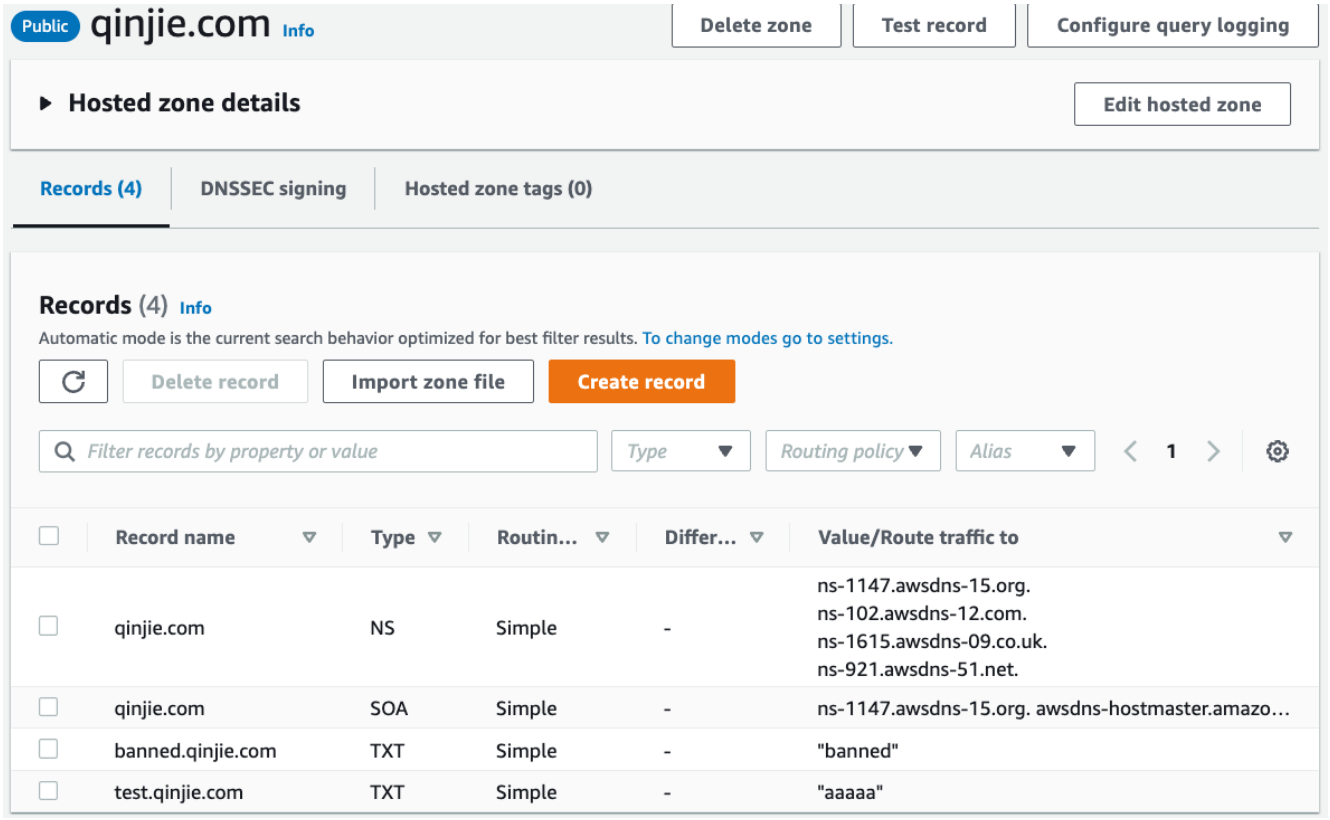
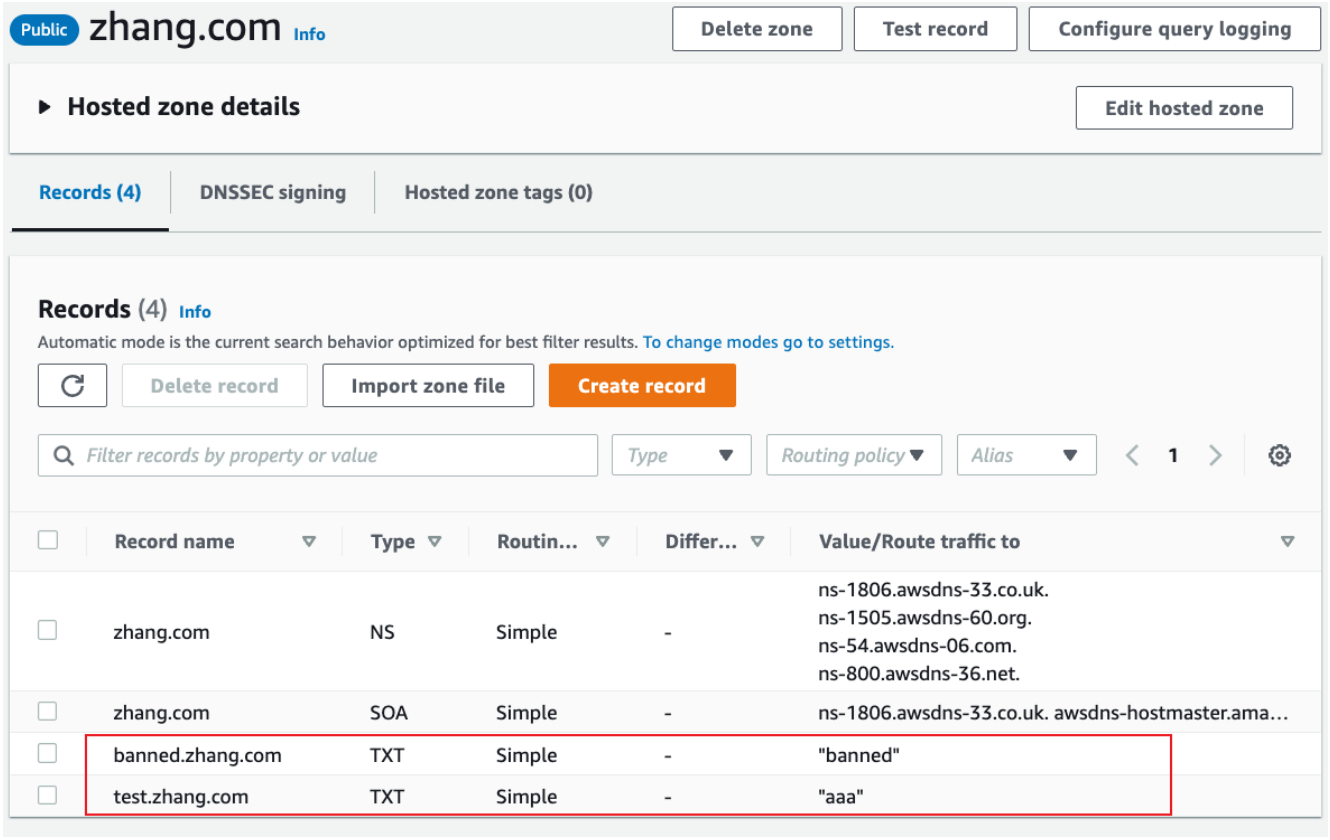
IAM Policy to Restrict Updating Route53 Recordsets

