Azure GAHP Server for HT Condor

Authors:

Rangarajan Srirangam (Microsoft)

Rakesh Patil (Microsoft)

Reviewers:

Jamie Frey (University of Wisconsin)

Brij Jashal (Tata Institute of Fundamental Sciences)

Tony Wu (Microsoft)

Xavier Pillons (Microsoft)

Suren Machiraju (Microsoft)

# Overview

The University of Wisconsin and the Microsoft Azure CAT team have collaborated to develop an Azure GAHP specification and an Azure GAHP Server. Together they enable the HT Condor framework to leverage the Azure Cloud for running high-throughput computing (HTC) workloads using Azure Virtual Machines and Virtual Machine Scale Sets. The Azure GAHP specification lays out simple commands using which compute resources can be created on Azure utilizing features of Azure Infrastructure as a Service (IAAS) such as Virtual Machines, Virtual Networks, Disks etc. There are also command options by which Azure PAAS services such as Azure Automation and Azure Key Vault can be invoked for specific purposes. The Azure GAHP Server is written in Python and is open-sourced and available for download and use.

# HT Condor

[HT Condor](https://research.cs.wisc.edu/htcondor/description.html) is an open-source [HTC](https://research.cs.wisc.edu/htcondor/htc.html) (high-throughput computing) software framework. It is developed by the HTCondor team at the Computer Sciences Department at the University of Wisconsin-Madison. The [HT Condor software](https://research.cs.wisc.edu/htcondor/description.html) provides a job queueing mechanism, scheduling mechanism, policy based control, priority scheme, resource monitoring, and resource management. Jobs submitted to HTCondor are placed into a queue, scheduled for execution at a time and location controlled by a policy, monitored for progress, and their completion reported to the user. You can find the above and more information [here](https://research.cs.wisc.edu/htcondor/description.html).

# GAHP

The compute resources leveraged by HT Condor in executing jobs could be on an on-premise compute grid or on public cloud. HT Condor requires that a [GAHP](https://research.cs.wisc.edu/htcondor/gahp/) (Grid ASCII Helper Protocol) server process be implemented to integrate with compute grids or clouds. The GAHP server process which runs as a separate process, is implemented as multi-threaded server facilitating non-blocking calls. HT Condor issues commands to the GAHP server process via standard input (STDIN) of the process and accepts responses via the standard output (STDOUT) of the process. The GAHP commands use a simple ASCII based command syntax and conform to a GAHP protocol that is designed to communicate with a particular job submission/execution system. There are therefore several GAHP protocols for interfacing with several grid/cloud systems.

