Alerts For Elastic Jobs Failure

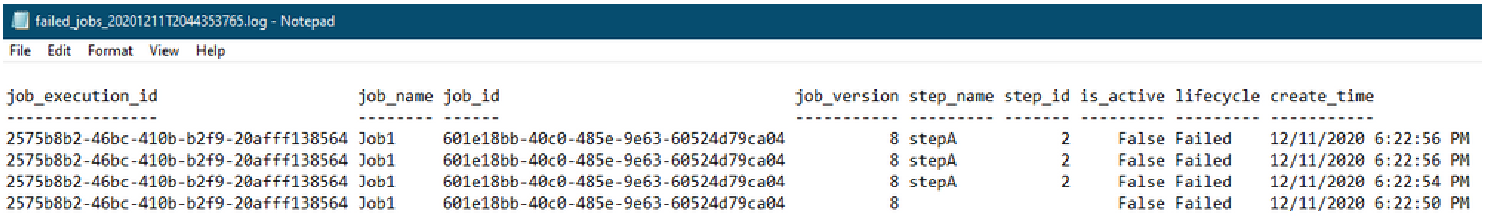
Last updated by | Melania Nitu (RINF) | Dec 21, 2020 at 7:07 AM PST

Contents

Automate and Send Alerts for Elastic Jobs Failures

Alerts can be sent for any particular state of an elastic job. In this TSG, we discuss the case of 'Failed' jobs, however the solution can be easily extended to other scenarios. The script can be executed on client machines in background on a schedule or in Azure Automation Runbook. If failed jobs are identified, email notifications are sent, with an attachment of a log file containing only the failed jobs.

Snippet of generated log file:

