WHITEPAPER

Load Testing Cerberus FTP Server 11 on the AWS and Azure platforms.

Phone
(703) 372-9750

Website
CerberusFTP.com

CONTENTS

1.0 INTRODUCTION 3

2.0 BACKGROUND 3

3.0 DESCRIPTION OF TESTS 4

4.0 TESTING SYSTEM SPECIFICATIONS 5

4.1 Cerberus Server Specifications 5

4.2 Cloud Client Server Specifications 5

4.3 Laptop Client Specifications 6

5.0 TESTS SETUP 7

6.0 CONDUCTING THE TESTS 8

6.1 Test One 8

6.1 Test Two 8

7.0 RESULTS 9

7.1 Test One 9

7.2 Test Two 10

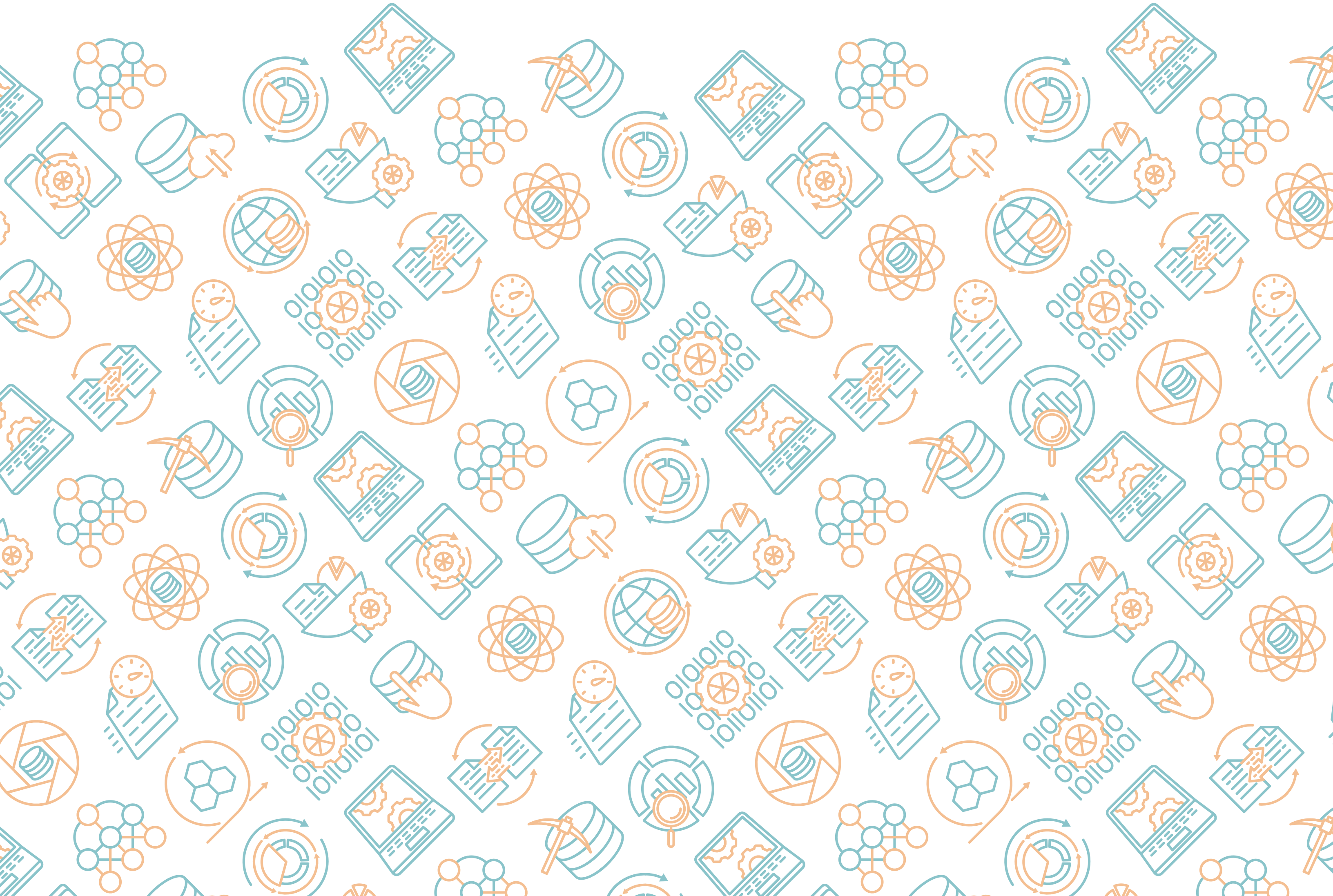
8.0 CONCLUSION 11

1.0 INTRODUCTION

Cerberus FTP Server is a robust, easy to manage secure file transfer server solution. Properly configured, and with sufficient network bandwidth, a single instance of Cerberus can handle large numbers of connections transferring large files. This document summarizes load testing conducted by Cerberus on virtual servers in the Amazon and Azure Clouds.

2.0 BACKGROUND

The purpose of this white paper is to document and benchmark Cerberus FTP Server's performance in a typical environment so that current and potential customers can be confident that Cerberus FTP Server will meet any connection and load requirements they may have.



3.0 DESCRIPTION OF TESTS

The testing was conducted with Cerberus running on virtual servers in both the Amazon and Azure cloud environments. The servers running Cerberus were configured in each cloud environment to be as similar as possible based on the specifications below:

Test 1 10 AWS client instances connecting to one AWS server in the same datacenter.

Test 2 10 AWS instances and 4 laptops connected to the office network or broadband wifi in dispersed locations connecting to one Azure server. Each instance or laptop ran an in-house developed application that rapidly executed randomized command-line client actions, such as uploading and downloading files of various sizes via FTP, FTPS and SFTP, deleting files and changing directories.

In order to focus our test on Cerberus FTP Server's performance as much as possible (as opposed to bandwidth, server performance, etc.), we made a deliberate decision to use a straightforward configuration.

