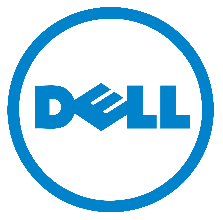
**Brocade Vepc Automation scripts**

User Guide



**Copyright © 2015 Dell Inc. All rights reserved.** This product is protected by U.S. and international copyright and intellectual property laws. Dell™ and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Table of Contents

[**Revision History: 3**](#_Toc431910414)

[**Table of Acronyms 4**](#_Toc431910415)

[**1 Introduction 5**](#_Toc431910416)

[**2 Pre-requisite Knowledge 5**](#_Toc431910417)

[**3 System Requirements 5**](#_Toc431910418)

[**4 How to get the scripts 6**](#_Toc431910419)

[**5 How to run 7**](#_Toc431910420)

[**6 Summary of Scripts 8**](#_Toc431910421)

[**7 Execution time of scripts 9**](#_Toc431910422)

[**8 Login Credentials 9**](#_Toc431910423)

[**9 Logs 10**](#_Toc431910424)

[**10 Known bugs 10**](#_Toc431910425)

# Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Description** |
| 1.0 | Sep 14, 2015 | Abdul Rehman | Initial Version |
| 1.1 | Sep 30, 2015 | Abdul Rehman | Updated Version 1 |

# Table of Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| CPE | Control Plane Engine |
| CDF | Charging Data Function |
| CLI | Command Line Interface |
| CIDR | Classless Inter-domain Routing |
| DPE | Data Path Engine |
| DHCP | Dynamic Host Control Protocol |
| EPC | Evolved Packet Core |
| EMS | Element Management System |
| GUI | Graphical User Interface |
| HTTP | Hyper Text Transfer Protocol |
| IP | Internet Protocol |
| IXIA | A company that provides tools for Network Testing |
| PDN | Public Data Network |
| RAN | Radio Access Network |
| RIF | RAN Interface |
| SDB | Session Database |
| SSH | Secure Shell |
| UDB | User Database |
| URL | Uniform Resource Locator |
| VEPC | Virtual Evolved Packet Core |
| VCM | Virtual Core for Mobile, a Product by Brocade. |
| VEM | VCM Element Management |

# Introduction

VEPC Automation scripts are developed to fully automate the Brocade VCM deployment process in high availability. User doesn’t need to perform any configuration except editing some required parameters that are required for running the scripts and vary according to the Openstack deployment in use. Once the user enters those required parameters the deployment process is fully automated. There is no need to alter any parameter later at any stage of vEPC deployment. The document is intended for anyone deploying Brocade vEPC using Dell Automation scripts and will explain the environment and resources required for the scripts to run.

# Pre-requisite Knowledge

To be able to use this guide effectively, users need to have a general understanding of operating systems based on Linux kernel and Openstack. Knowledge of following will be helpful but not required:

* Linux and Openstack command line tools
* Basic knowledge of Python would be helpful but not required

# System Requirements

Two compute nodes are required for running these scripts. Scripts deploy fifteen instances during initial deployment process while scale-up consists of three instances each time it executes.

Scripts are tested in the following environment and it is recommended to have this setup before running the scripts:

* OpenStack Juno
* CentOS 6.5
* Python 2.6

Along with following hardware resources on each Compute Node:

|  |  |  |  |
| --- | --- | --- | --- |
| **Node** | **Memory** | **Storage** | **Logical CPUs** |
| Compute 1 | 64GB | 640GB | More than 48 |
| Compute 2 | 64GB | 640GB | More than 48 |

EMS instance consists of standard Openstack flavor “m1.large” which has the following specifications:

|  |  |  |  |
| --- | --- | --- | --- |
| **Flavor** | **Memory** | **Storage** | **Logical CPUs** |
| m1.large | 8GB | 80GB | 4 |

All other instances except EMS consists of standard Openstack flavor “m1.medium” which has the following specifications:

|  |  |  |  |
| --- | --- | --- | --- |
| **Flavor** | **Memory** | **Storage** | **Logical CPUs** |
| m1.medium | 4GB | 40GB | 2 |

The script will create availability zones for two compute nodes. Name of zones are ‘Compute 1’ and ‘Compute 2’. There are no specific hardware requirements for Control nodes but they need to have enough resources available to work with compute nodes.

