AWS Resilience and Chaos Engineering Day

Gunnar Grosch

Sr. Developer Advocate, AWS

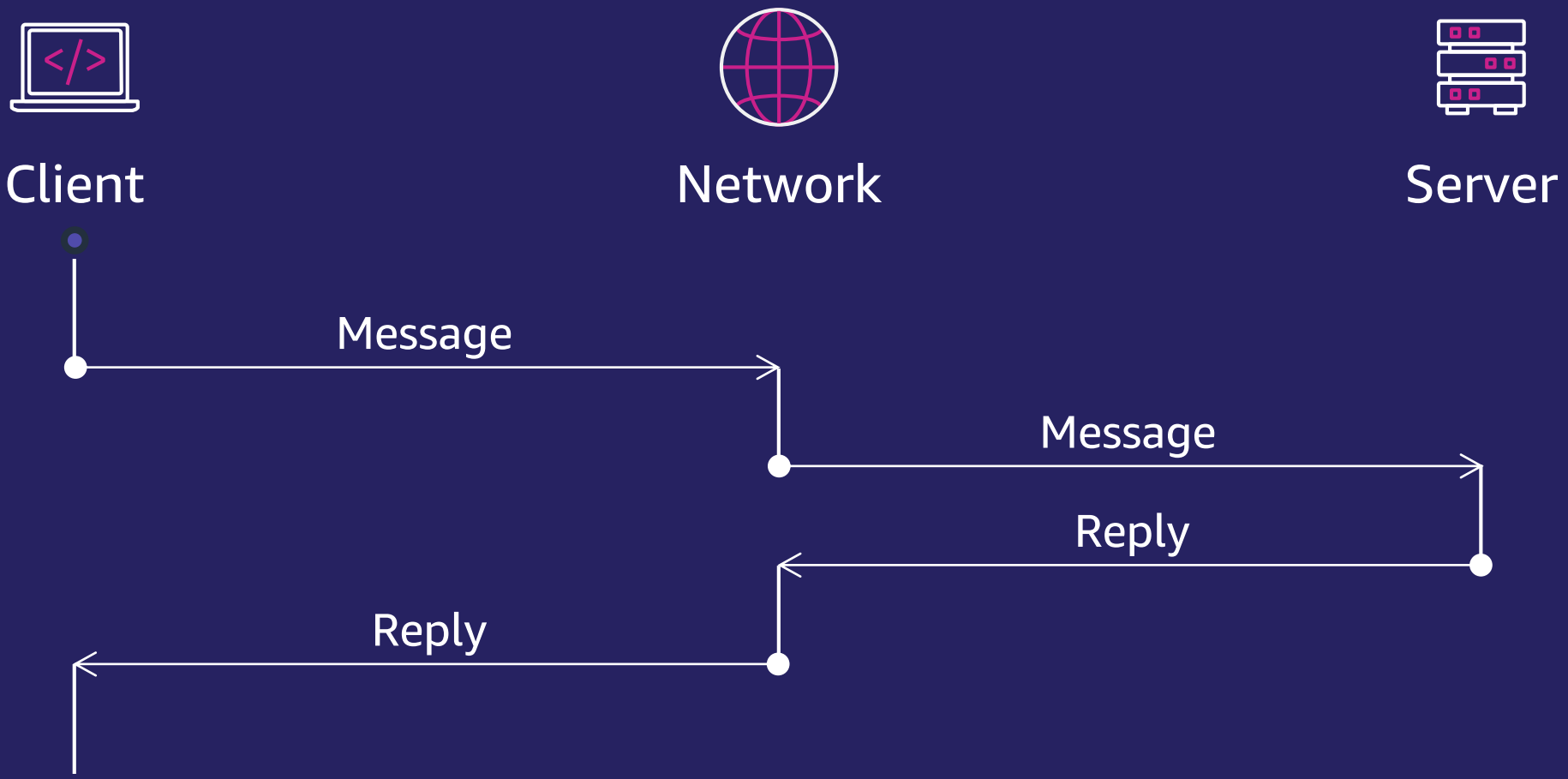
 @gunnargrosch

What we'll cover in this session

- Challenges with distributed systems
- What chaos engineering is
- Phases of chaos engineering
- Common use cases for chaos engineering