Some example configurations that could cause bottlenecks that indirectly affect Cerberus performance include:

- AD/LDAP user management
- Reporting to a database
- Storing files on hard disk drives or network storage.

If customers are using these features and are concerned about performance, we recommend conducting load testing with those features enabled as performance is dependent on the setup and configuration of the environment within which Cerberus is operating.

4.0 TESTING SYSTEM SPECIFICATIONS

Specifications of the hardware and software used to run the test appear below.

4.1 Cerberus Server Specifications

AWS Cloud VM (1 instance)

- 4 vcpus, 2.3 GHz, Intel Broadwell E5-2686v4, x64 processor
- 16 GB RAM
- Windows Server 2019 Datacenter, 64-bit
- Storage: EBS Only (The local instance store volumes that are available to the instance. The data in an instance store is not permanent - it persists only during the lifetime of the instance.)
- Cerberus 11.1.0.0 with FTP, FTPS and SFTP listeners
- Network capacity up to 5 Gigabits/second

Azure Cloud VM (1 instance)

- AMD EPYC 7452 32-Core Processor with 1 Processor, 2 Cores (4 Threads), x64 processor
- 16 GB RAM
- Windows Server 2019 Datacenter, 64-bit
- Storage: 127 GB Premium SSD
- Cerberus 11.1.0.0 with FTP, FTPS and SFTP listeners
- Network capacity unlimited pay-as-you-go

4.2 Cloud Client Server Specifications

AWS Cloud VM (x10 instances)

- 1 vCPU, 2.5 GHz, Intel Xeon Family, x64 processor
- 1 GB RAM
- Windows Server 2019
- Network capacity up to 5 Gigabits/second
- In house developed Perl based application, initiating command-line cURL client actions

4.3 Laptop Client Specifications

Specific specifications varied, but, on average:

- 1 CPU, 1.9 GHz, Intel Core, x64 processor
- 8 GB RAM
- Windows 10
- Home Wifi connections of varying capabilities
- In house developed Perl based application, initiating command-line cURL client actions

5.0 TESTS SETUP

For the first test, Cerberus FTP Server version 11.1.0.0 was installed and configured on one AWS virtual server of moderate capability in order to reproduce the most common set up by the average Cerberus user.

