Sample Scripts to deploy Db2 Analytics Accelerator for z/OS on IBM Z

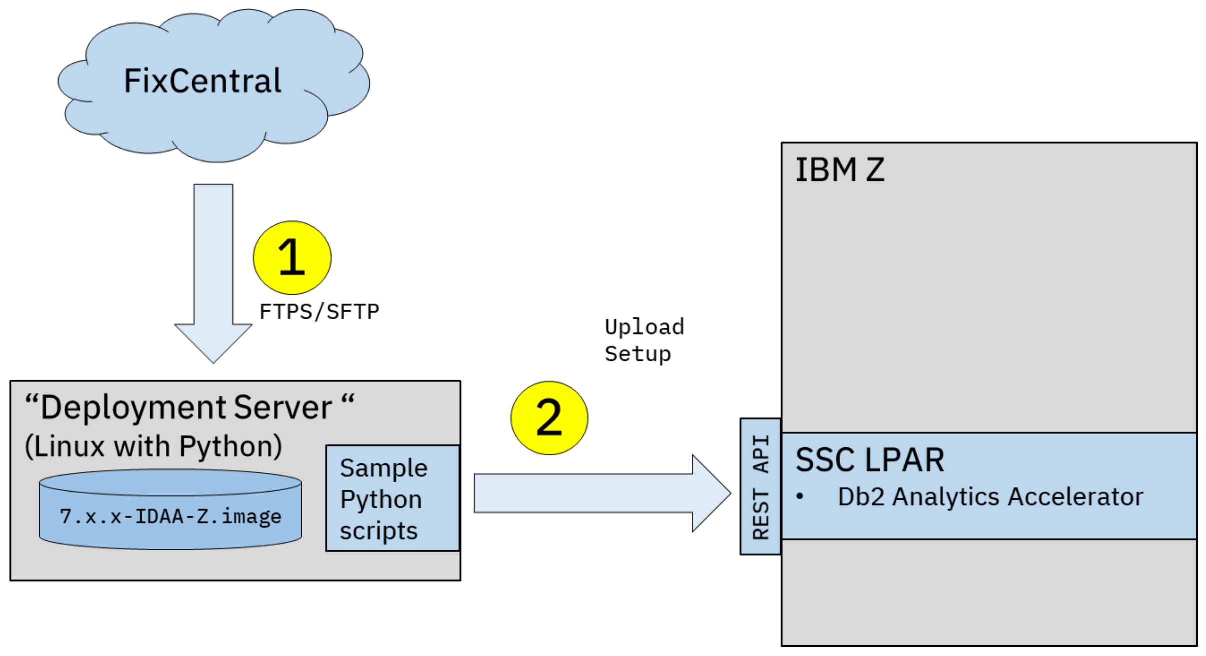
Installation and deployment of Db2 Analytics Accelerator for z/OS on IBM Z (short: the Accelerator) includes the following activities, as described in the installation/deployment section of the IBM Knowledge Center:

- Definition of a Secure Service Container (SSC) LPAR using the Hardware Management Console (HMC), and specifying required resources such as IFLs and RAM
- Retrieving the latest Accelerator image from FixCentral
- Uploading the image file to the LPAR, using the SSC installer web interface
- Create a definition configuration file for additional network interfaces, storage (disks) and optionally settings for additional sites
- Starting initialization of the Accelerator based on this definition configuration file

After these steps, the Accelerator is available for the Db2 for z/OS “pairing” process.

The scripts and procedures provided in this package support an alternative way for deployments by using a set of Python scripts instead of a Web user interface. This could make deployment easier in a production environment. Because Accelerator images are large (several GB in size), the scripts can also be beneficial in environments where system programmers work remotely and otherwise would have to run a web based upload with limited network bandwidth.

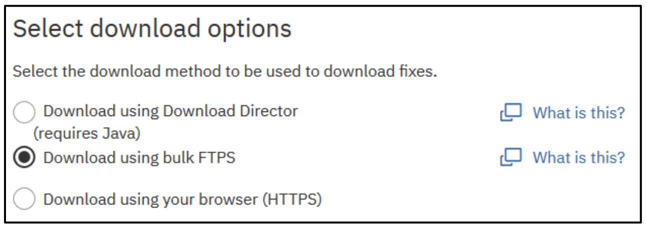
The following picture shows the involved components and flow.



The concept introduces a deployment server running Linux and Python (Version 3). This deployment server needs network interfaces to connect to IBM's Fix Central servers, as well as to the SSC LPAR.

Downloading the Accelerator Image from Fix Central

The Db2 Analytics Accelerator pre-requisite web page (http://www-01.ibm.com/support/docview.wss?uid=swg27050440) includes links which point to a quick order in Fix Central. In addition to web based download options, pick "Download using bulk FTPS".

