Shibboleth-IdP Release 17040

Introduction

The packaged TIER Shibboleth-IdP release is a Docker container-based implementation of the Shibboleth IdP. The release consists of 2 Docker containers for the Shibboleth IdP and one for a load balancer on a Centos Linux Virtual Machine image that is used to configure, build, and run the containers. Release 17040 of the TIER-Shibboleth-IdP appliance contains the following components:

- CentOS 7.3.1611
- Shibboleth IdP: 3.3.1
- Java: 8u131-b11
- Tomcat: 8.0.43
- HAProxy: 1.7.5

At the current point in time, TIER Release 17040, we have not written documentation on how to use the containers outside of the VM environment. This release is primarily available for use with Oracle VirtualBox, though an Amazon AMI is available (please contact us).

A few words on VirtualBox:

- If you are not familiar with VirtualBox, you can read the documentation and download the software from Oracle's web site.
- Once VirtualBox is installed and running, you import the .ova distribution image using the File / Import Appliance function.
- The default network connection for the Shibboleth IdP packaged Virtual Machine is *Bridged*. With a bridged network connection, the VM will use dhcp to obtain its IP address from your local network. If you have a network registration process in place on your campus network, you may need to register the MAC address of your VM before you obtain an IP address. You can change the VirtualBox network configuration to NAT mode to look at the VM and its components this way but it is not recommended for general testing. *Remember, if you are on a public network, the VM will be exposed to the world and we publish the password on this web site.* Note that Virtual Box bridge mode can not work with many wireless network adapters since the don't support promiscuous mode. A wired network connection is generally better for use with Bridge Mode. Note that is *possible* to do some testing in NAT mode, but involves the inclusion of port numbers into the process. This works e.g. https://127.0.0.1:8443 but remember that only one person can use a URL at a time in the TIER Tested.
- If you choose to connect this VM to the TIER Testbed, upon success you will see a set of attributes supplied by your Shibboleth IdP VM being displayed by the TIER testbed Shibboleth SP.

Installation

For quick deployment of the TIER Shibboleth IdP Release:

Step 1 – boot the VM (see 'A few words on VirtualBox' above)

Step 2 – login to the VM via SSH (using the local IP address on the VM) using a username of *shibboleth* and a password of *shibboleth*.

Step 3 – Do a little housecleaning

- *sudo /bin/bash*
- *passwd shibboleth*
- *passwd centos*

Step 4 – *cd work* (/home/shibboleth/work)

Step 5 – run *./setup.sh* (you'll be asked 7 questions - see below**)

Step 6 – run *./build.sh*

Then, wait for a couple of minutes and your IdP virtual cluster should be up and running (verify with 'docker ps')...

If your installation was successful, the Shibboleth status page should display the normal information (version, uptime) from within the VM:
curl -k https://127.0.0.1/idp/status

** The setup.sh script will ask you 7 questions (in addition to asking you to agree to Java):

1. FQDN (or IP address) - this is used to build the SAML entityID
    a. Example: idp.example.org
2. Scope (for attributes)
    a. Example: example.org
3. LDAP URL
    a. Example: ldap://ldap.example.org
4. LDAP Base DN
    a. Example: ou=people,dc=example,dc=org
5. LDAP Svc Acct DN
    a. Example: uid=IdPServiceAcct,ou=Service Accounts,dc=example,dc=org
6. LDAP Svc Acct password
    a. Example: mySecretPassword
7. Whether to burn/copy the config or mount it***
    a. Example: burn

The VM Environment

The VM is designed to support TIER's Docker distribution of the Shibboleth IdP. The VM can be used to both build Docker containers for use elsewhere and/or be a fully operational environment that is also used to run the containers. For example, a VM could sit behind a campus load balancer and be one instance of the campus Shibboleth deployment. The instructions on this page are all geared towards using the VM as your production environment. Those wanting to use the Docker images directly can review the VM to see how to configure and use them.

The VM contains a full Shibboleth IdP configuration tree located at */home/shibboleth/build/shibboleth_idp/root*. The existing scripting makes changes to this tree but these changes are limited to the initial setup process. You make operational changes to the Shibboleth IdP configuration by editing the configuration tree in the normal manner (see normal Shibboleth documentation) and running the ~/bin/*rebuild.sh* command described above. Typically (unless you chose to mount the config instead of "burn" it), all Shibboleth IdP configuration information is built into the container, you create a new container to change the configuration.

The operational VM runs three containers: two instances of the Shibboleth IdP and a copy of haproxy. The haproxy container receives user requests and load balances across the two Shibboleth instances. The ~/bin/*restart.sh* script leverages ansible to automate the process of making Shibboleth IdP configuration changes without downtime. The script works by quiescing traffic to one Shibboleth container, stopping the container, swapping in a new previously built Docker image with the updated configuration, and then resuming traffic to the new container. This process is then repeated with the second Shibboleth container.

Other VM Notes:

1. Virtual Box Console Window
   The console window provided by Virtual box is not over friendly. The VM accepts SSH on port 22 when the VirtualBox network adapter is configured for a Bridged Adapter and on port 2222 when the VirtualBox network adapter is configured for NAT. You will find that a SSH connection with a decent terminal emulator is much easier to use.
2. User Shibboleth does have sudo capability (remember to change the password).
3. User Shibboleth Home Directory
   The shibboleth user's home directory contains the build process and access to make configuration changes.
4. Debugging Problems
   The key to debugging issues with Shibboleth remains with the log data, By default, log data is automatically mapped to the host VM. In /home /shibboleth/logs you will find access to the usual Shibboleth and Tomcat logs.

