Sandboxing with Splunk

...while you get settled...

▶ Latest Slides:
  • https://splunk.box.com/v/blueprints-docker-sandbox

▶ Handout:
  • https://splunk.box.com/v/blueprints-docker-sandbox-ref

▶ Collaborate: #docker-sandbox
  • Sign Up @ http://splk.it/slack

▶ Load Feedback -------------------------------
Sandboxing with Splunk
(with Docker)

Burch | Senior Best Practices Engineer

.conf2017 | Washington, DC
During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.
Who are we?
Who are you?

This should make sense

► Experience installing Splunk
► Familiar with Splunk’s Ports
► Comfortable using Command Line

It’s totally cool if you wanna bounce!
What’s a “Burch”?  

Senior Best Practices Engineer

- Was a Senior Sales Engineer
- Before that, Splunk Customer
- Before that, Middleware Eng
- Before that, Computer Science
- Before that, an idea of my parents

Docker Expert
Follow up != Prerequisite

Splunk Snacks

Your Splunk Sandbox
Wednesday, September 28, 2016 | 11:00 AM-11:15 AM

**Track**: Community Theater | **Session Focus**: Using Splunk | **Other Topics**: Best Practices, Getting Data In, Dev Tools

**Speakers**
Burch Simon, Senior Sales Engineer, Splunk

**Recording** | **Slides**
I sense much fear in you.
Greatest Sandbox == Least Effort
Build a Splunk Sandbox: Options

Localhost
Admin Rights?

Cloud
Exfiltration?

Virtual
Effort?

Container
Coolness?
Install & Setup

Already done to your lab machines
Download Docker

https://www.docker.com/
Hosted in Docker Store

Docker Community Edition for Mac

By Docker

The fastest and easiest way to get started with Docker on Mac

Categories: Docker Community Editions

Get Docker Community Edition for Mac

Docker for Mac is available for free.

Requires Apple Mac OS Yosemite 10.10.3 or above. Download Docker Toolbox for previous OS versions.

By downloading this, you agree to the terms of the Docker Software End User License Agreement

Get Docker  Usage Instructions

Docker CE for Mac

Docker CE for Mac is an easy-to-install desktop app for building, debugging, and testing Dockerized apps on a Mac. Docker for Mac is a complete development environment deeply integrated with the Mac OS Hypervisor framework, networking, and filesystem. Docker for Mac is the fastest and most reliable way to run Docker on a Mac.

Features and Benefits

- Easy installation and setup of a complete Docker development environment for the Mac.
- Integrated Docker platform and tools (Docker command line, Docker Compose, and Docker Notary) command line.
- Automatic updates with channels for monthly Edge and quarterly Stable versions of Docker.
- Fast and reliable performance with native macOS virtualization running a custom minimal Linux distro.
- Seamless volume mounting for code and data, including file change notifications that unlock fast edit-test cycles.
Great Instructions

Installation

System requirements
Docker for Mac works on OS X El Capitan 10.11 and newer macOS releases.

Get Docker

Stable
The Stable version is fully baked and tested, and comes with the latest GA release of Docker.

Edge
The Edge version offers cutting edge features and comes with experimental features turned on.

- Get Docker CE for Mac (Stable)
- Get Docker CE for Mac (Edge)

Install it

Double-click Docker.pkg to start the install process.

When the installation completes and Docker starts, the whale in the top status bar shows that Docker is running, and accessible from a terminal.

Run it

Open a command-line terminal, and try out some Docker commands.

- Run `docker version` to check that you have the latest release installed.
- Run `docker run hello-world` to verify that Docker is pulling images and running as expected.
Splunk Images

https://store.docker.com/profiles/splunk
$ docker pull splunk/splunk
Using default tag: latest
latest: Pulling from splunk/splunk
ad74af05f5a2: Pull complete
6ed26c881126: Pull complete
0efc5ebe5075: Pull complete
123d19a3ee15: Pull complete
6fe48f1452ee: Pull complete
fc6bbc9992f6: Pull complete
8ebdf9134129: Pull complete
Digest: sha256:1be3208a6c1d96ca5ad320fc21cbfcf06428e3ea12f10773e2efc7d2db4b522
Status: Downloaded newer image for splunk/splunk:latest

$ docker images
REPOSITORY          TAG           IMAGE ID       CREATED          SIZE
splunk/splunk       latest        9cad3d52dc92   9 days ago       736MB
Command Crash Course

Put on your helmet
Instantiate & Start Splunk

- docker run -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk
Instantiate & Start Splunk

- docker run -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk

  - run

    Run a command in a new container
Instantiate & Start Splunk

- docker run -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk
  
  • run  Run a command in a new container
  
  • -P, --publish-all  Publish all exposed ports to random ports
Instantiate & Start Splunk

