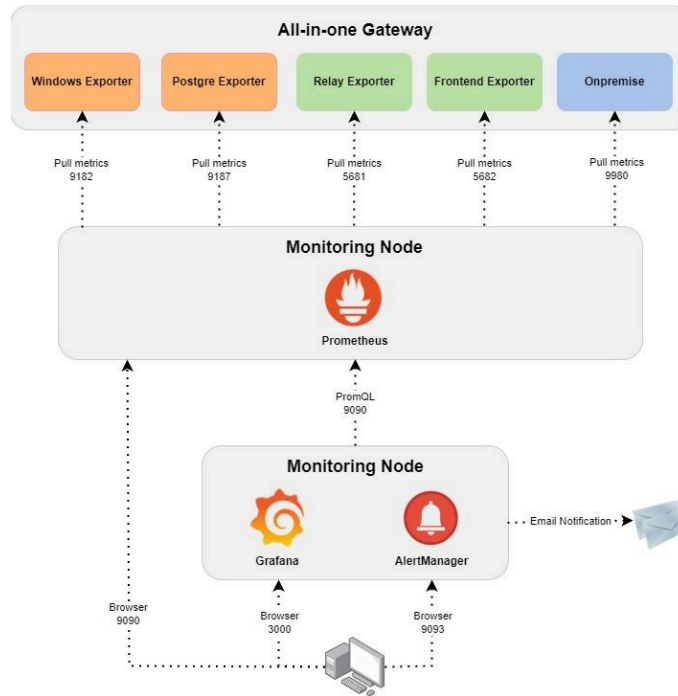


All-in-one Gateway Monitoring Setup Guide v1.0

1 Architecture



2 Preparation

2.1 WinSW

2.1.1 Introduction

Windows Service Wrapper (WinSW) is a binary, which can be used to wrap and manage custom executables as Windows services. The project has an MIT License, hence it can be used just as a binary even in commercial systems.

WinSW offers executables for .NET Framework 2.0, 4.0 and 4.6.1. It can run on Windows platforms which have these versions of .NET Framework installed. For systems without .NET Framework, the project provides native 64-bit and 32-bit executables based on .NET Core 3.1.

2.1.2 Download WinSW

Go to <https://github.com/winsw/winsw/releases> and download the suitable WinSW according to your .NET version.

.NET versions:

Win10 later, Win Server 2016 later.	• WinSW.NET461.exe
Win7/Win8/Win Server 2012/2008 R2 SP1	• WinSW.NET4.exe
Win Server 2008	• WinSW.NET2.exe

2.2 Exporter

2.2.1 Introduction

These exporters are the components that collect monitoring data and make it available externally through the Prometheus monitoring format, collecting data from On-Prem Gateway and converting it into a format supported by Prometheus. Prometheus periodically calls the metrics data interface provided by the Exporters to obtain the data.

2.2.2 Download Windows Exporter

Go to https://github.com/prometheus-community/windows_exporter/releases and download the suitable windows_exporter.

i Tips

- If you need basic auth, please download windows_exporter.exe.

- If you do NOT need basic auth, please download windows_exporter.msi.

2.2.3 Download Postgres Exporter

Go to https://github.com/prometheus-community/postgres_exporter/releases and download the suitable postgres_exporter.

2.2.4 Download Relay Export and Frontend Exporter

Download exporter.zip from [this link](#) and unzip exporter.zip. The folder should include five files

- exporter.exe
 - Copy exporter.exe to {InstallDir}.
- relay_exporter.yaml
 - Edit relay_exporter.yaml and replace %InstallPATH% with your actual {InstallDir} on filename: '%InstallPATH%\log\frontend_statistics.last.log'. Then save the file.
- frontend_exporter.yaml
 - Edit frontend_exporter.yaml and replace %InstallPATH% with your actual {InstallDir} on filename: '%InstallPATH%\log\frontend_statistics.last.log'. Then save the file.
- RelayExporterservice.xml
- FrontendExporterService.xml

Tips

The default {InstallDir} : C:\Program Files (x86)\Splashtop\Splashtop Remote\Splashtop Gateway