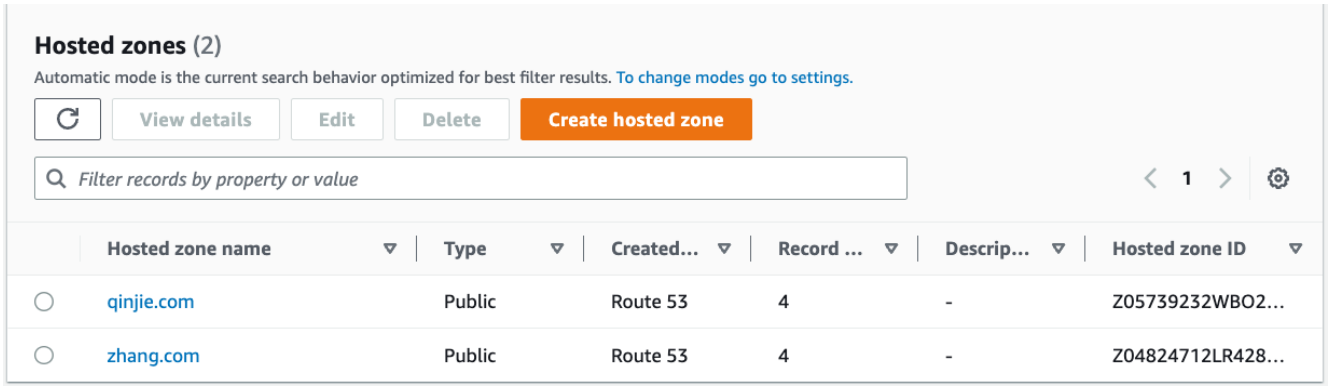
1. Create a public domain `qinjie.com` in Route53.
    - Add 2 TXT records `test.qinjie.com` and `banned.qinjie.com`



