

---

# DesignSpark Documentation

DesignSpark

Mar 31, 2022



# DESIGNSPARK CLOUD

<b>1</b>	<b>DesignSpark Cloud</b>	<b>3</b>
<b>2</b>	<b>Account</b>	<b>5</b>
<b>3</b>	<b>Metrics</b>	<b>7</b>
<b>4</b>	<b>Air Quality Project</b>	<b>17</b>
<b>5</b>	<b>Documentation Developer Guide</b>	<b>19</b>



The DesignSpark Documentation Library is used to host technical reference documentation for DesignSpark Cloud and collaborative open source projects. Documentation is written in reStructuredText (RST), hosted in GitHub repositories and generated using Sphinx, which makes it easier to update, manage and release documentation alongside code, using a “continuous documentation” paradigm.

Should you identify a mistake or otherwise room for improvement, a pull request can be submitted using the *Edit on GitHub* link top-right of the page in question.

---

**Note:** Documentation which is not hosted as part of the main docs project and instead in a GitHub repository alongside code or PCB design files, is published under `/projects/`. When navigating to such “sub-projects”, the site navigation will change to that of the sub-project. This can be a little confusing at first, but the browser *Back* button can be used to return to the main documentation site.

---



## DESIGNSPARK CLOUD

DesignSpark Cloud is software-as-a-service (SaaS) for building IoT and other data driven solutions. It is presently an invitation-only “closed alpha” and the initial focus is on supporting the *Air Quality Project*.

Please see the menu on the left for further details.



## 2.1 Registration

After an admin has created your account you will receive an e-mail invitation to complete the sign-up process, which must be completed within 7 days of receiving the e-mail. As part of which you will need to set a password and may have the option of configuring multi-factor authentication (MFA).

If you appear to have not received an e-mail, please check your spam folder.

---

**Note:** DesignSpark Cloud uses Okta for identity management, hence e-mails will be sent from okta.com and you will sometimes see this name in web pages.

---

## 2.2 Password

If you need to change your password, please visit [id.designspark.io](https://id.designspark.io), log in and select the menu by your name top-right, followed by *Settings*.

Should you forget your password, visit [id.designspark.io](https://id.designspark.io) and select the *Forgot password?* link.

<p><b>Warning:</b> If an incorrect password is entered too many times, your account will become locked and it will be necessary to raise a <a href="#">support ticket</a> to have this unlocked.</p>
--



## METRICS

The DesignSpark Metrics platform provides time series data storage and visualisation. It is built upon the Prometheus time series database (TSDB) and Grafana analytics and visualisation software.

### 3.1 Prometheus

When access to DesignSpark Metrics is provisioned you will be provided with an `instance ID` and at least one key for accessing the instance. These should be kept secret and not shared or otherwise disclosed to any third parties.

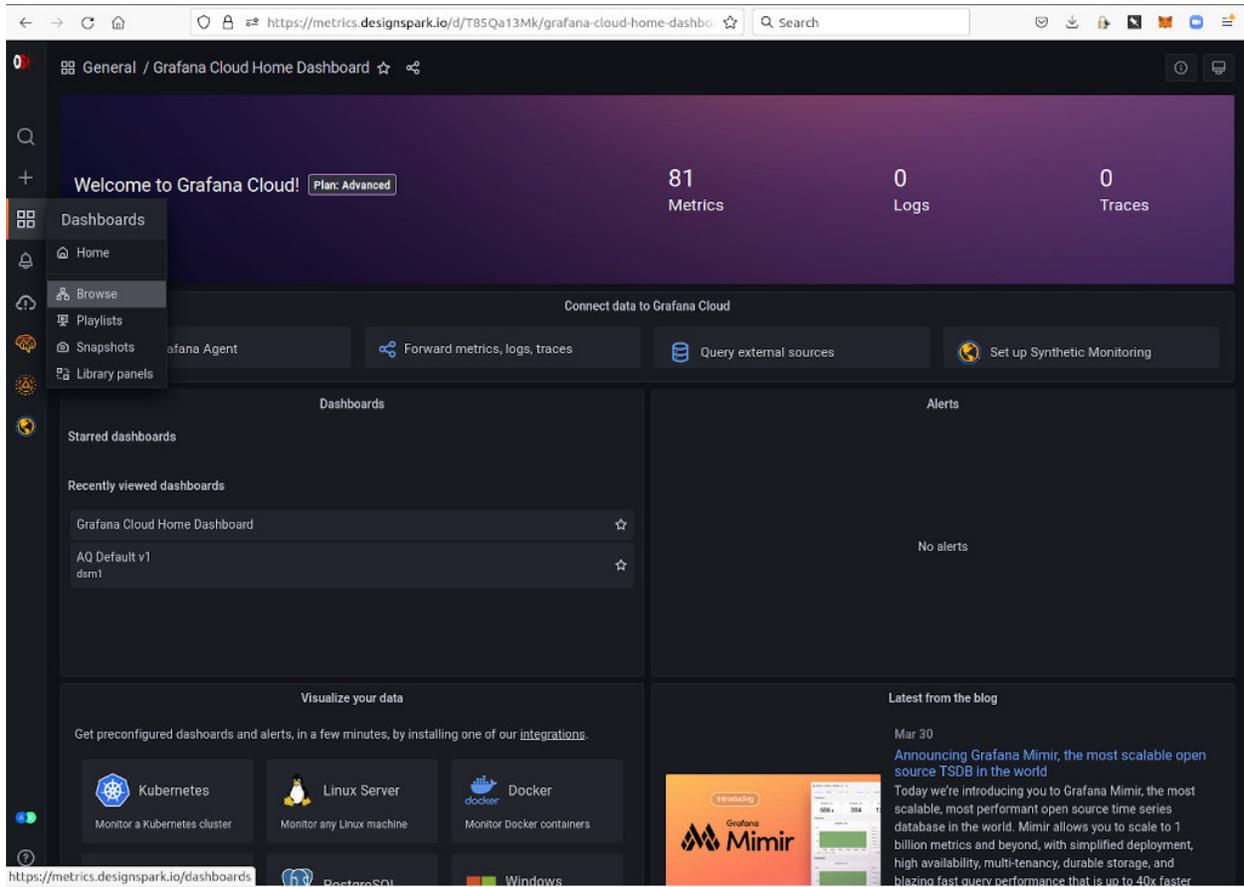
Keys can be of the `metricsPublisher (R/W)` or `Viewer (R/O)` type and it's important that the correct type is used in applications. For example, attempting to use a `Viewer` key in an application that needs to publish metrics will result in failure.

