DNS Demystified

Global Traffic Management with Amazon Route 53

AWS re:Invent

aws

This Hour of Your Short Life

Theme: Large-Scale DNS

- Lightning Intro to DNS & Core Route 53 Features
- DNS Resolution inside EC2
- What's New in Route 53
- Customer Case Study: MuleSoft
- Route 53 Advanced Traffic Management
- DNS Operational Excellence

Lightning Tour of

DNS & Core Route 53 Features

What Problem Does DNS Solve?

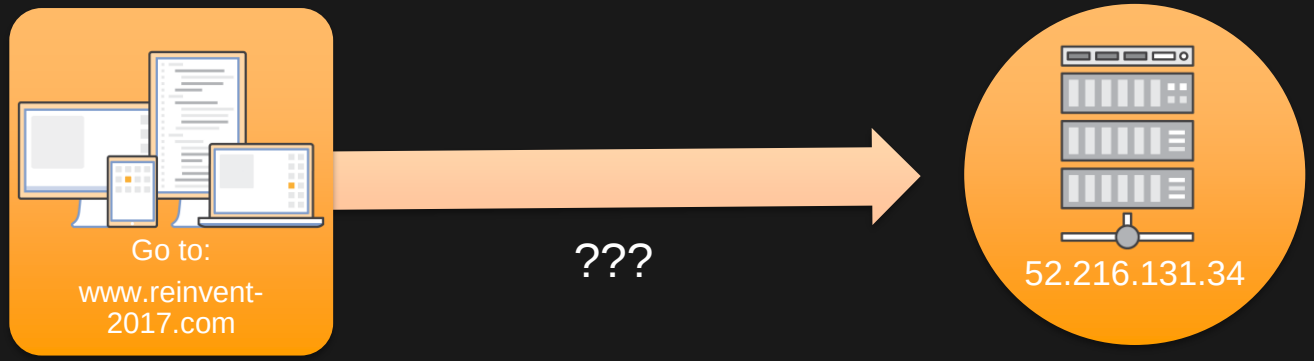
How Does It Work?

How Do You Set It up?

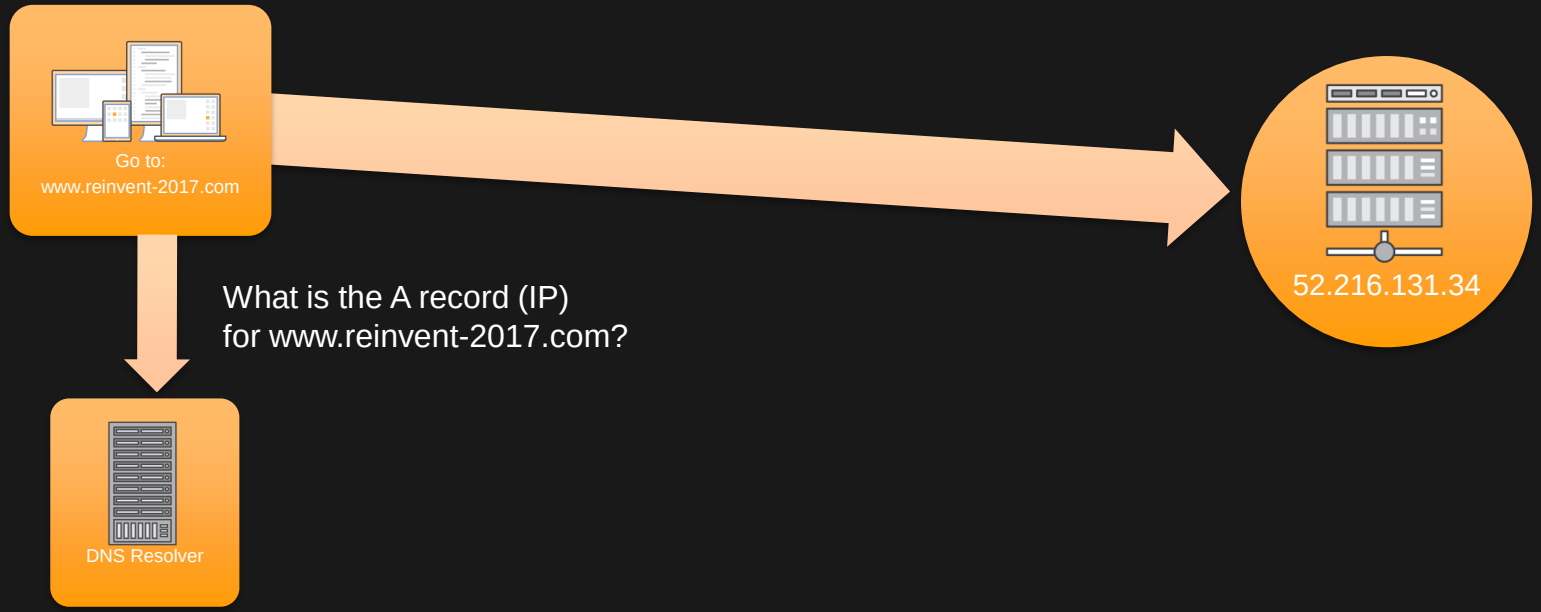
AWS re:Invent

aws

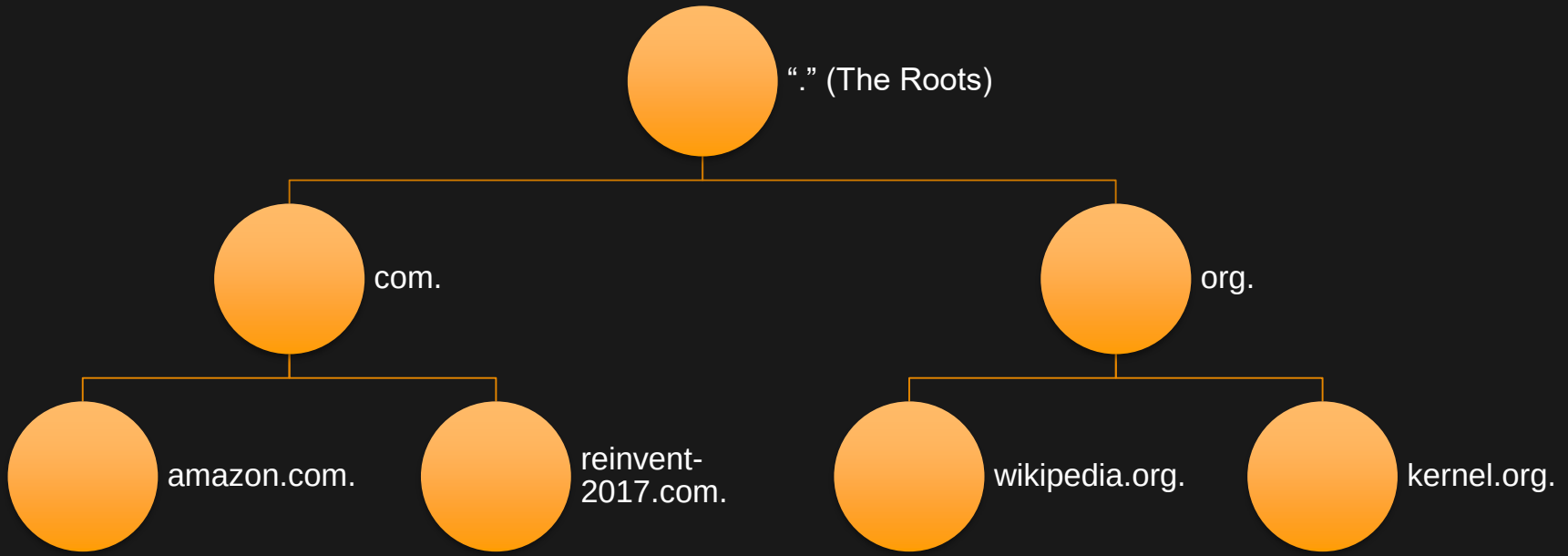
The Problem DNS Solves



The Problem DNS Solves



How Do the Resolvers Know?



How Does a Resolver Work?

root nameservers
(Various)

- A record for www.reinvent-2017.com?
- NS records for com

com nameservers
(Verisign)

- A record for www.reinvent-2017.com?
- NS records for reinvent-2017.com

reinvent-2017.com
Nameservers
(Route 53)

- A record for www.reinvent-2017.com?
- Yes! 52.216.131.34

Recursive Resolution

```
ubuntu@ip-172-31-9-203:~$ dig +trace www.reinvent-2017.com

; <<>> DiG 9.9.5-3ubuntu0.16-Ubuntu <<>> +trace www.reinvent-2017.com
;; global options: +cmd
.                   518400      IN    NS    A.ROOT-SERVERS.NET.
.                   518400      IN    NS    B.ROOT-SERVERS.NET.
.                   518400      IN    NS    C.ROOT-SERVERS.NET.
.                   518400      IN    NS    D.ROOT-SERVERS.NET.
.                   518400      IN    NS    E.ROOT-SERVERS.NET.
.                   518400      IN    NS    F.ROOT-SERVERS.NET.
.                   518400      IN    NS    G.ROOT-SERVERS.NET.
.                   518400      IN    NS    H.ROOT-SERVERS.NET.
.                   518400      IN    NS    I.ROOT-SERVERS.NET.
.                   518400      IN    NS    J.ROOT-SERVERS.NET.
.                   518400      IN    NS    K.ROOT-SERVERS.NET.
.                   518400      IN    NS    L.ROOT-SERVERS.NET.
.                   518400      IN    NS    M.ROOT-SERVERS.NET.
;; Received 239 bytes from 172.31.0.2#53(172.31.0.2) in 192 ms
```

Recursive Resolution

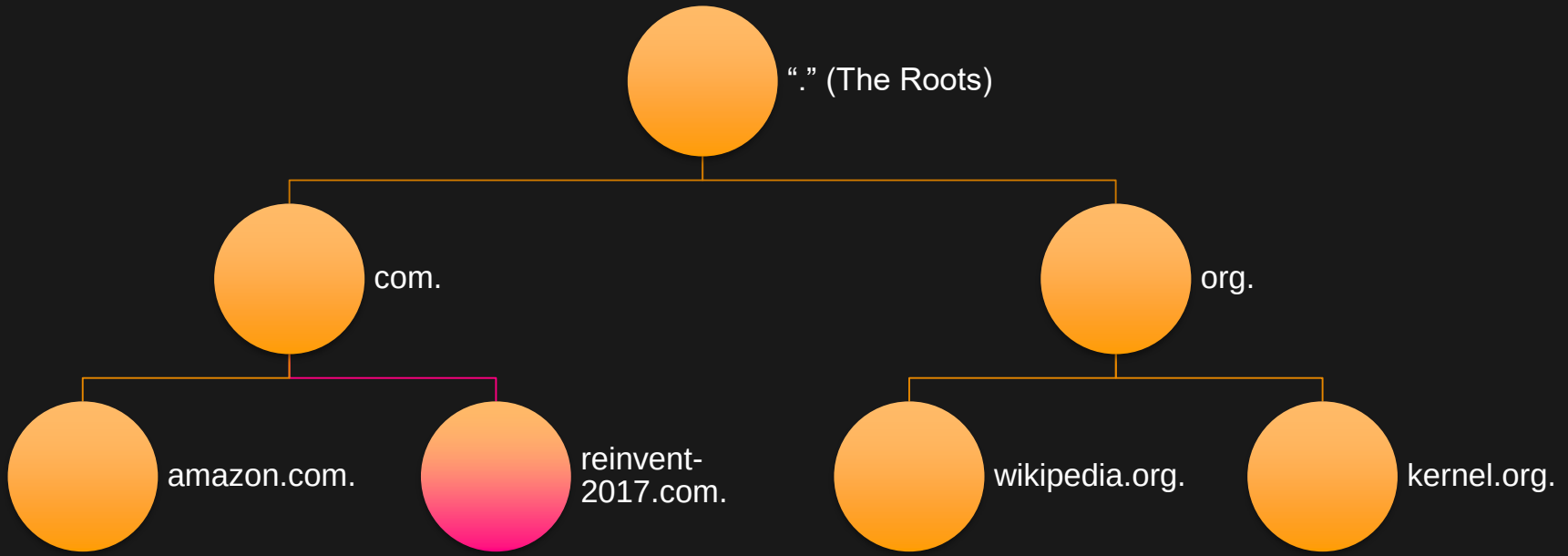
```
com.            172800    IN    NS    b.gtld-servers.net.
com.            172800    IN    NS    e.gtld-servers.net.
com.            172800    IN    NS    j.gtld-servers.net.
com.            172800    IN    NS    i.gtld-servers.net.
com.            172800    IN    NS    l.gtld-servers.net.
com.            172800    IN    NS    k.gtld-servers.net.
com.            172800    IN    NS    g.gtld-servers.net.
com.            172800    IN    NS    f.gtld-servers.net.
com.            172800    IN    NS    m.gtld-servers.net.
com.            172800    IN    NS    h.gtld-servers.net.
com.            172800    IN    NS    a.gtld-servers.net.
com.            172800    IN    NS    d.gtld-servers.net.
com.            172800    IN    NS    c.gtld-servers.net.
;; Received 1181 bytes from 202.12.27.33#53(M.ROOT-SERVERS.NET) in 88 ms
```

Recursive Resolution

```
reinvent-2017.com.  172800    IN   NS   ns-1900.awsdns-45.co.uk.
reinvent-2017.com.  172800    IN   NS   ns-1526.awsdns-62.org.
reinvent-2017.com.  172800    IN   NS   ns-709.awsdns-24.net.
reinvent-2017.com.  172800    IN   NS   ns-250.awsdns-31.com.
;; Received 704 bytes from 192.41.162.30#53(l.gtld-servers.net) in 38 ms
```

```
www.reinvent-2017.com.   60   IN   A    52.216.131.34
reinvent-2017.com.  172800    IN   NS   ns-1900.awsdns-45.co.uk.
reinvent-2017.com.  172800    IN   NS   ns-1526.awsdns-62.org.
reinvent-2017.com.  172800    IN   NS   ns-250.awsdns-31.com.
reinvent-2017.com.  172800    IN   NS   ns-709.awsdns-24.net.
;; Received 203 bytes from 205.251.192.250#53(ns-250.awsdns-31.com) in 9 ms
```

DNS for a Website



What Is Route 53?

