



Installation Guide

version 1.8.0

Web Server, LLC

Dec 20, 2024

CONTENTS

1	Annotation	1
2	General Information	2
3	Package Installation of Angie PRO	4
3.1	Distros	4
3.1.1	Alma, CentOS, MSVSphere, Oracle, RED OS, Rocky, ROSA, SberLinux	4
3.1.2	Alpine	7
3.1.3	ALT Linux	8
3.1.4	Astra SE	9
3.1.5	Debian, Ubuntu	10
3.1.6	FreeBSD	11
3.1.7	openSUSE	12
3.2	Extras	13
3.2.1	Console Light	13
3.2.2	Dynamic Modules	14
3.3	License File	16
4	Intellectual Property Rights	17

CHAPTER

ONE

ANNOTATION

This document contains a step-by-step guide for deploying Angie PRO. Angie PRO is an efficient, powerful, and scalable web server.

GENERAL INFORMATION

Angie PRO is the only commercial web server developed and localized in Russia.

A web server is a class of software that provides access to network resources via the HTTP protocol to end users. Angie PRO, for example, can be used to operate websites, mobile applications, self-service kiosks in the subway, and multimedia systems on long-distance trains. Every time a user opens a website, uses a mobile application, interacts with a self-service kiosk in the subway, or even with a multimedia system on the "Sapsan" train, the user's request can be processed by Angie PRO.

Angie PRO is:

- **A general-purpose web server.** Written in C.
- **An L4-L7 load balancer.** Allows load balancing between servers for both TCP/UDP protocols and HTTP.
- **A proxy and caching server.** Enables faster operation of web services through a flexible caching mechanism.
- **Available on all popular platforms.** Compiled and tested on Alpine, Debian, Oracle, RED OS, Rocky, and Ubuntu.
- **High performance.** One of the most efficient web servers in the world.

Why choose Angie PRO:

- **Compatibility with NGINX OSS.** Angie PRO is fully compatible with Nginx, allowing any existing Nginx user to transition to Angie PRO without significant costs or service downtime.
- **Enhanced statistics and real-time monitoring.** Angie PRO offers complete real-time server load monitoring, enabling dynamic configuration management based on load profiles and ensuring full service availability.
- **Dynamic configuration of proxied server groups.** Allows management of proxied server group settings through a convenient REST interface without service interruption.
- **Cache element removal.** Provides the ability to remove cache elements via a user-friendly API without service downtime.
- **Active health checks for proxied servers.** Checks for "liveness" and proxies only to those groups of proxied servers that respond according to a specified algorithm.
- **Dynamic key-value storage.** Enables dynamic management of Angie PRO configuration variables via HTTP API.
- **Dynamic DNS updates.**
- **Session-affinity proxying.**
- **Repository with dynamic third-party modules.** Angie PRO supports most NGINX third-party modules and allows for seamless installation, guaranteeing functionality and support.
- **Shared memory zone synchronization.** Capability to use cache zones, `limit_req`, etc., in the Angie PRO cluster.

- **Hiding or personal branding of the server name in response headers.** Ability to change or hide the name and version of the web server from users.

A list of foreign software with similar functional characteristics to Angie PRO includes Nginx, Nginx Plus, Apache, Envoy, products utilizing NGINX solutions (OpenResty, Tengine, Cloudflare), and Yandex's cloud solutions.

PACKAGE INSTALLATION OF ANGIE PRO

To access the package repository, you need to sign a contract and purchase a license. For questions about licenses, contracts, and custom builds, contact:

- info@wbsrv.ru
- <https://angie.software/>
- +7 (495) 120 50 33

Then, configure the repository for your distro's package manager to install and update Angie PRO and the *dynamic modules* you need. Finally, install the *license file* and update the limits.

