

Who's watching the watcher?

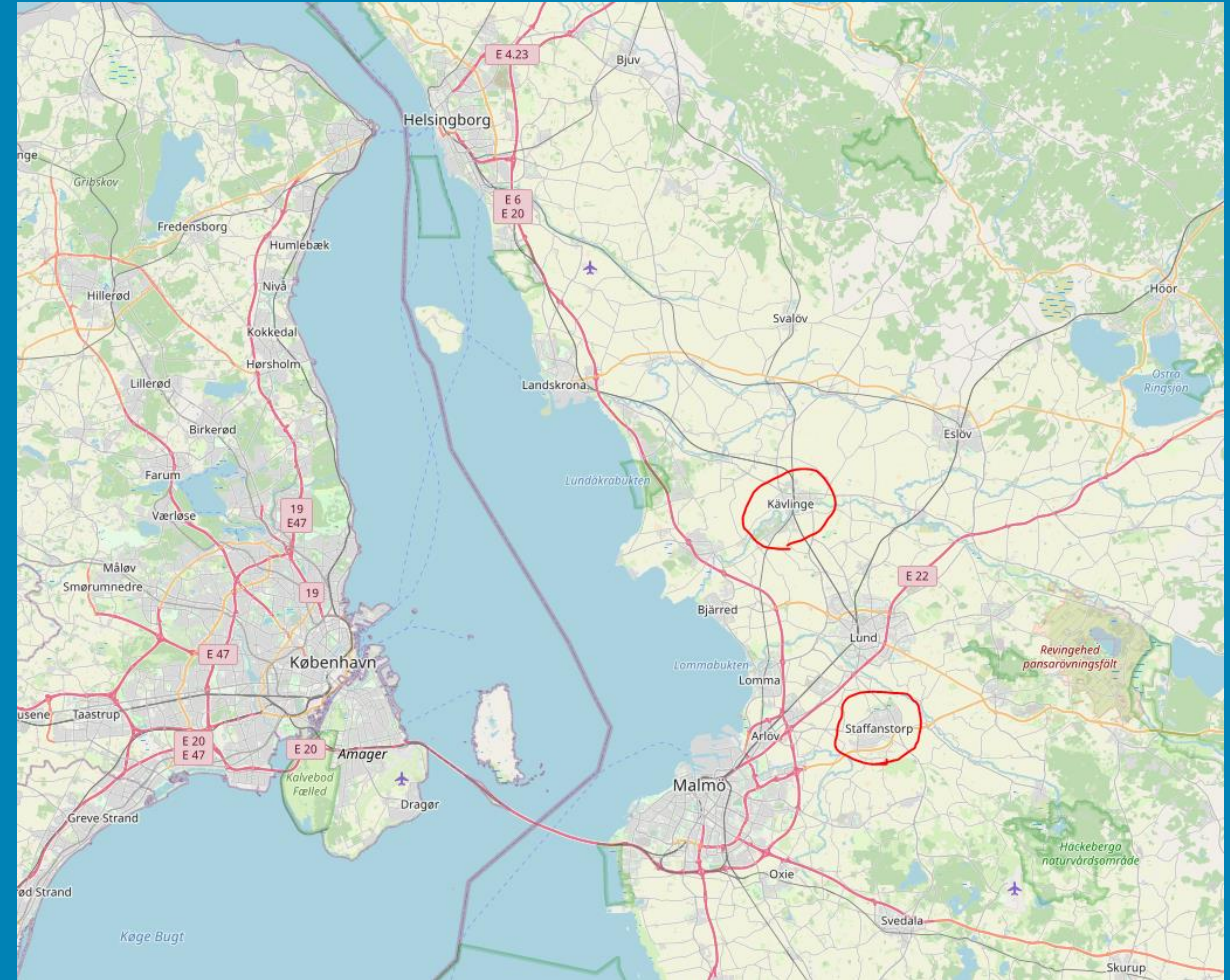
Martin Ekstrand, Geoinfo Staffanstorp-Kävlinge

Hur använder vi FME?

