

SonarQube Script를 통해 알람메일 보내기

이 문서는 SonarQube Script를 통해 알람메일 보내기 가이드를 공유하기 위해 작성되었다.

도구명	Sonarqube
버전	9.5
비고	Sonarqube 6.1 이상 가능

- [Sonarqube LOC 임계값 설정 시 메일 안 보내지는 이유](#)
- [Script 설정하기](#)
 - 리눅스
 - 윈도우
- [참조](#)

Sonarqube LOC 임계값 설정 시 메일 안 보내지는 이유

글로벌 관리자 알림

License Manager에는 두 가지 기본 제공 알림 메커니즘이 있습니다(알림은 전역 관리자에게 전송 됨). 전역 관리자는 다음과 같은 경우 알림을 받습니다.

- 라이선스가 곧 만료 됩니다. 라이선스가 만료되기 2개월 전에 다시 알림이 전송됩니다.
- 구성 가능한 **LOC 임계값**이 초과 되었습니다. LOC 게이지의 표시기를 사용하여 이 임계값을 변경할 수 있습니다.
 - 백그라운드 작업은 서버 시작 시 실행된 다음 24시간마다 실행되어 LOC 임계값을 확인하고 알림을 보낼지 여부를 결정합니다.
 - 백그라운드 작업은 SMTP 서버 설정의 유효성과 글로벌 관리자 이메일 주소가 설정되었는지 여부를 확인하지 않습니다. 전역 관리자가 알림을 수신하려면 LOC 임계값에 도달하거나 초과하기 전에 이러한 설정을 올바르게 구성해야 합니다.

Script 설정하기

리눅스

vi mailsend.sh

- alerday: 만료날짜 이전에 받고 싶은 날 입력
- baseurl: Sonarqube 주소
- admin: admin 계정의 ID, password: 해당 계정의 password, email: 받고 싶은 이메일 주소 입력
- 25줄 MSG1: LOC 임계값 이상 시 받고 싶은 메시지 입력
- 28줄 MSG2: 만료날짜 경고 시 받고 싶은 메시지 입력

특정 사용자만 추가시

```
#!/bin/bash
#
alertyday=60
baseurl=http://localhost
admin=admin
password=
email=
MSG1=""
MSG2=""
MSG3="\n\n\n Sonarqube  Crontab  ."
CURLS=`curl -u $admin:$password ${baseurl}/api/editions/show_license`
MAX=`echo $CURLS | json_reformat | grep -w maxLoc | awk '{print $2}' | cut -d ',' -f 1`
THRESHOLD=`echo $CURLS | json_reformat | grep -w remainingLocThreshold | awk '{print $2}'`
LOC=`echo $CURLS | json_reformat | grep -w loc | awk '{print $2}' | cut -d ',' -f 1`
EXPIRE=`echo $CURLS | json_reformat | grep -w 'expiresAt' | awk '{print $2}' | cut -d '"' -f 2`
DATE=$(date +%Y-%m-%d)
#
EXPIRESEC=`date -d $EXPIRE "+%s"`
DATESEC=`date -d $DATE "+%s"`
#
RESULT=`echo "${MAX} - ${THRESHOLD} - ${LOC}" | bc`
EXPIREDAY=`echo "($EXPIRESEC - $DATESEC) / 86400" | bc`
#
if (( ${RESULT} <= 0 )) ; then
    MSG1=" . LOC: ${LOC}\n"
fi
if (( ${EXPIREDAY} <= ${alertyday} )) ; then
    MSG2=" . : ${EXPIRE}"
fi
MESSAGE=`echo -e ${MSG1}${MSG2}${MSG3}`
#
if (( ${RESULT} <= 0 )) || (( ${EXPIREDAY} <= ${alertyday} )) ; then
    curl -X POST -u $admin:$password "${baseurl}/api/emails/send" -H "Content-Type: application/x-www-form-urlencoded" -d message="${MESSAGE}" -d subject="Sonarqube  " -d to="${email}"
    #Crontab
    ,
    SCRIPT=`echo $0 | cut -d '/' -f 2`
    crontab -l | grep -v "${SCRIPT}" > crontab.txt
    crontab crontab.txt
fi
```