### 3.2 Grafana

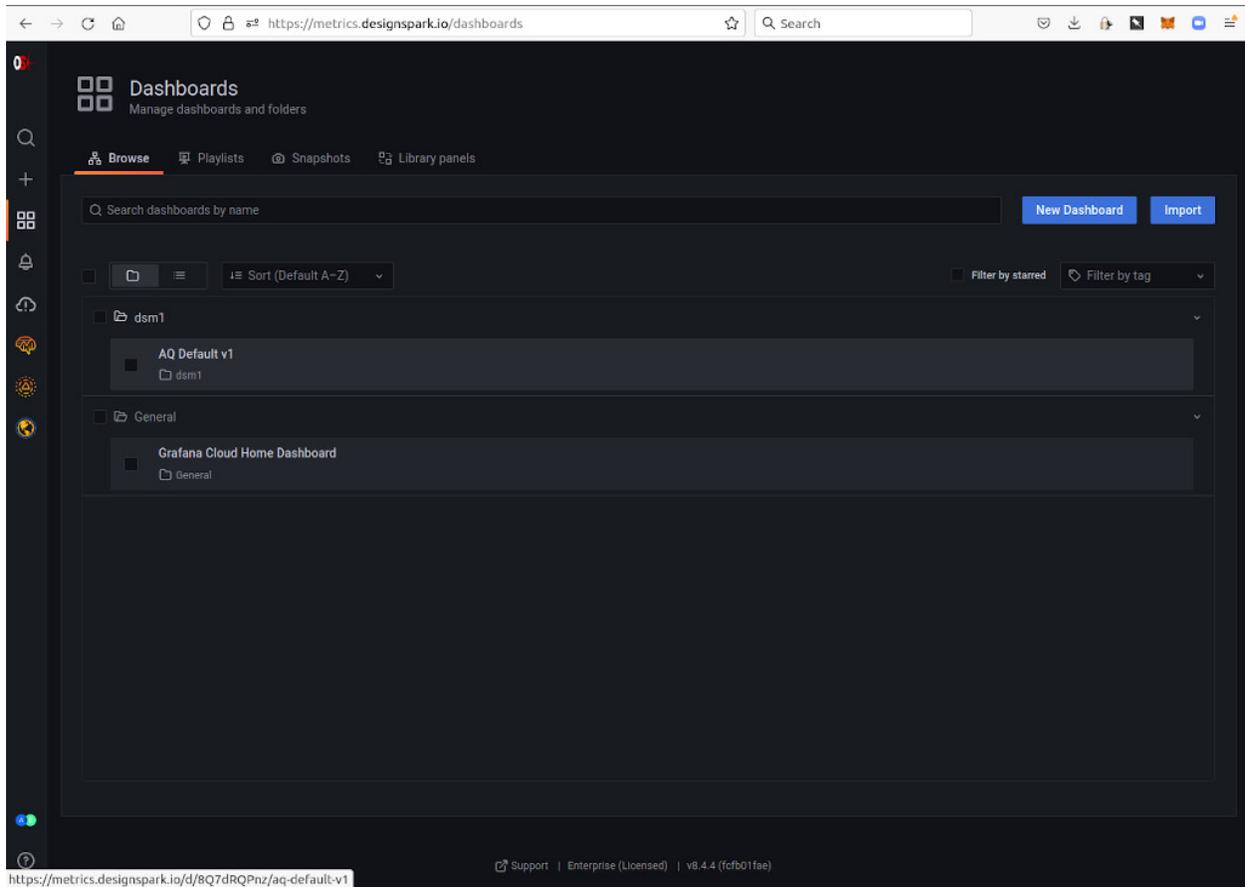
The Grafana web application is accessed via [metrics.designspark.io](https://metrics.designspark.io). A DesignSpark Cloud *account* is required to log in and your DesignSpark or RS Components website credentials will not work.

