EX Engage Connector Deployment Guide

Maintenance

12/14/2025

Contents

Maintenance

Once you have deployed and started the EX Engage Connector service, you can upgrade the EXEC services to a new version, monitor the status of containers, and access logs.

Upgrade EXEC Services

- Ensure that the new container version of EXEC service is available in the docker registry.

- Update the _TAG parameters in the .env file with the new version of the EXEC service. You can update all or individual EXEC services:

```
EXCS_TAG=100.0.100.0001
EXAS_TAG=100.0.100.0001
EXCP_1_TAG=100.0.100.0001
```

- Run `./bootstrap.py --action update`. This command will download the .env and docker-compose files from the remote EXEC Service VMs.

    - The script compares the new **.env** file parameters and **docker-compose** files with the files present in remote EXEC service VMs.

    - If the files differ, the bootstrap script will generate backups of the **remote.env** and **docker-compose** files in the respective service VMs under the backup directory.

    - The backed-up filenames will be appended with the current date in the format of "_<ddmmyyyy>". For instance, .env_23052023 and docker-compose-EXAS.yml_23052023.

    - Update parameters downloaded remote .env file to add any newly added parameters or remove any deprecated parameters, and reupload the .env file into the respective service VMs.

    - Update parameters downloaded remote docker-compose files to add any newly added parameters or remove any deprecated parameters, and reupload the docker-compose file into the respective service VMs

- Restart the exec services using start/stop or restart commands for the changes to take effect.

You can also update individual services by using the `--service` option as shown below:

- `./bootstrap.py --action update --service=exas` - Update the EXAS service.

- `./bootstrap.py --action update --service=excs` - Update the EXCS service.

- `./bootstrap.py --action update --service=excp --pair 1` - Update the EXCP service for excp pair 1.

- `./bootstrap.py --action update --service=excp --pair 1 --host 1` -Update the EXCp service running on VM host 1 belonging to excp pair 1.

Rollback EXEC Services

After upgrading the EXEC service to a newer version or updating any parameters and if the service is not running as expected then the admin can roll back the service.

- Run `./bootstrap.py --action rollback`. The bootstrap script will find the latest backup of the EXEC service taken during the update operation.
    - The script will copy the .env and docker-composer files from the backup folder into the **exec** directory.
- Restart the EXEC services using start/stop or restart commands for the changes to take effect.

You can also rollback individual services by using the --service option as shown below:

- `./bootstrap.py --action rollback --service exas` - Roll back .env and docker-compose file to the pre-update state for the exas service.
- `./bootstrap.py --action rollback --service excs` - Roll back .env and docker-compose file to the pre-update state for the excs service.
- `./bootstrap.py --action rollback --service excp --pair 1` - Roll back .env and docker-compose file to the pre-update state for the for excp pair 1
- `./bootstrap.py --action rollback --service excp --pair 1 --host 1` - Roll back .env and docker-compose file to the pre-update state for the host 1 belonging to excp pair 1

Cleaning up recordings not injected to GC

In some scenarios, recordings may fail to be injected into the GC. Those recordings are accumulated in two places:

- Recording MCP hosts: in the directory configured in the MCP option *msml.record.irrecoverablerecordpostdir*.
- WebDAV directory where MCP uploads the recordings.

It is recommended that the customers automate a cleanup of those storages.