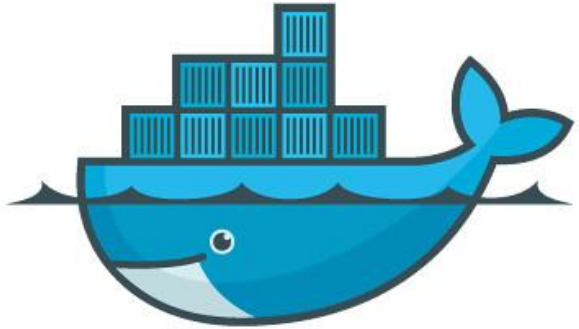

Docker for Network Operators (Part 2)

Anurag Bhatia, Hurricane Electric

Housekeeping rules & announcements!

1. Questions will be taken at regular intervals, usually when one section of the presentation ends. Feel free to type questions in Q&A in meantime. There would also be Q&A and open discussion in the end.

2. We now call these sessions formally as “INNOG Tech sessions” and past tech sessions have been documented here.

3. These sessions are **not** recorded and that’s by intention. That is done to promote open discussions without worrying about showing up on YouTube videos at some point. Sessions at INNOG annual conference are recorded.

4. This talk is part 2 the Docker session. You can find part 1 here.

5. For getting notified of future such sessions you can signup here.

Plan for the day

1. Setup of DNS server (auth + recursor) via Docker containers
2. Setup of mail server - mailcow
3. Setup of ticketing system - OS Ticket
4. Automated backup!

DNS Setup via Docker containers

Type of DNS operations

1. Authoritative DNS
2. Recursive DNS / DNS resolver
3. DNS forwarder

Note: If want to read about DNS in detail, checkout DNS 101 session

Bind9 as authoritative DNS

Image Link:
https://hub.docker.com/_/bind9/plans/3af94cc6-b9c6-43c2-8658-e617ef977949?tab=instructions

Setup steps

1. On Ubuntu 20.04, DNS port 53 is binded to systemd-resolved. Disabling it frees up the port 53 but stops DNS resolution. Thus it's better to stop, disable it & put put hardcoded nameserver values

```
systemctl stop systemd-resolved.service
systemctl disable systemd-resolved.service
Add nameserver in /etc/resolv.conf
```

2. Add recursion no; in named.conf.options since it's authoritative server

3. Add reference to the zone in name.conf.local

4. Add zone file

Reverse DNS

How to setup rDNS PTR

1. Write IP in reverse. Thus for 203.0.113.0/24 zone is: 113.0.203.in-addr.arpa
   For IPv6 2001:db8::/32 zone is 8.b.d.0.1.0.0.2.ip6.arpa.

2. IPv6 PTR is harder to type because one has to fill with zeros. Use dig -x trick to find query zone.

Live demo of rDNS PTR...

Bind9 as DNS resolver instructions

1. Bring up containers to get base config ready
2. Put ACL with your network IPs (aggregates)
3. Enable recursion and apply ACL

Live demo of bind as DNS resolver

Unbound as DNS resolver

https://hub.docker.com/r/mvance/unbound

Note official docker image. Ensure to read dockerfile to check source of program code.

Mailcow - self-hosted mail server

Why self host email?

1. It's way cheaper when you pay for hardware resources (RAM, HDD, CPU) instead of INR/user/month basis. Incremental cost per user is negligible
2. Control over your own data, rules and mail server behaviour
3. With docker it's easy to setup & even restore if hardware breaks
4. It's easier and very much doable with all modern features
5. You can still have split delivery setup if you want to test or move certain users to this new system
6. One can bundle it as a "service" to your enterprise/leased line users user@your-company.com

Microservice architecture...

Different containers performing different actions as needed to run a full fledged mail operation.

Think about what are the features one needs in modern email system?

Microservice architecture...

mailcow: dockerized comes with multiple containers linked in one bridged network. Each container represents a single application.

- ACME
- ClamAV (optional)
- Dovecot
- ejabberd
- MariaDB
- Memcached
- Netfilter (Fail2ban-like integration by @mkuron)
- Nginx
- Oletools via Olefy
- PHP
- Postfix
- Redis
- Rspamd
- SOGo
- Solr (optional)
- Unbound
- A Watchdog to provide basic monitoring

Source: https://mailcow.github.io/mailcow-dockerized-docs/

Hardware requirement

Resource	mailcow: dockerized
CPU	1 GHz
RAM	**Minimum** 6 GiB + 1 GiB swap (default config)
Disk	20 GiB (without emails)
System Type	x86_64

Discussion on where to host it?

Live demo of mail server setup...

Reference documentation here

Note on running mailcow behind NGINX Proxy Manager

1. Understand how ACME works to get you SSL certificate

2. Use different hostname for mail server & web access. For mailserver create reverse proxy rule to redirect only port 80 (not 443). So that included ACME client with mailcow can get certificate for mails

3. For web access let NGINX proxy manager to manage it including SSL

Test of mail server

Send a mail to check-auth@verifier.port25.com (no content needed in email) and read the reply with report on SPF, DKIM etc.

Split delivery mail setup

1. Run two mail servers in parallel - have certain users on one server and certain on others

2. Quite useful if migrating - setup Mailcow with split delivery and point MX to it. It can deliver mails for existing users to existing email server & specified accounts on new server

3. “innog.net” domain does that for mailing list hosting!

SPF and DKIM

1. SPF defines mail servers which are allowed to send emails on behalf of a domain. It’s defined by administrator of the domain who has access to the DNS and it is published as TXT record.

2. DKIM enables adding cryptographic signature to emails, thus receiver can validate those signatures against the public key published by the domain administrator.

Ticketing system

Typical workflow of ticketing systems

- End users log complaints/issues/fresh sales requests etc via email
- End users call via phone & person on phone creates a “ticket”
- Handling of tickets via different groups - Sales, Support (L1/L2/L3), field team
- Creating basic SLA & alert emails when SLA is breached
- Option to create easy to type quick replies and help topics

Live demo of OS Ticket Setup

docker-compose reference here

Automated backup

1. You can use any popular tool like Duplicati
2. Backups must be off-site
3. Ideally multiple copies of backups should be kept
4. Ideally backups should be encrypted before they are uploaded on the remote server
5. Data de-duplication should be done to optimise storage

Options for object storage

Discussion on possible options these days for object storage

Live demo remote backup and restoration

Questions?

anurag@he.net
Web: he.net