### 3.2.1 Home dashboard

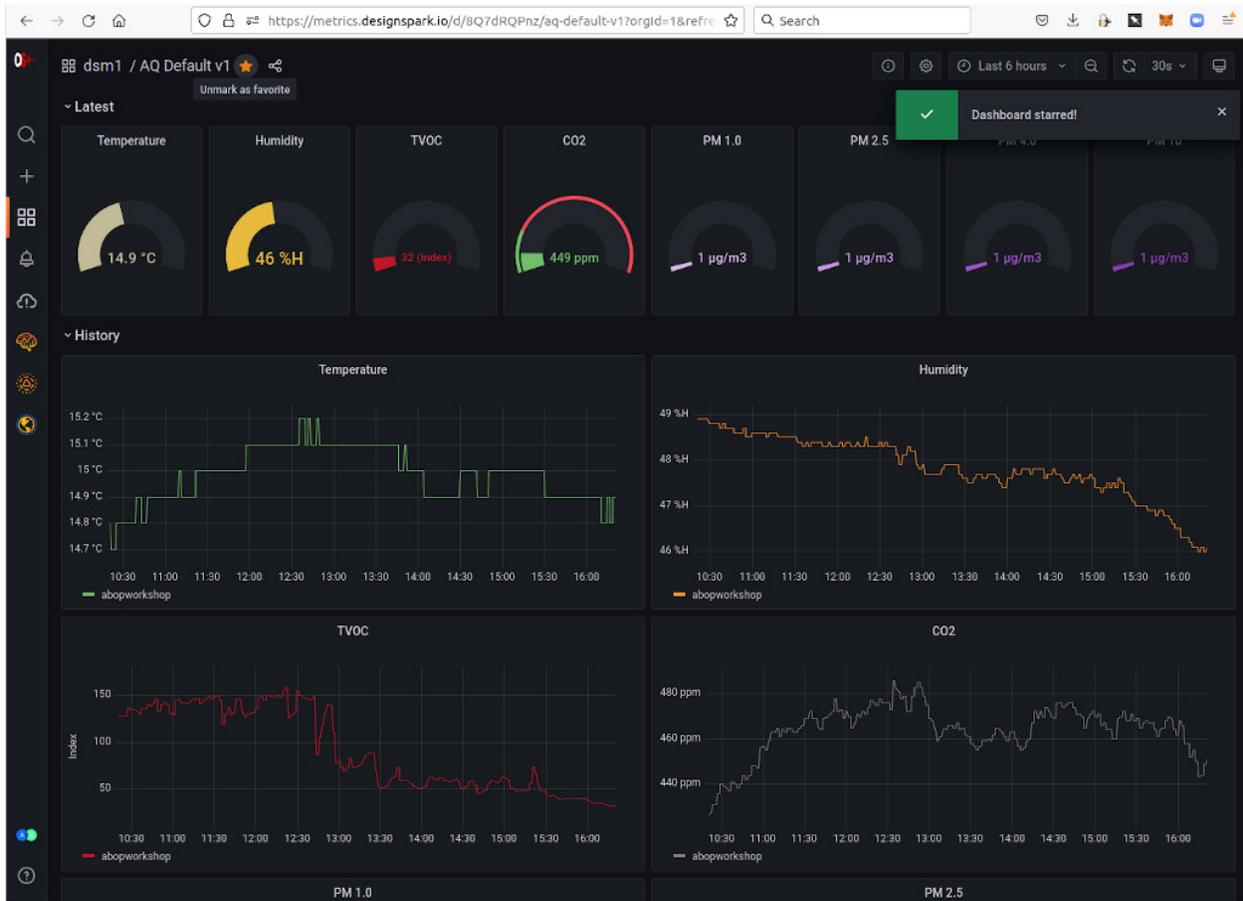
A dashboard can be set to be the home dashboard, such that this is immediately displayed whenever you log in. Next we will demonstrate how this can be done with the default *Air Quality Project* dashboard.



