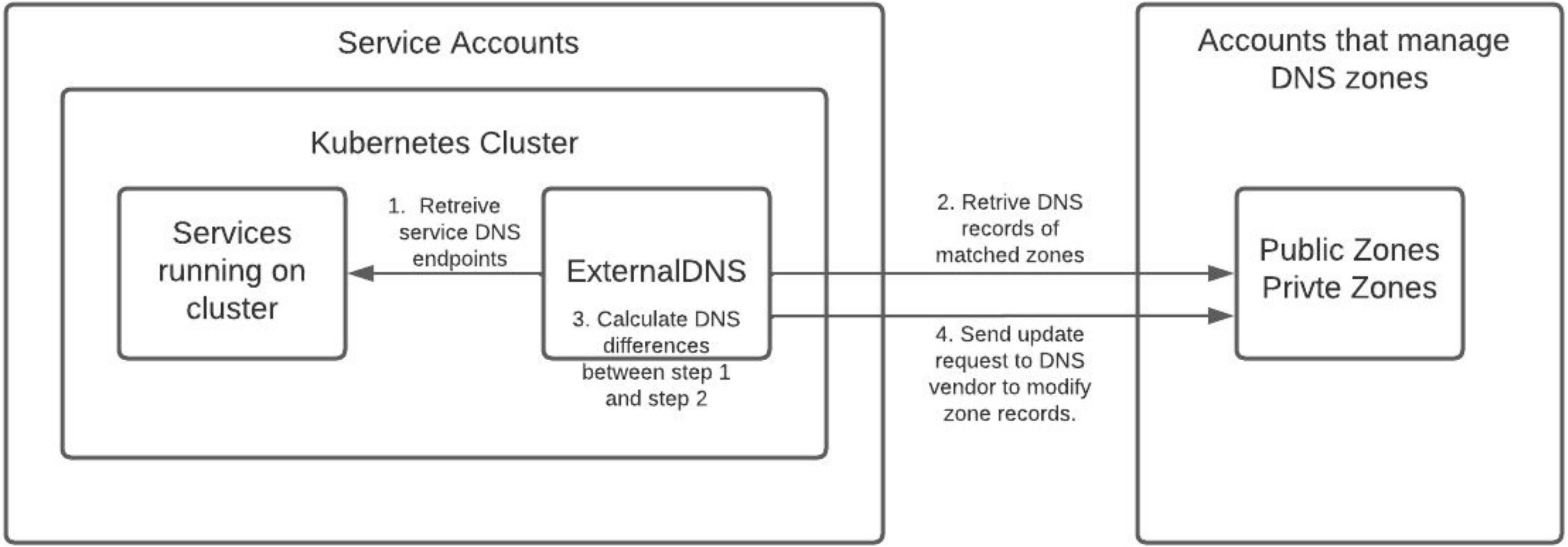
ExternalDNS on AWS in Large Scale

Sidan Qi (sidan.qi@salesforce.com)
Sile Yang (sile.yang@salesforce.com)

