FAASTED, KEYLESS ENTRY TO MULTI-HOST DOCKER

Gabe Beged-Dov Prismod Systems, LLC

Background

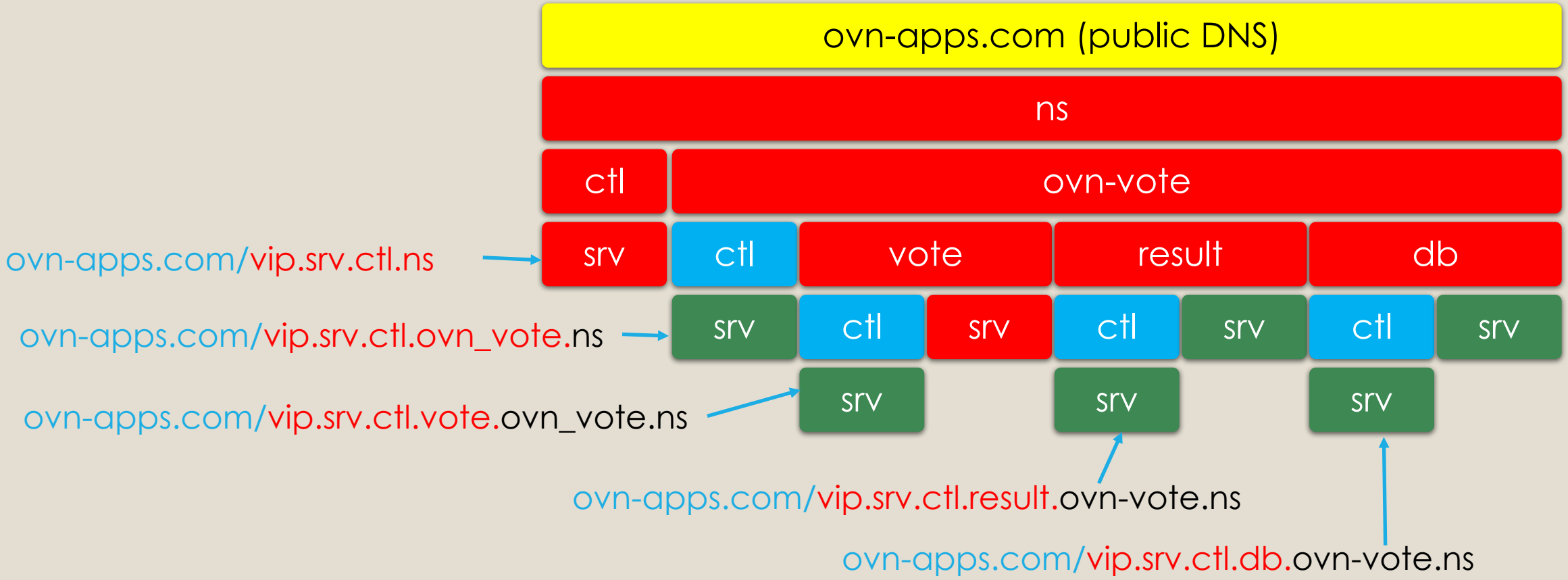
- Docker is developing tech in the FAAS/Serverless space.
  - https://github.com/bfirsh/serverless-docker
- FAASted (Function as a Service Trampoline Extension to Docker) is a vehicle for investigating alterative approaches by focusing on OVN and any necessary extensions
- FAASted uses the server side of Wepoq-OVN (modified for absence of managed clients) and a form of socket activation that may provide benefits over the current Docker approach

Overview

- OVN already provides explicit multi-host docker support via CNM plugin.
  - cluster KV db is required to store state which OVN doesn't need.
- Can run Docker in single node mode by connecting docker0 to br-int
- stable naming hierarchy for discovery
- Docker/Kubernetes as conceptual underlay for Wepoq-OVN
- "socket activation" and caching to speed up FAAS launch

- Mismatch in Docker and Wepoq-OVN network topologies
  - from within
    - ovn-kubernetes (CNI plugin)
    - ovn-docker (CNM plugin)
  - from without
    - Wepoq L4 tagging interlay
    - OVN VLAN tagging interlay
    - OVN tunneling overlay
    - next-to-lay
      - a second vif in each container/pod for Wepoq-OVN
- Current approach is 2nd vif on server side
- "next-to-lay" endpoint attachment triggered by docker events and labels, not plugin

Naming/Dispatch Hierarchy



Dispatch (part 1)

- ovn-apps GW receives the HTTP get request
- GW rewrites
  - ovn-apps.com/vip.srv.result.ovn-vote.ns to
  - vip.srv.result.ovn-vote.ns.ovn.wepoq.net
- OVN DNS resolves it to the VIP ip on the srv.result LS
- The IP is load balanced across three servers on the srv.result LS
- The packet is dispatched to the IP of s3.result for this example
- In the egress host chassis, kernel upcalls the socket listener of container attached to s3.result

Dispatch (part 2)

- FaaSTeD uses a split container model
  - In FAAS, the container should only run on-demand, but then how do you make it start fast?
  - Docker uses a dispatcher container (entrypoint) that dispatches based on the URL path
    - The services are not 1st class in that they are not visible on the network at layer 3
- FaaSTed uses a stub container that listens on the IP
  - accepts the socket
  - starts a pre-warmed implementation container
  - passes the accept socket via the Linux ancillary message feature of Domain sockets