SEP 03 – 04, 2024
CONTAINER days CONFERENCE

Demystifying DNS: A Guide to Understanding and Debugging Request Flows in Kubernetes Clusters

Qasim Sarfraz

About me

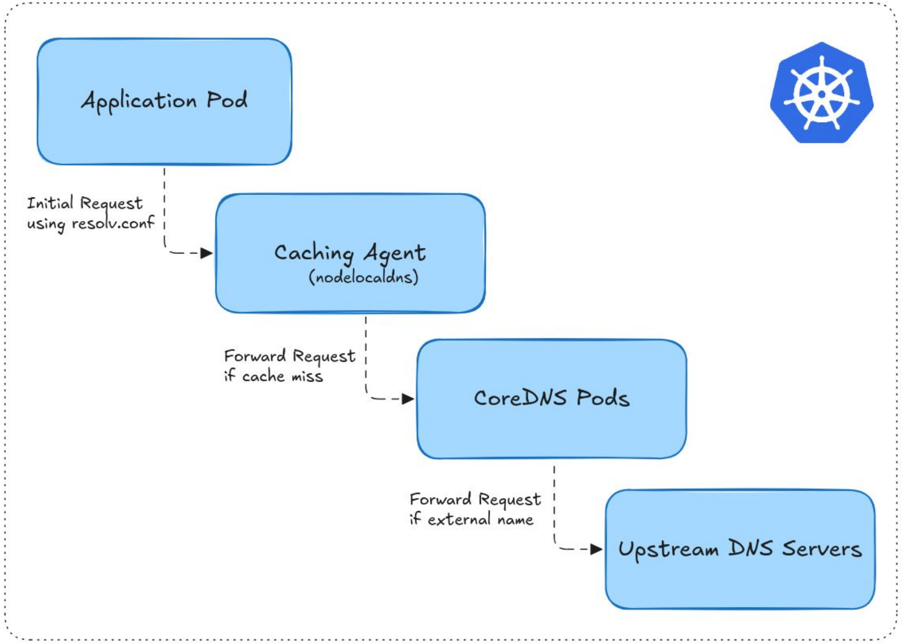
- Pakistan (Lahore) -> Germany (Hamburg)
- Software Engineer @ Microsoft
- Inspektor Gadget, CoreDNS header plugin, other cloud-native projects!

Agenda

- Understanding DNS Components and Request Flows
- Deep Dive into Request Flows using:
  - CoreDNS log plugin
  - Hubble
  - Inspektor Gadget
- Debugging Scenarios

Understanding DNS Components / Request Flows

- In order to debug DNS issues we first need to understand different components:
  1. Application Pod
  2. Node Local DNS
  3. kube-dns Service
  4. CoreDNS Pod
  5. Upstream DNS Server
- Challenges:
  - Too many hidden systems.
  - Not easy to trace the DNS request flows across the cluster.
  - Which tools to use to deep dive into request flows?


Application Pod
Initial Request using resolv.conf
Caching Agent (nodelocaldns)
Forward Request if cache miss
CoreDNS Pods
Forward Request if external name
Upstream DNS Servers

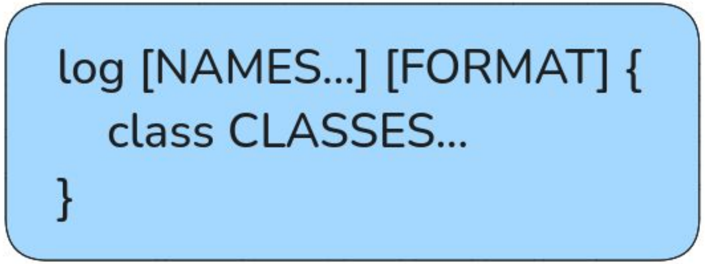
Deep Dive into Request Flows

Deep Dive into Request Flows - Disclaimer

We will trace a DNS request (name=**example.com.**) from pod (name=**mypod**) (namespace=**demo**) on a **minikube** Kubernetes Cluster

