#_ Essential Nginx Commands / Concepts [+100]

1. Basic Nginx Operations

- nginx: Starts the NGINX server.
- nginx -s stop: Stops the NGINX server immediately.
- nginx -s quit: Shuts down the NGINX server gracefully.
- nginx -s reload: Reloads the NGINX configuration.
- nginx -t: Tests the NGINX configuration.
- nginx -v: Displays the NGINX version.
- nginx -V: Displays the NGINX version, compiler version, and configured modules.

2. Understanding NGINX Configuration

- **The main context:** Refers to the top level of the configuration.
- **Events context:** Contains directives that define operating system specific options.
- **HTTP context:** Contains directives for handling web traffic.
- **Server context:** Specifies configurations for a specific virtual server.
- **Location context:** Contains directives for processing specific types of requests.

3. Directives

- worker_processes: Defines the number of worker processes.
- worker_connections: Defines the maximum number of simultaneous connections for each worker process.
- sendfile: Enables or disables the use of sendfile().
- tcp_nopush: Enables or disables the use of the TCP_CORK socket option.
- keepalive_timeout: Sets the timeout for keep-alive connections with the client.
- include: Includes another file, or files matching the specified mask.
- default_type: Defines the default MIME type of a response.
- gzip: Enables or disables compression of responses.

4. Serving Static Content

- root: Sets the root directory for requests.
- index: Sets the default file to serve when a directory is requested.
- autoindex: Turns on or off the autoindex module, which automatically generates directory listings.

5. Reverse Proxy and Load Balancing

- proxy_pass: Sets the address of a proxied server and passes the request to the proxied server.
- proxy_set_header: Allows redefining or appending fields to the request header passed to the proxied server.
- upstream: Defines a group of servers for proxying or load balancing.

6. Load Balancing Methods

- **Round Robin:** The default method, requests are distributed evenly across the servers.
- **Least Connections:** A request is sent to the server with the least number of active connections.
- **IP Hash:** The client's IP address is used in the hash function to determine what server should be selected for the next request.

7. HTTPS and SSL/TLS

- ssl_certificate: Specifies the location of the SSL certificate to use for establishing secure connections.
- ssl_certificate_key: Specifies the location of the SSL certificate key.
- ssl_protocols: Enables the specified protocols for SSL/TLS.
- ssl_ciphers: Specifies the cipher list for SSL/TLS.
- ssl_prefer_server_ciphers: Specifies that server ciphers should be preferred over client ciphers when using the SSL/TLS protocols.

8. Caching

- proxy_cache_path: Defines the path and other parameters of a cache.
- proxy_cache: Sets the shared memory zone used for caching.
- proxy_cache_valid: Sets caching time for different response codes.
- add_header: Adds a field to a response header provided on the condition that the response code equals 200, 201 (1.3.10), 204, 206, 301, 302, 303, 304, 307 (1.1.16, 1.0.13), or 308 (1.13.0).

9. Rate Limiting

- limit_req_zone: Sets parameters for a request rate limit defined by a key.
- limit_req: Limits the request processing rate for a given location.
- limit_conn_zone: Sets parameters for a connection limit defined by a key.
- limit_conn: Limits the maximum allowed number of connections for a given key.

10. Server and Location Blocks

- Configuring server blocks: Server blocks are similar to virtual hosts in Apache. They allow you to configure NGINX to serve multiple domains out of a single server.
- Location blocks: The location context is used to decide how to process a request based on its URI.

11. Logging and Monitoring

- access_log: Defines the access log's location and format.
- error_log: Defines the error log's location and logging level.
- log_format: Defines the format of the access log.

12. Nginx Modules

- **Core Module:** Provides directives for configuring basic functionality and resources.
- **Events Module:** Provides directives for setting up base event handling attributes.
- **HTTP Module:** Provides directives for handling web traffic.
- **Mail Module:** Provides directives for handling mail traffic.
- **Stream Module:** Provides directives for handling TCP and UDP traffic.

13. Security

- server_tokens: Enables or disables emitting nginx version on error pages and in the "Server" response header field.
- add_header X-Frame-Options: Protects your website against clickjacking attacks.
- add_header X-Content-Type-Options: Stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type.
- add_header X-XSS-Protection: Enables cross-site scripting filter built into most recent web browsers.