sonar-administrator 그룹들 메일 보내기

- 그룹 변경 시 34줄 그룹 변경

특정 그룹에 속한 사용자들 대상

```
#!/bin/bash
#
alertyday=60
baseurl=http://localhost
admin=
password=
MSG1=""
MSG2=""
MSG3="\n\n Sonarqube  Crontab  ."
CURLS=`curl -u $admin:$password ${baseurl}/api/editions/show_license`
MAX=`echo $CURLS | json_reformat | grep -w maxLoc | awk '{print $2}' | cut -d ',' -f 1`
THRESHOLD=`echo $CURLS | json_reformat | grep -w remainingLocThreshold | awk '{print $2}'`
LOC=`echo $CURLS | json_reformat | grep -w loc | awk '{print $2}' | cut -d ',' -f 1`
EXPIRE=`echo $CURLS | json_reformat | grep -w 'expiresAt' | awk '{print $2}' | cut -d '"' -f 2`
DATE=$(date +%Y-%m-%d)
#
EXPIRESEC=`date -d $EXPIRE "+%s"`
DATESEC=`date -d $DATE "+%s"`
#
RESULT=`echo "${MAX} - ${THRESHOLD} - ${LOC}" | bc`
EXPIREDAY=`echo "($EXPIRESEC - $DATESEC) / 86400" | bc`
#
if (( ${RESULT} <= 0 )) ; then
    MSG1=" . LOC: ${LOC}\n"
fi
if (( ${EXPIREDAY} <= ${alertyday} )) ; then
    MSG2=" . : ${EXPIRE}"
fi
MESSAGE=`echo -e ${MSG1}${MSG2}${MSG3}`
#
if (( ${RESULT} <= 0 )) || (( ${EXPIREDAY} <= ${alertyday} ))
then
    # , curl (grep -B ) )
    for EMAIL in `curl -u $admin:$password ${baseurl}/api/users/search | json_reformat | grep -B 4 "sonar-
administrators" | grep -w 'email' | awk 'x[$0]++' | awk '{print $2}' | cut -d '"' -f 2`
    do
        curl -X POST -u $admin:$password "${baseurl}/api/emails/send" -H "Content-Type: application/x-
www-form-urlencoded" -d message="${MESSAGE}" -d subject="Sonarqube " -d to="${EMAIL}"
    done
    #Crontab ,
    SCRIPT=`echo $0 | cut -d '/' -f 2`
    crontab -l | grep -v "${SCRIPT}" > crontab.txt
    crontab crontab.txt
fi
```

■ 권한 부여

```
chmod 755 mailsend.sh
```

Crontab으로 시간 설정

```
crontab -e
```

```
# , , , ,
0 9 * * 1-5 $script_location/mailexend.sh
```

윈도우

Powershell 사용

- alertday: 만료날짜 이전에 받고 싶은 날 입력
- baseurl: Sonarqube 주소
- admin: admin 계정의 ID, password: 해당 계정의 password, email: 받고 싶은 이메일 주소 입력
- 31줄 MSG1: LOC 임계값 이상 시 받고 싶은 메시지 입력
- 34줄 MSG2: 만료날짜 경고 시 받고 싶은 메시지 입력

