# TBS Monitoring and Alerting Automation Solution Design

## Requirements

The monitoring and alerting framework is designed to meet the following requirements:

* Flexible while minimizing tag maintenance effort
* Support for one or more teams to be notified for a particular type of alert, without notifying teams that shouldn’t be notified
* Fully automated so that when new resources are created that require monitoring and alerting, the applicable alert rule sets are automatically created for that resource.
* When broadscale changes are required to an alert trigger condition (eg. across hundreds of resources), the changes can be rolled out automatically to all affected alert rules

## Solution design

The Azure native components leveraged by the framework are:

* Action Groups which define who or what to notify (ie. contact information for a support team member, an Azure runbook or Azure Function, etc)
* Alert Rules which define the conditions under which the alert should trigger, as well as which Action Groups to notify. The solution uses both Metric and Log Query alert rules types.
* Azure resource group and resource level tags. The tags described below will be used to control which Action Groups are notified as well as which Alert Rules sets apply for a particular resource.

Related sets of Alert Rules are grouped into “alert rule sets” (this is our own concept). Each resource type in azure will have a corresponding default set of alert rules, referred to as “standard” rule sets. These are implicit and do not need to be specified in the alertRuleExtensions tag (eg. windowsVMStandard, applicationInsightsStandard, etc).

The solution uses resource level and resource group level tags to define whether alerting should be enabled for a given resource or resource group, what alert rules should be applied, and which operations support team(s) to notify.

The tags that are used are as follows:

1. The “alert” tag, which when set to True enables alerting for an individual resource or all resources in a resource group (see table below for more details).
2. The “alertRulesExtensions” tag, which defines which alert rule extension sets that should be created for a particular resource, in addition to the standard set of alert rules corresponding to the resource. For example, the sqlServerIaaSLogAnalyticsQueries extension set defines a set of alert rules for SQL Server performance metrics.
3. The “alertActions” tag, which defines which support teams (defined via Action Groups) should be notified for one or more alert rule sets. This tag can be defined at the individual resource level and/or the resource group level (see table below for more details).

For new app services or VMs experiencing too many alerts at the beginning, or for resources that require special alert trigger conditions, the alert tag can be set to False after the alert rules have been initially created in order to customize the trigger conditions without having the alert rules automatically update on the next run of the script. Later on, if desired, the alert tag can be reset to True in order to have the alert rules automatically update to the default configuration.

There are 3 different types of alert rules:

1. Metric alert rules based on Azure Monitor supported metrics. These alert rule are scoped to a particular resource (eg. a particular VM, app service, application insights instance, etc).
2. Metric alert rules based on Log Analytics Workspace metrics. These alert rules are scoped to the log analytics workspace that a resource is logging to. The names of log analytics metric alert rule sets must terminate with “LogAnalyticsMetrics”.
3. Log analytics query based alert rules. These alert rules are scoped to the log analytics workspace that a resource is logging to. The names of log analytics query alert rule sets must terminate with “LogAnalyticsQueries”.

### Alert Rule template definitions

The solution uses two Alert Rule templates defined in the templates folder:

1. metric-alert-rule-template.json which is a parameterized ARM template for creating both regular metric alert rules and log analytics metric alert rules.
2. log-analytics-query-alert-rule-template.json which is a parameterized ARM template for creating log analytics query based alert rules.

### Alert Rule set definitions

The default trigger conditions for alert rules and other alert related parameter values are stored in JSON parameter files, and referenced when deploying the alert rule ARM templates.

### File structure in Git

The alert rule JSON parameter files are organized into folders matching the name of the corresponding alert rule set, as shown below:

log-analytics-query-alert-rules

sqlServerIaaSLogAnalyticsQueries

lock-wait-time.json

…

linuxVMStandardLogAnalyticsQueries

syslog-errors.json

…

windowsVMStandardLogAnalyticsQueries

netlogon-failure.json

…

metric-alert-rules

windowsVMStandardLogAnalyticsMetrics

low-memory-alert.json

low-disk-space.json

…

linuxVMStandardLogAnalyticsMetrics

low-memory-alert.json

low-disk-space.json

…

…

templates

metric-alert-rule-template.json

log-analytics-query-alert-rule-template.json

Although the script itself is stored in the Git repo and pulled by the runbook, it is not actually used for security reasons (in case an unauthorized change was made to the repo that could delete resource in the TBS subscription ). The script that is run is the one directly associated to the runbook.

