Say you have a JAR file and you need to run it as a service. Additionally, you want it to start automatically if/when system restarts.

Ubuntu has a built-in mechanism to create custom services, enabling them to get started at system boot time and start/stop them as a service. In this post, I am going to share a simple and elegant way to create a service wrapper for your JAR file so you can run it as a service. Here we go.

**Step 1: Create a Service**

sudo vim /etc/systemd/system/my-webapp.service

Copy/paste the following into the file /etc/systemd/system/my-webapp.service:

**[Unit]**

Description=My Webapp Java REST Service

**[Service]**

User=ubuntu

# The configuration file application.properties should be here:

#change this to your workspace

WorkingDirectory=/home/ubuntu/workspace

#path to executable.

#executable is a bash script which calls jar file

ExecStart=/home/ubuntu/workspace/my-webapp

SuccessExitStatus=143

TimeoutStopSec=10

Restart=on-failure

RestartSec=5

**[Install]**

WantedBy=multi-user.target

**Step 2: Create a Bash Script to Call Your Service**

Here’s the bash script that calls your JAR file: **my-webapp**

#!/bin/sh

sudo /usr/bin/java -jar my-webapp-1.0-SNAPSHOT.jar server config.yml

Don't forget to give your script execute permission: sudo chmod u+x my-webapp

**Step 3: Start the Service**

sudo systemctl daemon-reload

sudo systemctl enable my-webapp.service

sudo systemctl start my-webapp

sudo systemctl status my-webapp

**Step 4: Set Up Logging**

First, run: sudo journalctl --unit=my-webapp . See real-time logs by using the -f option.

If you want to trim them, use -n <# of lines> to view the specified number of lines of the log:

sudo journalctl -f -n 1000 -u my-webapp

Tail the live log using the -f option:

sudo journalctl -f -u my-webapp

Stop the service by using:

sudo systemctl stop my-webapp

That's it! Enjoy and show your support if you like it. Thanks!