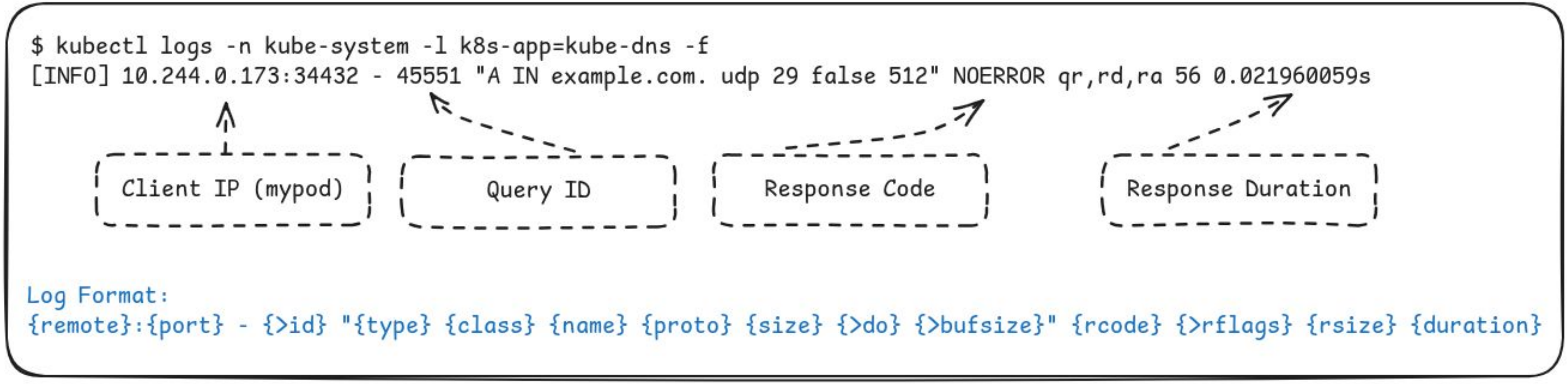
CoreDNS log plugin

- CoreDNS core plugin: https://coredns.io/plugins/log/
- Needs a keyword **log** in Corefile.
- Logs all the requests to stdout.



```
log [NAMES...] [FORMAT] {
    class CLASSES...
}
```

CoreDNS log plugin



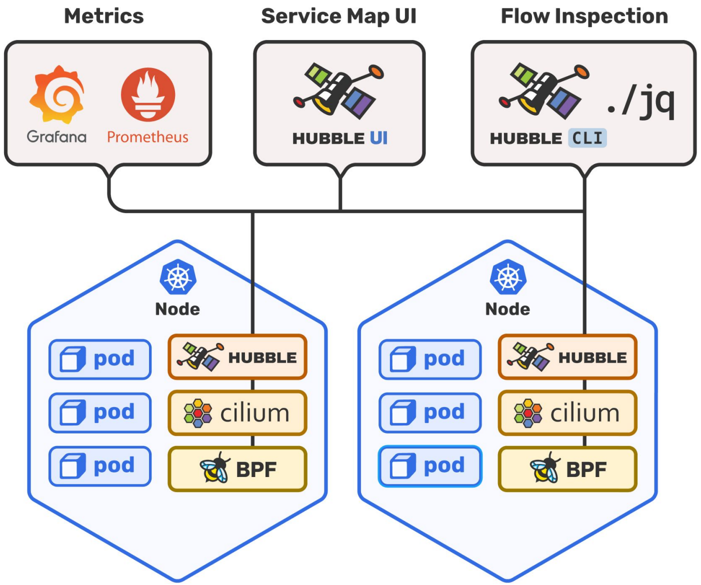
```
$ kubectl logs -n kube-system -l k8s-app=kube-dns -f
[INFO] 10.244.0.173:34432 - 45551 "A IN example.com. udp 29 false 512" NOERROR qr,rd,ra 56 0.021960059s
```

Log Format:

```
{remote}:{port} - {>id} "{type} {class} {name} {proto} {size} {>do} {>bufsize}" {rcode} {>rflags} {rsize} {duration}
```