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



3. Now we have 2 public domains for testing.



4. Create an IAM Role `lambda_update_route53_records` for Lmabda with following policy, which allows role to update recordset `test.qinjie.com` in domain `qinjie.com`.
   - Permissions (Inline Policy), where `Z0573923n2WBO20BEZE484` is the Zone ID of `qinjie.com`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "route53:ChangeResourceRecordSets",
            "Resource":
"arn:aws:route53:::hostedzone/Z05739232WBO20BEZE484",
            "Condition": {
                "ForAllValues:StringEquals": {
```

```
                "route53:ChangeResourceRecordSetsNormalizedRecordNames": [
                                "test.qinjie.com"
                            ]
                        }
                    }
                ]
}
```

- Trust Relationships

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

5. Create a lambda function with following code.
    - Set role to `lambda_update_route53_records`.

```
import json
import boto3

def lambda_handler(event, context):

    client = boto3.client('route53')

    # TEST UPDATING OTHER RECORDSETS IN CORRECT DOMAIN

    zone_id = 'Z05739232WBO20BEZE484' # qinjie.com
    value = "aaaaa"
    record_names = ['test.qinjie.com', 'banned.qinjie.com']

    for record_name in record_names:
        print(f"Updating {record_name}")
        try:
            response = client.change_resource_record_sets(
                HostedZoneId=zone_id,
                ChangeBatch={
                    "Comment": f"Update TXT Fields {record_name}",
```

```
                    "Changes": [
                        {
                            "Action": "UPSERT",
                            "ResourceRecordSet": {
                                "Name": record_name,
                                "Type": "TXT",
                                "TTL": 180,
                                "ResourceRecords": [
                                    {
                                        "Value": f'"{value}"'
                                    },
                                ],
                            }
                        },
                    ]
                }
            )
            print(f"Success: {response}")
        except Exception as ex:
            print(f"Error: {ex}")
        print('++++++++++++++++++++++++++++++++++++++')

    # TEST UPDATING RECORDSETS IN WRONG DOMAIN

    zone_id = 'Z04824712LR428SMYXA4A' # zhang.com
    value = 'aaaaa'
    record_names = ['test.zhang.com', 'banned.zhang.com']

    for record_name in record_names:
        print(f"Updating {record_name}")
        try:
            response = client.change_resource_record_sets(
                HostedZoneId=zone_id,
                ChangeBatch={
                    "Comment": f"Update TXT Fields {record_name}",
                    "Changes": [
                        {
                            "Action": "UPSERT",
                            "ResourceRecordSet": {
                                "Name": record_name,
                                "Type": "TXT",
                                "TTL": 180,
                                "ResourceRecords": [
                                    {
                                        "Value": f'"{value}"'
                                    },
                                ],
                            }
                        },
```

```
                    ]
                }
            )
            print(f"Success: {response}")
        except Exception as ex:
            print(f"Error: {ex}")
        print('++++++++++++++++++++++++++++++++++++++')

    return {
        'Message': "DONE"
    }

```

6. Run the Lambd 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
++++++++++++++++++++++++++++++++++++++
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