QueueMetrics
call center solution

Asterisk & the Docker revolution

Some lessons from the trenches

Asterisk Africa – Johannesburg - March 14, 2018

Asterisk Community Conference Africa 2018
14th & 15th March 2018, Johannesburg

Presented by:

Lenz Emilitri
Founder, Loway

@lenz

Loway
Measure. Improve.

Today's presentation

Loway
Measure. Improve.

- Docker
  - Benefits
  - How it works
  - Pros & cons
- Asterisk on Docker
  - Why and why not
  - Pitfalls & pain points
- Lessons learnt

Why me?

Loway
Measure. Improve.

- In 2012 I started thinking about hosted service for QueueMetrics
- Found Docker in 2013
- First beta customers in 2014
- Launched as public product in October 2015 – http://queuemetrics-live.com
- Expect to have ~2000 instances by end of 2018

What is Docker?

Loway
Measure. Improve.

- Docker is an engine and ecosystem built to run containers
- Containers run images – stand-alone executable packages that include everything needed to function
- You *can* think of containers as micro-Vms
  - That's what they were designed for…
  - ...but maybe you shouldn't.

Why is Docker a “revolution”?

Loway
Measure. Improve.

- Improves server density
  - Bare metal → VM → Container
- Installation simplicity
  - ISO → Package manager → Container
- Reduces TCO
  - You manage a server – you kill a container

How does Docker work?

Loway
Measure. Improve.

- Uses Linux LXC: cgroups, namespaces and iptables
  - LXC (LinuX Containers) available since 2008
  - Similar in spirit to FreeBSD jails
- Private perspective of Kernel resources
  - PIDs
  - Network
  - File-systems

Docker: PIDs

Loway
Measure. Improve.

- Processes running in the same container “see” each other
- Processes are visible from the host container
- A container works like a stand-alone VM
- Container can apply/enforce resource limits
- You can “join” a container (run process or even a shell)

Docker: Networking

Loway
Measure. Improve.

- Expose a set of inbound TCP/UPD ports on the host that will “tunnel” to a specific Docker instance
- Can use fixed IP addresses
- Can create different types of networks, even spanning multiple hosts and resolving DNS queries

Docker: Filesystems

Loway
Measure. Improve.

- Docker uses “overlay” filesystems
- Usually mounts all layers read-only but the last ones
- Flexible mount policy
- Big performance hit compared to “normal” FS
  - *Defaults not suitable for production*
- Can mount data volumes and local dirs

Docker: Images & Base images

Loway
Measure. Improve.

- Images: a complete, installed, ready-to-run file-system
- Made of multiple layers of customization, from a “base image”
- Live on private or public registry servers
- Striving for minimum size – is it worth it?
  - Big images are shared
  - The container only runs your own processes
  - Sane environment helps troubleshooting

Docker: creating an image

Loway
Measure. Improve.

- Create an image by defining a Dockerfile
  - Starts from a “base image”
  - Does stuff (downloads, compiles and installs packages, copies config files)
  - Exposes a process to be run, a set of ports/services and mount points
- Repeat and throw away until satisfied.
  - Push to a repository for consumption

Docker: Running

Loway
Measure. Improve.

- Install Docker (CentOS 7 – 1.12):
  ```
  yum install -y docker-io
  ```
- From a system running Docker, just run:
  ```
  docker run -p 12345:5038 -P -d
  lenz/asterisk-load-test-13
  ```
- Magic just happens
- Behaves as a single black-box

Docker: What goes on

Loway
Measure. Improve.

- You can see a container from its host:

```
0:36 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libexec/docker/docker-runc-current
0:00  \_ /usr/bin/docker-containerd-current -l unix:///var/run/docker/libcontainerd/docker-conta
0:00  |   \_ /usr/bin/docker-containerd-shim-current e18626543d15f37b7725abcea73175042ca347d274f
0:00  |       \_ /bin/bash /ww/run
0:08  |           \_ asterisk
0:00  |           \_ sleep 30
0:00  \_ /usr/libexec/docker/docker-proxy-current -proto tcp -host-ip 0.0.0.0 -host-port 32768 -
0:00  \_ /usr/libexec/docker/docker-proxy-current -proto tcp -host-ip 0.0.0.0 -host-port 12345 -
```

- You can see the proxies that allow network comunication

Docker: Listing

Loway
Measure. Improve.

- Check the state of a container you just run:

```
CONTAINER ID  e18626543d15
IMAGE         lenz/asterisk-load-test13
COMMAND       "/ww/run"
CREATED       9 seconds ago
STATUS        Up 7 seconds
PORTS         0.0.0.0:12345->5038/tcp,
              0.0.0.0:32768->8088/tcp
NAMES         nostalgic_mccarthy
```

Docker: Joining

Loway
Measure. Improve.