3 Onpremise

- Go to {InstallDir} and edit config.yaml.
- Add the below parameters at the top layer to the file to enable metrics.

```
1 Metrics:
2   Enable: true
3   Addr: 0.0.0.0:9980
4   Period: 15
```

Tips

- Addr
 - /metrics will listen on tcp 0.0.0.0:9980
- Period
 - Prometheus call /metrics every 15 seconds

- Save the file config.yaml.
- Run Command Prompt as administrator, then run the below script to restart Onpremise service.

```
1 net stop OnpremiseSvr
2 net start OnpremiseSvr
```

4 Postgresql

4.1 Postgres configuration

- Go to {InstallDir}\Data and edit postgresql.conf.
- Add the below section to the postgresql.conf.

```
1 shared_preload_libraries = 'pg_stat_statements'
2 pg_stat_statements.max = 1000
3 pg_stat_statements.track = all
```

Tips

- Remove "#" from the beginning of shared_preload_libraries.
- Add these lines
 - "pg_stat_statements.max = 1000"
 - "pg_stat_statements.track = all"

- Save the file postgresql.conf.
- Go to {InstallDir}\Data and edit pg_hba.conf.
- Add the below section to the section IPv4 local connections.

```
1 # IPv4 local connections:
2 host all postgres 127.0.0.1/32 md5
3 host all postgres_exporter 127.0.0.1/32 md5
```

- Run Command Prompt as administrator, then run the below script to restart PostgreSQL service.

```
1 net stop OnpremiseSvr
```

```
2 net start OnpremiseSvr
```

- Go to `{InstallDir}` and create `prepare-postgres-exporter.sql`.
- Copy the below information to `prepare-postgres-exporter.sql` and change the `password` on line 23 with your actual password.

```
1 create extension if not exists pg_stat_statements;
2
3 -- To use IF statements, hence to be able to check if the user exists before
4 -- attempting creation, we need to switch to procedural SQL (PL/pgSQL)
5 -- instead of standard SQL.
6 -- More: https://www.postgresql.org/docs/9.3/plpgsql-overview.html
7 -- To preserve compatibility with <9.0, DO blocks are not used; instead,
8 -- a function is created and dropped.
9 CREATE OR REPLACE FUNCTION __tmp_create_user() returns void as $$
10 BEGIN
11     IF NOT EXISTS (
12         SELECT
13             FROM pg_catalog.pg_user
14             WHERE username = 'postgres_exporter') THEN
15         CREATE USER postgres_exporter;
16     END IF;
17 END;
18 $$ language plpgsql;
19
20 SELECT __tmp_create_user();
21 DROP FUNCTION __tmp_create_user();
22
23 ALTER USER postgres_exporter WITH PASSWORD 'password'; -- change me
24 ALTER USER postgres_exporter SET SEARCH_PATH TO postgres_exporter,pg_catalog;
25
26 -- If deploying as non-superuser (for example in AWS RDS), uncomment the GRANT
27 -- line below and replace <MASTER_USER> with your root user.
28 -- GRANT postgres_exporter TO <MASTER_USER>;
29 CREATE SCHEMA IF NOT EXISTS postgres_exporter;
30 GRANT USAGE ON SCHEMA postgres_exporter TO postgres_exporter;
31 GRANT CONNECT ON DATABASE postgres TO postgres_exporter;
32
33 CREATE OR REPLACE FUNCTION get_pg_stat_activity() RETURNS SETOF pg_stat_activity AS
34 $$ SELECT * FROM pg_catalog.pg_stat_activity; $$
35 LANGUAGE sql
36 VOLATILE
37 SECURITY DEFINER;
38
39 CREATE OR REPLACE VIEW postgres_exporter.pg_stat_activity
40 AS
41     SELECT * from get_pg_stat_activity();
42
43 GRANT SELECT ON postgres_exporter.pg_stat_activity TO postgres_exporter;
44
45 CREATE OR REPLACE FUNCTION get_pg_stat_replication() RETURNS SETOF pg_stat_replication AS
46 $$ SELECT * FROM pg_catalog.pg_stat_replication; $$
47 LANGUAGE sql
48 VOLATILE
49 SECURITY DEFINER;
50
51 CREATE OR REPLACE VIEW postgres_exporter.pg_stat_replication
52 AS
53     SELECT * FROM get_pg_stat_replication();
54
55 GRANT SELECT ON postgres_exporter.pg_stat_replication TO postgres_exporter;
56
57 CREATE OR REPLACE FUNCTION get_pg_stat_statements() RETURNS SETOF pg_stat_statements AS
58 $$ SELECT * FROM public.pg_stat_statements; $$
59 LANGUAGE sql
60 VOLATILE
61 SECURITY DEFINER;
62
63 CREATE OR REPLACE VIEW postgres_exporter.pg_stat_statements
64 AS
65     SELECT * FROM get_pg_stat_statements();
66
67 GRANT SELECT ON postgres_exporter.pg_stat_statements TO postgres_exporter;
68
```

