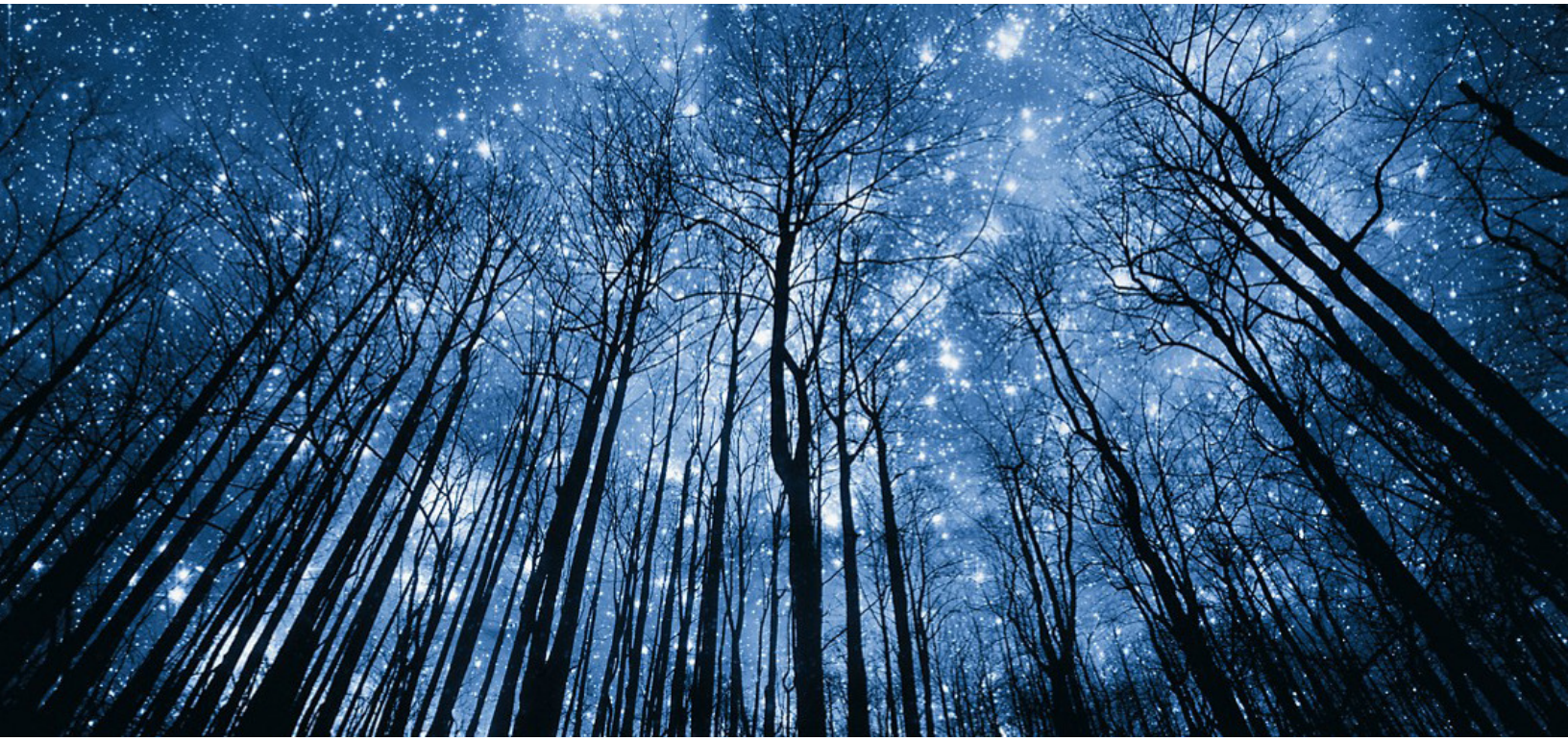
DELL Technologies

CHAOS ENGINEERING
A JOURNEY INTO RESILIENCE



Afeefa Shaista
Supervisor, Inside Sales Management
Dell Technologies

Mahesh GR

DELL Technologies
Proven Professional

Knowledge Sharing Article

The Dell Technologies Proven Professional Certification program validates a wide range of skills and competencies across multiple technologies and products.

