

Exporters

Pushgateway

The [pushgateway](#) exporter will allow you to export data for a watcher to a Prometheus gateway. To install the exporter, you will need to install its dependencies:

```
pip install watchme[exporter-pushgateway]
```

The dependencies include the [prometheus client](#).

Quick Start

Setup a gateway

```
$ docker run -d -p 9091:9091 prom/pushgateway
```

Create and add the exporter to an existing task. The only required parameter is the url for the gateway (above we use `http://localhost:9091`).

```
$ watchme add-exporter <watcher> exporter-pushgateway  
url@http://localhost:9091 type@pushgateway task-<task>
```

Detailed Start

1. Setup a Gateway

To set up a gateway, you can find full instructions (section “Run It”) at <https://github.com/prometheus/pushgateway>. A brief summary using Docker is provided here. First, bring up the gateway:

```
$ docker run -d -p 9091:9091 prom/pushgateway
```

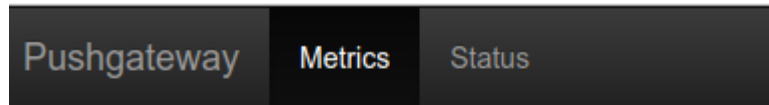
The image will pull and run, exposing the gateway on port 9091. First use `docker ps` to ensure that it’s running without issue:

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
	PORTS	NAMES	

```
fd6e3d3fb8db      prom/pushgateway  "/bin/pushgateway"  7 seconds ago
Up 6 seconds      0.0.0.0:9091->9091/tcp  gifted_merkle
```

Then, open your browser to <http://localhost:9091/>. You should see an empty portal.



Code Community © Prometheus Authors 2014

It's a fairly boring interface, but that's okay - we will fix it up when we push data to it!

2. Create a Watcher

Next, let's create a watcher that will collect the data that we want to push to the exporter. Imagine that we are interested in monitoring the

temperate for a region. Let's call the watcher "weather-watcher"

```
$ watchme create temp-watcher
Adding watcher /home/vanessa/.watchme/temp-watcher...
Generating watcher config /home/vanessa/.watchme/temp-watcher/watchme.cfg
```

This is an empty watcher in that there aren't any tasks to run.

3. Create a Task

Let's create a task to get the temperature for Luxembourg.

```
$ watchme add-task temp-watcher task-luxembourg
url@https://www.accuweather.com/en/lu/luxembourg/228714/weather-
forecast/228714 selection@.local-temp get_text@true func@get_url_selection
type@urls regex@[0-9]+

[task-luxembourg]
url = https://www.accuweather.com/en/lu/luxembourg/228714/weather-
forecast/228714
selection = .local-temp
get_text = true
func = get_url_selection
regex = [0-9]+
```

```
active = true
type = urls
```

Above, we create a task called “task-luxembourg” that will go to the URL with the weather, select the class “.local-temp” on the page, get the text for the class, and filter to the regular expression for one or more numbers (`[0-9]+`). If you are interested in these details, see the [urls watcher tasks](#).

Test the task

We haven’t scheduled it to run, but we might try testing it out to make sure that it runs as we expect.

```
$ watchme run temp-watcher task-luxembourg --test
Found 1 contender tasks.
[1/1] |=====| 100.0%
{
  "task-luxembourg": [
    "54"
  ]
}
```

The `--test` flag will ensure that the data is not saved and written to git. The above worked great, because we see that the temperature is 54.

4. Add an Exporter

Let's now add an exporter. An exporter is like a task in that it can be activated (or not), and you can define more than one for any particular watcher. If you want to see the exporters available:

```
$ watchme list --exporters
watchme: exporters
pushgateway
```

Adding an exporter is similar to adding a task! The command asks you to name the exporter, and the watcher to which you are adding it:

```
$ watchme add-exporter <watcher> <exporter>
```

And you can follow these labels by any parameters required for the exporter. For the pushgateway exporter, along with specifying the type, you are only required one parameter (url) that must start with http.

```
$ watchme add-exporter temp-watcher exporter-pushgateway
url@http://localhost:9091 type@pushgateway
```