Hubble

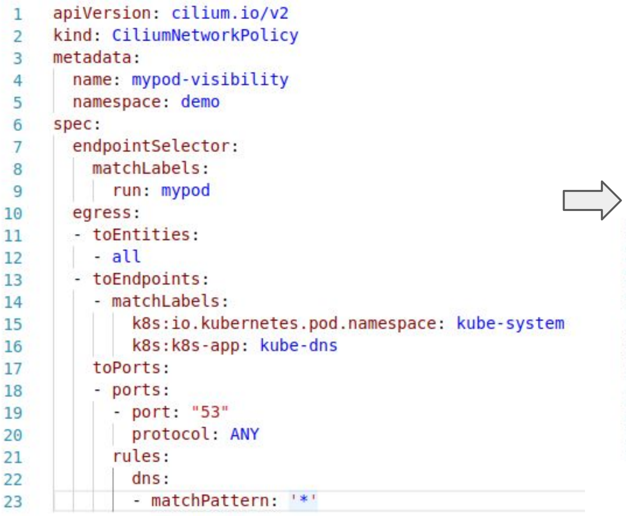
- Provies Network, Service & Security Observability for Kubernetes.
- Built on top of Cilium and eBPF.
- Installed as Kubernetes Daemonset.



source: https://github.com/cilium/hubble

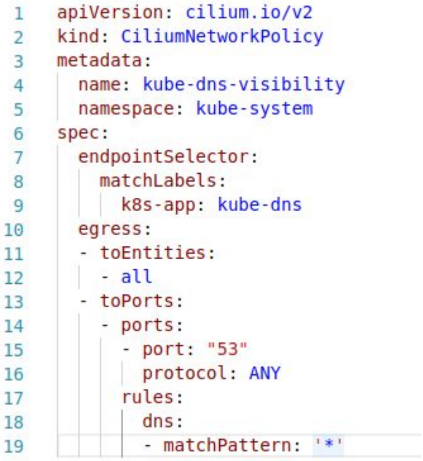
Hubble

- Install Cilium with L7 Proxy Support.
- Create cilium network policies for visibility of DNS traffic for:
  - mypod
  - CoreDNS



```yaml
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: mypod-visibility
  namespace: demo
spec:
  endpointSelector:
    matchLabels:
      run: mypod
  egress:
  - toEntities:
    - all
  - toEndpoints:
    - matchLabels:
        k8s:io.kubernetes.pod.namespace: kube-system
        k8s:k8s-app: kube-dns
    toPorts:
    - ports:
      - port: "53"
        protocol: ANY
      rules:
        dns:
        - matchPattern: '*'
```

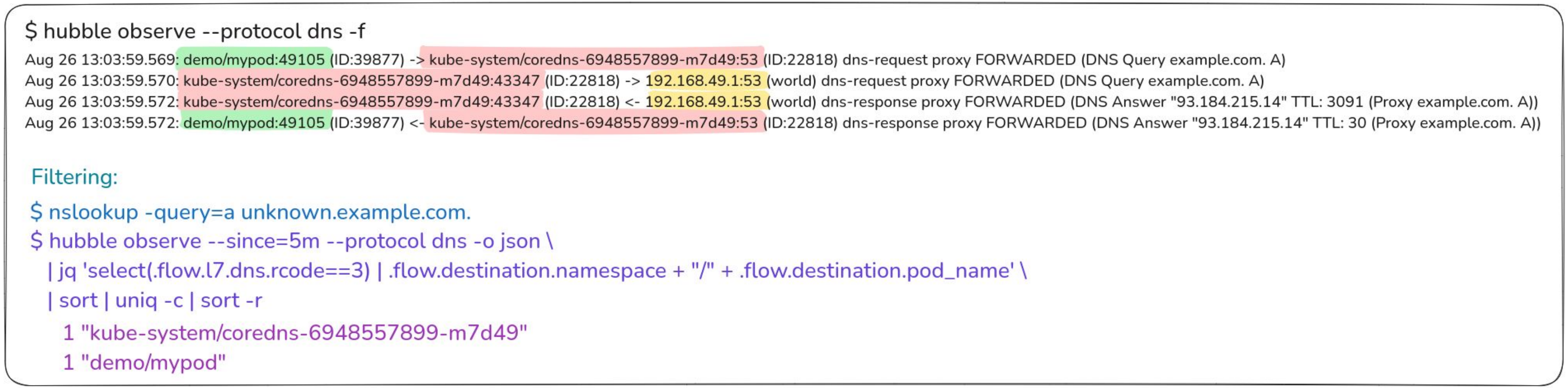
mypod



```yaml
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: kube-dns-visibility
  namespace: kube-system
spec:
  endpointSelector:
    matchLabels:
      k8s-app: kube-dns
  egress:
  - toEntities:
    - all
  - toPorts:
    - ports:
      - port: "53"
        protocol: ANY
      rules:
        dns:
        - matchPattern: '*'
```