특정 사용자만 추가시

```
#
$alertday=60
$baseurl="http://localhost"
$admin=""
$password=""
$MSG1=""
$MSG2=""
$MSG3="\n\n Sonarqube ."
$email = ""
#Access
function Get-BasicAuthCreds {
    param([string]$Username,[string]$Password)
    $AuthString = "{0}:{1}" -f $Username,$Password
    $AuthBytes = [System.Text.Encoding]::Ascii.GetBytes($AuthString)
    return [Convert]::ToBase64String($AuthBytes)
}
$BasicCreds = Get-BasicAuthCreds -Username $admin -Password $password;
$CURLS=(Invoke-WebRequest -Uri "${baseurl}/api/editions/show_license" -Method Get -Headers @{"Authorization"="Basic $BasicCreds"}).Content | ConvertFrom-Json

$MAX=$CURLS.maxLoc
$THRESHOLD=$CURLS.remainingLocThreshold
$LOC=$CURLS.loc
$EXPIRE=$CURLS.expiresAt
$DATE=(Get-Date).ToString("yyyy-MM-dd")
#
$RESULT= ${MAX} - ${THRESHOLD} - ${LOC}
$EXPIREDAY = (New-TimeSpan -Start $DATE -End $EXPIRE).Days

#
if(${RESULT} -le 0 ){
    $MSG1=" . LOC: ${LOC}\n"
}
if(${EXPIREDAY} -le ${alertday}){
    $MSG2=" . : ${EXPIRE}"
}
$MESSAGE="${MSG1}${MSG2}${MSG3}"
#
if ((${RESULT} -le 0) -or (${EXPIREDAY} -le ${alertday})){
    $body = @{"message"=${MESSAGE}}
    subject="Sonarqube "
    to="${email}"
}
Invoke-WebRequest -Uri "${baseurl}/api/emails/send" -Method POST -Headers @{"Authorization"="Basic $BasicCreds"} -ContentType "application/x-www-form-urlencoded" -Body $body
}
```

