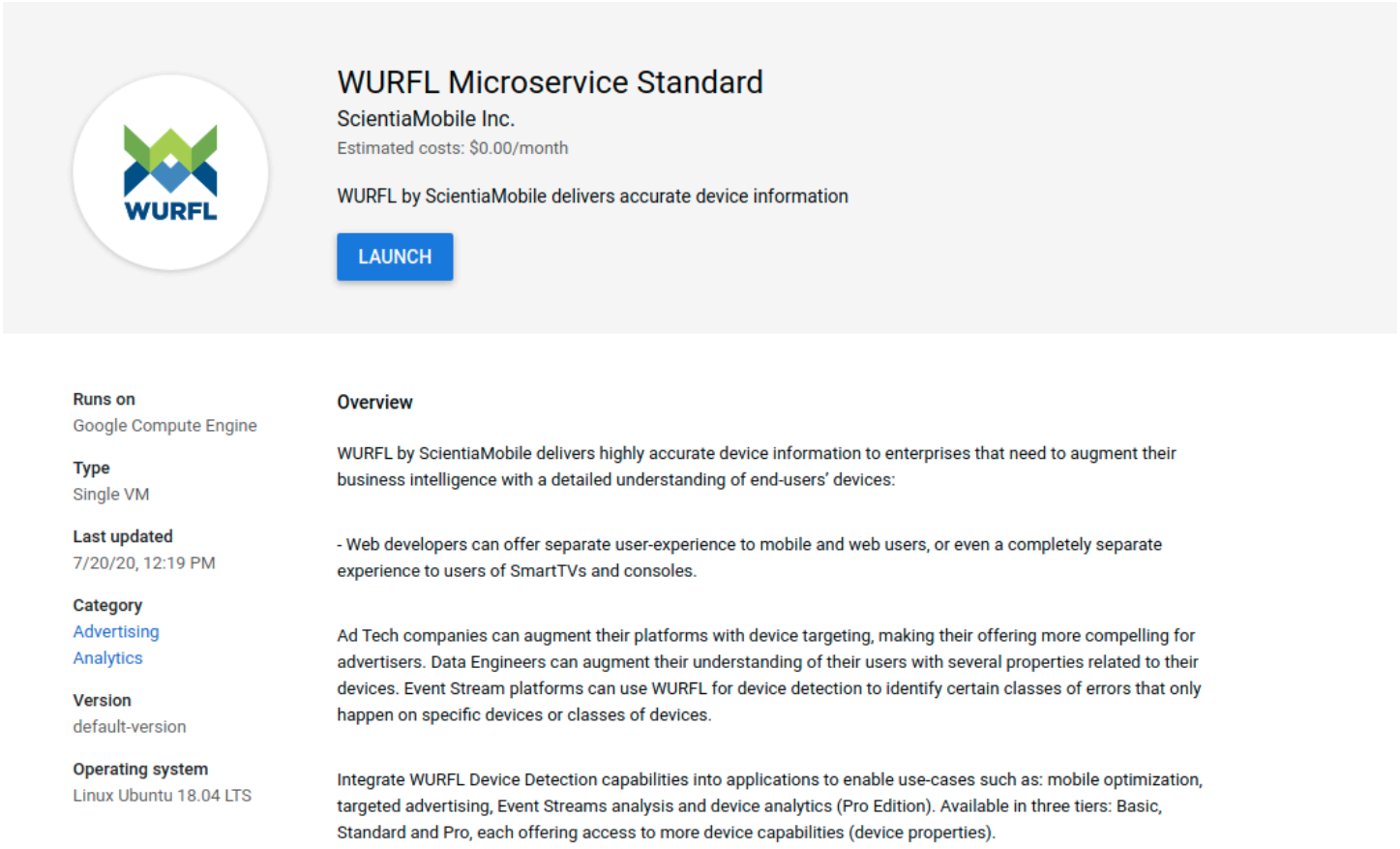
For the purposes of this document, we will pick the Standard edition, but the process is identical for the other versions (AKA "tiers"). Only the number of supported capabilities differs among GCEs.