CoreDNS

Hubble



```
$ hubble observe --protocol dns -f
Aug 26 13:03:59.569: demo/mypod:49105 (ID:39877) -> kube-system/coredns-6948557899-m7d49:53 (ID:22818) dns-request proxy FORWARDED (DNS Query example.com. A)
Aug 26 13:03:59.570: kube-system/coredns-6948557899-m7d49:43347 (ID:22818) -> 192.168.49.1:53 (world) dns-request proxy FORWARDED (DNS Query example.com. A)
Aug 26 13:03:59.572: kube-system/coredns-6948557899-m7d49:43347 (ID:22818) <- 192.168.49.1:53 (world) dns-response proxy FORWARDED (DNS Answer "93.184.215.14" TTL: 3091 (Proxy example.com. A))
Aug 26 13:03:59.572: demo/mypod:49105 (ID:39877) <- kube-system/coredns-6948557899-m7d49:53 (ID:22818) dns-response proxy FORWARDED (DNS Answer "93.184.215.14" TTL: 30 (Proxy example.com. A))
```

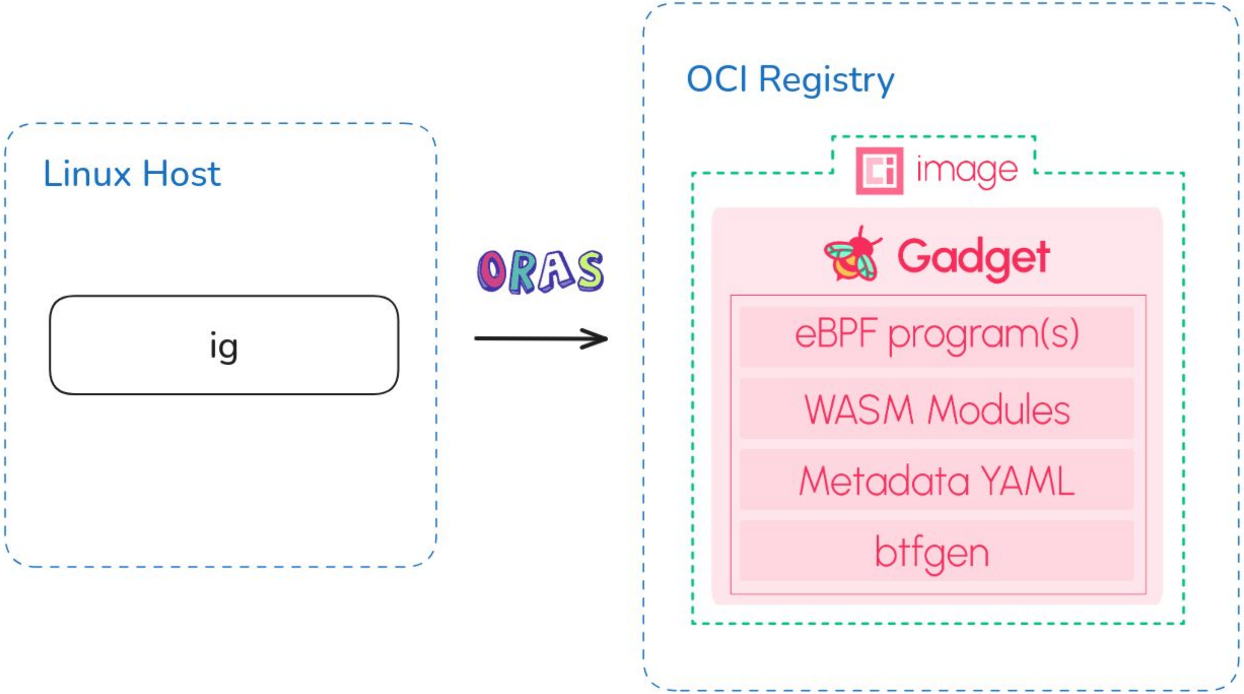
Filtering:

```
$ nslookup -query=a unknown.example.com.
$ hubble observe --since=5m --protocol dns -o json \
  | jq 'select(.flow.l7.dns.rcode==3) | .flow.destination.namespace + "/" + .flow.destination.pod_name' \
  | sort | uniq -c | sort -r
    1 "kube-system/coredns-6948557899-m7d49"
    1 "demo/mypod"
```