ExternalDNS in Cloud

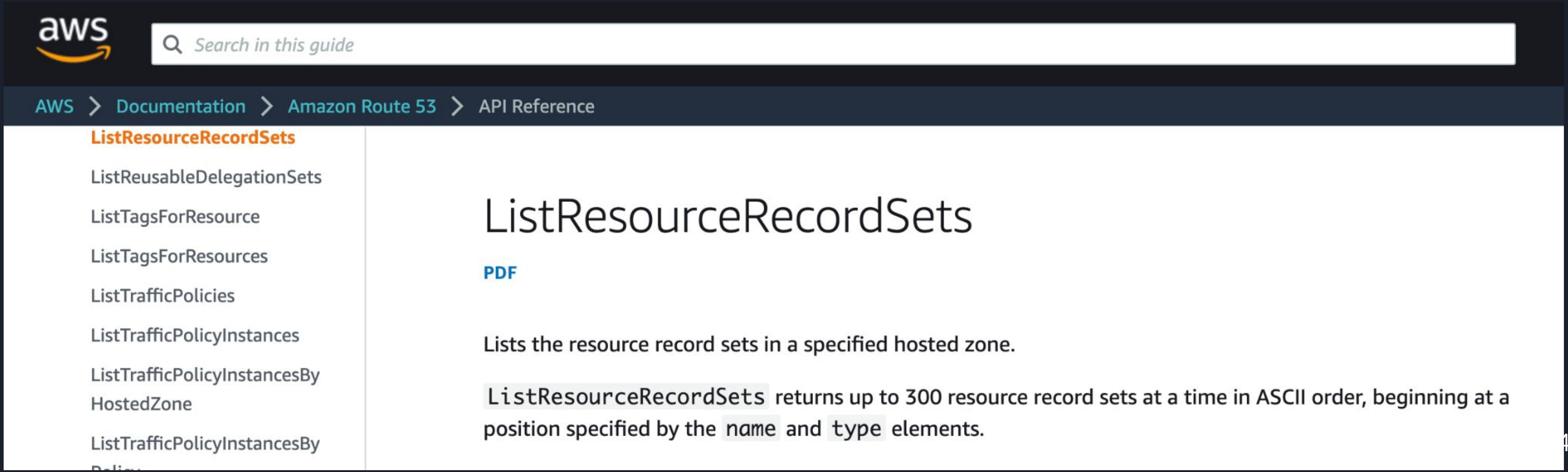

Service Accounts
Kubernetes Cluster
Services running on cluster
1. Retreive service DNS endpoints
ExternalDNS
3. Calculate DNS differences between step 1 and step 2
2. Retrive DNS records of matched zones
4. Send update request to DNS vendor to modify zone records.
Accounts that manage DNS zones
Public Zones
Privte Zones

AWS Usage and Limit

Entity	Quota
Frequency of Amazon Route 53 API requests	5 calls per second
Number of elements in ListResourceRecordSets requests	300 Records per call in ASCII order
Number of Records per zone	10,000 Records (adjustable)

ExternalDNS in large scale

- ExternalDNS downloads all records via Route53 API call
- ExternalDNS creates two ResourceRecordSets for one endpoint of Kubernetes resource.
  - We should estimate zone size in double.



ListResourceRecordSets

Lists the resource record sets in a specified hosted zone.

`ListResourceRecordSets` returns up to 300 resource record sets at a time in ASCII order, beginning at a position specified by the `name` and `type` elements.

Implementation Challenges

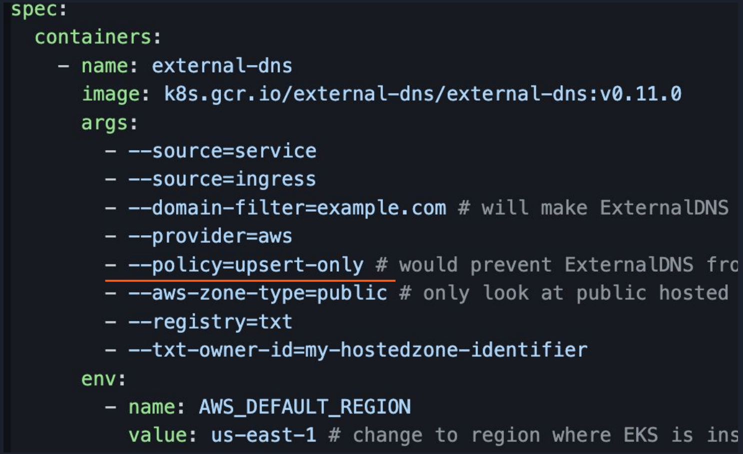
- Number of records reaches AWS Default Quota
- Route53 API throttling issue due to large number of API requests

