Contents

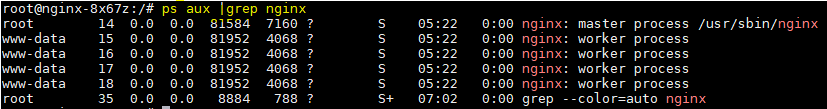
1. **What is Nginx --------------------------------------------------------------------------- 02**
   1. **How to install nginx on ubuntu ----------------------------------------------- 02**
   2. **Some useful default location -------------------------------------------------- 03**
   3. **How to test nginx config and reload ----------------------------------------- 03**
2. **What is reverse proxy ----------------------------------------------------------------- 03**
3. **Configuration File’s Structure ------------------------------------------------------- 04**
4. **Some important diective/context -------------------------------------------------- 05**
   1. **proxy\_pass --------------------------------------------------------------------------- 05**
   2. **main ------------------------------------------------------------------------------------ 05**
   3. **events ---------------------------------------------------------------------------------- 06**
   4. **http ------------------------------------------------------------------------------------- 06**
   5. **server ---------------------------------------------------------------------------------- 06**
   6. **location -------------------------------------------------------------------------------- 07**
   7. **upstream ------------------------------------------------------------------------------ 08**
   8. **mail ------------------------------------------------------------------------------------- 08**
5. **Steps to add tools to nginx for reverse proxy ------------------------------------ 09**
6. **What is Nginx:**

Nginx is an open source web server which can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache. It is much faster and lighter web server than apache.

Nginx is one of the very few systems that can patched or upgraded without having to take any downtime.

Nginx has one master process and several worker processes. The main purpose of the master process is to read and evaluate configuration and maintain worker processes. Worker processes do actual processing of requests.

**Command to check process:** **ps aux |grep nginx**



**1.1 How to Install nginx server on ubuntu:**

sudo apt-get update

sudo apt-get install nginx

To Check Nginx version: nginx –v

Output: nginx version: nginx/1.9.0

* Official docker image: docker pull nginx
* By default, Nginx HTTP server listens for incoming connection and binds on **port 80**.
* By default, Nginx listens for secure connections on **port 443**.

**1.2 Some Useful Default Location,**

|  |  |  |
| --- | --- | --- |
| **File Name** | **Location** | **Comments** |
| All nginx file/folder | /etc/nginx | The nginx configuration folders, as all the configuration files are located at this place. |
| Main config file | /etc/nginx/nginx.conf | This configuration file is the main files where we can change the global configuration of the Nginx. |
| Custom config file | /etc/nginx/sites-enabled/tools-context.conf | Here we update the configuration for new tool. |
| Access log file | /var/log/nginx/access.log | All client requests to the server are record in the access log |
| Error log file | /var/log/nginx/error.log | Any Nginx Web Server errors are recorded in the error log file |

**1.3 To Test nginx config file is ok or not**

* Command: **nginx -t**

**To Stop or Reload Configuration**

* Command: **nginx -s signal** (Where signal may be stop, quit, reload)
* In our project reload nginx with script **./reloadNginx.sh** in /home/ubuntu/wrappers directory.

1. **What is Reverse Proxy:**

A proxy server is a go‑between or intermediary server that forwards requests for content from multiple clients to different servers across the Internet. A reverse proxy server is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate backend server. A reverse proxy provides an additional level of abstraction and control to ensure the smooth flow of network traffic between clients and servers.

**Jenkins**

**Nginx**

**Client**

**LDAP**

**SonarQube**

1. **Configuration File’s structure: -** Nginx consists of modules which are controlled by directives specified in the configuration file. Directives are divided into simple directives and block directives (Context). Default config file looks like-



**Directive**: - A simple directive consists of the name and parameters separated by spaces and ends with a semicolon (;).

Like: listen 80;

**Context**: - A block directive has the same structure as a simple directive, but instead of the semicolon it ends with a set of additional instructions surrounded by braces ({ and }). If a block directive can have other directives inside braces, it is called a context (examples: [events](http://nginx.org/en/docs/ngx_core_module.html#events), [http](http://nginx.org/en/docs/http/ngx_http_core_module.html#http), [server](http://nginx.org/en/docs/http/ngx_http_core_module.html#server), and [location](http://nginx.org/en/docs/http/ngx_http_core_module.html#location)). Like

location / {

proxy\_pass http://localhost:8080/;

}

1. **Some important Directive/Context: -**

**4.1 proxy\_pass directive:**

This directive sets the address of the proxied server and the URI to which location will be mapped. When we say nginx to proxy\_pass that means, we are saying Pass this request on to this proxy URL.

