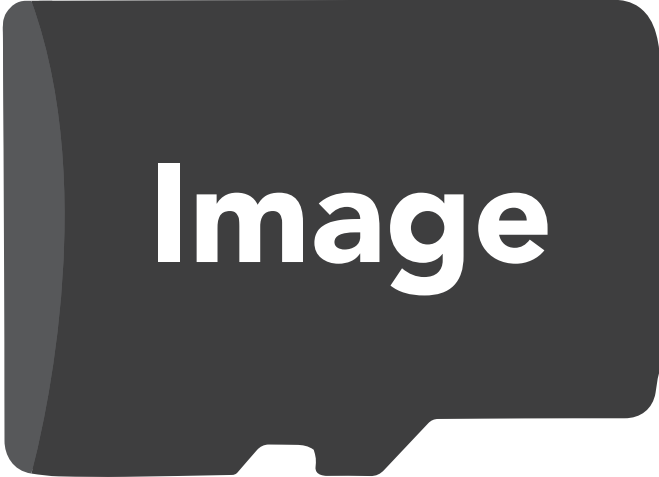


Atlas*·i*otTM
Installation guide

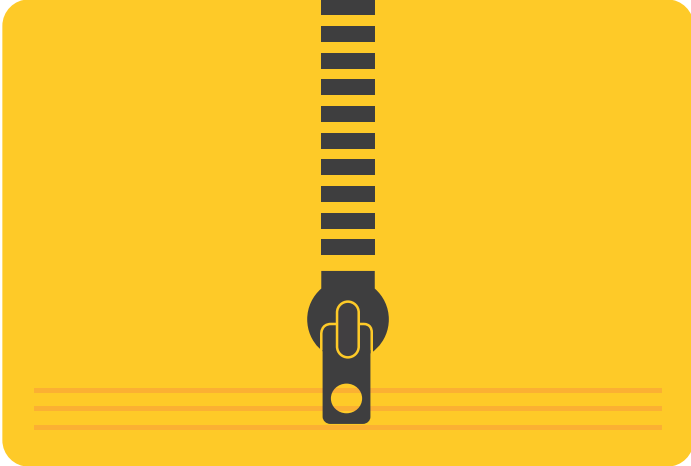
There are 2 different ways to install the Atlas iot software

Click here if you want to install using the Atlas iot image file.



~2gb

Click here if you want to install using the Atlas iot zip file.



~60mb

Preferred method of installation

Atlas iot image file

This is the preferred method of installation

Requirements

- Raspberry Pi 3b or later
- 16 Gb Micro SD card

The Atlas iot will work on a Raspberry Pi 3b or later.
The compressed file is ~2gb and extracts to ~4.66gb.

Download the image file from the Atlas Scientific website or by clicking [HERE](#) and extract the compressed file. You will need to mount the *.img file onto an SD card.

To do so, you will need to use:

Windows - Win32 Disk imager

Mac - Raspberry Pi Imager*

**Atlas Scientific has not tested this software, this is what is recommended by the Raspberry Pi community.*



We recommend using class 10 SD / UHS-I or higher speed Micro SD card, with a minimum of 16GB.

For more information on how to mount the *.img file to the SD card

Windows - [click here](#)

Mac - [click here](#)

You can now insert the SD card into the Raspberry Pi and power it up.

Installation Complete!

Atlas iot zip file

Prepare the Raspberry Pi

Requirements

- Raspberry Pi 3b or later
- 16 Gb Micro SD card

Download and Install Raspberry Pi OS (AKA Raspian)

Download and install the Raspberry Pi OS (32 bit) onto an SD Card per the instructions here:

<https://www.raspberrypi.org/documentation/installation/installing-images/>

Make sure you enable I2C

Install the SD card and boot your Raspberry Pi

Step 1 Download Atlas iot Software

Download the application from the Atlas Scientific web site or by clicking [HERE](#) and unzip into a temporary directory on your computer.

Step 2 Install Atlas iot Software

While logged in as an admin (*we'll assume Pi is still the username for this example*) create a directory for the application:

```
mkdir AtlasIoTCore
```

Using SSH or or similar, copy the contents of the directory you unzipped to the directory you just created on the RPi (~/.AtlasIoTCore)

Change directory to the directory on the RPi (cd ~/.AtlasIoTCore)

Change the permissions on the AtlasIoTCore application

```
chmod 777 AtlasIoTCore
```

Step 3 Run the program

To Run the program type:

```
./AtlasIoTCore -urls "http://*:8080"
```

This will start the program, and make the user interface available on port 8080 of your Raspberry Pi

Installation Complete!

Recommendation

While the above will run the program, every time you reboot, you would have to run the program again. You probably want to install something like NGINX and setup AtlasIoTCore as a daemon

Host by Nginx with auto-restart capability

First, install and start Nginx:

```
sudo apt-get install nginx
sudo /etc/init.d/nginx start
```

Open Nginx config file:

```
sudo nano /etc/nginx/sites-available/default
```

Replace its content with:

```
server {
    listen      80 default_server;
    server_name _;
    location / {
        proxy_pass      http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection keep-alive;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

server_name is set to `_;` means I don't limit domain names, just accept any hostnames.
proxy_pass is the default Kestrel HTTP endpoint, which runs our ASP.NET Core website.

Check and apply the config file:

```
sudo nginx -t  
sudo nginx -s reload
```

Now we have one last step to go, make dotnet process auto-restart in case your code accidentally blows up. To do that, we will create a systemd service.

```
sudo nano /etc/systemd/system/AtlasIoT.service
```

```
[Unit]  
Description=AtlasIoT Service  
  
[Service]  
WorkingDirectory=/home/pi/AtlasIoTCore  
ExecStart=/home/pi/AtlasIoTCore/AtlasIoTCore  
Restart=always  
# Restart service after 10 seconds if the dotnet service crashes:  
RestartSec=10  
KillSignal=SIGINT  
SyslogIdentifier=AtlasIoT  
User=pi  
Environment=ASPNETCORE_ENVIRONMENT=Production  
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false  
  
[Install]  
WantedBy=multi-user.target
```

Note, we can only use absolute path in systemd configuration.
Register and start the service:

```
sudo systemctl enable AtlasIoT.service  
sudo systemctl start AtlasIoT.service  
sudo systemctl status AtlasIoT.service
```

Now, you have the auto start and failure restart capability.
Try to reboot your Raspberry Pi, the website will auto start!