- Route 53 Public DNS (zone hosting)
    - 100% availability SLA
    - Currently 76x PoPs worldwide
- Route 53 Registrar
- Route 53 Private DNS (inside EC2)
- Route 53 Health Checking and Traffic Management
- Public APIs and CLIs for automated operation
    - SDK, AWS CLI, CLI53, Boto, Denominator …

DNS for a Website

```
a45e60d6bfe1:~ gavinmc$ aws route53domains check-domain-availability --domain-name reinvent-2017.com
{
    "Availability": "AVAILABLE"
}

a45e60d6bfe1:~ gavinmc$ aws route53domains register-domain --cli-input-json file://route53domains-skeleton.json \
                        --domain-name reinvent-2017.com
{
    "OperationId": "eae4c84c-606c-4453-89d8-e752ff614337"
}

a45e60d6bfe1:~ gavinmc$ aws route53domains list-operations
{
    "Operations": [
        {
            "Status": "IN_PROGRESS",
            "Type": "REGISTER_DOMAIN",
            "SubmittedDate": 1510208466.623,
            "OperationId": "eae4c84c-606c-4453-89d8-e752ff614337"
        }
    ]
}
```

Zone Created Automatically

```
a45e60d6bfe1:~ gavinmc$ aws route53 list-hosted-zones-by-name
{
    "HostedZones": [
        {
            "ResourceRecordSetCount": 2,
            "CallerReference": "RISWorkflow-RD:8498650d-2e08-4b12-b099-8c30e4ff5f61",
            "Config": {
                "Comment": "HostedZone created by Route53 Registrar",
                "PrivateZone": false
            },
            "Id": "/hostedzone/Z2PBIU2YUKVPKA",
            "Name": "reinvent-2017.com."
        }
    ],
    "IsTruncated": false,
    "MaxItems": "100"
}
```

Create Records

```
a45e60d6bfe1:~ gavinmc$ aws route53 change-resource-record-sets --cli-input-json file://change-rrset.json \
                            --hosted-zone-id Z2PBIU2YUKVPKA
{
    "ChangeInfo": {
        "Status": "PENDING",
        "Comment": "Creating_a_first_record",
        "SubmittedAt": "2017-11-09T06:43:01.339Z",
        "Id": "/change/C23K4GSRVZTUAG"
    }
}

a45e60d6bfe1:~ gavinmc$ aws route53 get-change --id C23K4GSRVZTUAG
{
    "ChangeInfo": {
        "Status": "INSYNC",
        "Comment": "Creating_a_first_record",
        "SubmittedAt": "2017-11-09T06:43:01.339Z",
        "Id": "/change/C23K4GSRVZTUAG"
    }
}
```

What Was in the JSON Doc?

```
{
    "ChangeBatch": {
        "Comment": "Creating a first record",
        "Changes": [
            {
                "Action": ”CREATE",
                "ResourceRecordSet": {
                    "Name": "www.reinvent-2017.com",
                    "Type": "A",
                    "SetIdentifier": "1",
                    "Weight": 0,
                    "TTL": 900,
                    "ResourceRecords": [
                        {
                            "Value": ”52.216.131.34"
                        }
                    ]
                }
```

DNS inside EC2

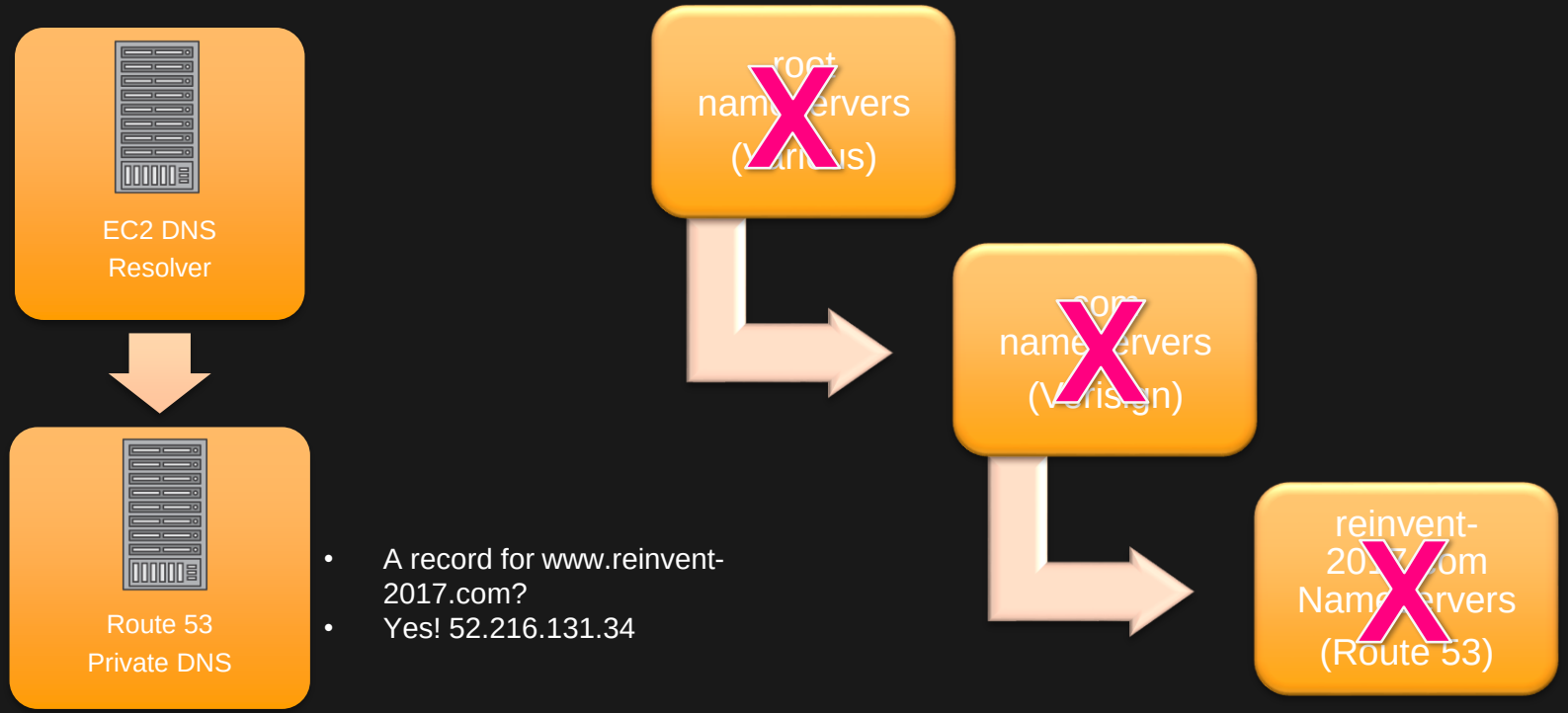
Amazon-Provided DNS

Route53 Private DNS

DNS Resolution in VPCs

- “Amazon-Provided DNS” in EC2
    - Public DNS resolver
    - Internal instance hostnames
- Route 53 Private DNS
    - Create/Associate Private DNS zone
    - Cross-region, cross-account zones

How Does Private DNS Work?

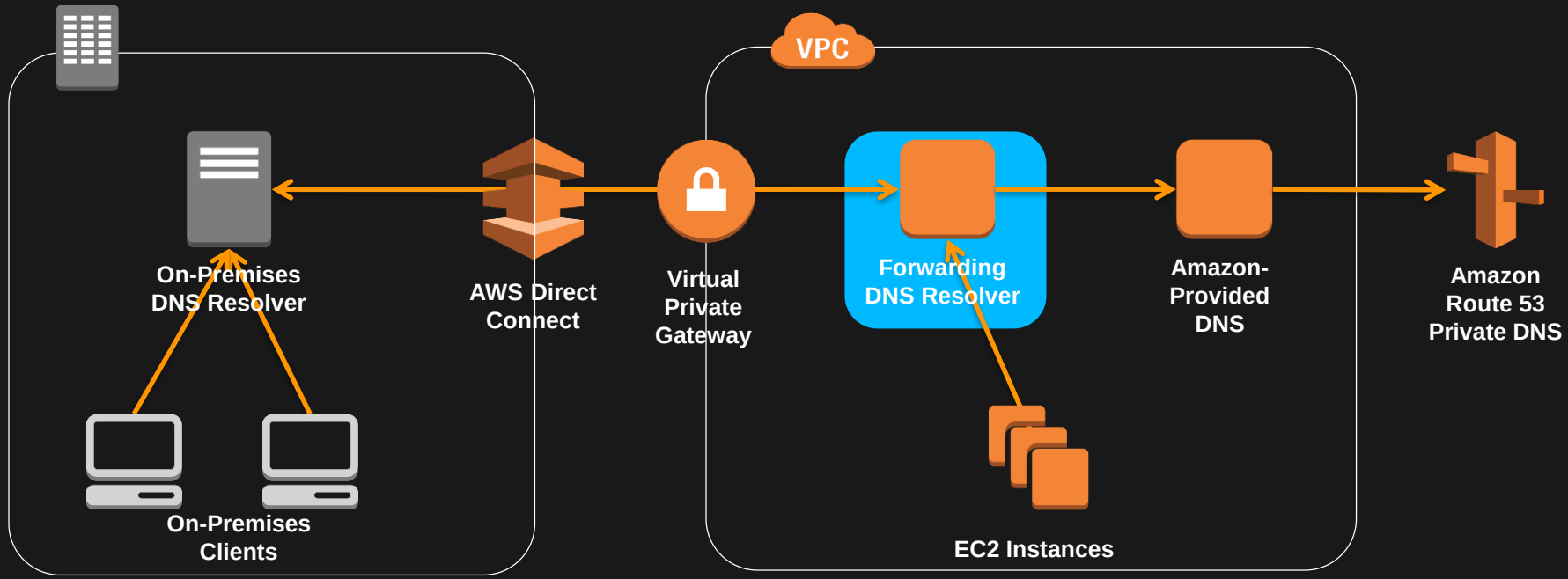


- A record for www.reinvent-2017.com?
- Yes! 52.216.131.34

DNS In "Hybrid" Clouds

- DirectConnect or VPN between data center and VPC
- A single view of DNS for data center hosts and VPC instances
- Solutions
    - Forwarding Resolver instances, e.g., Unbound, DNSMasq, Simple AD
    - Lambda-based sync from DC into Route53 Private DNS
    - Per-instance conditional forwarding, e.g., Unbound
- Whitepaper
    - *Hybrid Cloud DNS Solutions For Amazon VPC* (October 2017)

Hybrid Cloud – Forwarding Instances



What's New in Route 53?

