

Setup Azure for Student Projects

This document is an attempt to describe the installation and configuration of everything that is needed for “Azure for Student Projects” (A4SP). Using A4SP it is possible to provide access to a group of students within an azure environment in which costs are controlled, access to resources are limited and administration tasks are reduced to a minimum.

As a starting point for the installation

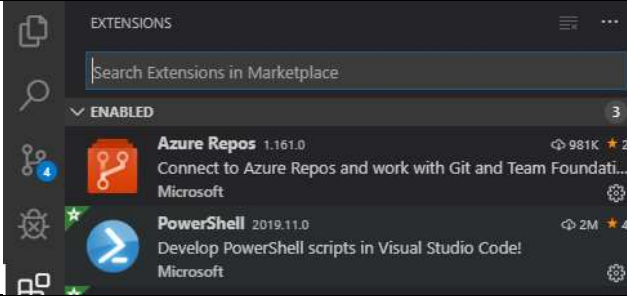
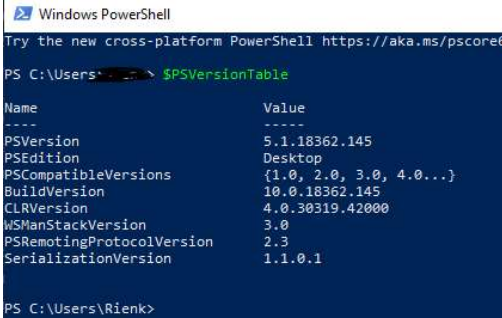
- This document is based upon an Azure customer with an Enterprise Agreement (EA).
- Everything that must be done just once will be done using the Azure Portal. Rational is that automation would take too much time and moreover, it would be less clear for an administrator what is being accomplished by running the script.
- Everything that runs on a regular basis is scripted using PowerShell (Az-modules only where possible).

Setup

To do the Setup, follow all steps in this document and in this order using the exact names as provided. It is assumed that you are the owner of a new Dev/Test subscription in your Azure account having a trust relationship with an Azure Active Directory containing the Users of your organization.

Prerequisites

Setup your environment, first steps.

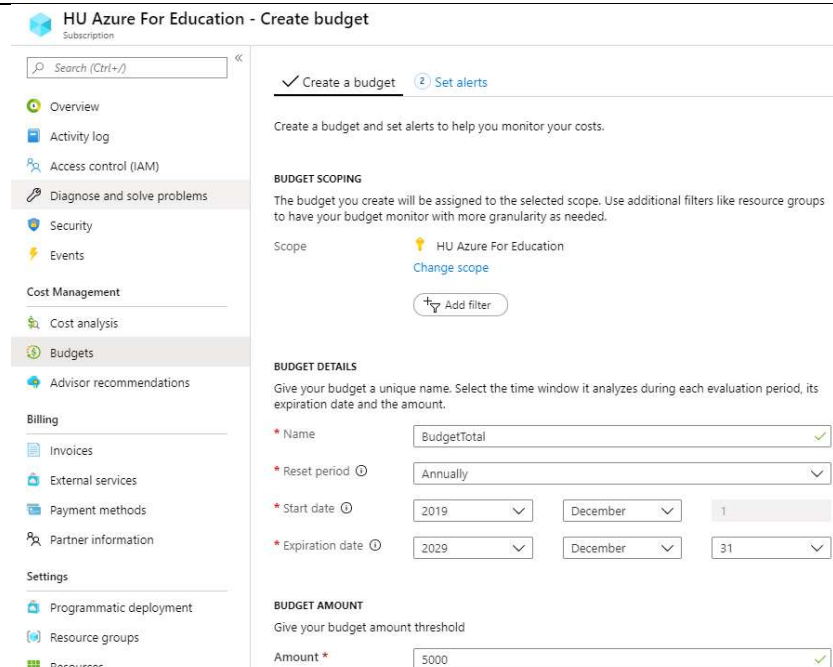
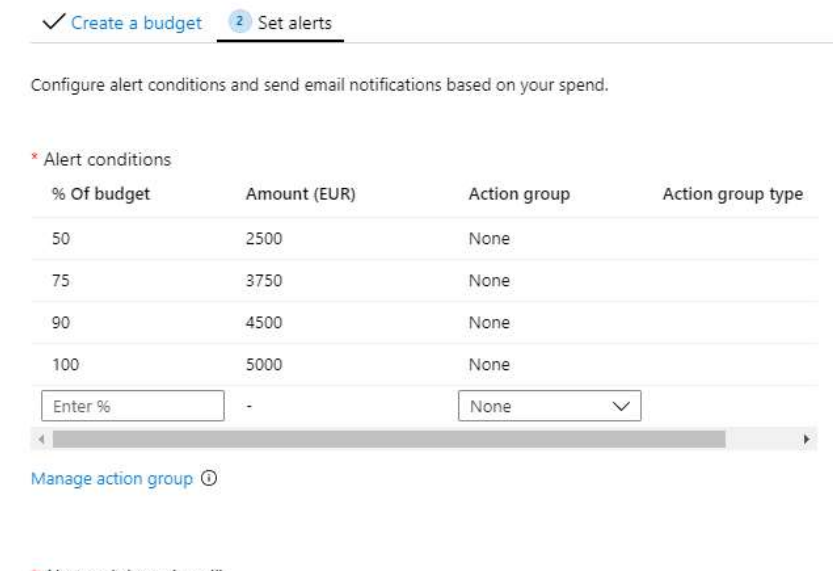
#	Description	Image
1	Install Visual Studio Code using these extensions.	
2	Download / Clone / Whatever: Get the files from this url.	https://dev.azure.com/rienkvanderploeg0902/Azure4StudentProjects/_git/AzureForStudentProjects
3	Rename .\Settings_Settings.json to .\Settings\Settings.json and provide the information that is appropriate for your organization. Note: SendGrid info is part of a later step, don't worry.	<pre>1 { 2 "general": { 3 "name": "HU Azure for Education", 4 "subscription": "e1xxxxxxxxxxxxxxxx", 5 "tenant": "989xxxxxxxxxxxxxxxx", 6 "location": "West Europe", 7 "SendGridPassword": "xxxxxxxxxxxx", 8 "SendGridAPIKey": "xxxxxxxxxxxx" 9 } 10 }</pre>
4	Install Powershell for Azure. Check your PowerShell version first: ;# versie >= 5.1	
5	Install Az –PowerShell module. Accept NuGet provider and PSGallery as being trusted repositories if requested.	Install-Module -Name Az -AllowClobber -Scope CurrentUser
6	From a PowerShell console, run the Setup script	.\Setup\setup.ps1

