Setting up Turbonomic Action Notifications to Slack Channels

An interesting use-case that I've bumped into lately is where folks want to enable automation, but they also need to know when automated things happen. Email was the common platform for notifications, and still is, but there are many more organizations adoption Slack for day-to-day activity monitoring and building out interesting interactive ways to enable the ChatOps approach to IT operations management.

Since you may have followed along my first article which showed you how to set up a custom WebHook integration for your Slack team channel, we will take that one step further and show you how to configure Turbonomic to send notifications of actions to your Slack channel.

Setting up Action Scripts in Turbonomic

One of the cool features within Turbonomic is something called Action Scripts. These are scripts that are run when a particular actions happens on a particular entity within the environment. Action scripts run at different times in the process including before (PRE) and after (POST) the action so that you can either get notification or to trigger some interaction with the action.

Action Scripts run for every action time available including moves, scale/resize, and more. The naming of each Action Script is relative to the timing (PRE/POST) and the action type. You only need to create one Action Script which is hosted on your Turbonomic control instance and launched by the Turbonomic engine as actions are triggered.

The official documentation on using Action Scripts is here, but for our purposes here I will give you a crash course in creating a PRE move script so that we can send Slack notifications when an application workload is about to move.

Variables Accessible during Action Script Execution

There are a number of environment variables which are generated when a Turbonomic action is instantiated. Some of these include:

$VMT_TARGET_NAME – the entity which is subject to the move action
$VMT_CURRENT_NAME – the source location where the entity is located
$VMT_NEW_NAME – the destination where the entity will be moved
$VMT_ACTION_NAME – the unique ID for the action

These are the ones that I've chosen to include for my Slack notifications because I will want to know the workload which is subject to the move, the source location, target location, and then having the ID of the action is helpful for auditing and also for more deeper integration with a true ChatOps approach that we will dive into further in another post.

For now, the Slack notifications will be simply to log for us using our Slack channel whenever there are moves occurring. You can select from any of the different actions in the Action Scripts, so this is a good place to start.

The PRE_MOVE_VirtualMachine.sh Script

The simplest view of the script is as follows. Simply create a file named **PRE_MOVE_VirtualMachine.sh** which is the one that is called by a move action. This could be anything from a VM migration across hosts, clusters, or also container pod changes and more.

We need to leverage the action variables that we have been given and pass them into the our Slack API call. The simplest method for this is to inject a cURL command into the Action Script that will run using the native cURL command available on your Turbonomic instance.

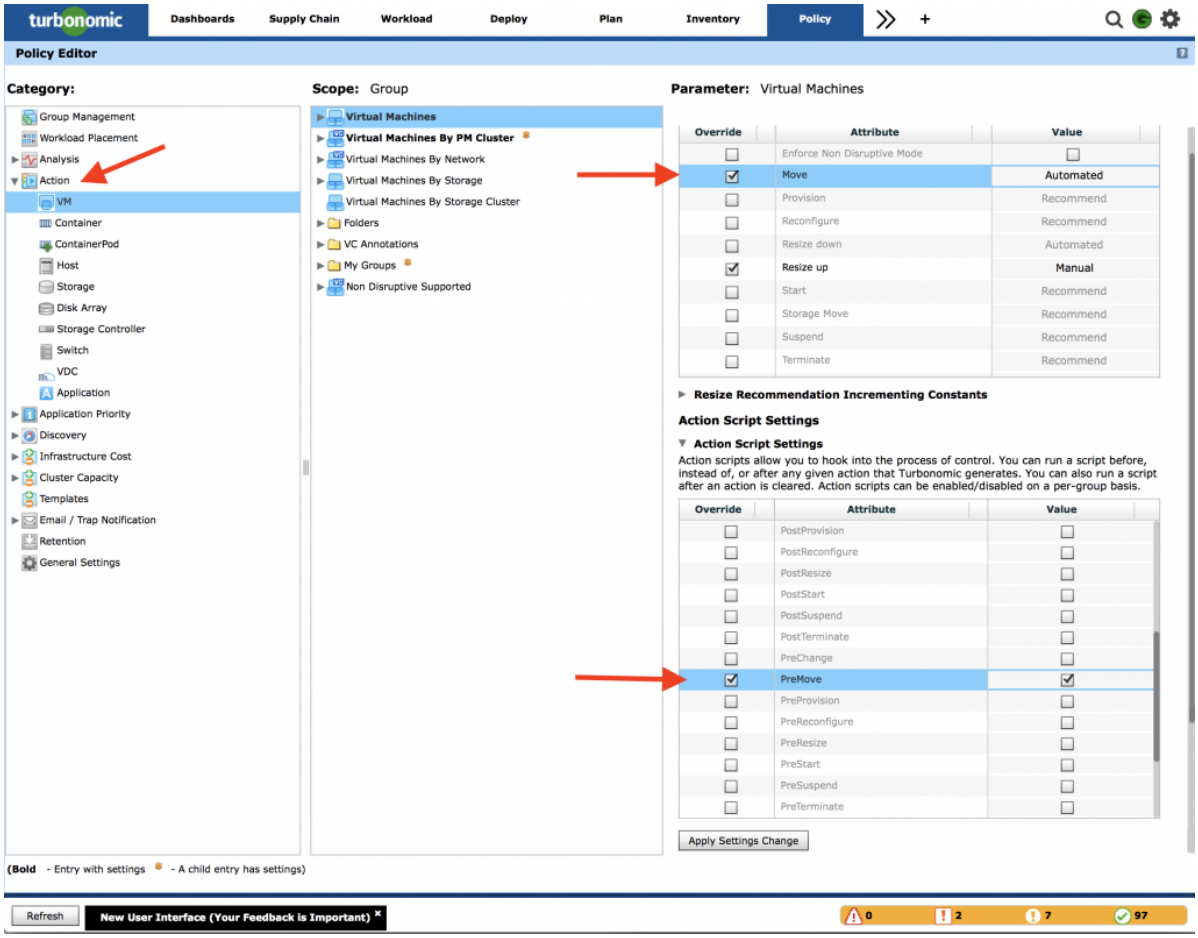
The command to post to the API for Slack requires your WebHook URL [which you can get by following this guide that helps you get the WebHook set up](#).