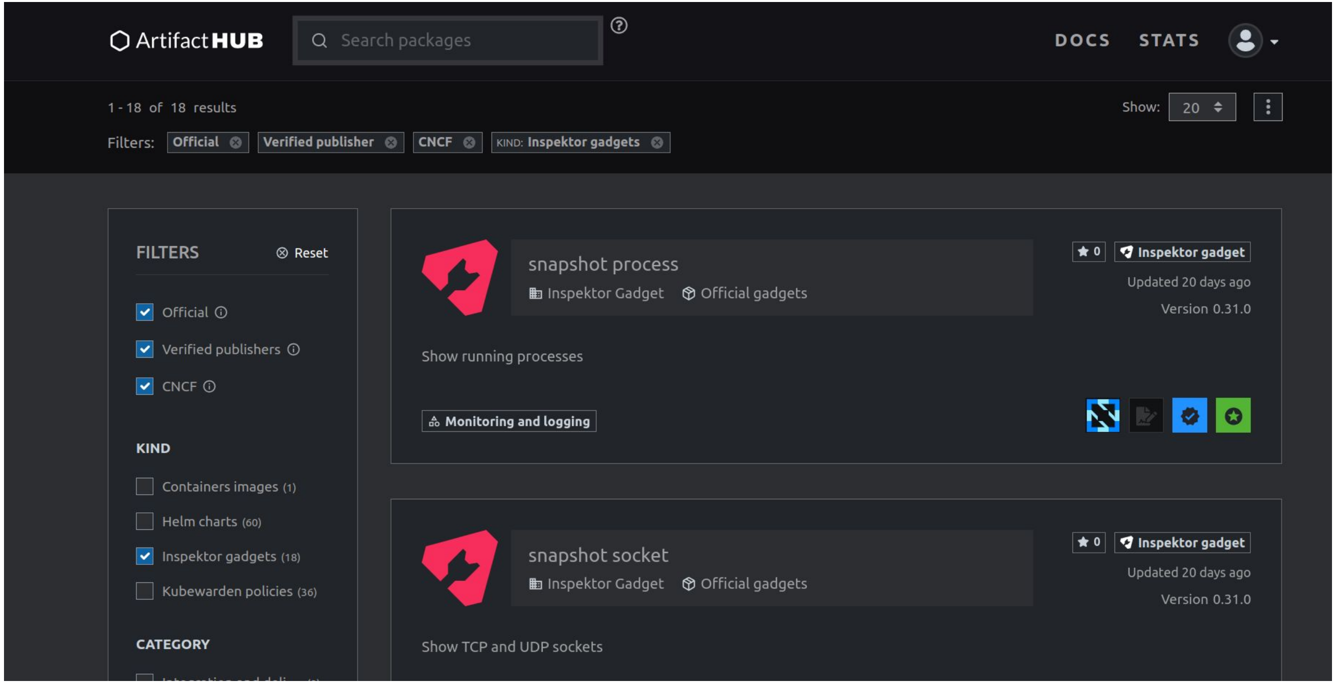
Inspektor Gadget

- **Tool:** A set of tools (gadgets) that empower users to inspect Kubernetes and Linux Systems.
- **Framework:** An observability framework that allows you to build and share custom "gadgets".
- It can be used in Kubernetes as:
  - Daemonset + kubectl gadget.
  - kubectl debug node.

Inspektor Gadget - Gadget



Inspektor Gadget - Gadget



1 - 18 of 18 results

Show: 20

Filters: Official ⊗ Verified publisher ⊗ CNCF ⊗ KIND: Inspektor gadgets ⊗

FILTERS ⊗ Reset

☑ Official ⓘ

☑ Verified publishers ⓘ

☑ CNCF ⓘ

KIND

□ Containers images (1)

□ Helm charts (60)

☑ Inspektor gadgets (18)

□ Kubewarden policies (36)