The Azure Automation powershell runbook will:

* Pull the above file structure from the GIT repo to the local filesystem temp folder (using Azure DevOps credentials stored as an Azure Automation account credential)
* Reads the tags of resource groups and resources within those resource groups and determines for each resource what Alert Rules sets should be created and which Action Groups should be notified based on tag settings.
* For each Alert Rule set that applies, the script then loops through all Alert Rule JSON parameter files defined for that Alert Rule set does an ARM template deployment for each one by passing in the JSON parameter file and the corresponding Action Group Ids as parameters to create the Alert Rules.
* The runbook powershell script will run once a day (scheduled) , picking up any newly tagged resources that need alert rules and rolling out any changes made to existing alert rule definitions.

Note that the powershell script will only create or update alert rules, it will not delete them, even if the “alert” tag is toggled from True to False. Alert rule deletion is anticipated to be rare, and as such will need to be done manually.

## Resource Group Level tagging

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag name** | **Purpose** | **Format** | **Example** |
| alert | Indicates whether alerting should be enabled or disabled for all resources in the resource group. This setting can be overridden at the individual resource level. | String, with a value of True or False |  |
| alertActions | For one or more alert rule sets, defines who should be notified if one of the alerts in the set triggers. | JSON string of the format:  {  “comma separated list of alert rule set names 1”: “comma separated list of action group names”,  “comma separated list of alert rule set names 2”: “comma separated list of action group names”,  …  }  In place of a list of alert rule set names, a wildcard (\* character) can be used to designate “All Alert Rule Sets”. | An example value for the alertActions tag:  {  “\*”: “Resource Group XYZ Support Team”,  “windowsVMStandard,  linuxVMStandard,  windowsVmStandardLogAnalyticsMetrics,  linuxVmStandardLogAnalyticsMetrics”: “Infrastructure Support Team”  }  In the above example the Infrastructure Support Team will be notified if one of the standard Alert Rules for VMs triggers for any resource in the resource group, and the Resource Group XYZ Support Team will be notified if any Alert Rule for any resource in the resource group triggers. |

## Resource Level tagging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tag name** | **Purpose** | **Format** | **Example** | **Notes** |
| alert | Indicates whether alerting should be enabled or disabled for the resource. If defined, it will override the setting at the resource group level. | String, with a value of True or False |  | If not defined at the resource level, it will default to the resource group’s “alert” tag value; otherwise if the tag is not defined at either level, a value of False is assumed |
| alertRuleExtensions | Names of any additional alert rule sets (ie. alert rule extensions) that should be applied in addition to the standard set. The standard set is derived based on resource’s resource type. | Comma separated list of alert rule set names. | For example, for a VM that is running SQL Server, the value could be set to **sqlServerIaaSLogAnalyticsQuery** in order to include alerting on SQL Server performance counters. | Not applicable |
| alertActions | For one or more alert rule sets, defines who should be notified if one of the alert rules in the set triggers for the resource. | JSON string of the format:  {  “comma separated list of alert rule set names 1”: “comma separated list of action group names”,  “comma separated list of alert rule set names 2”: “comma separated list of action group names”,  …  }  In place of a list of alert rule set names, a wildcard (\* character) can be used to designate “All Alert Rule Sets”. | An example value for the alertActions tag:  {  “sqlServerIaasLogAnalyticsQueries”: “SQL Server Support Team”  }  In the above example the  SQL Server Support Team is notified if one of the SQL Server Alert Rules trigger. | Any alert actions defined in the “alertActions” tag at the resource group level will be additive to those defined at the resource level. |