- docker run -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk

- run
  Run a command in a new container

- -P, --publish-all
  Publish all exposed ports to random ports

- -d, --detach
  Run container in background and print container ID
Instantiate & Start Splunk

- docker run -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk

- run
  Run a command in a new container

- -P, --publish-all
  Publish all exposed ports to random ports

- -d, --detach
  Run container in background and print container ID

- -e, --env list
  Set environment variables
Instantiate & Start Splunk

- docker run -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk

  - run
    Run a command in a new container
  
  - -P, --publish-all
    Publish all exposed ports to random ports
  
  - -d, --detach
    Run container in background and print container ID
  
  - -e, --env list
    Set environment variables

  - SPLUNK_START_ARGS="--accept-license"
Instantiate & Start Splunk

- `docker run -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk`

  - `run` Run a command in a new container
  - `-P, --publish-all` Publish all exposed ports to random ports
  - `-d, --detach` Run container in background and print container ID
  - `-e, --env list` Set environment variables
    - `SPLUNK_START_ARGS="--accept-license"`
  - `splunk/splunk` Image name
Now what?

$ docker container list

- CONTAINER ID    unique id
- IMAGE           splunk/splunk
- COMMAND         out of scope for us today
- CREATED         relative time existing
- STATUS          relative time running
- PORTS           Port mappings. See below
- NAMES           Random name (unless --name used)

0.0.0.0:32784->8000/tcp

- localhost:32784 traffic to container port 8000 (splunk web!)
- 32784 different each time (–P)
Look ma! Splunk!
But wait, what about conf files?

Terminal Access

- Command Line:
  - `docker exec -it <container name|id> bash`
    - `-i, --interactive`  Keep STDIN open even if not attached
    - `-t, --tty`  Allocate a pseudo-TTY

- But not much installed
  - Not even vi
  - `apt-get FTW!`
But wait, what about conf files?

Local Editing

▶ Or, mount a folder!
  • `-v, --volume list` Bind mount a volume
  • Example: `-v apps/local_app:/opt/splunk/etc/apps/remote_app`
    • `local_app` is a folder on the host; `remote_app` lives in the container

▶ Directly edit on your host (GUI editor)

▶ Direct link == edits reflected in container
Key Docker Commands
https://splunk.box.com/v/blueprints-docker-sandbox-ref

- Create and start a Splunk container
  - `docker run --name foo -P -d -e SPLUNK_START_ARGS="--accept-license" splunk/splunk`
  - Optional: `...license" -v ~/Desktop/myapp:/opt/splunk/etc/apps/conf2017_app splunk/s...

- Navigate web browser to container
  - `docker container list`

- Stop a container
  - `docker stop foo`

- Start a container
  - `docker start foo`

- Destroy a container
  - `docker rm -fv foo`
    - `-f, --force` Force the removal of a running container (uses SIGKILL)
    - `-v, --volumes` Remove the volumes associated with the container
Create a Splunk Sandbox

This is where you come in...lazy bones ;)}
Your Mission
should you choose to accept it…

▶ Create and Connect many containers

▶ Change settings
  • Suggestion: http port and try UI restart ;)

▶ Destroy!

▶ Change environment variables, set hostname, etc…
  • [https://store.docker.com/community/images/splunk/splunk](https://store.docker.com/community/images/splunk/splunk)

▶ Load Tutorial App & add web.conf settings
Next Steps!

TREAT YO SELF
Container Resources

Adjust the computing resources dedicated to Docker.

CPUs: 4

Memory: 2.0 GB

Disk image location
/users/osimon/Library/Containers/c...er.driver.amd64--linux/Docker.qcow2

Move disk image  Open in Finder

Apply & Restart

Docker is running

CPUs: 8

Memory: 16.0 GB

Disk image location
/users/osimon/Library/Containers/c...er.driver.amd64--linux/Docker.qcow2

Move disk image  Open in Finder

Apply & Restart

Docker is running
Do More!
https://store.docker.com/community/images/splunk/splunk

Basic configuration with Environment Variables

You can use environment variables for basic configuration of the indexer and forwarder. For more advanced configuration, create configuration files within the container or use a Splunk deployment server to deliver configurations to the instance.