CATEGORY

snapshot process

Inspektor Gadget · Official gadgets

★ 0 · Inspektor gadget

Updated 20 days ago

Version 0.31.0

Show running processes

Monitoring and logging

snapshot socket

Inspektor Gadget · Official gadgets

★ 0 · Inspektor gadget

Updated 20 days ago

Version 0.31.0

Show TCP and UDP sockets

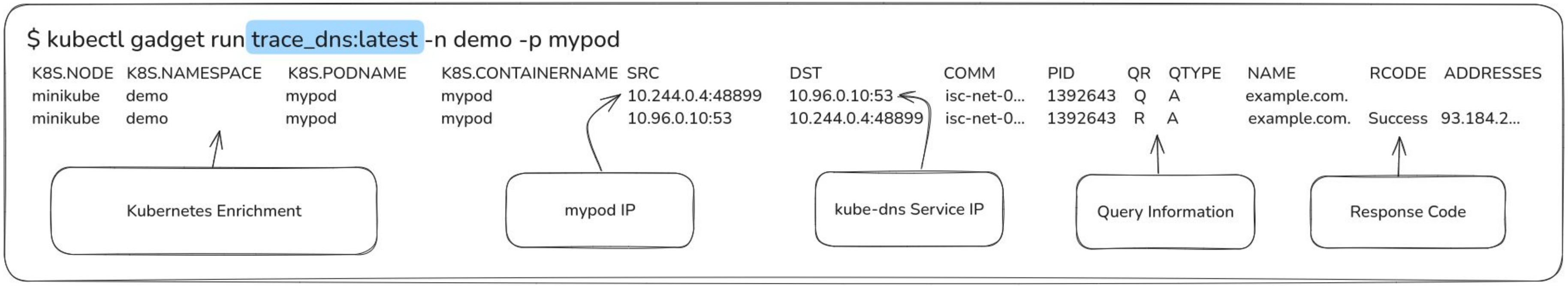
Inspektor Gadget - DNS Gadget

- DNS Gadget.
- Traces DNS requests and responses using eBPF.
- Uses WASM for post-processing.
- Source code. (Modify -> Build -> Push -> Run)

Guide

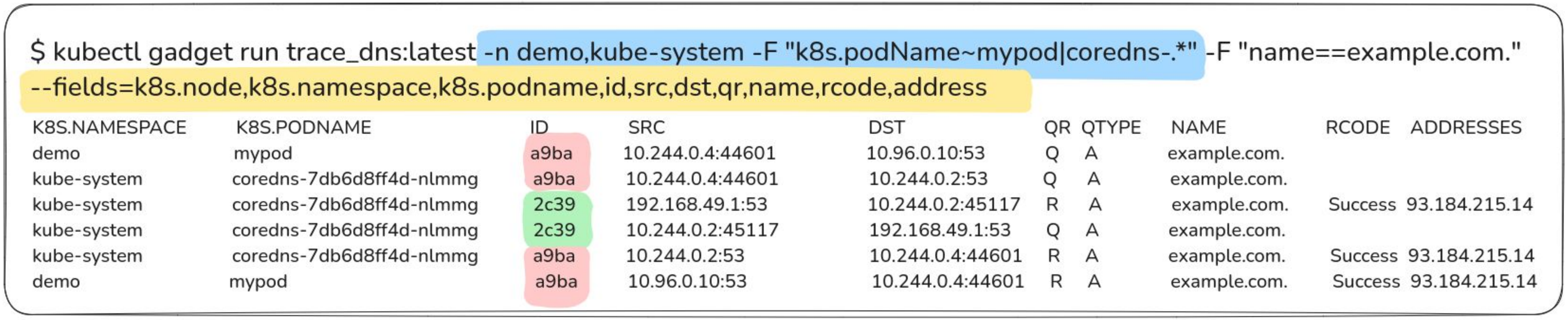
Source Code

Inspektor Gadget (Application Pod)



