Heureka Appliance
Installation Guide

CONTENTS

Background

The Heureka appliance runs on Fedora Core OS (FCOS). It might be very different from distributions you have worked with previously. This documentation applies to appliance version 1500.0. The version number of an appliance can be viewed with:

```
grep ^version= /home/core/config/VERSION
```

Please read all instructions thoroughly and follow them step by step.

Installation

Import OVA File

1.1 Import the provided .ova into a virtualization environment. Set the CPU and memory settings to the appropriate values for the targeted data and performance.

- Recommended Settings: 16 core CPU, 32 GiB RAM
- Minimum Settings: 8 core CPU, 8 GiB RAM

1.2 Configure the virtual machine to have both reserved memory and cpu's to minimize performance impact from other virtual machines.

Configure Appliance

2.1 Start up the virtual machine, log into the command line console with the default credentials:

- Username: `core`
- Password: `core`

2.2 You will be prompted to set a new password for the core user. Ensure you save this password somewhere safe where it cannot be lost such as an encrypted password manager. If you lock yourself out and have console access, see the Password Recovery instructions in this document.

2.3 Optionally, copy ssh keys to the appliance to allow for ssh-key based authentication.

```
ssh-copy-id core@<Appliance DNS Name>
```

2.4 The system is already configured for DHCP. If you require a hardcoded static IP, then you need to create a file in `/etc/NetworkManager/system-connections` with your network adapter settings. More information can be found here.

2.5 At this point is recommended that you switch from the VMware console to ssh for the remainder of the steps.

```
ssh core@<Appliance DNS Name>
```

2.6 If the `ssl.pem` file has not been created yet, follow the instructions found at SSL Certificate Format in this document to create it.

2.7 Copy the `ssl.pem` file to this location on the appliance: `/home/core/config/haproxy/ssl.pem`. This can be accomplished like so with `scp` and the desired certificate named `ssl.pem`:

```
scp ssl.pem core@<Appliance DNS Name>:/home/core/config/haproxy/ssl.pem
```

2.8 Run the appliance configuration script:

```
/home/core/scripts/appliance_setup.sh
```

Note: This script is designed to be run anytime you wish to make a configuration change.

2.9 Record the password for the sftp service and the database from the output, `HEUREKA_SFTP_PASSWORD` and `HEUREKA_CORE_DATABASE_PASSWORD`. Keep these passwords in a secure location.

2.10 Restart the appliance:

```
sudo reboot now
```

2.11 Wait for the VM to come back up, either by watching the console or waiting for ssh connectivity to become available, then visit the web console at https://<appliance_dns_name>. It should come up within a minute of startup.

2.12 Upload the license key issued to you for this appliance

2.13 Contact Heureka to arrange for your initial account username and password to be added to the appliance

2.14 Once you have logged in, there should be an example endpoint running, run a search against the example endpoint to verify the application is working as expected. If you wish to shut off and disable the example endpoint, run the following on the command line:

```
/home/core/scripts/after_testing.sh
```

Install an Endpoint

Detailed documentation on endpoint installations will be provided by a Heureka representative. The installation packages and configuration files are available on the sftp share. Connect to `sftp` with the following information:

- Address: `sftp://<Appliance DNS Name>`
- Port: `2222`
- Username: `ftp_user`
- Password: <sftp password recorded during installation>
    - This password is also available after install at `/home/core/config/generated/sftp_password.env`

The installation packages are in the installers directory and the configuration files are in the `endpoint_config_templates` directory.

Updating the Appliance

These instructions apply to appliances already running 1492.0 or later. If you are looking to update an appliance from a version earlier than 1492.0 to version 1492.0 or later, you will need to consult the Heureka Appliance Migration Guide for instructions.

Update FCOS

1.1 FCOS requires internet access during the upgrade, specifically access to `*.fedoraproject.org`, this can be tested by making sure you eventually get a `200` response code after redirects:

```
curl -I -L https://coreos.fedoraproject.org
```

1.2 Run the update script. If there are updates that were applied, the system will automatically reboot.

```
sudo /home/core/scripts/update_fcos.sh
```

Note: If you are running an older version of FCOS, the updates may be trying to contact an update server that is no longer in commission. To fix this, run the update script after updating the Heureka Software. The newer version of the software will contact the latest update server.

Note: Depending on how many updates are available, it may take several update/reboot cycles to fully update the system. If the update script reboots your system, just keep running the `update_fcos.sh` script until you see:

[SUCCESS] No updates found

1.3 Optional: If you want to be able to restore to this point in the future, this would be a good time to take a snapshot of your current VM.