Capability to send Alerts related Costs on the Subscription Level

First things first:

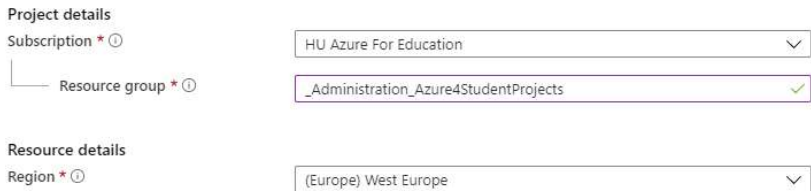
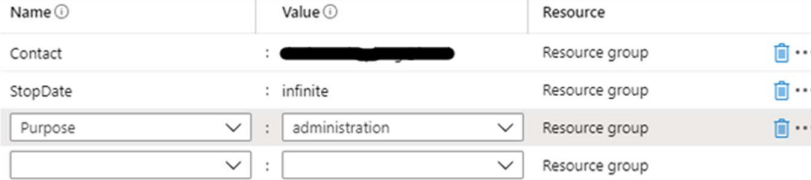
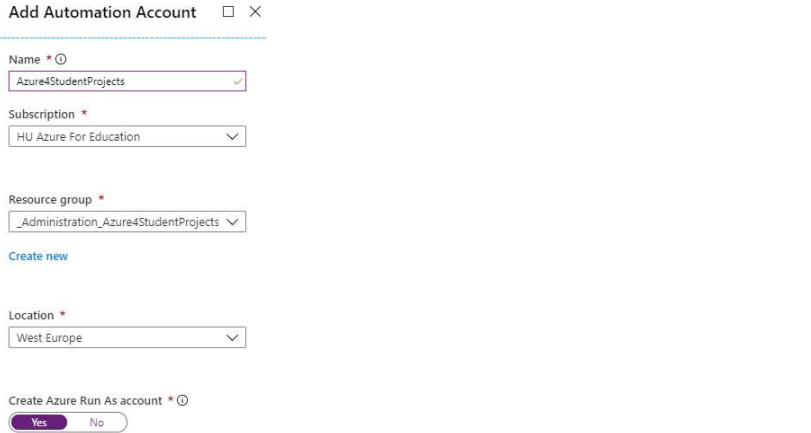
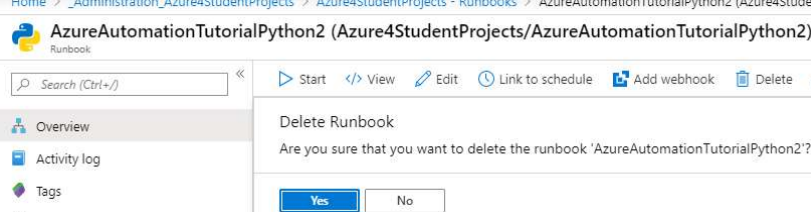
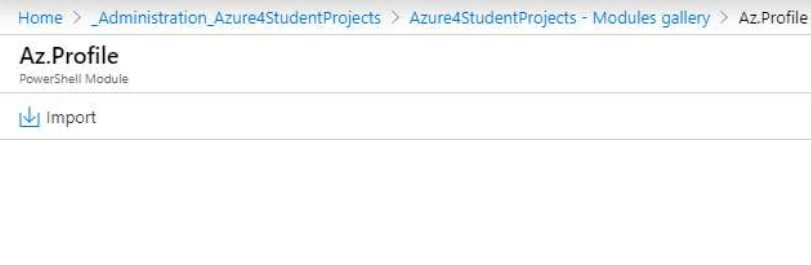
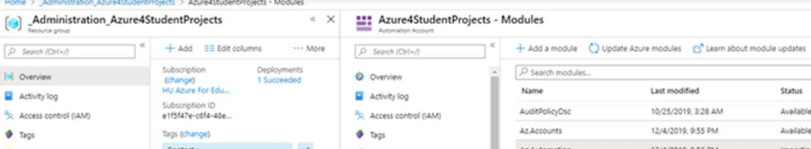
You should be warned about high expenses. This can be done by creating a budget on the subscription level.

