How to manually use FTD 1.0.2 Device Package

https://github.com/cisco-security/FMC-REST-API-scripts

Scripts: config-ftd-routed.py unconfig-ftd-routed.py ftd-reg.pl

What we need before we start:

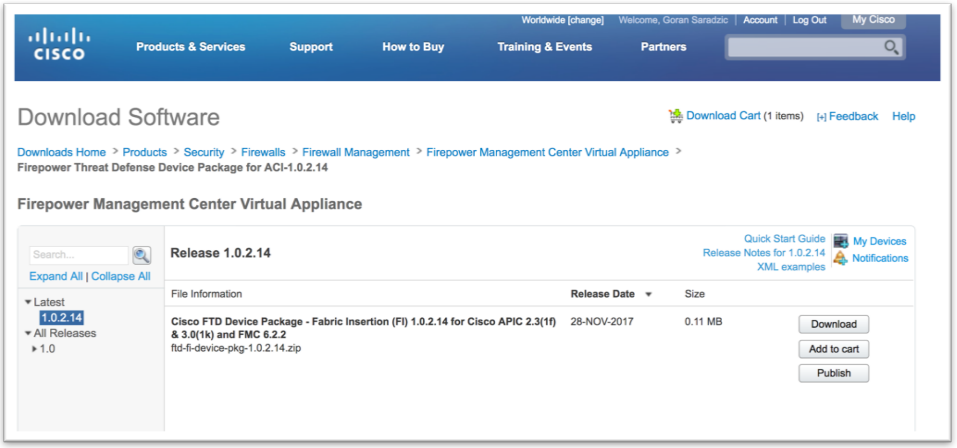
1. Scripting Host that you can use to program FMC (I suggest Linux)
2. Install Python interpreter, 2.7.3 or later
3. Download FTD 1.0.2 Device Package for ACI from Cisco.com
4. Download Github config and unconfig python scripts
5. Create manual-devpkg directory (choose the name as appropriate)

Prepare FMC and FTD for Orchestration:

1. Setup FMC with necessary licenses (use demo with evaluation mode)
2. Ensure connectivity between FMC and FTD
3. You can register FTD(s) via FMC GUI, or you can modify and use the script ftd-reg.pl
4. FTD HA must be pre-configured manually via FMC
5. Pre-configure FMC Policy and Rule(s) you want FTD interfaces to use
6. Create a separate admin account for API communication (i.e. apiuser)

The python scripts included here are stand-alone scripts that use/reference procedures build into FTD device package, version 1.0.2. This package is available on Cisco.com.

FTD 1.0.2 FI Device Package Posted



Scripts configure and unconfigure FTD in Routed (L3FW) Firewall mode, with two interfaces. FTD device package should be unzipped into ftd-fi directory (this is the default location for the zip). Script can be placed in the same directory as ftd-fi, and it will reference the procedure within ftd-fi. You will need to modify this path in the script:

```
import sys
sys.path.insert(0, '/home/user/manual-devpkg/ftd-fi')
```

Scripts include three basic sections: setting parameters, the device spec, and device configuration sections. Last two sections are based on what APIC would use to configure an FTD device in a managed service graph. In our case, we can use a stand-alone python script to apply the same configuration instead of using APIC.

The sample scripts are created by watching an APIC debug.log in FTD device package logs directory (/data/devicescript/CISCO.FTD_FI.1.0/logs/debug.log). Captured json device spec and device config sections were sanitized and updated with parameters. Config creation script uses serviceAudit procedure built into device package.

To run the script, you must add python interpreter in front:

```
python config-ftd-routed.py
```

Enjoy!