Update Heureka Software

2.1 Obtain the upgrade archive file from Heureka. The filename of this file will be `appliance-<version>.tar.xz`.

2.2 Copy the upgrade archive to the appliance and place it in the `/home/core` directory

```
scp appliance-<version>.tar.xz core@<Appliance DNS Name>:/home/core
```

2.3 Log into the appliance on the console as the `core` user

```
ssh core@<Appliance DNS Name>
```

2.4 Extract the archive

```
cd /home/core
tar xvf appliance-<version>.tar.xz
```

2.5 Run the installation script to install the new components:

```
/home/core/dist/scripts/install/install.sh
```

Note: If you see the message:

```
Database upgrade necessary. Please run db_upgrade.sh and then rerun
this script.
```

Then perform the actions in the section below entitled "Database Upgrade", reperform step 2.5, and then continue to step 2.6.

2.6 Run the configuration script to update the configuration as necessary and start the new components:

```
/home/core/scripts/appliance_setup.sh
```

Database Upgrade

Sometimes we need to upgrade the database to a newer version which requires data to be ported from the old database to the new database. When this is needed, the `install.sh` script will inform you that a database upgrade is needed before you can continue.

Since we are porting the database, you will need to have enough free disk space to duplicate your data. The db_upgrade.sh will do this estimation, but if you want to check for yourself, you can get the number of MB free with this command:

```
df --block-size=M --output="avail" /home/core/volumes/postgres
```

And the amount of space used by your current database with this command:

```
sudo du -sm /home/core/volumes/postgres
```

If the amount free is less than the size of your current database, you will have to free up disk space by deleing unneeded collection data or add more space to your appliance to accommodate the upgrade. To add more disk space, see "Expanding the Disk" in the FAQ at the end of this document.

To upgrade the database, run the upgrade script and follow the prompts:

```
/home/core/dist/scripts/db_upgrade.sh
```

Once complete, your old database data will be in the `/home/core/volumes/postgres-<old version number>` directory. If you ever need to roll back the change, you can simply rename the old data directory back to the original. If the upgrade is successful and the appliance is running properly, you can safely remove this old directory.

SSL Certificates

Format of the Certificate File

The appliance uses a certificate container in PEM format named `ssl.pem`. This certificate must be provisioned from a public certificate authority that has been verified to be in the list of trusted certificate authorities of the Heureka endpoint software.

You must provision a certificate that matches the specific domain name of the appliance or a wildcard certificate for your domain and then obtain the private key and any other certificates in the trust chain up to the root certificate. This documentation refers to all these certificates separately, the file names and trust chain may be different depending on what certificate authority is used.

For example, for certificates from authorities like Comodo or GoDaddy, you create your ssl `.key` and `.csr` (Certificate Signing Request) file. You then submit the `.csrto` the authority to confirm and get signed. Once signed you will usually receive two `.crt` files. One of them is your domain's certificate file and the other is an authority chain consisting of the root certificate along with any other intermediate certificates.

Open a text editor (such as wordpad or vim) and paste the entire body of each certificate into one text file in the following order:

- The Private Key - `appliance_domain_name.key`
    - This key must not be encrypted with a passphrase.
- The Primary Certificate - `your_domain_name.crt`
- The Intermediate Certificate - `example_ca.crt`
- The Root Certificate - `trusted_root.crt`
    - This must be a public certificate authority.

- The intermediate and root certificates may be sent to by your provider in a single file

Make sure to include the beginning and end tags on each certificate. The result should look like this:

```
-----BEGIN RSA PRIVATE KEY-----
(Private Key: appliance_domain_name.key)
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
(Primary SSL certificate: your_domain_name.crt)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
(Intermediate certificate: example_ca.crt)
-----END CERTIFICATE----
-----BEGIN CERTIFICATE-----
(Root certificate: trusted_root.crt)
-----END CERTIFICATE-----
```

Here is what it would look like when done. Note: The entries have been shortened for this example. Your key and certificates should be much longer.

```
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDFB/A4MB9QGP0e
7KZd63iwgI36vbkXQg+vWPZ7FQ==
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIEhjCCA26gAwIBAgIBATANBgkqhkiG9w0BAQsFADCBpzELMAkGA1UEBhMCVVMx
VVck4ak8y84Xmw==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEPjCCAyagAwIBAgIUXJrvv5CWlRfD77HN03mHpO5q9ogwDQYJKoZIhvcNAQEL
LwOReMjokiYlEFg83ePRw6AftMAY3ZF9ET/Fzl89LMkXdw==
----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEczCCA1ugAwIBAgIQJiEb9SrrUbAL+p/djTbanjANBgkqhkiG9w0BAQUFADBv
F9UCjiJE/JDErEafSDkqqQ5qlwYgPz9a+qyVI3rchsjq2Tt4jCZE
----END CERTIFICATE-----
```

Save the combined file as ssl.pem. The .pem file is now ready to use.

