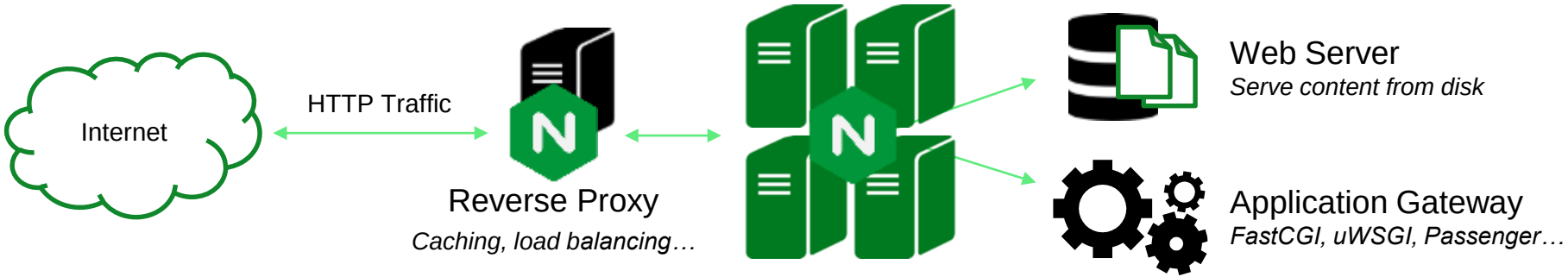
NGINX: Basics and Best Practices

NGINX

NGINX Overview



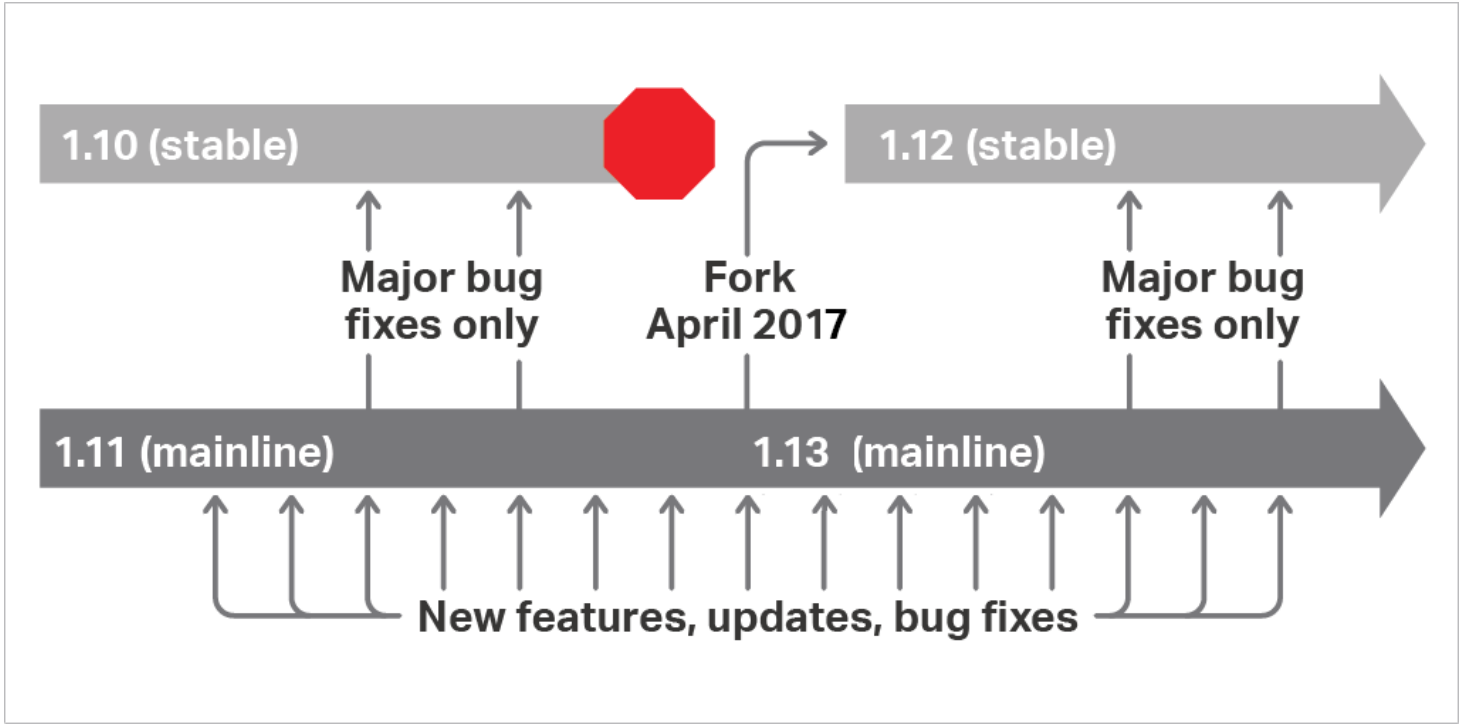
Agenda

- Installing NGINX and NGINX Plus
- Basic Configurations
- Improving Performance and Reliability
- Debugging and Troubleshooting

NGINX Installation Options

- Official NGINX repo
  - Mainline (recommended) – Actively developed; new minor releases made every 4-6 weeks with new features and enhancements.
  - Stable − Updated only when critical issues or security vulnerabilities need to be fixed.
- OS vendor and other third-party repos
  - Not as frequently updated; Debian Jessie has NGINX 1.6.2
  - Typically built off NGINX Stable branch

NGINX Mainline vs. Stable



NGINX Installation: Debian/Ubuntu

Create **/etc/apt/sources.list.d/nginx.list** with the following contents:

```