From Associate, entry-level courses to Expert-level, experience-based exams, all professionals in or looking to begin a career in IT benefit from industry-leading training and certification paths from one of the world's most trusted technology partners.

Proven Professional certifications include:

- Cloud
- Converged and Hyperconverged Infrastructure
- Data Protection
- Data Science
- Networking
- Security
- Servers
- Storage

Courses are offered to meet different learning styles and schedules, including self-paced On Demand, remote-based Virtual Instructor-Led and in-person classrooms.

Whether you are an experienced IT professional or just getting started, Dell Technologies Proven Professional certifications are designed to clearly signal proficiency to colleagues and employers.

Learn more at www.dell.com/certification

Contents

Introduction

Chaos engineering is a method of testing distributed software that intentionally introduces failure and faulty scenarios to evaluate its ability to withstand random disruptions. These disruptions can cause applications to respond in an unpredictable manner and break under pressure. Chaos engineers aim to understand the root cause of these issues.
Practitioners' subject software to a controlled, simulated crisis to test for unstable behavior. These crises can be technical, natural, or malicious events such as an earthquake affecting data center availability or a cyberattack infecting applications and websites. As software performance degrades or fails, the chaos engineers analyze the results and make recommendations to improve the code's resiliency, ensuring the application remains intact during an emergency.
As chaos engineers gain confidence in their testing methods, they gradually increase the complexity of the scenarios and broaden the scope of the simulated disaster. By simulating a wide range of disaster scenarios and outcomes, chaos engineers can better model the behavior of applications and microservices. This enables them to provide developers with valuable insights to improve software and cloud-native infrastructure.

Chaos engineering is a discipline that involves intentionally introducing controlled failures or disruptions into a system to test its resilience and identify weaknesses. The goal is to increase the overall reliability and fault-tolerance of the system by exposing and addressing potential issues before they cause problems in production. This is done by running experiments, called "chaos experiments," in which various failure scenarios are simulated, such as network outages, server failures, and other types of disruptions.

The process of chaos engineering typically includes the following steps:

1. Define the system's expected behavior: Define the normal behavior of the system, including its inputs, outputs, and expected response times.
2. Identify failure scenarios: Identify potential failure scenarios that could occur in the system, such as network outages, server failures, and other types of disruptions.
3. Design experiments: Design experiments that simulate the identified failure scenarios and measure the system's response.
4. Run experiments: Run the experiments in a controlled environment, such as a staging or test environment.
5. Analyze results: Analyze the results of the experiments to determine how the system responded to the simulated failures and identify areas for improvement.
6. Implement improvements: Use the insights gained from the experiments to make improvements to the system's design, configuration, and monitoring.
7. The importance of doing chaos engineering is that it allows you to test the resiliency of your systems in a controlled environment before it causes issues in production. This can help you identify and fix potential issues before they cause problems for your customers.

History of Chaos Engineering

The concept of chaos engineering has its origins in the field of systems biology, where scientists studied the behavior of complex systems by introducing small perturbations and observing how the system responds. The term "chaos engineering" itself was coined by Netflix in 2010, when the company introduced a tool called Chaos Monkey to randomly shut down production servers to test the resilience of their systems.
Netflix's use of chaos engineering was driven by the need to improve the resilience of their systems as they migrated from on-premises data centers to the cloud. The increased complexity and uncertainty of the cloud environment made it difficult to predict how systems would behave under different failure scenarios. By intentionally causing failures in a controlled environment, Netflix was able to identify and address potential issues before they caused problems in production.
Since then, other companies, such as Amazon, Google, and Microsoft, have also adopted chaos engineering to improve the reliability and fault-tolerance of their systems. The practice has become increasingly popular in recent years as organizations have come to realize the benefits of identifying and addressing potential issues before they cause problems in production.
As the popularity of chaos engineering grew, new tools and frameworks have emerged to make it easier for organizations to run chaos experiments. These include Gremlin, Chaos Kong, and Litmus Chaos. Additionally, a community of practitioners has formed around the practice of chaos engineering, sharing best practices, and developing new techniques.