Note: not described, but if you like, create budgets for administrative resource groups like "_Administration_Azure4StudentProjects", "_AppServicePlans" and "NetworkWatcherRG" (standard: name=rg-name, budget=500 euro, alert=50%, action=sendAlertToAdministrator)

#	Description	Image																								
1	<p>Goto Subscription and “Create budget”</p> <p>Scope: Subscription!</p> <p>Name: BudgetTotal</p> <p>Reset periode: Annually</p> <p>Start date: First of this month</p> <p>Expiration date: Last day of Last available year in the drop down box.</p> <p>Budget: Depending on your budget / calendar year, for instance 5000 (euro).</p>	 <p>HU Azure For Education - Create budget</p> <p>Search (Ctrl+J)</p> <p>Overview Activity log Access control (IAM) Diagnose and solve problems Security Events Cost Management Cost analysis Budgets Advisor recommendations Billing Invoices External services Payment methods Partner information Settings Programmatic deployment Resource groups Resources</p> <p>✓ Create a budget 2 Set alerts</p> <p>Create a budget and set alerts to help you monitor your costs.</p> <p>BUDGET SCOPING The budget you create will be assigned to the selected scope. Use additional filters like resource groups to have your budget monitor with more granularity as needed.</p> <p>Scope HU Azure For Education Change scope Add filter</p> <p>BUDGET DETAILS Give your budget a unique name. Select the time window it analyzes during each evaluation period, its expiration date and the amount.</p> <p>* Name BudgetTotal ✓</p> <p>* Reset period Annually</p> <p>* Start date 2019 December 1</p> <p>* Expiration date 2029 December 31</p> <p>BUDGET AMOUNT Give your budget amount threshold</p> <p>Amount * 5000 ✓</p>																								
2	<p>Set appropriate alerts, for instance at 50% / 75% / 90% and 100%.</p> <p>Fill in alert recipients, more than one of course!</p>	 <p>✓ Create a budget 2 Set alerts</p> <p>Configure alert conditions and send email notifications based on your spend.</p> <p>* Alert conditions</p> <table><thead><tr><th>% Of budget</th><th>Amount (EUR)</th><th>Action group</th><th>Action group type</th></tr></thead><tbody><tr><td>50</td><td>2500</td><td>None</td><td></td></tr><tr><td>75</td><td>3750</td><td>None</td><td></td></tr><tr><td>90</td><td>4500</td><td>None</td><td></td></tr><tr><td>100</td><td>5000</td><td>None</td><td></td></tr><tr><td>Enter %</td><td>-</td><td>None</td><td></td></tr></tbody></table> <p>Manage action group ⓘ</p> <p>* Alert recipients (email)</p>	% Of budget	Amount (EUR)	Action group	Action group type	50	2500	None		75	3750	None		90	4500	None		100	5000	None		Enter %	-	None	
% Of budget	Amount (EUR)	Action group	Action group type																							
50	2500	None																								
75	3750	None																								
90	4500	None																								
100	5000	None																								
Enter %	-	None																								

Create an Automation Account

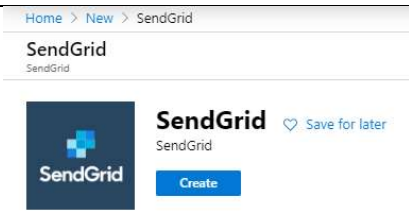
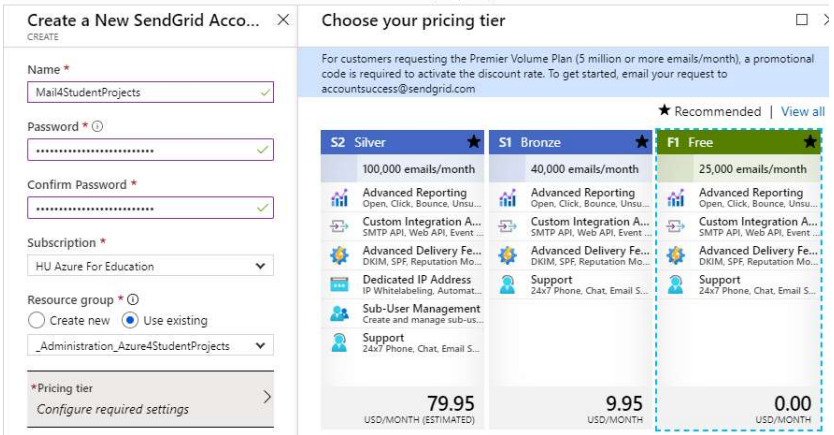
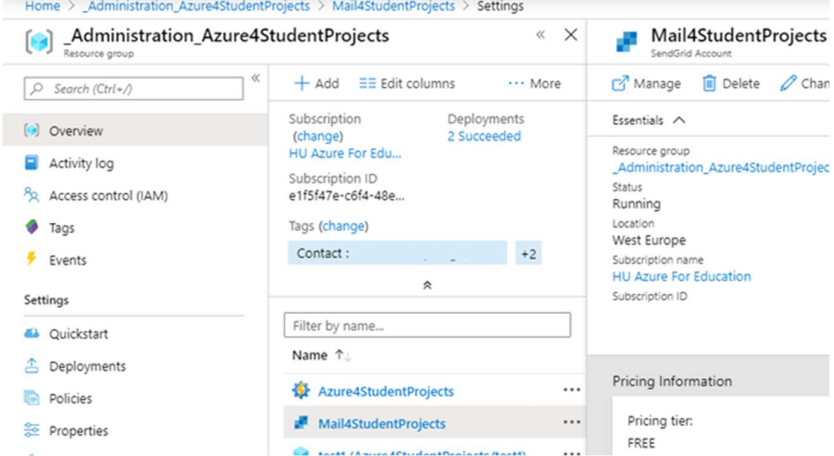
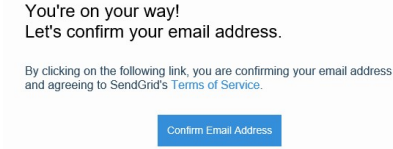
In order to be able to run scripts automatically, for instance to send reports or check budgets and take action if budgets are exceeded you must create an automation account within your subscription.
This should be done by the owner of the subscription.