3.1 Distros

Name	Versions	Architectures
<i>AlmaLinux</i>	9, 8	x86-64, arm64
<i>Alpine</i>	3.21, 3.20, 3.19, 3.18	x86-64, arm64
<i>Alt</i>	10 8	x86-64, arm64 x86-64
<i>Astra SE</i>	4.7 1.8, 1.7	arm64 x86-64
<i>CentOS</i>	9	x86-64, arm64
<i>Debian</i>	12, 11	x86-64, arm64
<i>FreeBSD</i>	14, 13	x86-64, arm64
<i>MSVSphere</i>	9	x86-64
<i>openSUSE</i>	15	x86-64, arm64
<i>Oracle Linux</i>	9, 8	x86-64, arm64
<i>RED OS</i>	8 7	x86-64 x86-64, arm64
<i>Rocky Linux</i>	9, 8	x86-64, arm64
<i>ROSA</i>	Chrome 12 Fresh 12	x86-64, arm64 x86-64
<i>SberLinux</i>	9	x86-64
<i>Ubuntu</i>	24.04, 22.04, 20.04	x86-64, arm64


3.1.1 Alma, CentOS, MSVSphere, Oracle, RED OS, Rocky, ROSA, SberLinux

1. Create the `/etc/ssl/angie/` directory:

```
$ sudo mkdir -p /etc/ssl/angie/
```

2. Transfer the files you received with your license:

File Type	Original Name	Where
Certificate	<code>angie-repo.crt</code>	<code>/etc/ssl/angie/angie-repo.crt</code>
Private Key	<code>angie-repo.key</code>	<code>/etc/ssl/angie/angie-repo.key</code>

 **Hint**

If you have the license but not these files, email us at support@angie.software.

- To add the repo, create the following file named `/etc/yum.repos.d/angie.repo`:

Alma

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/almalinux/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

CentOS

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/centos/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

MSVSphere

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/msvsphere/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

Oracle

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/oracle/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

RED OS

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/redos/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
```

```
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

Rocky

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/rocky/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

ROSA Chrome

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/rosa-chrome/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

ROSA Fresh

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/rosa/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

SberLinux

```
[angie-pro]
name=Angie PRO repo
baseurl=https://download.angie.software/angie-pro/sberlinux/$releasever/
sslclientcert=/etc/ssl/angie/angie-repo.crt
sslclientkey=/etc/ssl/angie/angie-repo.key
gpgcheck=1
enabled=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

4. Install the Angie PRO package:

```
$ sudo yum install -y angie-pro
```

5. (Optional) Install any *extra* packages you need:

```
$ sudo yum install -y <PACKAGE NAME>
```

6. Start the service:

```
$ sudo systemctl start angie
```


- To autostart Angie PRO after server reboot:

```
$ sudo systemctl enable angie
```

3.1.2 Alpine

- Transfer the files you received with your license:

File Type	Original Name	Where
Certificate	angie-repo.crt	/etc/apk/cert.pem
Private Key	angie-repo.key	/etc/apk/cert.key

Hint

If you have the license but not these files, email us at support@angie.software.

- Install the prerequisites for adding the Angie PRO repo:

```
$ sudo apk update
$ sudo apk add curl ca-certificates
```

- Download the public key of the Angie PRO repo for package verification:

```
$ sudo curl -o /etc/apk/keys/angie-signing.rsa \
  https://angie.software/keys/angie-signing.rsa
```

- Add the Angie PRO repo:

```
$ echo "https://download.angie.software/angie-pro/alpine/v$(egrep -o \
  '[0-9]+\.[0-9]+' /etc/alpine-release)/main" \
  | sudo tee -a /etc/apk/repositories > /dev/null
```

- Update the repo indexes:

```
$ sudo apk update
```

- Install the Angie PRO package:

```
$ sudo apk add angie-pro
```

- (Optional) Install any *extra* packages you need:

```
$ sudo apk add <PACKAGE NAME>
```

- Start the service:

```
$ sudo service angie start
```

- To autostart Angie PRO on server reboot:

```
$ sudo rc-update add angie
```

3.1.3 ALT Linux

1. Create the `/etc/ssl/angie` directory:

```
$ sudo mkdir -p /etc/ssl/angie/
```

2. Transfer the files you received with your license:

File Type	Original Name	Where
Certificate	angie-repo.crt	/etc/ssl/angie/angie-repo.crt
Private Key	angie-repo.key	/etc/ssl/angie/angie-repo.key

Hint

If you have the license but not these files, email us at support@angie.software.

3. Download the public key of the Angie PRO repo for package verification:

```
$ curl -o ~/angie-signing.gpg https://angie.software/keys/angie-signing.gpg && \
  sudo gpg --no-default-keyring --keyring /usr/lib/alt-gpgkeys/pubring.gpg -
  →-import ~/angie-signing.gpg
```

4. Save the key's signature:

```
$ echo 'simple-key "angie-pro" {
  Fingerprint "EB8EAF3D4EF1B1ECF34865A2617AB978CB849A76";
  Name "Angie PRO (Signing Key) <devops@tech.wbsrv.ru>";
}' | sudo tee /etc/apt/vendors.list.d/angie.list > /dev/null
```

5. Add the Angie PRO repo:

ALT 10

```
$ echo "rpm [angie-pro] https://download.angie.software/angie-pro/altlinux/10/
  →$(uname -m) main" \
  | sudo tee /etc/apt/sources.list.d/angie.list > /dev/null
```

ALT 10 SP

```
$ echo "rpm [angie-pro] https://download.angie.software/angie-pro/altlinux-sp/10/
  → $(uname -m) main" \
  | sudo tee /etc/apt/sources.list.d/angie.list > /dev/null
```

ALT 8 SP

```
$ echo "rpm [angie-pro] https://download.angie.software/angie-pro/altlinux-sp/8/
  →$(uname -m) main" \
  | sudo tee /etc/apt/sources.list.d/angie.list > /dev/null
```

6. Save an apt configuration file for the Angie PRO repo as `/etc/apt/apt.conf.d`:

```
$ ( echo 'Acquire::https::Verify-Peer "true";';
  echo 'Acquire::https::Verify-Host "true";';
  echo 'Acquire::https::SslCert "/etc/ssl/angie/angie-repo.crt";';
  echo 'Acquire::https::SslKey "/etc/ssl/angie/angie-repo.key";';
  ) | sudo tee -a /etc/apt/apt.conf >/dev/null
```

- Update the repo indexes:

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

- Install the Angie PRO package:

```
$ sudo apt-get install -y angie-pro
```

- (Optional) Install any *extra* packages you need:

```
$ sudo apt-get install -y <PACKAGE NAME>
```

- Start the service:

```
$ sudo systemctl start angie
```

- To autostart Angie PRO on server reboot:

```
$ sudo systemctl enable angie
```

3.1.4 Astra SE

- Create the `/etc/ssl/angie/` directory:

```
$ sudo mkdir -p /etc/ssl/angie/
```

- Transfer the files you received with your license:

File Type	Original Name	Where
Certificate	angie-repo.crt	/etc/ssl/angie/angie-repo.crt
Private Key	angie-repo.key	/etc/ssl/angie/angie-repo.key

Hint

If you have the license but not these files, email us at support@angie.software.

Restrict the access to the directory and the files:

```
$ sudo chown -R _apt:nogroup /etc/ssl/angie/
```

- Install the prerequisites for adding the Angie PRO repo:

```
$ sudo apt-get update
$ sudo apt-get install -y apt-transport-https lsb-release \
  ca-certificates curl gnupg2
```

- Download the public key of the Angie PRO repo for package verification:

```
$ sudo curl -o /etc/apt/trusted.gpg.d/angie-signing.gpg \
  https://angie.software/keys/angie-signing.gpg
```

- Add the Angie PRO repo:

```
$ echo "deb https://download.angie.software/angie-pro/astra-se/$(egrep -o \
  '[0-9]+\.[0-9]+' /etc/astra_version) unstable main" \
  | sudo tee /etc/apt/sources.list.d/angie.list > /dev/null
```

- To configure the repo, create a file named `/etc/apt/apt.conf.d/90download-angie` with the following contents:

```
Acquire::https::download.angie.software::Verify-Peer "true";
Acquire::https::download.angie.software::Verify-Host "true";
Acquire::https::download.angie.software::SslCert      "/etc/ssl/angie/angie-repo.
→cert";
Acquire::https::download.angie.software::SslKey       "/etc/ssl/angie/angie-repo.
→key";
```

- Update the repo indexes:

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

- (Optional) When running a Closed Software Environment (CSE), install the key package for Angie PRO binary verification:

```
$ sudo apt-get install -y angie-digsig-key
```

Update the CSE:

```
$ sudo update-initramfs -uk all
```

Then restart the server:

```
$ sudo shutdown -r now
```

- Install the Angie PRO package:

```
$ sudo apt-get install -y angie-pro
```

- (Optional) Install any *extra* packages you need:

```
$ sudo apt-get install -y <PACKAGE NAME>
```

3.1.5 Debian, Ubuntu

- Create the `/etc/ssl/angie/` directory:

```
$ sudo mkdir -p /etc/ssl/angie/
```

- Transfer the files you received with your license:

File Type	Original Name	Where
Certificate	<code>angie-repo.crt</code>	<code>/etc/ssl/angie/angie-repo.crt</code>
Private Key	<code>angie-repo.key</code>	<code>/etc/ssl/angie/angie-repo.key</code>

Hint

If you have the license but not these files, email us at support@angie.software.

Restrict the access to the directory and the files:

```
$ sudo chown -R _apt:nogroup /etc/ssl/angie/
```

- Install the prerequisites for adding the Angie PRO repo:

```
$ sudo apt-get update
$ sudo apt-get install -y apt-transport-https lsb-release \
    ca-certificates curl gnupg2
```

- Download the public key of the Angie PRO repo for package verification:

```
$ sudo curl -o /etc/apt/trusted.gpg.d/angie-signing.gpg \
    https://angie.software/keys/angie-signing.gpg
```

- Add the Angie PRO repo:

```
$ echo "deb https://download.angie.software/angie-pro/$(. /etc/os-release && \
→echo "$ID/$VERSION_ID $VERSION_CODENAME") main" \
    | sudo tee /etc/apt/sources.list.d/angie.list > /dev/null
```

- To configure the repo, create a file named `/etc/apt/apt.conf.d/90download-angie` with the following contents:

```
Acquire::https::download.angie.software::Verify-Peer "true";
Acquire::https::download.angie.software::Verify-Host "true";
Acquire::https::download.angie.software::SslCert      "/etc/ssl/angie/angie-repo.
→crt";
Acquire::https::download.angie.software::SslKey      "/etc/ssl/angie/angie-repo.
→key";
```

- Update the repo indexes:

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

- Install the Angie PRO package:

```
$ sudo apt-get install -y angie-pro
```

- (Optional) Install any *extra* packages you need:

```
$ sudo apt-get install -y <PACKAGE NAME>
```

3.1.6 FreeBSD

- To add the Angie PRO repo, create these directories:

```
$ sudo mkdir -p /usr/local/etc/pkg/angie/ /usr/local/etc/pkg/repos/
```

- To configure the repo, create a file named `/usr/local/etc/pkg/repos/angie.conf` with the following contents:


```
angie: {
    url: "https://download.angie.software/angie-pro/freebsd/${VERSION_MAJOR}/${
→ARCH}",
    signature_type: "pubkey",
    pubkey: "/usr/local/etc/pkg/angie/angie-signing.rsa",
    enabled: yes
}
```

- Download the public key of the Angie PRO repo for package verification:

```
$ sudo curl -o /usr/local/etc/pkg/angie/angie-signing.rsa \
    https://angie.software/keys/angie-signing.rsa
```

- Transfer the files you received with your license:

File Type	Original Name	Where
Certificate	angie-repo.crt	/usr/local/etc/pkg/angie/angie-repo.crt
Private Key	angie-repo.key	/usr/local/etc/pkg/angie/angie-repo.key

 **Hint**

If you have the license but not these files, email us at support@angie.software.

- Add the certificate and the key to the package manager's configuration:

```
$ echo '
PKG_ENV: {
  SSL_CLIENT_CERT_FILE: "/usr/local/etc/pkg/angie/angie-repo.crt",
  SSL_CLIENT_KEY_FILE:  "/usr/local/etc/pkg/angie/angie-repo.key"
}' | sudo tee -a /usr/local/etc/pkg.conf > /dev/null
```

- Update the repo indexes:

```
$ sudo pkg update
```

- Install the Angie PRO package:

```
$ sudo pkg install -y angie-pro
```

- (Optional) Install any *extra* packages you need:

```
$ sudo pkg install -y <PACKAGE NAME>
```

- Start the service:

```
$ sudo service angie start
```

- To autostart Angie PRO on server reboot:

```
$ sudo sysrc angie_enable=YES
```

3.1.7 openSUSE

- Create the `/etc/ssl/angie` directory:

```
$ sudo mkdir -p /etc/ssl/angie/
```

- Transfer the files you received with your license:

File Type	Original Name	Where
Certificate	angie-repo.crt	/etc/ssl/angie/angie-repo.crt
Private Key	angie-repo.key	/etc/ssl/angie/angie-repo.key

And bundle them into `/etc/ssl/angie/angie-repo-bundle.crt`:

```
$ cat /etc/ssl/angie/angie-repo.crt /etc/ssl/angie/angie-repo.key | \
sudo tee -a /etc/ssl/angie/angie-repo-bundle.crt > /dev/null
```

Hint

If you have the license but not these files, email us at support@angie.software.

- To add the repo, create a file named `/etc/zypp/repos.d/angie.repo` with the following contents:

```
[angie-pro]
enabled=1
autorefresh=1
baseurl=https://download.angie.software/angie-pro/opensuse/$releasever_major?ssl_
↪clientcert=/etc/ssl/angie/angie-repo-bundle.crt&ssl_verify=peer
gpgcheck=1
gpgkey=https://angie.software/keys/angie-signing.gpg.asc
```

- Update the repo indexes:

```
$ sudo zypper refresh
```

- Install the Angie PRO package:

```
$ sudo zypper install -y angie-pro
```

- (Optional) Install any *extra* packages you need:

```
$ sudo zypper install -y <PACKAGE NAME>
```

- Start the service:

```
$ sudo systemctl start angie
```

- To autostart Angie PRO after server reboot:

```
$ sudo systemctl enable angie
```

3.2 Extras

Besides the packages that provide the basic functionality, we also publish a few extra packages, both our own and built from curated third-party sources.

3.2.1 Console Light

Console Light is a lightweight, real-time monitoring interface for Angie PRO, published as `angie-pro-console-light` in our repositories. It is installed in the same way as the `angie` package in the steps above; see the configuration steps in monitoring.

3.2.2 Dynamic Modules

To extend the basic functionality of Angie PRO, you can add various dynamic modules, available as ready-made packages from our repository:

angie-pro-module-image-filter	Adds a filter to transform JPEG, GIF, PNG, and WebP images.
angie-pro-module-njs: HTTP JS module, streaming JS module	Enables using njs (a JavaScript subset) in Angie configuration in the <code>http</code> and <code>stream</code> contexts, respectively.
angie-pro-module-perl	Enables writing <code>location</code> and variable handlers in Perl, and also invoking Perl from SSI.
angie-pro-module-xslt	Adds a filter to transform XML responses with XSLT stylesheets.

We also build and publish in our repository the following third-party module packages:

angie-pro-module-auth-jwt	Adds client JWT authentication.
angie-pro-module-auth-ldap	Adds support for authentication against multiple LDAP servers.
angie-pro-module-auth-spnego	Adds SPNEGO, GSSAPI support.
angie-pro-module-brotli	Enables Brotli for dynamic and static response compression.
angie-pro-module-cache-purge	Enables purging FastCGI, proxy, SCGI, and uWSGI cache contents.
angie-pro-module-dav-ext	Extends WebDAV method support, adding PROPFIND and OPTIONS.
angie-pro-module-dynamic-limit-req	Enables dynamically locking IPs and releasing them periodically.
angie-pro-module-echo	Enables <code>echo</code> , <code>sleep</code> , <code>time</code> , <code>exec</code> , and other shell-style commands in the configuration file.
angie-pro-module-enhanced-memcached	Enhances the capabilities of the built-in Memcached module.
angie-pro-module-eval	Enables capturing subrequest response bodies in variables.
angie-pro-module-geoip2: http_geoip2, stream_geoip2	Enables geodata lookup in the MaxMind GeoIP2 database.
angie-pro-module-headers-more	Enables setting and clearing input and output headers.
angie-pro-module-http-auth-radius	Enables Radius support.
angie-pro-module-keyval	Enables variables created from key-value pairs.
angie-pro-module-lua: http_lua_module, stream_lua_module	Enable using Lua in Angie PRO configuration in the <code>http</code> and <code>stream</code> contexts, respectively.
angie-pro-module-modsecurity	Adds a connector that enables ModSecurity rules.
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p>Note</p> <p>The package needs extra setup.</p> </div>	
angie-pro-module-ndk	Adds Nginx Development Kit (NDK) for module development.
angie-pro-module-otel	Enables sending telemetry data to an OpenTelemetry collector.
angie-pro-module-opentracing	Adds distributed tracing of Angie PRO requests via OpenTracing; includes data export plugins for Zipkin and DataDog.
angie-pro-module-postgres	Enables direct communication with PostgreSQL databases.
angie-pro-module-redis2	Enables Redis 2.0 for HTTP upstreams.
angie-pro-module-rtmp	Enables RTMP for live streaming and video on demand.
angie-pro-module-set-misc	Adds various <code>set_xxx</code> directives to the Rewrite module.
angie-pro-module-sub	Enables regex string substitutions in the HTTP response body.
angie-pro-module-testcookie	Enables robot mitigation using a cookie-based challenge-response mechanism.
angie-pro-module-upload	Adds the <code>multipart/form-data</code> encoding (RFC 1867) for file upload, enabling resumable uploads.
angie-pro-module-vod	Enables repackaging <code>.mp4</code> files for HLS, HDS, MSS, and DASH streaming.
angie-pro-module-vts: module-vts, module-sts, module-stream-sts	Packs the three listed modules for traffic monitoring.
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p>Note</p> <p>Their respective template files are installed as follows:</p> <pre style="margin: 0;">/usr/share/angie-pro-module-vts/status.compress.html /usr/share/angie-pro-module-vts/status.template.html /usr/share/angie-pro-module-vts/stream/status. ↳compress.html /usr/share/angie-pro-module-vts/stream/status. ↳template.html</pre> </div>	
angie-pro-module-zip	Enables packaging ZIP archives dynamically.
angie-pro-module-zstd	Enables Zstandard compression.

To use an installed module in the configuration, load it with the `load_module` directive.

3.3 License File

To configure the license for Angie PRO:

1. Save the license file as `/etc/angie/license.pem`, setting the permissions you use for your client certificates.
2. Verify the license is valid; otherwise, check the details:

```
$ sudo angie -t

angie: Valid license found:
angie:   - owner: CN=Angie Client License
angie:   - period: Jul  8 21:00:00 2024 GMT .. Jul 17 20:59:59 2024 GMT
angie:
angie: Limitations:
angie:   - worker_processes_limit: 8
angie:   - worker_connections_limit: 0
```

3. Monitor the console and logs for any licensing issues. If the license expires mid-operation, Angie PRO periodically logs warnings indicating this. Additionally, on reload, it reports configuration errors if, for example, the number of worker processes set forth in the license terms is exceeded.
4. Update `/etc/angie/angie.conf`; after installation, it has two settings that limit the operation:

```
worker_processes 1;
worker_connections 256;
```

After saving the license file, update the settings according to your license terms, for example:

```
worker_processes 8;
worker_connections 65535;
```

**CHAPTER
FOUR**

INTELLECTUAL PROPERTY RIGHTS

The documentation for the software product Angie PRO is the intellectual property of Web Server, LLC. The documentation was created as a result of modifications (revisions) to the documentation for the software product Nginx.