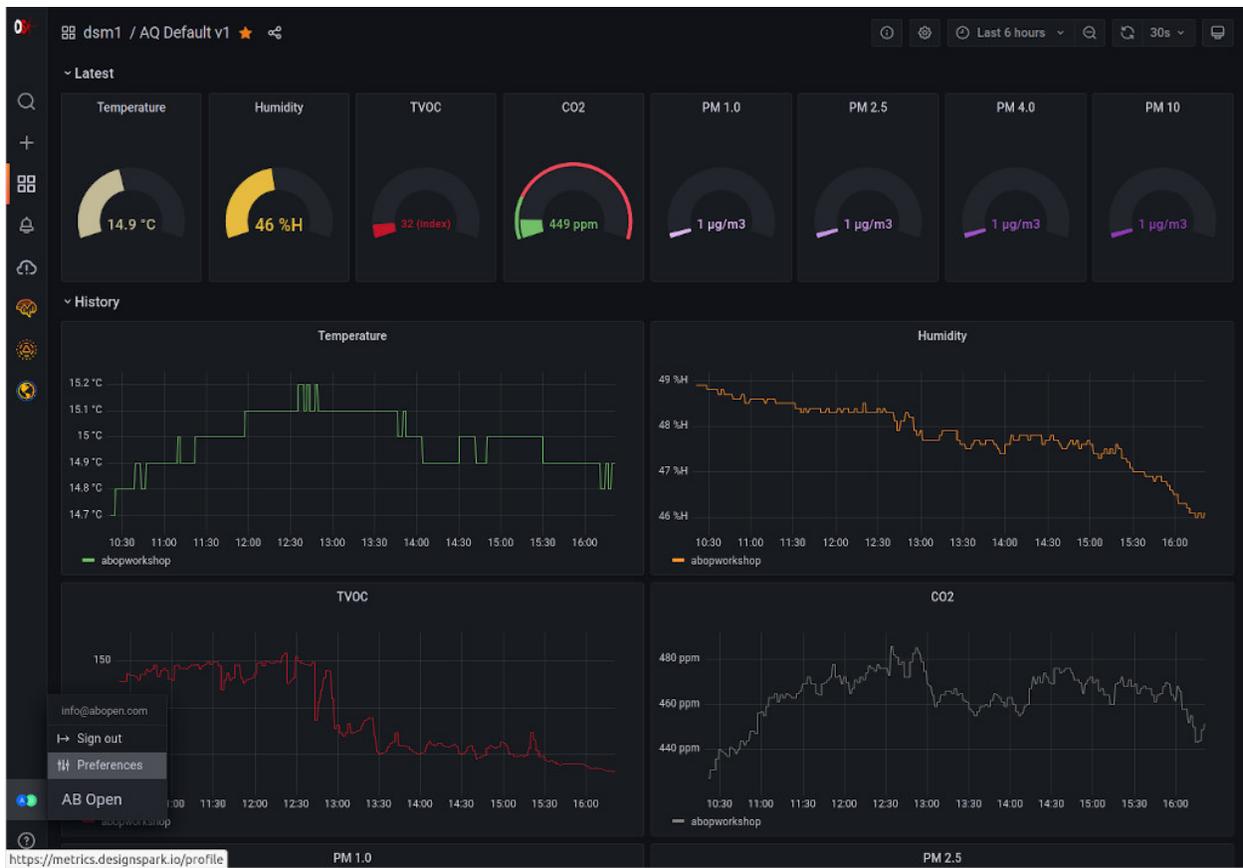
Above can be seen the default Grafana home dashboard, which displays some headline statistics and useful links. To browse our other dashboards we use the left-hand menu and select *Dashboards* (four squares icon), followed by *Browse*.