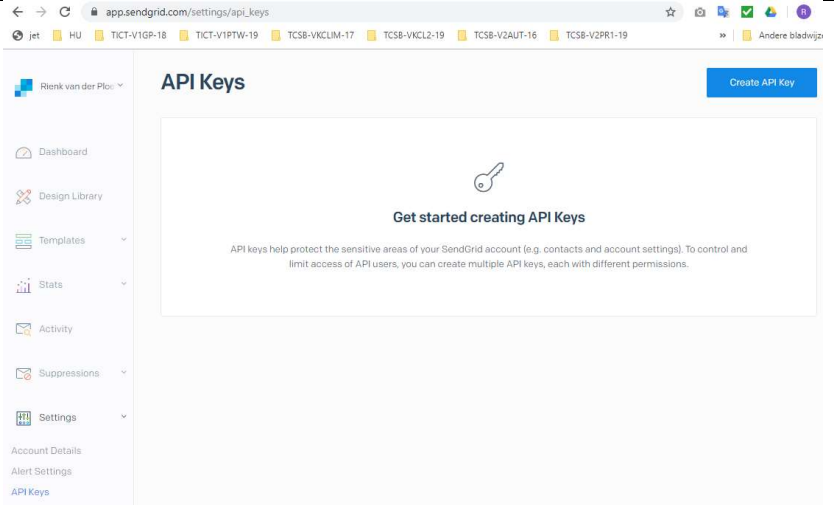
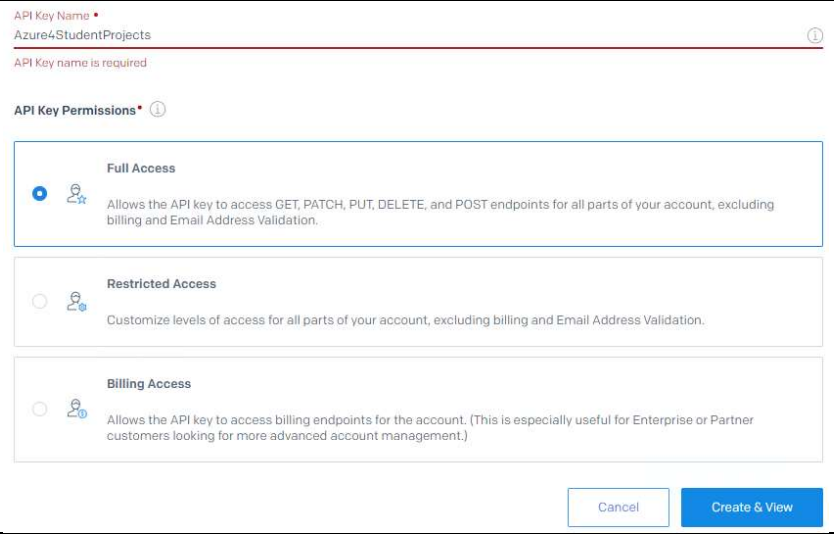
#	Description	Image
1	Create a new ResourceGroup (RG) starting with underscore to distinguish from StudentProjects. Name is: _Administration_Azure4StudentProjects	
2	Provide it with a few tags: Contact, StopDate and Purpose. This is important, for instance, this will prevent being deleted later.	
3	Add an automation account in this RG. Name must be: Azure4StudentProjects Type is: "Run As Account"	
4	Remove all (3) Tutorial scripts from the Resource Group.	
5	Import the Az.Profile module from the Modules Gallery in this automation account. Import more Az.modules in this order: Az.Accounts Az.Consumption Az.Automation Az.Resources Az.Profile Az.KeyVault	
6	Check if import succeeded (might take a long time!) if you like.	

Capability to Send E-mail from an automation runbook Using SendGrid in Azure

In order to send e-mails originating from your runbooks, for instance to send usage reports, you must create a SendGrid account within your subscription. SendGrid is a cloud-based email service that provides reliable transactional email delivery, scalability, and real-time analytics along with flexible APIs that make custom integration easy. Azure customers can unlock 25,000 free emails each month.

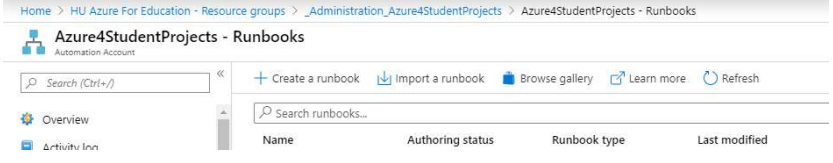
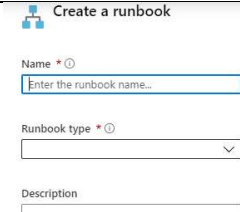
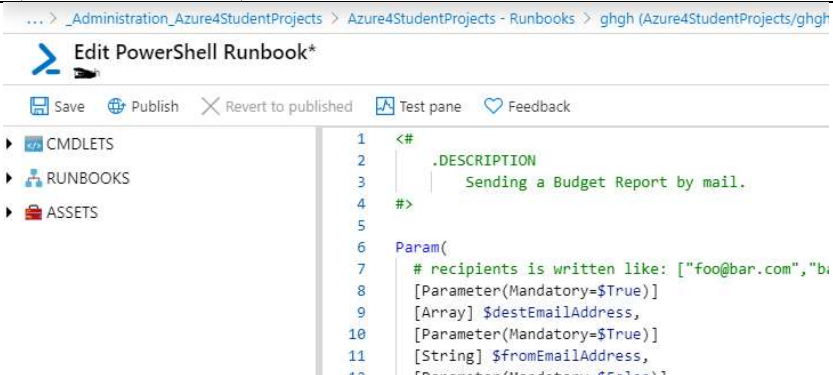
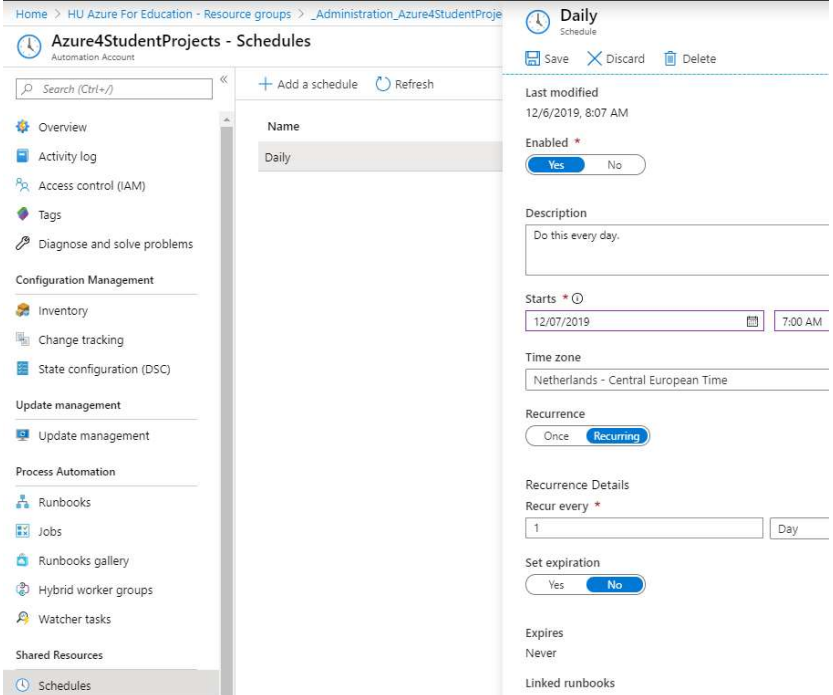
<https://docs.microsoft.com/nl-nl/azure/sendgrid-dotnet-how-to-send-email#create-a-sendgrid-account>

