CONTENTS

// Tutorial //

How To Use Certbot Standalone Mode to Retrieve Let's Encrypt SSL Certificates on Ubuntu 16.04

Published on July 28, 2017

Security Ubuntu Let's Encrypt Ubuntu 16.04

By Brian Boucheron

Not using Ubuntu 16.04?
Choose a different version or distribution.

Ubuntu 16.04

Introduction

Let's Encrypt is a service offering free SSL certificates through an automated API. The most popular Let's Encrypt client is EFF's Certbot.

Certbot offers a variety of ways to validate your domain, fetch certificates, and automatically configure Apache and Nginx. In this tutorial, we'll discuss Certbot's *standalone* mode and how to use it to secure other types of services, such as a mail server or a message broker like RabbitMQ.

We won't discuss the details of SSL configuration, but when you are done you will have a valid certificate that is automatically renewed. Additionally, you will be able to automate reloading your service to pick up the renewed certificate.

Prerequisites

Before starting this tutorial, you will need:

- An Ubuntu 16.04 server with a non-root, sudo-enabled user and basic firewall set up, as detailed in this Ubuntu 16.04 server setup tutorial.
- A domain name pointed at your server, which you can accomplish by following "How to Set Up a Host Name with DigitalOcean." This tutorial will use `example.com` throughout.

- Port 80 **or** 443 must be unused on your server. If the service you're trying to secure is on a machine with a web server that occupies both of those ports, you'll need to use a different mode such as Certbot's *webroot* mode.

Step 1 – Installing Certbot

Ubuntu includes the Certbot client in their default repository, but it's a bit out of date. Instead, we'll install it from Certbot's official Ubuntu PPA, or *Personal Package Archive*. These are alternative repositories that package more recent or more obscure software. First, add the repository:

```
$ sudo add-apt-repository ppa:certbot/certbot
```

Copy

You'll need to press `ENTER` to accept. Afterwards, update the package list to pick up the new repository's package information:

```
$ sudo apt-get update
```

Copy

And finally, install the `certbot` package:

```
$ sudo apt-get install certbot
```

Copy

Now that we have Certbot installed, let's run it to get our certificate.

Step 2 – Running Certbot

Certbot needs to answer a cryptographic challenge issued by the Let's Encrypt API in order to prove we control our domain. It uses ports `80` (HTTP) or `443` (HTTPS) to accomplish this. Open up the appropriate port in your firewall:

```
$ sudo ufw allow 80
```

Copy

Substitute `443` above if that's the port you're using. `ufw` will output confirmation that your rule was added:

```
Output
Rule added
Rule added (v6)
```

We can now run Certbot to get our certificate. We'll use the `--standalone` option to tell Certbot to handle the challenge using its own built-in web server. The `--preferred-challenges` option instructs Certbot to use port 80 or port 443. If you're using port 80, you want `--preferred-challenges http`. For port 443 it would be `--preferred-challenges tls-sni`. Finally, the `-d` flag is used to specify the domain you're requesting a certificate for. You can add multiple `-d` options to cover multiple domains in one certificate.

```
$ sudo certbot certonly --standalone --preferred-challenges http -d example. Copy
```

When running the command, you will be prompted to enter an email address and agree to the terms of service. After doing so, you should see a message telling you the process was successful and where your certificates are stored:

```
Output
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at
   /etc/letsencrypt/live/example.com/fullchain.pem. Your cert will
   expire on 2017-10-23. To obtain a new or tweaked version of this
   certificate in the future, simply run certbot again with the
   "certonly" option. To non-interactively renew *all* of your
   certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

We've got our certificates. Let's take a look at what we downloaded and how to use the files with our software.

Step 3 – Configuring Your Application

Configuring your application for SSL is beyond the scope of this article, as each application has different requirements and configuration options, but let's take a look at what Certbot has downloaded for us. Use `ls` to list out the directory that holds our keys and certificates:

```
$ sudo ls /etc/letsencrypt/live/example.com
```

Copy

```
Output
cert.pem  chain.pem  fullchain.pem  privkey.pem  README
```

The `README` file in this directory has more information about each of these files. Most often you'll only need two of these files:

- `privkey.pem`: This is the private key for the certificate. This needs to be kept safe and secret, which is why most of the `/etc/letsencrypt` directory has very restrictive permissions and is accessible by only the **root** user. Most software configuration will refer to this as something similar to `ssl-certificate-key` or `ssl-certificate-key-file`.
- `fullchain.pem`: This is our certificate, bundled with all intermediate certificates. Most software will use this file for the actual certificate, and will refer to it in their configuration with a name like 'ssl-certificate'.

For more information on the other files present, refer to the "Where are my certificates" section of the Certbot docs.

Some software will need its certificates in other formats, in other locations, or with other user permissions. It is best to leave everything in the `letsencrypt` directory, and not change any permissions in there (permissions will just be overwritten upon renewal anyway), but sometimes that's just not an option. In that case, you'll need to write a script to move files and change permissions as needed. This script will need to be run whenever Certbot renews the certificates, which we'll talk about next.

Step 4 — Handling Certbot Automatic Renewals

Let's Encrypt's certificates are only valid for ninety days. This is to encourage users to automate their certificate renewal process. The `certbot` package we installed takes care of this for us by adding a renew script to `/etc/cron.d`. This script runs twice a day and will renew any certificate that's within thirty days of expiration.

With our certificates renewing automatically, we still need a way to run other tasks after a renewal. We need to at least restart or reload our server to pick up the new certificates, and as mentioned in Step 3 we may need to manipulate the certificate files in some way to make them work with the software we're using. This is the purpose of Certbot's `renew_hook` option.

To add a `renew_hook`, we update Certbot's renewal config file. Certbot remembers all the details of how you first fetched the certificate, and will run with the same options upon renewal. We just need to add in our hook. Open the config file with you favorite editor:

```
$ sudo nano /etc/letsencrypt/renewal/example.com.conf
```

Copy

A text file will open with some configuration options. Add your hook on the last line:

/etc/letsencrypt/renewal/example.com.conf

```
renew_hook = systemctl reload rabbitmq
```

Update the command above to whatever you need to run to reload your server or run your custom file munging script. Usually, on Ubuntu, you'll mostly be using `systemctl` to reload a service. Save and close the file, then run a Certbot dry run to make sure the syntax is ok:

```
$ sudo certbot renew --dry-run
```

Copy

If you see no errors, you're all set. Certbot is set to renew when necessary and run any commands needed to get your service using the new files.

Conclusion

In this tutorial, we've installed the Certbot Let's Encrypt client, downloaded an SSL certificate using standalone mode, and enabled automatic renewals with renew hooks. This should give you a good start on using Let's Encrypt certificates with services other than your typical web server.

For more information, please refer to Certbot's documentation.

About the authors

Brian Boucheron Author

Still looking for an answer? Ask a question

Search for more help

Was this helpful? Yes No

Comments

8 Comments

Leave a comment...

This textbox defaults to using `Markdown` to format your answer.

You can type `!ref` in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

Sign In or Sign Up to Comment

jamesburns1977 • April 23, 2018

Around a year ago, I used great DO articles such as this one to teach myself how to get LetsEncypt up and running on my first site. First time round, I used letsencrypt command manually (`sudo letsencrypt --apache --expand -d mysite.com- d www.mysite.com`) and have been updating the cert manually every 3 months (`sudo letsencrypt renew --agree-tos`).

I am now on my second site and decided to go through the latest tutorial updates and now have this working using certbot! And it appears that certbot will auto handle all renewals - nice:)

So the questions is, how do I get my original site working with certbot when I have been using letsencrypt commands directly. I am within 30 days of my original sites cert running out. Is it as simple as just running certbot like it’s the first time to ensure it takes over? Or is there a migration process?

Help gratefully appreciated…

Reply

Nearly Normal • January 16, 2018

Is this the easiest way to set this up? There seem to be way too many ways online. Some seem easier than others. This is more complex.

Reply

Lewis Cowles • December 17, 2017

Is there a way to have certbot not check for updates every single time it’s run. It’s only run every 3 months, but it always seems to be updating. damn thing.

Reply

altfatterz • October 29, 2017

I got this error, any ideas?

