How to use AutoElastic Manager

This document describes how to use AutoElastic Manager in an OpenNebula Cloud. This document is divided into two sections: setting UI parameters and application communication protocol.

User Interface Parameters

In this section, you will learn about all parameters that you can set in AutoElastic Manager. These parameters can be provided in the User Interface and then exported to a config.xml file. In the following each parameter is described in details. The values in parenthesis represent the configuration in the XML file.

Server

- FrondEnd (<FRONTEND_ADDRESS>):
  Cloud FrontEnd address.

- User (<FRONTEND_USER>):
  Cloud administrator user.

- Password (<FRONTEND_PWD>):
  Cloud administrator user password.

- Cluster ID (<CLUSTER_ID>):
  Cluster ID parameter to create new hosts in the cloud.

- Image Manager (<IM>):
  Cloud image manager driver to create new hosts in the cloud.

- Virtual Machine Manager (<VNM>):
  Cloud virtual machine manager driver to create new hosts in the cloud.

- Virtual Network Manager (<VMM>):
  Cloud network machine manager driver to create new hosts in the cloud.

Parameters

- SLA (<SLA>):
  XML file with SLA parameters.

- Log Path (<LOG_PATH>):
  Directory where AutoElastic Manager saves logs.

- Execution Log (<EXEC_LOG>):
  Word that AutoElastic Manager uses in the execution log name.

- VM Template ID (<TEMPLATE_ID>):
  Template ID for instantiating new virtual machines.

- Monitoring Interval (<MON_INTERVAL>):
  Time in seconds between each operation/observation (synchronization and elasticity verification).

- Virtual Machines (<NUM_VMS>):
  Number of virtual machines to add or remove in each elasticity operation.

- Upper Threshold (<UPPER_THRESHOLD>):
  Threshold to add virtual machines (0 to 100).

- Lower Threshold (<LOWER_THRESHOLD>):
  Threshold to remove virtual machines (0 to 100).

- Monitoring Window (<MON_WINDOW>):
  Number of observations. This value defines how many of the last observations must be considered by the evaluation algorithm.

- Cool-down Observations (<COOL_DOWN>):
  Number of monitoring observations after an elasticity action that AutoElastic Manager cannot do any new elasticity operation.

- Threshold Type (<THRESHOLD_TYPE>):
  The values in the XML file can be “static” or “live”.
  - Static: The upper and lower thresholds are the same in the entire execution.
  - Live: The upper and lower thresholds are automatically calculated in each observation.

- Evaluation Algorithm (<EVALUATION_ALGORITHM>):
  The values in the XML file can be “window_aging”, “full_aging” or “generic”.
  - Window Aging:
    AutoElastic Manager calculates the load value considering the Monitoring Window parameter comparing this value with the thresholds.
  - Full Aging:
    AutoElastic Manager calculates the load value considering all observations (do not use Monitoring Window) comparing this value with the thresholds.
  - Generic:
    AutoElastic Manager uses the current load. Here, Monitoring Window defines the consecutive times that a threshold must be violated to a resource reorganization occurs.

- General
  - Laboratory Mode (<LAB_MODE>):
    Activate automation tests programmed directly in the code. (Do not use)
  - Read Only (<READ_ONLY>):
    Simulation mode where AutoElastic Manager does not add/remove resources in the cloud. Data is only read from the cloud and elasticity operations occur only locally and logically. This mode operates only when Manage Hosts is active.
  - Manage Hosts (<MANAGE_HOSTS>):
    Activate host monitoring. Elasticity operations add and remove hosts and its virtual machines from the cloud. To monitor only virtual machines uncheck this box.

Communication

- Data Server (<DATA_SERVER_ADDRESS>):
  Server address to access a shared memory area (NFS).

- SSH User (<DATA_SERVER_PORT>):
  User to access the (NFS) server.

- SSH Password (<DATA_SERVER_USER>):
  User password to access the (NFS) server.

- Message Source Dir <DIR_MSG_SOURCE>):
  Data Server directory where AutoElastic Manager reads message files. (must end with "/")

- Message Target Dir (<DIR_MSG_TARGET>):
  Data Server directory where AutoElastic Manager creates message files. (must end with "/")

- Local Temp Dir (<DIR_MSG_LOCAL>):
  The Local directory where AutoElastic Manager creates message files to send to Data Server. (must end with "/")

- Warning Remove Resources Message (<MSG_WARNING_REMOVE>):
  Name of the file that AutoElastic Manager creates inside Message Target Dir to inform that he will remove resources.