Self-Signed Certificates

It is possible to generate a self-signed certificate for use with an appliance with the caveat that the certificate is installed in the JRE truststore on each endpoint and indexing appliance. Please contact Heureka support for the exact process to generate a valid self-signed certificate.

Once you have a valid `.pem` file, that file needs to be converted to a `.der` file. Use the following command to convert your ssl.pem to ssl.der:

```
openssl x509 -outform der -in ssl.pem -out ssl.der
```

Once you have the .der file, you can import this into the cacerts file of the Java Runtime Environment (JRE) used by the endpoint software. Look in the directory the Heureka endpoint software was installed into (e.g. `/usr/local/opt/heureka-agent` on MacOS). From that directory you will find the truststore file at `jre/lib/security/cacerts`. To add the `.der` file, run the following command:

```
jre/bin/keytool -import -alias appliance-internal \
-keystore jre/lib/security/cacerts -file <path_to_your_ssl_der>/ssl.der
```

The default password for the `cacert` file is `changeit`. When asked if you trust this certificate, answer yes.
Once the `.der` has been added, it should be able to securely communicate with any appliance with the matching `ssl.pem`.

In addition to adding the .der file to the JRE truststore, you should also add entries to the system configuration files on the endpoint (e.g. `/etc/hosts`) so the domain name you used in your self-signed cert resolves to the IP address of the appliance.

Once everything is in place, restart the indexing service on the endpoint so the new configuration takes effect.

Verifying the SSL Certificate

After creating the `ssl.pem` file, putting it into place, and completing the install process, the certificate is put into use and can verified to be fully working. If the certificate file has been created successfully, the `heureka-haproxy` service will be up and running. To check the status of this service use this command:

```
sudo systemctl status heureka-haproxy
```

If it has encountered a problem, there is a good possibility the `heureka-haproxy` is in an auto-restart loop. This can be checked by looking at the history of services:

```
sudo journalctl _PID=1 -e -n 100 | grep "heureka-.*\.service:" \
| tail -n 20
```

If it appears the `heureka-haproxy` service is restarting over and over, take a look at its logs with the following command:

```
sudo journalctl -u heureka-haproxy -e -o cat
```

There should be an error message in the log that can be used to troubleshoot the created certificate.

Updating or Changing the SSL Certificate

1.1 Create the `ssl.pem` file described in SSL Certificate Format.

1.2 Place the new `ssl.pem` file onto the appliance in this location: `/home/core/config/haproxy/ssl.pem`.

1.3 Restart the service responsible for SSL termination with this command to pick up the change:

```
sudo systemctl restart heureka-haproxy
```

SAML SSO Setup

Using Okta

If you are not using Okta, you can skip to step 2.1.

1.1 When running the appliance configuration script (step 2.8 of the installation instructions), ensure you use your application DNS name (not the IP address) as the root URL for the appliance.

1.2 After the appliance is setup is complete, log into the web interface as an administrator and enable SCIM via the System Configuration page. Note the Bearer Token that is generated.

1.3 Add the Heureka Platform Connector from the Okta Integration Network to your Okta configuration and configure it to use the SCIM bearer token.

Configure Appliance to use SAML

2.1 If you are using file-based metadata, copy the metadata file to configuration directory `/home/core/volumes/config-data` on the appliance.

2.1 From the command line of the appliance, re-run the application setup script

```
/home/core/scripts/application_setup.sh
```

2.2 When prompted whether you want to use SAML single sign-on, type `true`

2.3 If you provided file-based metadata, select your metadata file from the list. If you don't want to use a metadata file, just leave it blank and you will be prompted for a URL instead. If you did not prove file-based metadata, enter the metadata URL from your SAML-enabled application.

2.4 Complete the setup and allow the services to restart

Disabling Single Sign-On

If you ever with to disable SSO, do the following:

1.1 From the command line of the appliance, re-run the application setup script

```
/home/core/scripts/application_setup.sh
```

1.2 When prompted whether you want to use SAML single sign-on, type `false`

1.3 Complete the setup and allow the services to restart

Extending Collect Storage

Since the collection point in the appliance is a directory on the appliance (`/var/home/core/volumes/endpoint-api/ftp_root`), that directory can be mounted with an additional disk on the virtual machine or with network storage.

Move Files (Optional)

Before starting, make sure you have moved any existing collections to a temporary location if you need access to these files. Once the collection point is mounted, the existing files will not be available via the FTP server.