AWS re:Invent

aws

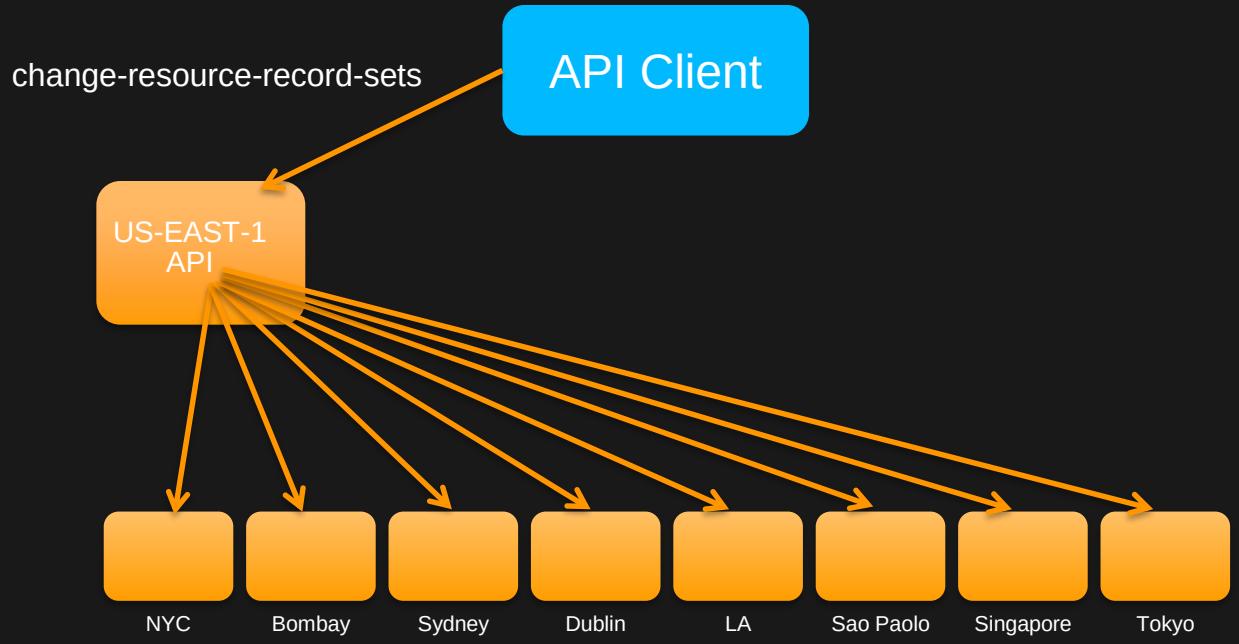
What's New?

- Geo-Proximity Routing Type
- Public Route 53 Query Logs
- Registrar Bulk (100x) Domain Transfer
- Multi-Response Answers for Weighted Round Robin
- CAA Record Support
- Service Limits API

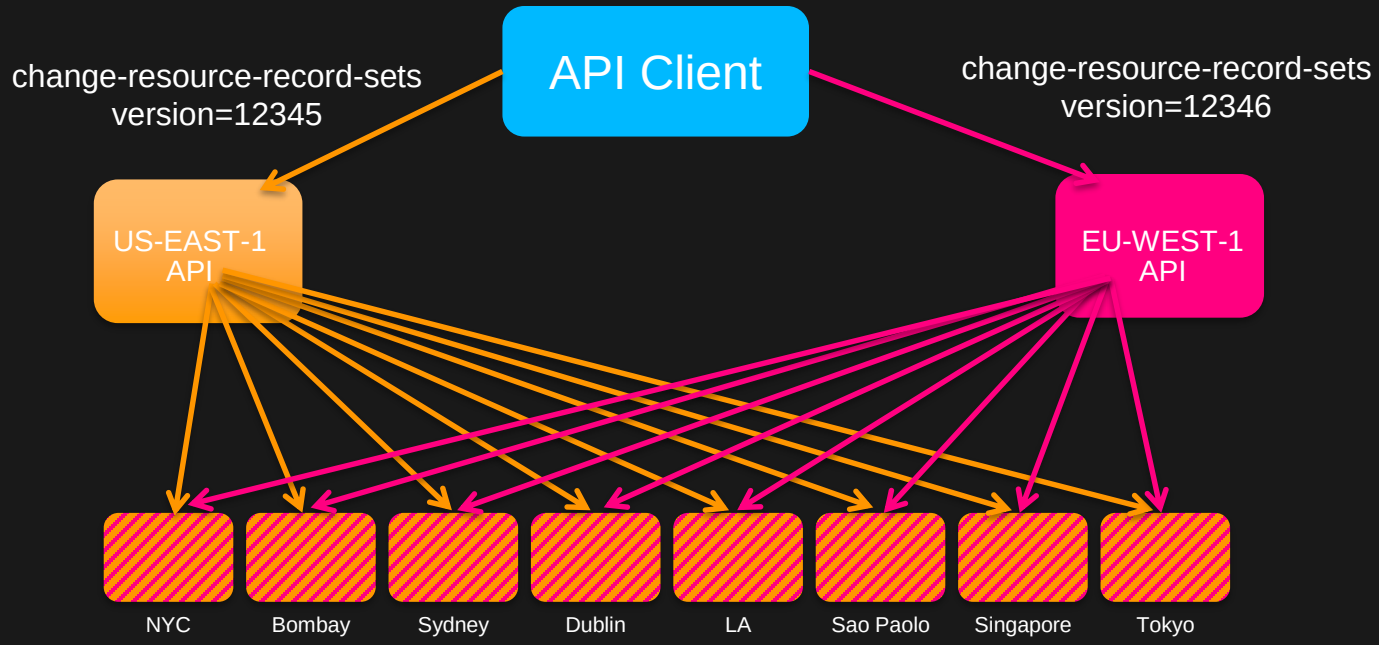
API and Change Propagation

- Route 53 100% SLA on DNS Queries
- Making DNS record changes: standard API in US-EAST-1
- Optional: second Route 53 API endpoint available in EU-WEST-1
- Tradeoffs in using standard versus multi-region API
- Customer use case: MuleSoft

Route 53 Multi-Region API



Route 53 Multi-Region API



MuleSoft's Propagation-Sensitive Use Case

About MuleSoft

Company overview

- Help organizations change and innovate faster by making it easy to connect the world's applications, data, and devices
- Founded in 2006, HQ: San Francisco
- 1,000+ employees worldwide

Business momentum

- IPO in March 2017
- Over 1,000 customers in approximately 60 countries
- Global presence: offices in five continents
- Global footprint: sales, services, support

MuleSoft's Anypoint Platform

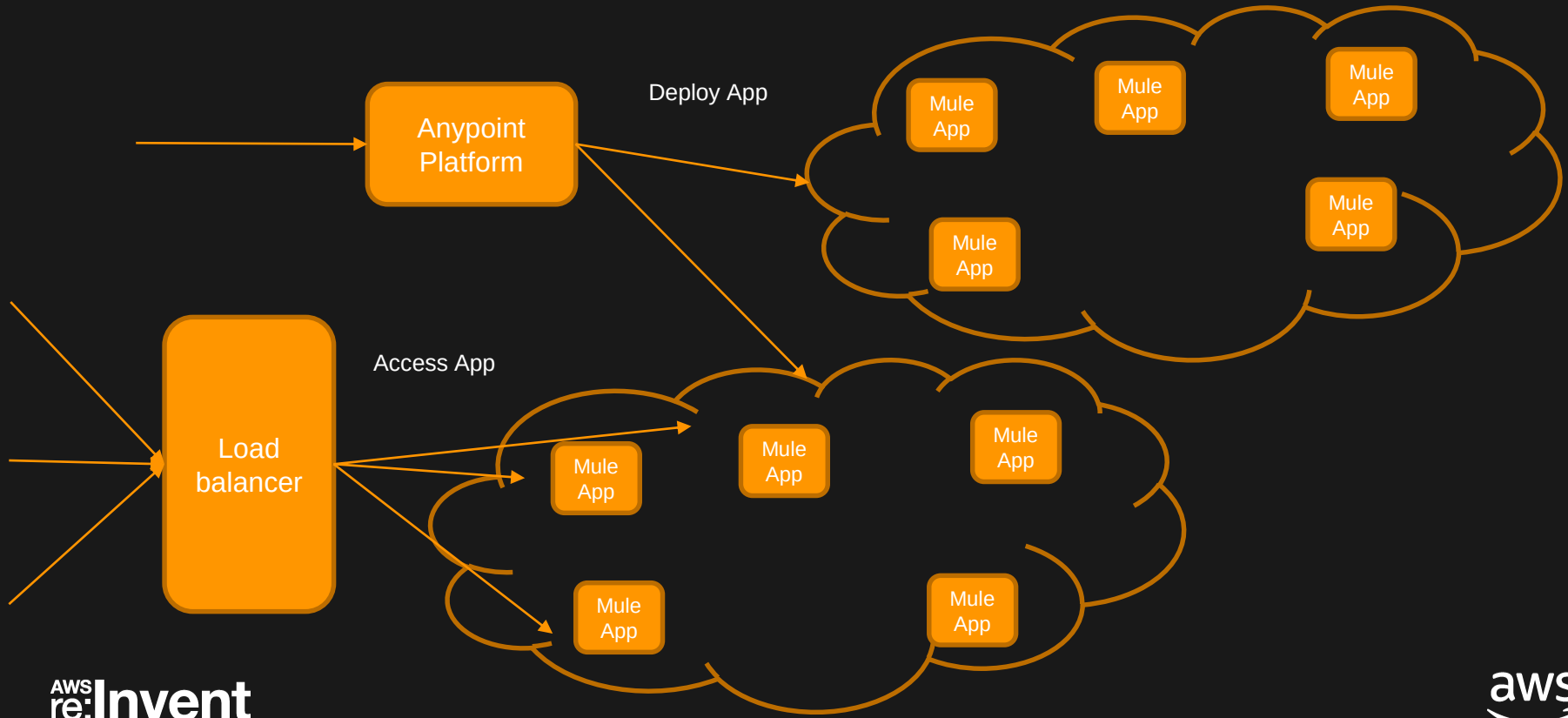
- Multi-tenant hybrid integration platform for APIs and integration
- iPaaS:
    - 99.99% uptime
    - One-click multi-DC HA
    - One-click scalability
    - Zero down-time upgrades
    - Smart healing
    - Built-in security

MuleSoft's Anypoint Platform in AWS

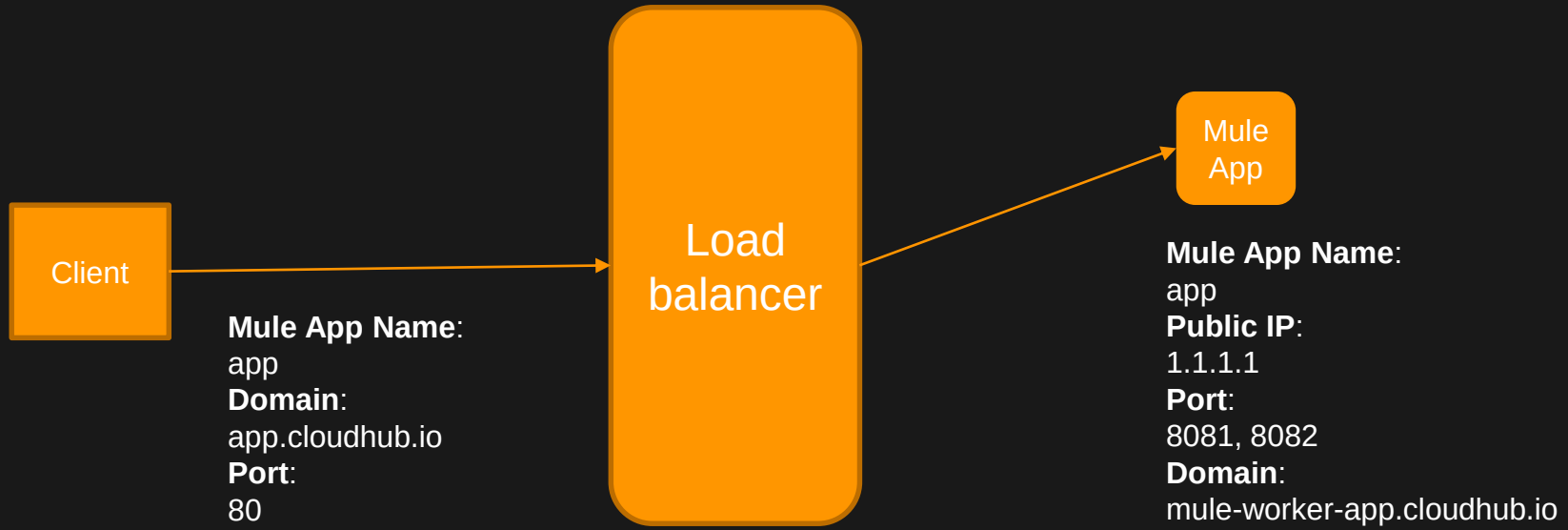
- Build on top of EC2, CloudWatch, SQS, S3, Route 53, etc.
- Run instances in 12 regions
- Active instances over 25,000
- CloudWatch Alarms over 30,000

- Daily application deployment over 9,000
    - EC2 Instance Provisions: 4,500
    - EC2 Instance Shutdown: 4,000
- **Daily Route 53 DNS Record Operations**
    - **Insert Resource Records: 5,000**
    - **Delete Resource Records: 4,000**