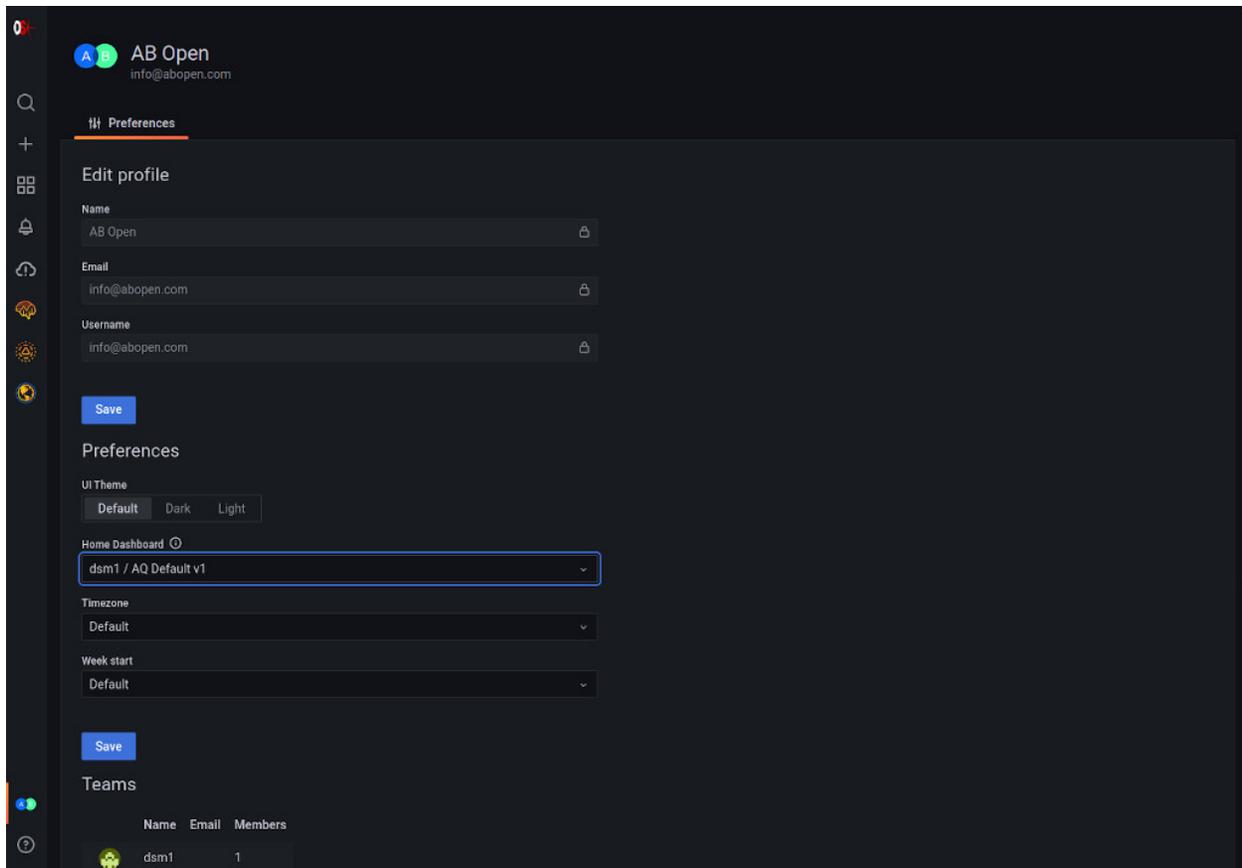
Here we can see we have a folder named after our DesignSpark Cloud UID, which for this user is *dsm1*. As an Air Quality project member, the user has been provisioned with a default dashboard named *AQ Default v1*, which we can select.



To set this as our home dashboard we need to first *star* it, by clicking on the star icon top-left.



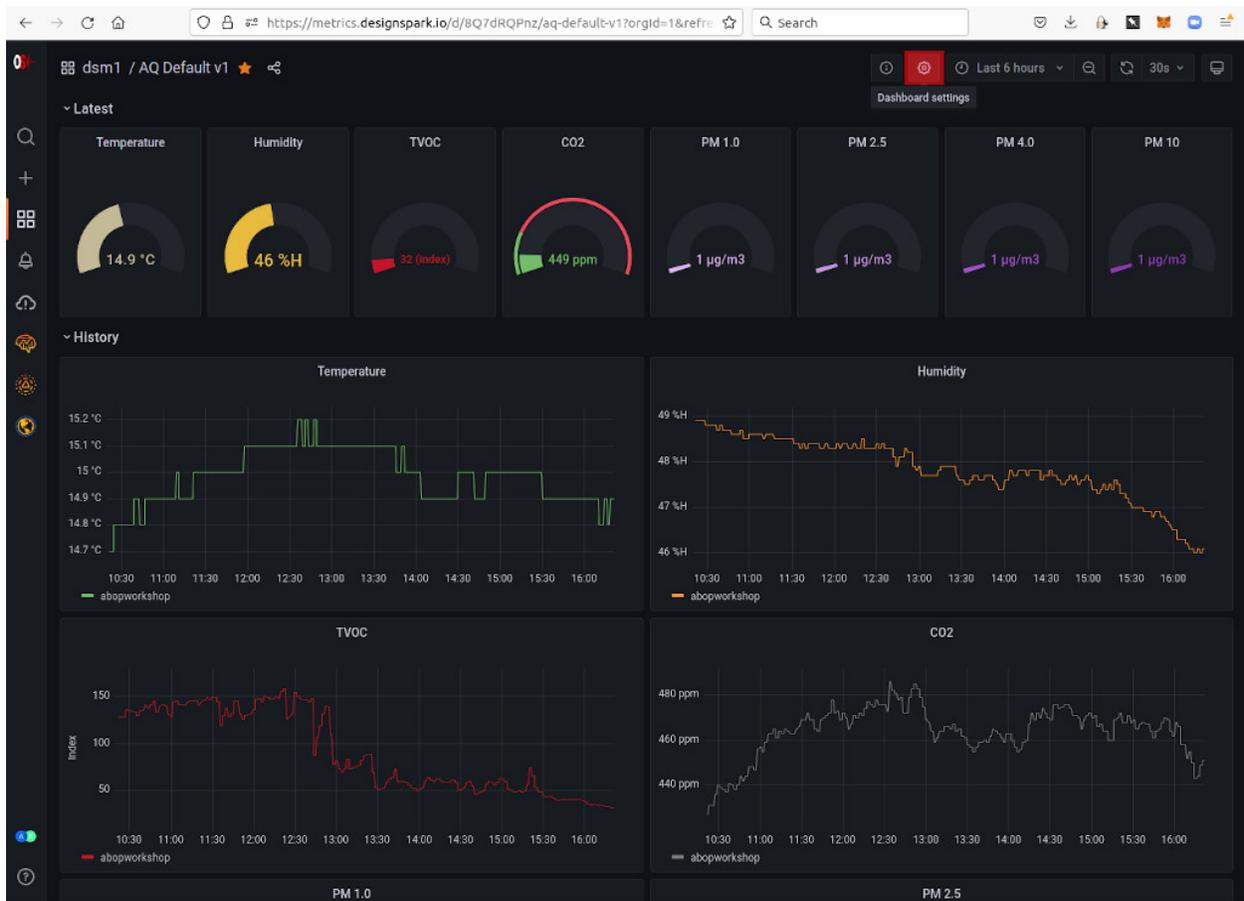
To now set our home dashboard we need to select *Preferences* via the menu on the user icon bottom-left.