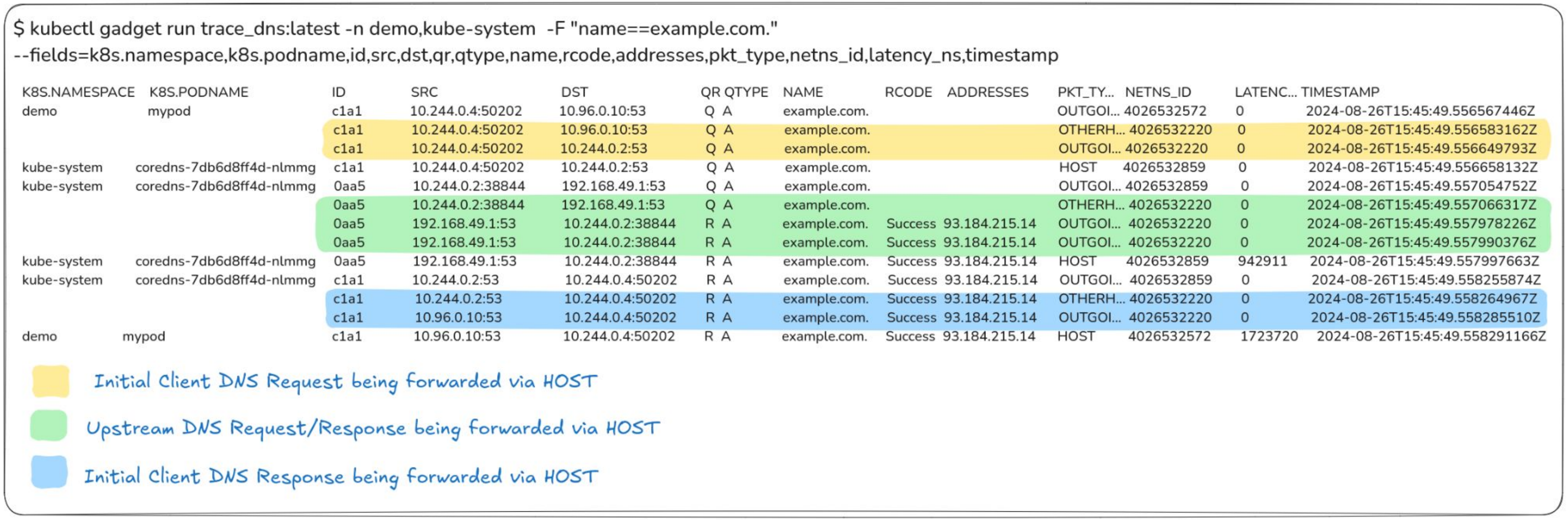
```
$ kubectl gadget run trace_dns:latest -n demo -p mypod
K8S.NODE  K8S.NAMESPACE  K8S.PODNAME  K8S.CONTAINERNAME  SRC                DST                COMM          PID      QR  QTYPE  NAME           RCODE    ADDRESSES
minikube  demo           mypod        mypod              10.244.0.4:48899   10.96.0.10:53      isc-net-0...  1392643  Q   A      example.com.
minikube  demo           mypod        mypod              10.96.0.10:53      10.244.0.4:48899   isc-net-0...  1392643  R   A      example.com.   Success  93.184.2...
```

Inspektor Gadget (Application + CoreDNS Pods)



```
$ kubectl gadget run trace_dns:latest -n demo,kube-system -F "k8s.podName~mypod|coredns-.*" -F "name==example.com."
--fields=k8s.node,k8s.namespace,k8s.podname,id,src,dst,qr,name,rcode,address
K8S.NAMESPACE   K8S.PODNAME                 ID     SRC                 DST                 QR  QTYPE  NAME           RCODE    ADDRESSES
demo            mypod                       a9ba   10.244.0.4:44601    10.96.0.10:53       Q   A      example.com.
kube-system     coredns-7db6d8ff4d-nlmmg    a9ba   10.244.0.4:44601    10.244.0.2:53       Q   A      example.com.
kube-system     coredns-7db6d8ff4d-nlmmg    2c39   192.168.49.1:53     10.244.0.2:45117    R   A      example.com.   Success  93.184.215.14
kube-system     coredns-7db6d8ff4d-nlmmg    2c39   10.244.0.2:45117    192.168.49.1:53     Q   A      example.com.
kube-system     coredns-7db6d8ff4d-nlmmg    a9ba   10.244.0.2:53       10.244.0.4:44601    R   A      example.com.   Success  93.184.215.14
demo            mypod                       a9ba   10.96.0.10:53       10.244.0.4:44601    R   A      example.com.   Success  93.184.215.14
```