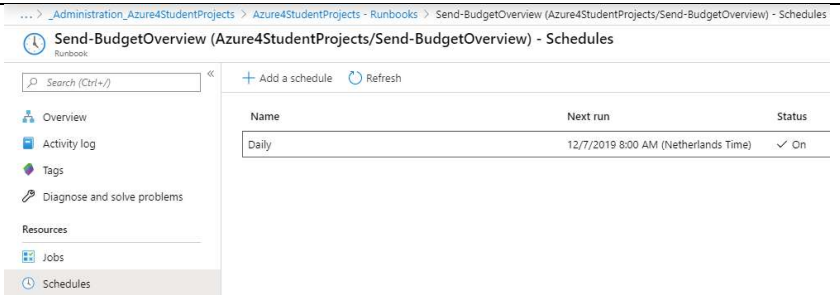
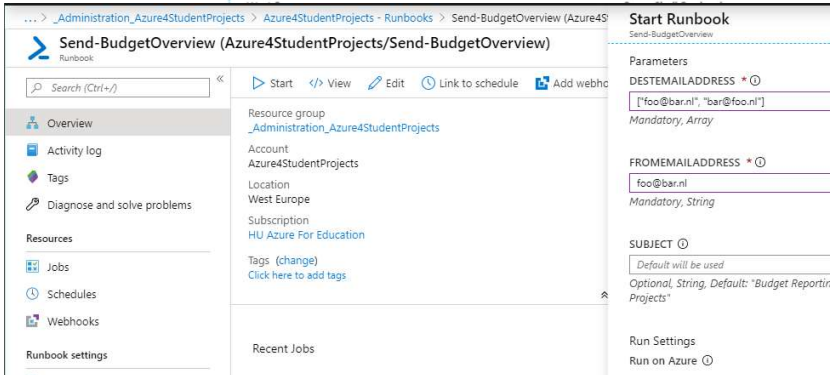
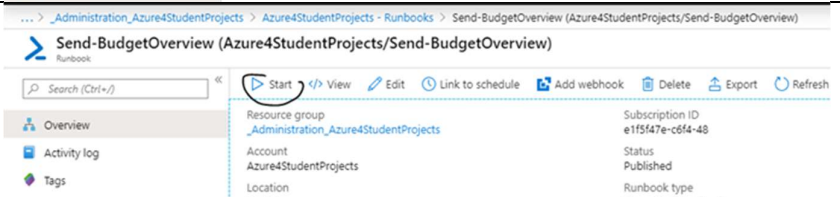
#	Description	Image
1	Logon to the azure portal as subscription owner and Create (Add) SendGrid.	
2	Complete the signup form. Select the F1 Free version. Name: Mail4StudentProjects Password: to be found in Settings\Settings.json ResourceGroup: _Administration_Azure4StudentProjects Contact Information: Owner of the subscription. Click "Create".	
3	After deployment of the SendGrid account succeeded, go to the Resource Group and click on the SendGrid Account Resource. Click on "Manage". You'll be redirected to the app.sendgrid.com for e-mail confirmation.	
4	Click on the button in the mail received. You'll redirected to: https://app.sendgrid.com/	

5	<p>To send an email using SendGrid, you must supply your API Key.</p> <p>Go to your dashboard → Settings → API Keys → Create API Key.</p> <p>Note: If you perform this step at a later moment, goto the website via “Manage”, see step number “7”.</p>	
6	<p>Provide the name: “Azure4StudentProjects”</p> <p>Select “Full Access”</p> <p>and click “Create&View”.</p>	
7	<p>Next step is to Create an Azure KeyVault and store your SendGridAPIKey in that vault. The API key is needed every time you will send an e-mail.</p>	<p>This is done by executing the script: <code>.\Setup\Scripts\Create-AzureAutomationVault.ps1</code></p> <p>References: https://docs.microsoft.com/nl-nl/azure/key-vault/quick-create-portal https://docs.microsoft.com/nl-nl/azure/automation/automation-send-email</p>