Netflix was one of the first companies to adopt chaos engineering as a method for testing their distributed systems. In 2009, the company migrated to AWS cloud infrastructure to deliver its online videos to a growing audience. However, the move to the cloud brought new complexities and uncertainties, such as an increase in connections and dependencies. To address these issues, Netflix sought to reduce complexity and improve production quality by introducing a tool called Chaos Monkey in 2010. This technology randomly switched off production software instances, allowing the company to test how the cloud handled its services.

As chaos engineering matured at Netflix and other organizations, innovative technologies such as Gremlin (2016) emerged, becoming more targeted and knowledge based. This has led to the emergence of specialized chaos engineers who are dedicated to disrupting cloud software and on-premises systems to make them more resilient. Today, chaos engineering is an established profession, and it is widely used to stabilize cloud software by intentionally causing failures in a controlled environment.

Since then, other companies, such as Amazon, Google, and Microsoft, have also adopted chaos engineering to improve the reliability and fault-tolerance of their systems. The practice has become increasingly popular in recent years as organizations have come to realize the benefits of identifying and addressing potential issues before they cause problems in production.
As the popularity of chaos engineering grew, new tools and frameworks have emerged to make it easier for organizations to run chaos experiments. These include Gremlin, Chaos Kong, and Litmus Chaos. Additionally, a community of practitioners has formed around the practice of chaos engineering, sharing best practices, and developing new techniques.
Chaos Engineering is now widely accepted as a best practice for ensuring the resiliency of distributed systems and is increasingly being adopted by companies in various industries, including finance, healthcare, retail, and more. Some companies have also developed their own internal tools and practices for implementing chaos engineering, further increasing its popularity and reach.

Why Chaos Engineering?

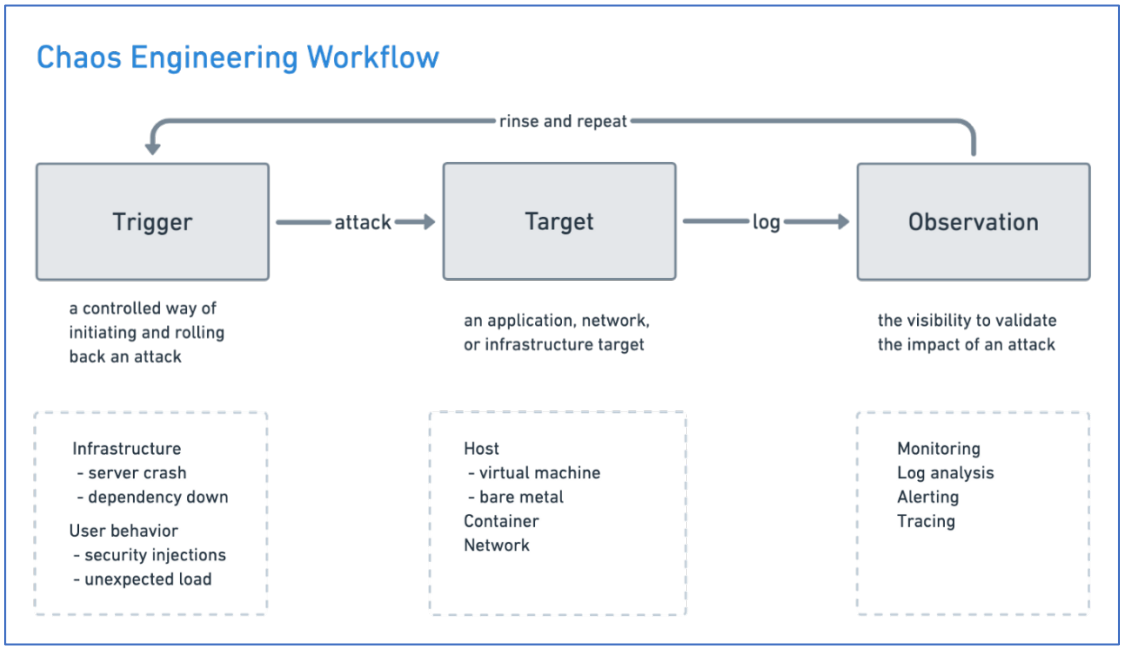


Figure 1: Chaos Engineering Workflow [Ref 1]