Inspektor Gadget (complete trace with OS context)



```
$ kubectl gadget run trace_dns:latest -n demo,kube-system  -F "name==example.com."
--fields=k8s.namespace,k8s.podname,id,src,dst,qr,qtype,name,rcode,addresses,pkt_type,netns_id,latency_ns,timestamp

K8S.NAMESPACE  K8S.PODNAME                ID    SRC                 DST                 QR QTYPE  NAME          RCODE    ADDRESSES      PKT_TY...  NETNS_ID     LATENC... TIMESTAMP
demo           mypod                      c1a1  10.244.0.4:50202    10.96.0.10:53       Q  A      example.com.                          OUTGOI...  4026532572   0         2024-08-26T15:45:49.556567446Z
                                          c1a1  10.244.0.4:50202    10.96.0.10:53       Q  A      example.com.                          OTHERH...  4026532220   0         2024-08-26T15:45:49.556583162Z
                                          c1a1  10.244.0.4:50202    10.244.0.2:53       Q  A      example.com.                          OUTGOI...  4026532220   0         2024-08-26T15:45:49.556649793Z
kube-system    coredns-7db6d8ff4d-nlmmg   c1a1  10.244.0.4:50202    10.244.0.2:53       Q  A      example.com.                          HOST       4026532859   0         2024-08-26T15:45:49.556658132Z
kube-system    coredns-7db6d8ff4d-nlmmg   0aa5  10.244.0.2:38844    192.168.49.1:53     Q  A      example.com.                          OUTGOI...  4026532859   0         2024-08-26T15:45:49.557054752Z
                                          0aa5  10.244.0.2:38844    192.168.49.1:53     Q  A      example.com.                          OTHERH...  4026532220   0         2024-08-26T15:45:49.557066317Z
                                          0aa5  192.168.49.1:53     10.244.0.2:38844    R  A      example.com.  Success  93.184.215.14  OUTGOI...  4026532220   0         2024-08-26T15:45:49.557978226Z
                                          0aa5  192.168.49.1:53     10.244.0.2:38844    R  A      example.com.  Success  93.184.215.14  OUTGOI...  4026532220   0         2024-08-26T15:45:49.557990376Z
kube-system    coredns-7db6d8ff4d-nlmmg   0aa5  192.168.49.1:53     10.244.0.2:38844    R  A      example.com.  Success  93.184.215.14  HOST       4026532859   942911    2024-08-26T15:45:49.557997663Z
kube-system    coredns-7db6d8ff4d-nlmmg   c1a1  10.244.0.2:53       10.244.0.4:50202    R  A      example.com.  Success  93.184.215.14  OUTGOI...  4026532859   0         2024-08-26T15:45:49.558255874Z
                                          c1a1  10.244.0.2:53       10.244.0.4:50202    R  A      example.com.  Success  93.184.215.14  OTHERH...  4026532220   0         2024-08-26T15:45:49.558264967Z
                                          c1a1  10.96.0.10:53       10.244.0.4:50202    R  A      example.com.  Success  93.184.215.14  OUTGOI...  4026532220   0         2024-08-26T15:45:49.558285510Z
demo           mypod                      c1a1  10.96.0.10:53       10.244.0.4:50202    R  A      example.com.  Success  93.184.215.14  HOST       4026532572   1723720   2024-08-26T15:45:49.558291166Z
```

🟨 Initial Client DNS Request being forwarded via HOST

🟩 Upstream DNS Request/Response being forwarded via HOST

🟦 Initial Client DNS Response being forwarded via HOST

Deep Dive into Request Flows - Summary

CoreDNS log plugin	Hubble	Inspektor Gadget
Ideal for initial inspection; however, its scope is limited to CoreDNS.	Provides a compact overview of request flows with enrichment and retrospective analysis.	Offers rich DNS traces with OS context and Kubernetes enrichment/filtering.
Requires configuration changes for use.	Requires Cilium CNI / specific policies to be in place for L7 flow visibility.	Extensible through custom gadget images ; No Support for TCP.

Debugging Scenarios

Debugging Scenario 1 - Verify Health of an Upstream DNS Server

Debugging Scenario 2 - Identify Unsuccessful DNS Response

Conclusion

- First step for debugging DNS is understanding the request flows and:
  - CoreDNS log plugin.
  - Hubble.
  - Inspektor Gadget.
- This was just a starting point so feedback is appreciated.
- Finally, reach out communities if you have any questions/suggestions.

Thank you for Listening!