**PL/Container python3 설치 가이드**

* 이 문서는 CentOS 7.x 환경의 Greenplum 6.x 버전에서 PL/Container python3를 설치 및 구성하는 방법을 설명 함
* 참조 URL : https://gpdb.docs.pivotal.io/6-2/ref\_guide/extensions/pl\_container.html

1. **사전 요구사항**

* For PL/Container 2.1.x use Greenplum Database 6 on CentOS 7.x (or later), RHEL 7.x (or later), or Ubuntu 18.04.

|  |
| --- |
| Note: PL/Container 2.1.x supports Docker images with Python 3 installed. |

* For PL/Container 3 Beta use Greenplum Database 6.1 or later on CentOS 7.x (or later), RHEL 7.x (or later), or Ubuntu 18.04.
* The minimum Linux OS kernel version supported is 3.10. To verfiy your kernel release use:

|  |
| --- |
| $ uname -r |

* The minimum Docker versions on all hosts needs to be Docker 19.03.

1. **Docker 설치**

* PL/Container 설치를 위해서는 모든 Greenplum 클러스터 서버에 Docker 버전19.03 이상이 설치 되어야 함
* Docker 설치는 sudo 권한이 있는 계정 또는 root 계정으로 진행 (이 문서에서는 root 계정으로 설치 진행)

1. OS kernel 버전 확인

* 준비사항 : root 계정으로 gpssh 명령을 이용하여 Greenplum 클러스터의 모든 서버에 접속 가능하도록 설정 필요
* 참고자료 : GPDB6\_Install\_SOP\_20201023.xlsx 파일 (다운로드 URL : https://github.com/zhyun-pivotal/gpdb6\_install)

|  |
| --- |
| [root@mdw ~]# gpssh -f /data/staging/hostfile  => uname -r  [smdw] 3.10.0-1062.12.1.el7.x86\_64  [sdw1] 3.10.0-1062.12.1.el7.x86\_64  [sdw2] 3.10.0-1062.12.1.el7.x86\_64  [mdw] 3.10.0-1062.12.1.el7.x86\_64 |

1. Docker에 필요한 종속성 설치

* Local repo를 구성하기 위해서 다운로드만 받을 경우 “yum install –downloadonly” 명령어를 이용하여 아래 패키지 다운로드
* 기본 다운로드 경로는 “/var/cache/yum/x86\_64/7”이며, “--downloaddir=<directory>” 옵션으로 다운로드 경로 지정도 가능함

|  |
| --- |
| [root@mdw ~]# gpssh -f /data/staging/hostfile  => yum install -y yum-utils device-mapper-persistent-data lvm2  [smdw] Loaded plugins: fastestmirror, langpacks  [smdw] Loading mirror speeds from cached hostfile  [smdw] \* base: data.aonenetworks.kr  [smdw] \* extras: data.aonenetworks.kr  [smdw] \* updates: data.aonenetworks.kr  …  [smdw] Trying other mirror.  updates | 2.9 kB 00:00  [smdw] Package yum-utils-1.1.31-52.el7.noarch already installed and latest version  [smdw] Package device-mapper-persistent-data-0.8.5-1.el7.x86\_64 already installed and latest version  [smdw] Package 7:lvm2-2.02.185-2.el7\_7.2.x86\_64 already installed and latest version  [smdw] Nothing to do |

1. Docker repo 추가

* Local repo를 구성하기 위해서 docker-ce를 다운로드 받기 위해서는 yum-utils를 설치 필요
* yum-utils를 설치해야 yum-config-manager 명령어 사용이 가능함

|  |
| --- |
| [root@mdw ~]# gpssh -f /data/staging/hostfile  => yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo  [smdw] Loaded plugins: fastestmirror, langpacks  [smdw] adding repo from: https://download.docker.com/linux/centos/docker-ce.repo  [smdw] grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo  [smdw] repo saved to /etc/yum.repos.d/docker-ce.repo |

1. yum cache 업데이트

|  |
| --- |
| [root@mdw ~]# gpssh -f /data/staging/hostfile  => yum makecache fast  [smdw] Loaded plugins: fastestmirror, langpacks  [smdw] Loading mirror speeds from cached hostfile  [smdw] \* base: data.aonenetworks.kr  [smdw] \* extras: data.aonenetworks.kr  [smdw] \* updates: data.aonenetworks.kr  …  [smdw] Trying other mirror.  updates | 2.9 kB 00:00  (1/2): docker-ce-stable/7/x86\_64/updateinfo | 55 B 00:00  (2/2): docker-ce-stable/7/x86\_64/primary\_db | 46 kB 00:00  [smdw] Metadata Cache Created |

