

Source: C# Corner (www.c-sharpcorner.com)

PRINT

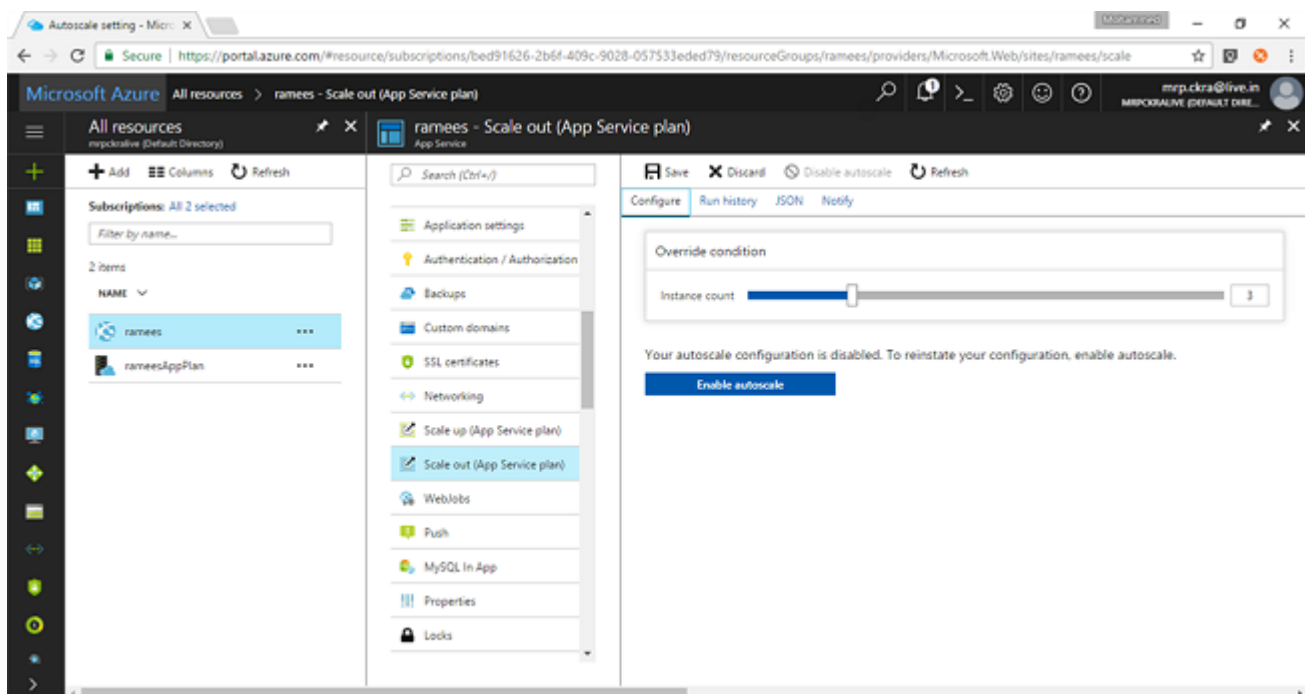
Article

Azure Web App Scaling - Part Two

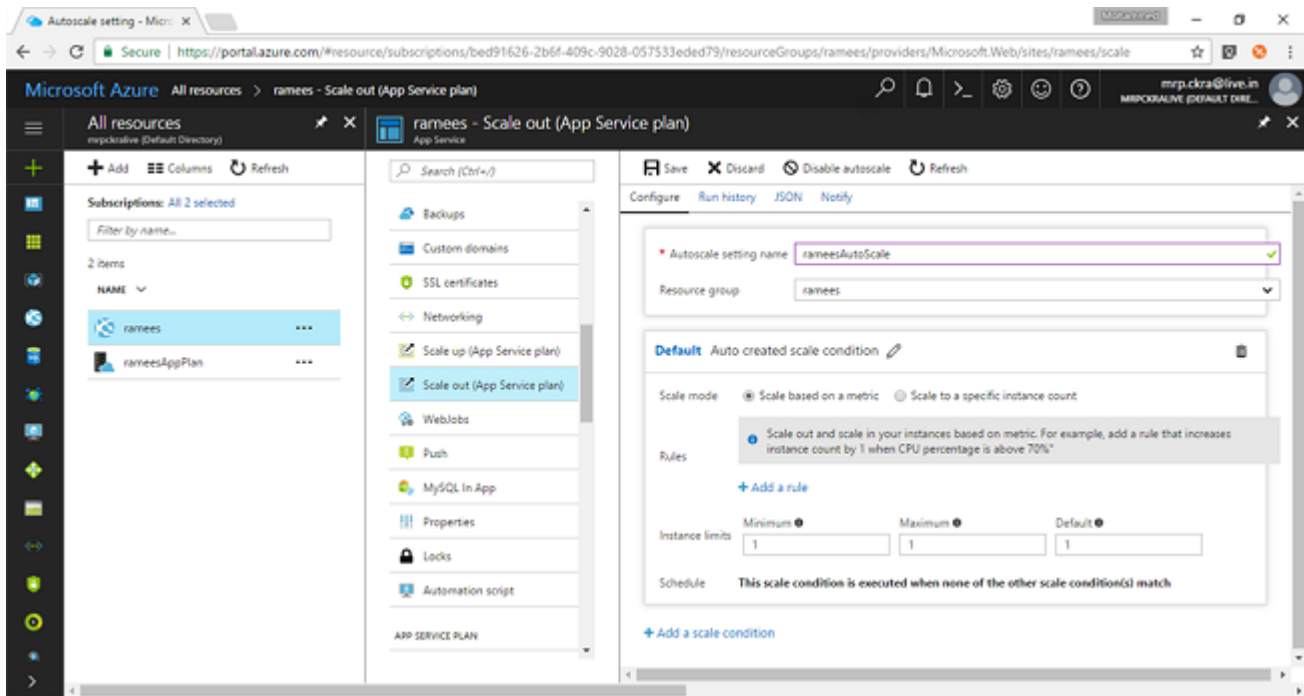
By [Ramees](#) on May 14 2017

In part one, we saw how to manually scale our app, and here, we are going to see how we can enable auto scale in our app. Auto scale happens based on some metrics and parameters like usage of disk, and memory or CPU beyond a limit. It