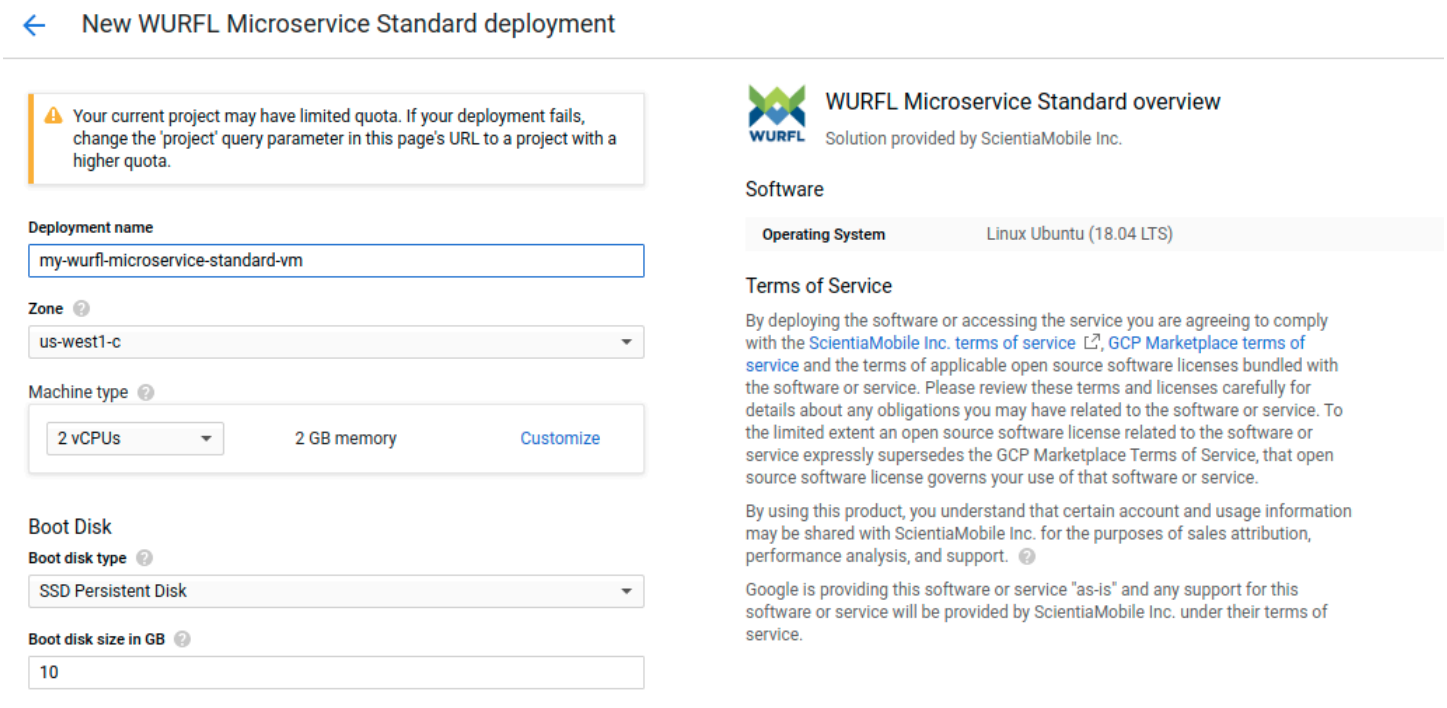
Capability Sets supported for each of the three tiers:

- Basic
- Standard

- Pro

We are now ready to launch our first Google Cloud Engine instance. To do that, click on the blue "**Launch**" button on the VM launch page. You'll land on the solution launch page.