- Save `prepare-postgres-exporter.sql`.
- Run Command Prompt as administrator, then run the below script to connect the Database.

```
1 cd {InstallDir}
2 postgres\bin\psql.exe -U postgres -d postgres -h 127.0.0.1 -p 5432 -f prepare-postgres-exporter.sql
```

4.2 Postgres exporter configuration

- Create `PostgresExporterService.xml` at `C:\Program Files\PostgresExporter`.
- Copy the below information to `PostgresExporterService.xml` and change the `password` on line 6 with your actual password.

```
1 <service>
2   <id>PostgresExporter</id>
3   <name>PostgresExporter</name>
4   <description>PostgresExporter</description>
5   <workingdirectory>C:\Program Files\PostgresExporter</workingdirectory>
6   <env name="DATA_SOURCE_NAME" value="postgresql://postgres_exporter:password@localhost:5432/postgres?sslmode=disable" />
7   <executable>postgres_exporter.exe</executable>
8   <log mode="roll" />
9   <onfailure action="restart" />
10 </service>
```

- Copy `postgres_exporter.exe` to `C:\Program Files\PostgresExporter`.
- Copy `WinSW.exe` to `C:\Program Files\PostgresExporter` and change the file name to `PostgresExporterService.exe`.
- Run Command Prompt as administrator, then run the below script to install Postgres Exporter.

```
1 cd C:\Program Files\PostgresExporter
2 PostgresExporterService.exe install
3 PostgresExporterService.exe start
```

5 Relay

5.1 Relay configuration

- Go to `{InstallDir}\relay\relay` and edit `relay.conf.ori`.
- Add the below parameters in the file. (Remove the `#` when updating the parameters and set `"supportRCT=true"` and `"supportSON=true"`)

```
1 cmdSendStatistics=../../exporter.exe -p ../../log/relay_statistics.last.log
2 supportRCT=true
3 supportSON=true
```

- Save the file `relay.conf.ori`.
- Run Command Prompt as administrator, then run the below script to restart Relay service.

```
1 net stop OnpremiseSvr
2 net start OnpremiseSvr
```

5.2 Relay exporter configuration

- Create `RelayExporterService.xml` at `C:\Program Files\RelayExporter`.
- Copy the below information to `RelayExporterService.xml`.

```
1 <service>
2   <id>RelayExporter</id>
3   <name>RelayExporter</name>
4   <description>RelayExporter</description>
5   <workingdirectory>C:\Program Files\RelayExporter</workingdirectory>
6   <executable>exporter.exe</executable>
7   <arguments>-s -c relay_exporter.yaml</arguments>
8   <log mode="roll" />
9   <onfailure action="restart" />
10 </service>
```

- Copy `relay_exporter.yaml` to `C:\Program Files\RelayExporter` (Please ensure replace `%InstallPATH%` with your actual `{InstallDir}`).
- Copy `exporter.exe` to `C:\Program Files\RelayExporter`.
- Copy `WinSW.exe` to `C:\Program Files\RelayExporter` and change the file name to `RelayExporterService.exe`.
- Run Command Prompt as administrator, then run the below script to install Relay Exporter.

```
1 cd C:\Program Files\RelayExporter
2 RelayExporterService.exe install
3 RelayExporterService.exe start
```