# How to get the scripts

To download scripts, log in to Dell Box account using the URL <http://app.box.com/>, in case if you don’t have Box account please contact your dell representative. After login process is complete, go to the directory NFV/vnf/partner\_brocade/ and download *vEPC.tar.gz* file which contains all the required files for running vEPC scripts and a checksum.txt file containing checksum of *vEPC.tar.gz.*

SSH into the control node and copy the vEPC.tar.gz file in /root/

After copying *vEPC.tar.gz* file on control node, calculate its checksum using *md5sum vEPC.tar.gz* command:

Verify the value printed on CLI with value given in *checksum.txt* file to be sure that tar file is not corrupted.

Now run the command to untar the vEPC.tar.gz:

tar –zxf vEPC.tar.gz vEPC/

After the untar completes change the directory by entering the following command:

cd vEPC

# How to run

To execute scripts, SSH into the CLI of control node and go inside vEPC directory. Before running the scripts, user need to edit the “creds.txt” file provided in vEPC scripts directory based on the required parameters explained below for successful execution of deployment scripts.

Values should be entered in inverted commas as currently entered as a sample in creds.txt file shown in Figure 1, the order and format of parameters should not be changed.

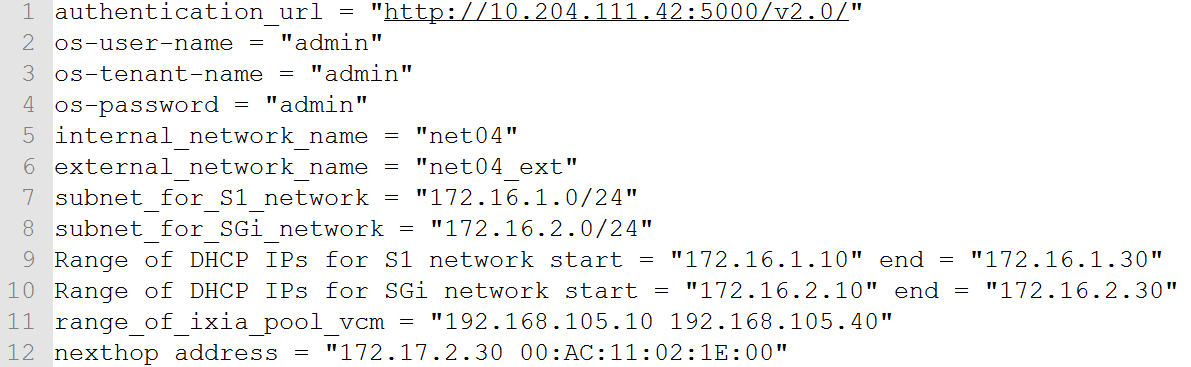


Figure 1: Sample values of creds.txt file

1. authentication\_url = “OS\_AUTH\_URL from openrc/keystonerc\_admin file located at /root/ openrc/keystonerc\_admin in Control Node”
2. os-user-name = “OS\_USERNAME from openrc/keystonerc\_admin file located at /root/ openrc/keystonerc\_admin in Control Node”
3. os-tenant-name = “OS\_TENANT\_NAME from openrc/keystonerc\_admin file located at /root/ openrc/keystonerc\_admin in Control Node”
4. os-password = “OS\_PASSWORD from openrc/keystonerc\_admin file located at /root/ openrc/keystonerc\_admin in Control Node”
5. Private network name is the name of the Private network in Openstack.

Note: This should already exist in Openstack networks and if it isn’t created during deployment, please create these before running the scripts.

1. Public network name is the name of the External network in Openstack.
2. Subnet value for S1 network is the value of subnet to be assigned to S1 network.
3. Subnet value for SGi network is the value of subnet to be assigned to SGi network.
4. Range of DHCP IPs for S1 network is used to assign IPs to ports (s1\_u, s1\_mme, s1\_u2, s1\_mme2 etc.) within this range.
5. Range of DHCP IPs for SGi network to assign IPs to ports (sgi, sgi2 etc.) within this range.

It is required to have any value between 24 and 27 for CIDR in S1 and SGi networks.

1. Range of IP addresses to be allowed for IXIA traffic. Sample value:

range\_of\_ixia\_pool\_vcm = "192.168.105.10 192.168.105.40"

First IP specifies the start of pool and second IP address specifies the end of pool.