Challenges with distributed systems

Distributed systems are complex

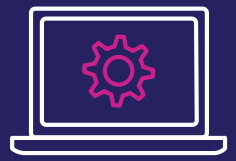


https://aws.amazon.com/builders-library/challenges-with-distributed-systems/

Traditional testing is not enough

Unit testing of components

Tested in isolation to ensure function meets expectations

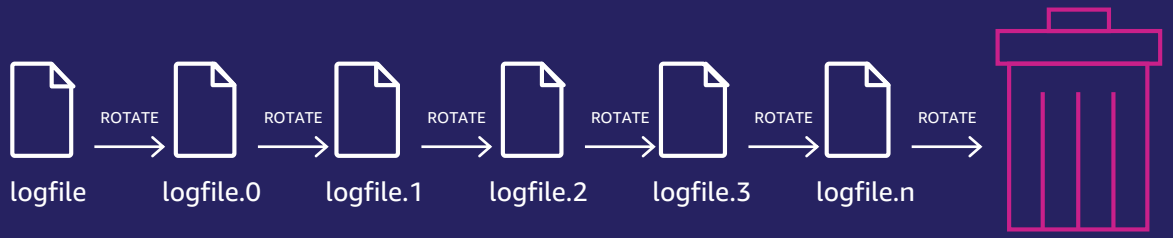


Functional testing of integrations

Each execution path tested to assure expected results

TESTING = VERIFYING A KNOWN CONDITION

And it can get more complicated...



```
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
close failed in file object destructor:
IOError: No space left on device
```

What is chaos engineering?

“Chaos engineering is the discipline of experimenting on a system in order to build confidence in the system’s capability to withstand turbulent conditions in production”

principlesofchaos.org

"Chaos engineering is the discipline of **experimenting** on a system in order to build confidence in the system's capability to withstand turbulent conditions in production"

principlesofchaos.org

“Chaos engineering is the discipline of experimenting on a system in order to **build confidence** in the system’s capability to withstand turbulent conditions in production”

principlesofchaos.org

"Chaos engineering is the discipline of experimenting on a system in order to build confidence in the system's capability to withstand **turbulent conditions in production**"

principlesofchaos.org

“Chaos engineering is the discipline of experimenting on a system in order to build confidence in the system’s capability to withstand turbulent conditions in production”

principlesofchaos.org

The process of chaos engineering

- Stressing an application by creating disruptive events
- Observing how the system responds
- Implementing improvements

Fundamental goals with chaos engineering

- Improve resilience and performance
- Uncover hidden issues
- Expose blind spots
  Monitoring, observability, and alarm
- And more