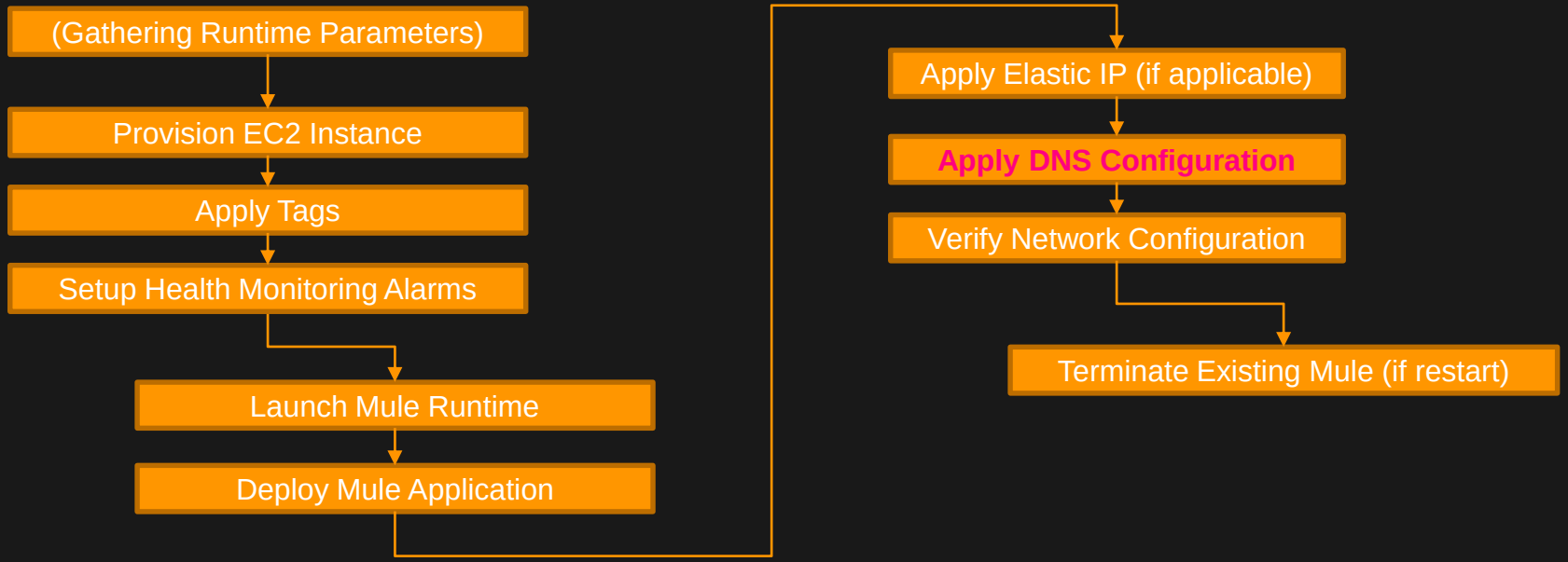
MuleSoft's Anypoint Platform in AWS



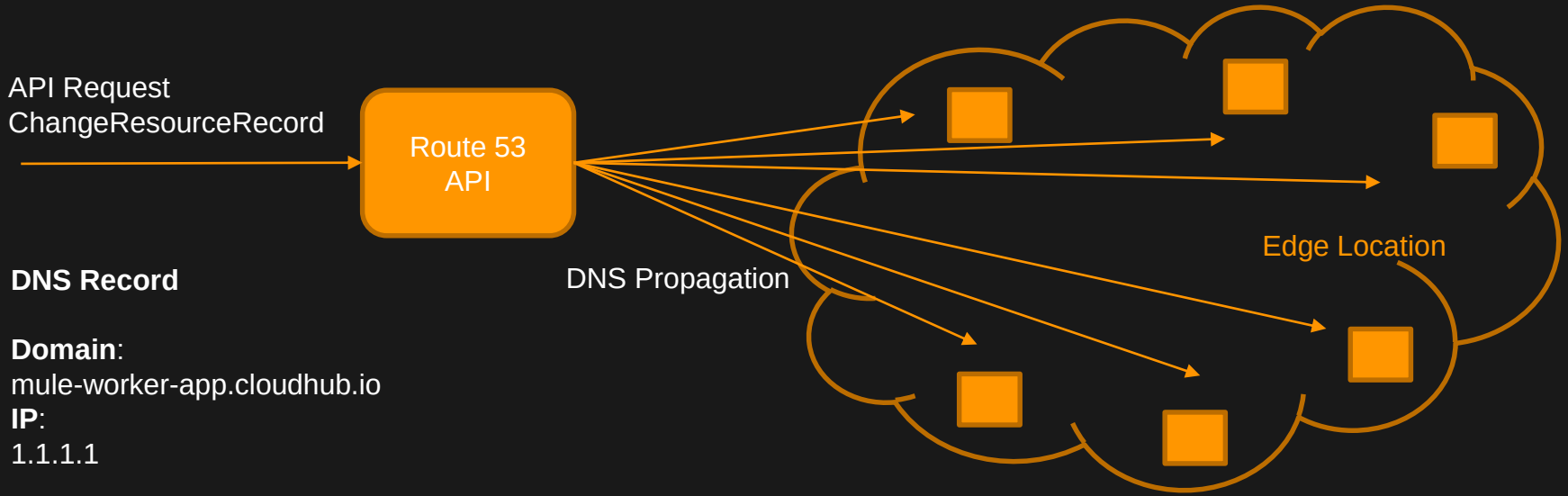
MuleSoft's Anypoint Platform in AWS



Mule App Deployment



Mule App Deployment—Route 53



DNS Record

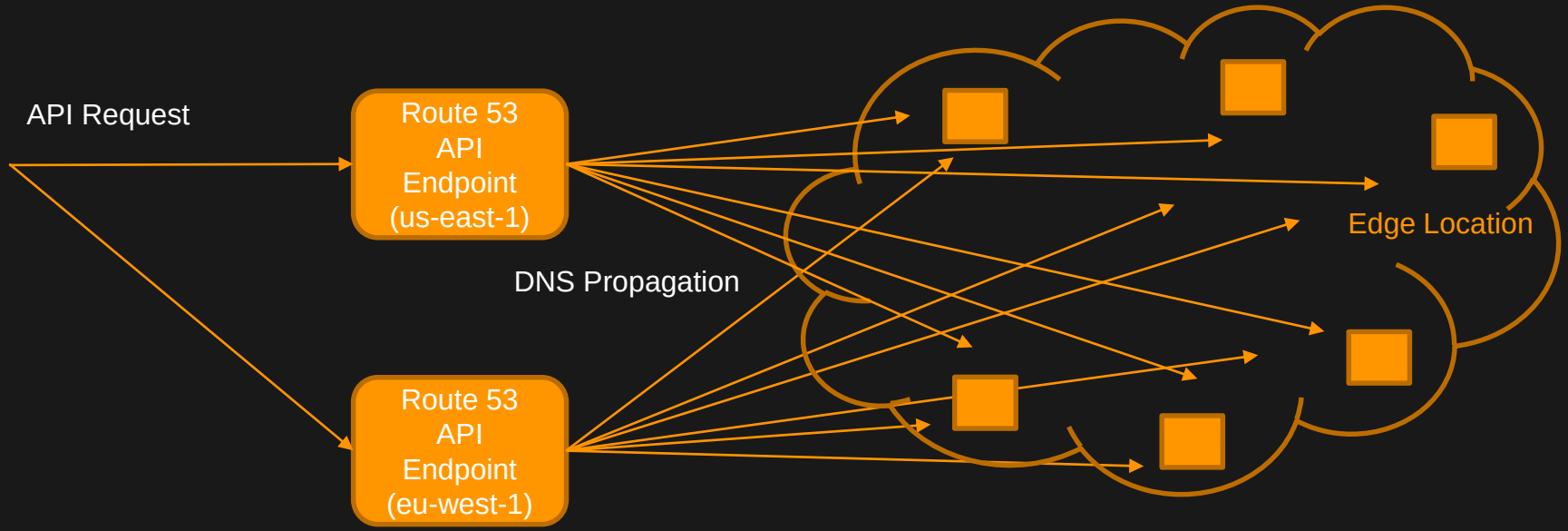
Domain:
mule-worker-app.cloudhub.io
IP:
1.1.1.1

Importance of Route 53

- Potential failure mode of Route 53
    - Unreachable API
    - Slow DNS records propagation

- Impact on MuleSoft's Anypoint Platform
    - Mule application deployment failures
    - Unresolvable Mule applications

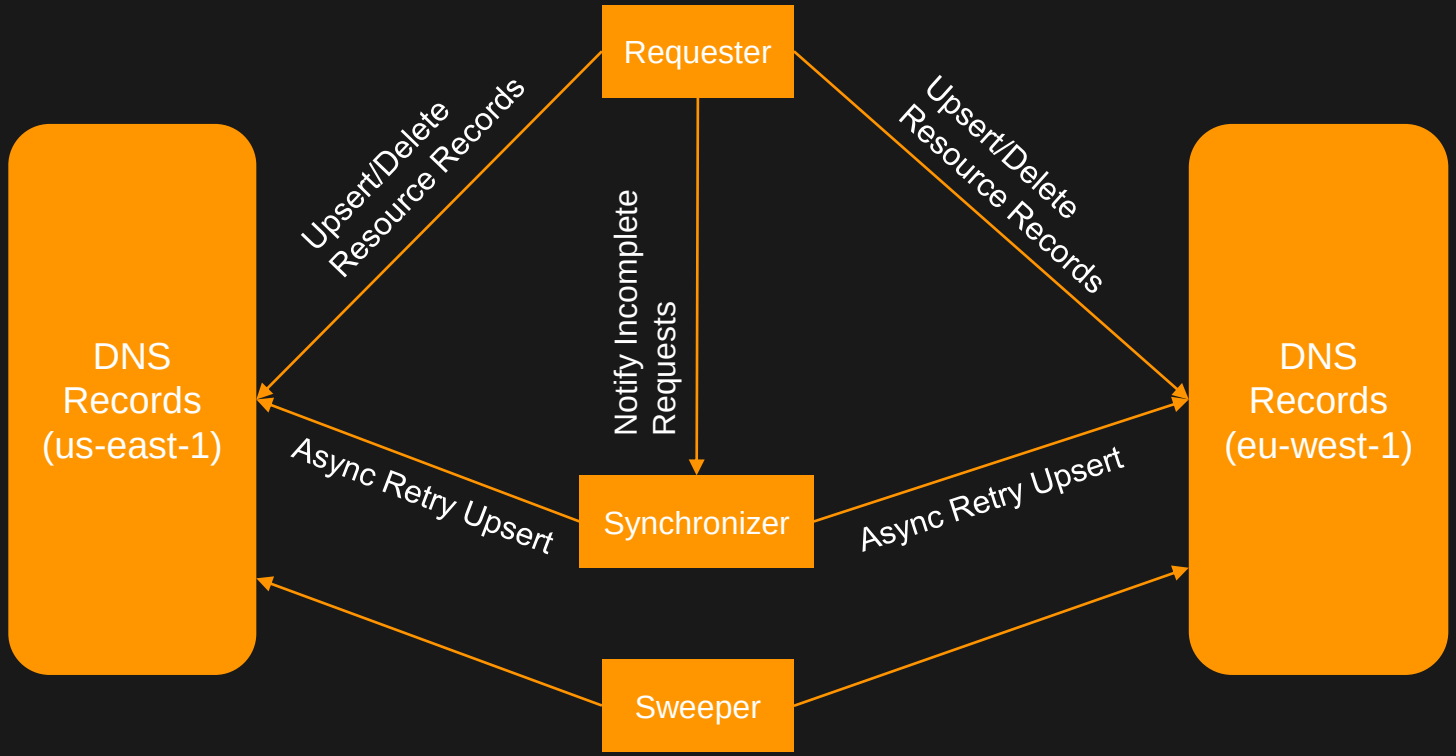
Extra Resilience of Route 53 Services



Options on Multi-Region Route 53

- How it works
  - Two independent Route 53 API endpoints in two regions
  - Operate separately and propagate two sets of versioned DNS records
  - DNS query gets the higher-versioned record
- Options
  - Failover (not recommended)
    - Use the secondary one only if the primary doesn't respond
      - Distinguish various types of failures
      - Deal with two sets of DNS records
  - Redundancy
    - Goal is to keep records in both API regions

Utilization Multi-Region Route 53 Services



Iterate and remove garbage records (infrequently)

Implementation Limitations and Concerns

- Independence of multi-region Route 53 servers
    - High availability and redundancy
    - Additional synchronization for consistency between the two regions

- Migration from current Route 53 services to multi-region Route 53
    - Backwards compatible
    - Complicated rollback procedures in case of problems

- Rare theoretical case of delayed DNS update
    - The Route 53 services in two regions experience problems in reachability and propagation respectively in turn

Conclusion

- Increased uptime and resilience for MuleSoft's Anypoint Platform
- Real-time alert on Route 53 DNS status
- 100% success rate on application DNS resolve so far

Multi-Region API Recap

- Twin APIs provide redundant paths to provision records
- Best practice is to write all changes to all zones (constant work)
- Not currently released. Private beta only.
- Complexity trade-offs are only appropriate for customers with very sensitive, synchronous workflows