# Azure GAHP Protocol

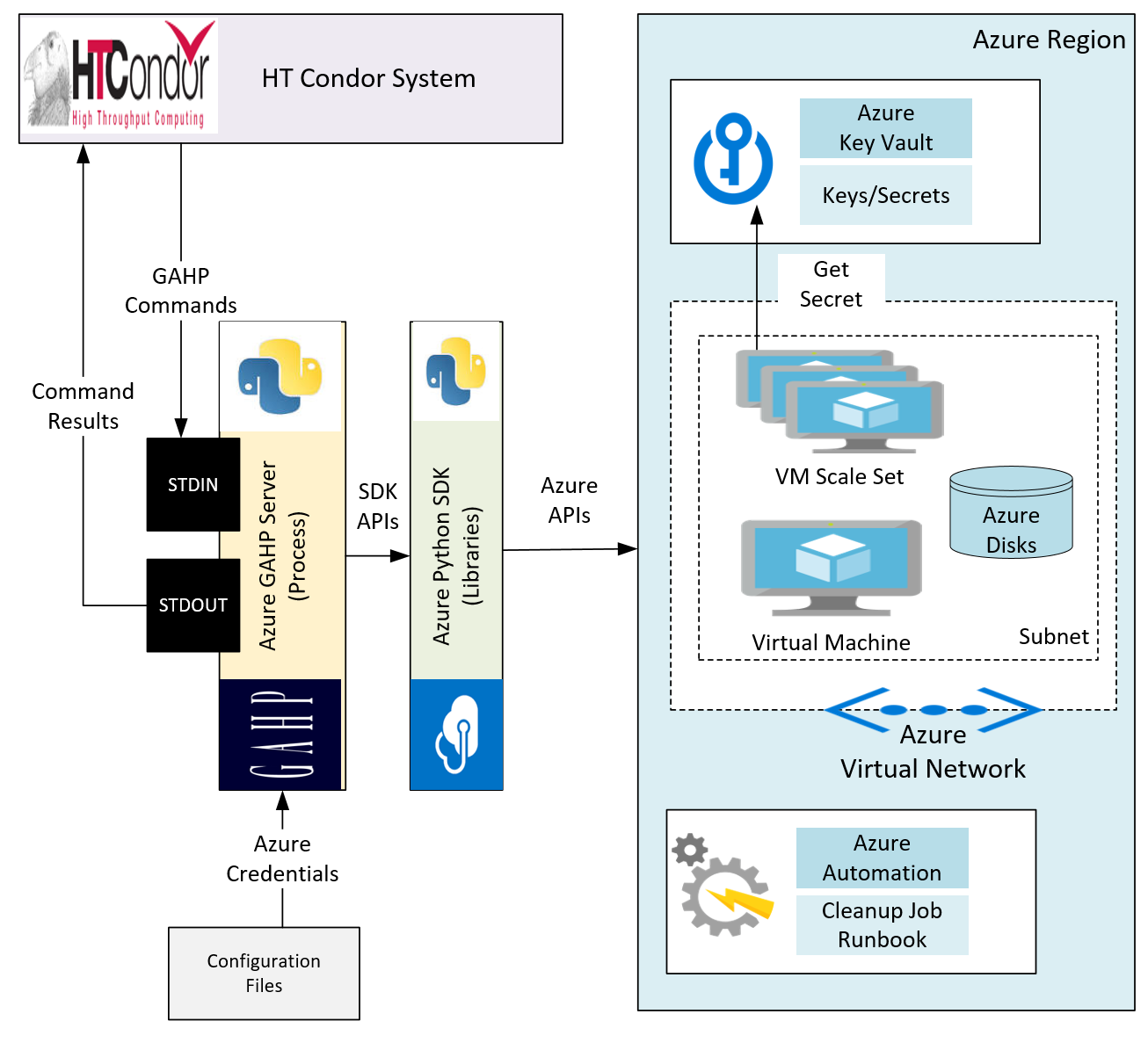
One of the GAHP protocols is the [Azure GAHP](https://research.cs.wisc.edu/htcondor/gahp/) protocol that is designed for the Azure Cloud. The Azure GAHP protocol specifies the commands that an Azure GAHP Server must implement. It firstly allows for individual Virtual Machines (VM) to be created, deleted, listed on Azure via the AZURE\_VM\_\* commands. To allow for fast creation of hundreds of Virtual Machines should the need arise, the Azure GAHP protocol provides AZURE\_VMSS\_\* commands that can leverage [Azure Virtual Machine Scale Sets](https://azure.microsoft.com/en-us/services/virtual-machine-scale-sets/) (VMSS) to create, delete, scale-up or scale-down VMs in large numbers. There are options provided with Azure GAHP VM\* and VMSS\* commands to allow certain Azure specific options such as Virtual Networks, data disks, IP addresses etc., to be specified. The Azure GAHP protocol also allows the user to request for an automatic clean up action that leverages [Azure Automation](https://docs.microsoft.com/en-us/azure/automation/automation-intro) to delete running VMs or VMSS at a user specified future time. Finally, the Azure GAHP Protocol provides a way for user secrets to be securely installed into VMSS during creation so that such secrets can be used to securely access external resources from within the VMs. This capability uses the [Managed Service Identity (MSI) feature with VMSS](https://github.com/simongdavies/azuremsi/blob/master/windowsvmssmsi/README.md) and the [Azure Key Vault](https://azure.microsoft.com/en-us/services/key-vault/) service.