- Entering a container:

```
docker exec -it nostalgic_mccarthy
bash

[root@e18626543d15 /]# ps
PID TTY          TIME CMD
  160 ?        00:00:00 bash
  172 ?        00:00:00 ps
[root@e18626543d15 /]# exit
```

Docker: Monitoring

Loway
Measure. Improve.

- Resource usage:
  `docker stats`

```
CONTAINER   CPU %       MEM USAGE / LIMIT
MEM %     NET I/O             BLOCK I/O           PIDS

e186263d15  1.19%       28.69 MiB / 1.797 GiB
1.56%     1.527 kB / 804 B    106.1 MB / 454.7 kB   0
```

- Processes in container:
  `docker top`

Docker: Stopping / Removing

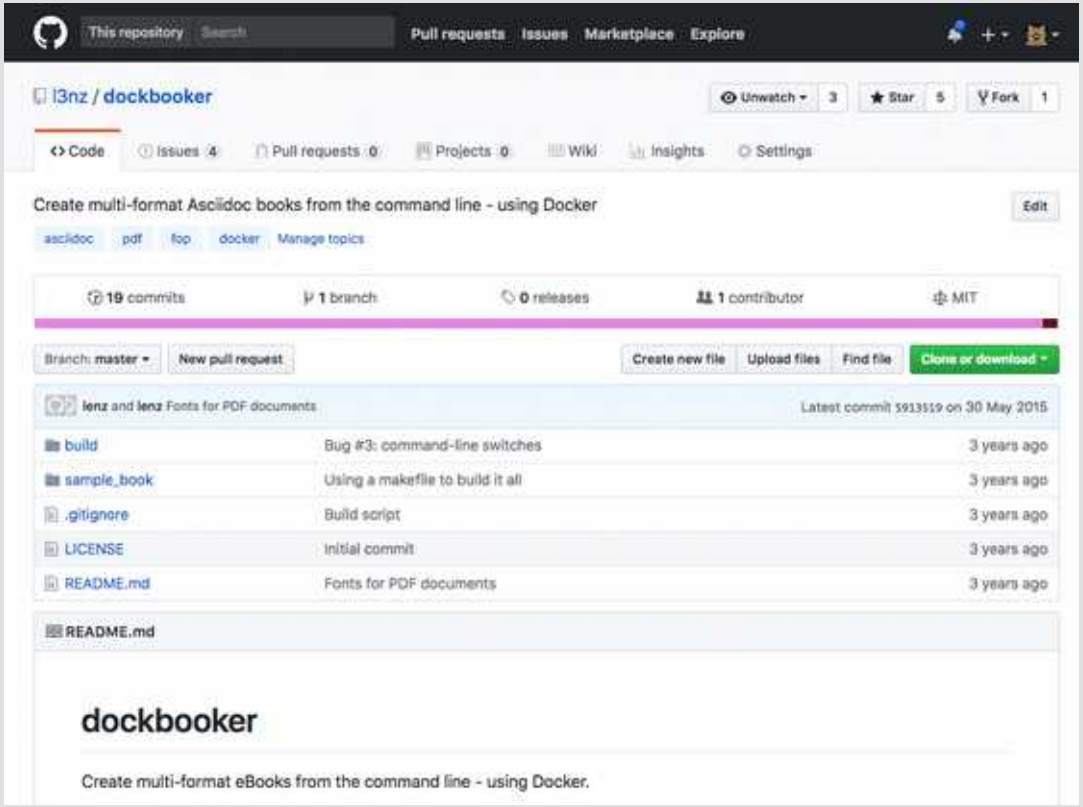
Loway
Measure. Improve.

- To stop a container:
  ```
  # docker stop nostalgic_mccarthy
  nostalgic_mccarthy
  ```
- Container appears as: *Exited (137) 6 seconds ago*
- Remove container (purges storage):
  ```
  # docker rm e18626543d15
  e18626543d15
  ```

Example: Data Transformer

Loway
Measure. Improve.

- Non-persistent process
- Avoid installation / dependency hell
- Stuff in → Stuff out
- Don't care *how* it's done
- Very common: compile pipelines, transcoding, watermarking...



Docker pro's

Loway
Measure. Improve.

- Easy to run
- No configuration mismatches
- "Disposable" mentality for upgrades
- Encourages decomposition
- Use different base distros as needed
- Performance
  - Little overhead and latency compared to VM
  - IO / Networking generally adequate

Docker cons

Loway
Measure. Improve.

- Managing state / distributed state
- Managing data volumes
- Managing product life-cycle
- Production monitoring
- What actually happens within a third-party container?

Today's presentation

Loway
Measure. Improve.