sonar-administrator 그룹들 메일 보내기

- 받고자 하는 그룹 변경시 : 44줄 부분 변경

특정 그룹에 속한 사용자들 대상

```
#
$alertday=60
$baseurl="http://localhost"
$admin=""
$password=""
$MSG1=""
$MSG2=""
$MSG3="\n\n Sonarqube ."
$email = @()
#Access
function Get-BasicAuthCreds {
    param([string]$Username,[string]$Password)
    $AuthString = "{0}:{1}" -f $Username,$Password
    $AuthBytes = [System.Text.Encoding]::Ascii.GetBytes($AuthString)
    return [Convert]::ToBase64String($AuthBytes)
}
$BasicCreds = Get-BasicAuthCreds -Username $admin -Password $password;
$CURLS=(Invoke-WebRequest -Uri "${baseurl}/api/editions/show_license" -Method Get -Headers @{"Authorization"="Basic $BasicCreds"}).Content | ConvertFrom-Json

$MAX=$CURLS.maxLoc
$THRESHOLD=$CURLS.remainingLocThreshold
$LOC=$CURLS.loc
$EXPIRE=$CURLS.expiresAt
$DATE=(Get-Date).ToString("yyyy-MM-dd")
#
$RESULT= ${MAX} - ${THRESHOLD} - ${LOC}
$EXPIREDAY = (New-TimeSpan -Start $DATE -End $EXPIRE).Days

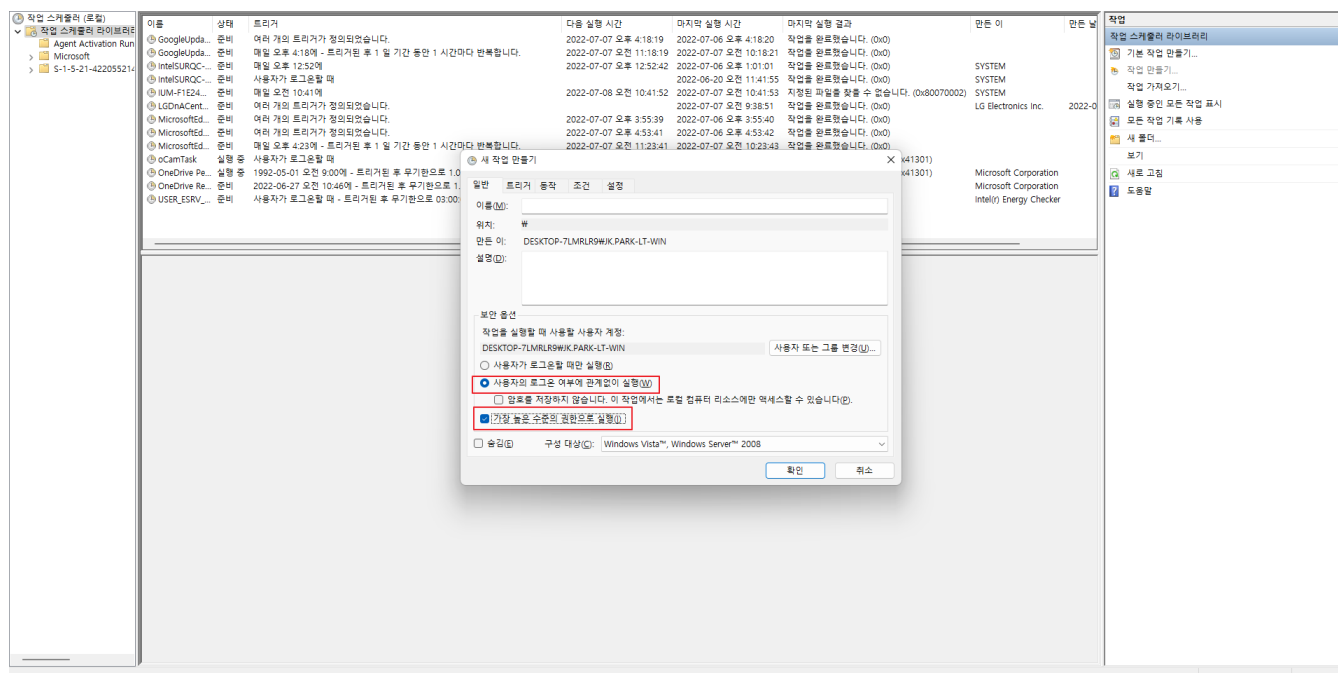
#
if(${RESULT} -le 0 ){
    $MSG1=" . LOC: ${LOC}\n"
}
if(${EXPIREDAY} -le ${alertday}){
    $MSG2=" . : ${EXPIRE}"
}
$MESSAGE="${MSG1}${MSG2}${MSG3}"
#
if ((${RESULT} -le 0) -or (${EXPIREDAY} -le ${alertday})){
    #
    $Users= (Invoke-WebRequest -Uri "${baseurl}/api/users/search" -Method Get -Headers @{"Authorization"="Basic $BasicCreds"}).Content | ConvertFrom-Json
    #
    $Users.users | ForEach-Object{
        if($_.groups.Contains("sonar-administrators")){
            $email += $_.email
        }
    }
    #
    $email | sort | unique | ForEach-Object{
        $body = @{"message"="${MESSAGE}"
        subject="Sonarqube "
        to="$_"}
        Invoke-WebRequest -Uri "${baseurl}/api/emails/send" -Method POST -Headers @{"Authorization"="Basic $BasicCreds"} -ContentType "application/x-www-form-urlencoded" -Body $body
    }
}
}
```