Capability to Send Budget Reports on a daily basis to one or more administrators using a runbook

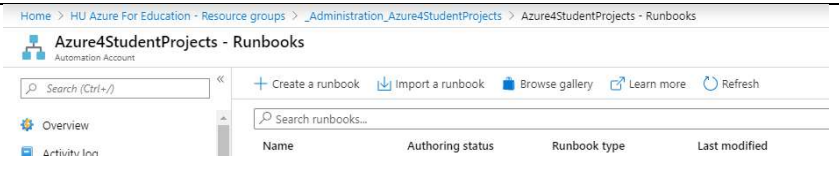
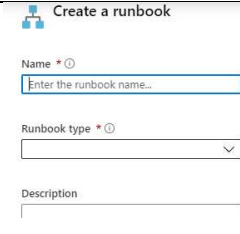
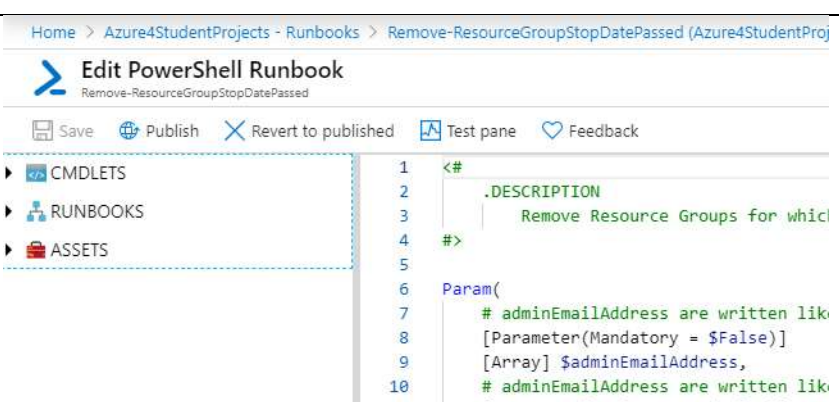
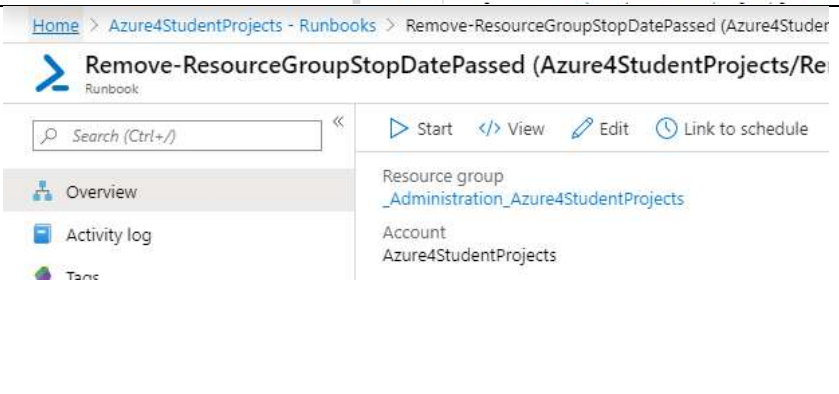
Next step is to create a runbook that sends a budget report every day by mail. It provides an overview of the subscription and budget reports of every resource group sorted by budget that is left and a color of red when budget that is left is < 10 euro.

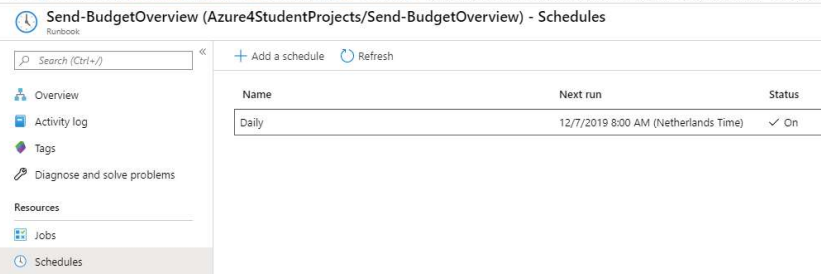

#	Description	Image
1	Create a new Runbook by clicking "Create a runbook" from RG=_Administration_Azure4StudentProjects→ Automation account=Azure4StudentProjects → "Runbooks"	
2	Click "Create a runbook" Name: "Send-BudgetOverview" Type: PowerShell Description: short description if you like. Click: "Create".	
3	Contents of the runbook can be found in the repository with files: .\Setup\Runbooks\Send-BudgetOverview.ps1 Paste it in the editor and press "Save". Press "Publish".	
4	Add a schedule called "Daily" with these properties: Starts: choose the day of tomorrow.	

5	Link (Add) the schedule to your Runbook.													
6	Don't forget to test the runbook.	<div></div>												
7	Result should be something like this in your mailbox:	<div><p>Report: Azure for Student Projects Spending</p><p>Total Cost this year: 1.85 Euro (from a budget of 5000 Euro in total).</p><p>Note, this table is sorted: Resource Groups with the least remaining budget are shown at the top of the list.</p><table><tr><th>Resource Group</th><th>Budget</th><th>Spent</th><th>Budget Left</th></tr><tr><td>simpleVM-Team1</td><td>10</td><td></td><td>10</td></tr><tr><td>simpleVM-Team2</td><td>10</td><td></td><td>10</td></tr></table></div>	Resource Group	Budget	Spent	Budget Left	simpleVM-Team1	10		10	simpleVM-Team2	10		10
Resource Group	Budget	Spent	Budget Left											
simpleVM-Team1	10		10											
simpleVM-Team2	10		10											

Capability to Remove the Resource Group when the Stop Date has passed using a runbook