Chaos engineering is a powerful tool for improving the reliability and fault-tolerance of distributed systems. By intentionally causing failures in a controlled environment, chaos engineering allows organizations to:

1. Identify weaknesses: Chaos engineering helps identify weaknesses in a system that may not be apparent under normal conditions. By simulating different failure scenarios, organizations can better understand how their systems respond to disruptions and identify areas for improvement.
2. Improve resilience: By identifying and addressing potential issues before they cause problems in production, chaos engineering can help improve the resilience of a system. This can lead to fewer outages, faster recovery times, and a better user experience.
3. Reduce complexity: Chaos engineering can help reduce the complexity of distributed systems by exposing and addressing issues that may be hidden due to the complexity of the system.
4. Enhance monitoring and observability: By running chaos experiments, organizations can gain a better understanding of their systems and identify areas where they need to improve monitoring and observability.
5. Prepare for the unexpected: Chaos engineering helps organizations prepare for unexpected events by simulating a wide range of failure scenarios. This allows organizations to better understand how their systems will respond in the event of a real crisis and make any necessary adjustments to improve their resiliency.

6. Improve overall reliability: By regularly performing chaos experiments, organizations can improve the overall reliability of their systems, which can lead to fewer outages, faster recovery times, and a better user experience.
7. Increased customer satisfaction: By improving the reliability and fault-tolerance of systems, organizations can provide a better user experience for their customers, leading to increased satisfaction and loyalty.
8. Better incident management: By simulating different failure scenarios and testing their response, organizations can improve their incident management procedures and be better prepared to handle real-world crises.
9. Testing in production environment: By performing chaos experiments in a production-like environment, organizations can test their systems in a more realistic setting and gain a better understanding of how they will perform in real-world situations.
10. Improve team collaboration: By involving different teams such as development, operations, and security in the chaos engineering process, organizations can improve collaboration and communication among teams, leading to better results.
11. Better understanding of inter-dependencies: By running chaos experiments, organizations can gain a better understanding of the inter-dependencies between different systems and services, allowing them to make more informed decisions about how to improve their overall architecture.
12. Continuous improvement: By regularly performing chaos experiments, organizations can continuously improve the reliability and fault-tolerance of their systems, leading to a more robust and resilient infrastructure.

How does chaos engineering work?

Chaos engineering works by intentionally introducing controlled failure or disruption into a system to test its resilience and identify weaknesses.

Chaos engineering starts with understanding the software's expected behavior.

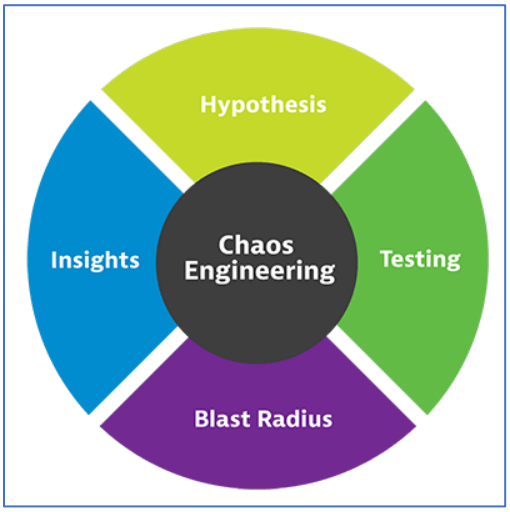


Figure 2: Chaos Engineering Workflow [Ref 2]

The process typically includes the following steps:

1. Define the system's expected behavior: Define the normal behavior of the system, including its inputs, outputs, and expected response times.
2. Identify failure scenarios: Identify potential failure scenarios that could occur in the system, such as network outages, server failures, and other types of disruptions.
3. Design experiments: Design experiments that simulate the identified failure scenarios and measure the system's response.
4. Run experiments: Run the experiments in a controlled environment, such as a staging or test environment.
5. Analyze results: Analyze the results of the experiments to determine how the system responded to the simulated failures and identify areas for improvement.
6. Implement improvements: Use the insights gained from the experiments to make improvements to the system's design, configuration, and monitoring.