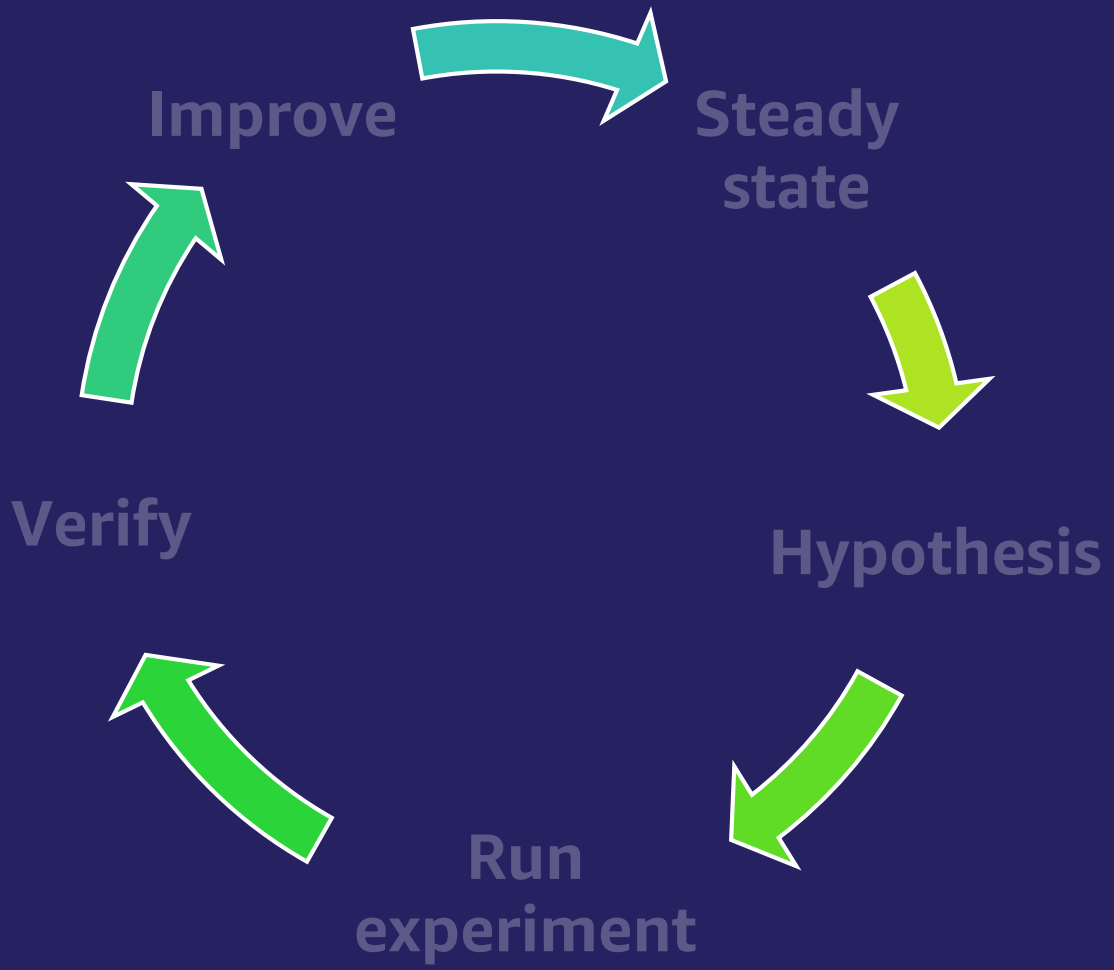
When to do chaos experiments

- Adding new code
- Creating new features
- Adding services
- Adding or changing dependencies