Advanced Traffic Management

DNS for High Availability

DNS for Failover

DNS for Global Least Latency

DNS for Regional Content

AWS re:Invent

aws

Traffic Management

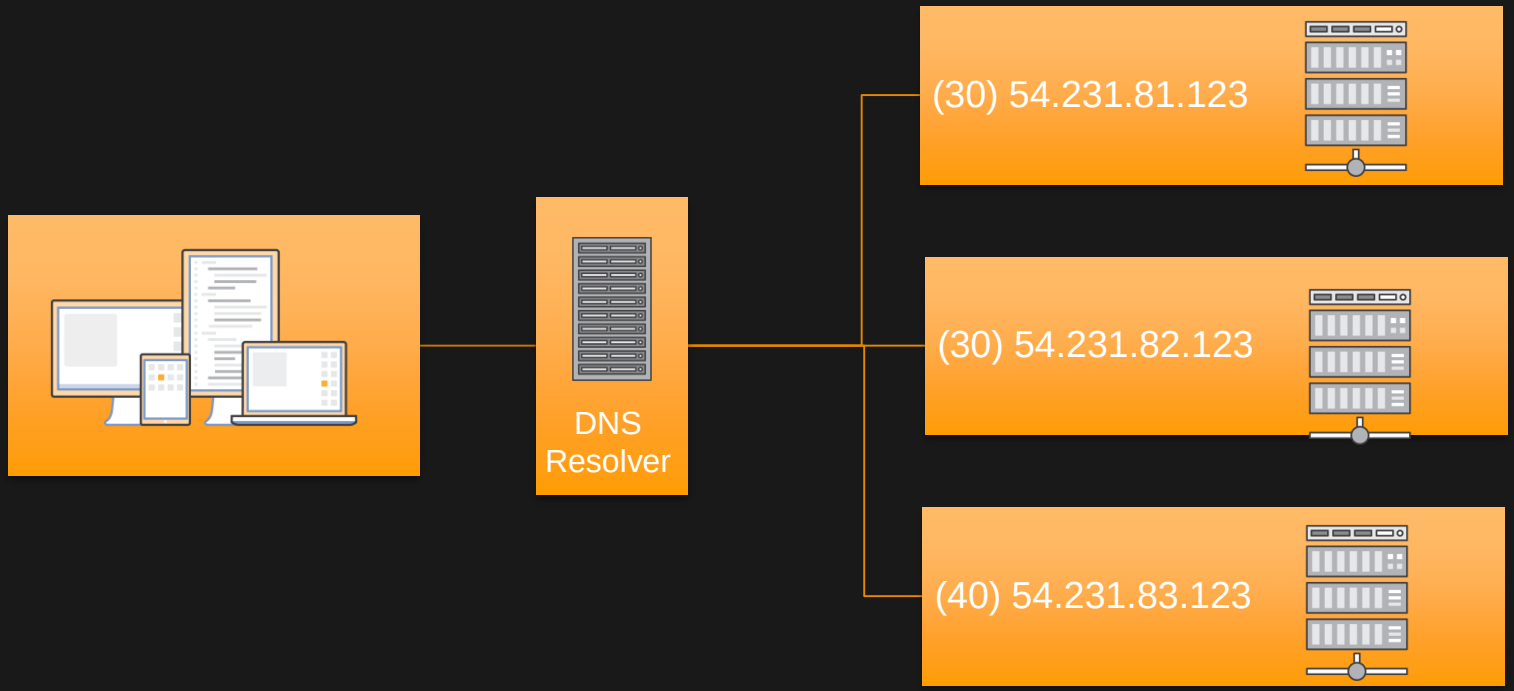
- DNS is an abstraction layer
- Smart answers to manage traffic
- Let’s look at some customer use cases

DNS for Load Balancing

- *Customer: I run an online bookshop and I want to be able to run my service active-active across multiple redundant data centers.*

- *Feature: Weighted Round Robin records will return each endpoint IP in the ratio you determine.*

DNS for Load Balancing

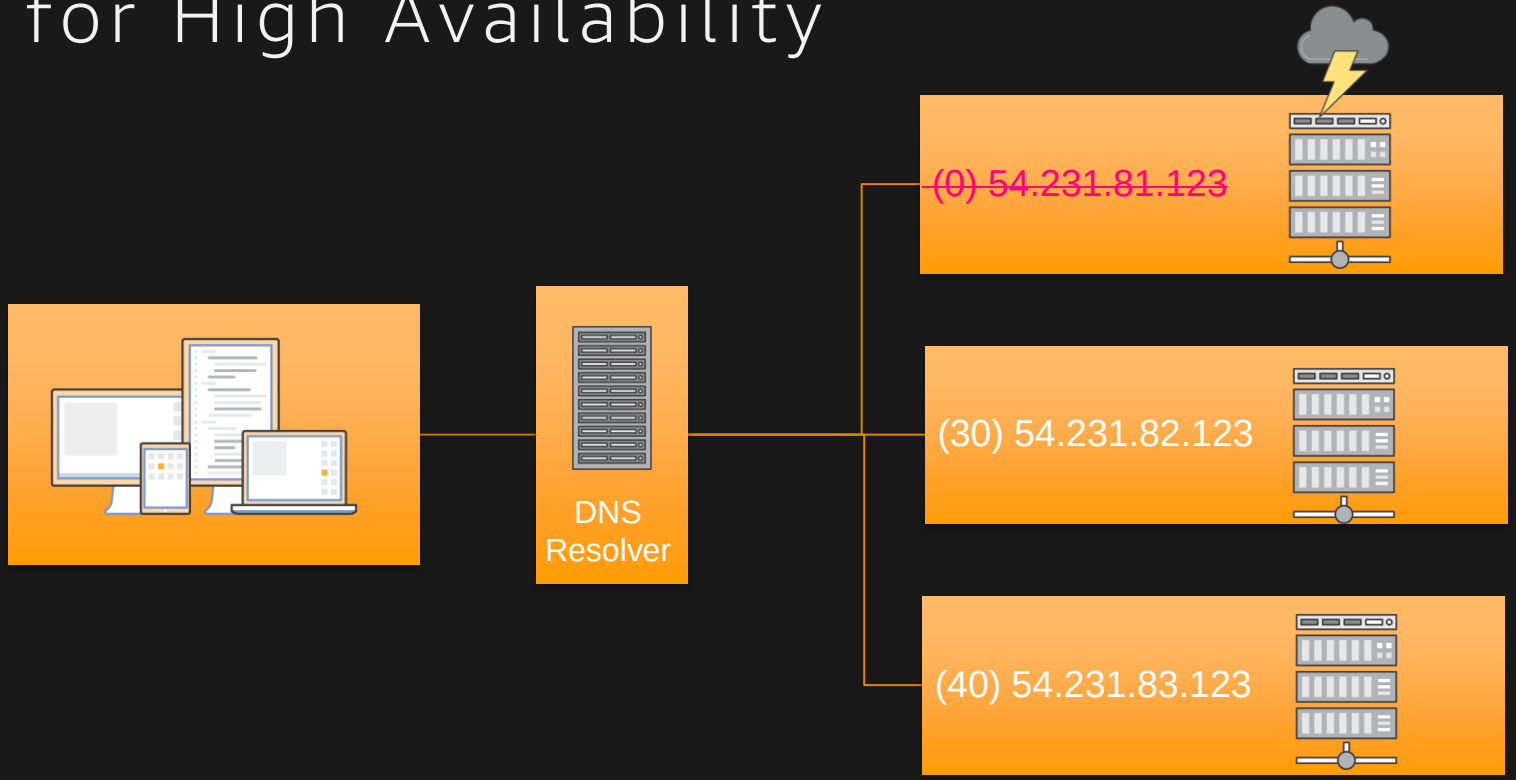


DNS for High Availability

- *Customer: Now, how do I recover automatically when one of my endpoints or data centers fails?*

- *Feature: Route 53 Health Checks will poll each configured endpoint regularly, establish which ones are healthy and only return those.*

DNS for High Availability

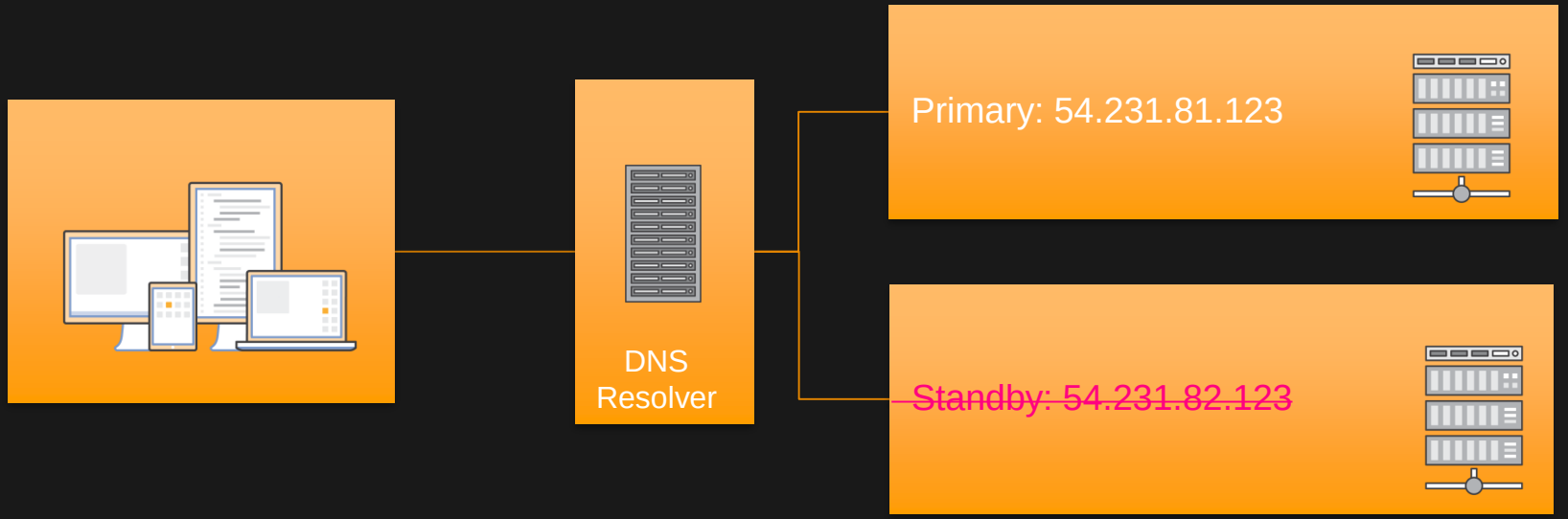


DNS for Failover

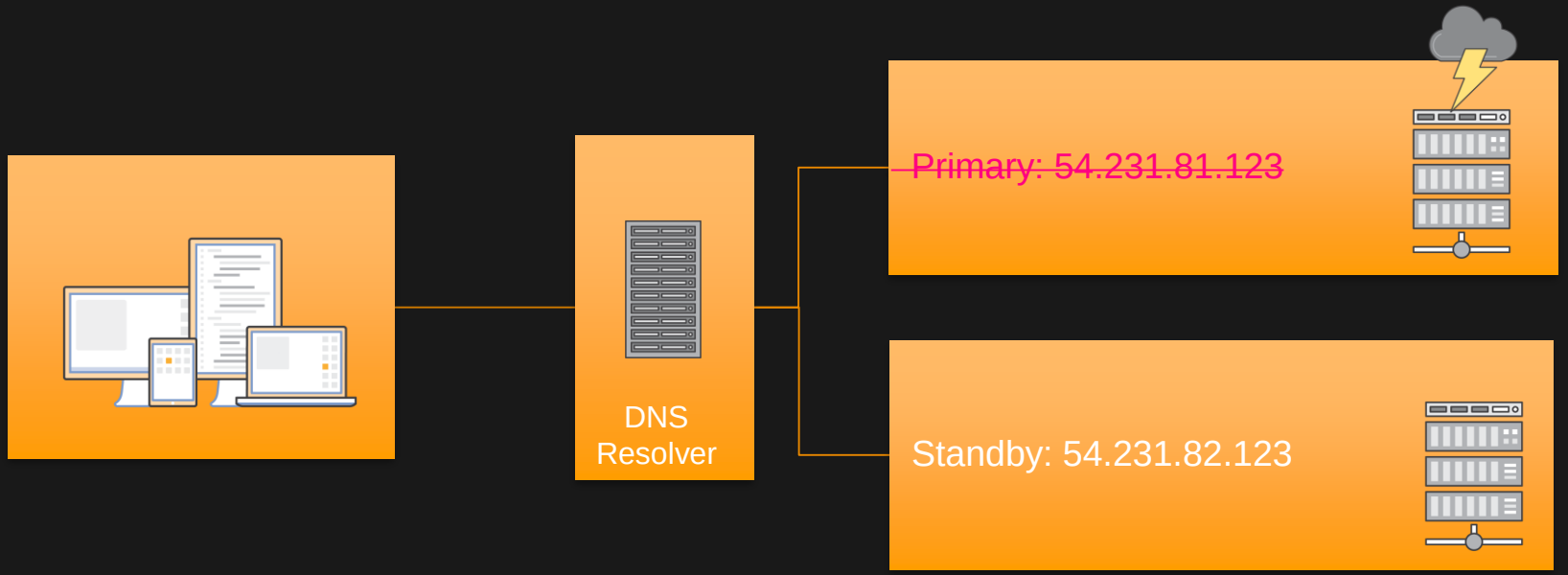
- *Customer: I'm running a web service. If it fails, I need to fail over to a second standby system (for example, a second stack, an S3 bucket website).*