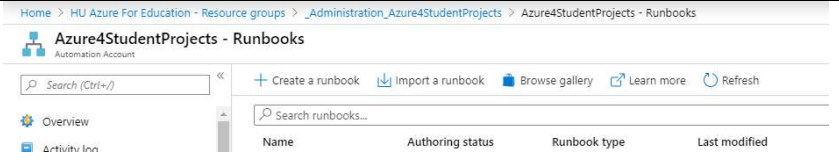
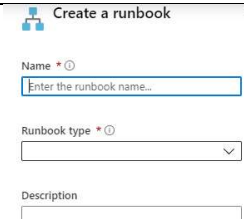
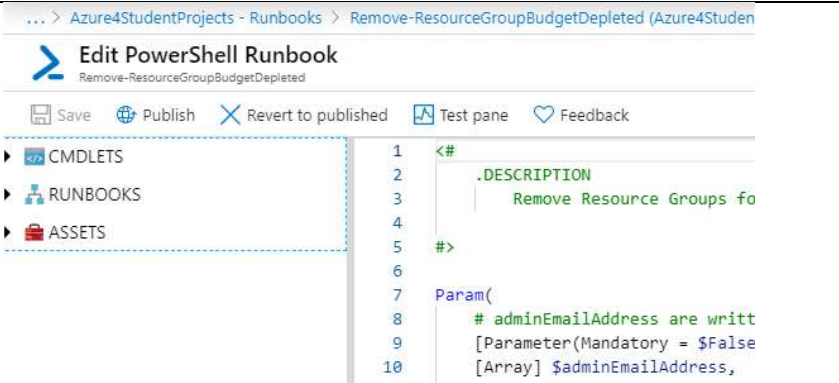

Next step is to create a runbook that removes a resource group when the stop date, that was provided during onboarding, has passed. A notification will be send to its users and administrator.

#	Description	Image
1	Create a new Runbook by clicking "Create a runbook" from RG=_Administration_Azure4StudentProjects→ Automation account=Azure4StudentProjects → "Runbooks"	
2	Click "Create a runbook" Name: "Remove-ResourceGroupStopDatePassed" Type: PowerShell Description: short description if you like. Click: "Create".	
3	Contents of the runbook can be found in the repository with files: .\Setup\Runbooks\Remove-ResourceGroupStopDatePassed.ps1 Paste it in the editor and press "Save". Press "Publish".	
4	Link this runbook to the previous created daily schedule, by clicking "Link to Schedule". Starts: choose the day of tomorrow.	

5	<p>Link (Add) the schedule to your Runbook:</p> <p>“Add a schedule”.</p> <p>Link a schedule to your runbook → Select the “Daily” schedule.</p> <p>Provide the adminemailaddress for recipient and sender and leave the other two fields empty.</p> <p>Press twice “Ok” → “Ok”.</p>	 <p>... > Administration_Azure4StudentProjects > Azure4StudentProjects - Runbooks > Send-BudgetOverview (Azure4StudentProjects/Send-BudgetOverview) - Schedules</p> <p>Send-BudgetOverview (Azure4StudentProjects/Send-BudgetOverview) - Schedules</p> <p>Runbook: Search (Ctrl+/) + Add a schedule Refresh</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Next run</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Daily</td> <td>12/7/2019 8:00 AM (Netherlands Time)</td> <td>✓ On</td> </tr> </tbody> </table> <p>Overview Activity log Tags Diagnose and solve problems</p> <p>Resources Jobs Schedules</p>	Name	Next run	Status	Daily	12/7/2019 8:00 AM (Netherlands Time)	✓ On
Name	Next run	Status						
Daily	12/7/2019 8:00 AM (Netherlands Time)	✓ On						
6	<p>Result should be something like this in your mailbox when the stop date has passed:</p>	 <p>Send Subject Azure Resource Group was removed (Budget depleted).</p> <p>Resource Group(s) removed (stop <u>Date</u> has passed).</p> <p>Via this e-mail we want to inform you that the Resource Group with name 'simpleVM-TeamEndDateTest2' was removed from the 'Azure for Students Platform' because its Stop Date has passed.</p> <p>Unfortunately, there is no possibility to reverse this, or to restore information that was part of this Resource Group.</p> <p>0 Euro was spent in this Resource Group until today.</p>						

Capability to Remove the Resource Group when budget is depleted using a runbook

Next step is to create a runbook that removes the Resource Group and send an e-mail when budget is depleted.

#	Description	Image
1	Create a new Runbook by clicking "Create a runbook" from RG= _Administration_Azure4StudentProjects → Automation account=Azure4StudentProjects → "Runbooks"	
2	Click "Create a runbook" Name: "Remove-ResourceGroupBudgetDepleted" Type: PowerShell Description: short description if you like. Click: "Create".	
3	Contents of the runbook can be found in the repository with files: .\Setup\Runbooks\Remove-ResourceGroupBudgetDepleted.ps1 Paste it in the editor and press "Save". Press "Publish".	
3	Go to your subscription → Budgets → BudgetTotal → Edit budget → "Set alerts". Click "Manage action group" Click "add action group"	

4

Add the action group using the exact names as provided in the image:

get > Manage actions > Add action group > Configure Runbook

Add action group

Action group name *

Short name *

Subscription *

Resource group *

Actions

Action group name *	Action Type *	Status	Configure	Actions
RemoveRGBudgetAlert	Automation Runbook		Edit details	✕
Unique name for the action	Select an action type			

[Privacy Statement](#)

[Pricing](#)

i Have a consistent format in emails, notifications and other endpoints irrespective of monitoring source. You can enable per action by editing details. [Learn more](#)

Configure Runbook

Run runbook *

Runbook source *

Subscription *

Automation account *

Runbook *

Parameters [Configure parameters](#)

i When the alert is triggered, the alert data will be passed to the runbook in the \$WebhookData input parameter. [See this article for more information.](#)

Enable the common alert schema *

5

Click "Configure parameters" in the same screen and choose the right e-mail addresses for the sender and the recipients of the removal (end users will be informed as well).