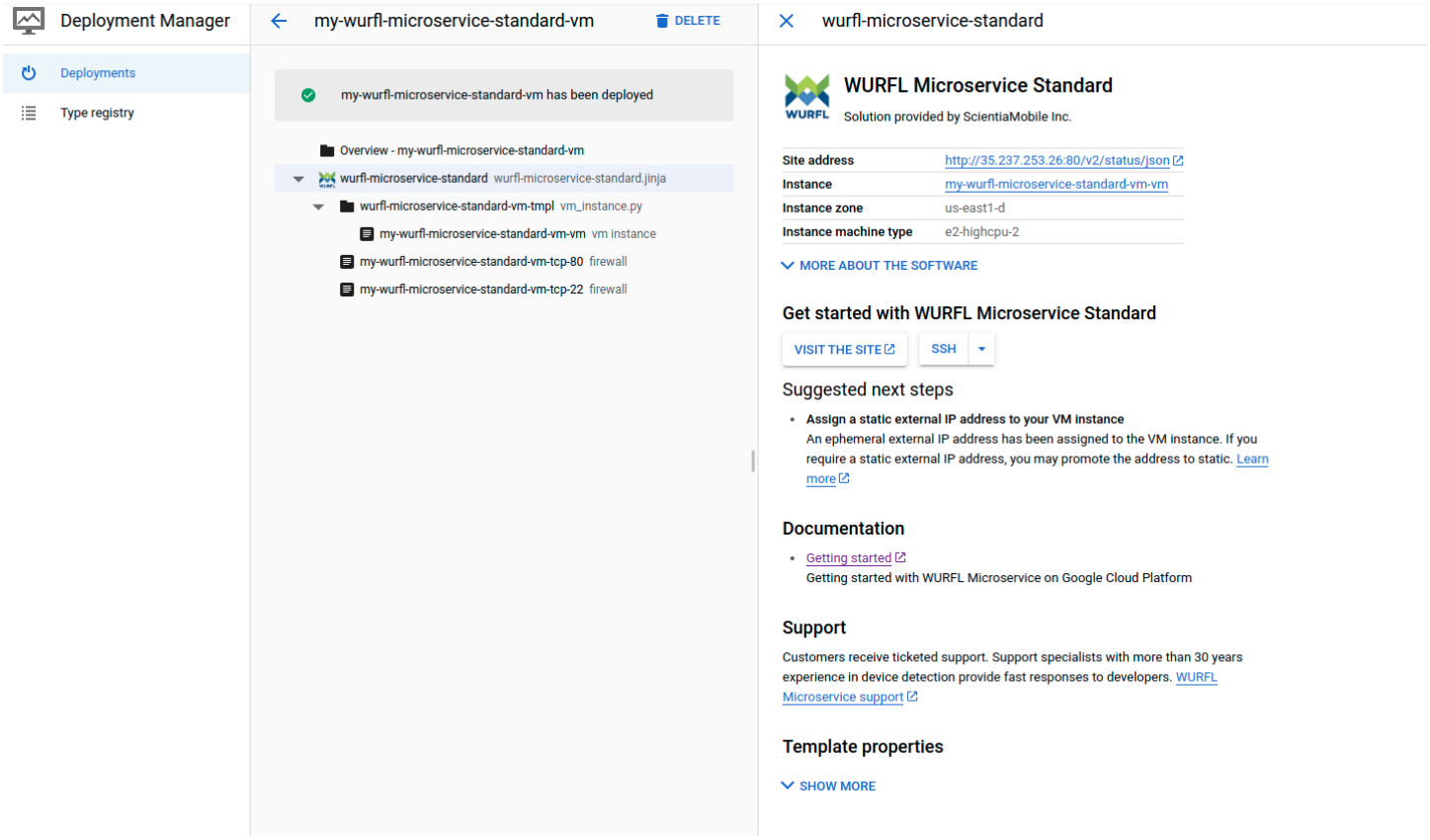
(The screenshot that follows was “split” to fit the screen.)



Choose a **deployment name**. Review and the machine type, zone, boot disk and networking options. Scientiamobile WM images are provided with default values for these options but, most of the times, you may want to modify them to fit your needs.

Then click on the blue “**Deploy**” button.

Once the GCE launch is complete, the page will show links to the WURFL Microservice server app status, SSH access to the VM, documentation and support.



> ***Note:*** *Your choice of GCE instance should depend on how much traffic you need to process through device detection. As a rule of thumb, using a single CPU, 2GB Memory will allow about 1000 lookups/sec. Actual throughput will be much higher for users of any of the APIs thanks to local caching. While an* ***N1*** *machine type is sufficient for development and testing, a larger instance is recommended for production. To get more info about how to choose your machine type check* ***GCP docs here***

Ports `80` and `22` are open by default. Port 80 is used to talk HTTP to the WURFL Microservice HTTP server, while port 22 allows SSH access to the GCE instance.