1. Nexthop address with MAC Address of outbound traffic to PDN server

nexthop address = "172.17.2.30 00:AC:11:02:1E:00"

After successfully updating the above parameters, following are the commands to run the vEPC scripts.

1. To deploy vEPC, run “python vEPC\_deploy.py”
2. For scale-up, run “python scale\_up.py”
3. For scale-down, run “python scale\_down.py”
4. To terminate vEPC, run “python vEPC\_termination.py”

# Summary of Scripts

There are total of 4 scripts for vEPC deployment process, each script performs a specific operation:

1. **vEPC\_deploy.py**

* This script deploys vEPC VCM components in high availability. Total of 15 instances are deployed in the form of VCM-1 (7 instances), VCM-2 (7 instances) and 1 instance for EMS along with networks S1 and SGi

1. **scale\_up.py**

* This script scales-up VCM components by creating a set of instances SDB, CPE and DPE each time script runs.

1. **scale\_down.py**

* This script terminates the set of single instances SDB, CPE and DPE created during scale-up process

1. **vEPC\_termination.py**

* Terminates the vEPC all components and prompts the user if he wants to delete EMS, aggregate groups and glance images of VCM and performs the action based on the user input

Detailed functionality of each script is explained in the document name “vEPC Automation Scripts Design Reference” along with diagrams for better understanding without going deeper into the inner details of APIs and functions used to perform such tasks.

# Execution time of scripts

Depending upon the computational power, all the four scripts have variant execution time. Approximate time of completion is stated below:

1. Initial deploy scripts takes 30-40 minutes
2. Scale-up script take 7-10 minutes
3. Scale-down script take 2-4 minutes
4. vEPC termination script takes 3-5 minutes

# Login Credentials

To SSH into VCM and EMS instances use the following credentials:

Username: root

Password: root123

To login to the EMS GUI, use the following credentials:

Username: admin

Password: Root1234

# Logs

All the scripts generate logs during execution. To see the logs generated during the execution of scripts, go into the /vEPC/logs/ directory. It contains a total of 8 files, 2 for each script; one file is for activity log i.e. the total flow of script in which it deploys VCM components and the second is error log in case if any error occurs during the execution of script.

1. deploy\_date\_time.log and deploy\_error\_date\_time.log
2. scale\_up\_date\_time.log and scale\_up\_error\_date\_time.log
3. scale\_down\_date\_time.log and scale\_down\_error\_date\_time.log
4. terminate\_date\_time.log and terminate\_error\_date\_time.log

# Known bugs

If you see any of the following bugs in log files or CLI please refer to the resolution/possible reason of the bug to know the actual cause of bug.

|  |  |  |
| --- | --- | --- |
| **Sr. #** | **Bugs** | **Resolution/Possible reason** |
| 1. | Unauthorized: The request you have made requires authentication. (HTTP 401) | This error usually occurs when the authentication URL, OS\_USERNAME, OS\_PASSWORD or OS\_TENANT\_NAME entered in creds.txt doesn’t match with the values of keystonerc\_admin/openrc file. Please double check the values and try running the scripts again. |
| 2. | novaclient.exceptions.NotFound: No more floating ips in pool. (HTTP 404) | This error usually occurs when there are no more floating IPs available in the pool to assign to instances. Please increase the number of floating IP pool and then re-try. |
| 3. | Floating IP assignment error. Retrying...  Error: Connection to neutron failed: HTTPConnectionPool  Request timed out. (timeout=30) (HTTP 400) | This error is generated when keystone is unable to make a connection with neutron or neutron server is not responding. Please make sure your neutron/nova server are active and verify by using following commands:  neutron net-list  nova service-list |
| 4. | BadRequest: No nw\_info cache associated with instance (HTTP 400) | This error occurs because of race when associating floating IP to fixed IP. Originally, the fixed IPs are associated and "virtual\_interface\_id" is updated and saved to the DB at a later time. When floating IP is tried to be allocated before the update in database, it can cause this error. |
| 5. | No valid host was found. | This issue usually occurs because of not enough resources available to deploy the instances. There is a resource check function which gives warning to the user if the resources are not enough and if user still continues deployment this could lead to an error. |
| 6. | EMS status Failed | Initial Deployment script deploys the VCM-EMS and tries to start the EMS service, if you see the FAILED status on CLI try to login using the URL provided as the EMS sometimes show FAILED status while it’s still working. This is probably the issue with EMS image provided by Brocade. |