will automatically add more instances when the usage is exceeded. Also, we can schedule it for some specific time or day, even though there are some limitations per plan, and you can't scale beyond a limit.

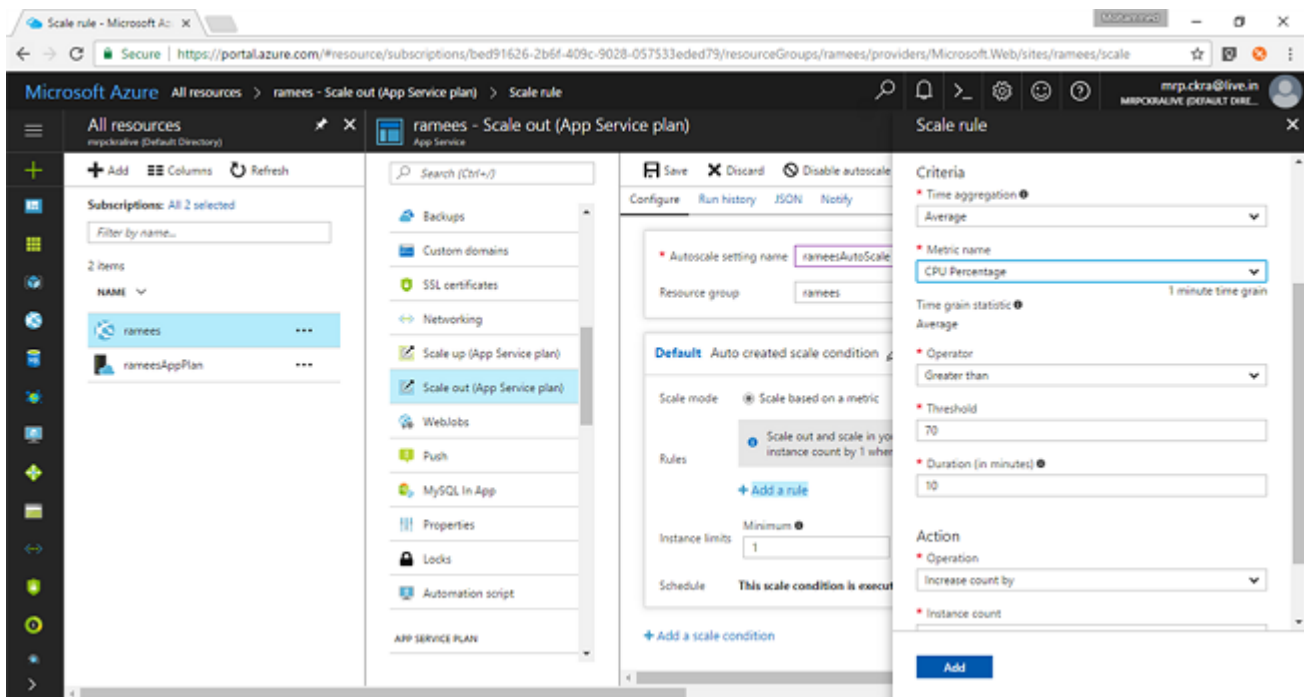
We are back to our scale out section of our app created in [part one](#) in which we set three instances manually and you can see that the auto scale functionality is disabled by default.