Zone record limit

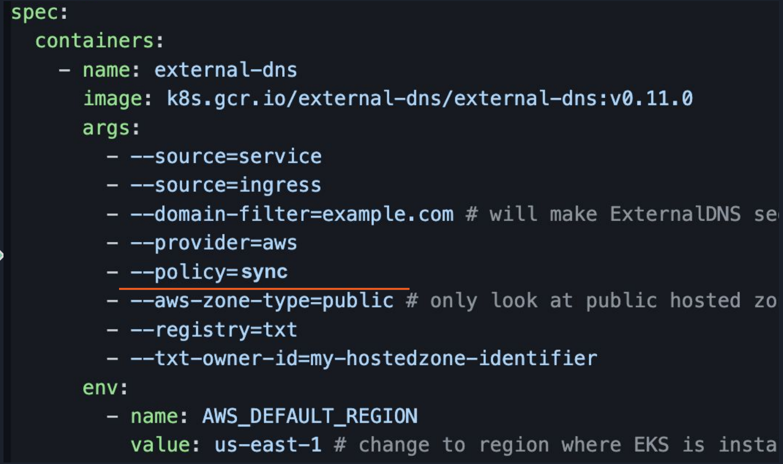
- Issue
  - As Salesforce service scales up, some zones hosted on AWS Route53 reach the AWS record limit of 10,000 records per zone (adjustable).
- Impact
  - No new records can be added to the zone
- Solution
  - Create zone size monitor and alert
    - Set up a real-time zone size monitoring and alert using AWS Python SDK and Lambda, an AWS serverless compute service, to send real-time zone size to AWS Cloudwatch. When the zone size reaches 80% of the Quota, engineer should request an higher Quota from AWS.
  - Reduce zone size - remove stale DNS reocrds
    - Deploy ExternalDNS pod using “sync” mode. Stale DNS records will be removed automatically by ExternalDNS.

Zone record limit (cont’d)

Example of modifying ExternalDNS policy configuration in manifest



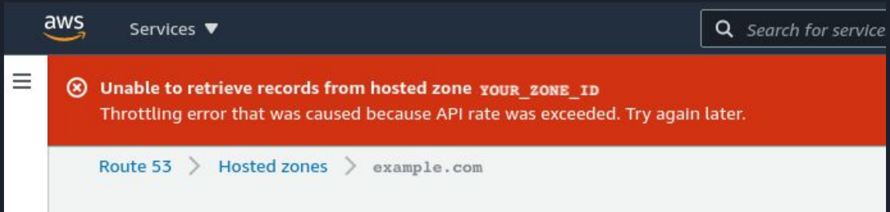
```
spec:
  containers:
  - name: external-dns
    image: k8s.gcr.io/external-dns/external-dns:v0.11.0
    args:
    - --source=service
    - --source=ingress
    - --domain-filter=example.com # will make ExternalDNS
    - --provider=aws
    - --policy=upsert-only # would prevent ExternalDNS fro
    - --aws-zone-type=public # only look at public hosted
    - --registry=txt
    - --txt-owner-id=my-hostedzone-identifier
    env:
    - name: AWS_DEFAULT_REGION
      value: us-east-1 # change to region where EKS is ins
```



```
spec:
  containers:
  - name: external-dns
    image: k8s.gcr.io/external-dns/external-dns:v0.11.0
    args:
    - --source=service
    - --source=ingress
    - --domain-filter=example.com # will make ExternalDNS se
    - --provider=aws
    - --policy=sync
    - --aws-zone-type=public # only look at public hosted zo
    - --registry=txt
    - --txt-owner-id=my-hostedzone-identifier
    env:
    - name: AWS_DEFAULT_REGION
      value: us-east-1 # change to region where EKS is insta
```

