EBOOK

testhouse

Azure Load Testing

www.testhouse.net

CONTENT INSIDE

INTRODUCTION

Web sites and mobile applications have emerged as significant avenues for conducting business. In today's business landscape, it is crucial for companies to cultivate a positive digital experience to effectively engage customers. By providing a positive digital experience, companies can increase customer satisfaction, loyalty, and engagement, and ultimately drive revenue and growth.

Performance of an application plays a critical role in providing consistent digital experience to customers. Any performance issues in an application will directly result in loss of revenue, harm to brand reputation and lead to dissatisfied customers. Hence, it’s essential to use performance tools that can measure digital experience of a website or mobile application. These tools help simulate user behavior under different load conditions and identify potential bottlenecks in the application’s performance. Identifying performance, scalability, or resiliency issues in production or near-production stages can pose significant challenges and incur substantial expenses when it comes to resolution.

This ebook will help guide you in generating Azure Load Tests without the need for expertise in load-testing tools, or seamlessly import your pre-existing Apache JMeter scripts. Attain valuable insights into performance, scalability, and capacity, and promote ongoing enhancements through automated continuous integration and continuous delivery (CI/CD) workflows.

BACKGROUND

For businesses, the performance of their applications holds the key to unlocking the full potential of their digital endeavors. A seamless and high-performing application can elevate user satisfaction and brand perception, translating into tangible business success. On the contrary, performance issues can lead to customer frustration, revenue loss, and reputational damage, derailing even the most well-crafted digital strategies.

To navigate this critical aspect of the digital landscape, organizations must embrace performance testing as a strategic imperative. Thoroughly evaluating applications under varying load conditions is essential to identify and eliminate performance bottlenecks, ensuring optimal user experiences even during peak usage periods.

At the forefront of empowering businesses with unrivaled performance testing capabilities stands "Azure Load Testing" - an innovative and fully managed service by Microsoft Azure. Tailored to the needs of modern enterprises, this service enables organizations to effortlessly generate large-scale loads and gain valuable insights into application performance.

In this ebook, we embark on a transformative journey, guiding organizations towards harnessing the potential of Azure Load Testing, regardless of their prior experience with load-testing tools. Our focus is on unraveling the key features, benefits, and best practices of Azure Load Testing, equipping organizations with the knowledge and tools to supercharge their application performance.

Our mission is to empower organizations to craft exceptional digital experiences that captivate their customers, inspire unwavering loyalty, and pave the way for enduring success. By embracing Azure Load Testing, businesses can confidently propel their digital initiatives to new heights, confidently venturing into the future with a customer-centric approach that sets them apart as true industry leaders.

PERFORMANCE TEST TYPES AND OBJECTIVES

Performance testing is a type of software testing that is used to evaluate the performance of an application or system under various load conditions. The goal of performance testing is to identify and measure key performance indicators such as response time, throughput, resource utilization, and stability of the system under various load conditions.

We can leverage Azure load testing service to conduct various types of tests. Following are the key test types & their objectives.

Test type	Key objective
Load Testing	Simulating realistic load conditions to evaluate how the system performs under different workloads.
Stress Testing	Simulating beyond its expected load capacity to identify the breaking point or number of users system can handle before it starts to degrade its performance.
Capacity Testing	Testing the system's ability to handle future growth by evaluating its ability to handle increasing levels of load over time.
Endurance or Soak Testing	Evaluate the system's ability to handle a sustained workload over an extended period.
Spike Testing	Evaluate system’s ability to handle sudden and extreme increases in workload or user activity.

KEY FEATURES AND BENEFITS OF AZURE LOAD TESTING

Azure Load testing service helps effortlessly generate large-scale load without the requirement of intricate infrastructure. Efficiently create tests without any prior familiarity with load testing tools or execute existing test scripts at scale with comprehensive support for Apache JMeter. Simplify load testing with a fully managed service that seamlessly integrates networking best practices, guaranteeing a smooth testing experience for both public and private endpoints hosted on Azure, multi-cloud, on-premises, or hybrid environments.
Here are some of the key features of Azure load test service.