When I enable the auto scale, you can see that it will create a default Auto scale condition. It is possible to have more than one condition and to have the name "Auto created scale condition" of your choice. You have to give a name to this Auto scale setting also, from the top. So, this setting may have multiple conditions.



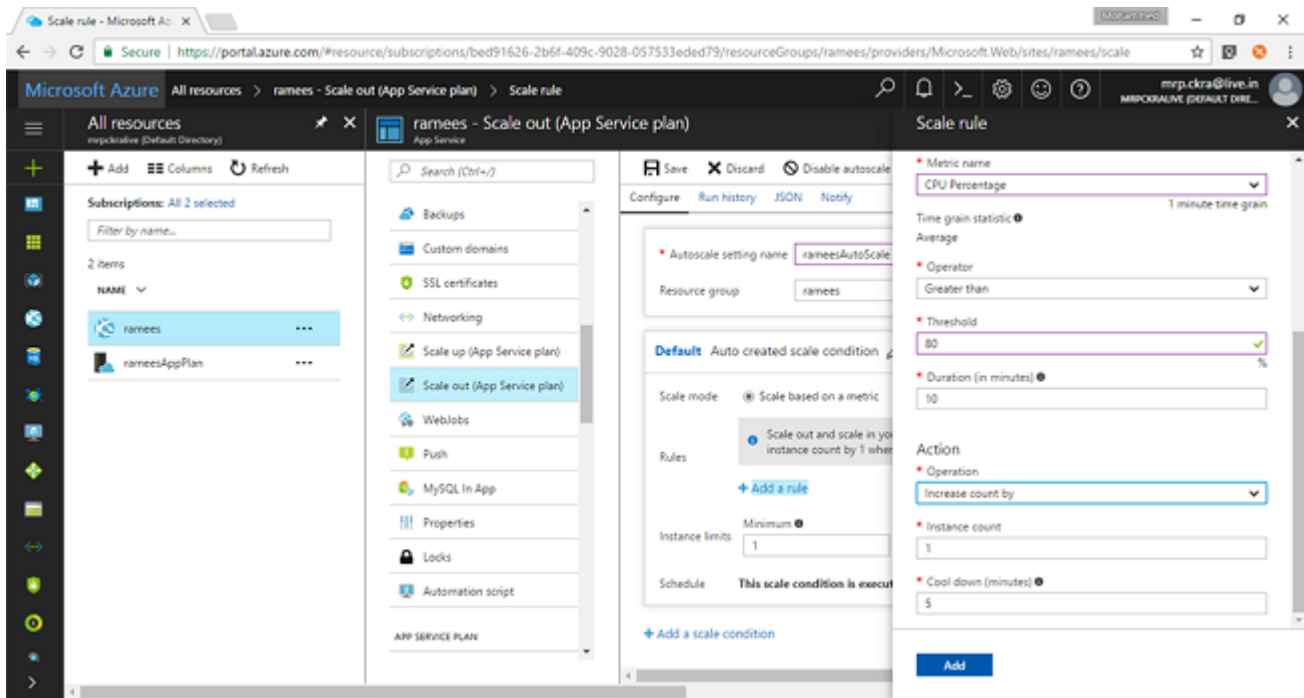
When we click on "Add a rule", it will pop up the following window to define the rule.



