EXAM BLUEPRINT

NGINX OSS

ABOUT THE NGINX CERTIFICATION EXAMS

The NGINX certification is earned by passing four exams, in any order:

Management
Configuration: Knowledge
Configuration: Demonstrate
Troubleshooting

Successful completion of the NGINX exam series acknowledges the skills and understanding necessary for day-to-day management of the NGINX web server platform.

WHAT IS THE NGINX EXAM BLUEPRINT?

The F5 NGINX Certification exam blueprint provides a list of objectives and a detailed breakdown of the skills and knowledge a candidate should have to pass each of the four exams and earn the NGINX certification credential. The objectives and examples can be used to identify areas for additional study, experience and knowledge. The examples provided are illustrative, not exhaustive.

PREREQUISITE:

None.

THIS EXAM IS BASED ON NGINX OSS BASIC

EXAM: Management		
Objectives and examples		CC*
1.1	**Given a scenario identify when to use NGINX** • Describe NGINX as a web server • Describe NGINX as a reverse proxy • Describe NGINX as a load balancer • Describe NGINX as a caching solution • Describe NGINX as an API gateway	**U/A**
1.2	**Explain the NGINX configuration directory structure** • Identify the default NGINX core config file • Identify the included directories/files • Describe the order of how the included files will be 'merged' into the running configuration • Describe directive inheritance and overriding properties	**U/A**
1.3	**Demonstrate how to manage user permissions** • Identify user context (i.e. using the configuration file) • Describe how and when to give read/write/execute access • Describe how to run NGINX as a specific user type • Describe the relationship between NGINX processes and users	**R**
1.4	**Manage shared memory zones** • Describe how and why NGINX uses shared memory zones • Describe why directives use a shared memory zone	**U/A**

EXAM: Configuration: Knowledge		
Objectives and examples		CC*
2.1	**Configure NGINX as a load balancer** • Define the load balancing pools/systems • Explain the different load balancing algorithms • Describe the process used to remove a server from the pool • Describe what happens when a pool server goes down • Explain what is unique to NGINX as a load balancer • Describe how to configure security • Modify or tune a memory zone configuration • Describe how to configure NGINX as mirroring server • Describe how to configure NGINX as a layer 4 load balancer • Describe how to configure NGINX as an API Gateway	**U/A**

* Cognitive Complexity Key: **R** = Remember, **A/E** = Analyze/Evaluate, **U/A** = Understand/Apply

2.2	**Configure NGINX as a content cache server** • Define a minimum retention policy • Describe how to configure path regex routing • Describe the why and how of caching in NGINX • Define the cache in the http context • Enable the cache • Specify the content that should be cached • Describe different types of caching • Explain what is unique to NGINX as a cache server	**U/A**
2.3	**Configure NGINX as a web server** • Demonstrate how to securely serve content (HTTP/HTTPS) • Describe the difference between serving static content and dynamic content. (REGEX, and variables) • Describe how server and location work • Explain what is unique to NGINX as a web server	**U/A**
2.4	**Configure NGINX as a reverse proxy** • Explain how traffic routing is handled in NGINX as a reverse proxy • Explain what is unique to NGINX as a reverse proxy • Configure encryption • Demonstrate how to manipulate headers • Describe the difference between proxy_set_Header and add_header • Modify or tune a memory zone configuration • Describe how to configure NGINX as socket reserve proxy • Describe how open source NGINX handles health checks in different situations	**U/A**

EXAM: Configuration: Demonstrate		
Objectives and Examples		**CC***
3.1	**Demonstrate how to manage connections and bandwidth** • Describe the difference between rate limiting and bandwidth throttling • Describe security • Demonstrate how to limit the amount of connections that are made to the NGINX server and its upstreams • Demonstrate how to set a bandwidth limit • Understand how to enable and optimize keep-alives for the NGINX server and its upstreams	**U/A**
3.2	**Demonstrate how to restrict access** • Demonstrate how to restrict access to NGINX based on IP address • Demonstrate how to restrict access to NGINX based on HTTP method • Demonstrate how to authenticate (auth basic / auth request) • Demonstrate how to restrict URIs	**U/A**

* Cognitive Complexity Key: **R** = Remember, **A/E** = Analyze/Evaluate, **U/A** = Understand/Apply

3.3	**Demonstrate how to configure logging** • Demonstrate how to customize the format of log files • Demonstrate how to customize the location of log files • Demonstrate how to set log levels (severity) • Describe the difference between an error log and an access log	**U/A**
3.4	**Demonstrate how to configure certificates** • Define the difference between a server certificate and a client certificate • Describe the components necessary to use an SSL certificate • Describe how to protect the SSL certificate and key	**U/A**
3.5	**Demonstrate how to enable HTTPS and associated security settings** • Compare the advantages of TLS termination, end to end encryption, and TLS passthrough • Demonstrate how to enable TLS encryption • Enable/Disable ciphers and TLS version • Describe how force all traffic to redirect to HTTPS	**U/A**

EXAM: Troubleshooting

Objectives and Examples		CC*
4.1	**Demonstrate how to stop, start, and reload NGINX binary** • Describe how to send signals to the NGINX process • Describe the difference between a reload and a stop/start • Describe how to test a new configuration before applying it	**U/A**
4.2	**Troubleshoot basic use cases** • Interpret logs • Identify start up failures • Describe how to deal with HTTP error codes • Describe how to troubleshoot various response • Describe how to troubleshoot use cases with multiple virtual hosts, multiple ports, and default servers • Describe how to troubleshoot location precedence and add_header inheritance • Describe how to troubleshoot client and server connections • Describe basic SELinux use cases	**U/A**
4.3	**Troubleshoot TLS security settings** • Identify TLS connection errors • Describe how to troubleshoot invalid certificates	**U/A**

* Cognitive Complexity Key: **R** = Remember, **A/E** = Analyze/Evaluate, **U/A** = Understand/Apply

Cognitive Complexity Descriptions

Lower Order Thinking Skills → Higher Order Thinking Skills

Remember	**Understand/Apply**	**Analyze/Evaluate**	**Create**
Information retrieval	Knowledge transfer	Critical thinking and reasoning	Innovation or creative thinking
Rote memorization	Comprehension or ability to apply knowledge to a standard process	Determine how parts relate to whole or knowledge integration and application to new situations	Forming an original work product
Retrieve relevant knowledge from long-term memory	Construct meaning from information	Make judgments based on criteria	Combine or reorganize parts to form a new pattern or structure
E.g., recall, retrieve, recognize	E.g., interpret, classify, compare, explain, implement	E.g., troubleshoot, attribute, diagnose, critique	E.g., generate, plan, produce

Alpine Testing Solutions' suggested cognitive complexity levels and associated verb references consider multiple approaches to defining cognitive processing (e.g., Anderson et al., Webb, Bloom, Frisbie). Above material created with assistance from Alpine and distributed with Alpine's permission as an attachment to certification test

Alpine Testing Solutions, Inc. (Alpine) gives F5 Networks permission to distribute the PDF "Cognitive Complexity Description 20130418.pdf" as an attachment to certification test blueprints created with assistance from Alpine into the exam blueprint.