International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I6P104

Volume 72 Issue 6, 25-30, June 2024
© 2024 Seventh Sense Research Group®

Original Article

Harnessing Chaos: The Role of Chaos Engineering in Cloud Applications and Impacts on Site Reliability Engineering

Rahul Yadav

Lead Application Architect, Humana Inc, Louisville, Kentucky, USA.

Received: 19 April 2024 Revised: 23 May 2024 Accepted: 04 June 2024 Published: 15 June 2024

Abstract *- In the ever-evolving landscape of cloud computing, where reliability and resilience are paramount, the concept of chaos might seem counterintuitive. However, within this realm, chaos is not only embraced but actively harnessed as a means of ensuring systems are robust and capable of withstanding unexpected failures. At the heart of this approach lies a powerful methodology known as Chaos Engineering. Chaos Engineering is a disciplined approach to experimenting on distributed systems to build confidence in their resilience. It involves intentionally introducing controlled disruptions or failures into a system to observe how it responds under adverse conditions.* This paper investigates and examines how Chaos Engineering techniques might be integrated into cloud-based systems and how this can affect Site Reliability Engineering (SRE) techniques. *By simulating real-world failures in a controlled environment, organizations can identify weaknesses, uncover hidden dependencies, and improve the overall reliability of their systems.*

Keywords *- Azure chaos studio, Cloud technologies*, *Chaos Engineering, Enterprise architecture, Site reliability engineering.*

1. Introduction

Cloud computing offers unparalleled scalability, flexibility, and cost-effectiveness, but it also introduces new challenges in terms of reliability and resilience. Cloud applications are inherently distributed and complex, with dependencies spanning multiple services and infrastructure components. Chaos engineering provides a systematic way to test the resilience of cloud applications by subjecting them to various failure scenarios, such as service outages, network latency, or resources. According to Cambridge, the term resilence is about to consider that return to good condition after the problems get solved. Resilience in cloud computing is the ability of a cloud system to recover from failures and continue functioning as usual. This paper investigates and examines how Chaos Engineering techniques might be integrated into cloud-based systems and how this can affect Site Reliability Engineering (SRE) techniques.

2. Literature Review

A key technique for improving the dependability and robustness of contemporary cloud-based applications is chaos engineering. The purpose of this study of the literature is to present an overview of the field's body of knowledge about the application of Chaos Engineering to cloud systems, with a focus on how it affects Site Reliability Engineering (SRE).

System dependability and operational efficiency can be significantly increased by incorporating Chaos Engineering into SRE frameworks. Through early identification and resolution of probable malfunctions, establishments can reduce the consequences of events and improve overall service dependability. (Miles, 2019) The significance of taking preventative action to preserve system dependability is emphasized by site reliability engineering. According to (Beyer et al., 2016), chaos engineering is strongly aligned with SRE concepts as it offers an ongoing monitoring and improvement strategy for resilient systems. The case studies from the industry show how well Chaos Engineering works to find and fix system flaws. Chaos Engineering techniques have been effectively applied by Netflix and Amazon, amongst others as well, to improve system stability (Nygaard, 2007; Amazon Web Services, 2020).

Several platforms and technologies make it easier to conduct Chaos Engineering experiments in cloud systems. Notable instances that offer the ability to inject faults and evaluate the resilience of systems are Gremlin and Litmus (Gremlin, n.d.; LitmusChaos, n.d.). The techniques used by Netflix engineers to test the system's resilience to unforeseen failures gave rise to the field of chaos engineering (Nygard, 2012). To find weaknesses and increase system resilience, it makes use of concepts like fault injection and regulated experimenting (Miles, 2019).

3. Benefits of Chaos Engineering in Cloud Applications

3.1. Proactive Identification of Weaknesses

By intentionally introducing failures into cloud applications, organizations can identify weaknesses and vulnerabilities that may otherwise remain hidden until they manifest during an actual outage or incident.

3.2. Improved System Reliability

Chaos engineering helps organizations build more resilient cloud applications by exposing and addressing potential points of failure. This results in increased uptime, improved user experience, and enhanced customer satisfaction.

3.3. Optimized Resource Utilization

By testing the resilience of cloud applications under different load and failure conditions, organizations can optimize resource allocation and ensure efficient utilization of cloud resources.