Route53 API Throttling Error

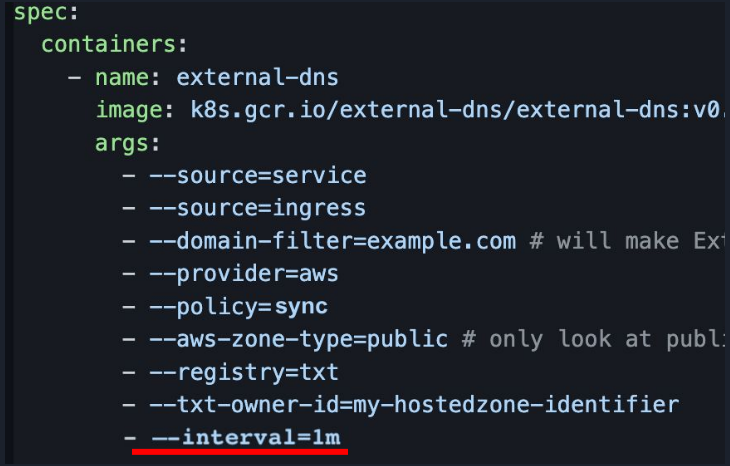
- Issue
  - Exhaust Route53 API rate limit when multiple external-dns instances perform bulk API calls to Route 53
  - Route53 throttling error
- Impact
  - ExternalDNS fails to update Route53 hosted zone
- Solution
  - Reduce Zone Size
    - Deploy ExternalDNS using “sync” mode
  - Enhance Cache Mechanism
    - Reduce the polling loop's synchronization interval at the possible cost of slower change propagation



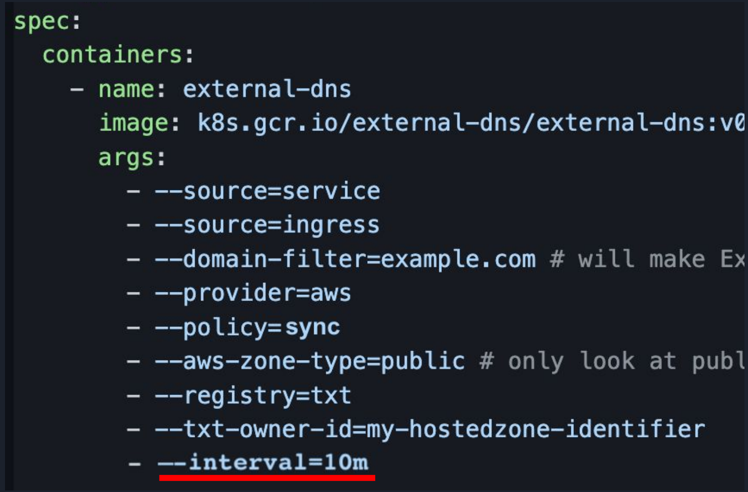
Unable to retrieve records from hosted zone YOUR_ZONE_ID
Throttling error that was caused because API rate was exceeded. Try again later.

Route53 API Frequency (Cont'd)

Example of modifying configuration of DNS update interval



```
spec:
  containers:
    - name: external-dns
      image: k8s.gcr.io/external-dns/external-dns:v0
      args:
        - --source=service
        - --source=ingress
        - --domain-filter=example.com # will make Ext
        - --provider=aws
        - --policy=sync
        - --aws-zone-type=public # only look at publ
        - --registry=txt
        - --txt-owner-id=my-hostedzone-identifier
        - --interval=1m
```



```
spec:
  containers:
    - name: external-dns
      image: k8s.gcr.io/external-dns/external-dns:v0
      args:
        - --source=service
        - --source=ingress
        - --domain-filter=example.com # will make Ex
        - --provider=aws
        - --policy=sync
        - --aws-zone-type=public # only look at publ
        - --registry=txt
        - --txt-owner-id=my-hostedzone-identifier
        - --interval=10m
```

Learnings

- Somewhere in your cloud (e.g. AWS), you may use Kubernetes with ExternalDNS making resources discoverable via public DNS servers
- ExternalDNS is not designed in a way that’s distributed and scalable
- Always be careful about the underlying rate limit in your cloud
- Design to scale out when a zone is updated by multiple ExternalDNS instances

Questions?

Thank you

Appendix

1. ExternalDNS Open source: https://github.com/kubernetes-sigs/external-dns
2. AWS Route53 Quotas: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/DNSLimitations.html#limits-api-entities