For the second test, Cerberus FTP Server version 11.1.0.0 was installed and configured on one Azure virtual server of moderate capability in order to reproduce the most common set up by the average Cerberus user.

A custom application was developed by the Cerberus engineering team that uses Perl to rapidly generate and issue FTP, FTPS and SFTP commands to cURL to upload and download files of random sizes ranging from 10240 to 1048576 bytes in size via cURL, as well as transactions such as file deletes, status requests, and folder changes. The application was initiated via a command line containing the number of threads to initiate, from 1 to 64, and how many times to repeat the threads (effectively extending the time of the test). The application was installed on each client and laptop instance.

Each client instance and desktop used separate native Cerberus user accounts with password authentication. The installations of Cerberus on the AWS and Azure servers were configured with the default self-signed certificate provided by Cerberus at installation.

6.0 CONDUCTING THE TESTS

6.1 Test One

For test one, as a proof of concept of the test design, 10 AWS servers were combined to conduct the test against another AWS server in the same datacenter. This was to reduce the effect of potential network congestion on the test.

On all servers, a Windows Command line window was opened on each client and laptop instance. The command issues to the custom application initiated 40 threads repeated 50 times and were executed as closely to simultaneously as possible.

For test monitoring, the Cerberus Summary Page was used to monitor the number of uploads, downloads and concurrent connections. Task Manager was used to monitor CPU and Memory usage.

6.1 Test Two

For test two, we sought to reproduce real world conditions as much as possible. As such, several employee laptops and 10 AWS servers were combined to conduct the test to ensure that connections to the Cerberus Server were coming from several IP addresses dispersed across the United States.

The command issued to the custom application was agreed upon in advance among the team running the test. Each client and desktop initiated 36 threads repeated 100 times and were executed as closely to simultaneously as possible.

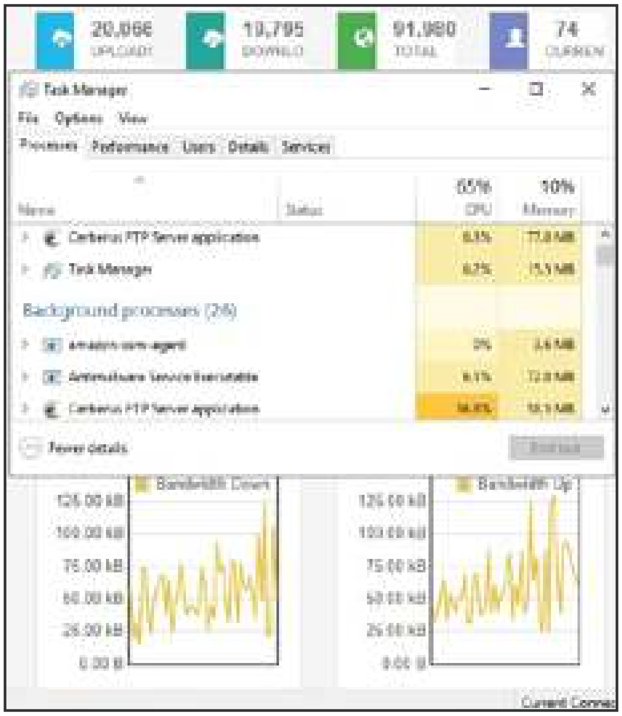
For test monitoring, the Cerberus Summary Page was used to monitor the number of uploads, downloads and concurrent connections. Task Manager was used to monitor CPU and Memory usage.

7.0 RESULTS

7.1 Test One

For test one, running 40 threads repeated 50 times by 10 AWS client servers against an AWS server running Cerberus in the same datacenter. This test was designed to re-create a 'medium-high' usage of Cerberus. We found the following:

- Concurrent connections peaked at about 74 and usually ranged between 50 - 70 throughout the test
- CPU usage of the AWS Cerberus server peaked at 65% but usually ranged between 50 - 55%. CPU usage on each of the client servers was at 100% throughout the test. This test scenario was possibly bottle-necked by the clients having only a single core.
- RAM usage on the AWS Cerberus server was low: it peaked at 77MB and was usually around 10% of total available.