3.4. Enhanced Incident Response Preparedness

Chaos engineering prepares organizations to respond effectively to real-world incidents by simulating various failure scenarios and providing insights into how the system behaves under stress.

3.5. Cultural Shift Towards Reliability

Embracing chaos engineering fosters a culture of reliability and resilience within organizations, where failure is seen as an opportunity for learning and improvement rather than a source of fear or blame.

4. Chaos Engineering and Site Reliability Engineering (SRE)

Chaos Engineering and Site Reliability Engineering (SRE) share a common goal: to build and maintain reliable, scalable, and resilient systems. Chaos Engineering complements the principles and practices of SRE by providing a systematic approach to testing the resilience of systems and identifying weaknesses before they impact users or customers. SRE teams can leverage chaos engineering techniques to validate assumptions, test hypotheses, and improve the reliability of their systems. By integrating chaos engineering into their workflows, SRE teams can proactively identify and address reliability issues, optimize incident response processes, and ultimately deliver a better user experience.

5. Approach

Before discussing the retry approaches, let us define core principles that will be at the center of the approach to be chosen for the scenarios.

5.1. Core Principles

5.1.1. Define Objectives

Clearly define what you want to achieve through chaos engineering. Whether it's improving system resilience, identifying weaknesses, or testing failure recovery mechanisms, having clear objectives is crucial.

5.1.2. Identify Critical Systems

Determine which systems are critical to your organization's operations and prioritize them for chaos engineering experiments.

5.1.3. Hypothesize Scenarios:

Brainstorm potential failure scenarios that could occur in your systems, such as server crashes, network failures, or database outages.

5.1.4. Design Experiments

Design controlled experiments to simulate these failure scenarios in a controlled environment. Start with small, non-critical components before moving on to larger, more complex systems.

5.1.5. Implement Chaos Tools

Choose and implement chaos engineering tools that fit your requirements. Tools like Chaos Monkey, Gremlin, or custom scripts can help inject failures into your systems.

5.1.6. Execute Experiments

Execute the chaos experiments during off-peak hours or in a staging environment to minimize impact on production systems. Monitor the experiments closely to ensure they don't cause any irreversible damage.

5.1.7. Analyze Results

Analyze the impact of the chaos experiments on your systems. Identify weaknesses, areas for improvement, and unexpected behavior.

5.1.8. Iterate and Improve

Based on the results, iterate on your systems and processes to improve resilience and reliability. Continuously refine your chaos engineering practices, staying ahead of potential failures.

5.1.9. Document and Share Learnings

Document the results of your chaos experiments and share them with your team and the broader community. Encourage knowledge sharing and collaboration to improve overall system resilience.

5.2. Implementation

5.2.1. Enable Chaos Studio on the VM you Created

1. Open the Azure portal.
2. Search for Chaos Studio in the search bar.

3. Select Targets and go to the VM you created.
4. Select the checkbox next to your VM. Select Enable targets > Enable service-direct targets from the dropdown menu.

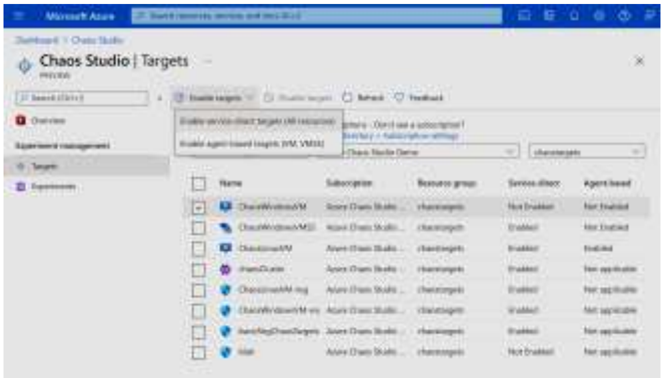


Fig. 1 Quickstart-virtual-machine-enabled

5. Confirm that the desired resource is listed. Select Review + Enable, then Enable.
6. A notification appears and indicates that the resource selected was successfully enabled.