deb http://nginx.org/packages/mainline/OS/ CODENAME nginx
deb-src http://nginx.org/packages/mainline/OS/ CODENAME nginx
```

- *OS* – `ubuntu` or `debian` depending on your distro
- *CODENAME* –
  - With `debian`: `wheezy`, `jessie`, or `stretch` (7.0, 8.0, 9.0)
  - With `ubuntu`: `precise`, `trusty`, `xenial`, or `yakkety` (12.04, 14.04, 16.04, 16.10)

```
$ wget http://nginx.org/keys/nginx_signing.key
$ apt-key add nginx_signing.key
$ apt-get update
$ apt-get install –y nginx
```

NGINX Installation: CentOS/Red Hat

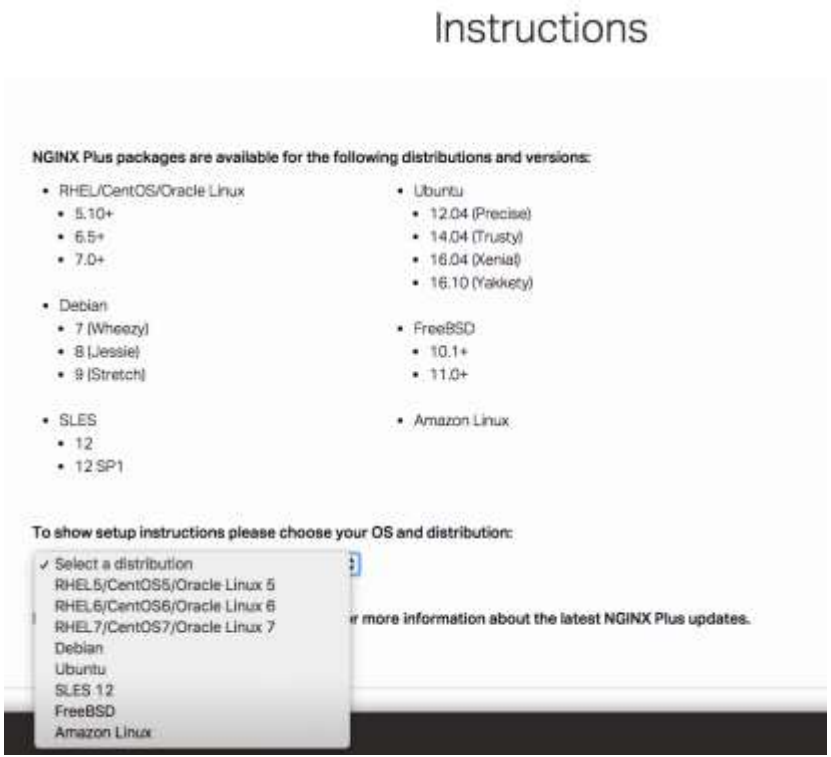
Create **/etc/yum.repos.d/nginx.repo** with the following contents:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/mainline/OS/OSRELEASE/$basearch/
gpgcheck=0
enabled=1
```

- *OS* – `centos` or `rhel` depending on your distro
- *OSRELEASE* – `6` or `7` for 6.x or 7.x versions, respectively