1.1 Make temp dir

```
sudo mkdir -p /tmp/ftp_root
```

1.2 Move existing files

```
mv /var/home/core/volumes/endpoint-api/ftp_root/* /tmp/ftp_root
```

Storage Option 1: Adding an attached drive

2.1 If not done already, create and attach a new disk to the virtual machine

2.2 Identify the disk to format

```
sudo lsblk
```

2.3 Identify the desired device and set it to a variable. For ***example***:

```
DEVICE=/dev/sdb
```

2.4 Format the device

```
sudo mkfs.ext4 -m 0 -F -E \
lazy_itable_init=0,lazy_journal_init=0,discard $DEVICE
```

2.5 Determine UUID of disk

```
UUID=$(sudo blkid -s UUID -o value $DEVICE)
```

2.6 Backup fstab

```
sudo cp /etc/fstab /etc/fstab.orig
```

2.7 Add a new fstab entry for the new disk

```
echo "UUID=$UUID /var/home/core/volumes/endpoint-api/ftp_root ext4
discard,defaults,nofail 0 0" | sudo tee -a /etc/fstab
```

2.8 Mount all the disks

```
sudo mount -av
```

Storage Option 2: Add network storage

You will need to have a network location available, reachable from the virtual machine. These instructions are for mounting a SMB file share using the CIFS driver. Other types of mounts, such as NFS or SSHFS, are possible. If you need help configuring the appliance to mount these types of drives, please contact Heureka for support.

3.1 Determine the UNC network path to your drive and set it to a variable. For ***example***:

```
DEVICE=//0.0.0.0/ftp_storage
```

3.2 If credentials are necessary to mount the network file share, create and place the credentials file for the mount at `/home/core/.smbcredentials`. The format of the file contents is:

```
domain=<domain>
username=<username>
password=<password>
```

Set the file to be assesible only to the core user:

```
chmod 600 /home/core/.smbcredentials
```

Set the credentials file to a variable:

```
CREDENTIALS=/home/core/.smbcredentials
```

3.3 Backup fstab

```
sudo cp /etc/fstab /etc/fstab.orig
```

3.4 Add fstab entry. If credentials were necessary:

```
echo "$DEVICE /var/home/core/volumes/endpoint-api/ftp_root cifs
noperm,iocharset=utf8,rw,credentials=$CREDENTIALS 0 0" \
| sudo tee -a /etc/fstab
```

If credentials were not necessary:

```
echo "$DEVICE /var/home/core/volumes/endpoint-api/ftp_root cifs
noperm,iocharset=utf8,rw 0 0" | sudo tee -a /etc/fstab
```

3.5 Mount all the disks

```
sudo mount -av
```

Move Files Back (Optional)

Only do this after the storage option has successfully mounted.

4.1 Move files back after mounting

```
sudo mv /tmp/ftp_root/* /var/home/core/volumes/endpoint-api/ftp_root
```

4.2 Cleanup the temp directory

```
sudo rm -rf /tmp/ftp_root
```

Finish

5.1 Make sure `ftp_root` directory is writable by the `heureka_run` user. The following is one way to do it:

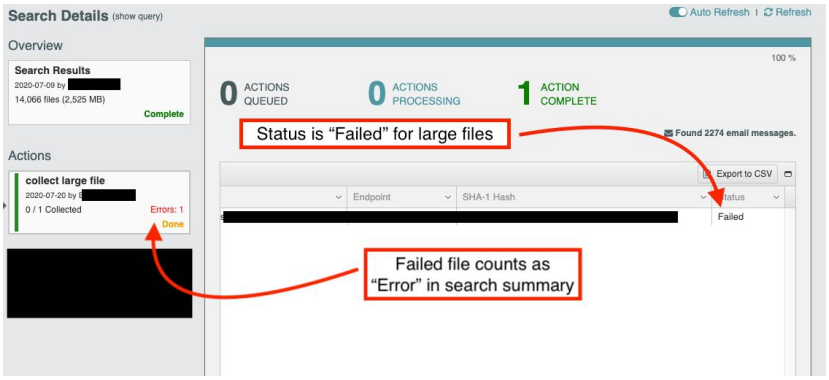
```
sudo chmod 777 /var/home/core/volumes/endpoint-api/ftp_root
```

5.2 Make sure to reboot after finishing the above steps.

```
sudo reboot now
```

Large File Collection

Out of the box, the Heureka Platform has a safety measure regarding what are deemed large files during collection – essentially so you don't collect the world. A default size of 100 MB is the limit for any single file to be collected. These files can be identified; if you have tried to collect a file over the limit the status will be "Failed" and will count as an "Error" on the search summary.