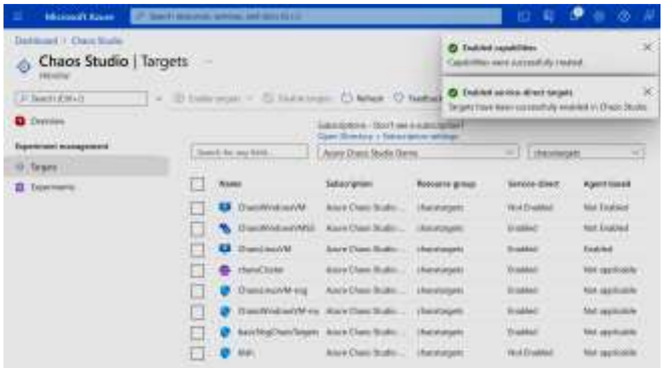


Fig. 2 Tutorial-service-direct-targets-enable-confirm

5.2.2. Create an Experiment

1. Select Experiments.

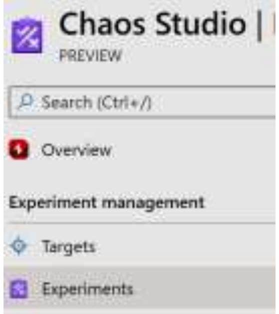
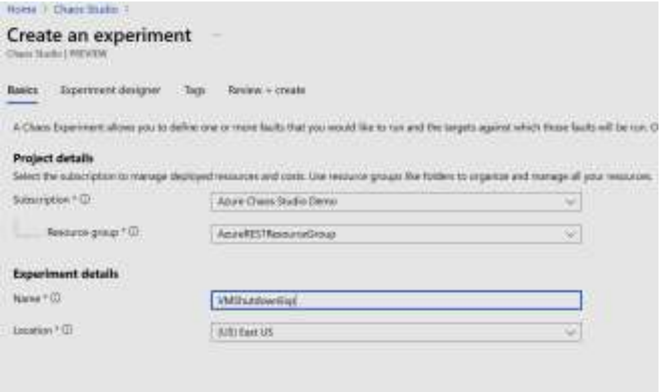


Fig. 3 Quickstart-left-experiment

2. Select Create > New experiment.
3. Fill in the Subscription, Resource Group, and Location boxes where you want to deploy the chaos experiment. Give your experiment a name. Select Next: Experiment designer.