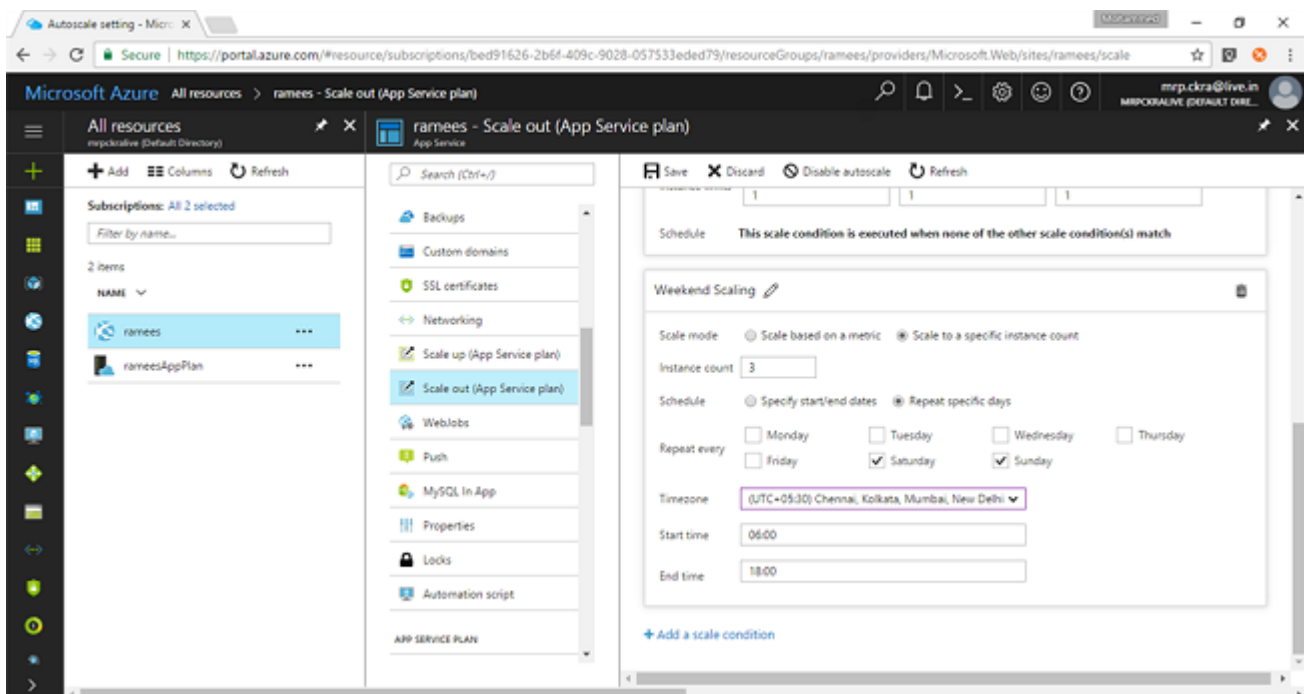
Let us define a new rule to increase the instance count to three in case when CPU usage exceeds 80% for an average of 10 minutes.

For that, I am selecting the metric name to CPU Percentage with the operator greater than and the threshold to 80 and the duration for 10 minutes.

