|  |
| --- |
| Push Agent  User Guide |
| Jan 2017 |

Notices

Legal notice

© 2016, Hewlett Packard Enterprise Development LP.

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document History | | | | |
| Version No. | Version date | Updates | Approved by |  |
| 0.1 | 01/19/2017 | Base Line |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| \* Action types: Approve, Review, Inform, File, Action Required, Attend Meeting, Other (please specify) | | | | |

# Table of contents

[1 Document History 2](#_Toc472628240)

[2 Table of contents 3](#_Toc472628241)

[3 Introduction 6](#_Toc472628242)

[3.1 Purpose 6](#_Toc472628243)

[3.2 Audience 6](#_Toc472628244)

[3.3 References 6](#_Toc472628245)

[4 Installation 7](#_Toc472628246)

[4.1 Precondition 7](#_Toc472628247)

[4.2 Installation Steps 7](#_Toc472628248)

[5 Post-Installation 7](#_Toc472628249)

[5.1 JDK Configuration 7](#_Toc472628250)

[5.2 PushAgent Configuration 8](#_Toc472628251)

[5.2.1 Configuration Items 8](#_Toc472628252)

[5.2.2 Multiple PushAgent Configuration 9](#_Toc472628253)

[6 Operation 11](#_Toc472628254)

List of figures

**No table of figures entries found.**

List of tables

**No table of figures entries found.**

# Introduction

## Purpose

The purpose of this document is to guide users through deploy, configuration and use PushAgent.

## Audience

The audience includes end users of the solution.

## References

# Installation

## Precondition

|  |  |
| --- | --- |
| Software | Version |
| JDK | 1.7 + |
| bdap.pushagent | VVERSION |

## Installation Steps

After get <bdap.pushagent-VVERSIONN-package.tar.gz>, please execute following command to install the PushAgent

|  |
| --- |
| 1. mkdir –p <PushAgentHome> 2. copy bdap.pushagent-VVERSIONN-package.tar.gz to <PushAgentHome> 3. cd <PushAgentHome> 4. tar -xvf bdap.pushagent-VVERSIONN-package.tar.gz |

When complete the installation, you will see such structure:

|  |
| --- |
| <PushAgentHome>  |----README.md  |----bin  |----bdap.pushagent-VVERSIONN.jar  |----config.json  |----start.sh  |----stop.sh |

# Post-Installation

## JDK Configuration

Make sure the JDK is set in the PATH environment variable. After complete please execute following command on shell to verify the jdk version is greater than 1.7

|  |
| --- |
| java -version |

## PushAgent Configuration

## Configuration Items

We can find the PushAgent configuration file : <PushAgentHome>/bin/config.json. Following is a sample to running PushAgent on IPv6 environment.

|  |
| --- |
| {  "OMFile" : { -----------------------------------------------------------------------------------------------------------------------------1  "id" : "OMFile", -----------------------------------------------------------------------------------------------------------------------2  "directory" : "/tmp/test/source/om", ------------------------------------------------------------------------------------------3  "elements" : [ {------------------------------------------------------------------------------------------------------------------------4  "name" : "usplselux185.eslabs.ssn.hp.com",-------------------------------------------------------------------------------5  "hostname" : "usplselux185.eslabs.ssn.hp.com",---------------------------------------------------------------------------6  "ip" : "fe80::ae16:2dff:fe98:ae64"----------------------------------------------------------------------------------------------7  } ],  "category" : "OM",--------------------------------------------------------------------------------------------------------------------8  "timeZone" : "GMT",-----------------------------------------------------------------------------------------------------------------9  "cronExpr" : "0/1 \* \* \* \* ?", --------------------------------------------------------------------------------------------------------10  "recursive" : true, ---------------------------------------------------------------------------------------------------------------------11  "filenameFilterExpr" : "new('bdap.tools.pushagent.PathFilter', WorkingDir, '2016\*/\*\*', false)", ----------------12 "filesPerBatch" : 1, -----------------------------------------------------------------------------------------------------------------------13  "processRecordFile" : "./omfile\_process\_record",----------------------------------------------------------------------------14  "destServer" : "fe80::ae16:2dff:fe98:b1e8",------------------------------------------------------------------------------------15  "destServerPort" : 22, ----------------------------------------------------------------------------------------------------------------16  "destServerUser" : "dbadmin",-----------------------------------------------------------------------------------------------------17  "destServerPass" : "password",----------------------------------------------------------------------------------------------------18  "destServerDirRule" : "`/data/femtocell/omfiles/raw/${WorkingElement.IP}`"---------------------------------------19  }  } |

1. One kind of push agent job name, it should unique. And can configure multiple such element.
2. Job Id, it should be unique as #1
3. The scan/monitor folder on the push agent deployed server.
4. It is an array, used to specify which PushAgent execute this job. If there are multiple PushAgent to execute this job, we can configuration another set of “name”, “hostname” and “ip”.
5. The PushAgent server name
6. The PushAgent server hostname
7. The IP of PushAgent sever, it could be ipv4 or ipv6. PushAgent server will use this to identify which job it should execute.
8. Category of job
9. Timezone used for the job trigger time
10. It is a cron expression to specify when the job is executed. For detail how to configure, please refer to <http://www.quartz-scheduler.org/documentation/quartz-2.x/tutorials/crontrigger.html>
11. To indicate whether scan the subfolder of the monitored folder.
12. It is a filter to filter out the matched folder/files need to push.
13. Every time the job is executed, maximum how many files to be pushed.
14. It is a history file to record the latest push files’ status to avoid duplicate process same file. For each job, should have a unique file to store the status.
15. The destination server ipv4/ipv6 address, to which the PushAgent will push the files.
16. Destination server sftp port
17. Destination server username
18. Destination server password
19. The path on destination server to store the pushed files.
    1. For the config item - destServerDirRule, has below variable context:
    2. \* WorkingElement {Name, Hostname, IP}
    3. \* WorkingDir
    4. \* DestServer



## Multiple PushAgent Configuration

There is only one config.json file, and every PushAgent can find out which job this PushAgent need to execute by its IP address. Following is a sample:

Config.json

|  |
| --- |
| {  "Job1" : {  "id" : "Job1",  "directory" : "/tmp/test/source/om",  "elements" : [ {  "name" : "PushAgent A name",  "hostname" : "PushAgnet A hostname",  "ip" : "PushAgnet A IP address"  },{  "name" : "PushAgent B name",  "hostname" : "PushAgnet B hostname",  "ip" : "PushAgnet B IP address"  } ],  ......  }, "Job2" : {  "id" : "Job2",  "directory" : "/tmp/test/source/sysperf",  "elements" : [ {  "name" : "PushAgent C name",  "hostname" : "PushAgnet C hostname",  "ip" : "PushAgnet C IP address"  } ],  ......  }  } |



* Job1’s Elements contains PushAgent A & B IP address as above. PushAgent A&B will know the Job1 they should execute.
* Job2’s Elements doesn’t contain Push Agent A&B, so Job2 only executed by PushAgent C

# Operation

After complete above steps. User can start or stop the PushAgent as following:

Start PushAgent

|  |
| --- |
| <PushAgentHome>/bin/start.sh |

Monitor PushAgent after start

|  |
| --- |
| tail -f <PushAgentHome>/bin/log.log |

Stop PushAgent

|  |
| --- |
| <PushAgentHome>/bin/stop.sh |