14. Nginx Plus

- **Advanced load balancing:** Load balance with session persistence, health checks, and DNS SRV records.
- **Media streaming:** Improved live and on-demand streaming to multiple devices.
- **Monitoring and diagnostics:** Additional metrics, plus a live activity monitoring interface.

15. HTTP/2

- http2: Enables HTTP/2 for a server.
- Server Push: An HTTP/2 feature where the server sends resources to the client before the client requests them.

16. Nginx Ingress Controller

- **Basic concept:** A Kubernetes Ingress Controller that uses ConfigMaps to store the NGINX configuration.
- **Annotations:** Used to customize behavior.
- Custom templates: Used to customize the NGINX configuration.

17. Configuration Optimization

- **Tuning worker processes and worker connections:** Optimizing these can help handle more simultaneous clients.
- **Buffer and timeout optimization:** Optimizing these can help handle large files or slow clients.

18. Regular Expressions

- **Basic Regular Expressions:** Used in location matching and rewrite rules.
- **Regular Expression Modifiers:** Used to change the behavior of regular expressions.

19. Rewrite Rules

- rewrite: Generates an internal rewrite of the request.
- return: Stops processing and returns the specified code to a client.

20. Nginx Variables

- Predefined variables: Variables like $host, $uri, $args, etc., that can be used in configuration.
- set: Allows creation of custom variables.

21. Nginx Maps

- map: Helps to set variable's value based on another variable's value.

22. GeoIP Module

- geoip_country: Enables a country database.
- geoip_city: Enables a city database.

23. Load Balancing Algorithms

- **Weighted Load Balancing:** Assigns weight to backend servers for traffic distribution.
- **Least Connections Load Balancing:** Requests are distributed to the server with the fewest connections.
- **IP Hash Load Balancing:** The client's IP address is used to determine the backend server.

24. Advanced Proxying and Caching

- proxy_cache_bypass: Defines conditions under which the response will not be taken from a cache.
- proxy_no_cache: Defines conditions under which the response will not be saved to a cache.

25. Failover and Backup

- backup: Marks the server as a backup server.
- down: Marks the server as permanently unavailable.

26. Nginx command-line interface (CLI)

- **Master and Worker Processes:** Understanding NGINX architecture.
- **Signal a Master Process:** Send signals to NGINX processes.

27. Error Handling

- error_page: Configures responses to various error codes.
- Custom Error Pages: Create custom error pages.

28. Gzip Compression

- gzip: Enables or disables compression of responses.
- gzip_comp_level: Sets a gzip compression level of a response.

29. Nginx and PHP-FPM

- **PHP processing:** Configuration for processing PHP files with PHP-FPM.
- **FastCGI parameters:** Configure these to handle PHP request and response.

30. Third-Party Modules

- **Google PageSpeed:** Optimizes your site automatically by reducing the size of images, minifying CSS and JavaScript, and applying other speed enhancements.
- **Lua module:** Embed the power of Lua into Nginx HTTP Servers.
- **Brotli module:** Provides Brotli compression for NGINX.

31. WebSocket proxying

- **WebSocket configuration:** Configuration for proxying WebSocket connections.

32. Nginx and SSL/TLS

- **OCSP Stapling:** Allows the server to check if a SSL certificate has been revoked.
- **HSTS (HTTP Strict Transport Security):** Ensures the browser never visits the HTTP version of a website.
- **SSL Session Cache and Session Tickets:** Optimizing SSL with session cache and session tickets.
- **DH (Diffie-Hellman) key exchange:** Protecting against attacks on SSL.

33. Rate Limiting and IP Blacklisting

- IP Whitelisting and Blacklisting: Limit access to your server by IP.
- deny: Denies access for the specified IP address or addresses.
- allow: Allows access for the specified IP address or addresses.
- limit_req: Limits the request processing rate.

34. HTTP/3 and QUIC

- **Understanding HTTP/3:** An overview of the HTTP/3 and QUIC protocols.
- **Configuring HTTP/3:** How to configure HTTP/3 and QUIC with NGINX.

This should give you a comprehensive understanding of Nginx, serving as a useful reference for both beginners and experienced users.