



# High Availability & Performance. Effective Resource Usage. Low Cost.

---

Regarding the Performance Efficiency Pillar of AWS' Well-Architected Framework focused on Load and Performance Testing



# What's Inside

<b>Introduction</b>	<b>3</b>
What is the AWS Well-Architected Framework?	3
What is the Performance Efficiency Pillar?	3
What is StormForge – Load and Performance Testing?	3
<b>Five Design Principles for Performance Efficiency in the Cloud</b>	<b>5</b>
1. Democratize advanced technologies	5
Introducing and using the right technology is complex	5
Gain insights with load testing experiments	6
2. Go global in minutes	6
The need for speed – globally...	6
...without the burden of low latencies.	7
3. Use Serverless architectures	8
Get rid of your servers...	8
...and understand what's up next	8
4. Experiment more often	9
You're maybe stuck	9
Infrastructure-as-code will help and the future of testing is in production	9
5. Mechanical sympathy	10
Separation of concerns, decoupling and design patterns in general	10
Complexity has not gone away – it shifted to a managed service	10
<b>Thank You</b>	<b>10</b>
<b>About the Authors</b>	<b>11</b>



# Introduction

## What is the AWS Well-Architected Framework?

The AWS Well-Architected Framework “documents a set of foundational questions that allow you to understand if a specific architecture aligns well with cloud best practices”<sup>1</sup> in the form of a white paper. AWS published the first version of it in 2016. We [wrote a blog article](#) about it.

AWS reworked the guidelines in November 2018 and added five extra white papers covering each of the five pillars introduced in the Well-Architected Framework (Operational Excellence, Security, Reliability, Performance Efficiency, and Cost Optimization).

We will have a look at the white paper about the Performance Efficiency Pillar focusing on load and performance testing.

## What is the Performance Efficiency Pillar?

The Performance Efficiency Pillar deals with “the efficient use of computing resources to meet requirements and how to maintain that efficiency as demand changes and technologies evolve”<sup>2</sup>. You might have created a great architecture and built your infrastructure in a very reasoned way but you cannot rest upon your work. Technologies change and are refined and so has your system.

---

1 AWS Well-Architected Framework – November 2018, p. 1c, <https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>

2 Performance Efficiency Pillar. AWS Well-Architected Framework – July 2018, p. 1, <https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/welcome.html>

## What is StormForge Performance Testing?

StormForge Performance Testing is a cloud-based Software as a Service (SaaS) solution for continuous load and performance testing. As an AWS Advanced Technology Partner StormForge empowers agile and DevOps teams to create, manage and run large scale load and performance tests – automated, integrated and in real time.

StormForge takes advantage of AWS' cloud infrastructure to scale test clusters based on the customers' needs so they can explore the behavior of their API and web application, investigate potential issues and focus on reliability and scalability.

# Five Design Principles for Performance Efficiency in the Cloud

The Performance Efficiency Pillar of the AWS Well-Architected Framework suggests the following five design principles to “help you achieve performance efficiency”<sup>3</sup>:

- Democratize advanced technologies
- Go global in minutes
- Use Serverless architectures
- Experiment more often
- Mechanical sympathy

In the following, we will examine each one of these five design principles focusing on how load and performance testing can empower you to optimize your performance efficiency.

## 1. Democratize advanced technologies

*We are creating and dealing with complex systems!*

- Everybody today

### Introducing and using the right technology is complex

Using the right tool for the job (a.k.a. deciding for the right technology to solve a very particular problem in your business space) is critical and very complicated. To run that technology which is maybe new to your team is even more complex.

---

3 Performance Efficiency Pillar AWS Well-Architected Framework, p. 2

## Gain insights with load testing experiments

A major problem of each and every team member in a traditional or DevOps environment is to create a common understanding of the behavior of:

- single services,
- interacting services,
- and of course the whole distributed system.

Doing load testing experiments is a common practice to create that understanding. Even the simplest test scenario answers questions about responsiveness, runtime behavior, resource consumption, monitoring and logging insights as well as operational excellence.

**When conducting their first load test most StormForge customers realize that they have an incomplete, missing or even broken monitoring or logging setup. That realization leads to one of the baselines for democratization.**

[REQUEST A DEMO TO SEE FOR YOURSELF >](#)

## 2. Go global in minutes

*46 TIMES MORE frequent code deployments*

*2,555 TIMES FASTER lead time from commit to deploy*

*7 TIMES LOWER change failure rate*

*2,604 TIMES FASTER time to recover from incidents*

- DORA State of DevOps Report 2018

Comparing the elite group against the low performers

### The need for speed – globally...

DevOps has changed the world. Talking about speed we have to distinguish at least between two perspectives.

First, you need many, short release cycles to quickly gather feedback from the end user to quickly iterate over the product. In the process, you opt for one or the other technology that helps you to further increase speed-to-market and/or enhance the end-customer advantage. There is not only Moore's Law<sup>4</sup> behind that but also a lot of learnings in the Lean Enterprise<sup>5</sup>.

Second, the need for speed of delivery to a customer with great user experience is a major need for product success:

- If Search Engine Optimization (SEO) is important for your business:  
Loading pages must be fast<sup>6</sup>.
- If transactional conversion like for ecommerce is important to your business:  
It must be fast and scalable<sup>7</sup>.
- If connectivity and security like for IoT is important to your business:  
It must be scalable and reliable<sup>8</sup>.

Today, we need to take these perspectives into account – globally.