# Azure GAHP Server

An Azure GAHP Server is an implementation of the Azure GAHP Protocol. Since the GAHP does not mandate a particular programming language, platform or technology, there can be multiple Azure GAHP implementations which conform to the Azure GAHP protocol but differ in implementation choices. Currently. there is a freely downloadable implementation of an Azure GAHP Sever, tested by the University of Wisconsin-Madison to work with HT Condor that can be downloaded from [here](https://htcondor-wiki.cs.wisc.edu/index.cgi/dir?d=src/azure_gahp) . It is written in Python and runs as a separate process with a command line. It needs to be configured with a valid Azure subscription and credentials that are provided in a text file. It works as follows

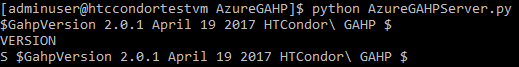
* HT Condor system sends GAHP commands such as “AZURE\_VM\_CREATE” to the STDIN of the Azure GAHP Server with the requisite parameters.
* The Azure GAHP servers uses a configuration file to read the Azure subscription and credential information
* The Azure GAHP Server then translates the GAHP commands into Azure API calls and command parameters to Azure API arguments using the Azure python SDK libraries.
* These Azure API calls connect over the internet to the Azure cloud (in a chosen Azure region) and cause VM/VMSS getting created, deleted etc., depending on the command executed.
* While creating VM/VMSS, the dependent Azure resources needed for the running VM/VMSS such as virtual networks, subnets, NICs, IP addresses etc are also created using the specified parameters
* The Azure GAHP server reports the results of each command execution on the STDOUT in response to the GAHP “RESULTS”
* The HT Condor system receives the results from the STDOUT of the Azure GAHP server

This workflow is illustrated in the figure below



The Azure GAHP server can optionally use the Azure Key Vault to store secrets that need to be used by VMs to access resources while running. The VMs authenticate to the Azure Key vault securely by using a combination of the specified command parameters and the Managed service identity feature supported by VMSS. If an automatic resource deletion job is specified, then the Azure GAHP servers creates a run book in Azure automation that is invoked upon a user specified schedule to delete VM/VMSS created by the user.

While the HT Condor system interacts via STDIN/STDOUT in an automated fashion, the GAHP server can also be manually tested via the command line. In the scree shot below the command line is shown with the result of the execution of the Azure GAHP “Version” command



# Installing and using the Azure GAHP Server

To install and use the Azure GAHP server, you need to

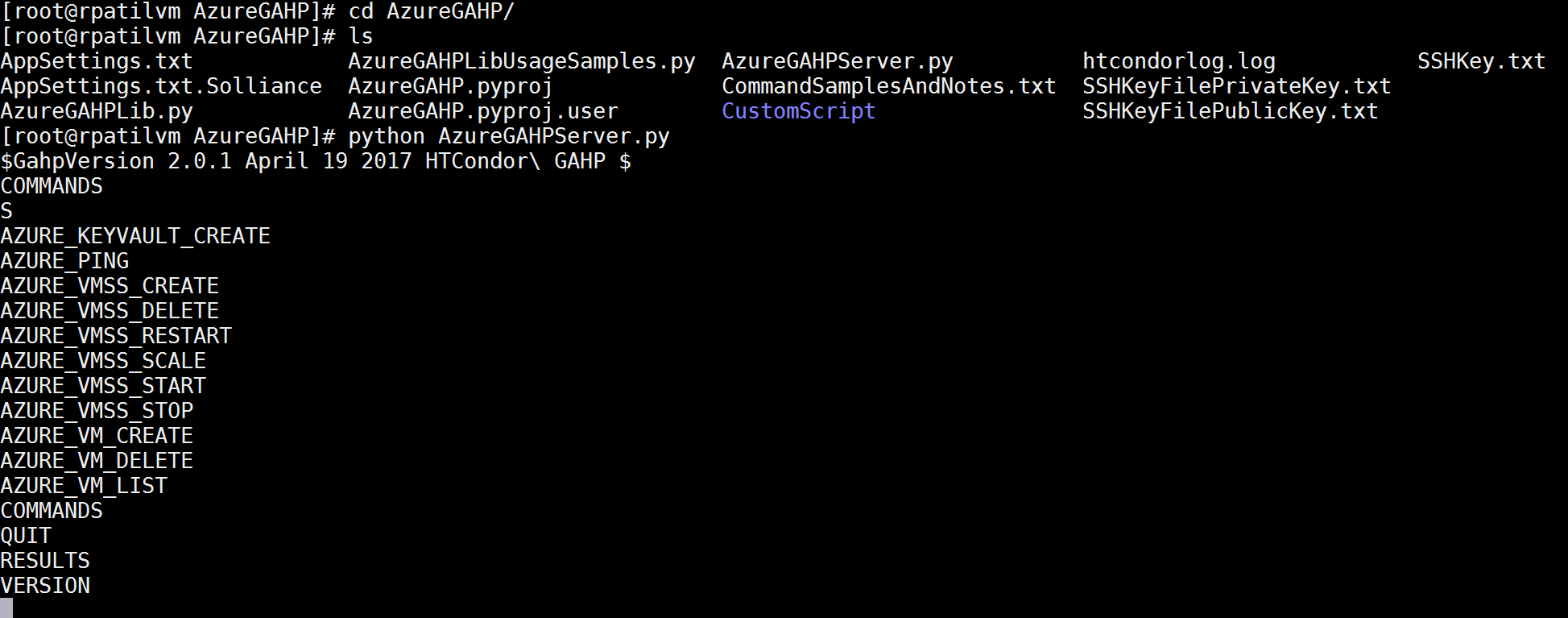
* Install Prerequisites including Python 3.5, PIP, Azure SDK for Python
* Create a Service Principal for the Azure GAHP Server and Key Vault
* Optionally complete the pre-requisites for accessing Azure Key Vault
* Optionally complete the prerequisites for creating the automatic deletion job
* Enter Azure credentials in the Azure GAHP Server configuration file
* Download the Azure GAHP server package
* Run the GAHP Server
* Execute Azure GAHP commands