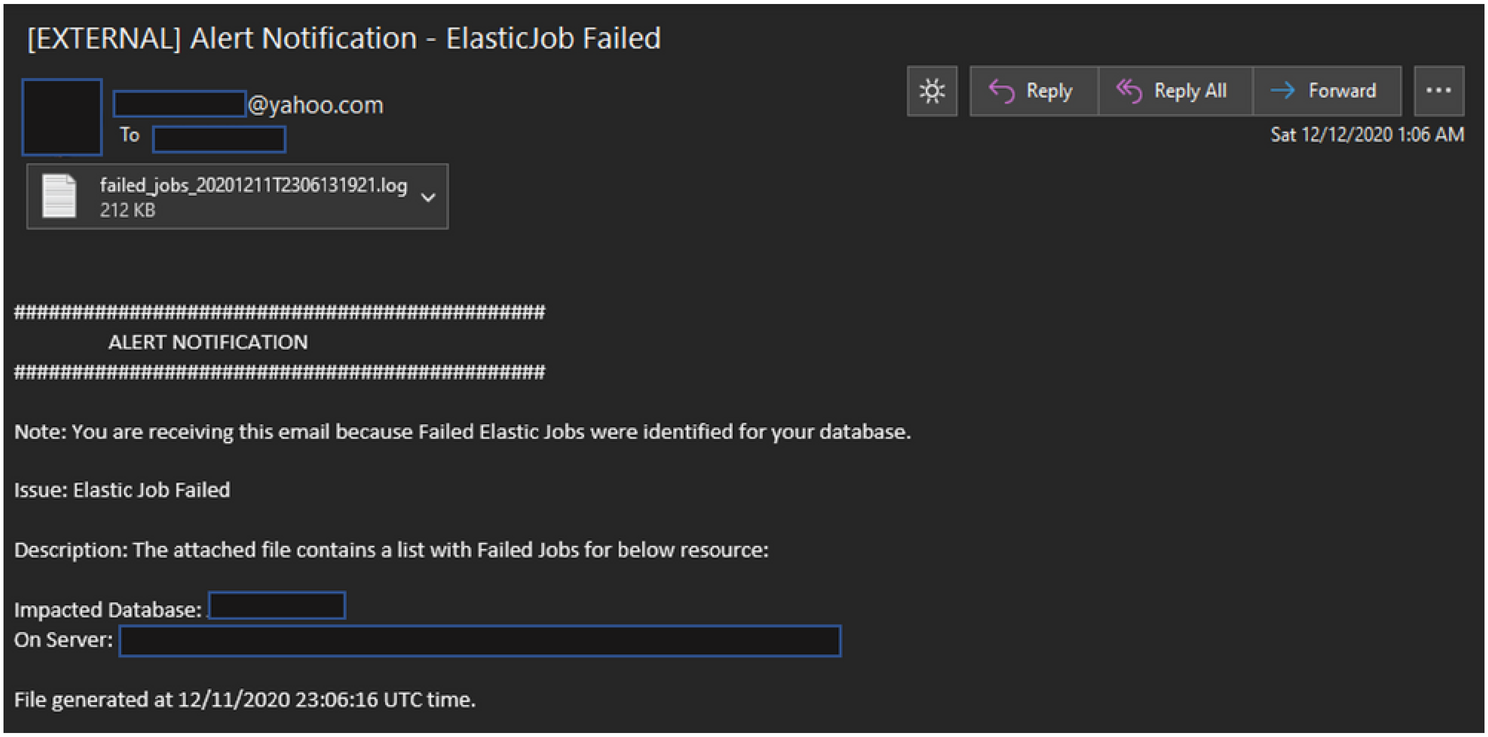


failed_jobs_20201211T2044353765.log - Notepad
File Edit Format View Help

```
job_execution_id                     job_name job_id                               job_version step_name step_id is_active lifecycle create_time
----------------                     -------- ------                               ----------- --------- ------- --------- --------- -----------
2575b8b2-46bc-410b-b2f9-20afff138564 Job1     601e18bb-40c0-485e-9e63-60524d79ca04           8 stepA           2     False Failed    12/11/2020 6:22:56 PM
2575b8b2-46bc-410b-b2f9-20afff138564 Job1     601e18bb-40c0-485e-9e63-60524d79ca04           8 stepA           2     False Failed    12/11/2020 6:22:56 PM
2575b8b2-46bc-410b-b2f9-20afff138564 Job1     601e18bb-40c0-485e-9e63-60524d79ca04           8 stepA           2     False Failed    12/11/2020 6:22:54 PM
2575b8b2-46bc-410b-b2f9-20afff138564 Job1     601e18bb-40c0-485e-9e63-60524d79ca04           8                       False Failed    12/11/2020 6:22:50 PM
```

The email functionality can be leveraged through any Simple Mail Transfer Protocol server. The proposed script is using smtp.mail.yahoo.com on port 587.

Note: For sending email alerts, an alternative to SMTP is to use a SendGrid Account. Send an email from an Azure Automation runbook: https://docs.microsoft.com/en-us/azure/automation/automation-send-email.



[EXTERNAL] Alert Notification - ElasticJob Failed

[redacted]@yahoo.com

To [redacted]

Sat 12/12/2020 1:06 AM

failed_jobs_20201211T2306131921.log
212 KB

```
##################################################
        ALERT NOTIFICATION
##################################################
```

Note: You are receiving this email because Failed Elastic Jobs were identified for your database.

Issue: Elastic Job Failed

Description: The attached file contains a list with Failed Jobs for below resource:

Impacted Database: [redacted]
On Server: [redacted]

File generated at 12/11/2020 23:06:16 UTC time.

How to run and automate alerts

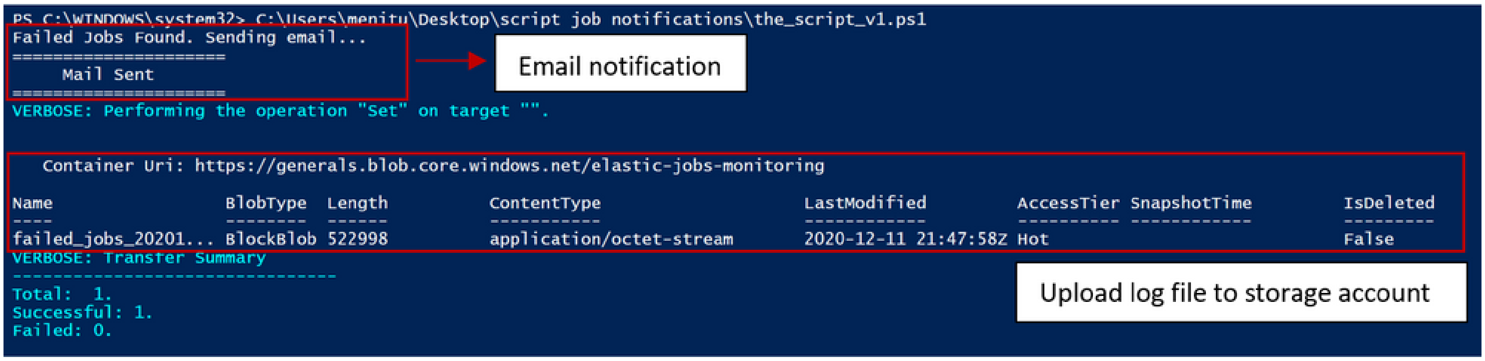
For any of the below options, you have to set the required parameters in the script: Insert SQL Server Details, Credentials, Storage Account (optional) & SMTP server details:

```
$server = '.database.windows.net' # Set the name of the server
$jobDB = ''    # Set the name of the job database you wish to test
$user = '' # Set the login username you wish to use
$passw = '' # Set the login password you wish to use
$FromEmail = 'xxxxxxxxxxxx@yahoo.com' # "from" account must match the smtp server domain
$FromEmailPassw = '' # use app/server access token - it works with account passw as well
$ToEmail = 'xxxxxxxxxxxx@yyyyy.com'
$SMTPServer = "smtp.mail.yahoo.com" #insert SMTP server
$SMTPPort = "587" # or port 465
$StorageAccountName = ''
$StorageContainerName = ''
```