Subscription * Azure Chaos Studio Demo

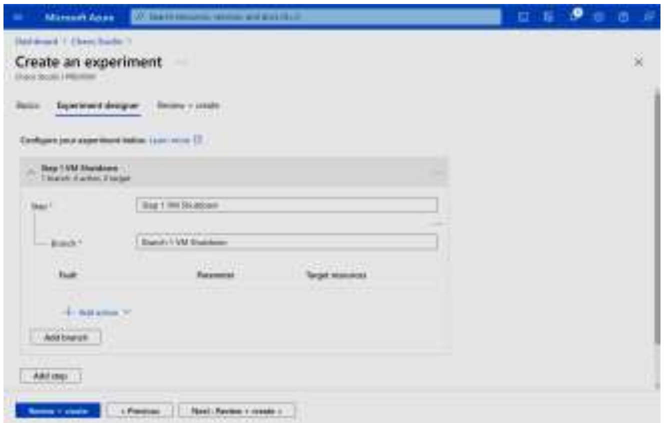
Resource group * AzureRESTResourceGroup

Name * VMShutdownExp

Location * (US) East US

Fig. 4 Quickstart-service-direct-add-basics

4. In the Chaos Studio experiment designer, give a friendly name to your Step and Branch. Select Add action > Add fault.

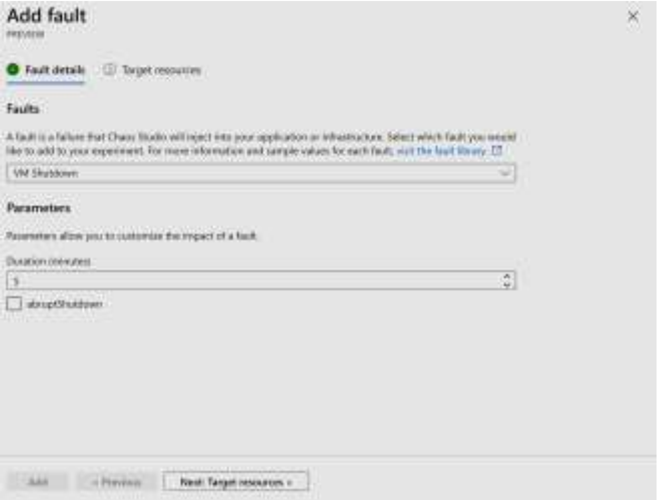


Step * Step 1 VM Shutdown

Branch * Branch 1 VM Shutdown

Fig. 5 Quickstart-service-direct-add-designer

5. Select VM Shutdown from the dropdown list. Then, fill in the Duration box with the number of minutes you want the failure to last.



A fault is a failure that Chaos Studio will inject into your application or infrastructure. Select which fault you would like to add to your experiment. For more information and sample values for each fault, visit the fault library

VM Shutdown

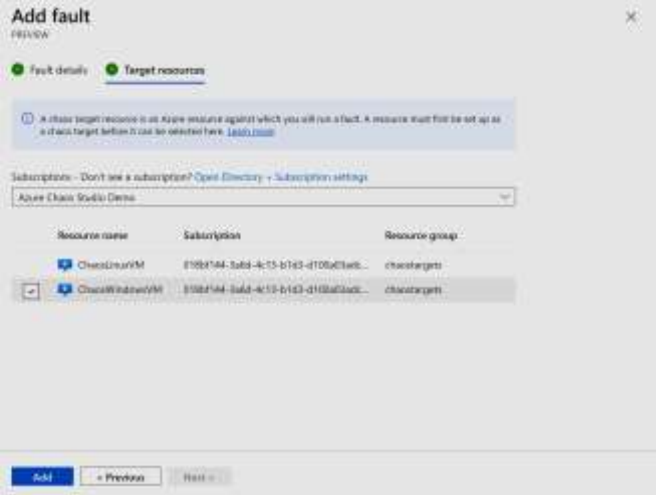
Parameters allow you to customize the impact of a fault.

Duration (minutes) 5

□ abruptShutdown

Fig. 6 Quickstart-service-direct-add-fault

6. Select Next: Target resources



Add fault
PREVIEW

● Fault details ● Target resources

ⓘ A chaos target resource is an Azure resource against which you will run a fault. A resource must first be set up as a chaos target before it can be selected here. Learn more

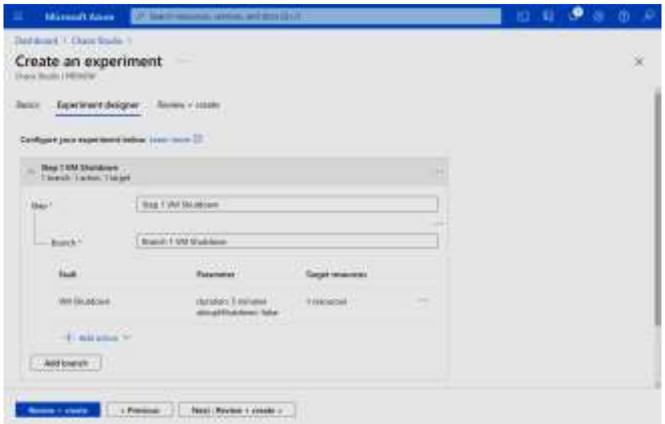
Subscriptions - Don't see a subscription? Open Directory + Subscription settings

Azure Chaos Studio Demo

	Resource name	Subscription	Resource group
	ChaosLinuxVM	818bf144-3a8d-4c13-b1d3-d108a01adb...	chaostargets
☑	ChaosWindowsVM	818bf144-3a8d-4c13-b1d3-d108a01adb...	chaostargets

Fig. 7 Quickstart-service-direct-add-targets

7. Select Add



Microsoft Azure
Search resources, services, and docs (G+/)
Dashboard > Chaos Studio >