- *Feature: Route 53 Failover routing type with health checks.*

DNS for Failover



DNS for Failover



DNS for Low Latency

- *Customer: I built my web service out in multiple EC2 Regions. Responsiveness is critical. How do I route customers to the lowest latency EC2 region?*

- *Feature: Route 53 Latency-Based Routing (LBR) Policy determines the IP address of the client and always returns the endpoints in the lowest latency (healthy) region for that client IP.*

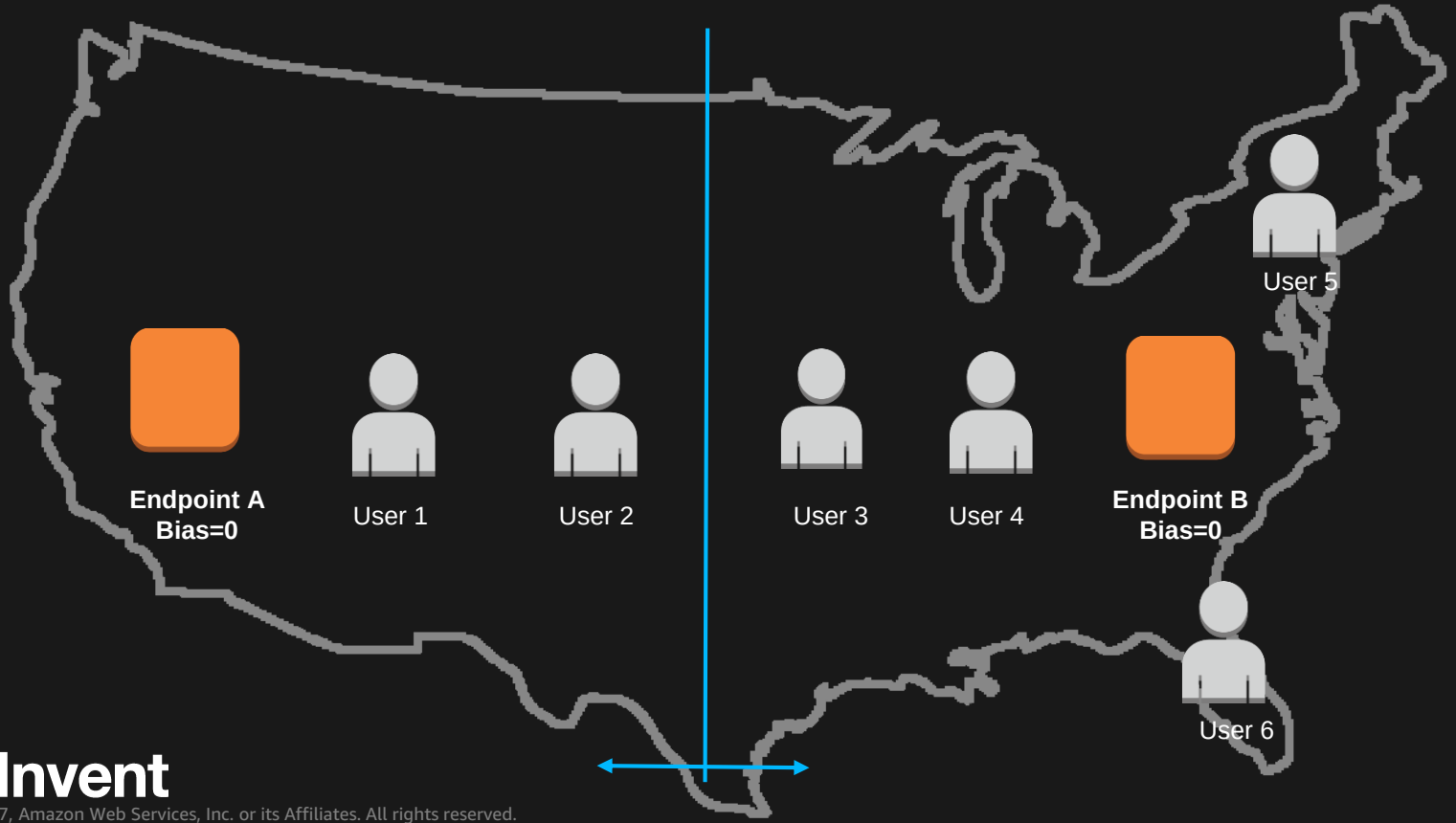
DNS for Low Latency (LBR)

DNS for Low Latency (Geoproximity)

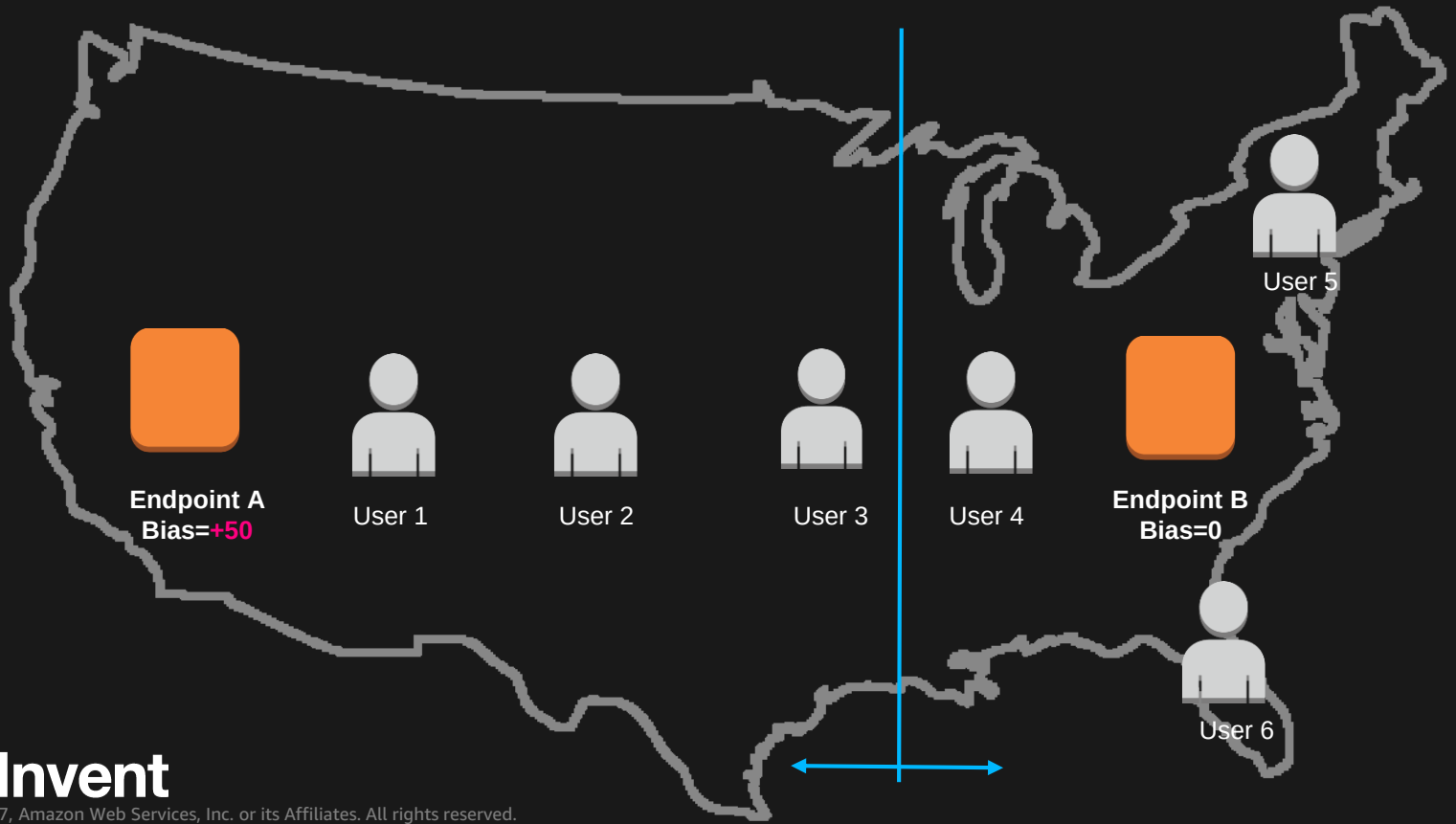
- *Customer: What if my endpoints are not inside EC2 Regions? What if I want to shift traffic between regions while preferring least latency?*

- *Feature: Route 53 Geo-Proximity estimates the map location of the client and returns endpoints which are at the minimal distance from the client.*

DNS for Low Latency (Geoproximity)



DNS for Low Latency (GeoProximity)

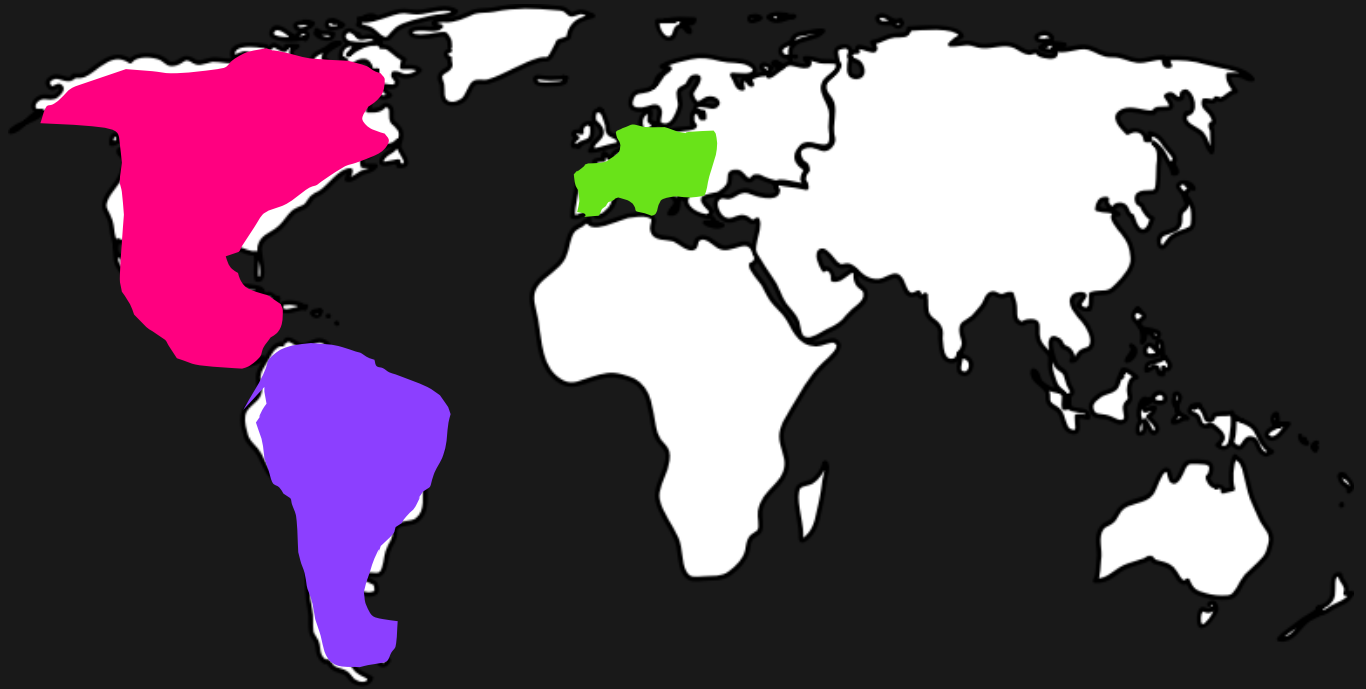


DNS for Regional Content

- *Customer: I need to offer different content based on the location of the client.*

- *Feature: Route 53 Geolocation determines the continent, country, and state (in USA) of the client. Customers can choose which locations go to which endpoints.*