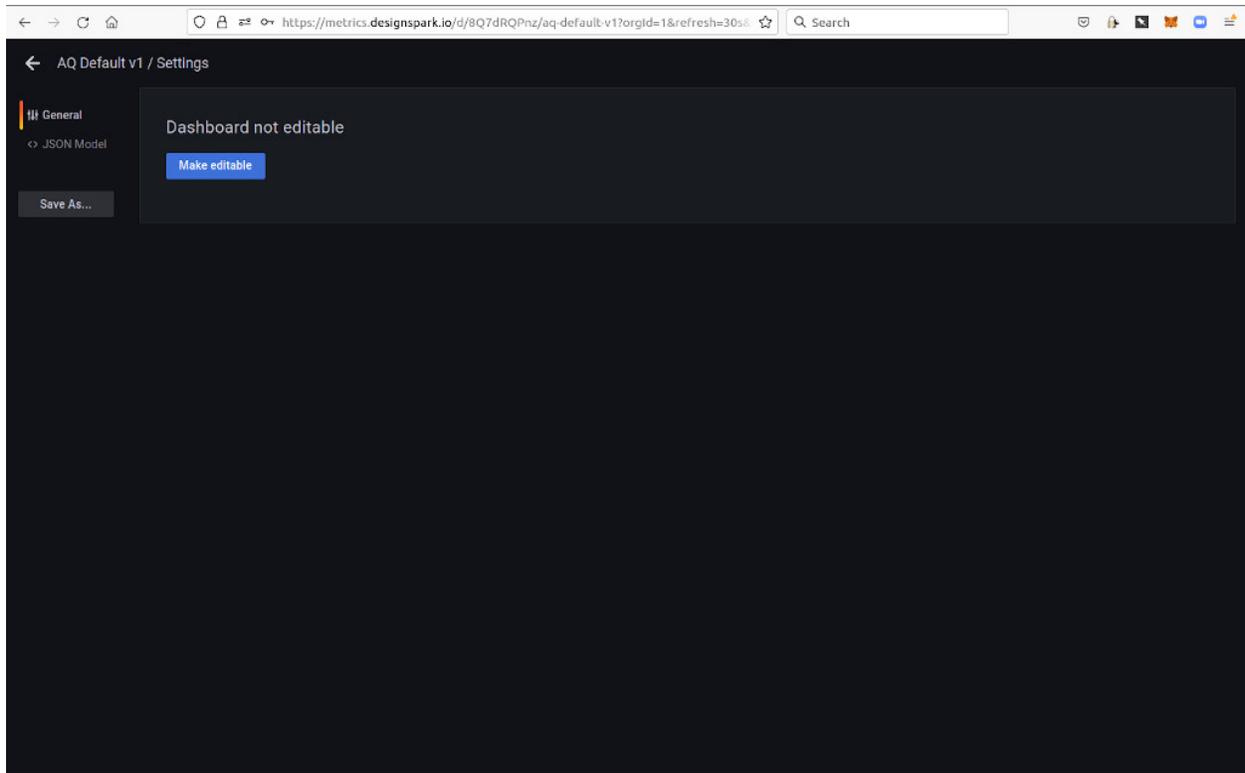
Finally, we can select the *Home Dashboard*, followed by *Save*.