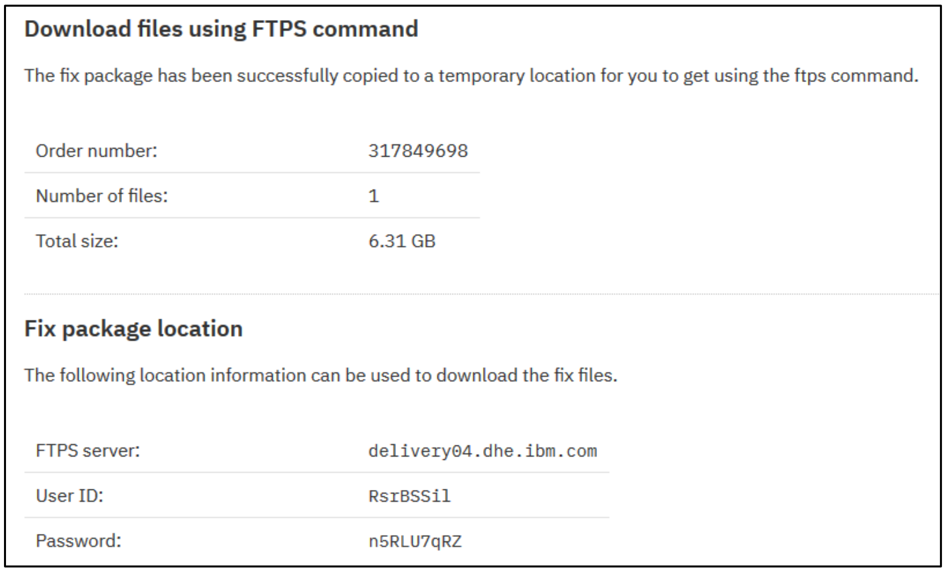


Select download options

Select the download method to be used to download fixes.

- ○ Download using Download Director (requires Java)
- ● Download using bulk FTPS
- ○ Download using your browser (HTTPS)

This will create a temporary location and credentials on an FTPS server:



Download files using FTPS command

The fix package has been successfully copied to a temporary location for you to get using the ftps command.

Order number:	317849698
Number of files:	1
Total size:	6.31 GB

Fix package location

The following location information can be used to download the fix files.

FTPS server:	delivery04.dhe.ibm.com
User ID:	RsrBSSil
Password:	n5RLU7qRZ

On the Linux Deployment Server, a script or the following sequence of shell commands can be used to download the image to a local file system:

```
$ sftp RsrBSSil@delivery04.dhe.ibm.com
RsrBSSil@delivery04.dhe.ibm.com's password:
Connected to delivery04.dhe.ibm.com.
sftp> ls
7.x.x.x-Information_Management-IDAA-Z.image
sftp> binary
sftp> passive
sftp> mget *
```

Python Sample Scripts

A set of Python sample scripts are available to handle the steps required during an installation and deployment. They can be invoked directly on the command line.

The “-h” command line option shows help and command line parameters:

```
> aqt-upload.py -h
***********************************************************

  Db2 Analytics Accelerator image upload

***********************************************************
usage: aqt-upload.py [-h] [--bootudid BOOT_UDID] [--licpath LICPATH] [-v]
                     LPAR_IP LPAR_USER LPAR_PASSWORD BOOTDEVICE_ID IMAGE_NAME

Db2 Analytics Accelerator image upload

positional arguments:
  LPAR_IP               The IP address or FQDN of the SSC LPAR
  LPAR_USER             Name of the Appliance user
  LPAR_PASSWORD         Password of the Appliance user
  BOOTDEVICE_ID         The device identifier to which the image is to be installed.
  IMAGE_NAME            Db2 Analytics Accelerator image to install

optional arguments:
  -h, --help            show this help message and exit
  --bootudid BOOT_UDID  UDID of a SCSI disk in 32 digit hexadecimal format.
                         This parameter is required if device is a SCSI disk.
  --licpath LICPATH     Path where license accept information has been stored
                        (default: lic).
  -v, --verbose         Increase output verbosity and logging
```

All of them need credentials and IP address of the SSC LPAR (both specified in the HMC for the SSC LPAR) and require that the LPAR is activated and accessible.

For specification of `BOOT_UDID for a` SCSI disk, here is a typical example:

- `device=0.0.1200`
- `bootudid=6005076241b5b0224000000021000028`

aqt-upload.py	Upload Accelerator image from local file system to an SSC LPAR.
aqt-license-accept.py	Accept software license for the Accelerator
aqt-first-time-setup.py	Start first time initialization of the Accelerator
aqt-quiesce.py	Quiesce Accelerator operation (before an update)
aqt-export-config.py	Export Accelerator configuration (before an update)
aqt-import-config.py	Import Accelerator configuration (after an update)
aqt-update-config.py	Update Accelerator configuration (during operation)
aqt-complete-update.py	Complete an update for a multiple node deployment
aqt-disk-fcp-list.py	Print a list of available FCP disk IDs

