PM2 Windows Deployment

This is an example of how to install, run and set as a service the PM2 manager on windows. It is assumed that you already have node.js installed on the machine.

# INSTALLATION

Open the console in **administrator mode** and type the following commands:

*npm install pm2 –g*

*npm install pm2-windows-service –g*

The above two commands will globally install the pm2 process and the windows service generator on the system.

After this is done, open the **System Environment Variables** and add a new variable named **PM2\_HOME**. This variable is the run location of the pm2 process. Ideally it should be outside of a user directory. In the test case we set this to **E:\PM2** which is an empty folder for now.

We will also install the log rotation module into pm2. To do this, execute the following:

*pm2 install pm2-logrotate*

This installs the log rotation module in the pm2 system. Next we can configure the log rotation. For example we can choose to rotate the logs when they become larger than 1KB or after one minute every minute:

*pm2 set pm2-logrotate:max\_size 1K (1KB)*

*pm2 set pm2-logrotate:rotateInterval '\*/1 \* \* \* \*' (force rotate every minute)*

We can also choose to compress the rotated logs with the following:

*pm2 set pm2-logrotate:compress true (compress logs when rotated)*

Please note that the output paths of the rotated logs are sent to **PM2\_HOME**.

# TESTING THE SERVICE

In order to test the service we require three files. One is the .json file which describes our ecosystem of processes we would like to run and the other two files are dummy node.js applications that we may want to manager in our system. Please see the below files for details.



The test.json file contains the properties that are fed into the pm2 service. The content is below:



The important piece to notice is the specification of different **env** declarations which allows us to set different environment variables on startup. For example we can change port numbers between development and production. Also notice that there are two processes being managed in the above file. For more details please visit: <http://pm2.keymetrics.io/docs/usage/application-declaration/>

# WINDOWS SERVICE

To make sure the process starts upon system reboot we need to add it as a windows service. To do this we are going to use the pm2 windows service tool. To install a service with the name **pm2test** we execute the following command:

*pm2-service-install –n pm2test*

When you execute this command it will prompt your for some permissions and also ask if you want to set the environment variables. Select NO to this option. Since we already defined the **PM2\_HOME** this is all we need. If you were to perform this setup it would cause some path issues later on.

If you open the Windows Service Panel you should see your new service active and running. If you go to the command line and type:

*pm2 list*

you will see that you have an empty set of apps and one module currently online (this is the rotate logging module). Now we will add our process to the windows service. Perform the following commands to start the service from the command line:

*pm2 start test.json –update-env –env production*

The above command should be done in the directory containing the .json file. The options for update-env will reset the given environment variables for the process of a change occurred between startups (ie if you changed the port number in the test.json you would need to include this on reload/restart of the pm2 service). The second command (env) lets you specify the environment you are running in. This is very important for deployment as it should be changed depending on the system it is being run on. (UAT should have UAT env, PROD should have PROD etc.)

If you were to run the list command again you should see the two applications online. Now we need to save this state by running the following commands:

*pm2 save*

This will save the current state of the process manager. At this point the service will recall this saved state upon system reboot. To mimic this behavior you can kill all node sessions and see if restarting the process:

*Taskkill /F /im node.exe*

This will kill all of the node sessions. Once you restart the server you should see the apps once again running on the production ports (as specified in the test.json file).

# UPDATING THE SERVICE

In the case that we want to make a change to the service all we must do is restart the process from the command line and perform another save command. For example to switch from the production to development environment (and make this persist across a reboot) we would perform the following.

*pm2 reload test.json –update-env –env development*

*pm2 save*

At this point the service has been updated with these specifications and will persist on a reboot event.