Load & Performance
Testing the Cloud??!!

Sebastian Cohnen, Founder & CTO
StormForger.com
2016-04-12
EHLO!

• 7+ years consulting and development
• Focus on web performance and architectures
• Founder & CTO StormForger.com

Sebastian Cohnen (@tisba)
Performance
Performance

Performance is the ability of a system to fulfill a task within defined dimensions.

It is a measurement for efficiency and can describe the relative capacity of a system.

“1 instance can handle 250 rps with p99 at 50ms…”
Performance & Scalability
Performance & Scalability

Performance
…is a measurement for efficiency and can describe the relative capacity of a system.

“1 instance can handle 250 rps with p99 at 50ms…”

Scalability
…will tell you if and how effectively you can grow capacity by adding resources.

“10 instances can handle 2500 rps with p99 at 50ms…”
Load & Performance Testing
“In software engineering, **performance testing** is in general, a testing practice performed to **determine how a system performs** in terms of responsiveness and stability under a particular workload.” – Wikipedia
Performance Testing

• A family of non-functional testing methods…
  
  • which **induce** a well defined **workload**
  • in order to **observe** the systems **behavior**
  • in order to **verify** and **understand** its characteristics
Performance Testing Family

• Load Testing
• Stress Testing
• Spike Testing
• Soak & Endurance Testing
• Resilience Testing
• Troubleshooting
• Configuration Testing
• Scalability Testing
Back to Topic: The Cloud!
The AWS Cloud

• IaaS, PaaS, …
• APIs and automation
• On-demand
• Cost-effective
• Scalable
Why should we care?
“The Cloud is Scalable!”
Scaling Resources ≠ Scaling Applications
Scaling Resources ≠ Scaling Applications

- Adding resources does not necessarily scale your application
- e.g. adding more EC2 Instances won't remove the bottleneck at your RDS
You need to understand...

- Your (distributed, μService, …) Application
- Your Software Architecture
- Your Cloud Environment
- Used Services and their behavior
Complexity

- Complexity has not simply vanished
- XaaS & Co have moved it “somewhere else”
- Impact on performance is still very important
Applying load tests to the cloud...
Load Testing

- Induce normal/expected workload to a given system and observe!

- Determine latency, throughput, capacity, ... of your system

- Ideally: Figure out the capacity per resource (e.g. per EC2 Instance)
Scalability Testing

- How does my **capacity** increase with additional resources?
- Requirement for **Capacity Planning** and Cost Estimation
Stress, Spike and Soak Testing

• How does your system and environment behave under serious stress? (Stress Testing)
• What happens when I'm on reddit/hacker news? (Spike Testing)
• What happens over longer period of times? (Soak Testing)
Configuration Testing

• How does the behavior change, if the configuration changes?
• Very useful to learn about the impact of environment to your system
Configuration Testing

- Instance Types
- Auto Scaling
- Throughput Provisioning
- Correct/Optimal Service usage?

- OS, Web/App Server Configuration
Availability & Resilience Testing

- (Zero-Downtime) Deployments under load
- Infrastructure changes
- Understanding failure scenarios
- Testing failover mechanisms
- Chaos Engineering
Is this different from what it used to be pre-cloud?
Yes and No!

- The testing needs and methods have not changed!
- BUT: Multiple, scalable performance test environments used to be cumbersome and very expensive
- You should utilize automation wherever you can:
  - Infrastructure, Services, Servers, Code
  - Test Data(!)
Conclusion
Conclusion

• Scaling Resources is not Scaling Applications
• Understanding is key, and testing is one tool to get there
• Complexity has not vanished, it has been shifted
• Utilize the cloud to make test setups simpler and achievable!
• …and…
Proof of Concept

• Design architecture on AWS
• Implement a prototype
• Discuss with AWS Solution Architects
• Validate!
  Run tests, observe, learn and improve!
Thank you!

Sebastian Cohnen – @tisba
StormForger.com
AWS Slide Deck Template
Deck Guidelines

Fonts, sizes, colors, and layouts are all pre-built in this template.

Color palette

Please do not use gradients, shadows, or outlines on shape elements. Limit color use for chart graphics to grayscale plus one accent color.
Helpful Resources

AWS Logo (logos for both web and print)
AWS Simple Icons (product and simple icons for architectural diagrams)
Design Request (AWS Marketing Design wiki)
Deck Asset Repository (up-to-date deck assets and templates)
Copy & Paste Content

When pasting content from another presentation please paste using “Destination Theme.”

Windows

Mac

Note: This works when copying entire slides from other presentations as long as the source presentation is also 16:9
Copy & Paste Code

When pasting content Code into a Code template please use the “Keep Text Only Function” for Windows and “Destination Theme” for Macs. If any additional coloring needs to be done to your code type please do it after pasting it into your slide.
Assets Usage

Multiple assets can be combined to create one graphic
Resizing Assets

Always hold down shift key and drag from corner when scaling assets.