It's now time to review our instance. Click on the **site address** link. It will open a web browser page with the WM applications status.

Alternatively, you can use `curl` from a shell terminal: assuming your IP is `35.237.253.26`, the following command will confirm that the GCE VM has been deployed successfully and that the WM Server is up and running.

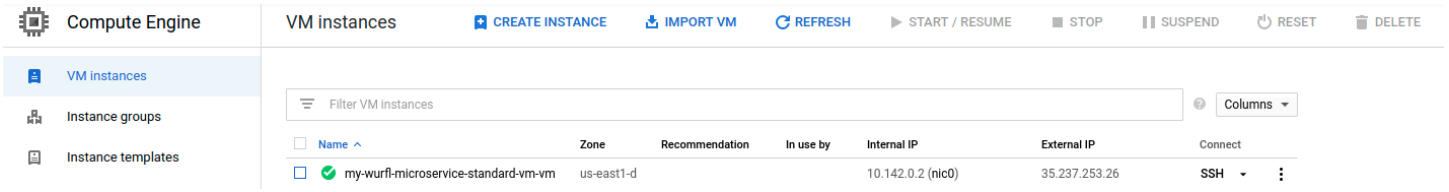
```
$ curl  http://35.237.253.26/v2/status/json
{
    "lookup_request": 0,
    "lookup_useragent": 0,
    "lookup_device_id": 0,
    "make_model_requests": 0,
    "server_info_requests": 8,
    "v1_capabilities_requests": 0,
    "not_found_404": 0,
    "server_uptime": 1172
}
```

if your instance responds with something like this, everything looks good. Your WURFL Microservice server is up and running!

The WM Client API now has a WM server to talk to. More about Client APIs later.

3. Viewing the purchased GCE under your Google Cloud Platform account.

You can acces the newly-created instance from your Google Cloud Platform Dashboard.



You will need to know the Instance IP address (`<external ip>`) to correctly configure your client API options (following step). The address can be found in your GCP dashboard.

> ***Note:*** *The configuration presented here is the simplest possible way to obtain a running instance of WURFL Microservice within Google Cloud Platform. In a real production environment, things are likely to be more complex than this and may involve making multiple instances of the product with a load balancer to distribute traffic and achieve fault tolerance. Discussion of the GCP infrastructure is outside the scope of this document.*

4. (Only for first-time WURFL Microservice for Google Cloud users) Install the WM Client API package and import the API in your project to start using it in your application.

WM Client APIs are available for the following supported languages (regardless of the capability set/tier): Go (golang), Java, .NET, Node.js and PHP.

WM Client APIs for the respective languages are provided on the ScientiaMobile public github repository:

- dotNET: https://github.com/WURFL/wurfl-microservice-client-dotnet
- Golang: https://github.com/WURFL/wurfl-microservice-client-golang
- Java: https://github.com/WURFL/wurfl-microservice-client-java

- Node.js: https://github.com/WURFL/wurfl-microservice-client-nodejs
- PHP: https://github.com/WURFL/wurfl-microservice-client-php
- Python: https://github.com/WURFL/wurfl-microservice-client-python
- Rust: https://github.com/WURFL/wurfl-microservice-client-rust

You can find the documentation for each API here:

1. Golang (GO Language)
2. Java
3. Microsoft .NET
4. PHP
5. Node.js
6. Python

Maven Central, Nuget and NPM also carry the WM Client API libraries for Java, .NET and Node.js respectively.

© 2026 ScientiaMobile Inc.
All Rights Reserved.

NOTICE: All information contained herein is, and remains the property of ScientiaMobile Incorporated and its suppliers, if any. The intellectual and technical concepts contained herein are proprietary to ScientiaMobile Incorporated and its suppliers and may be covered by U.S. and Foreign Patents, patents in process, and are protected by trade secret or copyright law. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from ScientiaMobile Incorporated.