```
$ yum –y install nginx
$ systemctl enable nginx
$ systemctl start nginx
$ firewall-cmd --zone=public --add-port=80/tcp –permanent
$ firewall-cmd --reload
```

NGINX Plus Installation

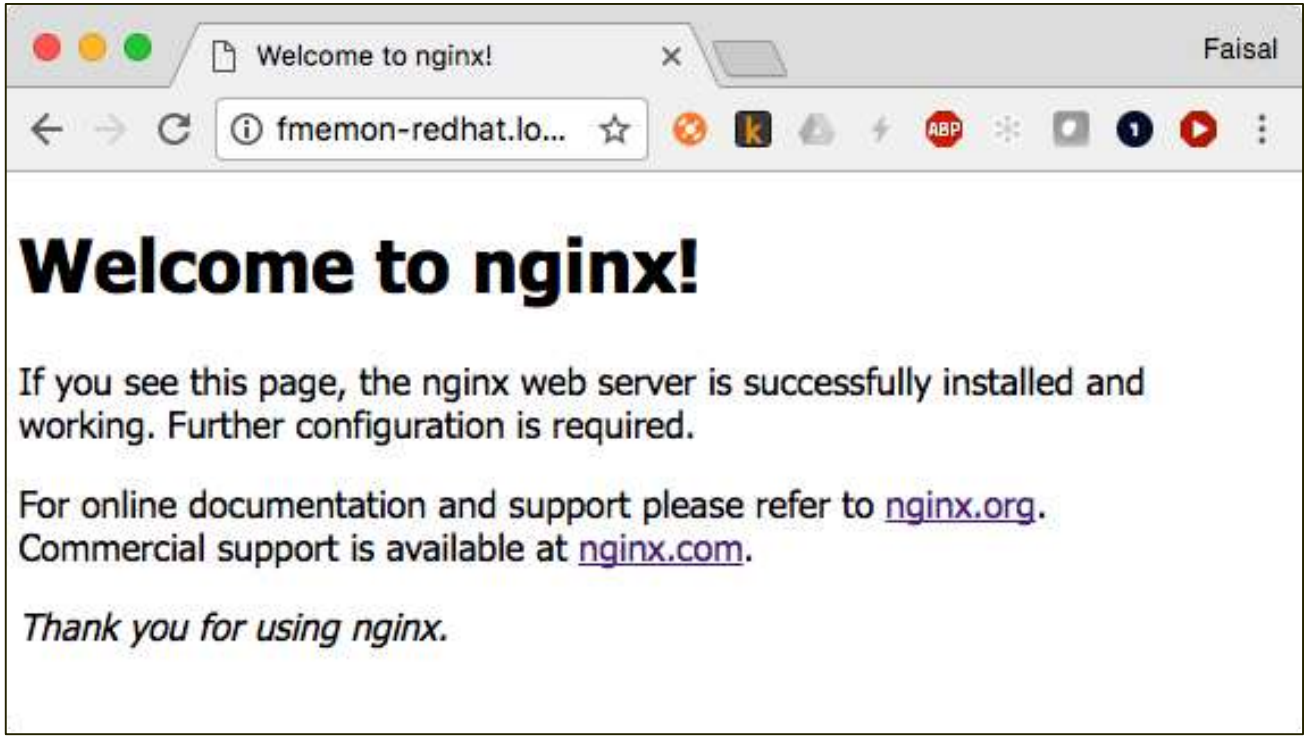


- Visit **cs.nginx.com/repo_setup**
- Select OS from drop-down list
- Instructions similar to OSS installation
- Mostly just using different repo and installing client certificate

Verifying Installation

```
$ nginx -v
nginx version: nginx/1.13.0
$ ps -ef | grep nginx
root       1088      1  0 19:59 ?        00:00:00 nginx: master process
/usr/sbin/nginx -c /etc/nginx/nginx.conf
nginx      1092   1088  0 19:59 ?        00:00:00 nginx: worker process
```

Verifying Installation



Key NGINX Commands

- `nginx -t` Check if NGINX configuration is ok
- `nginx -s reload` Gracefully reload NGINX processes
- `nginx -V` Similar to `-v`, but with more detailed information
- `nginx -T` Dump full NGINX configuration
- `nginx -h` Display NGINX help menu
- After config change, test and reload : `nginx -t && nginx -s reload`

NGINX Installation Misc

- For more installation details, see http://nginx.org/en/linux_packages.html
  - List of all supported distros and CPUs
  - SUSE Linux installation instructions

- For NGINX Plus, see https://cs.nginx.com/repo_setup
  - List of all supported distros and CPUs, including FreeBSD

Agenda

- Installing NGINX and NGINX Plus
- Basic Configurations
- Improving Performance and Reliability
- Debugging and Troubleshooting

Key Files and Directories

- **/etc/nginx/** − Parent directory for all NGINX configuration
- **/etc/nginx/nginx.conf** − Top-level NGINX configuration, not modified often
- **/etc/nginx/conf.d/*.conf** − Configuration for virtual servers and upstreams; for example, **www.example.com.conf**

Basic Web Server Configuration