Pay per use

Helps generate high scale load quickly and easily by generating load using virtual users. No additional set-up or configurations are required, Microsoft provides a built-in load testing service, available as subscription by paying for only what you use.

Secured

Highly secured platform, offers security and compliance features to safeguard confidentiality and integrity of test data. Also provides options for securing communication channels, encrypted data, and adherence to industry-standard compliance regulations.

Real-world conditions

Helps mimic real-world load conditions by distributing load across multiple geographical regions

Built-in best practices

Provides built-in networking best practices to avoid tests being mistaken for a security risk (man in the middle attack). Also provides recommendations to improve runtime reliability. Tests are stopped automatically if the application endpoints or Azure components start throttling requests.

Monitoring

Seamless integration with other Azure Services such as Application Insights and Azure monitors to collect additional telemetry data to gain deeper insights into application's performance during load tests.

CI/CD

Easily incorporate load tests into Azure DevOps workflows.

APPROACH – AZURE LOAD TESTING

The approach to Azure Load Testing typically involves following major steps.

1. Planning & Preparation

- Understand the application architecture.
- Define performance testing key goals and objectives – Identify key performance metrics you want to measure, such as response time, throughput, scalability, and resource utilization. These objectives will help guide testing approach.
- Identify scenarios - identify critical transactions and scenarios that need to be tested. This could include frequently performed transactions or activities by various users. It's recommended to identify realistic test scenarios that mimic real-world usage patterns.
- Identify workload profiles to simulate realistic load on the system.

2. Create Azure Load Testing Resource & Create Test

- First, you'll create the top-level resource for Azure Load Testing. It provides a centralized place to view and manage test plans, test results, and related artifacts.
- Once the resource is created, you create load tests. Following are two ways to create load tests.
    - Quick Test – Provide URL, specify number of virtual users, ramp-up time, and test duration.
    - Upload JMeter Scripts – Upload JMX (JMeter script), define environment variables if used in test. You can also provide number of test engines, pass / fail criteria and configure server monitoring.

3. Configure Server Monitoring Metrics

- To capture metrics during your load test, select the Azure components that make up your application. Azure Load Testing integrates with Azure Monitor to capture server-side resource metrics for Azure-hosted applications.
- Configure and Monitor server-side application metrics by using Azure Load Testing – Additional reference - https://learn.microsoft.com/en-us/azure/load-testing/how-to-monitor-server-side-metrics

Note: You will need an Azure Load Testing resource with at least one completed test run.

4. Execute & Analyse

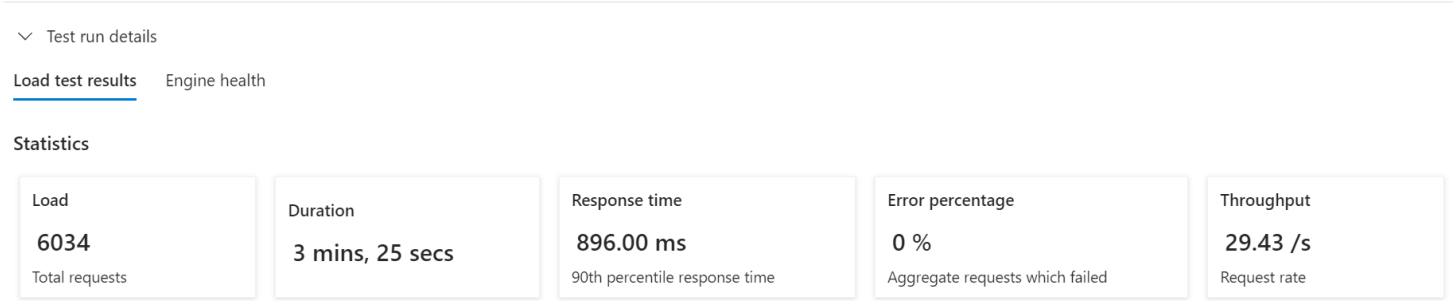
- Run performance tests using the defined test scripts and configurations.
- Monitor system health during the test capturing relevant performance metrics like response time, transaction throughput, error rates, and resource utilization