DNS for Regional Content

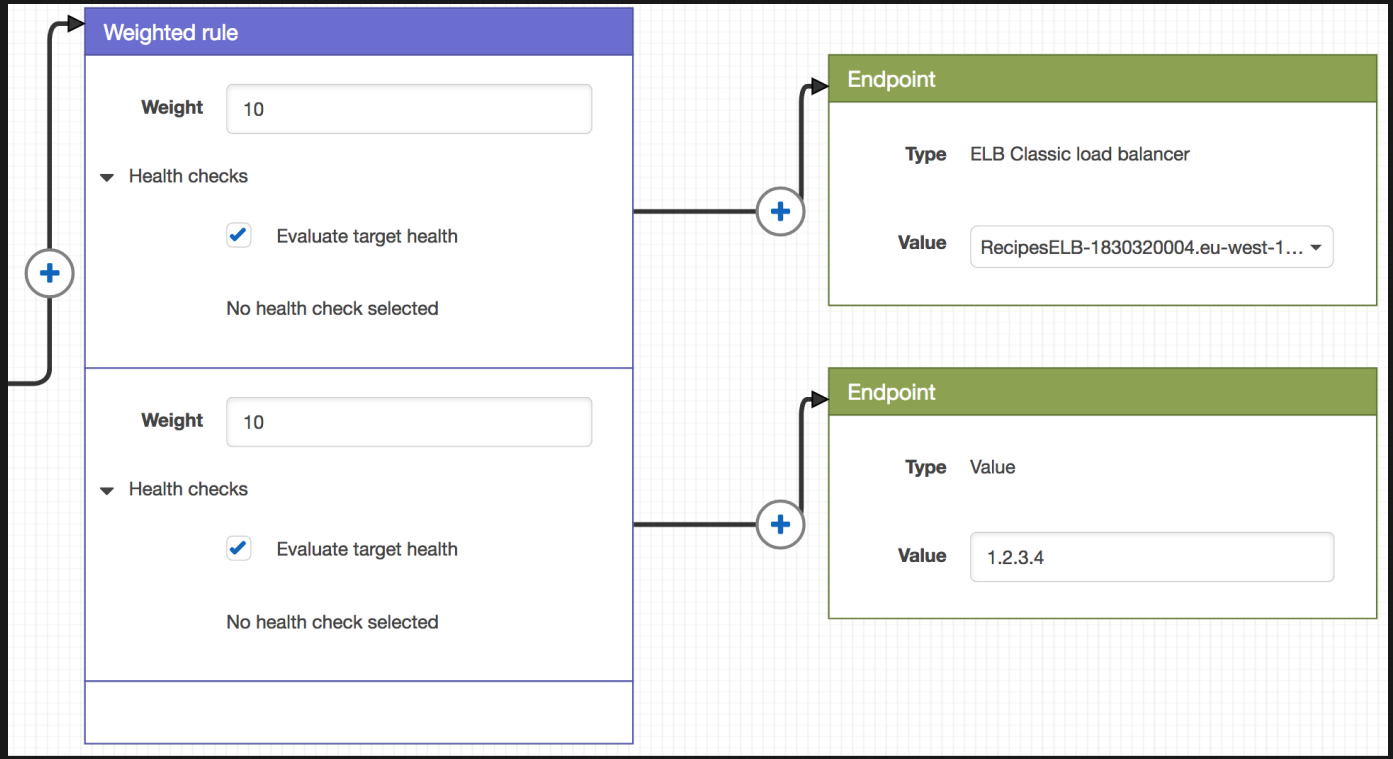


Route 53 Traffic Flow

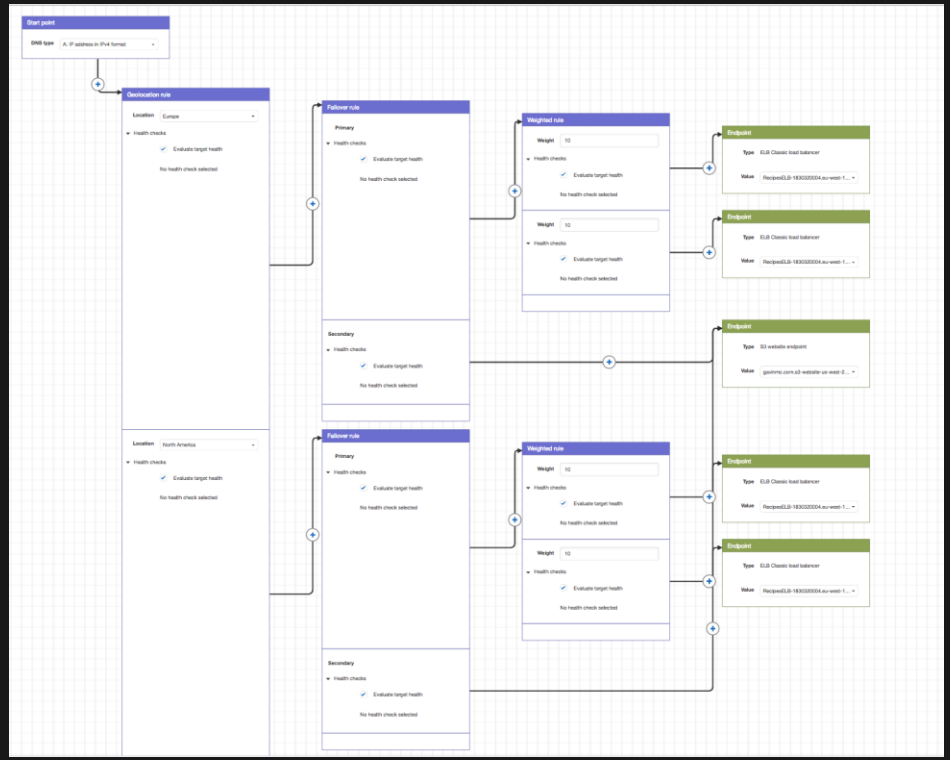
- *Customer: My routing policy combines Health Checks, Weighting, LBR, and Failover records. I need a way to visualise the flow.*
- *Customer: I need to be able to make multiple, atomic changes to my routing policy.*
- *Customer: I need to be able to rollback my routing policy.*

- *Feature: Route 53 Traffic Flow UI and API.*

Route53 Traffic Flow



Route53 Traffic Flow



Traffic Flow Policy as Code

```
{
    "AWSPolicyFormatVersion":"2015-10-01",
    "RecordType":"A",
    "Endpoints":{
        "endpoint-weighted-blue":{
            "Type":"value",
            "Value":"1.2.3.4"
        },
        "endpoint-weighted-green":{
            "Type":"value",
            "Value":"3.4.5.6"
        }
    },
    "Rules":{
        "weighted-start-purple":{
            "RuleType":"weighted",
            "Items":[
                {"Weight":"10", "EndpointReference":"endpoint-weighted-blue", "EvaluateTargetHealth":true},
                {"Weight":"10", "EndpointReference":"endpoint-weighted-green","EvaluateTargetHealth":true}
            ]
        }
    },
    "StartRule":"weighted-start-purple"
}
```

Creating a Policy

```
a45e60d6bfe1:~ gavinmc$ aws route53 create-traffic-policy --name reinvent-2017.com-policy \
                                --document file://policy.json
{
    "TrafficPolicy": {
        "Document": "{……}",
        "Version": 2,
        "Type": "A",
        "Id": "fd936505-74a3-4f58-8ebb-be270b426460",
        "Name": "reinvent-2017.com-policy"
    },
    "Location": "https://route53.amazonaws.com/2013-04-01/trafficpolicy/fd936505-74a3-4f58-8ebb-
be270b426460/1"
}
```

Apply the Policy

```
a45e60d6bfe1:~ gavinmc$ aws route53 update-traffic-policy-instance \
    --traffic-policy-version 2  --ttl 60 \
    --id=3a91fc71-76b6-4e8e-b7ad-a7b3eaa8b6ca \
    --traffic-policy-id fd936505-74a3-4f58-8ebb-be270b426460
{
    "TrafficPolicyInstance": {
        "Name": "www.route53-2017.com.",
        "TrafficPolicyVersion": 2,
        "TrafficPolicyType": "A",
        "State": "Updating",
        "TrafficPolicyId": "fd936505-74a3-4f58-8ebb-be270b426460",
        "TTL": 60,
        "HostedZoneId": "Z2DICVFGUDZI8F",
        "Id": "3a91fc71-76b6-4e8e-b7ad-a7b3eaa8b6ca"
    }
}
```

Rolling Back a Policy

```
a45e60d6bfe1:~ gavinmc$ aws route53 update-traffic-policy-instance \
     --traffic-policy-version 1  --ttl 60 \
     --id=3a91fc71-76b6-4e8e-b7ad-a7b3eaa8b6ca \
     --traffic-policy-id fd936505-74a3-4f58-8ebb-be270b426460
{
    "TrafficPolicyInstance": {
        "Name": "www.route53-2017.com.",
        "TrafficPolicyVersion": 1,
        "TrafficPolicyType": "A",
        "State": "Updating",
        "TrafficPolicyId": "fd936505-74a3-4f58-8ebb-be270b426460",
        "TTL": 60,
        "HostedZoneId": "Z2DICVFGUDZI8F",
        "Id": "3a91fc71-76b6-4e8e-b7ad-a7b3eaa8b6ca"
    }
}
```

DNS Operational Excellence Tips

Safely Switching Providers

AWS re:Invent

aws

TTLs Are Important!

- As a DNS operator, the TTL is your friend!
- Short TTL = fast traffic shifts and rollbacks
- Long TTL = better customer experience, lower query volume ($)
- For critical changes, lower the TTL temporarily
    - Faster rollback

Switching DNS Providers

- We want to move a production DNS zone to Route 53
- Simplest solution:
    1. Export the zone data
    2. Create a zone in Route 53 and import data
    3. Update registrar with new nameservers
- If you have a problem, roll back takes two days ☹

Safely Switching DNS Providers

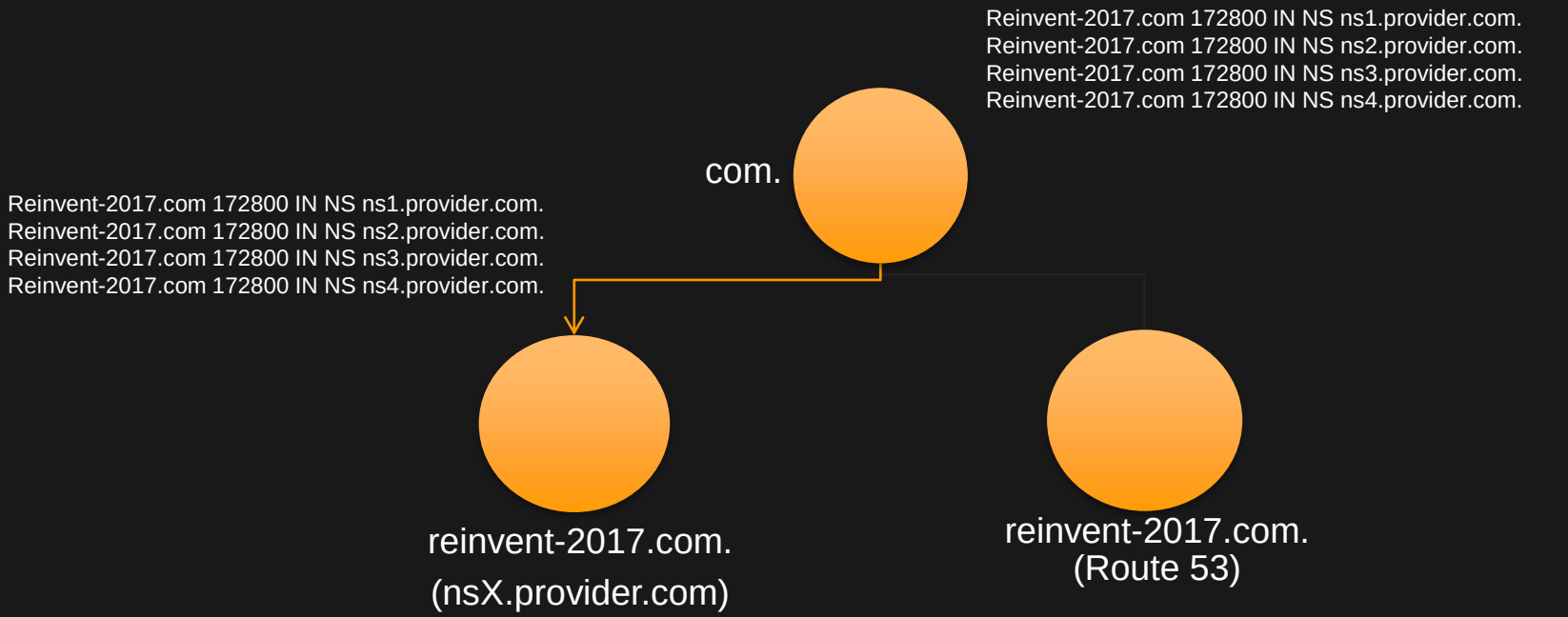
- Good change management always includes quick rollback steps
- Rollback of registrar: two days!
    - TTL on TLD NS records is 172800 sec
- There is a way to make it safer
    - Two copies of NS records
    - Most resolvers cache the in-zone copy

Recall: Recursive Resolution

```
reinvent-2017.com.  172800    IN   NS   ns-1526.awsdns-62.org.
reinvent-2017.com.  172800    IN   NS   ns-1900.awsdns-45.co.uk.
reinvent-2017.com.  172800    IN   NS   ns-709.awsdns-24.net.
reinvent-2017.com.  172800    IN   NS   ns-250.awsdns-31.com.
;; Received 704 bytes from 192.41.162.30#53(l.gtld-servers.net) in 38 ms

www.reinvent-2017.com.   60   IN   A    52.216.131.34
reinvent-2017.com.  172800    IN   NS   ns-1526.awsdns-62.org.
reinvent-2017.com.  172800    IN   NS   ns-1900.awsdns-45.co.uk.
reinvent-2017.com.  172800    IN   NS   ns-250.awsdns-31.com.
reinvent-2017.com.  172800    IN   NS   ns-709.awsdns-24.net.
;; Received 203 bytes from 205.251.192.250#53(ns-250.awsdns-31.com) in 9 ms
```

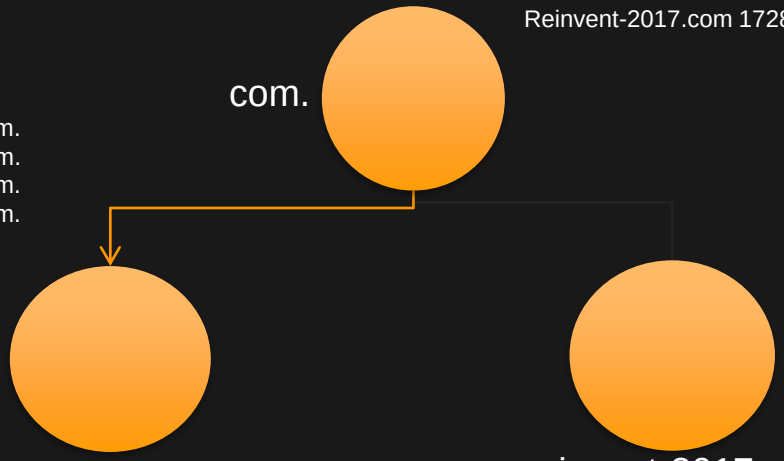
Starting Point



Step 1—Lower TTLs (t=0)

Reinvent-2017.com 172800 IN NS ns1.provider.com.
Reinvent-2017.com 172800 IN NS ns2.provider.com.
Reinvent-2017.com 172800 IN NS ns3.provider.com.
Reinvent-2017.com 172800 IN NS ns4.provider.com.