# 

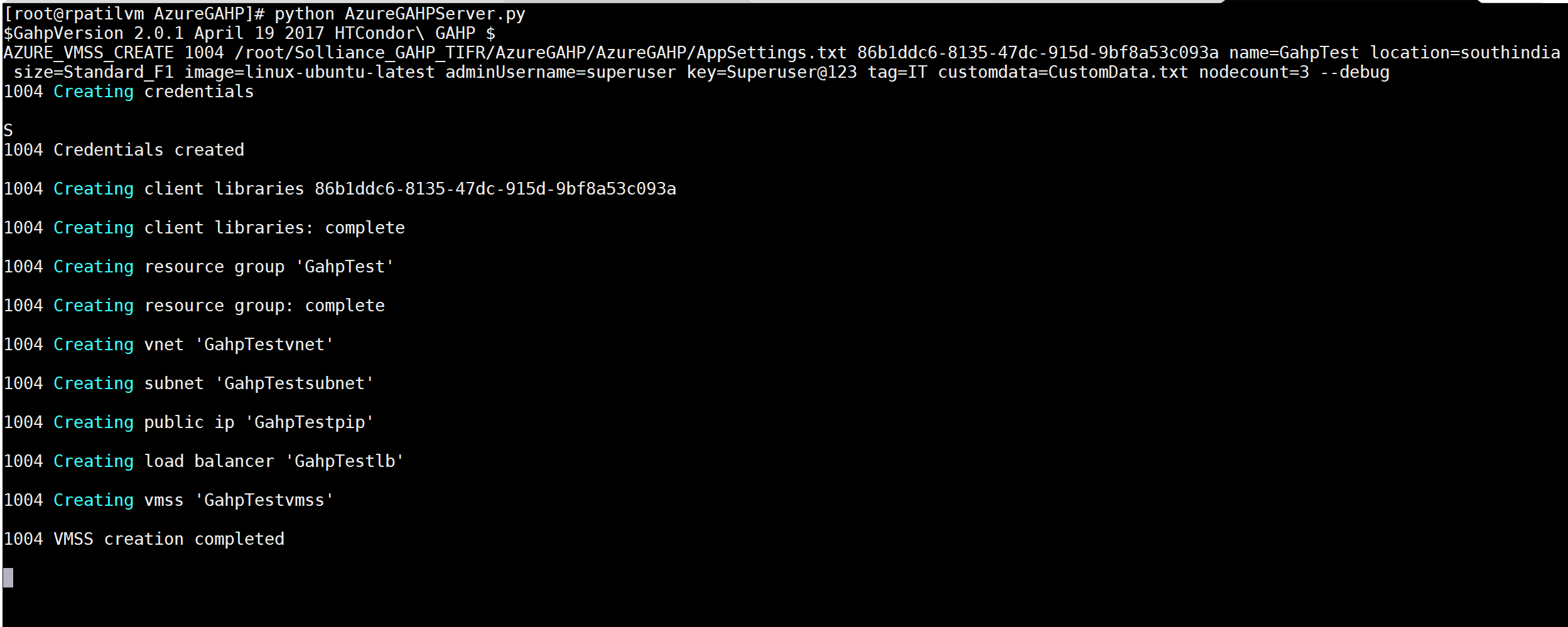
# Example Usage of the Azure GAHP Server