### **...without the burden of low latencies.**

Global setups have to deal with the burdens of latency, the interaction with important points-of-presence (PoP), the reliability of connection and of course with the availability and scalability in each and every location. AWS gives the ability to easily deploy a setup in every region around the world.

With StormForge and load testing, the impact of performance and latency of multi-region and global setups becomes inspectable. You gain insights *before* the user experience is affected.

**[LEARN MORE IN THIS BLOG ABOUT AWS FARGATE NETWORK PERFORMANCE >](#)**

---

4 Moore's Law, or how overall processing power for computers will double every two years, <http://www.moorelaw.org/>

5 Lean Enterprise. How High Performance Organizations Innovate at Scale, Barry O'Reilly, Joanne Molesky, Jez Humble

6 Analyze and optimize your website with PageSpeed tools, <https://developers.google.com/speed/>

7 Improve Your Ecommerce Site Performance & Speed to 2X Conversions, <https://www.shopify.com/enterprise/site-performance-page-speed-ecommerce>

8 AWS IoT Core, <https://www.amazonaws.cn/en/iot-core/>

### 3. Use Serverless architectures

*Serverless it is.*

*Serverless it will be.*

– Yan Cui, *AWS Serverless Hero*, developer advocate at Lumigo, author of *Production-Ready Serverless*

#### Get rid of your servers...

Creating Serverless Infrastructure allows you to decouple each and every business requirement in a single computable unit. Reconnecting many of these units using e.g. AWS Step Functions or queues like AWS SQS or AWS Kinesis gives you the ability to actually orchestrate whole complex business processes and workflows – decoupled at low cost and with the power of real independence.

#### ...and understand what's up next

With great power comes great responsibility. A Serverless Infrastructure is still a complex distributed system. The complexity just shifts from a monolithic setup to a decoupled managed setup. It's necessary to understand its behavior, have fallbacks and circuit breakers, understand retries and have deep insights through monitoring. Furthermore Serverless itself has its own – sometimes special – behavior.

**Did you know that you can reduce response times of AWS Lambdas in NodeJS up to 70% by only applying one configuration?**

[LEARN HOW IN THIS ARTICLE >](#)



## 4. Experiment more often

*We'll never have an adequate testing environment to do regular, automated functional, non-functional or exploratory testing.*

– Head of QA of some software enterprise

### You're maybe stuck

After years of software delivery and evolving infrastructure it's hard to move on and get away from the Innovator's Dilemma<sup>9</sup>. It's just not easy to get rid of technical debt and add new business value easily.

### Infrastructure-as-code will help and the future of testing is in production

The AWS Cloud provides an ecosystem for easily getting started with one single opportunity and move it out of the legacy as described in the previous section, "[Use Serverless architectures](#)". Did you ever think about the idea to furnish or rebuild a business critical component of a system by using Serverless technologies or whatever fits from the AWS Cloud ecosystem? Ask yourself: Will it be faster/better/cheaper, simpler to operate or probably give you the chance to start a new journey?

Load and performance testing enables you to answer these questions without any impact on your business. It helps to determine the impact of an innovative setup, its configuration or it just lets you test a new technology fast. Automation using infrastructure-as-code allows you to change components/services easily and to switch to another service/component without any effort.

The obvious idea of having a lot of different environments is maybe wrong. Testing functional and non-functional requirements will move closer to production soon.

**Do you know about configuration or availability & resiliency testing? Did you ever heard of the principles of chaos engineering and the future of testing in production?**

[LEARN MORE ABOUT THE TYPES OF PERFORMANCE TESTING >](#)

---

9 The Innovator's Dilemma, Clayton Christensen, [https://en.wikipedia.org/wiki/The\\_Innovator%27s\\_Dilemma](https://en.wikipedia.org/wiki/The_Innovator%27s_Dilemma)

## 5. Mechanical sympathy

*Complex systems are systems with high interdependence.*

– Physics

### Separation of concerns, decoupling and design patterns in general

Most of the known design patterns of software and architecture design do of course apply in to cloud. Some of them like the Data Access Pattern are highly necessary to actually be capable to enjoy the advantages of cloud-based and managed services.

### Complexity has not gone away – it shifted to a managed service

At some point it comes to a finding in the sphere of connecting to a data access layer and retrieving data when doing performance testing. With the flexibility of a cloud-based and managed service it is possible to adjust that data layer to fit your needs. But you have to pay attention and take care of design patterns.

**Respect design patterns, experiment more often, load test early and often!**

[READ MORE ABOUT PERFORMANCE TESTING PRE AND POST CLOUD >](#)

## Thank You

We hope you enjoyed this eBook. If you have any questions or comments please don't hesitate to contact us at [support@stormforge.io](mailto:support@stormforge.io) or contact one of the authors.

# About the Authors



## Lars Wolff

Director Performance Testing, [lars@stormforge.io](mailto:lars@stormforge.io)

Lars Wolff is co-founder of StormForger, a performance testing SaaS for agile and DevOps teams. In 2020, StormForger and the US company Carbon Relay merged to StormForge, combining load and performance testing with machine learning. Lars has been working in software development for web based systems as an engineer and agile coach for years. Today he mainly helps teams to deliver reliable, scalable and fast systems. He's in love with DevOps, Scrum, Kanban, Whiteboards and Post-its®.



## Denise Schynol

Marketing Manager, [denise@stormforge.io](mailto:denise@stormforge.io)

Denise Schynol is responsible for marketing and communications at StormForge Germany. With a background in community building and web development she takes care of everything that is available to our customers and prospects.

She is a Rails Girl, puts everything in lists to rule the chaos, and takes care of finding solutions as your personal MacGyver.