NOTES

- You can change any of the config/UI files in the config tree (located at /home/shibboleth/build/shibboleth_idp/root/) to suit your deployment needs (most are in 'conf' or 'views'),
- At that point, if you chose to mount your configuration, you're done (you may want to restart tomcat inside the container). If your configuration is burned into the containers, read on...
- The following command rebuilds the containers with the new configuration (not needed if you chose the "mount" option)
  - *bin/rebuild.sh*
- The following command does a rolling restart of the two IdPs (should be outage-free) (not needed if you chose the "mount" option)
  - *bin/restart.sh*
- Basic config backup/restore is provided by scripts in the /home/shibboleth/bin directory.
  - *shibb-config-snap.sh* - will create an archive of the current shibb config (the 'root' directory) and save it to a timestamped zip filename in /home/shibboleth (no command options are required)
  - *shibb-config-restore.sh* - can restore a filename that you specify or it can also set your config to the default release config or to the default TIER Testbed config. (run it with no parameters to display command options)
    - To set your config to the default **testbed** config, run "*shibb-config-restore -t*".
    - To set your config to the default **release** config, run "*shibb-config-restore -r*".
- Logfiles from both containers' instances of tomcat and shibboleth are mounted to the VM's filesystem at /home/shibboleth/logs

---IMPORTANT: It is **CRITICAL** to understand that the Docker containers generated and stored by this system, and your configurations in particular (as always), contain very sensitive information, in particular, the private keys used to sign and decrypt data. Please treat all old containers, archived containers, config trees, and archived config trees as sensitive data and protect and dispose of them accordingly. Specifically, do not upload or publish your containers to a public repository or one you do not control access to.

*******Burning/copying the config means that containers stand alone and rely on nothing external. They contain all needed secrets and config. This is the recommended approach. Mounting the config means that the containers maintain a live connection to the config files on the VM and so the IdP's normal mechanisms for config reloading will function as expected. However, it is expected that the performance of a mounted config will not be as robust as that of a burned config.

Operational Commands

- Shibb Status Page:
  - curl -k https://127.0.0.1/idp/status
- To setup the config for the testbed
  - /home/shibboleth/bin/shibb-config-restore -t
  - /home/shibboleth/bin/rebuild.sh

    - /home/shibboleth/bin/restart.sh (takes 10+ minutes due to waiting for drain)
- To reset to the default release config:
    - /home/shibboleth/bin/shibb-config-restore -r
    - /home/shibboleth/work/setup.sh
    - /home/shibboleth/bin/rebuild.sh
    - /home/shibboleth/bin/restart.sh (takes 10+ minutes due to waiting for drain)
- To backup your current config:
    - /home/shibboleth/bin/shibb-config-snap.sh
        - (will create a file similar to: /home/shibboleth/shib-idp-config_04082017-112546.zip)
- To restore a backup made from the previous command:
    - /home/shibboleth/shibb-config-restore.sh -f=/home/shibboleth/shib-idp-config_04082017-112546.zip
    - /home/shibboleth/bin/rebuild.sh
    - /home/shibboleth/bin/restart.sh (takes 10+ minutes due to waiting for drain)

- docker stop shibboleth_idp_0 shibboleth_idp_1 haproxy
  Shuts down the Shibboleth IdP containers
- /home/shibboleth/bin/rebuild.sh
  Run this command after making a change to the Shibboleth configuration tree to build your new configuration into the Docker images.
- /home/shibboleth/bin/restart.sh
  Running this command rebuilds the Docker containers based on your new configuration and sequentially restarts the running containers running behind the load balancer to minimize user disruption.

- If you run low on disk space
    - /usr/local/bin/remove-containers.sh
    - /usr/local/bin/remove-images.sh
- If you want to save a snapshot of the shibboleth container before making a configuration change, issue the following command:
    - /home/shibboleth/bin/tag.sh my/shibboleth_idp_0
    - Output will look similar to the following:

```
Searching for my/shibboleth_idp_0 image
Image my/shibboleth_idp found
Tagging the container
7313879f2d22-my/shibboleth_idp-161110122546

New docker image tagged $7313879f2d22-my/shibboleth_idp-161110122546
```

    - In order to revert to the save container, issue the following command (replacing the text in red with the container tag number output in the preceding command):
        - /home/shibboleth/bin/restore.sh my/shibboleth_idp_0 161110122546

Some Useful Docker Commands

While the normal idea is that you should never need to look inside a container, it is possible and is sometimes useful for debugging unusual issues. These commands may be helpful.

1. docker ps
   Shows the names and status of any running containers.
2. docker exec
   Run a command inside a running docker container. You will find **docker exec -it shibboleth_idp_0 bash** a handy command for debugging Shibboleth issues. This command will open a root shell inside the container and map the output back to your VM session. Inside the container you will find the familiar /opt/shibboleth-idp tree, including access to the configuration and logs.
3. docker start
   To start the IdP after rebooting the VM, run **docker start shibboleth_idp_0**
4. docker stop
   To stop the IdP, run **docker stop shibboleth_idp_0**
5. docker cp
   Used to copy files in to or out of a running container. The syntax is similar to scp. For example, to copy a logfile to the VM, run the command **docker cp shibboleth_idp_0:/opt/shibboleth-idp/logs/idp-process.log .**