6 Frontend

6.1 Frontend configuration

- Go to `{InstallDir}\relay\relay` and edit `frontend.conf.ori`.
- Edit the below parameter in the file. (Remove the `#` when updating the parameters.)

```
1 cmdSendStatistics=../../exporter.exe -p ../../log/frontend_statistics.last.log
```

- Run Command Prompt as administrator, then run the below script to restart Frontend service.

```
1 net stop OnpremiseSvr
2 net start OnpremiseSvr
```

6.2 Frontend exporter configuration

- Create `FrontendExporterService.xml` at `C:\Program Files\FrontendExporter`.
- Copy the below information to `FrontendExporterService.xml`.

```
1 <service>
2   <id>FrontendExporter</id>
3   <name>FrontendExporter</name>
4   <description>FrontendExporter</description>
5   <workingdirectory>C:\Program Files\FrontendExporter</workingdirectory>
6   <executable>exporter.exe</executable>
7   <arguments>-s -c frontend_exporter.yaml</arguments>
8   <log mode="roll" />
9   <onfailure action="restart" />
10 </service>
```

- Copy `frontend_exporter.yaml` to `C:\Program Files\FrontendExporter` (Please ensure replace `%InstallPath%` with your actual `{InstallDir}`).
- Copy `exporter.exe` to `C:\Program Files\FrontendExporter`.
- Copy `WinSW.exe` to `C:\Program Files\FrontendExporter` and change the file name to `FrontendExporterService.exe`.
- Run Command Prompt as administrator, then run the below script to install Frontend Exporter.

```
1 cd C:\Program Files\FrontendExporter
2 FrontendExporterService.exe install
3 FrontendExporterService.exe start
```

7 Windows exporter

- Create `WindowsExporterService.xml` at `C:\Program Files\WindowsExporter`.
- Copy the below information to `WindowsExporterService.xml`.

```
1 <service>
2   <id>WindowsExporter</id>
3   <name>WindowsExporter</name>
4   <description>WindowsExporter</description>
5   <workingdirectory>C:\Program Files\WindowsExporter</workingdirectory>
6   <executable>windows_exporter.exe</executable>
7   <log mode="roll" />
8   <onfailure action="restart" />
9 </service>
```

- Copy `windows_exporter-xxxx-xxx.exe` to `C:\Program Files\WindowsExporter` and change the name to `windows_exporter.exe`.
- Copy `WinSW.exe` to `C:\Program Files\WindowsExporter` and change the file name to `WindowsExporterService.exe`.
- Run Command Prompt as administrator, then run the below script to install Windows Exporter.

```
1 cd C:\Program Files\WindowsExporter
2 WindowsExporterService.exe install
3 WindowsExporterService.exe start
```

8 Monitoring Server

Please download Prometheus, AlertManager, and Grafana first. Then, configure the server monitoring according to your needs. In addition, we have provided a monitoring server configuration template for your reference, please download [How to set up a monitoring server.pdf](#) to get more information.

- Prometheus: <https://github.com/prometheus/prometheus/releases>
- AlertManager: <https://github.com/prometheus/alertmanager/releases>
- Grafana: <https://dl.grafana.com/enterprise/release/grafana-enterprise-10.2.3.windows-amd64.msi>

9 Firewall configuration

Please make sure the below ports are added to specific programs. For more information, please refer to <https://learn.microsoft.com/en-us/windows/security/operating-system-security/network-security/windows-firewall/configure-with-command-line?tabs=powershell>.

Program	LocalPort
<code>{InstallDir}\Onpremise.exe</code>	9980
<code>C:\Program Files\RelayExporter\exporter.exe</code>	5681
<code>C:\Program Files\FrontendExporter\exporter.exe</code>	5682
<code>C:\Program Files\PostgresExporter\postgres_exporter.exe</code>	9187
<code>C:\Program Files\WindowsExporter\windows_exporter.exe</code>	9182

C:\Program Files\AlertManager>alertmanager.exe	9093
C:\Program Files\Prometheus\prometheus.exe	9090
C:\Program Files\GrafanaLabs\grafana\bin\grafana.exe	3000