### 3.2.2 Copying a dashboard

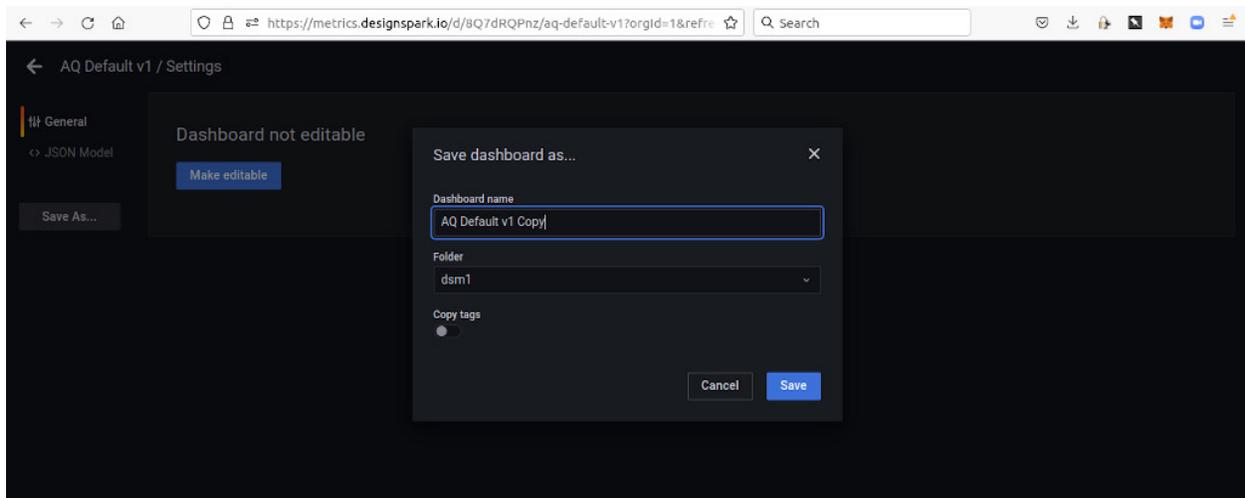
It is strongly recommended to copy default dashboards and make changes to copies, rather than the original. This way if mistakes are made, the default can be copied again. It is also possible that new default dashboards will be rolled out in future, which may involve deleting the old default dashboard.



To copy a dashboard, select *Dashboard settings* (gear icon, highlighted red).



Next, select *Save As...*



Finally, enter a new name for the dashboard and select *Save*.

### 3.2.3 Editing dashboards

For details of how to edit and create new dashboards, see the [Grafana documentation](#). However, note that anything in connection with installing software or administration tasks etc. will not be relevant.



## AIR QUALITY PROJECT

Air quality is a growing problem worldwide. DesignSpark has developed an open-source, cloud-connected, sensor platform as part of a wider programme of engineer activism. You can find out more about this and the contributor community via the [DesignSpark website](#).

In the left-hand navigation you can find links for the Air Quality Project technical reference documentation.

---

**Note:** The Air Quality project is currently operating as an invitation-only “closed beta”.

---

### 4.1 Air Quality Dashboards

The Air Quality project has both private and public dashboards. Private dashboards can only be accessed by individual users and are read-write (may be modified). Whereas public dashboards may be viewed by anyone, but are read-only (cannot be modified).

#### 4.1.1 Private

Private dashboards make use of the dedicated *Prometheus* instances that are provisioned for each DesignSpark Cloud user. When the user account is created a default private dashboard is also provisioned.

For details of how to browse dashboards, set a home dashboard, make copies and edit dashboards, see [DesignSpark Metrics documentation](#).

<p><b>Warning:</b> The default dashboard should not be edited, since a future update may delete this and provision an updated dashboard. Instead the default dashboard should be copied and the copy edited. Details can be found in the aforementioned documentation.</p>
--

#### 4.1.2 Public

Generally speaking, public dashboards will not show data from individual sensors and instead will provide statistics such as minimum, maximum and mean values. Exceptions to this include dashboards for Air Quality sub-projects, e.g. hackerspace air quality monitoring, where it may be useful to have things such as a league table with individual metrics from each shown.

*Further details will be added here when public dashboards launch.*

## 4.2 Data Model

*Stub - TBD.*

## 4.3 Taxonomy

An taxonomy is being developed so as to ensure consistency in the labelling of data gathered and avoid multiple labels being used for the same thing.

Should it be necessary to extend the taxonomy, e.g. to add a missing location, there are two routes to achieving this:

1. Submit a pull request to modify this page (see Edit on GitHub link top-right)
2. Raise a [support ticket](#) (Air Quality Project - Cloud Support)

### 4.3.1 Location

Value	Description
livingroom	Main living space
kitchen	Domestic or commercial kitchen
utilityroom	Utility room, e.g. laundry etc.
hallway	Hallway (any floor)
diningroom	Dining room
bedroom	Bedroom
office	Home or business office
attic	Attic
basement	Basement
conservatory	Conservatory
garage	Garage
greenhouse	Greenhouse
shed	Shed
workshop	Home, work or makerspace workshop
outdoor	Any outdoor location, e.g. house wall or garden etc.

