BROCADE

Release 1.1.1 for the Brocade Vyatta Controller

Release Notes v1.0

January 12, 2015

Document History

Document Title	Summary of Changes	Publication Date
Release 1.1.1 for the Brocade Vyatta Controller Release Notes	New document for the GA release	January 12, 2015

Contents

Introducing in This Release

Brocade Vyatta Controller release 1.1.0 introduced the Brocade Vyatta Controller, the first commercial platform built directly from the OpenDaylight Helium release. The controller is an open network platform that facilitates migration to software-defined networking. Release 1.1.1 of the Brocade Vyatta Controller is the latest release of the controller.

New Features

Release 1.1.1 includes the following new feature:

Brocade Vyatta vRouter 5600 Element Management System

The Brocade Vyatta vRouter 5600 Element Management System (EMS) is an app that is developed on the controller to manage the Vyatta 5600 vRouters. Although EMS is installed as a separate app, it is accessible on the controller GUI.

For more information about the Brocade Vyatta EMS, see *Brocade Vyatta vRouter 5600 EMS User Guide*.

Support for Brocade Vyatta Controller Apps

App	Brocade Vyatta Controller version	
	1.1.0	1.1.1
Path Explorer 1.1.0	Supported	Supported
Vyatta EMS 1.1.0	Unsupported	Supported

Security

There are no security issues in Release 1.1.1.

Behavior Changes

There are no new behavior changes in Release 1.1.1.

Limitations

Release 1.1.1 contains the limitations that are associated with the YANG UI, the EMS application, and Path Explorer.

Limitations of the YANG UI

While the YANG UI provides the REST APIs to configure the modules for mounted routers, as stated in the corresponding YANG configurations, some PUT, POST, and DELETE requests may generate errors. These errors may be caused by the router rejecting such requests. The error may be inherent in the logic of the request, and the YANG UI may not get complete error details about the request. If you have any questions about the errors for any particular API, see the configuration manual for the router.

Limitations of the Brocade Vyatta vRouter 5600 EMS

- The Brocade Vyatta vRouter 5600 Element Management System (EMS) uses IPsec VPN Site-to- Site to build the VPN tunnels.
- The EMS, by default, supports a maximum of 250 pairs. This limit is due to the IP address schema that EMS uses to automatically assign the tunnel IP addresses that range from 172.16.0.x/30 through 172.16.255.x/30 for subnets.
- The EMS IPSec VPN peer authentication mode uses, by default, a pre-shared-secret. It does not support the rsa-key-name or the x509 certificate.
- The EMS uses the same IKE group and ESP group configuration across multiple IPSec site-to-site peer connections.
- The EMS IPsec VPN does not support Network Address Translation (NAT) networks and NAT traversal.
- The device name for the router can be any name, but do not enter special characters such as the colon, semicolon, slash, backslash, space, and so on. These characters may interfere with the REST requests to the browser. A hyphen or an underscore is allowed.
- When you mount a device, it may sometimes take longer than usual for the controller to register the mount. If the system does not display the mounted device within 10 seconds, click **Refresh** or use the **Refresh** button on the browser to reload the data.
- When you unmount a device, sometimes it may take longer for the controller to unregister the mount. If the system does not remove the mounted device within 10 seconds, click **Refresh** or use the **Refresh** button on the browser to reload the data.
- When you create a tunnel between two routers, sometimes it may take longer to establish all the router configurations. If the system does not display the tunnel within 20 seconds, click **Refresh** or use the use the **Refresh** button on the browser to reload the data.

Limitations of Path Explorer

- Path Explorer calculates the shortest path between two hosts, and creates a path between them. The shortest path is based on the number of hops, and not based on any other cost-metric elements, such as bandwidth and latency.
- You must manually update the path in the Path Explorer app when the network topology changes.

Limitations of the OpenFlow plug-in

The following features are not supported on the OpenFlow plug-in:

- Table configuration and capabilities learning
- Port configuration
- Queue configuration
- DTLS-based connection establishment
- Role request

- Clustering
- Flow table features statistics

Limitations of Host Tracker

The ARP Handler module of Host Tracker has performance limitations. When starting an OpenFlow network of 50 or more switches, you have to wait for approximately one minute while the module processes the OpenFlow topology before you can start sending network traffic. The module needs that much time because of the following:

- When Host Tracker is in the Active mode, the application needs the time to push the flood flows to all switches.
- When Host Tracker is in the Passive mode, the application needs the time to handle and dispatch the ARP packets generated by the hosts.

To prevent further performance degradation, we recommend that you do not exceed 100 switches in your OpenFlow topology because ARP Handler might not behave correctly.

Upgrade Notes

For the procedure to upgrade the controller, see *Brocade Vyatta Controller Quick Start Guide*.

Known Issues

The following tables list the issues that are associated with the features of the Brocade Vyatta Controller.

OpenFlow
Bug ID: ODL-248
The NetIron family of Brocade switches does not support some of the flows that are injected by the Host Tracker feature. Consequently, the Brocade Vyatta Controller does not learn about the hosts that are attached to the OpenFlow network. **Solution:** Switch to the Passive mode of the Host Tracker. For more information, see the Host Tracker chapter in *Brocade Vyatta Controller User Guide*.

Clustering
Bug ID: ODL-306

When a transaction is created, the implementation of the clustered data store captures a stable snapshot of the current data tree in a logical store (configuration or operational) when the store is first accessed by means of a read, put, merge, or delete operation. This behavior differs from that of the DataBroker API contract, which states that the stable snapshot is captured when the transaction is created. The divergence is further complicated by the fact that each logical store subdivides its tree into any number of isolated shards; thus, a single client transaction may entail multiple shard subtransactions. The clustered data store is implemented in this manner for performance: to adhere to the current API contract, a subtransaction would have to be preallocated for every shard.
 	