* The following Screen Shot shows list of available Azure GAHP commands.

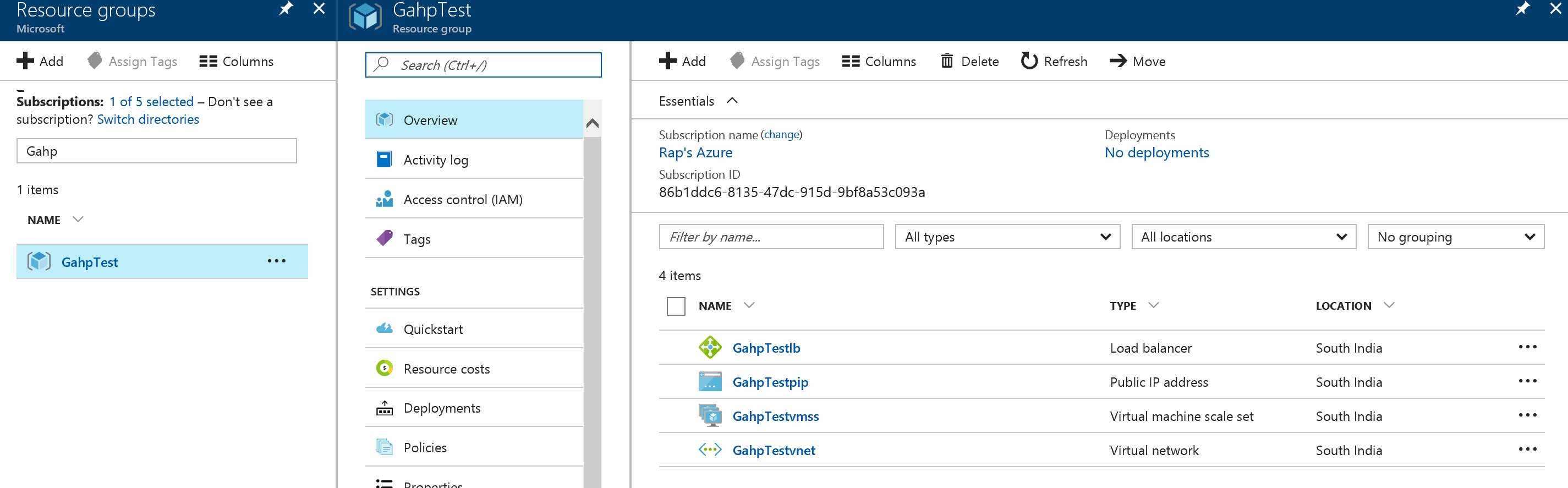
Invoke the python executable from the command prompt using command “python AzureGAHPServer.py” . To see list of available commands Type “COMMANDS” at STDIN.