Reinvent-2017.com 600 IN NS ns1.provider.com.
Reinvent-2017.com 600 IN NS ns2.provider.com.
Reinvent-2017.com 600 IN NS ns3.provider.com.
Reinvent-2017.com 600 IN NS ns4.provider.com.

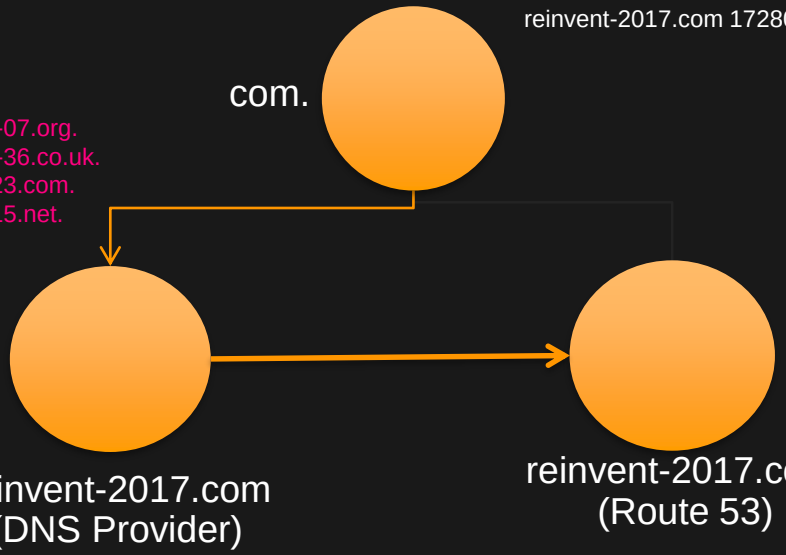


Reinvent-2017.com 600 IN NS ns-1084.awsdns-07.org.
Reinvent-2017.com 600 IN NS ns-1831.awsdns-36.co.uk.
Reinvent-2017.com 600 IN NS ns-190.awsdns-23.com.
Reinvent-2017.com 600 IN NS ns-634.awsdns-15.net.

Step 2—Alter NS In-Zone (t = 2 Days)

reinvent-2017.com 172800 IN NS ns1.provider.com.
reinvent-2017.com 172800 IN NS ns2.provider.com.
reinvent-2017.com 172800 IN NS ns3.provider.com.
reinvent-2017.com 172800 IN NS ns4.provider.com.

reinvent-2017.com 600 IN NS ns-1084.awsdns-07.org.
reinvent-2017.com 600 IN NS ns-1831.awsdns-36.co.uk.
reinvent-2017.com 600 IN NS ns-190.awsdns-23.com.
reinvent-2017.com 600 IN NS ns-634.awsdns-15.net.

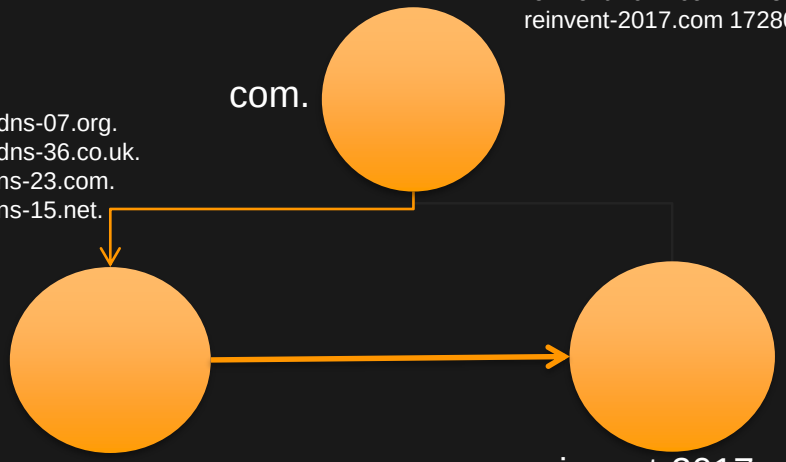


reinvent-2017.com 600 IN NS ns-1084.awsdns-07.org.
reinvent-2017.com 600 IN NS ns-1831.awsdns-36.co.uk.
reinvent-2017.com 600 IN NS ns-190.awsdns-23.com.
reinvent-2017.com 600 IN NS ns-634.awsdns-15.net.

Step 3—Raise TTLs (t = 3 Days)

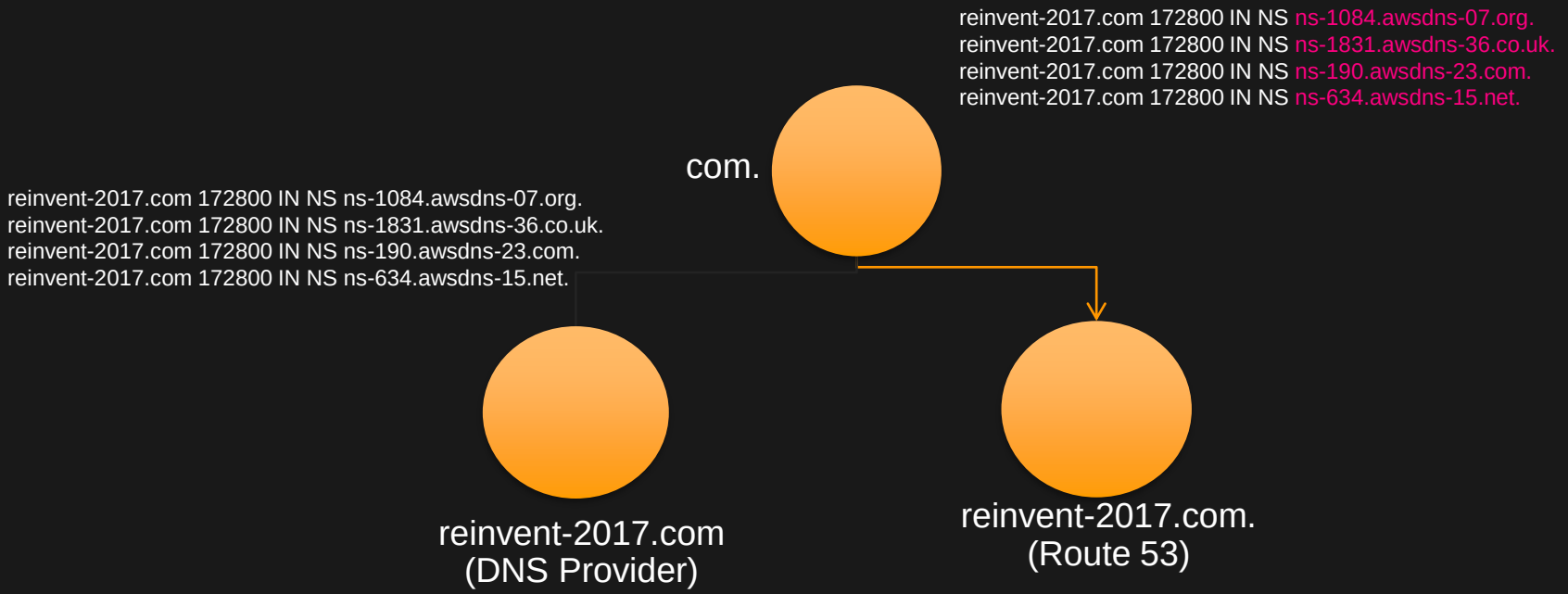
reinvent-2017.com 172800 IN NS ns1.provider.com.
reinvent-2017.com 172800 IN NS ns2.provider.com.
reinvent-2017.com 172800 IN NS ns3.provider.com.
reinvent-2017.com 172800 IN NS ns4.provider.com.

reinvent-2017.com 172800 IN NS ns-1084.awsdns-07.org.
reinvent-2017.com 172800 IN NS ns-1831.awsdns-36.co.uk.
reinvent-2017.com 172800 IN NS ns-190.awsdns-23.com.
reinvent-2017.com 172800 IN NS ns-634.awsdns-15.net.



reinvent-2017.com 172800 IN NS ns-1084.awsdns-07.org.
reinvent-2017.com 172800 IN NS ns-1831.awsdns-36.co.uk.
reinvent-2017.com 172800 IN NS ns-190.awsdns-23.com.
reinvent-2017.com 172800 IN NS ns-634.awsdns-15.net.

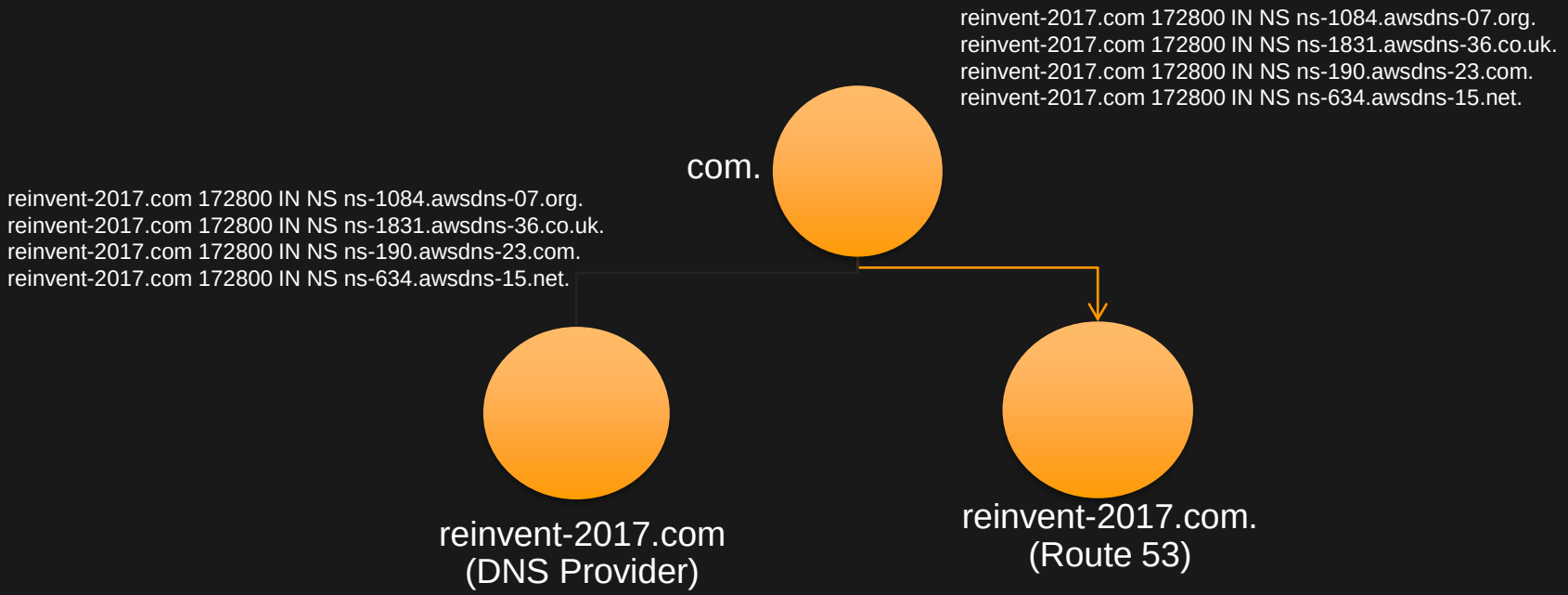
Step 4—Change the Registrar (t = 4 Days)



AWS re:Invent

aws

Step 5—Wait for Traffic to Drain



Summarizing

- Route 53: Low Cost, High Availability DNS services
- Flexible Traffic Management
- Fully Automatable
- MuleSoft—Time-Sensitive DNS Changes
- Operational Excellence

AWS re:Invent

Thank you!