A configuration feature allows you to set this size higher (or lower, even) to suit your organization's needs. There are two items to configure: 1. Endpoint API, and 2. Endpoint Service

Configure Endpoint API

To configure the Endpoint API, you will need to add a line to the environments file on the appliance virtual machine. You will need administrative access to this virtual machine.

1.1 Locate the environments file on the appliance:

```
/home/core/config/environment_file.env
```

1.2 Edit and save the file to add the following line, replacing `<SIZE>` with the upper limit you would like to use for collections. *{size}* is in bytes – e.g. 1000000000 to allow for files to be collected up to 1 GB. To allow unlimited size, use some large value (e.g. max size for Long data type: 9223372036854775807, which more than 9000 petabytes!)

```
HEUREKA_ENDPOINT_API_MAX_UPLOAD_SIZE=<SIZE>
```

1.3 Restart the Endpoint API service in order to reload the configuration.

```
sudo systemctl restart
```

Configure Endpoint Service

To configure the Endpoint Service, you will need to add a line to the configuration file for every endpoint that should adhere to the collection limit adjustment made above.

2.1 Locate the `agent.config` file for your endpoint's OS:

Linux (all):
`/var/lib/indexing-service/conf/agent.config`

macOS:
`/usr/local/opt/Heureka-agent/conf/agent.config`

Windows:
`C:/Program Files/Indexing Service/conf/agent.config`

2.2 Edit and save the file by adding the following line, replacing `<SIZE>` with the upper limit you would like to use for collections on this endpoint. Size is in bytes – e.g. 1000000000 to allow for files to be collected up to 1 GB. To allow unlimited size, use the value -1.

```
endpoint.upload.size.limit=<SIZE>
```

2.3 Restart the Endpoint's indexing service (named "Indexing Service" or "indexing-service", depending on OS) according to your OS service manager or restart the machine.

FAQ

Passwords

Generated passwords are stored in the `/home/core/config/generated` directory.

Troubleshooting

A support file for sending to Heureka for analysis can be generated with this command:

```
/home/core/scripts/generate_support_file.sh
```

Making configuration changes

If settings need to be changed that were entered during initial appliance configuration, re-run the setup script to go back through the prompts, change values as necessary. After the prompts have been filled in the services will restart automatically to pick up the changed values.

```
/home/core/scripts/appliance_setup.sh
```

Password Recovery

If you have forgotten your command line password to the appliance and have no SSH keys you can use to log in, you will have to manually reset the password via the console:

1.1 Reboot the virtual machine

1.2 At the BIOS screen, repeatedly press the up and down arrow keys to interrupt the GRUB bootloader

1.3 Highlight the image labeled `ostree:0` and press `e` to edit

1.4 Append the following to the line that starts with `linux`

```
init=/bin/sh console=tty0
```

1.5 Press `CTRL-X` to boot the system

1.6 This will boot the system and drop you into a root shell

1.7 Load the SELinux policy

```
/sbin/load_policy -i
```

1.8 Reset the core user password

```
passwd core
```

1.9 Restart the system

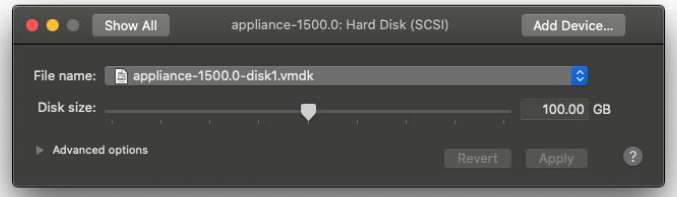
```
/sbin/reboot -f
```

Expanding the disk

If you are running low on disk space in the `/sysroot` partition and wish to expand it, do the following:

1.1 Shutdown the VM

```
sudo shutdown now
```



1.2 Expand the disk in your VM manager

1.3 Start the VM

1.4 Verify that the disk expanded using:

```
df -h
```

1.5 If the disk does not automatically expand, run the following command:

```
sudo /home/core/scripts/expand_disk.sh
```

If this command fails because `/sysroot` is mounted as read-only, use the following commands to remount the drive and run the `expand_disk.sh` script again:

```
sudo unshare --mount
mount -o remount,rw /sysroot
/home/core/scripts/expand_disk.sh
exit
```

Your disk should now be expanded and can be verified using the `df -h` command.

Heureka is a technical leader in endpoint search, identify and classification software. Our goal is to bring order to unstructured data by identifying risk while helping you realize the value of unstructured data across all endpoints.

Heureka, Inc
PH. 800.310.0981
info@heurekasoftware.com