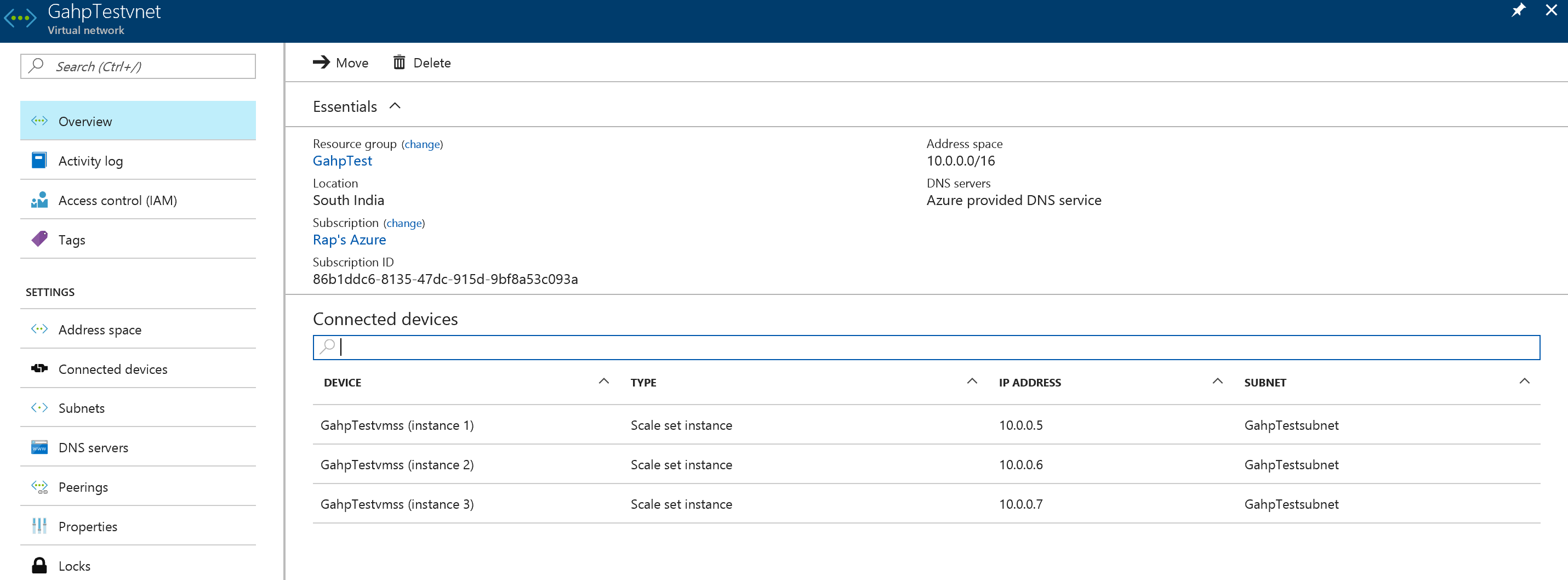


* The following screen shot shows command sample to create VM Scale Sets. Notice, that the scale set named GahpTestvmss with 3 instances is created within GahpTest Resource Group and located in South India Region. The size of VM instances within scale set is Standard\_F1 with Linux ubuntu OS installed. The username and password to login to these instances is also mentioned within the command. A sample customdata script that is to be executed at first boot of VM instances is also added as parameter to VMSS CREATE command.

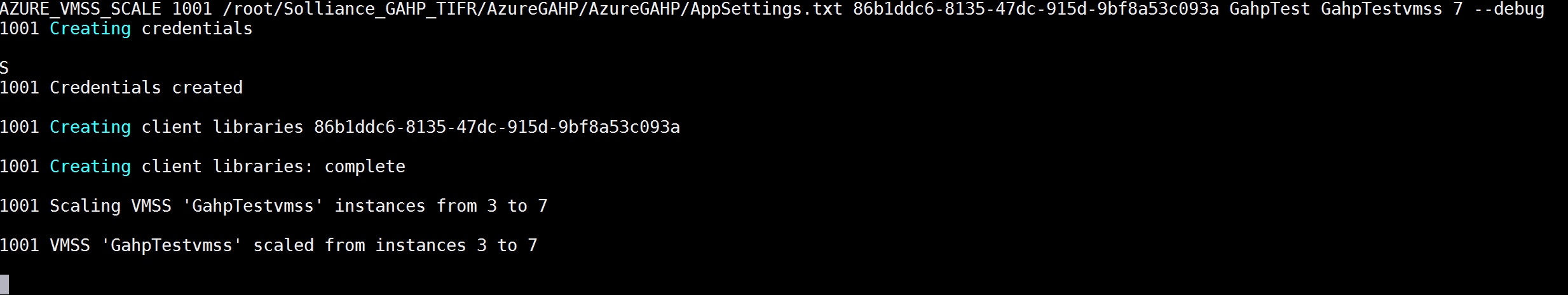


* The below two screen shots from Azure Portal also confirms that the GAHP Protocol has created a VMs Scale Set named GahpTestVMSS within GahpTest Resource Group.