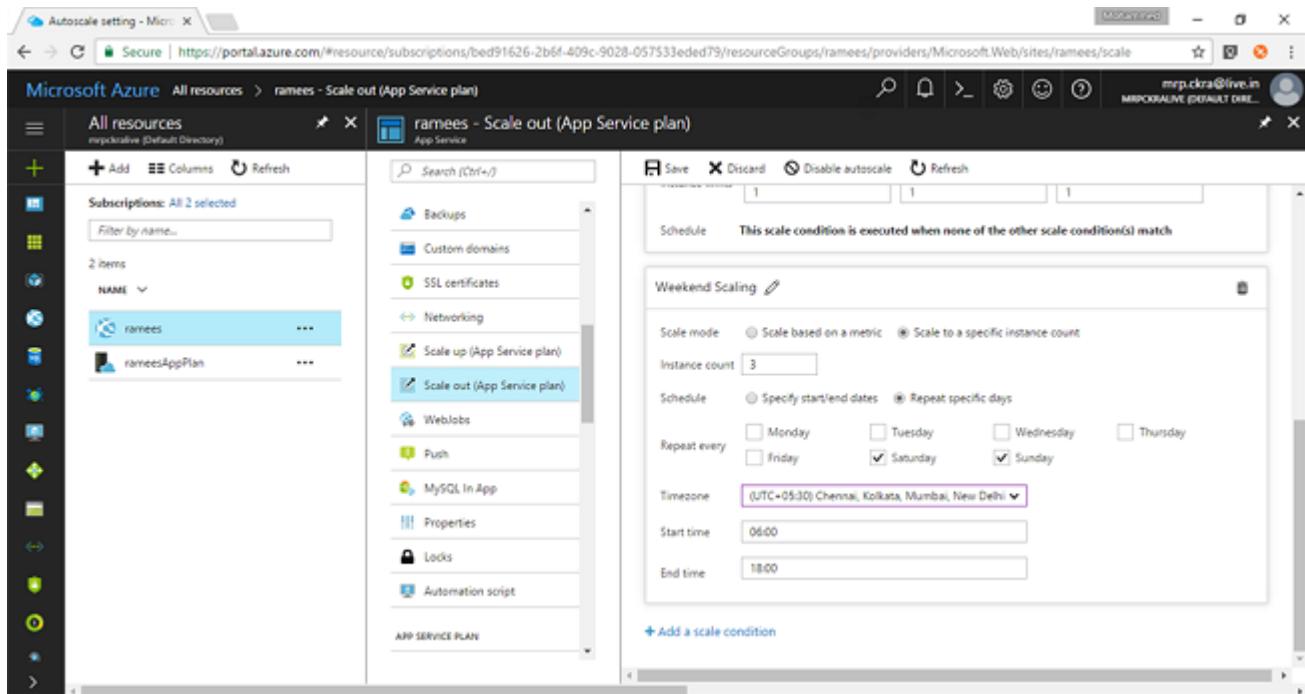
Here, the duration of 10 minutes means that every time the auto scale runs, it will query the metrics for that past 10 minutes which will allow the metrics to stabilize and avoid reaching transient spikes. Now, we have to increase the count by 3 and instance value to 3 and the cool down time to 5 minutes. Cool down time is the waiting time after a scale operation, before the scaling starts again.



This default scale will always happen irrespective of time, date, and other parameters, but if you want to add other scale conditions, you have to add them yourself outside of the default setting. For example, if you want to increase the instance count without any condition for all weekends (Saturday and Sunday), you can click on 'Add a Scale Condition' and select 'Scale to a specific instance count' with the start and end time, instead of 'Scale based on a metric'.



It is also possible to have a scale scheduled for a specific date, such as - any festival dates like New Year's Eve etc. You can also define rules to scale with maximum and minimum instances to run.



So, it is possible to have a default scale, recurrent scale, and fixed date scales.

One field I left unexplained was Time Aggregation on defining a rule which I left average. Think of a condition where you are having three instances running at a particular time. Then, it will not scale if one of the instances goes beyond the threshold. The scaling will happen only when the average of the three instances goes beyond the threshold.

You have to plan your rules very nicely in case of the production. The values I have given in the sample (like duration 10 minutes and cool of time 10 minutes) are for demo purpose only. In production environments, it may not be acceptable according to the organization rules.

Thank you for using C# Corner