Create an experiment
Chaos Studio | PREVIEW

Configure your experiment below. Learn more

Step 1 VM Shutdown
1 branch, 1 action, 1 target

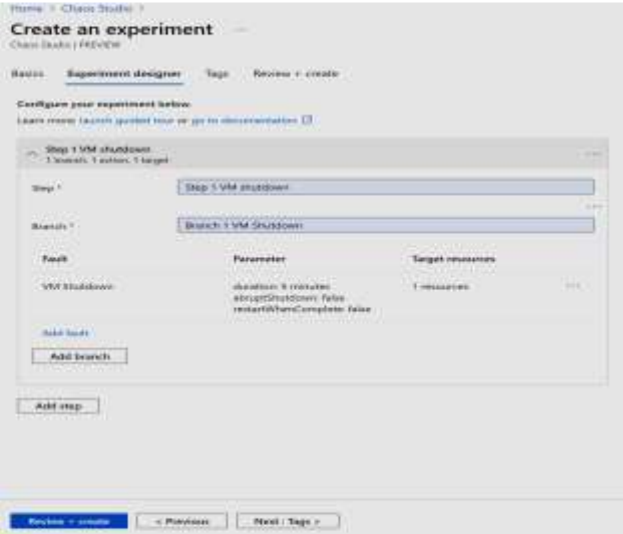
Step * Step 1 VM Shutdown

Branch * Branch 1 VM Shutdown

Fault	Parameter	Target resources
VM Shutdown	duration: 5 minutes abruptShutdown: false	1 resource(s)

Fig. 8 Quickstart-add-target

8. Verify that your experiment looks correct, and then select Review + Create > Create.



Create an experiment
Chaos Studio | PREVIEW

Configure your experiment below.
Learn more: launch guided tour or go to documentation

Step 1 VM shutdown
1 branch, 1 action, 1 target

Step * Step 1 VM shutdown

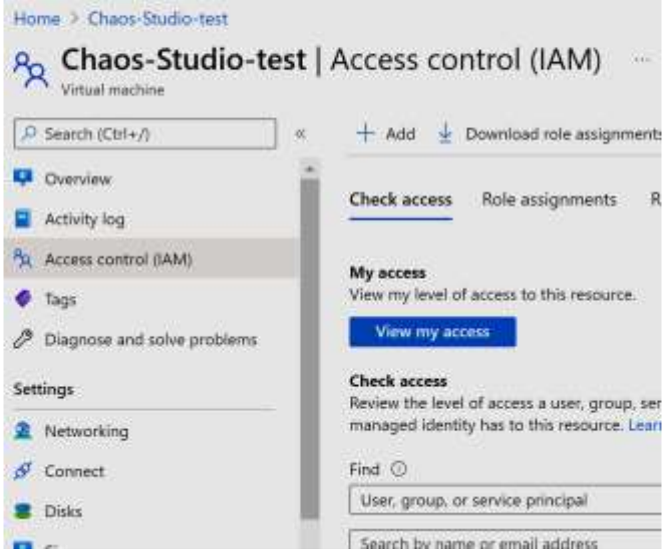
Branch * Branch 1 VM Shutdown

Fault	Parameter	Target resources
VM Shutdown	duration: 5 minutes abruptShutdown: false restartWhenComplete: false	1 resources

Fig. 9 Quickstart-review-and-create

5.3. Give Experiment Permission to your VM

1. Go to your VM and select Access Control (IAM).



Chaos-Studio-test | Access control (IAM)
Virtual machine

My access
View my level of access to this resource.

Check access
Review the level of access a user, group, ser managed identity has to this resource. Lear

Find ⓘ

User, group, or service principal

Search by name or email address

Fig. 10 Quickstart-access-control

2. Select Add.

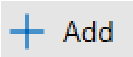


Fig. 11 Add

3. Select Add role assignment.

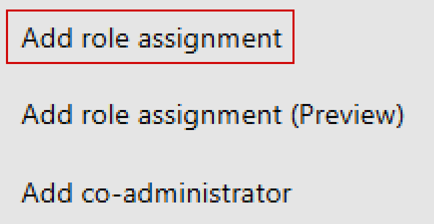


Fig. 12 Add-role-assignment

4. Search for Virtual Machine Contributor and select the role. Select Next.

Add role assignment

Fig. 13 Quickstart-virtual-machine-contributor

5. Select the Managed Identity option
6. Choose Select members and search for your experiment name. Select your experiment and choose Select.