1. Docker 설치

* Local repo를 구성하기 위해서 다운로드만 받을 경우 “yum install –downloadonly” 명령을 이용하여 아래 패키지 다운로드

|  |
| --- |
| [root@mdw ~]# gpssh -f /data/staging/hostfile  => yum -y install docker-ce  [smdw] Loaded plugins: fastestmirror, langpacks  [smdw] Loading mirror speeds from cached hostfile  [smdw] \* base: data.aonenetworks.kr  [smdw] \* extras: data.aonenetworks.kr  [smdw] \* updates: data.aonenetworks.kr  [smdw] Resolving Dependencies  ...  [smdw]  [smdw] Installed:  [smdw] docker-ce.x86\_64 3:19.03.13-3.el7  [smdw]  [smdw] Dependency Installed:  [smdw] container-selinux.noarch 2:2.107-3.el7 containerd.io.x86\_64 0:1.3.7-3.1.el7  [smdw] docker-ce-cli.x86\_64 1:19.03.13-3.el7  [smdw]  [smdw] Complete! |

1. Docker daemon 시작 및 확인

* Docker daemon이 서버 시작 시에 자동으로 기동 되도록 ‘systemctl enable docker.service’ 명령을 수행

|  |
| --- |
| [root@mdw ~]# gpssh -f /data/staging/hostfile  => systemctl enable docker.service  [smdw] Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.  => systemctl start docker.service  => systemctl status docker.service  [smdw] ● docker.service - Docker Application Container Engine  [smdw] Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)  [smdw] Active: active (running) since Thu 2020-10-29 21:27:42 KST; 22s ago  [smdw] Docs: https://docs.docker.com  …  [smdw] Oct 29 21:27:42 smdw systemd[1]: Started Docker Application Container Engine.  [smdw] Hint: Some lines were ellipsized, use -l to show in full.  => ps -ef | grep docker | grep -v grep  [smdw] root 2989 1 1 21:27 ? 00:00:01 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock |

1. Docker 그룹에 gpadmin 계정 추가

* Docker 데몬 및 docker 명령에 대한 액세스 권한을 gpadmin 계정에 부여하기 위해서 gpadmin 계정을 docker 그룹에 할당

|  |
| --- |
| [root@mdw ~]# gpssh -f /data/staging/hostfile  => usermod -aG docker gpadmin |

1. gpadmin계정으로 docker 명령 확인

* ‘docker ps’ 명령은 현재 실행중인 docker 컨테이너 목록을 확인

|  |
| --- |
| [root@mdw ~]# su - gpadmin  Last login: Thu Oct 29 21:02:14 KST 2020 from 192.168.23.151 on pts/5  [gpadmin@mdw ~]$ docker ps  CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES |

1. GPDB 재시작

* 모든 Greenplum 클러스터 서버에 Docker를 설치 한 후 Greenplum 데이터베이스 시스템을 다시 시작하여 GPDB에 Docker에 대한 액세스 권한을 부여

1. **PL/Container Language Extension Package 설치**
2. Language Extension Package 파일 업로드 및 설치 전 확인

* GPDB 버전과 호환되는 Language Extension Package 파일을 Tanzu Network 사이트 (https://network.pivotal.io/products/pivotal-gpdb)에서 다운로드 받아서 master 서버에 업로드 함
* Language Extension package 파일 및 python3 docker image 파일 모두 다운로드 받아야 함
* 업로드 한 파일은 gpadmin 계정의 소유인지 확인 필요
* /usr/local/greenplum-db-6.x.x 폴더가 gpadmin 계정의 소유인지 확인 필요

|  |
| --- |
| [gpadmin@mdw ~]$ ls -al /data/staging/ | grep plcontainer  -rw-r--r-- 1 gpadmin gpadmin 618027 Oct 29 22:32 plcontainer-2.1.1-gp6-rhel7\_x86\_64.gppkg  -rw-r--r--. 1 gpadmin gpadmin 1996485752 Oct 29 14:56 plcontainer-python3-image-2.1.1-gp6.tar.gz  [gpadmin@mdw ~]$ ls -al /usr/local | grep greenplum-db  lrwxrwxrwx 1 gpadmin gpadmin 29 Oct 29 16:10 greenplum-db -> /usr/local/greenplum-db-6.7.1  drwxr-xr-x 12 gpadmin gpadmin 202 Oct 29 16:10 greenplum-db-6.7.1 |

