Securely Send Files to SSDT for Support

While helping you trouble-shoot docker or application issues, the SSDT may ask you to send a file to the SSDT from your docker host. The SSDT Utils contains a script called `/ssdt/scripts/send.sh` which will transfer a file securely to an SSDT controlled server via HTTPS. The `init.sh` script (if sourced in `.bashrc` or linked in `profile.d`) provides a `send2ssdt` alias to easily invoke the script.

The script requires a file name parameter relative to the current working directory. The file must in contained in, or below, the current working directory. Alternatively, the file may be specified as a single dash (-) to read from standard input.

An optional second parameter may be specified to indicate the name, or short description, of the file (no spaces or special characters). If not specified, the name parameter defaults to the basename of the current directory:

```
/ssdt/scripts/send.sh {file} [name]
# or
send2ssdt {file} [name]
```

The file size is currently limited to 2GB.

The script does not notify the SSDT that a file has been sent. Therefore, you should communicate via other means when you send a file.

When transmitted to the SSDT, the file will be given a name prefixed by your docker host name and suffixed with a time stamp to ensure uniqueness.

Examples:

To send the log file from all containers, use:

```
#  docker-compose logs | /ssdt/scripts/send.sh -
```

To send the log from a single container (such as usaspp or uspsapp):

```
#  docker-compose logs usasapp | /ssdt/scripts/send.sh -
```

To send a log file from the current directory:

```
/ssdt/scripts/send.sh usasimport.log
# or:
send2ssdt usasimport.log
```

Backups can be sent, since binary files up to 2GB's are supported. To send a backup:

```
send2ssdt backup/nameofdistrict-usasdb.2016-03-11-15-20-10.backup.gz our_troubled_sampletown_db
```

To send the output of a command

```
/ssdt/scripts/info.sh | send2ssdt -
```

Large Files

Currently, files larger than 2GB cannot be sent via this procedure. These need to be split and each individual piece sent to the SSDT.

In order to send a file over 2GB, first the ITC needs to split the file and send the pieces individually, and then we need to put them back together. This has been successfully tested. The split command will split the file into pieces, each starting with xa (and then xb, etc. if there are several files created, but that should not happen with these backup files). The command provided splits the file into 1G chunks. Once they are split, then use the send script to send each individual file.

```
split -b 1G nameofbackupfile
```

SSDT note:

Once the files are received, the ssdt needs to put them back together. If possible, this should be done in the same location so the files will be sent to the processing area for automatic loading for troubleshooting (ssdt-docker-02:/data/upload.ssdt.io/uploads. In this case, the files resulting from the split command start with Fiscal-USxS-Pros-Backup_xa once uploaded. I downloaded and then combined them to re-create the backup file.

```
cat Fiscal-USxS-Prod2_backup_xa* > IRNdistrict-uspsdb.backup.gz
```

Example - in this case, we know they are usps files from Xenia. The IRN can be looked up here if it is not provided.

```
##Original files:
docker2_xe_xaa_2019-12-03T00-52-05.040Z
docker2_xe_xab_2019-12-03T00-52-36.533Z
docker2_xe_xac_2019-12-03T00-52-55.323Z

##Command to put them together:
cat docker2_xe_xa* > 044487NewPhil-uspsdb.2010-01-09.backup.gz
```

Technical Note:

The script uses the `ssdt-utils` image to execute the `curl` command to post the file to the SSDT's upload server at https://upload.ssdt.io/uploads. Since `curl` is running inside a temporary container, the file to be sent must be within or below your current working directory. It also means you do no have to have `curl` installed on your server.

This web site is "*write only*" so only SSDT personnel can access the transmitted files.

SSDT Internal Note:

Files uploaded by this process can be accessed by SSDT personnel at `upload.ssdt.io` (currently `ssdt-docker-02`) at `/data/uploads-sent` using `scp`. Note they used to be in /data/upload.ssdt.io/uploads. As of late 2019, a procedure was put in place to move backup from from uploads to the uploads-sent file. This is due to a mechanism put in place for restoring databases for support tickets. The naming convention for the files is:

`source_name_originalFileName`_timestamp

Where "source" is the hostname of the sending server (which may not be unique). "name" is the value entered in the second parameters or the parent directory of the file on the sending server. "originalFilename" is the file name on the sending system or dash (-) if from standard input.

Related articles

- Securely Send Files to SSDT for Support