작업스케줄러 등록

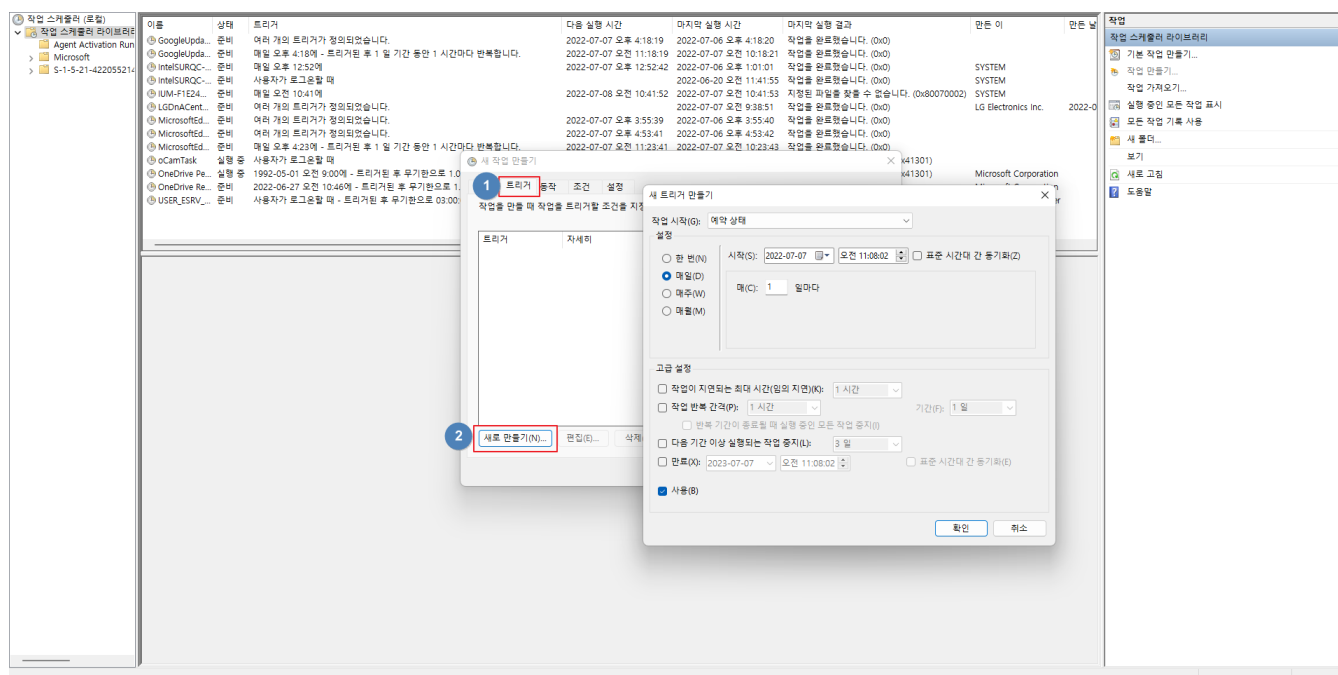
작업스케줄러 → 작업 만들기 → 일반

- 사용자 로그인 여부에 상관없이 실행 체크

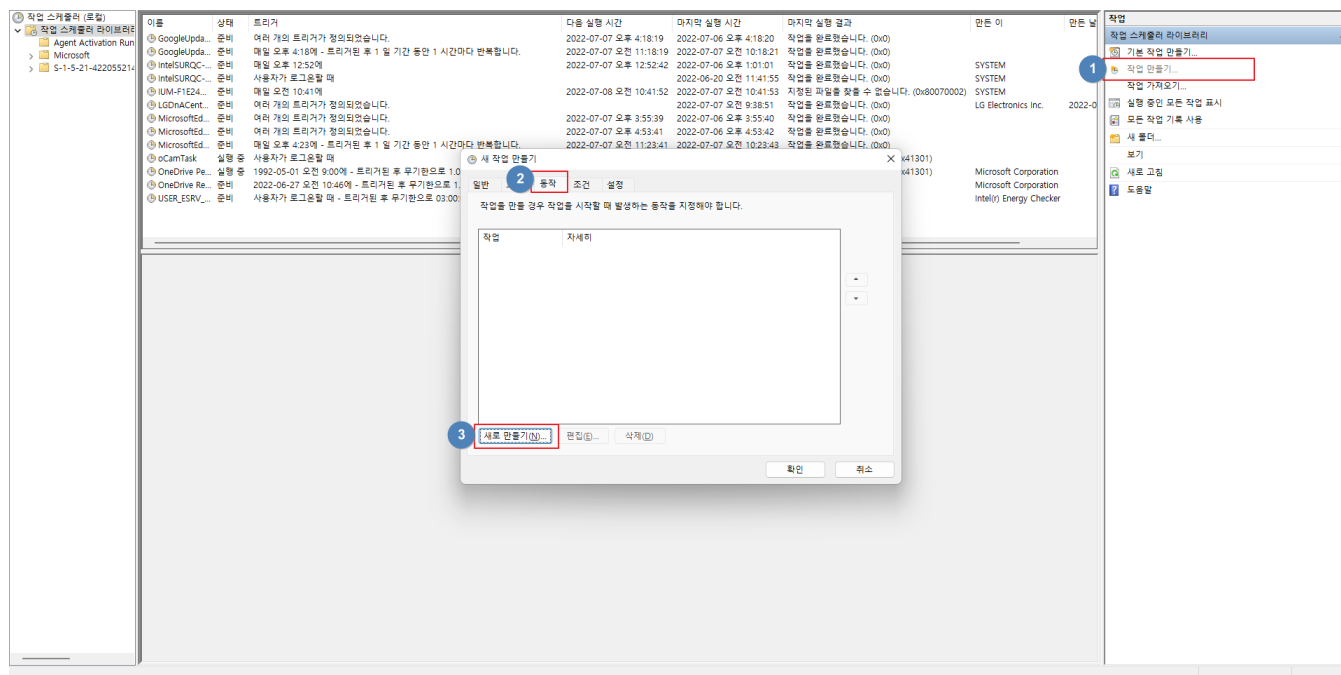
가장 높은 수준의 권한으로 실행 체크



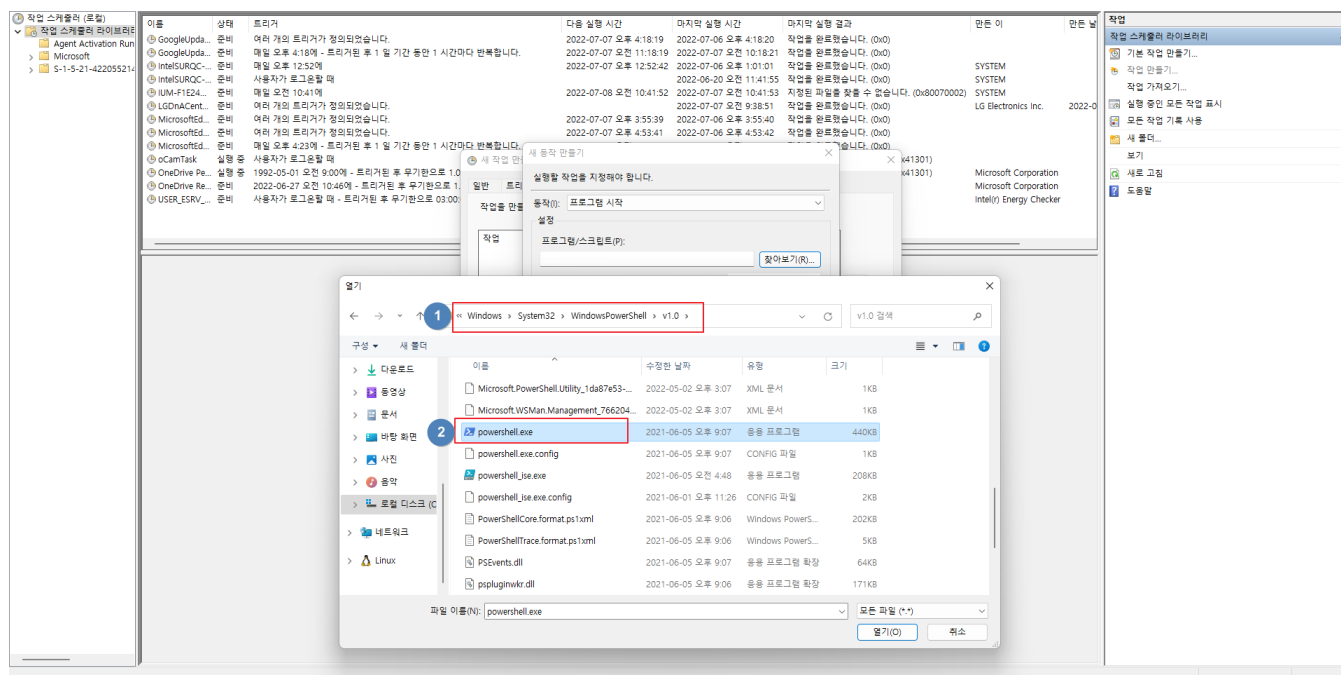
