

Confidence in Chaos



HOW PROPERLY APPLIED
OBSERVABILITY PRACTICES CAN
TAKE THE 'CHAOS' OUT OF
CHAOS TESTING

Narmatha Balasundaram

Senior Software Engineering
Manager,
CSE Microsoft

linkedin.com/in/narmathabala



About Me

Observability during chaos

- **Chaos Engineering** - Art of intentionally breaking the system
- **Observability** - Ability to answer new questions with existing data
- **Behavior** - 'Normal' vs 'unusual'



Preview

01

"True"
Observability

02

Golden Signals
& Actionable
Failures

03

Defining Service
Level
Agreements
and Objectives
(SLAs, SLOs)

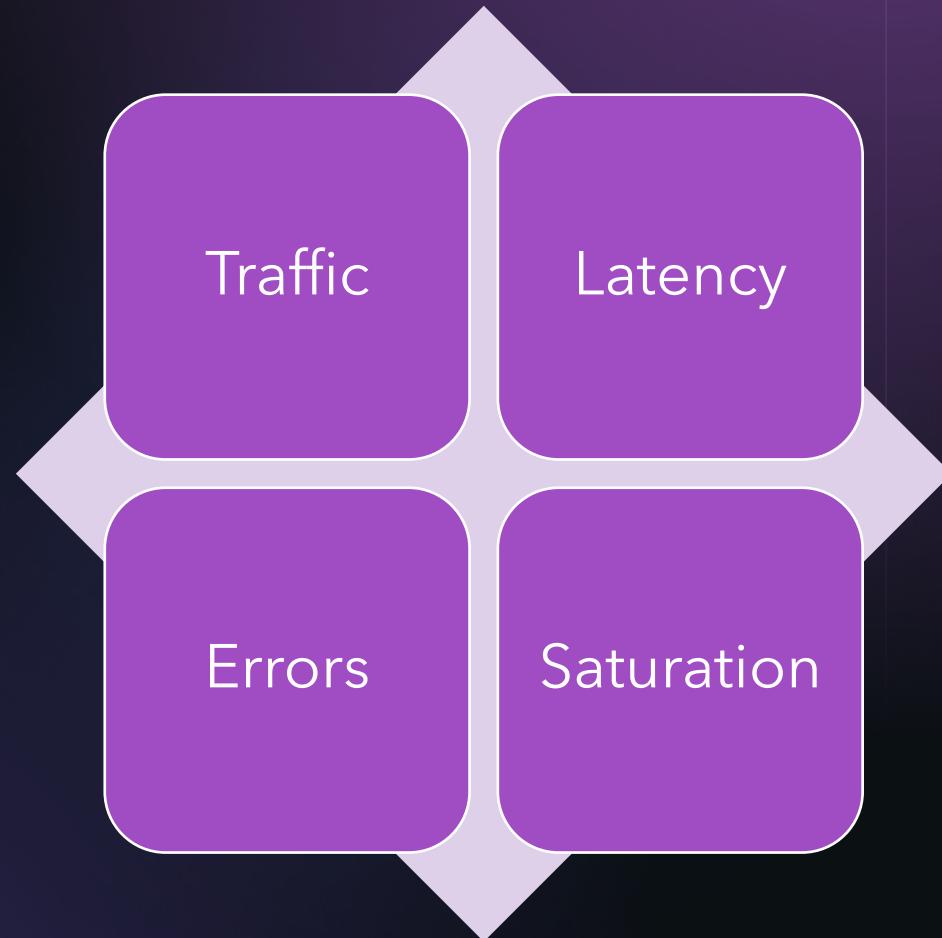
04

Monitoring &
Alerts

"True" Observability



Golden Signals



Signals	Transaction Metrics
Traffic	~200 requests/sec
Latency	500 msecs/request
Errors	0
Saturation	CPU < 20% Memory < 50% Network I/O

Actionable Failures

- Reduced Mean Time to Recovery (MTTR)
- Correlate cause and effect
- Minimizes the impact of the testing
- Runtime contextual for logs
- Build robust & relatable logs
- Ingestion and analytical engines

Service Level Agreements and Objectives (SLAs, SLOs)