APPROACH – AZURE LOAD TESTING

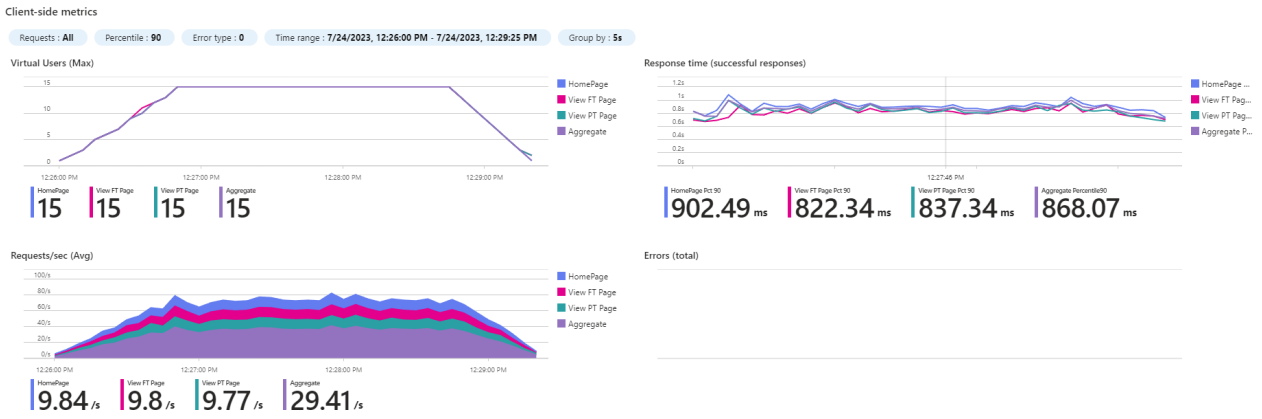
- Analyse the collected performance metrics to identify bottlenecks, performance issues, and areas for improvement. Compare the observed performance against defined benchmarks or service-level agreements (SLAs). Look for trends, anomalies, and areas of the system that may require optimization.
- Publish performance test reports.

We can collect following information during Azure Load Test Run:

Test run details – Displays key statistics such as total requests, duration of the test, errors encountered (if any), throughput / request rate.