```
server {
    listen       80 default_server;
    server_name  www.example.com;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
}
```

- `server` defines the context for a virtual server
- `listen` specifies IP address/port that NGINX listens on; if no IP address (as here), NGINX binds to all IP addresses on system
- `default_server` specifies to use this server if hostname is not known
- `server_name` specifies hostname of virtual server

`root` specifies that:

www.example.com maps to **/usr/share/nginx/html/index.html**
www.example.com/i/file.txt maps to **/usr/share/nginx/html/i/file.txt**

Basic SSL Configuration

```
server {
    listen       80 default_server;
    server_name  www.example.com;
    return 301 https://$server_name$request_uri;
}
server {
    listen 443 ssl default_server;
    server_name  www.example.com;
    ssl_certificate cert.crt
    ssl_certificate_key cert.key

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
}
```

- Force all traffic to SSL
- Good for SEO
- Use Let’s Encrypt to get free SSL certificates

Basic Reverse Proxy Configuration

```
server {
    location ~ [^/]\.php(/|$) {
    fastcgi_split_path_info ^(.+?\.php)(/.*)$;

    # fastcgi_pass 127.0.0.1:9000;
    fastcgi_pass unix:/var/run/php7.0-fpm.sock;

    fastcgi_index index.php;
    include fastcgi_params;
}
```

- Requires PHP FPM:
  `apt-get install –y php7.0-fpm`
- Can also use PHP 5
- Similar directives available for SCGI and uwsgi
- Additional PHP FPM configuration may be required

Basic Load Balancing Configuration

```
upstream my_upstream {
    server server1.example.com;
    server server2.example.com;
    least_conn;
}
server {
    location / {
        proxy_set_header Host $host;
        proxy_pass http://my_upstream;
    }
}
```

- Default load balancing algorithm is Round Robin
- `least_conn` selects server with fewest active connections
- By default NGINX rewrites `Host` header to name and port of proxied server
- `proxy_set_header` overrides and passes through original client `Host` header
- **`least_time`** factors in connection count and server response time (available in NGINX Plus only)

Basic Caching Configuration

```
proxy_cache_path /path/to/cache levels=1:2
                 keys_zone=my_cache:10m max_size=10g
                 inactive=60m use_temp_path=off;

server {
    location / {
        proxy_cache my_cache;
        proxy_set_header Host $host;
        proxy_pass http://my_upstream;
    }
}
```

- `proxy_cache_path` defines the size, location on disk, and other parameters of the cache
- `proxy_cache` enables caching for the local context

Agenda

- Installing NGINX and NGINX Plus
- Basic Configurations
- Improving Performance and Reliability
- Debugging and Troubleshooting

Modifications to Main nginx.conf

```
user  nginx;
worker_processes auto;

# ...

http {
    # ...

    keepalive_timeout 300s;
    keepalive_requests 100000;
}
```

- Set in main **nginx.conf** file.
- Default value for `worker_processes` varies by system and installation source.
- `auto` means to create one worker process per core. This is recommended for most deployments.
- `keepalive_timeout` controls how long to keep idle connections to clients open. Default: 75 seconds.
- `keeplive_requests` sets the limit on requests by a single client connection before it’s closed.
- `keepalive_*` can also be set per virtual server.

HTTP/1.1 Keepalive to Upstreams

```
upstream my_upstream {
    server server1.example.com;
    keepalive 32;
}
server {
    location / {
        proxy_set_header Host $host;
        proxy_http_version 1.1;
        proxy_set_header Connection "";

        proxy_pass http://my_upstream;
    }
}
```

- `keepalive` enables TCP connection cache
- By default NGINX uses HTTP/1.0 with `Connection: Close`
- `proxy_http_version` upgrades connection to HTTP/1.1
- `proxy_set_header` enables keepalive by clearing `Connection: Close` HTTP header

SSL Session Caching and HTTP/2