```
zoal@zoltans-mbp:~|⇒  sudo certbot certonly --standalone --preferred-challen
Password:
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for example.com
Waiting for verification...
Cleaning up challenges
Failed authorization procedure. example.com (http-01): urn:acme:error:unauthc
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type"

IMPORTANT NOTES:
 - The following errors were reported by the server:

   Domain: example.com
   Type:   unauthorized
   Detail: Invalid response from
   http://example.com/.well-known/acme-challenge/0hAZjaUob6DT83A689S8N_sq-b94
   "<!doctype html>
   <html>
   <head>
       <title>Example Domain</title>

       <meta charset="utf-8" />
       <meta http-equiv="Content-type"

   To fix these errors, please make sure that your domain name was
   entered correctly and the DNS A/AAAA record(s) for that domain
   contain(s) the right IP address.
```

Show replies   Reply

shared • October 17, 2017

I get this error, any ideas?

```
root@example:~# sudo certbot certonly --standalone --preferred-challenges htt
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for example.com
Cleaning up challenges
Problem binding to port 80: Could not bind to IPv4 or IPv6.
```

Show replies Reply

BN Bn • September 13, 2017

Hi all,

We follow the tutorial but had an error can you help on that please we still have some issue to activate SSL with certbot.

Error Message: Type: tls Detail: remote error: tls: handshake failure

To fix these errors, please make sure that your domain name was entered correctly and the DNS A/AAAA record(s) for that domain contain(s) the right IP address. Additionally, please check that you have an up-to-date TLS configuration that allows the server to communicate with the Certbot client.

Reply

Purwo Widodo • August 17, 2017

test

Reply

Alexey Deryagin • August 8, 2017

To avoid “required ports is already taken” error during renewal use hooks for web-server instance e.g. for nginx:

