Wildcard Certs via LetsEncrypt

Finally!

See also:

- https://www.codementor.io/slavko/generating-letsencrypt-wildcard-certificate-with-certbot-hts4aee8u
- https://community.letsencrypt.org/t/certbot-the-currently-selected-acme-ca-endpoint-does-not-support-issuing-wildcard-certificates/55667
- https://community.letsencrypt.org/t/google-domains-dns-api-support-not-google-cloud-dns/55480

Go to https://domains.google.com and create new wildcard record for *.whatever.

Clone the certbot repo and install the package (I'm using Docker).

```
$ git clone https://github.com/certbot/certbot
$ cd certbot
$ docker run -v `pwd`:/certbot -it python bash
# cd /certbot
# python setup.py install
```

Since Google Domains doesn't have an API, need to use the manual feature:

```
$ certbot certonly --manual -d *.whatever.ndslabs.org --agree-tos --no-bootstrap --server https://acme-v02.api.
letsencrypt.org/directory
...
Please deploy a DNS TXT record under the name
_acme-challenge.whatever.ndslabs.org with the following value:

XXuXXmIvjuvCNa-cXXoX4Xy0c2VDkbQrNnp3V4qrnXo

Before continuing, verify the record is deployed.
```

Now, go to Google domains and add a TXT record for *.whatever using the above value:

```
Login to Google Domains page.
Click DNS tab.
Scroll down to Custom resource records.
Name: *.whatever
Type:  TXT
TTL: 1h
Data: Value from above
```

Wait until the name resolves:

```
$ nslookup -type=TXT _acme-challenge.whatever.ndslabs.org
Server:                192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
_acme-challenge.whatever.ndslabs.org        text = "XXuXXmIvjuvCNa-cXXoX4Xy0c2VDkbQrNnp3V4qrnXo"
```

In the certbot window, Press Enter to Continue. This will create the certificate in /etc/letsencrypt/live/domain.

```
cp -r /etc/letsencrypy/archive/domain .
```

Exit the container

```
cd domain
kubectl create secret generic ndslabs-tls-secret --from-file=tls.crt=fullchain1.pem --from-file=tls.
key=privkey1.pem --namespace=default
```

A few things to note:

- Certificates are only valid for 90 days (https://community.letsencrypt.org/t/lets-encrypt-in-numbers-limits-restrictions-features/37113)
- Certbot can be used to automate certificate renewal
- cert-manager – successor to kube-lego – added support with https://github.com/jetstack/cert-manager/pull/309
    - Merged 2 days ago!
    - In theory, we could use cert-manager to generate and maintain wildcard certs via letsencrypt