
certbot-dns-route53 Documentation

Release 0

Certbot Project

Jul 11, 2019

Contents:

1	Named Arguments	3
2	Credentials	5
3	Examples	7
4	API Documentation	9
4.1	certbot_dns_route53.authenticator	9
4.2	certbot_dns_route53.dns_route53	9
5	Indices and tables	11
	Python Module Index	13
	Index	15

The *dns_route53* plugin automates the process of completing a `dns-01` challenge (DNS01) by creating, and subsequently removing, TXT records using the Amazon Web Services Route 53 API.

CHAPTER 1

Named Arguments

<code>--dns-route53-propagation-timeout</code>	The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (Default: 10)
--	--

Use of this plugin requires a configuration file containing Amazon Web Services API credentials for an account with the following permissions:

- `route53:ListHostedZones`
- `route53:GetChange`
- `route53:ChangeResourceRecordSets`

These permissions can be captured in an AWS policy like the one below. Amazon provides [information about managing access and information about the required permissions](#)

Listing 1: Example AWS policy file:

```
{
  "Version": "2012-10-17",
  "Id": "certbot-dns-route53 sample policy",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53:ListHostedZones",
        "route53:GetChange"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "route53:ChangeResourceRecordSets"
      ],
      "Resource" : [
        "arn:aws:route53:::hostedzone/YOURHOSTEDZONEID"
      ]
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
}  
  ]  
}
```

The [access keys](#) for an account with these permissions must be supplied in one of the following ways, which are discussed in more detail in the Boto3 library's documentation about [configuring credentials](#).

- Using the `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY` environment variables.
- Using a credentials configuration file at the default location, `~/.aws/config`.
- Using a credentials configuration file at a path supplied using the `AWS_CONFIG_FILE` environment variable.

Listing 2: Example credentials config file:

```
[default]  
aws_access_key_id=AKIAIOSFODNN7EXAMPLE  
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

Caution: You should protect these API credentials as you would a password. Users who can read this file can use these credentials to issue some types of API calls on your behalf, limited by the permissions assigned to the account. Users who can cause Certbot to run using these credentials can complete a `dns-01` challenge to acquire new certificates or revoke existing certificates for domains these credentials are authorized to manage.

Examples

Listing 1: To acquire a certificate for `example.com`

```
certbot certonly \  
  --dns-route53 \  
  -d example.com
```

Listing 2: To acquire a single certificate for both `example.com` and `www.example.com`

```
certbot certonly \  
  --dns-route53 \  
  -d example.com \  
  -d www.example.com
```

Listing 3: To acquire a certificate for `example.com`, waiting 30 seconds for DNS propagation

```
certbot certonly \  
  --dns-route53 \  
  --dns-route53-propagation-seconds 30 \  
  -d example.com
```


4.1 certbot_dns_route53.authenticator

Shim around `dns_route53` for backwards compatibility.

class `certbot_dns_route53.authenticator.Authenticator` (**args*, ***kwargs*)

Bases: `certbot_dns_route53.dns_route53.Authenticator`

Shim around `Authenticator` for backwards compatibility.

4.2 certbot_dns_route53.dns_route53

Certbot Route53 authenticator plugin.

class `certbot_dns_route53.dns_route53.Authenticator` (**args*, ***kwargs*)

Bases: `certbot.plugins.dns_common.DNSAuthenticator`

Route53 Authenticator

This authenticator solves a DNS01 challenge by uploading the answer to AWS Route53.

`_setup_credentials()`

Establish credentials, prompting if necessary.

`_perform(domain, validation_name, validation)`

Performs a dns-01 challenge by creating a DNS TXT record.

Parameters

- **`domain`** (*str*) – The domain being validated.
- **`validation_domain_name`** (*str*) – The validation record domain name.
- **`validation`** (*str*) – The validation record content.

Raises `errors.PluginError` – If the challenge cannot be performed

`__cleanup` (*domain, validation_name, validation*)

Deletes the DNS TXT record which would have been created by `__perform_achall`.

Fails gracefully if no such record exists.

Parameters

- **`domain`** (*str*) – The domain being validated.
- **`validation_domain_name`** (*str*) – The validation record domain name.
- **`validation`** (*str*) – The validation record content.

`__find_zone_id_for_domain` (*domain*)

Find the zone id responsible a given FQDN.

That is, the id for the zone whose name is the longest parent of the domain.

`__wait_for_change` (*change_id*)

Wait for a change to be propagated to all Route53 DNS servers. https://docs.aws.amazon.com/Route53/latest/APIReference/API_GetChange.html

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

C

certbot_dns_route53, 1
certbot_dns_route53.authenticator, 9
certbot_dns_route53.dns_route53, 9

Symbols

`_cleanup()` (*certbot_dns_route53.dns_route53.Authenticator*
method), 9

`_find_zone_id_for_domain()` (*cert-*
bot_dns_route53.dns_route53.Authenticator
method), 10

`_perform()` (*certbot_dns_route53.dns_route53.Authenticator*
method), 9

`_setup_credentials()` (*cert-*
bot_dns_route53.dns_route53.Authenticator
method), 9

`_wait_for_change()` (*cert-*
bot_dns_route53.dns_route53.Authenticator
method), 10

A

`Authenticator` (class in *cert-*
bot_dns_route53.authenticator), 9

`Authenticator` (class in *cert-*
bot_dns_route53.dns_route53), 9

C

`certbot_dns_route53` (module), 1

`certbot_dns_route53.authenticator` (mod-
ule), 9

`certbot_dns_route53.dns_route53` (module),
9