Chaos engineering experiments are typically run on a small scale and in a controlled environment, such as a staging or test environment, before being gradually scaled up to the production environment. This allows organizations to test their systems in a safe and controlled environment and make any necessary adjustments before implementing them in production.

The specific tools and techniques used in chaos engineering vary depending on the organization and the system being tested. Some organizations use open-source tools like Gremlin or Chaos Monkey, while others use in-house tools or frameworks. The key is to have a repeatable, controlled, and well-defined process in place to be able to learn from the results of the experiments.

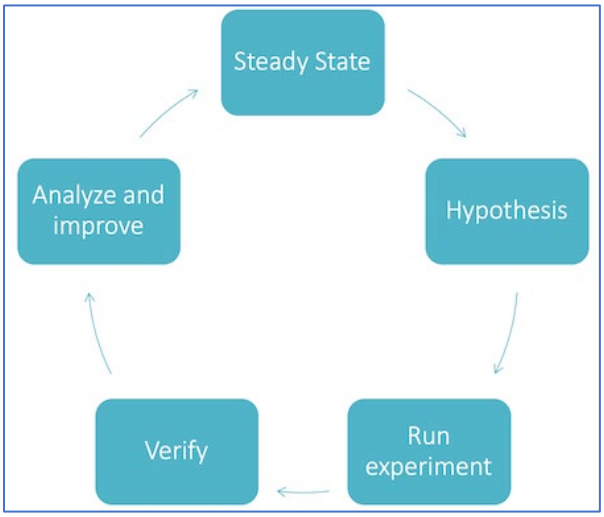


Figure 3: Chaos Engineering Workflow [Ref 3]

Who Uses Chaos Engineering?

Chaos engineering is used by organizations of all sizes and industries to improve the reliability and fault-tolerance of their systems. Some of the most common industries that use chaos engineering include:

1. Technology: Companies in the technology industry, such as Netflix, Amazon, Google, and Microsoft, have been among the early adopters of chaos engineering. They use it to test the resilience of their cloud-based systems and improve the user experience.
2. Financial Services: Banks and other financial institutions use chaos engineering to test the resilience of their systems and ensure that they can continue to provide services to customers even in the event of a crisis.
3. Healthcare: Hospitals and other healthcare providers use chaos engineering to test the resilience of their systems, including electronic health records and other critical systems, to ensure that they can continue to provide care to patients even in the event of a crisis.
4. Retail: Retail companies use chaos engineering to test the resilience of their systems, including e-commerce platforms and inventory management systems, to ensure that they can continue to provide services to customers even in the event of a crisis.
5. Government: Government agencies use chaos engineering to test the resilience of their systems, including critical infrastructure systems, to ensure that they can continue to provide services to citizens even in the event of a crisis.
6. Gaming: Gaming companies use chaos engineering to test the resilience of their systems.

The Benefits of Chaos Engineering

The benefits of chaos testing include:

1. Improved resilience: By simulating failures and disruptions, chaos testing helps identify and fix weaknesses in a system's design, making it more resilient to real-world failures.
2. Increased reliability: By identifying and mitigating potential points of failure, chaos testing can improve the overall reliability of a system.
3. Better incident response: By simulating failures and disruptions, chaos testing can help organizations develop and refine their incident response processes, making them more efficient and effective in the event of a real incident.
4. Cost savings: By identifying and fixing potential issues before they occur, chaos testing can help organizations save money by reducing the need for emergency repairs and reducing downtime.

5. Faster recovery: By identifying and mitigating points of failure, chaos testing can help organizations recover more quickly from real-world incidents.
6. Better understanding of the system: By simulating different scenarios, chaos testing can help organizations gain a better understanding of how their systems behave under different conditions.

