# [Monitoring overview - Infra](https://atc.bmwgroup.net/confluence/display/HDNAS/Monitoring+overview+-+Infra)

[Skip to end of metadata](https://atc.bmwgroup.net/confluence/display/HDNAS/Monitoring+overview+-+Infra#page-metadata-end)

* Created by [Michal Markus (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz2lpd), last modified [about 7 hours ago](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=2661902476&selectedPageVersions=8&selectedPageVersions=9)

[Go to start of metadata](https://atc.bmwgroup.net/confluence/display/HDNAS/Monitoring+overview+-+Infra#page-metadata-start)

Deadman check for **Grafana**, **Inlfux**, **Harvest**, **Telegraf**.

[ Solution deployed to ]

EU UAT telegraf host - [ITAHDNASUATTEL.bmwgroup.net](http://itahdnasuattel.bmwgroup.net/)  & EU Prod telegraf host - [lvashhdntelp1.bmwgroup.net](http://lvashhdntelp1.bmwgroup.net/)

[ script - cron ]

\*/1 \* \* \* \* source ~/.bashrc; /home/qqky020/monitoring\_services.sh

[ description ]

The script runs every minute, if the service is down for five consecutive minutes then it sends a ticket (NodeRed→Remedy)

[ Installation Steps ]

* + install OS packages via **root** user:

sudo yum install -y alsa-lib bc gcc gcc-c++ kernel-devel libffi-devel libXcomposite libXcursor libXdamage libXi libXrandr libXScrnSaver libxslt-devel libXtst mesa-libEGL mesa-libGL msodbcsql18.x86\_64 openssl-devel unixODBC-devel

* + update .bashrc (optional)
  + install miniconda (version: Miniconda3-py37\_4.8.2-Linux-x86\_64.sh) with **qqky020**user  
    chown -R <qqkyuser> <qqkygroup> <path\_to\_folder>  #replace with real values;  
    chmod +x Miniconda3-py37\_4.8.2-Linux-x86\_64.sh  
    ./Miniconda3-py37\_4.8.2-Linux-x86\_64.sh  
    # select "yes"

#reload bashrc via: . ~/.bashrc

* + install python modules  
    for package in Flask python-dotenv pandas pyodbc; do pip install $package --proxy "[http://qqky010:Kyndryl&BMW2022@192.109.190.88:8080](http://qqky010:Kyndryl&BMW2022@192.109.190.88:8080/)"; done
  + git clone scripts and flask\_UI (\*PLACEHOLDER\* repository link and command to be added) ; flask UI url:   
    tar -xvf ...
  + add firewall port:  
    firewall-cmd --add-port=8000/tcp  
    firewall-cmd --add-port=8000/tcp --permanent
  + update crontab: crontab -e  
    \*/1 \* \* \* \* source ~/.bashrc; /home/qqky020/monitoring\_services.sh  # backend script that checks service availability & sends ticket   
    5 0 \* \* \* source ~/.bashrc; insert\_uptime.sh  # sends uptime records to mssql database for reporting team
  + for reporting to MSSQL DB:  
    #create a **/etc/odbc.ini** file with the content below: (replace username and password with proper values!)  
    [DWH]  
    server = mvashhdnradp1  
    #driver = unixodbc  
    driver = /opt/microsoft/msodbcsql18/lib64/[libmsodbcsql-18.1.so](http://libmsodbcsql-18.1.so/).1.1  
    database = BMW\_Common\_View  
    username = "\*\*\*\*"  
    password = '\*\*\*\*'  
    TrustServerCertificate = yes  
    Trace = Yes  
    TraceFile = /home/qqky020/UI/flask\_wapi\_PROD/odbc.log
  + logrotation:

chmod 700 /home/qqky020/UI/flask\_wapi\_UAT   
vi /etc/logrotate.d/flask

# contents of the 'flask' file:  
/home/qqky020/UI/flask\_wapi\_UAT/\*uptime.log  
{  
    rotate 3  
    create 0644 [qqky020@europe.bmw.corp](mailto:qqky020@europe.bmw.corp) [qqky020@europe.bmw.corp](mailto:qqky020@europe.bmw.corp)  
    monthly  
    size 10M  
    missingok  
    dateext  
    copytruncate  
    notifempty  
    compress  
    delaycompress  
}

[ Monotirng Interface ]

http://<telegraf\_host\_IP>:8000

- EU UAT url: [http://10.2.58.59/8000](http://10.2.58.59:8000/)  
- EU Prod url: [http://10.31.51.55/8000](http://10.31.51.55:8000/)

Ansible jobs are monitored separately via @Gerhard's UI

**\* add link here** \*

[ to be added ]

* backup jobs monitoring
* qqky020 user password expire/revoke check
* performance monitoring