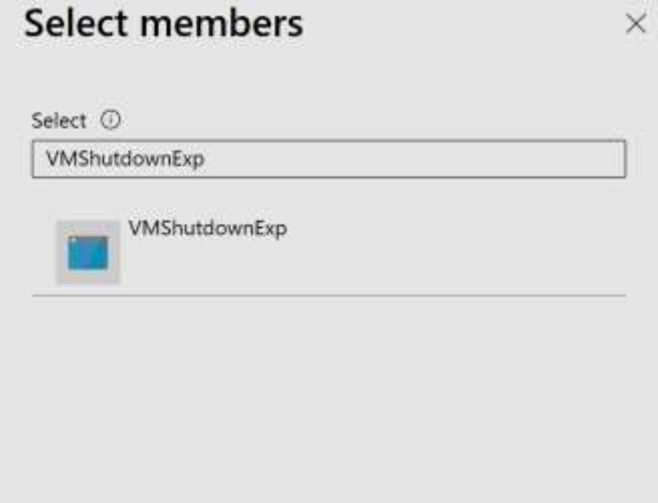


Fig. 14 Quickstart-select-experiment-role-assignment

7. Select Review + Assign.

5.4. Run the chaos experiment

1. Open the Azure portal:

- If you're using an @microsoft.com account, go to this website.
- If you're using an external account, go to this website.

2. Select the checkbox next to the experiment name and select Start Experiment.

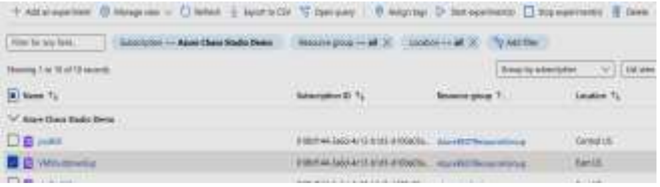
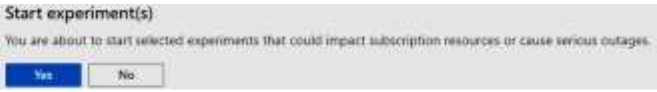


Fig. 15 Quickstart-experiment-start

3. Select Yes to confirm you want to start the chaos experiment.


Start experiment(s)

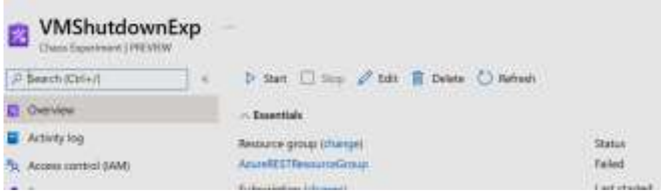
You are about to start selected experiments that could impact subscription resources or cause serious outages.

Fig. 16 Start-experiment-confirmation

4. (Optional) Select the experiment name to see a detailed view of the execution status of the experiment.

5.5. Clean up Resources

1. Select the checkbox next to the experiment name and select Delete.



Resource group (change): AzureRESTResourceGroup

Status: Failed

Fig. 17 Quickstart-delete-experiment

2. Select Yes to confirm that you want to delete the experiment.
3. Search the VM that you created on the Azure portal search bar.

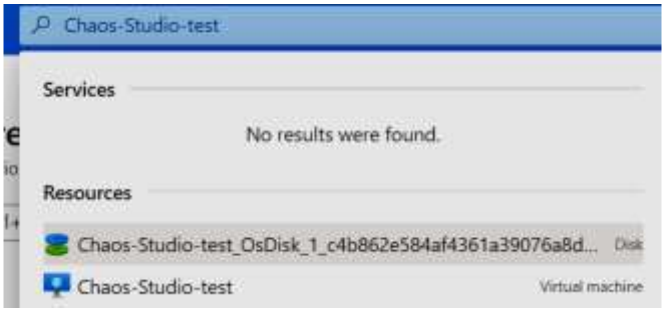
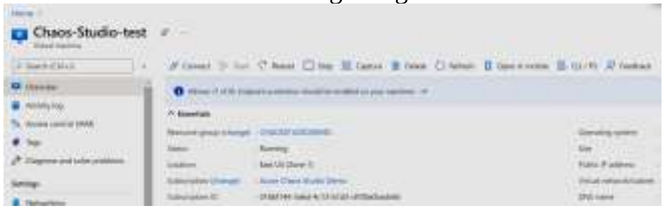


