TIER Instrumentation ("Beacon") Technical Specification

Version 1.0

Version 1.0 of the TIER Instrumentation specification calls for each TIER container to send a daily “beacon.” This beacon will contain only product (Shibboleth-IdP/Grouper/COmanage), product version (3.3.1/2.3/2.0), and a TIER release identifier.* This reporting data gives crucial insights to the TIER investor community regarding how and where the releases are being deployed so that adoption of deliverables can be directly measured.

The format of the beacon message will be JSON, which is already in use in other components and simple to implement. The message will be sent via HTTP/REST on a non-standard port (5001) to an AWS “collector” VM operated by TIER (collector.testbed.tier.internet2.edu). An example of the dashboard for the collector can be seen below. Each container will be configured at final build time with a random time for these daily messages to be sent (via cron). That random time will be between midnight and 4am. This is the format of the incoming message:

```
{
"host" : "IP Addr"
"msgType" : "TIERBEACON",
"msgName" : "TIER",
 "msgVersion" : "1.0",
"tbProduct" : "<COMPONENT_NAME>",
"tbProductVersion" : "<PRODUCT_VERSION>",
"tbTIERRelease" : "<TIER_RELEASE>",
"tbMaintainer" : "<MAINTAINER>"
}
```

tbProduct: Refers to component, or product.
tbProductVersion: Product Version
tbTIERRelease: TIER Release ID
***tbMaintainer**: A variable currently present in the containers to represent “maintainer”, but not yet used to convey meaningful information. In the future, it will likely be changed from “my” to “TIER”.

Within each container, the beacon will be implemented as a daily cron job, executed by crond. The job will call a script /usr/bin/sendtierbeacon.sh. That script will check whether an opt-out environment variable is present and, if not, will send the TIER beacon message via the curl command (using HTTP-POST). A timestamped confirmation message will also be written to a logfile at /var/log/cron.log within each container.

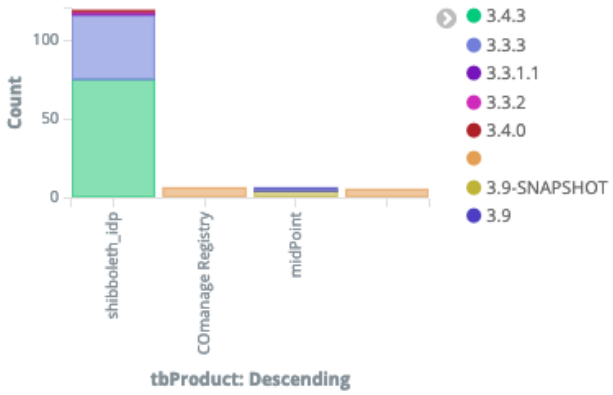
Crond will be started by a new script - /usr/bin/startup.sh - that will be the new target of the local Dockerfile’s CMD directive (replacing the previous target). The startup.sh script will do the following (it only runs once at build time):

Call the script /opt/tier/setenv.sh, which writes the environment variables needed by the beacon script to a local file - /opt/tier/env.bash. This is because cron jobs run in an isolated environment without these variables and docker provides no other local file to use for sourcing them, so the startup script writes them to a local file for later use by the cron job (sendtierbeacon.sh).

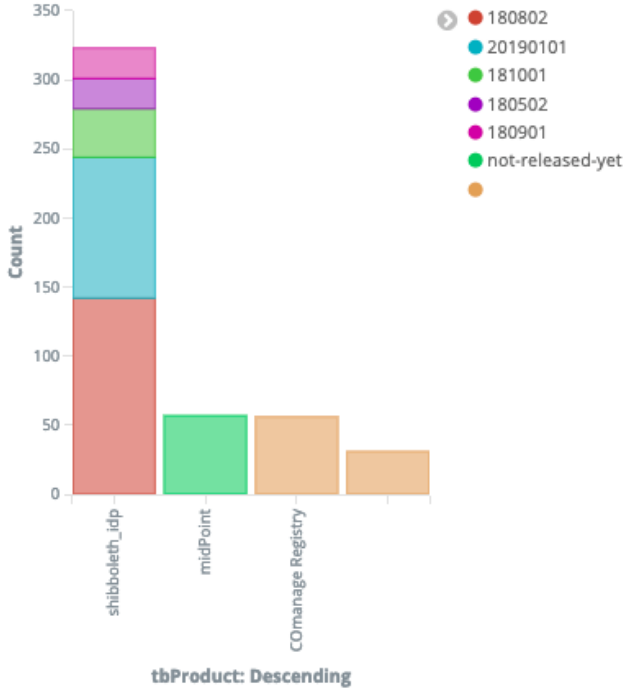
- Create a crontab file for the needed job set to execute at a random time between midnight and 4am. This file is written to /opt/tier/tier-cron
- Add that new crontab file to the root user’s crontab
- Create the cron log file - /var/log/cron.log
- Start crond
- Run the command line from the original CMD directive in the local Dockerfile

Example Dashboard

TIER Components by Product Version



TIER Components by Release



TIER_Component_Count