1. Language Extension Package 설치

|  |
| --- |
| [gpadmin@mdw ~]$ cd /data/staging/  [gpadmin@mdw staging]$ gppkg -i plcontainer-2.1.1-gp6-rhel7\_x86\_64.gppkg  20201029:22:36:07:006940 gppkg:mdw:gpadmin-[INFO]:-Starting gppkg with args: -i plcontainer-2.1.1-gp6-rhel7\_x86\_64.gppkg  20201029:22:36:08:006940 gppkg:mdw:gpadmin-[INFO]:-Installing package plcontainer-2.1.1-gp6-rhel7\_x86\_64.gppkg  …  20201029:22:36:10:006940 gppkg:mdw:gpadmin-[INFO]:-Completed local installation of plcontainer-2.1.1-gp6-rhel7\_x86\_64.gppkg.  20201029:22:36:10:006940 gppkg:mdw:gpadmin-[INFO]:--  ==========================================================================  PL/Container installation is complete! To proceed, follow these steps:  1. Source your new $GPHOME/greenplum\_path.sh file and restart the database.  2. Enable PL/Container language in one of your databases by running "CREATE EXTENSION plcontainer;" directly in your database  3. Add docker images on master node, for example add a python image by running "plcontainer image-add -f plcontainer-python-images.tar.gz;"  4. Add container runtimes on master node, for example add a runtime by running "plcontainer runtime-add -r plc\_python\_shared -i pivotaldata/plcontainer\_python\_shared:devel -l python;" Refer to the installation guide for options  ==========================================================================  20201029:22:36:10:006940 gppkg:mdw:gpadmin-[INFO]:-plcontainer-2.1.1-gp6-rhel7\_x86\_64.gppkg successfully installed. |

1. greenplum-path.sh 파일 적용 및 GPDB 재시작

|  |
| --- |
| [gpadmin@mdw ~]$ source /usr/local/greenplum-db/greenplum\_path.sh  [gpadmin@mdw ~]$ gpstop -raf  20201029:22:40:46:007182 gpstop:mdw:gpadmin-[INFO]:-Starting gpstop with args: -raf  20201029:22:40:46:007182 gpstop:mdw:gpadmin-[INFO]:-Gathering information and validating the environment...  20201029:22:40:46:007182 gpstop:mdw:gpadmin-[INFO]:-Obtaining Greenplum Master catalog information  20201029:22:40:46:007182 gpstop:mdw:gpadmin-[INFO]:-Obtaining Segment details from master...  …  20201029:22:40:50:007182 gpstop:mdw:gpadmin-[INFO]:-Database successfully shutdown with no errors reported  20201029:22:40:50:007182 gpstop:mdw:gpadmin-[INFO]:-Cleaning up leftover gpmmon process  20201029:22:40:50:007182 gpstop:mdw:gpadmin-[INFO]:-No leftover gpmmon process found  20201029:22:40:50:007182 gpstop:mdw:gpadmin-[INFO]:-Cleaning up leftover gpsmon processes  20201029:22:40:50:007182 gpstop:mdw:gpadmin-[INFO]:-No leftover gpsmon processes on some hosts. not attempting forceful termination on these hosts  20201029:22:40:50:007182 gpstop:mdw:gpadmin-[INFO]:-Cleaning up leftover shared memory  20201029:22:40:51:007182 gpstop:mdw:gpadmin-[INFO]:-Restarting System... |

1. 사용자 DB에 plcontainer language 생성

|  |
| --- |
| [gpadmin@mdw ~]$ psql  psql (9.4.24)  Type "help" for help.  gpdemo=# CREATE EXTENSION plcontainer;  CREATE EXTENSION |

1. **Docker image 설치**
2. docker image 설치

* master 서버에서 docker image 파일 설치를 실행하여 Greenplum 클러스터의 모든 호스트에 docker image 설치 진행