```
/usr/bin/certbot renew --pre-hook "systemctl stop nginx" --post-hook "systemc
```

Show replies Reply

Popular Topics

Ubuntu

Linux Basics

JavaScript

Python

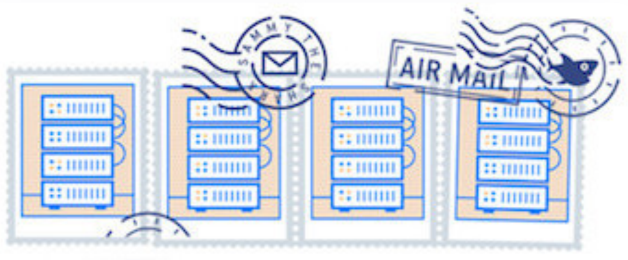
MySQL

Docker

Kubernetes

All tutorials →

Free Managed Hosting →



Get our biweekly newsletter

Sign up for Infrastructure as a Newsletter.

Sign up →

Hollie's Hub for Good

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.

Learn more →

Become a contributor

You get paid; we donate to tech nonprofits.

Learn more →

Featured on Community

Kubernetes Course   Learn Python 3   Machine Learning in Python

Getting started with Go   Intro to Kubernetes

DigitalOcean Products

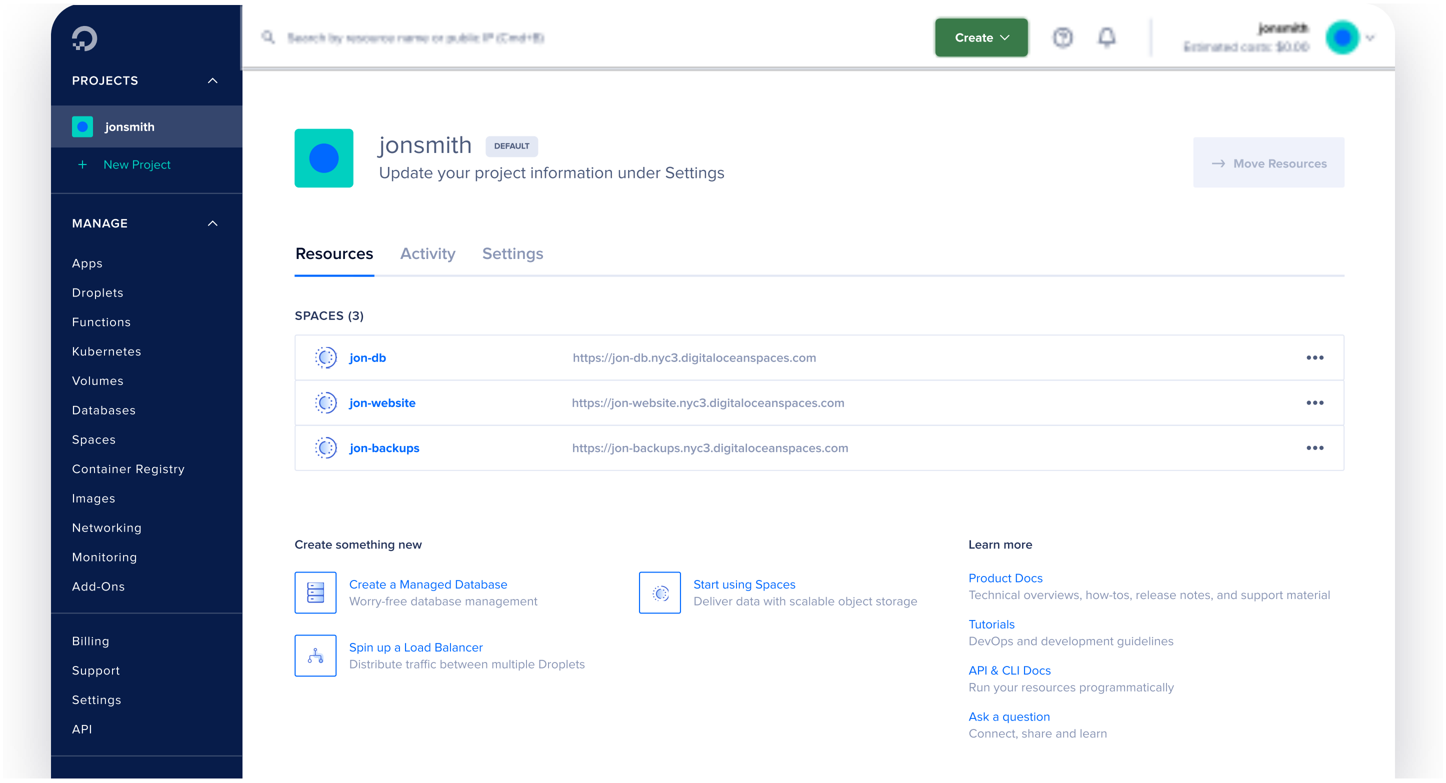
Cloudways   Virtual Machines   Managed Databases   Managed Kubernetes

Block Storage   Object Storage   Marketplace   VPC   Load Balancers

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

Learn more →


PROJECTS
jonsmith
New Project
MANAGE
Apps
Droplets
Functions
Kubernetes
Volumes
Databases
Spaces
Container Registry
Images
Networking
Monitoring
Add-Ons
Billing
Support
Settings
API
Create
jonsmith
DEFAULT
Update your project information under Settings
Move Resources
Resources
Activity
Settings
SPACES (3)
jon-db https://jon-db.nyc3.digitaloceanspaces.com
jon-website https://jon-website.nyc3.digitaloceanspaces.com
jon-backups https://jon-backups.nyc3.digitaloceanspaces.com
Create something new
Create a Managed Database
Worry-free database management
Start using Spaces
Deliver data with scalable object storage
Spin up a Load Balancer
Distribute traffic between multiple Droplets
Learn more
Product Docs
Technical overviews, how-tos, release notes, and support material
Tutorials
DevOps and development guidelines
API & CLI Docs
Run your resources programmatically
Ask a question
Connect, share and learn

Get started for free

Enter your email to get $200 in credit for your first 60 days with DigitalOcean.

Email address

Send My Promo

New accounts only. By submitting your email you agree to our Privacy Policy.

Company

Products

Community

Solutions

Contact