- 4 användare
- Desktop + server

Exempel på tillämpningar:

- Konverteringar
- Analyser
- Datakvalitetskontroll
- Automationer
- Self-service
- Övervakning av andra system



System utöver FME:

- ArcGIS Enterprise (multi machine deployment med hosting server + image server)
- ArcGIS Pro
- GEOSECMA for ArcGIS (egen fristående ArcGIS-server)
- Autocad
- Topocad
- SQL server (ca 40 databaser)
- Mapillary
- ClickUp

Övervakning

```
#LVDB.log
with open(lastLogFile) as logFile:
    data = logFile.read()
    data = data.splitlines(True)
    if 'Plan executed without errors.' in data[-1] and check_date(lastLogFile):
        print ("Inga fel inträffade.")
    else:
        str_data = str(data)
        if check_date(lastLogFile):
            msg_subject = 'PYTHON FAIL: LVDB - laddning med error.'
        else:
            msg_subject = 'PYTHON FAIL: LVDB - Gammal loggfil?'
        msg_middle = today + "<br>" + msg_subject + "<br>" + replace_func(str_data)
        msg = msg_start + msg_middle + msg_end
        with open(r"D:\Systemovervakning\geosecma\lvdb\lvdb.html", "w") as lvdbHTML:
            lvdbHTML.write(msg)

#LVDB_publicering.log
pub_log = r'D:/Logs/lvdb_publicering.log'
with open(pub_log) as pubLogFile:
    data = pubLogFile.read()
    data = data.splitlines(True)
    str_data = str(data)
    if 'Plan executed without errors.' in data[-1]:
        print ("Inga fel inträffade i publicering.")
    else:
        msg_subject = "LVDB publicering fail"
        msg_middle = today + "<br>" + msg_subject + "<br>" + replace_func(str_data)
        msg = msg_start + msg_middle + msg_end
        with open(r"D:\Systemovervakning\geosecma\lvdb\LVDB_publicering.html", "w") as LVDB_Publicering:
            LVDB_Publicering.write(msg)

#LVDBCompress.log
compress_log = r'D:/Logs/LVDBCompress.log'
with open(compress_log) as compressLogFile:
    data = compressLogFile.read()
    data = data.splitlines(True)
    regex = "End_state: 1"
    try:
        end_state_ok = True if re.search(regex, data[0]) and re.search(regex, data[1]) else False
    except:
        end_state_ok = False

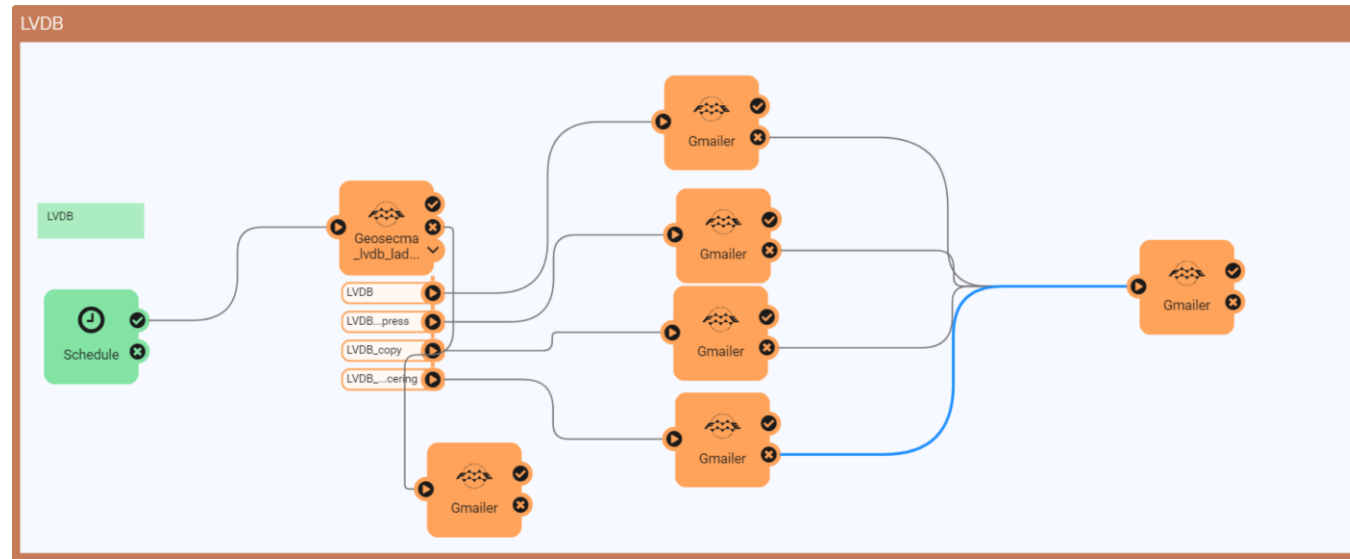
    if end_state_ok:
        print ("Inga fel inträffade i compress.")
    else:
        str_data = str(data)
        msg_subject = "LVDB compress fail"
        msg_middle = today + "<br>" + msg_subject + "<br>" + replace_func(str_data)
        msg = msg_start + msg_middle + msg_end
        print(msg)
        with open(r"D:\Systemovervakning\geosecma\lvdb\LVDBCompress.html", "w") as LVDBCompress:
            LVDBCompress.write(msg)

#lvdb_copy.log
copy_log = r'D:/Logs/lvdb_copy.log'
with open(copy_log) as copyLogFile:
    data = copyLogFile.read()
    data = data.splitlines(True)
    str_data = str(data)
    if "Failed to execute" in str_data:
        with open(r"D:\Systemovervakning\geosecma\lvdb\lvdb_copy.html", "w") as lvdbcopyHTML:
            msg_subject = "LVDB_copy fail: kunde inte kopiera GDB"
            msg_middle = today + "<br>" + msg_subject + "<br>" + replace_func(str_data)
            msg = msg_start + msg_middle + msg_end
            lvdbcopyHTML.write(msg)
```