- Permission to Remove Resources Message (<MSG_PERMISSION_REMOVE>):
  Name of the file that AutoElastic Manager reads from Message Source Dir to get permission to remove resources.

- Notification of New Resources Message (<MSG_NOTIFICATION_NEW_RESOURCES>):
  Name of the file that AutoElastic Manager creates inside Message Target Dir to inform that new resources are online.

Hosts

- List of host addresses that AutoElastic Manager can use in the cloud. In the config.xml file:
  <HOSTS>
  <HOST>HOST_NAME_OR_IP-1</HOST>
  <HOST>HOST_NAME_OR_IP-2</HOST>
  …
  <HOST>HOST_NAME_OR_IP-n</HOST>
  </HOSTS>

How Elasticity Works

The current version of AutoElastic Manager's features and limitations:

- AutoElastic adds and removes Hosts or Virtual Machines from the cloud environment;
- The elasticity grain is:
- When managing Hosts (flag Manage Hosts = true): ONE host and x virtual machines in this host, where x is defined by the "Virtual Machines" parameter;
- When managing VMs (flag Manage Hosts = false): x virtual machines, where x is defined by the "Virtual Machines" parameter. Here, the OpenNebula scheduler will choose the suitable host.
- AutoElastic Manager considers in the SLA the maximum and minimum number of HOSTS of VMs in the cloud;
- The initial resources in the cloud environment when the monitor starts must be the minimum number of resources defined in the SLA (hosts or virtual machines).

Removing Resources

When AutoElastic Manager detects that the load is low, he starts the operation to remove the last x resources (added in a later operation), where x is the "Virtual Machines" parameter). So the Manager creates in the "Message Target Dir" a message (file with the name configured in the "Warning Remove Resources Message" parameter) to the application to warn that the resources will be removed. After that, the Manager will only remove the resources when it receives a message from the application. The Manager waits for this message (file with the name configured in the "Permission to Remove Resources Message" parameter) from the application, looking for it in the "Message Source Dir". When the application creates the message, the Manager removes the resources.

Adding Resources

When AutoElastic Manager detects that the load is high, it starts the operation to add x resources (where x is the "Virtual Machines" parameter). After this, it returns to the monitoring cycle and in each observation, he checks if the new virtual machines are online. Once the resources are online, the Manager creates in the "Message Target Dir" a message (file with the name configured in the "Notification of New Resources Message" parameter) to inform that there are new resources available.

How To Develop the Elasticity in the Application

Here, important notes are listed that the programmer must have in mind to manually write the application code:

- The number of virtual machines that the AutoElastic Manager adds or removes is defined in a parameter. So, the application must know this value previously. We can send this as a parameter to the application;
- When AutoElastic Manager removes resources, he always removes the last resources he added. In this way, we need to disconnect always the younger processes.
- AutoElastic Manager only starts virtual machines. Then, in the template of the virtual machine the application must be configured to start when the virtual machine SO starts;
- The application code must have areas where processes can be added or disconnected;
- The application needs to know previously the names of messages that can be read or write.

Programming

In the application code, the programmer needs to add the elasticity code in a region where the number of processes can change. Therefore, he can create a method/function of elasticity and call it in these regions every time the processes can be added or removed. This method/function must implement two main operations:

1. Verification for messages: here, the code has to access a remote data directory looking for files from the AutoElastic Manager with specific names. The access to this directory can be implemented using NFS (the SO where the application is running can map the remote directory), SSH or another approach the programmer prefers;
2. Reorganization of resources: when there is a file in the remote data directory, depending on the file name, processes have to be disconnected or connected. So the code here has to reorganize the processes connections and the array/list of available connections. In the case where processes are disconnected, the younger processes have to be disconnected. So, the programmer needs to know the order as the connections occurred. After the disconnections, a code has to create a file with a specific name in the remote directory to allow the AutoElastic Manager to remove the resources. Finally, after both operations, a code must remove the file from the remote data directory.

As we saw, AutoElastic Manager works with three messages to synchronize information with the application. These messages are files with its names defined in the AutoElastic Manager UI. Here, we will see how the application must deal with it and what files can be created or read:

Messages that application can read

- "Warning Remove Resources Message": when the application reads a file with this name, processes have to be disconnected. After reading it, the application must remove the file;
- "Notification of New Resources Message": when the application reads a file with this name, new virtual machines are available and the application can use them and its processes. After reading it, the application must remove the file.

Messages that application can write

- "Permission to Remove Resources Message": after disconnecting resources, the application must write a file with this name in the remote data directory.