|  |
| --- |
| [gpadmin@mdw ~]$ plcontainer image-list  REPOSITORY TAG IMAGE ID CREATED SIZE  [gpadmin@mdw ~]$ plcontainer image-add -f /data/staging/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201029:23:06:34:008495 plcontainer:mdw:gpadmin-[INFO]:-Checking whether docker is installed on all hosts...  20201029:23:06:34:008495 plcontainer:mdw:gpadmin-[INFO]:-Distributing image file /data/staging/plcontainer-python3-image-2.1.1-gp6.tar.gz to all hosts...  20201029:23:08:37:008495 plcontainer:mdw:gpadmin-[INFO]:-Loading image on all hosts...  20201029:23:14:52:008495 plcontainer:mdw:gpadmin-[ERROR]:-error running command: docker load -i /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201029:23:14:52:008495 plcontainer:mdw:gpadmin-[CRITICAL]:-plcontainer failed. (Reason='error running command') exiting...  [gpadmin@mdw staging]$ plcontainer --verbose image-add -f /data/staging/plcontainer-python3-image-2.1.1-gp6.tar.gz  ...  20201030:11:16:03:003308 plcontainer:mdw:gpadmin-[INFO]:-Checking whether docker is installed on all hosts...  ...  20201030:11:16:03:003308 plcontainer:mdw:gpadmin-[INFO]:-Distributing image file /data/staging/plcontainer-python3-image-2.1.1-gp6.tar.gz to all hosts...  ...  20201030:11:16:42:003308 plcontainer:mdw:gpadmin-[INFO]:-Loading image on all hosts...  ...  20201030:11:18:23:003308 plcontainer:mdw:gpadmin-[INFO]:-Removing temporary image files on all hosts...  20201030:11:18:23:003308 plcontainer:mdw:gpadmin-[DEBUG]:-Adding cmd to work\_queue: rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201030:11:18:23:003308 plcontainer:mdw:gpadmin-[DEBUG]:-Adding cmd to work\_queue: rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201030:11:18:23:003308 plcontainer:mdw:gpadmin-[DEBUG]:-[worker6] got cmd: rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201030:11:18:23:003308 plcontainer:mdw:gpadmin-[DEBUG]:-Running Command: rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201030:11:18:23:003308 plcontainer:mdw:gpadmin-[DEBUG]:-[worker7] got cmd: rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201030:11:18:23:003308 plcontainer:mdw:gpadmin-[DEBUG]:-Running Command: rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  20201030:11:18:24:003308 plcontainer:mdw:gpadmin-[DEBUG]:-[worker7] finished cmd: Execute remote cmd on all hosts. cmdStr='ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=60 mdw ". /usr/local/greenplum-db/./greenplum\_path.sh; rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz"' had result: cmd had rc=0 completed=True halted=False  stdout=''  stderr=''  20201030:11:18:24:003308 plcontainer:mdw:gpadmin-[DEBUG]:-[worker6] finished cmd: Execute remote cmd on all hosts. cmdStr='ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=60 gps1 ". /usr/local/greenplum-db/./greenplum\_path.sh; rm -f /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz"' had result: cmd had rc=0 completed=True halted=False  stdout=''  stderr=''  20201030:11:18:24:003308 plcontainer:mdw:gpadmin-[DEBUG]:-Shutting down workers...  20201030:11:18:24:003308 plcontainer:mdw:gpadmin-[DEBUG]:-WorkerPool haltWork() |

1. plcontainer 이미지 테스트

* plcontainer image-add 명령이 실패할 경우 docker load 명령으로 image를 docker에 load 테스트 진행

|  |
| --- |
| [gpadmin@mdw ~]$ docker load -i /tmp/plcontainer-python3-image-2.1.1-gp6.tar.gz  cc967c529ced: Loading layer [============================>] 65.57MB/65.57MB  2c6ac8e5063e: Loading layer [============================>] 991.2kB/991.2kB  6c01b5a53aac: Loading layer [============================>] 15.87kB/15.87kB  e0b3afb09dc3: Loading layer [============================>] 3.072kB/3.072kB  0430d089b99f: Loading layer [============================>] 3.584kB/3.584kB  9c6f28497f21: Loading layer [============================>] 3.072kB/3.072kB  83fe36e29e55: Loading layer [============================>] 5.547GB/5.547GB  Loaded image: pivotaldata/plcontainer\_python3\_shared:devel |

1. plcontainer image-list 확인

|  |
| --- |
| [gpadmin@mdw staging]$ plcontainer image-list  REPOSITORY TAG IMAGE ID CREATED SIZE  pivotaldata/plcontainer\_python3\_shared devel 93f353853dbd 10 months ago 5.55GB |

1. PL/Container 구성 파일에 런타임 항목을 추가

* Docker 이미지를 설치 한 후 PL/Container 구성 파일에서 런타임 항목을 추가하거나 업데이트하여 PL/Container에 Docker 컨테이너를 시작하기 위해 Docker 이미지에 대한 액세스 권한을 부여

|  |
| --- |
| [gpadmin@mdw staging]$ plcontainer runtime-add -r plc\_python\_shared -i pivotaldata/plcontainer\_python3\_shared:devel -l python3  20201030:13:38:10:004275 plcontainer:mdw:gpadmin-[INFO]:-Distributing file plcontainer\_configuration.xml to all locations...  20201030:13:38:10:004275 plcontainer:mdw:gpadmin-[INFO]:-Configuration has changed. Run "select \* from plcontainer\_refresh\_config" in open sessions. New sessions will get new configuration automatically. |