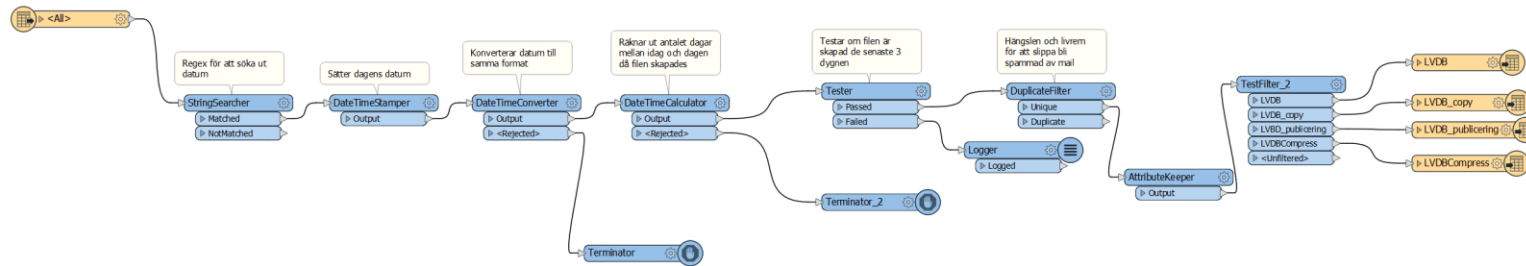
Varför inte låta python skicka mail direkt?

Dels för att smtp inte är helt robust men även...

Övervakaren som övervakar övervakningen



2022-06-21 (Martin) Anledningen till datatrukkningen är att directory watch inte fungerade. Möjliga pga UNC path eller bugg. Pythonscript skriver bara ut filer om fel uppstått, därav försöket med DW. Nu fick jag ritatlet modda det så att skriptet skriver ut datum längst upp och sedan använda det för att kontrollera.



När övervakaren inte fungerar

Vem övervakar övervakaren som övervakar övervakningen?



automations : Automation Manager

GET	/automations/workflows	Retrieve all automation workflows
POST	/automations/workflows	Create an automation workflow
GET	/automations/workflows/tags	Retrieve all automation tags
POST	/automations/workflows/tags	Create an automation tag
DELETE	/automations/workflows/tags/< tag >	Remove an automation tag
GET	/automations/workflows/< id >	Retrieve information about an automation workflow
PUT	/automations/workflows/< id >	Update information about an automation workflow
DELETE	/automations/workflows/< id >	Remove an automation workflow
PUT	/automations/workflows/< id >/enabled	Enable or disable an automation workflow
GET	/automations/workflows/< id >/log	Download automation workflow log file with filters
GET	/automations/workflows/< id >/log	Get the log of an automation workflow
DELETE	/automations/workflows/< id >/log	Clear automation workflow log file
GET	/automations/workflows/< id >/status	Get the status of an automation workflow
POST	/automations/workflows/< id >/trigger	Trigger the Manual node in an enabled automation workflow

Vår övervakning över övervakaren som övervakar övervakningen

- Pythonscript som körs varje morgon kl 06 via task scheduler på en av våra GIS-serverar

Målet är en "most bang for the buck"-kontroll:

- Är servern responding och redo att ta emot jobb?
- Finns det för många jobb i kön?
- Finns det något jobb som väntat i kön för länge?
- Har våra viktigaste automationer kastat errors eller warnings?
- Om något av detta inträffar skickas ett mail, denna gången via smtp.