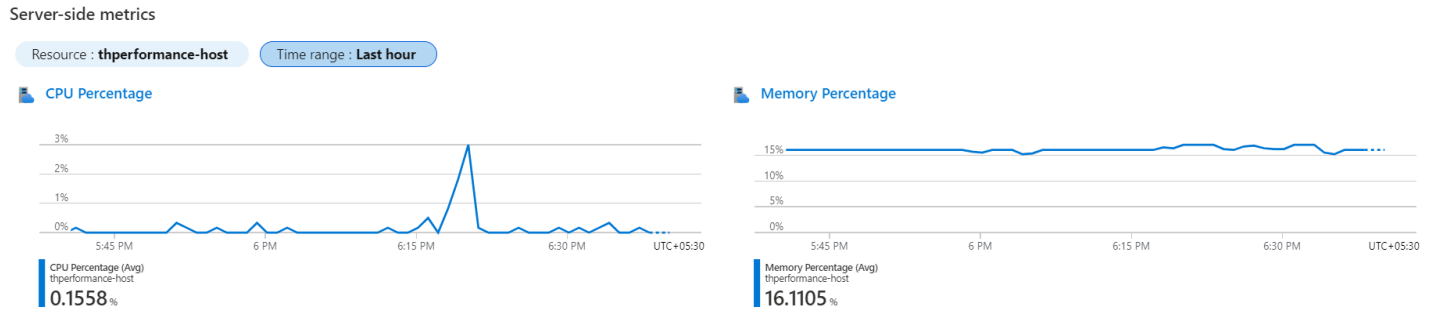


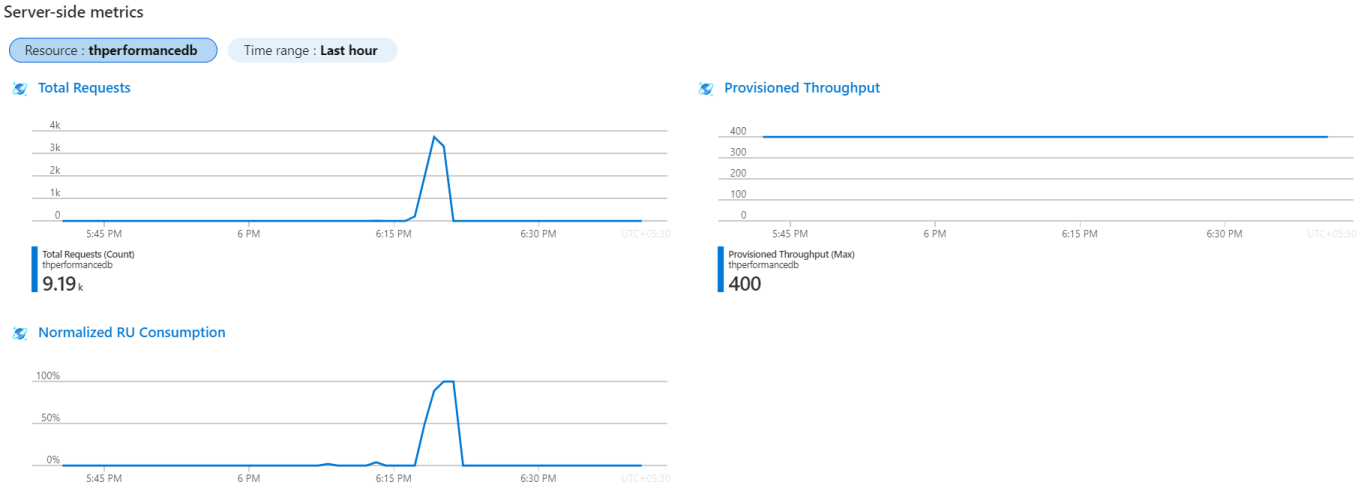
Client-side metrics – Displays number of virtual users, response time, request per second and any error encountered during execution.