- Docker
  - Benefits
  - How it works
  - Pros&cons
- Asterisk on Docker
  - Why and why not
  - Pitfalls & pain points
- Lessons learnt

Why Asterisk on Docker

Loway
Measure. Improve.

- Organizational advantages:
  - Rationalize deployment
  - Simplify testing and development
  - Simplify upgrades and rollbacks
  - Simplify production environment
- Technical advantages:
  - Lower latency compared to VM
  - Horizontal scalability

Asterisk: rationalize & tame

Loway
Measure. Improve.

- Asterisk is a prime example of “pet” vs “cattle”
  - Few, hand-tweaked servers
  - Local configuration
  - Sometimes compiled locally
  - Often exposed to clients (phones/providers)
  - “Just works”
- Opportunity: rationalize deployment

Asterisk: simpler development

Loway
Measure. Improve.

Simplify development / testing

- Use same production image
- Easier to run system tests
- Capture environment differences explicitly
- Less painful than spinning up a full VM
- Way less painful than installing from ISO

Asterisk: better upgrades

Loway
Measure. Improve.

- Easier and cleaner management of upgrades
- Expected downtime measured in seconds
- Simple to implement Blue/Green scenarios
- Easy roll-back in case upgrades abort!

Asterisk as a part of production

Loway
Measure. Improve.

- Asterisk is only a part of a real-world solution:
  - Reporting
  - Log storage
  - Dialers
  - Recordings
  - CRM
  - ...you name it
- All of them can benefit from Docker

Asterisk: latency

Loway
Measure. Improve.

- A container has better latency than a VM
  - Native scheduler
  - Native clock / interrupts
- A container pre-empts less resources than a VM

Asterisk: horizontal scalability

Loway
Measure. Improve.

- Pool of Asterisks + Kamailios: can grow quite a lot
- Favours a mindset of “many, simple, replaceable” and not monoliths

When NOT to use

Loway
Measure. Improve.

- Need access to specific hardware
- Need an all-in-one distro or app that is not available in Docker

Pitfalls & pain points

Loway
Measure. Improve.

- Asterisk specific
  - Networking / configuration
- Docker specific
  - Orchestration
  - Organization

Pitfalls: Asterisk specific

Loway
Measure. Improve.

- Networking:
  - prefer “host” networking and attach to specific IP
  - expose AMI, ARI, SIP/RTP
  - check *transport_udp* in *pjsip.conf*
- RTP: proxy very slow
  - reduce port range in *rtp.conf* (1000s)
- Resources & limits:
  - set ulimit for open files
  - can control CPU/IO/memory limits

Pitfalls: Asterisk file system

Loway
Measure. Improve.

Filesystem has five logical layers:

- Image (read only)
- Write layer on top of image (RW / slow)
- State you need to run
  - Configuration & AstDB
- State you want to keep
  - Logs
  - Recordings

Pitfalls: Managing state

Loway
Measure. Improve.

- Planning
- Try centralizing state:
  - Who does what
  - Who talks to whom
- Need external services:
  - Private registry
  - Notify containers (etcd)
  - Backup containers (S3)
  - Monitoring (Influx/Grafana)

Pitfalls: Data volumes

Loway
Measure. Improve.

- No “hot migration”
- Data-heavy containers tied to data:
  - Easier to define Hot-Backup policy
  - Will discover early if something missing!
- Data volumes or data directories?
  - Master state (only one - migrated)
  - Logs (one per run)
  - Resources (mounted externally)

Pitfalls: Application life-cycle

Loway
Measure. Improve.

- For Docker an instance is UP when started
  - *This may or may not be the case*
- Life-cycle is not managed
- Workaround: developed Whaleware in 2015
  - Managed life-cycle
  - Gathering of statistics
  - Self-healing when possible
- Still need to notify dependent services

Today's presentation

Loway
Measure. Improve.

- Docker
  - Benefits
  - How it works
  - Pros&cons
- Asterisk on Docker
  - Why and why not
  - Pitfalls & pain points
- Lessons learnt

Lessons learnt

Loway
Measure. Improve.

- Docker may make your life easier
- Strive for simplicity
- Stay away from bleeding edge
- Orchestration can work for you or against you.
- Ansible for effective management of host servers

Links

Loway
Measure. Improve.

- Excellent Asterisk Docker image by Doug Smith: https://github.com/dougbtv/docker-asterisk with HA example based on Kamailio
- Whaleware: managed life-cycle for Docker: https://github.com/l3nz/whaleware with Asterisk examples for AMI/ARI load-testing
- Dockbooker: https://github.com/l3nz/dockbooker

Thanks for attending!

Loway
Measure. Improve.

Pro Tip: You can run QueueMetrics and WombatDialer right out of Docker!