Scriptet tar bara några sekunder att köra på servern.

Det är parametriserat och indelat i funktioner.

```
I 2022-10-05T05:06:53+02:00 | 402622 : Job 1395: Translation failed: "BADNEWS: BADNEWS: GEODATABASE_SDE writer: A fatal error has occurred. Check the logfile above for details".
```

Exempel: FME server skickade aldrig mail... Men pythonscriptet fångade felet!

```
Automation 059b5545-a9e8-4feb-90bb-a44b53af32de report: {'errors': 1, 'warnings': 0}
```

Vad krävs för att du ska kunna använda samma?

- Windows Server eller dylikt för att schemalägga script (rimligen en annan maskin än den som kör FME server, om möjligt).
- En token i FME rest API som du kan använda – detta är enkelt att skapa! En del av kontrollerna funkar dessutom utan.
- Ladda ner scriptet (https://kartportal.se/staffanstopr/Python/FMEServer_status_check.zip)
- Om den ska skicka mail över smtp behöver du ett konto som stöder det.
- Ställ in parametrar i script och schemalägg.

```
#System variables
fme_server_url = r"https://{YOURFMESERVERURLHERE}/fmerest/v3" #The URL to your fme server
payload = {} #placeholder in case you need it
headers = {} #don't touch unless absolutely needed
token = "fmetoken token={YOURTOKENSTRINGHERE}" #Create a token in your fme server (set access automations and metrics and FME server will prompt you for additional permissions
needed), then set token string
error_report = {} #don't touch

#Parameters for settings
max_queue_size = 5 #This number controls how many jobs that can be queued before the script should consider it wrong
max_wait_time = 3600 #Maximum accepted waiting time in seconds for the job in the queue that has been waiting the longest
important_automations = [{"IDnr1"}, {"IDnr2"}] #list the ID:s of your most important automations
#These parameters can be set to false to skip certain tests
run_health_check = True
run_metrics = True
run_status_important_automations = True

#SMTP settings - you need to provide these (default office365)
smtp_server = "smtp.office365.com"
smtp_port = 587
mail_user = {EMAILUSERNAME}'
mail_password = "{EMAILPASSWORD}"
sent_from = "{MAILFROM}"
mail_to = [{"MAILTO1"}, {"MAILTO2"}]
```


Utveckling framåt statuskontroll

Funderingar:

- Kontroll om last run på en workspace och/eller automations – har den körts när den skulle?
- Runtime på workspaces:
 - – är det någon som tar för lång tid att köra?
 - – tagit för mycket RAM?
 - – för mycket CPU?

Utveckling framåt rest-api + ArcGIS

Få en del flöden utanför brandväggen genom att baka in i en custom geoprocessing tool

```
import requests
import json

url =
https://{FMESERVER}/fmerest/v3/transformations/transact/{REPOSITORY}/{WORKSPACE}fmw

payload = json.dumps({})
headers = {
    'Accept': 'application/json',
    'Authorization': 'fmetoken token={TOKEN FÖR DIN WORKSPACE}',
    'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

Spara som .py -> lägg in i toolbox -> publicera som gp-service -> lägg in i gp widget

Utveckling framåt övrigt

Håller på för ArcGIS Enterprise:

Ska kontrollera:

- Statuscheck portal machine(s)
- Portal index – automatisk reindexering om så specificeras
- Kontroll av portalens loggar
- Validation check av alla federerade servrar
- Validation av Enterprise data stores
- Kontroll av alla servrars loggar
- Kontroll av alla servrars error reports (ska skickas in till ESRI inc)
- Serviceusage: kollar max response time, failed requests, timed-out requests för VARJE service på respektive server

Sedan sammanställs allt till en HTML-fil med tabeller osv som ska övervakas av FME server. Alltid övervakning i 3 steg om möjligt 😊

Glöm inte övervaka övervakaren som övervakar övervakningen

Pythonscript: https://kartportal.se/staffanstorp/Python/FMEServer_status_check.zip

LinkedIn artikel: <https://www.linkedin.com/pulse/fme-server-basic-status-check-python-script-martin-ekstrand>

FME Server REST API kurs: <https://www.youtube.com/playlist?list=PLFxZDg3GNCguRCdExFnujegQjLmfMqh8> (sök på 2019 FME Server REST API Training Course)

Tack

Martin.Ekstrand@staffanstorp.se