Click "Ok", "Ok".

Configure Runbook

Run runbook *

Runbook source *

Subscription *

Parameters

Remove-ResourceGroupBudgetDepleted

Parameters

ADMINEMAILADDRESS

Optional, Array

FROMEMAILADDRESS

Optional, String

Capability to get an alert when a runbook fails

Runbooks are essential in this process. Therefore, it makes sense to forward Azure Automation job data to Azure Monitor logs. This provides insight on the automation jobs, gives the possibility to trigger an email or alert based on the runbook job status etcetera.

Reference information:

<https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/automation/automation-manage-send-joblogs-log-analytics.md>

#	Description	Image
1	<p>Create a Log Analytics workspace. Settings:</p> <p>Workspace name: Azure4StudentsLogAnalyticsWorkspace</p> <p>Resourcegroup: _administration_azure4studentprojects</p> <p>Pricingtier: Pay-as-you-go (Per GB 2018)</p> <p>Location: West Europe</p>	
2	Find the resourceid for your automation account	Get-AzResource -ResourceType "Microsoft.Automation/automationAccounts"
3	Find the resourceid for your log analytics workspace	Get-AzResource -ResourceType "Microsoft.OperationalInsights/workspaces"
4	<p>After running the commands to the right, it may take an hour before you start to see records in Azure Monitor logs of new JobLogs or JobStreams being written.</p> <p>To see the logs, run the following query in log analytics log search: AzureDiagnostics where ResourceProvider == "MICROSOFT.AUTOMATION"</p>	<pre>\$workspaceId = "[resource id of the log analytics workspace]" \$automationAccountId = "[resource id of your automation account]" Eg: \$workspaceId = "/subscriptions/<MYSUBSCRIPTIONID>/resourceGroups/_Administration_Azure4StudentProjects/providers/Microsoft.OperationalInsights/workspaces/Azure4StudentsLogAnalyticsWorkspace" \$automationAccountId = "/subscriptions/<MYSUBSCRIPTIONID>/resourceGroups/_Administration_Azure4StudentProjects/providers/Microsoft.Automation/automationAccounts/Azure4StudentProjects" Set-AzDiagnosticSetting -ResourceId \$automationAccountId -WorkspaceId \$workspaceId -Enabled 1</pre>
5	To Send an email when a runbook job fails or suspends: Click "Alerts" in loganalyticsworkspace. Click "New alert rule"	
6	<p>Select: your LogAnalyticsWorkspace resource as the resource to monitor.</p> <p>Click "Done"</p>	

7	<p>Add a condition: “Custom log search” (click on this link).</p>	<div><h3>Configure signal logic</h3><p>Choose a signal below and configure the logic on the next screen to define the alert condition.</p><div><div>Signal type ⓘ All</div><div>Monitor service ⓘ All</div></div><p>Displaying 1 - 20 signals out of total 112 signals</p><div><div>🔍 Search by signal name</div><div>Signal name↑↓Signal type</div><div>Custom log searchLog</div></div></div>					
8	<p>Add (log) search query:</p> <p>Alert logic: Threshold value=0</p> <p>Resulting in logic: number of results > 0.</p> <p>Period (in minutes): 1440, Frequency (in minutes): 15.</p> <p>Click “Done”.</p>	<p>AzureDiagnostics where ResourceProvider == "MICROSOFT.AUTOMATION" and Category == "JobLogs" and (ResultType == "Failed" or ResultType == "Suspended") and (TimeGenerated > now()-35m)</p>					
9	<p>Create an Action (group):</p> <p>Click “Edit details”</p>	<div><h3>Add action group</h3><div><div>Action group name * ⓘsendAlertToAdministrator</div><div>Short name * ⓘAzure Alert</div><div>Subscription * ⓘHU Azure For Education</div><div>Resource group * ⓘ_Administration_Azure4StudentProjects</div></div><div><div>Actions</div><table><thead><tr><th>Action group name *</th><th>Action Type *</th><th>Status</th><th>Configure</th></tr></thead><tbody><tr><td>AzureAdminAlert</td></tr></tbody></table></div><div>Email/SMS/Push/Voice</div><div></div><div>Edit details</div></div>	Action group name *	Action Type *	Status	Configure	AzureAdminAlert
Action group name *	Action Type *	Status	Configure				
AzureAdminAlert							

| 10 | Configure actions at your convenience: Click “OK”. | ☒ Email Email * email@example.com ☒ SMS (Carrier charges may apply) Country code * 1 Phone number * 1234567890 ☒ Azure app Push Notifications Azure account email * ⓘemail@example.com ☐ Voice Country code 1 Phone number 1234567890 Enable the common alert schema. [Learn more](#) Yes No |

11	<p>Create/modify alert details and click "Create Alert Rule".</p> <p>Test the rule by failing a runbook!</p>	<div><div>Customize Actions</div><div><div><input checked="" type="checkbox"/> Email subject ⓘ</div><div>Subject line * ⓘ Automation task Failed</div><div><input type="checkbox"/> Include custom json payload for webhook ⓘ</div></div></div> <div><div>ALERT DETAILS</div><div><div>Alert rule name ⓘ runbookFailed</div><div>Description Automation runbook failed.</div><div>Severity * ⓘ Sev 3 ▼</div><div><input checked="" type="checkbox"/> Suppress Alerts ⓘ</div><div>Suppress alerts for (in minutes) * 30</div></div></div>
----	--	--