- Continuously
- And more

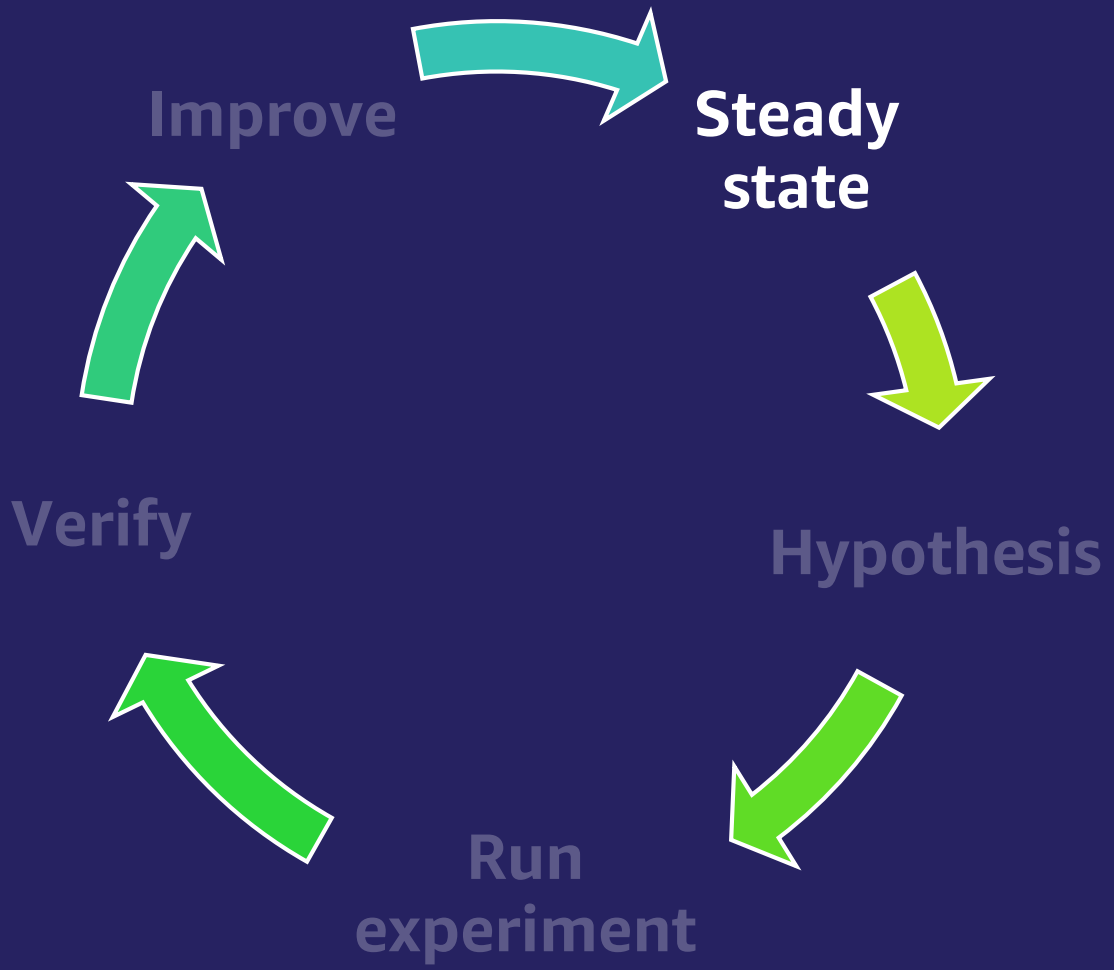
“Don’t ask what happens if a system fails; ask what happens when it fails”