The Challenges of Chaos Engineering

The challenges and pitfalls of chaos engineering include:

1. Complexity: Chaos engineering can be a complex process, involving the coordination of multiple teams and systems. It requires significant planning and organization to be done correctly.
2. Risk: Chaos engineering introduces the risk of causing unintended consequences and real damage to the system being tested. It is important to have proper safety measures in place and to limit the scope of testing.
3. Difficulty in measuring results: Measuring the results of chaos engineering can be difficult because it can be hard to quantify the benefits of increased resilience and reliability.
4. Lack of buy-in: Some team members and stakeholders may be resistant to the idea of intentionally causing failures and disruptions in the systems they are responsible for.
5. Limited applicability: Chaos engineering may not be appropriate for all systems or situations, and it may not be feasible to test certain types of failures or disruptions.
6. Limited resources: Chaos engineering may require additional resources, such as specialized tools or personnel, which can be difficult to obtain.

It is important to note that chaos engineering should only be done by experienced professionals, with proper planning and safety measures in place, and with a clear understanding of the risks and potential consequences.

How to get started with Chaos Engineering

Getting started with chaos engineering can involve the following steps:

1. Define your goals: Clearly define what you hope to achieve through chaos engineering, such as improved resilience or better incident response.
2. Identify your systems: Determine which systems and components will be included in the testing and identify any potential points of failure.
3. Create a plan: Create a detailed plan for the chaos engineering tests, including what types of failures and disruptions will be simulated and how the tests will be executed.

4. Implement safety measures: Implement safety measures to ensure that the testing does not cause unintended harm or damage to the systems being tested.
5. Execute the test: Run the chaos engineering tests, taking care to monitor the systems throughout the test and to have a clear rollback plan in case of unexpected results.
6. Analyze the results: Analyze the results of the chaos engineering tests and use them to improve the systems and processes that were tested.
7. Iterate and improve: Continuously iterate and improve the chaos engineering process by incorporating feedback, adjusting the testing scope, and incorporating new test scenarios.
8. Communicate and educate: Communicate the results and findings of the chaos engineering to the rest of the organization and educate team members on the benefits and best practices of chaos engineering.

Controlling the Chaos

Controlling the chaos in chaos engineering refers to the ability to limit the scope and impact of the tests to minimize the risk of unintended consequences. Some ways to control the chaos include:

1. Starting small: Start by conducting experiments on a small scale, with limited scope and impact. This will allow you to gain experience and confidence before moving on to more complex or risky tests.
2. Using simulated failures: Use simulated failures rather than real-world failures to minimize the risk of unintended consequences.
3. Using canaries: Use canary servers or test environments to conduct the experiments, which are isolated from the production environment.
4. Implementing rollback plans: Have a plan in place to roll back any changes made during the test in case of unexpected results.
5. Conducting post-mortems: After each test, conduct a post-mortem to review what worked and what did not and use those insights to improve the next test.
6. Coordinating with other teams: Coordinate with other teams and stakeholders in the organization to ensure that everyone is aware of the chaos engineering tests and understands their purpose and potential impact.
7. Continuously monitoring: Continuously monitor the systems during the test to ensure that everything is running as expected and to quickly respond to any unexpected behavior.
8. By following these steps and best practices, organizations can effectively control the chaos in their chaos engineering experiments and minimize the risk of unintended consequences.

Conclusion

In conclusion, chaos engineering is a technique used by gaming companies and other organizations to test the resilience of their systems by intentionally introducing failures and disruptions. The benefits of chaos testing include improved resilience, increased reliability, better incident response, cost savings, faster recovery, and better understanding of the system. However, chaos engineering also has its challenges and pitfalls, such as complexity, risk, difficulty in measuring results, lack of buy-in, limited applicability and limited resources. To get started with chaos engineering, one should define the goals, identify the systems, create a plan, implement safety measures, execute the test, analyze the results, iterate, and improve, and communicate and educate. To control the chaos, it is important to start small, use simulated failures, canaries, rollback plans, conduct post-mortems, coordinate with other teams, and continuously monitor the systems.

References

1. https://medium.com/better-practices/chaos-d3ef238ec328
2. https://www.dynatrace.com/news/blog/what-is-chaos-engineering/
3. https://www.microsoft.com/de-de/techwiese/cloud-native-community-blog/der-status-quo-des-chaos-engineerings.aspx
4. https://www.dynatrace.com/news/blog/what-is-chaos-engineering/