The `aqt-first-time-setup.py` script takes the filename of an Accelerator configuration JSON file as input. For multiple-node deployment, an additional JSON file with credentials for the data nodes in the cluster must be provided with the `--credentials_file` parameter. The layout of this file is as follows:

```
[
  { "name": "data1", "lpar": "LPAR1",
    "mgmt_ip": "10.1.0.1" , "uid": "admin", "pwd": "admin-password" },
  { "name": "data2", "lpar": "LPAR2",
    "mgmt_ip": "10.1.0.2" , "uid": "admin", "pwd": "admin-password" },
  { "name": "data3", "lpar": "LPAR3",
    "mgmt_ip": "10.1.0.3" , "uid": "admin", "pwd": "admin-password" },
  { "name": "data4", "lpar": "LPAR4",
    "mgmt_ip": "10.1.0.4" , "uid": "admin", "pwd": "admin-password" },
  { "name": "data5", "lpar": "LPAR5",
    "mgmt_ip": "10.1.0.5" , "uid": "admin", "pwd": "admin-password" }
]
```

The scripts can be used in a sequence to install and deploy an Accelerator without a web browser. Note that these scripts may start asynchronous operations, but poll the server for successful completion before they actually return to the caller. Therefore, a scripting sequence as outlined below can be easily implemented.

However, they can also be used in a mix of web based interaction and script execution. For example, use the script to upload the image but then continue configuration using the Web browser.

License Acceptance

Before working with the product, the software license has to be accepted. During web browser configuration this is enforced by a panel where IBM and non-IBM license texts are presented and users have to accept these before proceeding.

Similarly, script interaction can only continue after license acceptance. The script aqt-license-accept.py downloads the software license files to a local filesystem and requests confirmation from the user. After this, a file is written into a specified license path directory (command line parameter "`licpath`") which stores this information, along with date and timestamp. License acceptance is also noted in the LPAR so that subsequent scripts execution is enabled.

The license path directory information is relevant if a new installation of the same version should take place. Instead of manually accepting the license conditions again, the script could point to the license path directory where the previous accept notice has been stored.

It is suggested to use the same license path information with all scripts, and to consider license acceptance when automating deployments of new versions (keeping in mind that new versions need the manual license acceptance step in the first place).

Deployment and Installation Sequences

The package also comes with two sample shell scripts (sample-upload-initialize.sh and sample-update.sh) which orchestrate Python script invocation and also centralize configuration specifics for LPARs.

Additional sample scripts provide helper and checking functionality.

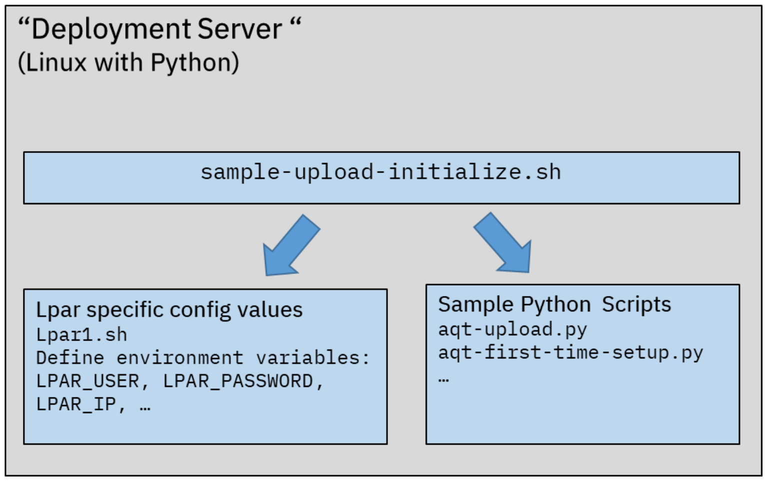
sample-upload-initialize.sh	Initial installation including image upload and configuration
sample-update.sh	Update an accelerator with a new image
sample-check-appliance-node-availability.sh	Check if a single node is available
sample-cluster-check-appliance-node-availability.sh	Check if all nodes in a cluster are available
sample-wait-operational.py	Wait until accelerator startup is complete
sample-check-installer.sh	Check if accelerator is in installer mode
sample-cluster-check-installer.sh	Check if all nodes of an accelerator are in installer mode
sample-reboot-to-installer.py	Reboot an accelerator into installer mode
sample-cluster-reboot-to-installer.sh	Reboot all nodes of an accelerator into installer mode

These scripts could be further customized and serve as a starting point for automated deployments.

As with a web-based configuration, the Accelerator definition is provided using a configuration file in json format. It has definitions for disks and networks and optionally also definitions for additional CPC/LPAR locations.

For a given FCP device, the script `aqt-disk-fcp-list` may be used to print a list of available disk IDs. This list can help to define the list of device IDs for data pool or runtime pool definition in the configuration file.

To use FCP/SCSI devices as the boot device during image upload, specify the FCP device ID as the `bootdeviceid` parameter and the UDID as `bootudid` parameter.



Sample Scripts, Support and Extensions

The scripts provided in this package are licensed to you ‘as-is’. They are not part of the licensed product and are also not supported in the same way. Future product capabilities (for example, advanced or changed network configurations) may not be reflected by the package. We welcome your feedback – you can provide it by opening an informational PMR or by writing to the Data Warehouse on System z Center of Excellence (dwhz@de.ibm.com).