APPROACH – AZURE LOAD TESTING

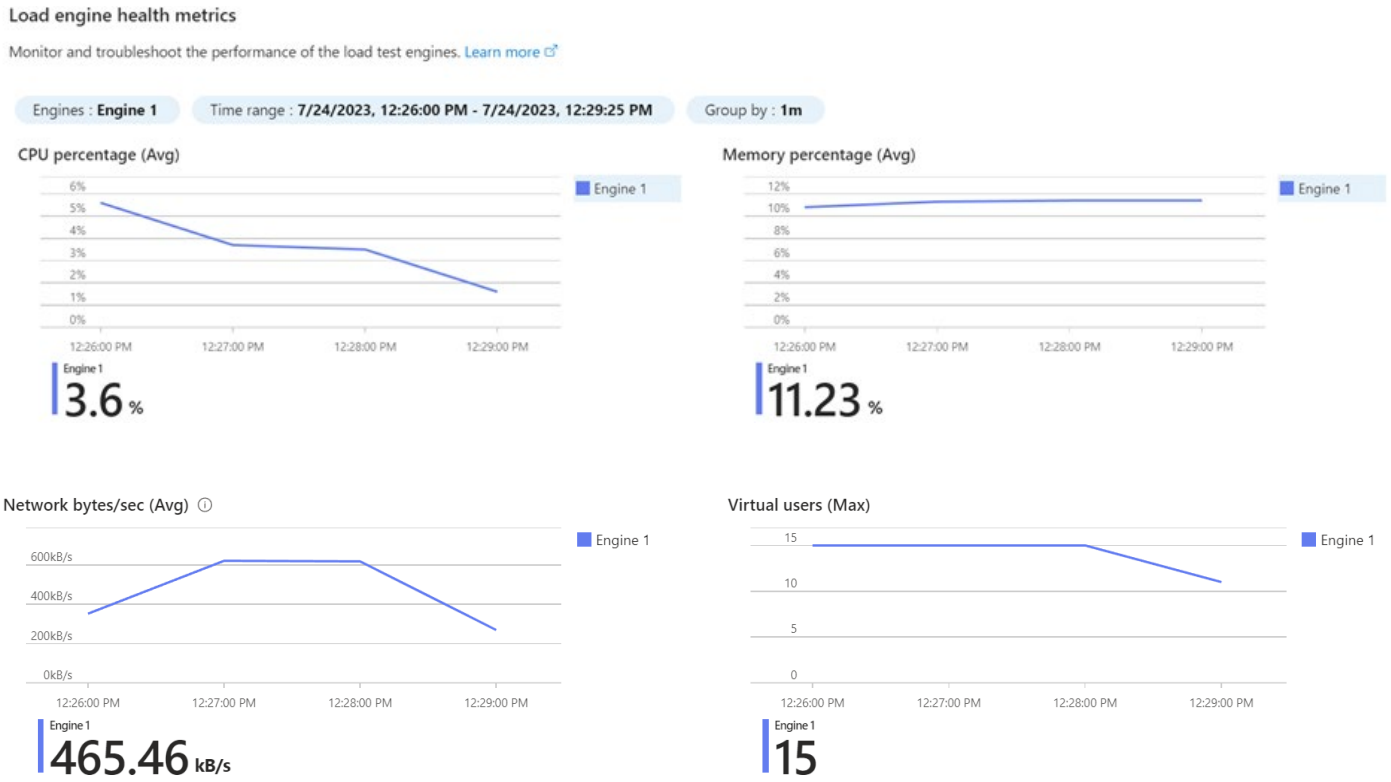
Server-side metrics – Displays information about Azure components (these needs to be configured prior running a test). Depending on the type of application components, you can monitor different metrics available out of the box. For example, metrics can be specific to Wed, App or DB servers. Following is an illustrative example of server-side metrics.





APPROACH – AZURE LOAD TESTING

Load engine health metrics – You can monitor and troubleshoot the health of test engine. This will help ensure the load test engine itself is not a bottleneck. Metrics captured include CPU, Memory & Network.



Note - You can Automate a load test with CI/CD pipeline. Refer below link for detailed steps.
https://learn.microsoft.com/en-us/azure/load-testing/quickstart-add-load-test-cicd?tabs=pipelines

5. Re-test post fixes or Iterate post enhancement

- Based on the analysis of the test results, optimize the application to address performance issues. This could involve tuning server configurations, optimizing database queries, improving caching mechanisms, or enhancing code efficiency. After implementing the optimizations, rerun the performance tests to validate the improvements.
- Use Azure Load test comparison feature to compare results of two or more runs.

BEST PRACTICES

When conducting Azure Load Testing, it's important to follow best practices to ensure accurate and meaningful results. Here are some best practices to consider.

I. *Number of users per Test engine* - Microsoft recommends 250 virtual users (threads) per engine instance. This varies from application to application depending on the complexity of test script and virtual user memory footprint.
II. Set engine instances accordingly to reach the desired number of threads. For example, set engine instances = 4 to reach a total of 1,000 threads.
III. In case your test plan uses a 'user.properties' file, please specify it using the File relevance dropdown of Test plan tab when creating a test.
IV. When you define the environment variable for the load test, its name must match the variable name that you used in the Apache JMeter script.
V. Azure Load Testing uploads the JMX file and all related files in a single folder. When you reference an external file in your JMeter script, verify that your only use the file name and remove any file path references.
VI. 6Azure Load Testing doesn't preserve the header row when splitting your CSV file. Before you add the CSV file to the load test, remove the header row from the file.