트리거 → 새로 만들기 선택: 주기 설정



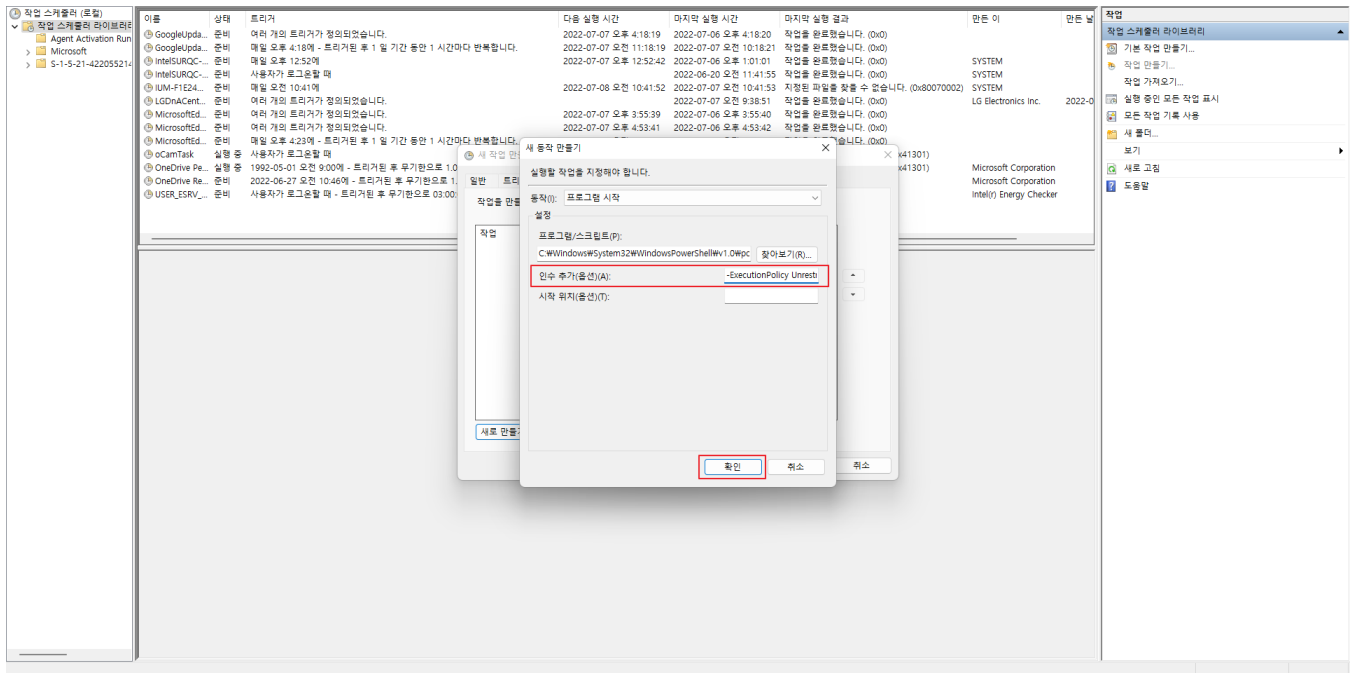
동작 → 새로 만들기 선택



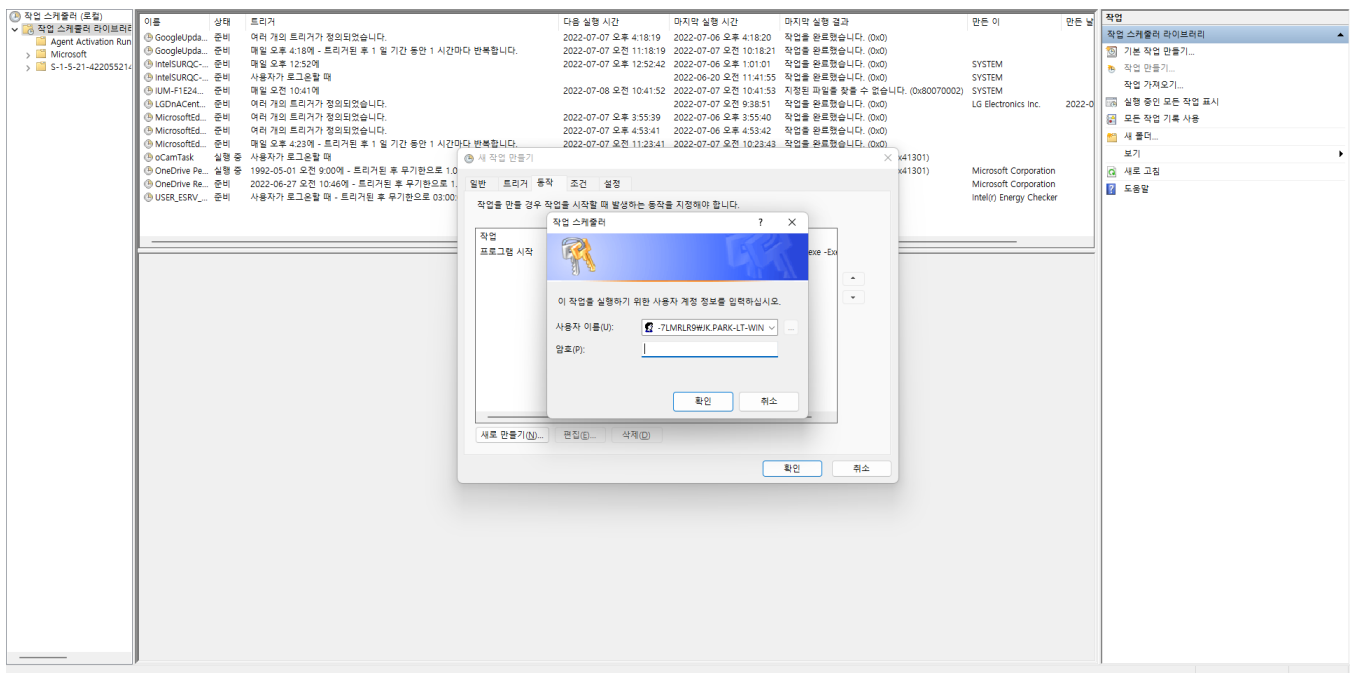
C:\Windows\System32\WindowsPowerShell\v1.0\W 경로의 Powershell.exe 선택



인수 추가(옵션) 칸에 -ExecutionPolicy Unrestricted -File "해당 파일"



확인 → 패스워드 입력



메일 보내고 난 후 작업스케줄러 중지 방법

해당 Powershell Script 문 메일 보내는 if문(단일메일: 44줄, 그룹메일: 56줄) 안에

```
($TaskScheduler = New-Object -ComObject Schedule.Service).Connect("localhost")
$TaskScheduler.GetFolder('\').GetTask("").Enabled = $false
```

해당 구문을 넣을 시 메일을 한번 보내고 작업스케줄러에서 사용 안함으로 변경된다.

참조

[Sonarqube Docs-License](#)

[Sonarqube Web Api](#)