1. 런타임 항목 확인

* 구성 파일에서 주어진 런타임 ID로 특정 런타임 표시

|  |
| --- |
| [gpadmin@mdw staging]$ plcontainer runtime-show -r plc\_python\_shared  PL/Container Runtime Configuration:  ---------------------------------------------------------  Runtime ID: plc\_python\_shared  Linked Docker Image: pivotaldata/plcontainer\_python3\_shared:devel  Runtime Setting(s):  Shared Directory:  ---- Shared Directory From HOST '/usr/local/greenplum-db/./bin/plcontainer\_clients' to Container '/clientdir', access mode is 'ro'  --------------------------------------------------------- |

1. 런타임 항목 삭제(항목 재 등록 시 필요)

|  |
| --- |
| [gpadmin@mdw ~]$ plcontainer runtime-show  PL/Container Runtime Configuration:  ---------------------------------------------------------  Runtime ID: plc\_python\_shared  Linked Docker Image: pivotaldata/plcontainer\_python3\_shared:devel  Runtime Setting(s):  Shared Directory:  ---- Shared Directory From HOST '/usr/local/greenplum-db/./bin/plcontainer\_clients' to Container '/clientdir', access mode is 'ro'  ---------------------------------------------------------  [gpadmin@mdw ~]$ plcontainer runtime-delete -r plc\_python\_shared  20201201:21:28:23:002669 plcontainer:gpm1:gpadmin-[INFO]:-Distributing file plcontainer\_configuration.xml to all locations...  20201201:21:28:23:002669 plcontainer:gpm1:gpadmin-[INFO]:-Configuration has changed. Run "select \* from plcontainer\_refresh\_config" in open sessions. New sessions will get new configuration automatically. |

1. python3 함수 테스트

|  |
| --- |
| [gpadmin@mdw staging]$ psql  psql (9.4.24)  Type "help" for help.  gpdemo=# CREATE or REPLACE FUNCTION pylog(a integer, b integer) RETURNS double precision AS $$  gpdemo$# # container: plc\_python\_shared  gpdemo$# import math  gpdemo$# return math.log(a, b)  gpdemo$# $$ LANGUAGE plcontainer;  CREATE FUNCTION  gpdemo=# select pylog(2,3);  pylog  -------------------  0.630929753571457  (1 row) |

1. **PL/Container python2 구성 (Optional)**
2. PL/Container python2 이미지 다운로드

* GPDB 및 plcontainer gppkg 버전에 맞는 PL/Container python2 이미지 파일(예 : plcontainer-python-image-2.1.1-gp6.tar.gz )를 다운로드

1. Docker image 설치

* python2 이미지 파일을 docker에 설치

|  |
| --- |
| plcontainer --verbose image-add -f /<path-to-file>/plcontainer-python-image-2.1.1-gp6.tar.gz |

1. PL/Container 구성 파일에 런타임 항목을 추가

* PL/Container 구성 파일에서 런타임 항목을 추가

|  |
| --- |
| plcontainer runtime-add -r plc\_python2\_shared -i pivotaldata/plcontainer\_python\_shared:devel -l python |

1. python2 함수 테스트

* PL/Container python2 구성 후 함수 생성 및 테스트 진행

|  |
| --- |
| /\* python2 function ddl \*/  CREATE FUNCTION dummyPython2() RETURNS text AS $$  # container: plc\_python2\_shared  return 'hello from Python'  $$ LANGUAGE plcontainer;  /\* python2 function call \*/  select dummyPython2(); |

1. **PL/Container R 구성 (Optional)**
2. PL/Container R 이미지 다운로드

* GPDB 및 plcontainer gppkg 버전에 맞는 PL/Container R 이미지 파일(예 : plcontainer-r-image-2.1.3-gp6.tar.gz )를 다운로드

1. Docker image 설치

* R 이미지 파일을 docker에 설치

|  |
| --- |
| plcontainer --verbose image-add -f /<path-to-file>/plcontainer-r-image-2.1.3-gp6.tar.gz |

1. PL/Container 구성 파일에 런타임 항목을 추가

* PL/Container 구성 파일에서 런타임 항목을 추가

|  |
| --- |
| plcontainer runtime-add -r plc\_r\_shared -i pivotaldata/plcontainer\_r\_shared:devel -l r |

1. R 함수 테스트

* PL/Container R 구성 후 함수 생성 및 테스트 진행

|  |
| --- |
| /\* R function ddl \*/  CREATE FUNCTION dummyR() RETURNS text AS $$  # container: plc\_r\_shared  return ('hello from R')  $$ LANGUAGE plcontainer;  /\* R function call \*/  select dummyR(); |