```
server {
    listen 443 ssl http2 default_server;
    server_name  www.example.com;

    ssl_certificate cert.crt
    ssl_certificate_key cert.key

    ssl_session_cache   shared:SSL:10m;
    ssl_session_timeout 10m;
}
```

- Improves SSL/TLS performance
- 1 MB session cache can store about 4,000 sessions
- Cache shared across all NGINX workers
- HTTP/2 improves performance
- Note: HTTP/2 requires OpenSSL 1.0.2 to work properly

Improved Caching Configuration

```
proxy_cache_path /path/to/cache levels=1:2
                 keys_zone=my_cache:10m max_size=10g
                 inactive=60m use_temp_path=off;

server {
    location / {
        proxy_cache my_cache;
        proxy_cache_lock on;
        proxy_cache_revalidate on;

        proxy_set_header Host $host;
        proxy_pass http://my_upstream;
    }
}
```

- `proxy_cache_lock` instructs NGINX to send only one request to the upstream when there are multiple cache misses for the same file
- `proxy_cache_revalidate` instructs NGINX to use `If-Modified-Since` when refreshing cache

Load Balancing with Health Checks Configuration

```
upstream my_upstream {
    zone my_upstream 64k;
    server server1.example.com slow_start=30s;
    server server2.example.com slow_start=30s;
}
server {
    location / {
        proxy_set_header Host $host;
        proxy_pass http://my_upstream;
    }
    location @health {
        health_check mandatory;
    }
}
```

- Polls **/health** every 5 seconds
- If response is not 2xx or 3xx, server is marked as failed
- Traffic to recovered/new servers slowly ramps up traffic over 30 seconds
- Many additional configurable parameters
- Exclusive to NGINX Plus

Agenda

- Installing NGINX and NGINX Plus
- Basic Configurations
- Improving Performance and Reliability
- Debugging and Troubleshooting

NGINX Stub Status Module

```
server {
    location /basic_status {
        stub_status;
    }
}
```

```
$ curl http://www.example.com/basic_status
Active connections: 1
server accepts handled requests
 7 7 7
Reading: 0 Writing: 1 Waiting: 0
```

- Provides aggregated NGINX statistics
- Restrict access so it’s not publicly visible

NGINX Plus Extended Status Module

```
$ curl https://www.nginx.com/resource/conf/status.conf
> /etc/nginx/conf.d/status.conf
```

```
upstream my_upstream {
    zone my_upstream 64k;
    server server1.example.com;
}
server {
    status_zone my_virtual_server;
    location / {
        proxy_set_header Host $host;
        proxy_pass http://my_upstream;
    }
}
```

- Provides detailed NGINX Plus statistics
- 40+ additional metrics
- Monitoring GUI also available; see **demo.nginx.com**
- Exclusive to NGINX Plus

Key Logging Files and Directories

- **/var/log/nginx/access.log** − Details about requests and responses

- **/var/log/nginx/error.log** − Details about NGINX errors

NGINX Access Logs

```
192.168.179.1 - - [15/May/2017:16:36:25 -0700] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36" "-"
192.168.179.1 - - [15/May/2017:16:36:26 -0700] "GET /favicon.ico HTTP/1.1" 404 571
"http://fmemon-redhat.local/" “Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36" "-"
192.168.179.1 - - [15/May/2017:16:36:31 -0700] "GET /basic_status HTTP/1.1" 200 100 "-"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36" "-"
```

- Enabled by default. Can be disabled with the `access_log off` directive.
- By default lists client IP address, date, request , referrer, user agent, etc. Can add additional NGINX variables; see **nginx.org/en/docs/varindex.html**.
- Log format configurable with the `log_format` directive

Summary

- We recommend using the NGINX mainline branch for most deployments
- Put all configuration in separate files in **/etc/nginx/conf.d/**
- Forcing all traffic to SSL improves security and improves search rankings
- Keepalive connections improve performance by reusing TCP connections
- SSL session caching and HTTP/2 improve SSL performance
- NGINX status module and logging capability provide visibility

Try NGINX Plus for free at **nginx.com/free-trial-request**

Upcoming Webinars

- Delivering High Performance Websites with NGINX (June 7, 2017, 11:00 AM CEST)
- Ask Me Anything about Microservices, Part 3 (June 14, 2017, 10:00 AM PDT)

Register at **nginx.com/webinars**