Phases of chaos engineering

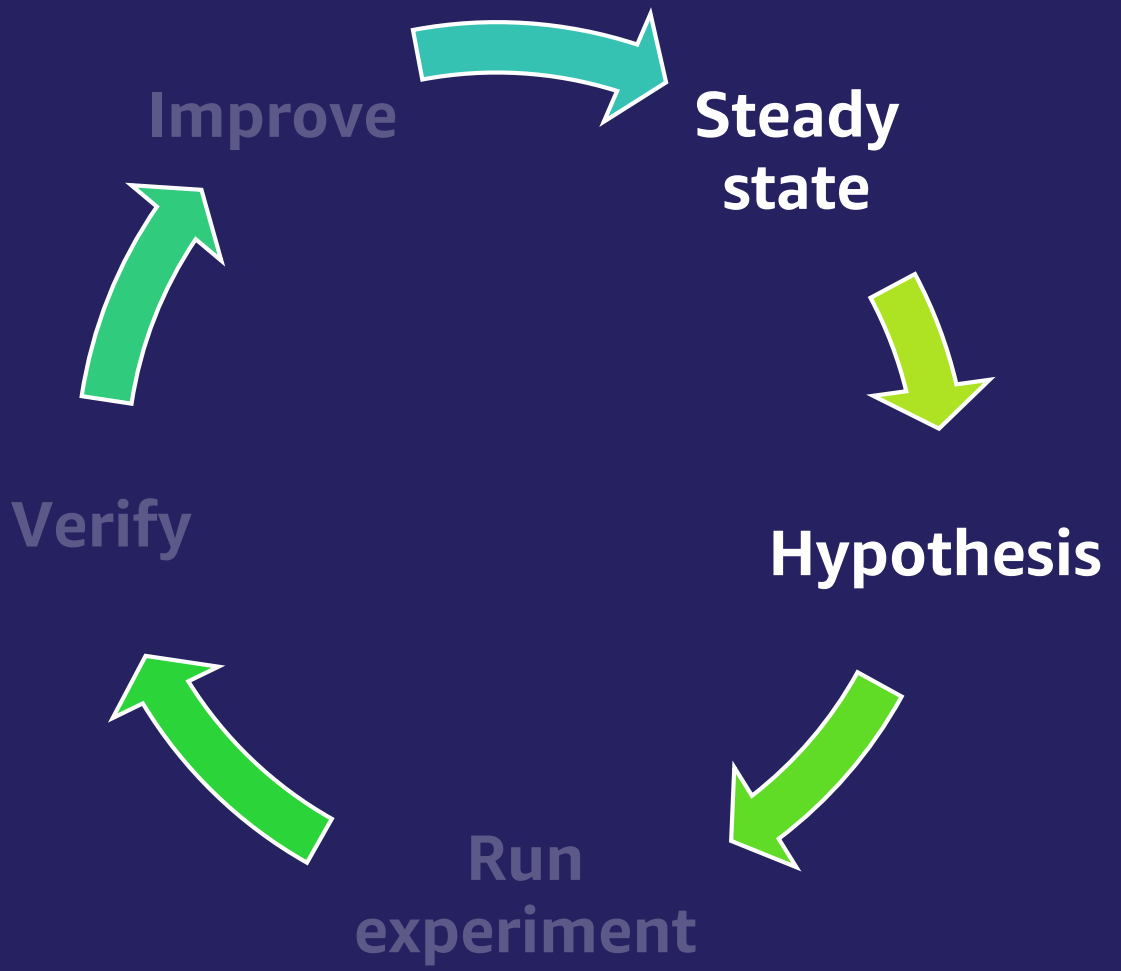
Phases of chaos engineering



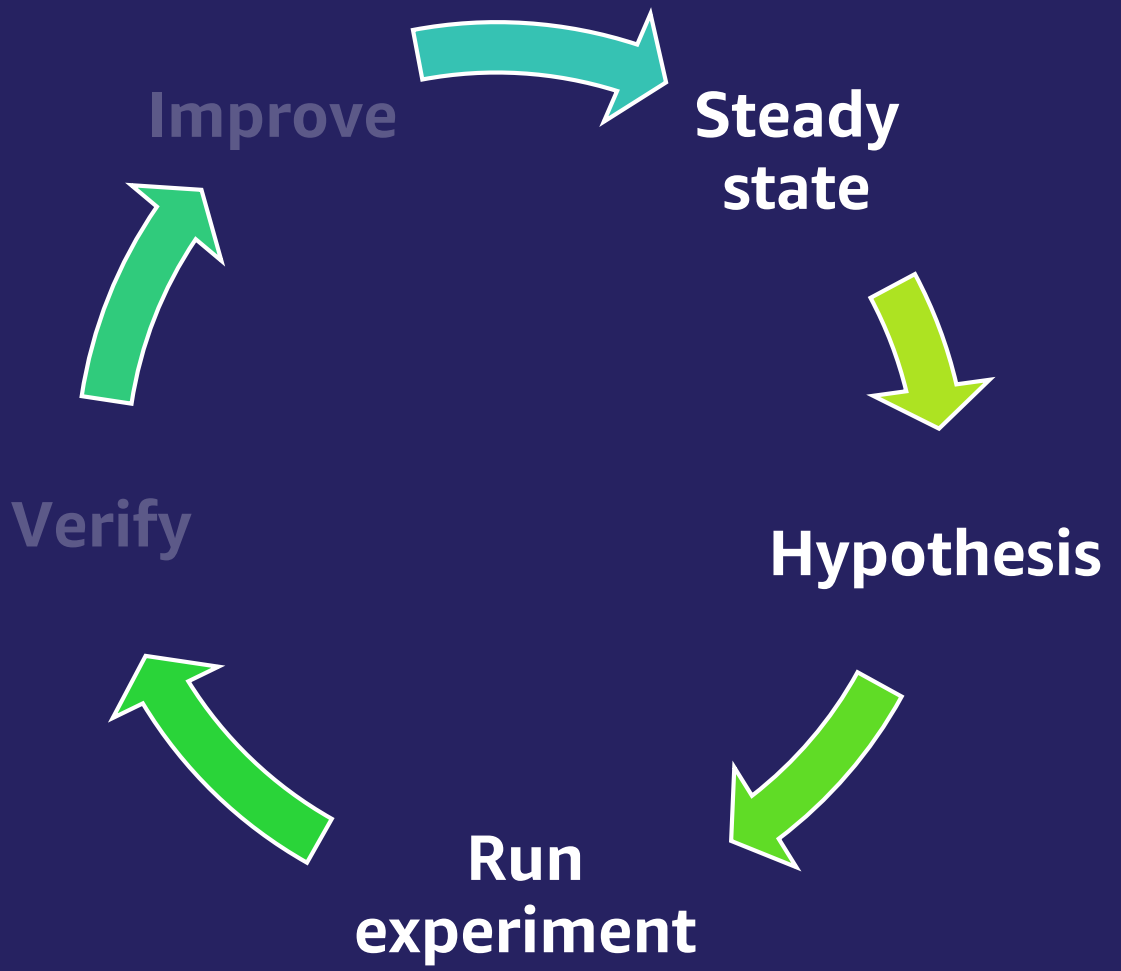
Phases of chaos engineering



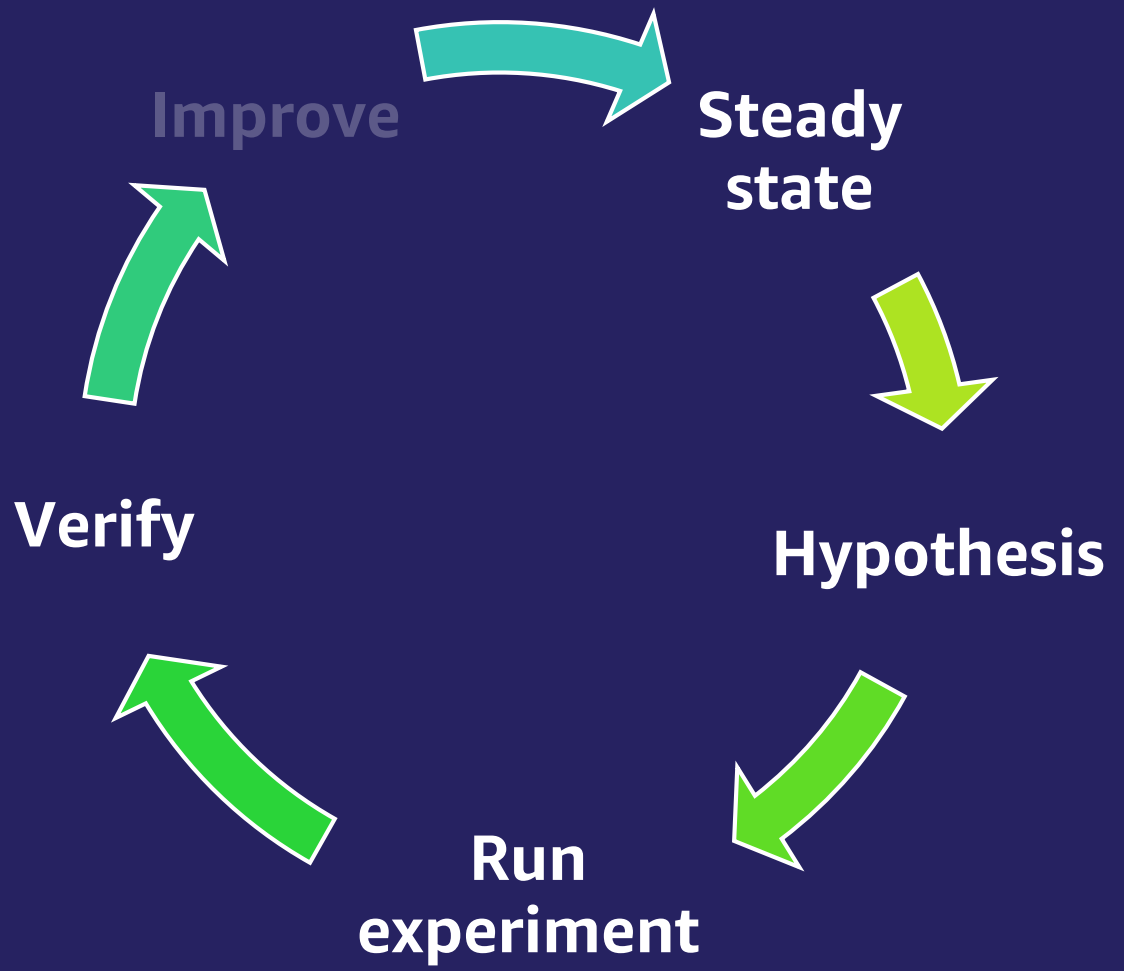
Phases of chaos engineering



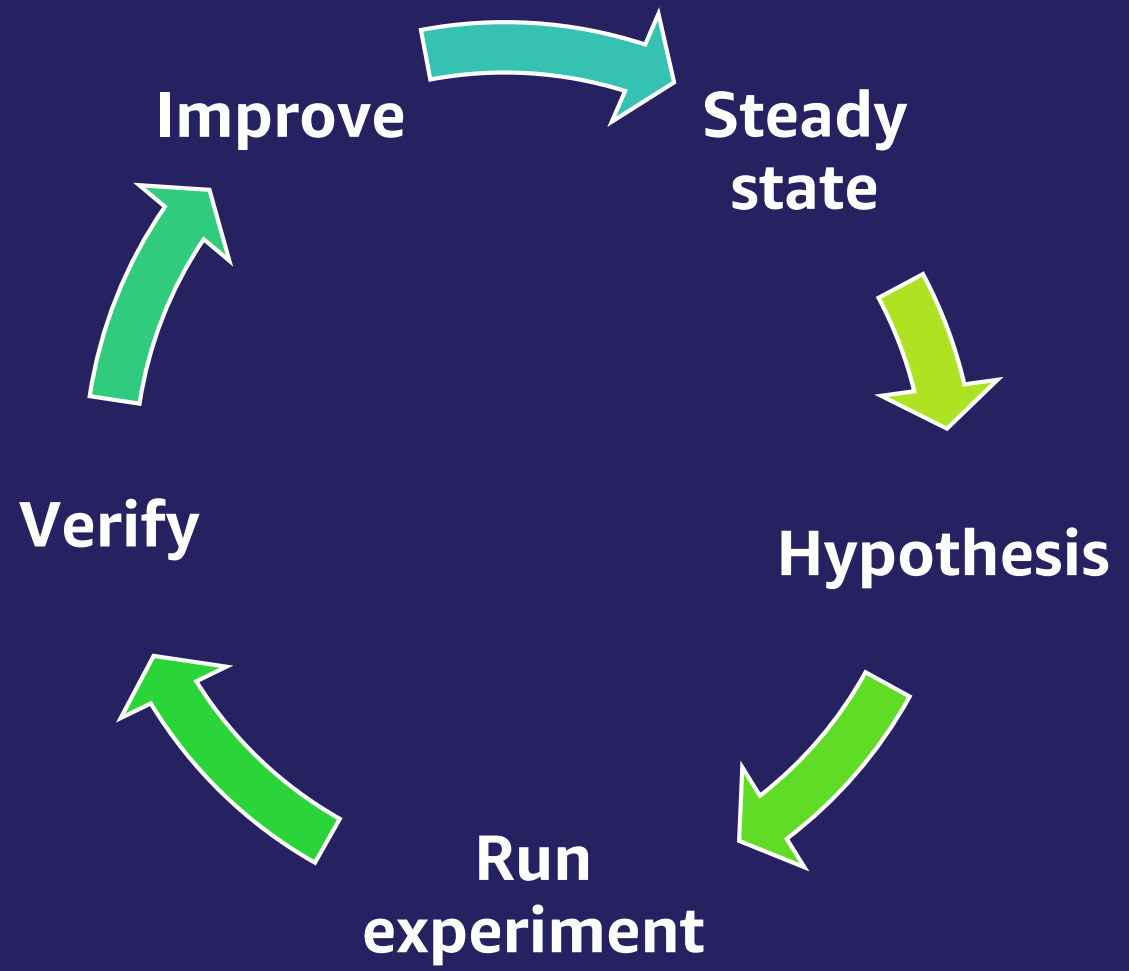
Phases of chaos engineering



Phases of chaos engineering



Phases of chaos engineering



Use cases

Use cases

One-off experiments

Periodic game days

Automated experiments

Use cases

One-off experiments

Periodic game days

Automated experiments

Use cases

One-off experiments

Periodic game days

Automated experiments

Use cases

One-off experiments

Periodic game days

Automated experiments

Automated experiments

Recurring scheduled experiments

Event-triggered experiments

Continuous delivery experiments

Automated experiments

Recurring scheduled experiments

Event-triggered experiments

Continuous delivery experiments

Automated experiments

Recurring scheduled experiments

Event-triggered experiments

Continuous delivery experiments

Automated experiments

Recurring scheduled experiments

Event-triggered experiments

Continuous delivery experiments

Automated experiments

Recurring scheduled experiments

Event-triggered experiments

Continuous delivery experiments

Use cases

One-off experiments

Periodic game days

Automated experiments

Recap

- Challenges with distributed systems
- What chaos engineering is
- Phases of chaos engineering
- Common use cases for chaos engineering

Thank you!

Gunnar Grosch

Sr. Developer Advocate, AWS

 @gunnargrosch