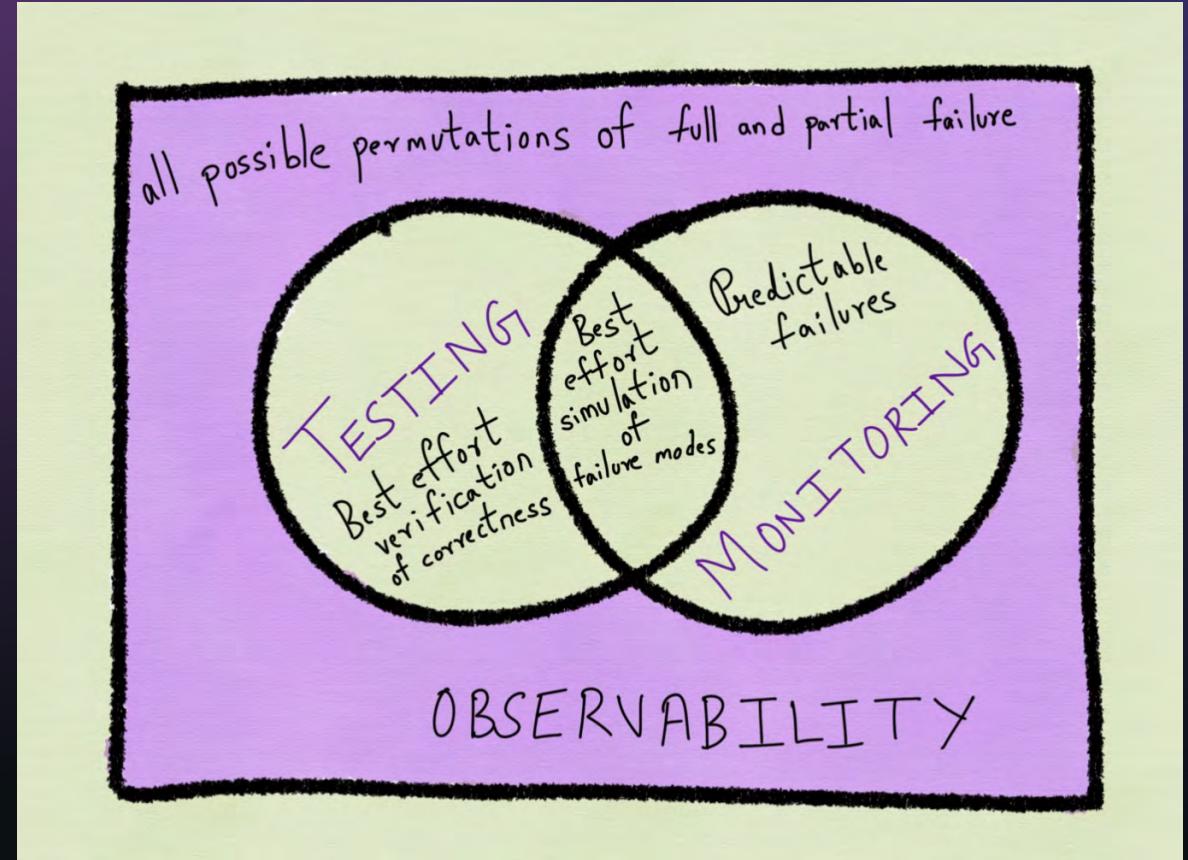
- **SLAs** - Agreement between two parties about services and their uptime & response times
- **SLOs** - Objectives the team must hit to meet agreements with clients/users
- **Error Budget** - Maximum amount of time that a technical system can fail without contractual obligations

Service Level Agreements and Objectives (SLAs, SLOs)

- **Before Chaos Testing**
 - Understanding SLOs helps evaluate the criticality of the failure
 - Helps in identifying critical issues affect user experience
- **During Chaos Engineering**
 - Measure how the system is doing without chaos
 - Helps in understand when to insert chaos in the system

Monitoring & Alerts

- Evaluate missing alarms
- Are the alarms measuring the right signals
- Evaluate thresholds for alerts
- Alerts are sent to the right teams



<https://copyconstruct.medium.com/>

Closing Thoughts

- **Start small**
 - Start with auto instrumentation
 - Leverage auto instrumentation
 - Correlate distributed logs & traces
- **Iterate Instrumentation**
 - Contextual Data => Targeted Conversations
 - Understand gaps
- **Lastly..... Celebrate learnings**

Resources



[CODE WITH ENGINEERING PLAYBOOK
\(MICROSOFT.GITHUB.IO\)](https://github.com/microsoft/CodeWithEngineeringPlaybook)



[CHAOS ENGINEERING - MICROSOFT
AZURE WELL-ARCHITECTED
FRAMEWORK | MICROSOFT DOCS](https://docs.microsoft.com/en-us/chaos-infra/well-architected-framework)



[GOOGLE - SITE RELIABILITY
ENGINEERING \(SRE.GOOGLE\)](https://sre.google/)