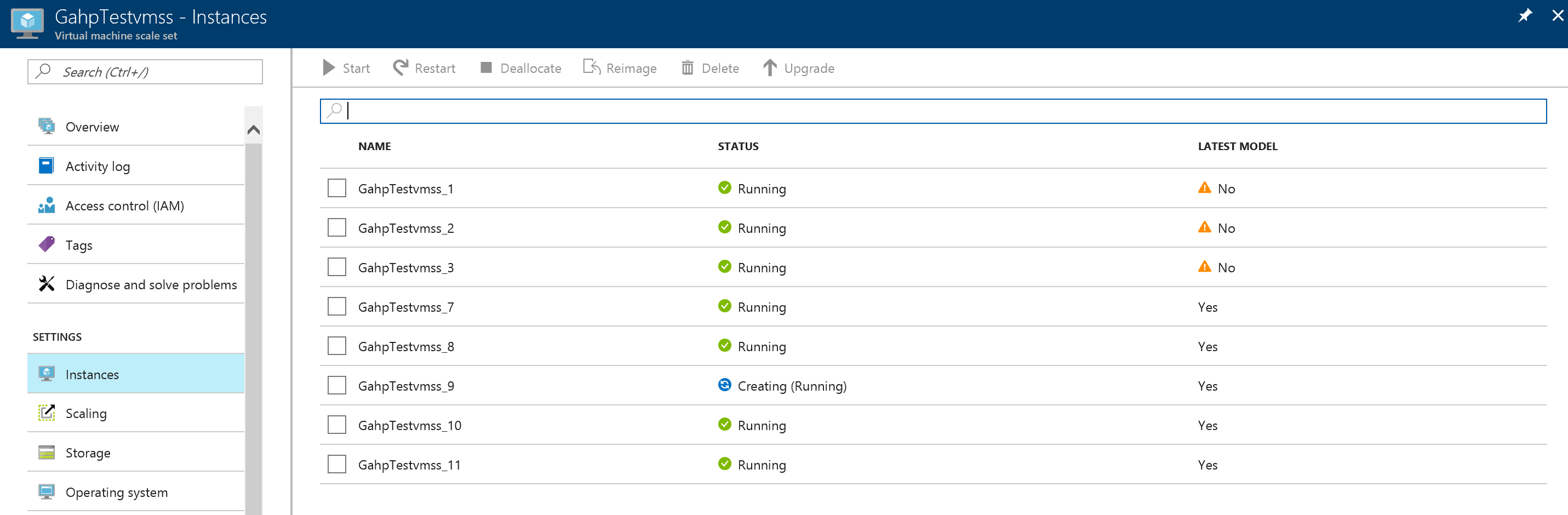




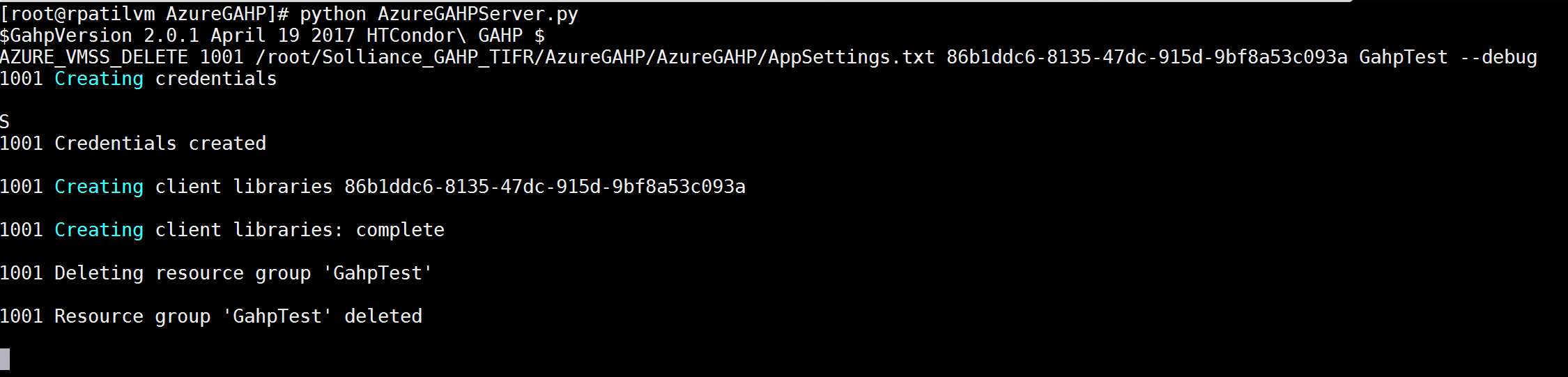
* Let us try scaling the VMSS from 3 to 7 instances. The screenshot below shows the scaling feature.