Along with seeing it on the screen, you can inspect the watcher to see that the exporter is added:

```
$ watchme inspect temp-watcher
[watcher]
active = false
type = urls

[task-luxembourg]
url = https://www.accuweather.com/en/lu/luxembourg/228714/weather-
forecast/228714
selection = .local-temp
get_text = true
func = get_url_selection
regex = [0-9]+
active = true
type = urls

[exporter-pushgateway]
url = http://localhost:9091
active = true
type = pushgateway
```

But notice how the exporter isn't added to the task? We can add it manually:

```
# watchme add-exporter <watcher> <exporter> <task>
$ watchme add-exporter temp-watcher exporter-pushgateway task-luxembourg
```

Or you could have added it when you first created the exporter. Let's show you the command to remove the exporter:

```
$ watchme remove temp-watcher exporter-pushgateway
Removing exporter-pushgateway from task-luxembourg
exporter-pushgateway removed successfully.
```

And then add it again, not only to the watcher but also to the task, with one command!

```
$ watchme add-exporter temp-watcher exporter-pushgateway
url@http://localhost:9091 type@pushgateway task-luxembourg
[exporter-pushgateway]
url = http://localhost:9091
active = true
type = pushgateway
```

You can then inspect the watcher to see the full addition (note that “exporter-pushgateway” is added to exporters for task-luxembourg:

```
vanessa@vanessa-ThinkPad-T460s:~/Documents/Dropbox/Code/Python/watchme$
watchme inspect temp-watcher[watcher]
active = false
type = urls

[task-luxembourg]
url = https://www.accuweather.com/en/lu/luxembourg/228714/weather-
forecast/228714
selection = .local-temp
get_text = true
func = get_url_selection
```



```
regex = [0-9]+
active = true
type = urls
exporters = exporter-pushgateway

[exporter-pushgateway]
url = http://localhost:9091
active = true
type = pushgateway
```

5. Run the Exporter

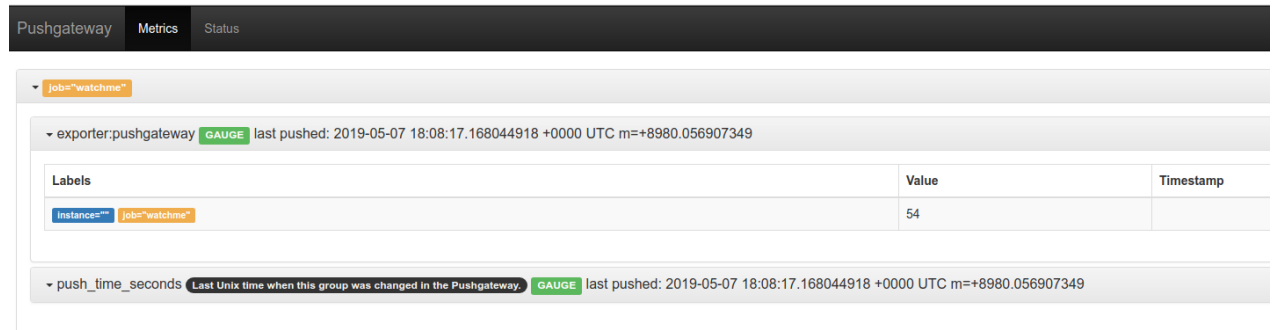
We can see that pushgateway is active, so when we run the task, it should push data there. To do it fully, we need to activate the watcher:

```
$ watchme activate temp-watcher
[watcher|temp-watcher] active: true
```

And then run the task.

```
$ watchme run temp-watcher task-luxembourg
Exporting list to exporter-pushgateway
```

If all goes well, you can return to the web interface and see your data added!



The screenshot shows the Pushgateway Metrics page. At the top, there are tabs for 'Pushgateway', 'Metrics', and 'Status'. Below the tabs, there is a dropdown menu showing 'job=""watchme""'. Underneath, there is a section for 'exporter.pushgateway' with a 'GAUGE' status and a 'last pushed' timestamp. A table with three columns: 'Labels', 'Value', and 'Timestamp' is displayed. The table has one row with the label 'instances=""' and the value '54'. Below the table, there is another section for 'push_time_seconds' with a 'GAUGE' status and a 'last pushed' timestamp.

Labels	Value	Timestamp
instances=""	54	

Great job! You can learn more about the use cases for [push_gateway](#) here.