This is the full GitHub Gist of the code. If you have existing Action Scripts in the folder, you can simply append these lines to your existing script.

Take note of the use of quotes within the command line as we need to pass the variables into the cURL command which requires additional double-quotes around the entire command.

Last step – Enable Action Script for Moves in Turbonomic

At the time of this writing, the Action Scripts features are still in the traditional flash UI. Go to the **Policy view** in your Turbonomic instance and expand the **Action | VM** section where we will enable the Action Scripts for Virtual Machines in this case.



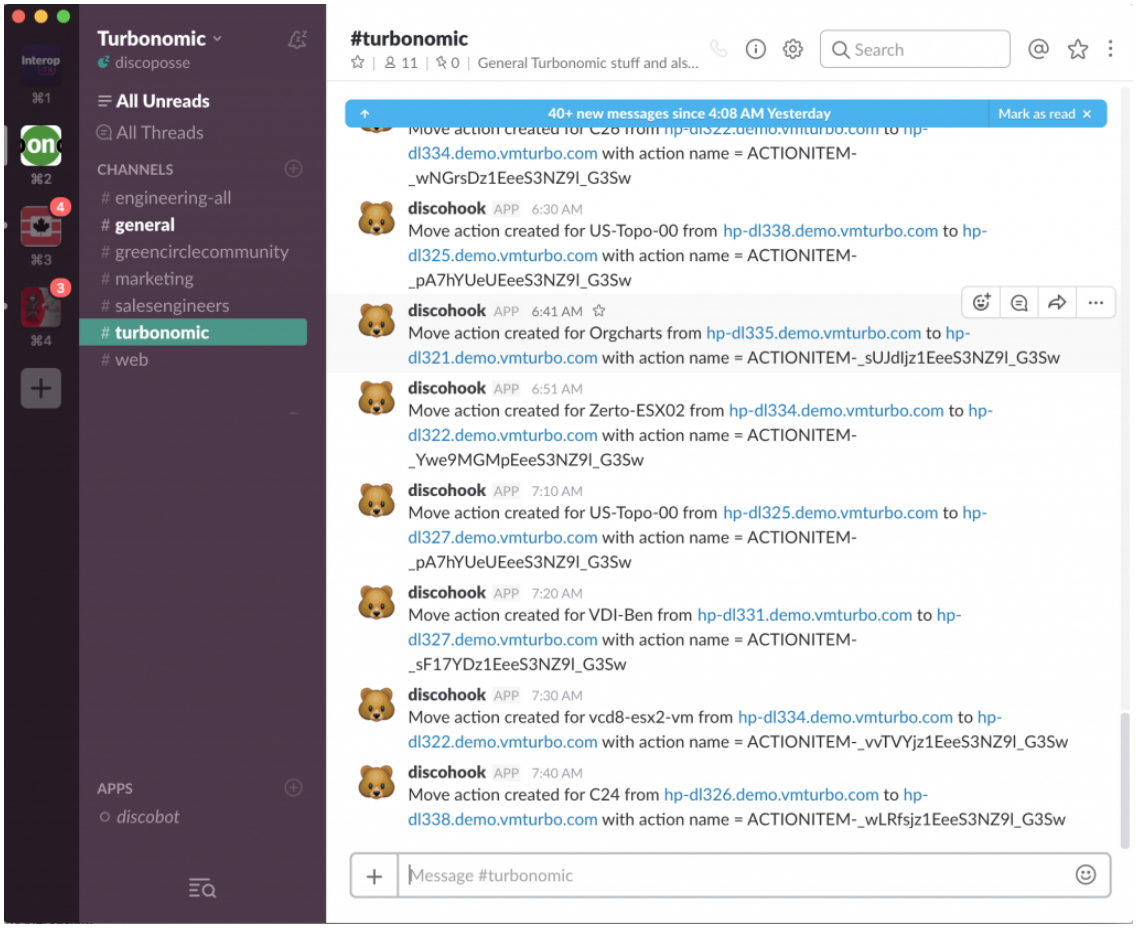
Simply check off the **Action Script Settings** setting for the **PreMove** action and you are all set. In

the image above we can see that I also have Move actions automated which may be set to Manual for your environment.

NOTE: Enabling policy changes within Turbonomic will trigger a refresh of the actions. This is because the state of your policies has changed and the entities in the environment must shop for the appropriate resources to satisfy their demand based on the newly formed policy configuration. This is the nature of the system being real-time so that no actions are held when they could be stale or unnecessary due to other environmental changes that have occurred.

The Slack View

Under your Slack channel, you will now begin seeing notifications whenever an action occurs. This is what your channel will start to look like as the moves take place:



In my case, I have enabled full automation; This means that these actions are triggered and the notification is done as the action is about to occur. We can also do POST_MOVE script which is handy if we are building out other hooks.

The goal with Action Scripts is to be able to integrate with any application lifecycle management process or product. Look for much more in the coming weeks as we walk through some more integrations that can be done with this method.