OPTION#1 Run the script on client machine on a schedule in background

In order to run it you need to:

1. Open Windows PowerShell ISE in Administrator mode
2. Open a New Script window
3. Paste the content in script window
4. Run it
5. If failed jobs are found, an alert email will be triggered and the log file containing details on the failed jobs will be attached to the email and either saved locally or sent to a storage account. The result of the script can be followed in the output window.



```
PS C:\WINDOWS\system32> C:\Users\menitu\Desktop\script job notifications\the_script_v1.ps1
Failed Jobs Found. Sending email...
=====================
     Mail Sent
=====================
VERBOSE: Performing the operation "Set" on target "".


   Container Uri: https://generals.blob.core.windows.net/elastic-jobs-monitoring

Name                 BlobType  Length          ContentType                    LastModified         AccessTier SnapshotTime         IsDeleted
----                 --------  ------          -----------                    ------------         ---------- ------------         ---------
failed_jobs_20201... BlockBlob 522998          application/octet-stream       2020-12-11 21:47:58Z Hot                             False
VERBOSE: Transfer Summary
--------------------------------
Total:  1.
Successful: 1.
Failed: 0.
```

5. To run the script in background on a schedule, you can use the following commands:

```
##  =========================================================================
##                Schedule Commands for Client Machines
##  =========================================================================
## To schedule script execution in background, please see below options

#insert parameter values
$script = 'script.ps1' # insert script location path
$Time= 'MM/DD/YYYY HH:MM' # insert desired start time for schedule
$jobName = ''#insert desired job name


# display all scheduled jobs
Get-ScheduledJob

# add new job to run at schedule
Register-ScheduledJob -Name $jobName -FilePath $script -Trigger (New-JobTrigger -Once -At $Time `
    -RepetitionInterval (New-TimeSpan -Minutes 1) -RepetitionDuration ([TimeSpan]::MaxValue))

# command to remove a scheduled job
Unregister-ScheduledJob $jobName
```

OPTION#2 Run the script from Azure Runbook

1. Create a new Automation Account as described here and make sure you choose "YES" for option "Create Azure Run As Account".

2. Import the following Azure Modules by browsing the gallery: Az.Accounts (≥ 2.2.2), Az.Storage, Az.Automation

3. Create a runbook to run the script and make sure you choose Powershell runbook type

4. Add the following login section when connecting from a Runbook

```
##  ============================================================================
##             Login when running from a Runbook
##  ============================================================================
## Enable this code section when running from Azure Runbook
## Get the connection "AzureRunAsConnection" when run from automation account

$connection = Get-AutomationConnection -Name AzureRunAsConnection
Connect-AzAccount `
    -ServicePrincipal `
    -Tenant $connection.TenantID `
    -ApplicationId $connection.ApplicationID `
    -CertificateThumbprint $connection.CertificateThumbprint

"Login successful..."
```

5. Schedule task

Note: Script execution can be monitored in the portal and alerts can be set for any script execution failure.

Public Doc Reference

- The script is publicly available in the following Git repository: https://github.com/MelaniaNitu/AzureSQL
- This blog post contains customer ready information and Powershell script that customer can use to send alerts for elastic jobs failures: https://techcommunity.microsoft.com/t5/azure-database-support-blog/automate-and-send-alerts-for-elastic-jobs-failures/ba-p/1981457

How good have you found this content?

🙂 🙁