Proxy\_pass http://jenkins-blueocean.ethan.svc.cluster.local:8080;

**4.2 main context:**

it is used to set the settings for NGINX globally and is the only context that is not surrounded by curly braces. Generally, the main context is placed at the beginning of the core nginx configuration file. The directives for the main context cannot be inherited in any other context and therefore cannot be overridden.

user nginx;

worker\_processes auto; # no. of worker process will be equal to no. of CPU

pid /run/nginx.pid;

**Note**: worker\_process multiply by worker\_connection will be equal to maximum no. of concurrent request our service should be able to accept

**4.3 events context:**

it is used to set global options for connection processing. The events context is included in the main context and there can be only one such context. The directives that we can place in this context are setting connection processing methods, the number of connections along with a few other directives.

# main context

events {

worker\_connections 768;

}

**4.4 http context:**

it declares settings for HTTP protocol. Virtual host settings are declared in the server context and are contained in the **http context.** Locations contexts that are used to store settings for URLs are contained within a server context. The HTTP context holds directives for handling HTTP and HTTPS traffic. The HTTP context is the child of the main context and is absolutely needed in every server context where virtual hosts settings are declared for our domains. Changes we want nginx to apply universally are in http block.

http {

sendfile on; # Skip buffering for static files

tcp\_nopush on; # Optimize sendfile packets

keepalive\_timeout 65; # Max time to keep a connection open

send\_timeout 10; # Max time for the client to accept/receive a response

...

}

**4.5 Server context:**

The NGINX virtual host settings are defined in the server contexts which in turn are contained in the HTTP context. There can be multiple server contexts inside the HTTP context. The directives inside server context govern the processing of requests for resources associated with a particular domain or IP address.

server {

listen 80;

server\_name domain1.com;

root /var/www/html/wordpress;

}

**4.6** [Location](https://www.journaldev.com/26342/nginx-location-directive)**contexts**:

it is used to define directives to handle client request. When a request for resource arrives at NGINX, it will try to match the URI to one of the locations and handle it accordingly.

Multiple Location contexts can be defined inside server blocks. Moreover, a location context can also be nested inside another location context.

The modifier in the location block is optional. Having a modifier in the location block will allow NGINX to treat a URL differently. Few most common modifiers are:

**Location** **Syntax**: location [ = | ~ | ~\* | ˆ~ ] uri { . . . }

* **none**: If no modifiers are present in a location block then the requested URI will be matched against the beginning of the requested URI.
* **=**: The equal sign is used to match a location block exactly against a requested URI.
* **~**: The tilde sign is used for case-sensitive regular expression match against a requested URI.
* **~\***: The tilde followed by asterisk sign is used for case insensitive regular expression match against a requested URI.
* **^~**: The carat followed by tilde sign is used to perform longest nonregular expression match against the requested URI. If the requested URI hits such a location block, no further matching will take place.

location / {

proxy\_pass http://cloveplatformui.ethan.svc.cluster.local:6000;

}

location ^~ /jenkinscore {

proxy\_pass http://jenkins-blueocean.ethan.svc.cluster.local:8081;

}

**4.7 upstream context**:

It is allowed to define a pool of back-end servers that NGINX can proxy the request. The context enables NGINX to perform load balancing while proxying the request. The upstream context is defined inside the HTTP context and outside any server context.

Once upstream servers have been defined, the name of the same is available within the server context to pass the request to the pool of back-end servers.

set $upstream\_clove http://cloveplatformui.ethan.svc.cluster.local:6000;

location / {

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header Host $host;

proxy\_redirect off;

proxy\_pass $upstream\_clove;

}

**4.8 mail context:**

NGINX can proxy the IMAP, POP3, and SMTP protocols to one of the upstream mail servers or to an external mail service where the mail accounts are hosted. To accomplish it, NGINX uses mail context that allows configuration directives for all aspects of proxying mail connections.

The mail context is defined in the main context i.e at the same level as the HTTP context.

mail {

server\_name mail.example.com;

auth\_http localhost:9000/cgi-bin/nginxauth.cgi;

proxy\_pass\_error\_message on;

...

}

1. **Steps to Add tools to Nginx for reverse proxy:**

* Write deployment file and Create pods of the application
* Write Service file and Create ClusterIP service for the application
* Configure/update reverse proxy setting in **tools-context.conf** file in /home/ubuntu/wrappers directory.
* Get ConfigMap (kubectl get cm)
* Delete nginx ConfigMap (kubectl delete cm nginx-conf)
* Create nginx ConfigMap from tools-context.conf file (kubectl create cm nginx-conf --from-file=tools-context.conf)
* Scale Down nginx pod (kubectl scale rs/nginx –replicaset=0)
* Scale Up nginx pod (kubectl scale rs/nginx –replicaset=1)
* Reload nginx (./reloadNginx.sh)