BEST PRACTICES

VII. Enable “Auto stop” test - Azure Load Testing will automatically stop a load test if the error percentage is high for a certain time window. This prevents incurring costs for an incorrectly configured test..

VIII. Monitor the health of Test Engine during test to ensure instances themselves are not a performance bottleneck

In addition, we recommend following best practices, which are specific to simulating realistic load on the system.

I. **Scenarios & workload** – Prioritise most business-critical scenarios and create end to end scenarios. Study usage pattern and behaviour of real users in production if available.

II. **Network and connectivity** - Test the application's performance under different network conditions, including high latency, low bandwidth, or intermittent connectivity. This helps identify any performance issues related to network communication, data synchronization. You can configure Azure Test engine from various geographically distributed locations that represent majority of customer base.

III. **Test data** - Use representative data sets that mimic the volume and complexity of real-world banking data. Consider a range of data sizes, such as different numbers of accounts, transaction histories, and user profiles. Performance testing with realistic data helps identify any performance issues related to data processing, retrieval, and storage.

IV. **Client-side performance** – Measure client-side performance during peak load conditions. For Internet banking application, measure page load time at UI layer, check for rendering time across various browsers. For Mobile banking application, check response time, CPU, battery and memory consumption using physical device during peak load condition.

CONCLUSION

In the digital age, where every interaction with customers is a make-or-break moment, delivering an exceptional user experience has become the cornerstone of business success. As web applications and mobile platforms take center stage in engaging customers, ensuring optimal performance has emerged as a critical imperative for organizations seeking to thrive in the highly competitive market.

The journey through the realms of "Azure Load Testing" has illuminated the path for organizations seeking to unlock the full potential of their applications. By embracing this innovative and fully managed service offered by Microsoft Azure, businesses can effortlessly generate large-scale loads, accurately assess application performance, and proactively address potential bottlenecks.
Azure Load Testing, with its wealth of features and seamless integration with Azure services, empowers organizations to optimize their digital experiences without the need for extensive load-testing expertise. From simulating real-world scenarios to capturing vital performance metrics, this service equips businesses with the tools necessary to elevate their application performance and deliver unmatched user satisfaction.
The best practices shared in this ebook serve as invaluable guides, paving the way for accurate and meaningful results. By adhering to these recommendations, organizations can fine-tune their testing endeavors, maximize the efficiency of their test engines, and confidently navigate the complexities of performance testing.

We urge organizations to embrace the spirit of innovation and customer-centricity embodied by Azure Load Testing. By harnessing the power of this cutting-edge service, businesses can build a resilient foundation for their digital initiatives, ensuring their applications thrive even under the most demanding conditions.
The quest for digital excellence is a journey that never ends, and Azure Load Testing provides organizations with a potent ally on this path. Let us embark on this journey together, driven by the vision of delivering extraordinary digital experiences, fostering unwavering customer loyalty, and transforming the future of business.

TESTHOUSE PERFORMANCE ENGINEERING SERVICES

At Testhouse, we take pride in our extensive experience in delivering large-scale performance engineering services for our clients. With the capability to deliver tests for up to 500,000 users within just 6 to 8 weeks, we are confident in our ability to optimize the performance and stability of your digital products. Our team consists of over 30 performance testing specialists who deliver services to customers across the globe. We have a wealth of experience in a variety of performance testing tools, including Jmeter, Gatling, HP LoadRunner, Neoload and more. One of our key strengths is our ability to apply shift-left testing practices around performance, which has helped us to reduce the cost of quality by identifying defects early in the life cycle.

testhouse

Let’s Connect

Whether you have a query or an enquiry, specific or general,
we are more than happy to help

Contact Us