This means, for example, that the following client scenario may not yield the behavior as documented in the DataBroker API contract:
 	
a = newReadWriteTransaction(); b = newReadWriteTranasction();
 	
// Returns null, PATH did not exist at time of allocation a and b. a.read(CONFIGURATION,PATH);
 	
// Write foo, submit it and assume the commit completes and the change is published.
 	
a.put(CONFIGURATION,PATH,foo);
 	
a.submit();
 	
// Should return null due to transaction isolation but will return foo since the shard sub-transaction was
 	
// lazily allocated after it was committed. b.read(CONFIGURATION,PATH)

NETCONF connector
Bug ID: ODL-307
• RESTCONF is a request-response model. So, basically, you select a tree, set the configuration for some entities under that tree, and issue a call. A corresponding request call is sent immediately as a part of this request. You cannot cache multiple RESTCONF requests and make a single NETCONF call. • RESTCONF does not support all operations supported by NETCONF such as copy-config and others. • The devices to which the controller connects need to implement NETCONF according to RFC specifications. Otherwise, issues may rise either in connecting to it or managing it. • Ensure that the NETCONF connector feature is up and running in the controller. • Otherwise, the controller cannot connect to NETCONF devices.

Controller shutdown
Bug ID: ODL-308
When the controller is shut down by using the “stop” script, the karaf.log file displays the following error. 2014-11-18 11:49:21,927 | ERROR | Bundle Shutdown | DestroyedModule | 126 - org.opendaylight.controller.config-manager - 0.2.6.Helium-SR1 | Error while closing instance of ModuleIdentifier{factoryName='host- tracker-impl', instanceName='host-tracker-impl'} java.lang.IllegalStateException: Transaction factory was closed. No further operations allowed. This error is benign, and the message can be ignored.

GUI
Bug ID: ODL-305, ODL-317
• The browser has to be occasionally refreshed to get the latest hops. • If something adversely affects the controller during a path operation, no error message is displayed. • Sometimes, it is not possible to access the web GUI from a remote host. If the controller is installed on a machine with more than one network card, the controller uses the IP address of the first network interface card that it gets by using the ifconfig command. If this IP address is not reachable from the external network, the user is not able to access the controller web GUI from the remote machine. Although the user can access the web GUI login page by using the externally reachable IP address of the controller machine, the user is not able to log in successfully. The Brocade Vyatta Controller web GUI uses the chosen IP address to send REST calls directly to the controller. As that IP address is not externally reachable, the call fails.

Solution:

1. Change the IP address in the following file to the externally reachable IP address.
   `/opt/bvc/web/config.json`
2. Refresh the web page, and try logging in again.

An alternate method to discover the IP address through the Development Tool of Google Chrome:

1. Go to **Settings** > **More Tools** > **Development Tools**. This action opens a window at the bottom of browser window.
2. Click the **Network** tab in the Development Tool window.
3. Access the controller web GUI, and log in with the credentials that were provided.
4. The Network tab shows the failed REST call that is sent by the web GUI to the controller. The REST call details include the controller IP address that is used by the web GUI to reach the controller.

Failure in GUI login after setting up HTTPS

Bug ID: ODL-384

As part of the checks before establishing a secure connection, the browser (client) uses the certification authority (CA) certificate to verify the CA signature on the server certificate.

After setting up the HTTPS service on the server that is running the controller, navigation to the URL of the GUI if possible. However, authentication with admin and admin as the username and password fails.

The issue is unavoidable with self-signed certificates that are used for authentication. If a user uses the certificate that is signed by a well-known CA, the browser accepts the certificate and authenticates the user.

Work-around

If the browser (client) is unaware of the real certificate in CA, use the following work-around:

1. Navigate to http://host:8443/restconf/modules.
2. Use the Advanced option to accept the certificate.

Bypassing the issue

The issue can be avoided the following way:

1. Enter the following command:
   `./bin/stop`
2. Enter the following command:
   `./bin/setup_https on`
3. Enter the following command:
   `./bin/start`
4. Navigate to https://host:8443/restconf/modules.
5. Set the browser to **Allow** non-verified certificates.

6. Navigate to http://host:9000.
7. Set the browser to **Allow** non-verified certificates.

Clustering functionality limitations
Bug ID: ODL-385
In some cases, even if a majority of the controller instances are available, some functionality may not work as expected for the clustering feature. Specifically, if the controller instance that is handling the connection between the OpenFlow plug-in and an OpenFlow switch is unavailable, attempts to program or query that switch by means of the controller fail even if a majority of the controller instances are available. This is because OpenFlow switch failover is not supported. Additionally, if REST calls are made to a specific controller instance, they fail if that instance is unavailable. Calls can be made to a different instance but do not automatically failover.

Upgrading a controller with clustering
Bug ID: ODL-417
Clusters may not persist after a controller upgrade. It is recommended that you uninstall the old version of the controller, install the new version, and set the cluster up again.

Loss of HTTPS configuration upon a controller upgrade
Bug ID: ODL-418
Upon an upgrade of the controller, the server is restarted with HTTP instead of HTTPS. **Resolution** 1. Enter the following commands to resolve the overwriting of HTTPS configurations and to re-enable HTTPS. a. ./bin/stop b. ./bin/setup_https off c. ./bin/setup_https on d. ./bin/start