BLUEDOT

OPERATIONS NOTE

Let's Encrypt on the BNG

Operations Note — Management-Interface Certificates

When Let's Encrypt is a viable option for the BNG management certificate, how to request one using either the HTTP-01 or DNS-01 challenge, and how to keep it renewed automatically. Most deployments sit on a management LAN with no inbound reachability and should use the alternatives instead.

Document Type	BlueDot Operations Note
Component	QuantumTouch — management TLS
Audience	Operators & Deployment Engineers
See Also	*Trusting the BNG TLS Certificate*
Last Updated	28 June 2026
Classification	**BlueDot Proprietary**
Date	October 2026

© 2026 BlueDot · All rights reserved.

Contents

When Let's Encrypt makes sense 3
HTTP-01 challenge (easiest, needs inbound port 80) 3
DNS-01 challenge (no inbound HTTP required) 4
Automatic renewal 4
Troubleshooting 5

What this covers: when Let's Encrypt is a viable option for the BNG management certificate, how to request a certificate using either the HTTP-01 or DNS-01 challenge method, and how to keep it renewed automatically.

Last updated: 2026-06-28

When Let's Encrypt makes sense

Let's Encrypt issues certificates for hostnames that are publicly resolvable and reachable. That requirement rules it out for most BNG deployments.

Good fit: - The BNG has a public IP address, OR - A public DNS record (e.g. `bng-site-a.example.com`) points at the box's management IP, AND that IP is reachable from the internet on port 80 or 443.

Not a fit (most deployments): - The box is on a management LAN (e.g. 192.168.x.x / 10.x.x.x) with no inbound internet access. - The management interface is behind a NAT with no inbound port forwarding.

If Let's Encrypt is not viable, see `trust-the-cert.md` for the alternatives (import an existing certificate, or distribute the local CA).

HTTP-01 challenge (easiest, needs inbound port 80)

The HTTP-01 method is the simplest: certbot starts a temporary HTTP server on port 80 and Let's Encrypt's servers fetch a verification token from it. The box must accept inbound TCP on port 80 from the internet during the request.

Steps:

1. In the BNG management UI, navigate to **About** → **Manage TLS cert** → **Generate certbot script** tab.
2. Select the **HTTP-01** option.
3. Fill in: - **Hostname** — the public DNS name pointing at this box (e.g. `bng-site-a.example.com`). - **Email** — your email address for Let's Encrypt renewal notices.
4. Click **Copy script** and then SSH into the box:

   ```
   ssh lanner@<management-ip>
   ```

5. Paste the script into a file and run it with sudo:

   ```
   sudo bash certbot-setup.sh
   ```

The script installs certbot (if absent), requests the certificate using the standalone HTTP-01 method, and copies the resulting cert and key into `/var/lib/quantum-controller/tls/`. The gunicorn process reloads the certificate without a full restart.

1. Verify the new certificate in your browser — it should now show as trusted.

Firewall note: port 80 only needs to be open during the initial request and during each renewal. You can open it temporarily via your firewall or provider security group, run certbot, then close it again.

DNS-01 challenge (no inbound HTTP required)

If the box is not reachable on port 80 from the internet, use the DNS-01 challenge. Certbot asks you to add a TXT record to your DNS zone to prove you control the domain. No inbound connection to the box is needed.

1. In the BNG management UI: **About** → **Manage TLS cert** → **Generate certbot script** → select the **DNS-01** tab.
2. Fill in the same hostname and email fields, then copy and run the script:

```
sudo bash certbot-dns01-setup.sh
```

3. Certbot will pause and display a TXT record to add, for example:

```
Please deploy a DNS TXT record under the name:
_acme-challenge.bng-site-a.example.com
with the following value: <token>
```

4. Add that TXT record in your DNS provider's control panel.
5. Wait at least 60 seconds for the record to propagate, then press Enter in the certbot terminal to continue. Certbot verifies the record and issues the certificate.
6. The script copies the cert and key to `/var/lib/quantum-controller/tls/` and signals gunicorn to reload.

Automatic renewal

Let's Encrypt certificates expire after 90 days. Add a cron job to the root crontab on the box to renew automatically:

```
sudo crontab -e
```

Add this line (adjust `<dns>` to your hostname):

```
0 3 * * * certbot renew --post-hook 'cp /etc/letsencrypt/live/<dns>/fullchain.pem /var/lib/quantum-controller/tls/
webui-cert.pem && cp /etc/letsencrypt/live/<dns>/privkey.pem /var/lib/quantum-controller/tls/webui-key.pem'
```

This runs at 3 AM every day. Certbot only contacts Let's Encrypt when the certificate is within 30 days of expiry, so daily invocations are harmless. The `--post-hook` copies the renewed files into the location gunicorn reads, which triggers an in-place reload.

To test that the renewal command works without actually contacting Let's Encrypt:

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

Troubleshooting

"Error: too many certificates already issued" Let's Encrypt enforces a rate limit of 5 duplicate certificates per week per registered domain. During testing, use the staging API instead:

```
sudo certbot certonly --staging --standalone -d bng-site-a.example.com -m you@example.com --agree-tos
```

Staging certificates are not trusted by browsers but allow you to verify the challenge flow works before using the production API.

"DNS-01 challenge failed — TXT record not found" DNS propagation can be slow. Wait 60 seconds and retry. If using a provider with high TTLs or slow propagation (some registrars take several minutes), wait longer. You can check current propagation with:

```
dig TXT _acme-challenge.bng-site-a.example.com @8.8.8.8
```

The token should appear in the output before certbot will succeed.

"Certificate applies but browser still shows the old warning" The gunicorn reload is immediate, but your browser may be caching the old certificate. Force a full reload:

- **Chrome / Edge**: `Ctrl+Shift+R` (Windows/Linux) or `Cmd+Shift+R` (macOS).
- **Firefox**: `Ctrl+Shift+R` or `Cmd+Shift+R`.

If the warning persists after a hard refresh, confirm gunicorn picked up the new certificate by checking the cert directly:

```
openssl s_client -connect <hostname>:8443 -servername <hostname> < /dev/null \
  | openssl x509 -noout -dates
```

The `notAfter` date should match the newly issued certificate's expiry.