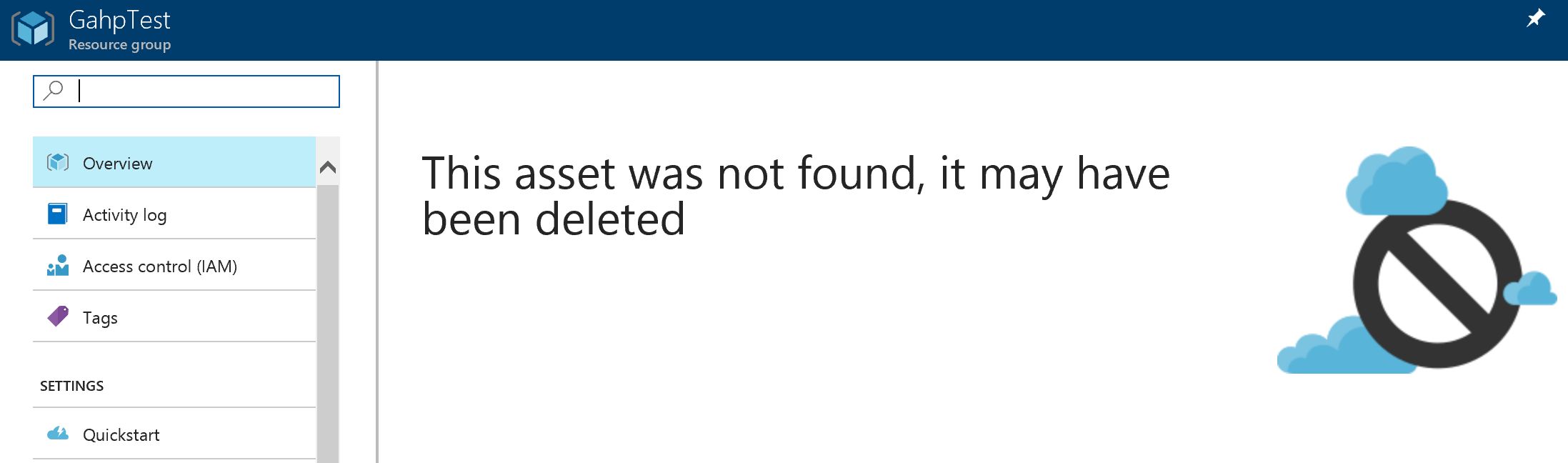
* The portal shows Instance count growing from 3 to 8. The reason for 1 extra VM is Over Provisioning feature of VMSS. At the end of the command completion Azure Portal would show only 7 running VMs as desired.



* To complete the loop, let’s try to delete the VMSS along with the Resource Group and other related artifacts within the resource group. The following screen shot shows Deletion Operation being accomplished via GAHP command.



* The same can also be confirmed by looking at the Azure Portal.



The detailed instructions for completing the above steps successfully are provided in the “[Pre-Requisites and Installation of GAHP interface](https://github.com/Azure-Samples/azure-sdk-python-gahp-grid-ascii-helper-protocol/blob/master/Pre-Requisites%20and%20installation%20of%20GAHP%20interface.docx)” and “[Azure GAHP Commands documents](https://github.com/Azure-Samples/azure-sdk-python-gahp-grid-ascii-helper-protocol/blob/master/Azure%20GAHP%20Commands.docx)”. You can down also load the Azure GAHP Server package from [here](https://github.com/Azure-Samples/azure-sdk-python-gahp-grid-ascii-helper-protocol/blob/master/Azure%20GAHP%20Commands.docx).