### 4.3.2 Project

Value	Description
makerspace	Makerspace/hackerspace subproject.

## DOCUMENTATION DEVELOPER GUIDE

Documentation is managed, built and published using:

- [reStructuredText \(RST\)](#) content stored in GitHub repositories.
- [Sphinx](#) documentation generator.
- [Read the Docs Sphinx Theme](#) for styling.
- [Read the Docs](#) for hosting.

When changes are pushed to the git repository this triggers the Read the Docs build pipeline and the resulting HTML is hosted via their servers. However, before this is done the site should be generated locally, so as to verify formatting and avoid continually pushing, having to wait for the site to build and risking errors going live. Only when the local build looks good do we commit and push.

RST content is located alongside project materials — e.g. software sources and PCB design databases — in the same git repository. This has a number of benefits:

- The same workflow with branching and pull requests etc. can be used to manage contributions.
- Sphinx can auto-document Python code and via plugins, other programming languages that are supported by Doxygen.
- Project materials + documentation are managed and tagged together.

Obviously projects such as this one that are purely documentation, will have only the Sphinx config and content in their repo.

To find out where the source RST for a page is located, click on the Edit on GitHub link top-right to locate the GitHub repository containing this.

## 5.1 Tooling

### 5.1.1 Setup

The Python venv module is used to create a virtual environment into which the Python dependencies are installed. These are specified in requirements.txt, which is also used by the Read the Docs build pipeline.

To install git and venv on Ubuntu:

```
sudo apt install git python3-venv
```

This only needs to be done once.

Then if we wanted to edit this documentation, for example:

```
git clone git@github.com:DesignSparkRS/DSDocs.git

cd DSDocs

python3 -m venv venv

source venv/bin/activate

pip install -r requirements.txt
```

The above steps need would to be carried out for each project where we want to edit documentation.

### 5.1.2 Use

Simply edit the RST content and then to build locally:

```
make html
```

Following which the generated HTML can be found in the `_build` directory.

With project repositories that contain documentation and other outputs, such as software sources or gateway etc., please prefix the git commit message with “Docs: ” so that it is clear that the commit pertains to documentation only.

---

#### Note:

1. With every new terminal it will be necessary to source the script `venv/bin/activate` in order to activate the Python virtual environment.
2. The `pip install` command may need to be run from time to time, e.g. after a dependency has been updated or a new one has been configured.
3. If your Python virtual environment somehow gets in a mess, just delete it, re-initialise and `pip install` the dependencies again.

---

**Warning:** Do not attempt to commit the contents of the `venv` directory, as we don’t want this to be pushed to GitHub. This usually shouldn’t be possible, as the `venv` directory should be listed in `.gitignore` for all projects, but it’s worth noting this here also just in case!

## 5.2 Formatting

### 5.2.1 General

For general notes on reStructuredText (RST) formatting see:

- [Introduction to reStructuredText](#)
- [reStructuredText Primer](#)

There are various editors available with RST support, such as Visual Studio Code with the [reStructuredText plugin](#).

Notes on less obvious formatting methods are provided below.

## 5.2.2 Linking to other project documentation

While it's possible to link to the documentation for another project just as you would a regular website, it's better to use Intersphinx where this is also Sphinx-based. Benefits include that you can then use a shorter cross-reference role each time, rather than a full URL, as well there being just one place to update the mapping should the host or URL path on the server change. For details, see:

<https://docs.readthedocs.io/en/stable/guides/intersphinx.html>

If the target Sphinx project has not previously been linked to, it will likely be necessary to update `conf.py` in the project you would like to link from, in order to add a new Intersphinx mapping.

## 5.2.3 Subscript and Superscript

Subscript and superscript characters can be generated using the `:sub:` and `:sup:` roles. For example:

```
Upper limit 10\ :sup:`6`. See note\ :sub:`2`.
```

Gives:

Upper limit 10<sup>6</sup>. See note<sub>2</sub>.

---

**Note:** Interpreted text needs to be surrounded by whitespace or punctuation, which means that if we don't want a gap, e.g. between 10 and 6, or between note and 2, we need to escape the whitespace with a backslash.

---

## 5.2.4 Non-ASCII Characters

Non-ASCII characters can be inserted by using substitutions and these are defined in the file `substitutions.txt` in the root of the Sphinx project. For example, the right arrow ( $\rightarrow$ ) is inserted by entering `|rarr|`. If a character is not currently supported, simply look up the Unicode ID and add it to `substitutions.txt`.