Fig. 18 Quickstart-cleanup

4. Select Delete to avoid being charged for the resource.



Status: Running

Subscription (change): Azure Chaos Studio Demo

Fig. 19 Quickstart-cleanup-virtual-machine

6. Impact

Chaos engineering and site reliability engineering (SRE) are closely related practices that aim to improve the reliability and resilience of cloud applications. Here's how chaos engineering impacts.

6.1. Improved Resilience

Chaos engineering helps identify weaknesses and vulnerabilities in cloud applications by intentionally introducing failures. By doing so, SRE teams can proactively. Address these issues, making the system more resilient to real-world failures.

6.2. Risk Mitigation

Chaos engineering allows SRE teams to assess the impact of potential. Failures in a controlled environment help them understand and mitigate risks before they impact. Users or business operations.

6.3. Continuous Improvement

Chaos engineering promotes a culture of continuous improvement within SRE teams. By regularly running chaos experiments and analyzing the results, teams can. Iteratively improve the reliability and performance of cloud applications over time.

6.4. Faster Incident Response

Through chaos engineering, SRE teams gain a deeper understanding of how their systems behave under stress and failure conditions. This knowledge enables them to respond more quickly and effectively to incidents, minimizing downtime and service disruptions.

6.5. Optimized Resource Allocation

By identifying and addressing weaknesses in cloud applications, chaos engineering helps SRE teams optimize resource allocation and infrastructure provisioning. This ensures that resources are allocated efficiently, leading to cost savings and improved performance.

6.6. Cultural Shift

Chaos Engineering fosters a culture of resilience and accountability within SRE teams. By embracing failure as a learning opportunity and actively seeking out weaknesses in their systems, teams can better prepare for unexpected events and adapt to changing circumstances. Overall, chaos engineering complements SRE practices by providing a systematic approach to identifying and addressing reliability and resilience challenges in cloud applications. By incorporating chaos engineering into their workflows, SRE teams can build more robust and dependable systems that meet the needs of their users and stakeholders.

7. Conclusion

In an era of increasingly complex and distributed cloud applications, chaos engineering emerges as a powerful tool for ensuring reliability and resilience. By intentionally introducing controlled disruptions into cloud applications, organizations can identify weaknesses, optimize resource utilization, and enhance incident response preparedness. When combined with the principles and practices of Site Reliability Engineering, chaos engineering becomes an integral part of building and maintaining reliable, scalable, and resilient systems in the cloud. As organizations continue to embrace the cloud, harnessing chaos engineering will be essential for staying ahead in an ever-changing landscape of technology and innovation.

References

[1] Principles of Chaos Engineering, Principlesofchaos, 2019. [Online]. Available: http://principlesofchaos.org/?lang=ENcontent

[2] Resilience, Cambridge Dictionary, 2020. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/resilience

[3] Russ Miles, *Chaos Engineering Observability*, O'Reilly Media, 2019. [Google Scholar] [Publisher Link]

[4] Betsy Beyer et al., *Site Reliability Engineering: How Google Runs Production Systems*, O'Reilly Media, 2016. [Google Scholar] [Publisher Link]

[5] Chaos Engineering, AWS Solutions Library. [Online]. Available: https://aws.amazon.com/solutions/resilience/chaos-engineering/

[6] Find and Fix Your Reliability Risks, Gremlin. [Online]. Available: https://www.gremlin.com/

[7] Open Source Chaos Engineering Platform, LitmusChaos. [Online]. Available: https://litmuschaos.io/

[8] Ali Basiri et al., "Chaos Engineering," *IEEE Software*, vol. 33, no. 3, pp. 35-41, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ali Basiri et al., "Automating Chaos Experiments in Production," *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice*, Montreal, QC, Canada, pp. 31-40, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Quickstart: Create and Run a Chaos Experiment by Using Azure Chaos Studio, Microsoft, pp. 1-287, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/chaos-studio/chaos-studio-quickstart-azure-portal