

IAM Policy to Restrict Updating Route53 Recordsets

- 1. Create a public domain `qinjie.com` in Route53.
 - Add 2 TXT records `test.qinjie.com` and `banned.qinjie.com`

Public **qinjie.com** Info

Delete zoneTest recordConfigure query logging

► Hosted zone details

Edit hosted zone

Records (4)DNSSEC signingHosted zone tags (0)

Records (4) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

↺

Delete record

Import zone file

Create record

🔍 Filter records by property or value

Type ▼

Routing policy ▼

Alias ▼

< 1 >

⚙

<input type="checkbox"/>	Record name ▼	Type ▼	Routin... ▼	Differ... ▼	Value/Route traffic to ▼
<input type="checkbox"/>	qinjie.com	NS	Simple	-	ns-1147.awsdns-15.org. ns-102.awsdns-12.com. ns-1615.awsdns-09.co.uk. ns-921.awsdns-51.net.
<input type="checkbox"/>	qinjie.com	SOA	Simple	-	ns-1147.awsdns-15.org. awsdns-hostmaster.amazo...
<input type="checkbox"/>	banned.qinjie.com	TXT	Simple	-	"banned"
<input type="checkbox"/>	test.qinjie.com	TXT	Simple	-	"aaaaa"

- 2. Create another public domain `zhang.com` in Route53.
 - Add 2 TXT records, `test.zhang.com` and `banned.zhang.com`.

Public
zhang.com
Info
Delete zone
Test record
Configure query logging

▶ Hosted zone details
Edit hosted zone

Records (4)
DNSSEC signing
Hosted zone tags (0)

Records (4) Info

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Refresh
Delete record
Import zone file
Create record

Filter records by property or value
Type
Routing policy
Alias
< 1 >

	Record name	Type	Routin...	Differ...	Value/Route traffic to
<input type="checkbox"/>	zhang.com	NS	Simple	-	ns-1806.awsdns-33.co.uk. ns-1505.awsdns-60.org. ns-54.awsdns-06.com. ns-800.awsdns-36.net.
<input type="checkbox"/>	zhang.com	SOA	Simple	-	ns-1806.awsdns-33.co.uk. awsdns-hostmaster.ama...
<input type="checkbox"/>	banned.zhang.com	TXT	Simple	-	"banned"
<input type="checkbox"/>	test.zhang.com	TXT	Simple	-	"aaa"

3. Now we have 2 public domains for testing.

Hosted zones (2)

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Refresh
View details
Edit
Delete
Create hosted zone

Filter records by property or value
< 1 >

	Hosted zone name	Type	Created...	Record ...	Descrip...	Hosted zone ID
<input type="radio"/>	qinjie.com	Public	Route 53	4	-	Z05739232WBO2...
<input type="radio"/>	zhang.com	Public	Route 53	4	-	Z04824712LR428...

4. Create an IAM Role `lambda_update_route53_records` for Lambda with following policy, which allows role to update recordset `test.qinjie.com` in domain `qinjie.com`.

- Permissions (Inline Policy), where `Z05739232WBO20BEZE484` is the Zone ID of `qinjie.com`.

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "route53:ChangeResourceRecordSets",
7              "Resource":
8                  "arn:aws:route53::hostedzone/Z05739232WBO20BEZE484",
9              "Condition": {
1             "ForAllValues:StringEquals": {

```

```

10     "route53:ChangeResourceRecordSetsNormalizedRecordNames": [
11         "test.qinjie.com"
12     ]
13     }
14 }
15 }
16 ]
17 }

```

- Trust Relationships

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Principal": {
7                  "Service": "lambda.amazonaws.com"
8              },
9              "Action": "sts:AssumeRole"
10         }
11     ]
12 }

```

5. Create a lambda function with following code.

- Set role to `lambda_update_route53_records`.

```

1  import json
2  import boto3
3
4  def lambda_handler(event, context):
5
6      client = boto3.client('route53')
7
8      # TEST UPDATING OTHER RECORDSETS IN CORRECT DOMAIN
9
10     zone_id = 'Z05739232WBO20BEZE484' # qinjie.com
11     value = "aaaaa"
12     record_names = ['test.qinjie.com', 'banned.qinjie.com']
13
14     for record_name in record_names:
15         print(f"Updating {record_name}")
16         try:
17             response = client.change_resource_record_sets(
18                 HostedZoneId=zone_id,
19                 ChangeBatch={
20                     "Comment": f"Update TXT Fields {record_name}",

```

```

21         "Changes": [
22             {
23                 "Action": "UPSERT",
24                 "ResourceRecordSet": {
25                     "Name": record_name,
26                     "Type": "TXT",
27                     "TTL": 180,
28                     "ResourceRecords": [
29                         {
30                             "Value": f'"{value}"'
31                         },
32                     ],
33                 }
34             },
35         ]
36     }
37 )
38     print(f"Success: {response}")
39 except Exception as ex:
40     print(f"Error: {ex}")
41     print('+++++')
42
43 # TEST UPDATING RECORDSETS IN WRONG DOMAIN
44
45 zone_id = 'Z04824712LR428SMYXA4A' # zhang.com
46 value = 'aaaaa'
47 record_names = ['test.zhang.com', 'banned.zhang.com']
48
49 for record_name in record_names:
50     print(f"Updating {record_name}")
51     try:
52         response = client.change_resource_record_sets(
53             HostedZoneId=zone_id,
54             ChangeBatch={
55                 "Comment": f"Update TXT Fields {record_name}",
56                 "Changes": [
57                     {
58                         "Action": "UPSERT",
59                         "ResourceRecordSet": {
60                             "Name": record_name,
61                             "Type": "TXT",
62                             "TTL": 180,
63                             "ResourceRecords": [
64                                 {
65                                     "Value": f'"{value}"'
66                                 },
67                             ],
68                         }
69                     },

```

```

70         ]
71     }
72 )
73     print(f"Success: {response}")
74 except Exception as ex:
75     print(f"Error: {ex}")
76     print('+++++')
77
78 return {
79     'Message': "DONE"
80 }
81

```

6. Run the Lambda to test. Can see that only updating of first record set is successful.

Test Event Name test	
Response { "Message": "DONE" }	
Function Logs START RequestId: f11ac10c-311a-41cd-9e57-2cc735eb0706 Version: \$LATEST Updating test.qinjie.com Success: {'ResponseMetadata': {'RequestId': '3c0667a7-b5a2-463d-bb15-1e000f4c7d07', 'HTTPStatusCode': 200, 'Headers': {'x-amzn-requestid': '3c0667a7-b5a2-463d-bb15-1e000f4c7d07'}, 'RetryAttempts': 0}} +++++ Updating banned.qinjie.com Error: An error occurred (AccessDenied) when calling the ChangeResourceRecordSets operation: User: +++++ Updating test.zhang.com Error: An error occurred (AccessDenied) when calling the ChangeResourceRecordSets operation: User: +++++ Updating banned.zhang.com Error: An error occurred (AccessDenied) when calling the ChangeResourceRecordSets operation: User: +++++ END RequestId: f11ac10c-311a-41cd-9e57-2cc735eb0706 REPORT RequestId: f11ac10c-311a-41cd-9e57-2cc735eb0706 Duration: 2886.38 ms Billed Duration: 28	
Request ID f11ac10c-311a-41cd-9e57-2cc735eb0706	