Power Management: Setup up custom power scripts for ProfitBricks

Custom Power Scripts for controlling ProfitBricks Instances

Prerequisites

To use dynamic scaling of clusters on ProfitBricks' IaaS platform you have to install a head node in a ProfitBricks virtual data center.

You can register as trial user (https://www.profitbricks.co.uk/trial) or directly sign up (https://www.profitbricks.co.uk/signup). Registration is also possible on ProfitBricks' German (https://www.profitbricks.de/) and US site (https://www.profitbricks.com/).

Your login data is also used for API access.

For setting up your virtual environment you could refer to the online help https://www.profitbricks.com/help/Main_Page. The setup consists of the following steps:

- Request a Bright Cluster Manager installation image from ProfitBricks' Support.
- ProfitBricks will provide this ISO image in your desired region (de/Frankfurt, de/Karlsruhe, us/LasVegas). Keep in mind that your virtual data center must be in the same region as the uploaded ISO image.
- Create a new virtual data center.
- Create a virtual server with following properties
    - 4 cores, 4GB RAM,
    - 60GB HDD without assigned disk image,
    - CDROM as boot device with Bright's ISO image assigned,
    - First NIC created without any connection (this will be internalnet),
    - Second NIC connected to Internet Access (this will be externalnet).
- After provisioning the virtual machine in ProfitBricks, open the server's remote console.
- Install Bright Cluster Manager according to Bright's Administrator manual. Choose a complex password - you are connected to the internet!

The following package should be installed on the Bright head:

Power Management: Setup up custom power scripts for ProfitBricks

1. Install cm-profitbricks package which will install

yum install cm-profitbricks

2. install profitbricks python module
   # pip install profitbricks

Notes

- The cm-profitbricks-1.0-1.noarch.rpm needs to be downloaded from special repository as it’s not publicly available in Bright repositories.
- The configuration files under /cm/local/apps/cm-profitbricks/1.0/etc needs to be adjusted to point to the current ProfitBricks data center, the login key which should be present in a file that is pointed to by PB_LOGIN_CONF directive, the number of CPUs and amount of memory in GB, and the LANID of the boot interface and the storage capacity in GB:

[root@bcnh-72-01 1.0]# cat etc/pb-powerscript.conf

config file for setting up BC/PB cluster

change settings to your own need

PB_LOGIN_CONF=/cm/local/apps/cm-profitbricks/1.0/etc/MyLogin.conf

PB_DCID=909c5078-057c-4189-8f24-578a70f22b6a

PB_LANID=2

PB_CREATE_CPU=2

PB_CREATE_RAM=2

PB_CREATE_STORAGE=10

[root@bcnh-72-01 1.0]#

- The login file pointed to by PB_LOGIN_CONF can be created by
  [root@bcnh-72-01 ~]# cd /cm/local/apps/cm-profitbricks/1.0/bin
  [root@bcnh-72-01 bin]# python -u ./pb_saveLoginFile.py -u <user> -p <password> -L <loginfile>

for example:
[root@bcnh-72-01 bin]# python -u ./pb_saveLoginFile.py -u admin@acme.com -p MySecret -L /cm/local/apps/cm-profitbricks/1.0/etc/MyLogin.conf

- Any power operation should not take more than 120 seconds (2 minutes). This assumption can been relaxed for the power ON/OFF by running the power ON/OFF operations in a screen session and detach from it once the commands are done.

Custom power script

Platform

The script is written in Python and tested on RHEL7 with Python version 2.7.5. The script accepts up to three parameters: <action> <node> ["volatile"]

Power operations

The power operation is either ON or OFF and any other value will be considered as UNKNOWN.

Setting up custom power

Custom power can be setup using CMSH as follows:

[root@profitbricks-test custompowerscripts]# cmsh

[profitbricks-test]% device use node001

[profitbricks-test->device[node001]]% set powercontrol custom

[profitbricks-test->device*[node001*]]% set custompowerscript /cm/local/apps/cm-profitbricks/1.0/bin/pbpowerscript

[profitbricks-test->device*[node001*]]% commit

[profitbricks-test->device[node001]]%

For disposable nodes, the “custompowerscriptargument” should be set to “volatile” so that the

scripts will destroy the node completely from ProfitBricks:

[profitbricks-test->device[node001]]% set custompowerscriptargument volatile

[profitbricks-test->device*[node001*]]% commit

[profitbricks-test->device[node001]]%

Notes

- The node objects should be defined in Bright BCM before they can be created dynamically in ProfitBricks using the power on/off operations.
- Because new nodes boot via PXE, their MAC address has to be configured in BCM to be accepted. As a virtual server’s MAC address is generated on creation, the custom power script must read out the MAC address and configure the node while the server is in a PXE boot loop.
- The cm-scale-cluster script runs periodically, as a cron job, every number of minutes. It checks the available jobs in the queues and powers on the required number of nodes to fulfill the job requirement. The following example shows a crontab setting to run the cm-scale-cluster script once every 12 minutes:

*/12 * * * * /cm/local/apps/cm-scale-cluster/bin/cm-scale-cluster

- The cm-scale-cluster uses /cm/local/apps/cm-scale-cluster/etc/default.conf as the default configuration file. A sample configuration file can be as follows:

QUEUE=defq NODEGROUP=profitgroup WHOLE_TIME=60
STOPPING_ALLOWANCE_PERIOD=10

QUEUE=hiprioq NODEGROUP=profitgroup WHOLE_TIME=60
STOPPING_ALLOWANCE_PERIOD=10

DEBUG = YES

WORKLOAD_MANAGER = slurm

POLICY_MODULE = /cm/local/apps/cm-scale-cluster/lib/default-policy.py

ACTION_MODULE = /cm/local/apps/cm-scale-cluster/lib/default-action.py

PARALLEL_REQUESTS = 1

LOG_FILE = /var/log/cm-scale-cluster.log

LOCK_FILE = /var/lock/subsys/cm-scale-cluster

SPOOL_DIR = /var/spool/cmd/cm-scale-cluster

RUN_INTERVAL = 15

- WHOLE_TIME can be used to keep idle nodes up and running the specified time. This was invented for environments where billing is based on an hourly schedule like AWS. In these environments, destroying a node too early would waste time that was already paid for. As ProfitBricks' billing is minute-base, nodes can be destroyed as soon as they are idle by setting WHOLE_TIME=0. On the other hand, creating a node from scratch again takes some time. Thus it may be more performant to keep nodes running some minutes if it's likely that they are needed by new jobs again.

- For more details on how to configure cm-scale-cluster, please refer to Bright Cluster Manager Administrator's manual.

Unique solution ID: #1315
Author: mohamed adel
Last update: 2016-07-25 16:51