Name	Status	65% CPU	10% Memory
Cerberus FTP Server application		0.3%	77.0 MB
Task Manager		0.2%	15.5 MB
Background processes (26)			
amazon-ssm-agent		0%	3.6 MB
Antimalware Service Executable		0.1%	72.0 MB
Cerberus FTP Server application		56.8%	50.1 MB

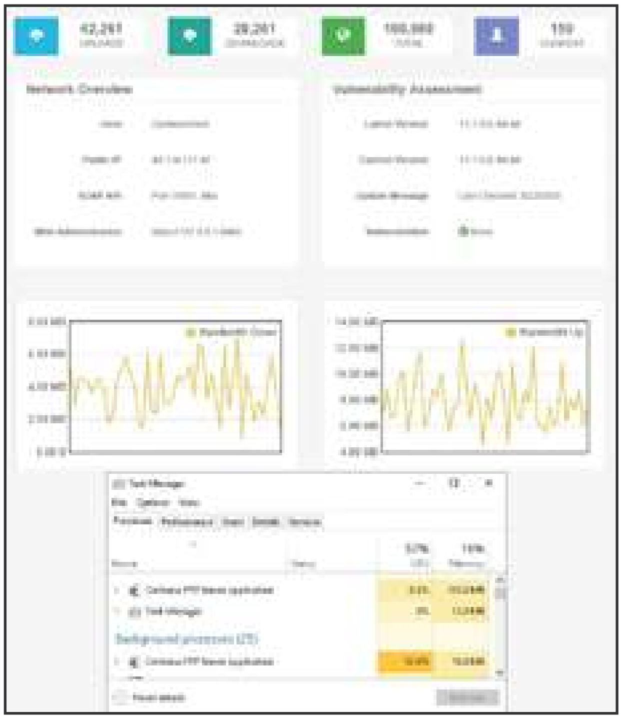
Test One (AWS): Screenshot of the Cerberus Uploads, Downloads, Total Connections and Current Connections

7.0 RESULTS
CONTINUED

7.2 Test Two

For test two, running 36 threads repeated 100 times by 10 AWS client servers and 4 laptops against an Azure server running Cerberus, from several IP addresses dispersed across the United States, we found the following:

- Concurrent connections peaked at 160+ and usually ranged between 100-150 throughout the test
- CPU usage of the AWS Cerberus server peaked at 62% but usually ranged between 50 - 55%. CPU usage on each of the client servers was at or close to 100% throughout the test
- RAM usage on the AWS Cerberus server was low: it peaked at 105MB and was usually around 12 - 15% of total available



Test Two (Azure): Screenshot of the Cerberus Uploads, Downloads, Total Connections and Current Connections

8.0 CONCLUSION

Cerberus is easily able to handle several hundred simultaneous connections without any delay in file transfer speed. Our tests were designed to simulate ‘medium-to-high’ usage of Cerberus, and indicate that the software possesses much higher transfer capacity than the typical customer hosting Cerberus on a standard 4-core server will ever use. For these customers, transfer limitations are much more likely to come from server hardware or bandwidth issues.

For extremely high-volume customers, we recommend load testing on more robust servers with as few limitations on the inbound network as possible in order to truly stress the product. If you would like to discuss further testing in this area, please contact us at support@cerberusftp.com

Both tests simulated what can be considered to be a ‘medium-high’ usage of Cerberus. Several hundred simultaneous connections is more than the vast majority of our customer base will ever handle. Despite this, Cerberus (and the AWS and Azure cloud servers hosting it) were able to rapidly process the transactions requested with no delays. We found that the capabilities of the client devices and networks is a much greater limiting factor. In order to really tax the capacities of a Cerberus instance being hosted on a standard 4 core server and simulate the highest volume customers we have, we would need to run the tests on much more capable client servers and ensure there are as few limitations on the inbound network as possible. Still, these basic tests have demonstrated that the Cerberus software itself is capable of handling a high volume of traffic with no issues.