- SPLUNK_ENABLE_DEPLOY_SERVER="true" - Enables deployment server on Indexer.
- SPLUNK_DEPLOYMENT_SERVER="<servername>:<ports>" - Configure deployment client. Set deployment server url.
- Example: --env SPLUNK_DEPLOYMENT_SERVER="splunkdeploymentserver:8889".
- SPLUNK_ENABLE_LISTEN="<ports>" - enable receiving.
  - Additional configuration is available using SPLUNK_ENABLE_LISTEN_ARGS environment variable.
- SPLUNK_FORWARD_SERVER="<servername>:<ports>" - forward data to indexer.
  - Additional configuration is available using SPLUNK_FORWARD_SERVER_ARGS environment variable.
  - Additional forwarders can be set up using SPLUNK_FORWARD_SERVER_<CL:30>_ARGS.
  - Example: --env SPLUNK_FORWARD_SERVER="splunkindexer:9997" --env SPLUNK_FORWARD_SERVER_ARGS="method=clone" --env SPLUNK_FORWARD_SERVER_1="splunk indexer2:9997" --env SPLUNK_FORWARD_SERVER_1_ARGS="method=clone".
- SPLUNK_ADD="<monitor add <what to monitor>what to add>" - execute add command, for example to monitor files or listen on specific ports.
  - Additional add commands can be executed (up to 30) using SPLUNK_ADD_<CL:30>.
  - Example: --env SPLUNK_ADD_1=udp 514 or --env SPLUNK_ADD_1="monitor/var/log/".
- SPLUNK_CMD="<any splunk command>" - execute any splunk command.
  - Additional commands can be executed (up to 30) using SPLUNK_CMD_<CL:30>.
  - Example: --env SPLUNK_CMD="edit user admin -password random_password -role admin -auth".

Ports

This Docker container exposes the following network ports:
- 8088/tcp - Splunk Web interface
- 8888/tcp - HTTP Event Collector
- 8888/tcp - Splunk Services
- 8191/tcp - Application Key Value Store
- 9997/tcp - Splunk receiving Port (not used by default) typically used by the Splunk Universal Forwarder
- 1514/tcp - Network Input (not used by default) typically used to collect syslog TCP data

This Docker image uses port 1514 instead of the standard port 514 for the syslog port because network ports below 1024 require root access. See Run Splunk Enterprise as a different or non-root user.

Hostname

When you use this Docker image, set a Hostname for it. If you recreate the instance later, the image retains the hostname.
Data Volume Increase
> 60 day then doing it wrong

Splunk Developer License Signup

With the Splunk developer license, you can use our SDKs and other developer tools to build big data applications that plug into Splunk’s map/reduce data-processing pipeline, storage technology, and management facilities. And, you can extend and enhance Splunk Web through our app framework. Just follow a few simple steps below and you’ll be on your way.

1. Get Splunk Enterprise.

You want to build applications that work on top of Splunk platform, then you need a license to Splunk Enterprise software, which is our flagship core product. Splunk Enterprise is a snap to install and easy to configure. Download it here.

2. Get your developer license.

You’ll need to request your developer license.

3. Get started and stay connected.

- Learn how to install your license.
- Get the SDKs from Splunk GitHub.
- Check out the apps on Splunkbase.
- Follow us on Twitter for latest updates @splunkdev.
- Join our Google Group.
Sample Data
Tutorial, Prod Export, _internal, Peer with Prod
Upgrade Splunk Image

docker pull splunk/splunk

$ docker pull splunk/splunk:latest
Using default tag: latest
latest: Pulling from splunk/splunk
ad74af05f5a2: Pull complete
6ed26c881126: Pull complete
0efc5eeb5075: Pull complete
2d31d03ee15: Pull complete
6fe48f1452ee: Pull complete
fc6bbcb9926e: Pull complete
8ebdf9134129: Pull complete
Digest: sha256:1be3208a6c1d96ca5ad320fc21cbfcf06428e3ea12f10773e2efc7d2dbb4b522
Status: Downloaded newer image for splunk/splunk:latest

$ docker images

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>splunk/splunk</td>
<td>latest</td>
<td>9cad3d52dc92</td>
<td>9 days ago</td>
<td>736MB</td>
</tr>
<tr>
<td>splunk/splunk</td>
<td>6.6.2</td>
<td>1b6fa73035a6</td>
<td>5 weeks ago</td>
<td>736MB</td>
</tr>
</tbody>
</table>
Splunk n' Box

https://github.com/mhassan2/splunk-n-box

Breakout Session

Splunk n' Box

Wednesday, September 27, 2017 | 3:30 PM-4:15 PM  INTERMEDIATE

Mo Hassan, Missouri, Splunk Inc.

I have written an extensive and feature-rich bash script (4000+ lines) that can be used by Splunk admins, regular users and Splunk employees to test multiple Splunk deployment scenarios using Docker (while shielding the user from learning Docker in the process). The script is widely used by Splunk customers, Splunk SEs and Splunk partners. The code base is the result of five months of development and testing.

Less
What Now?

Related breakout sessions and activities…

1. Rate this! (be honest)

2. Collaborate: #docker-sandbox
   • Sign Up @ http://splk.it/slack

3. Customer Success Studio

4. More talks, search for
   • Blueprints
   • Burch
   • Champagne
   • Delaney
   • Optimization
   • Best Practices
   • Veuve
Questions & Discussion?

Don't forget to rate this session in the .conf2017 mobile app