Chaos Engineering with Azure OpenAI: Building the Unbreakable

By: Nicholas Chang & Peter de tender

Who is Nicholas Chang

Senior Platform Engineer at Kainos and Microsoft Azure MVP

Blogger, Speaker and enjoying learning about technology

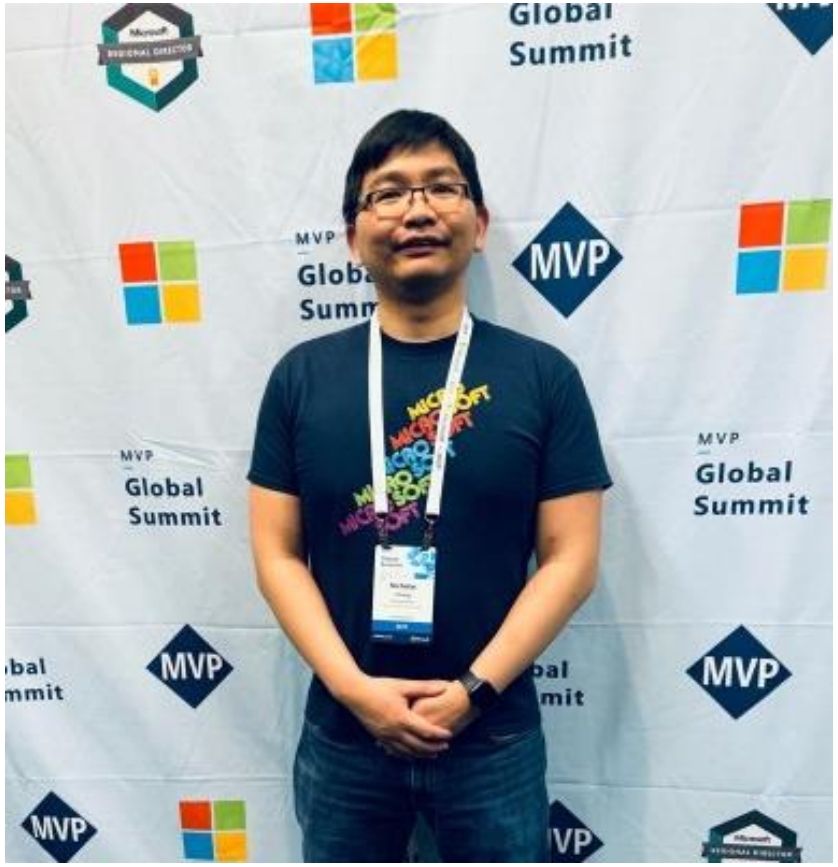
Communities

Organizer of Microsoft Azure Community User Group

Organizer of Microsoft Community Insights Podcast

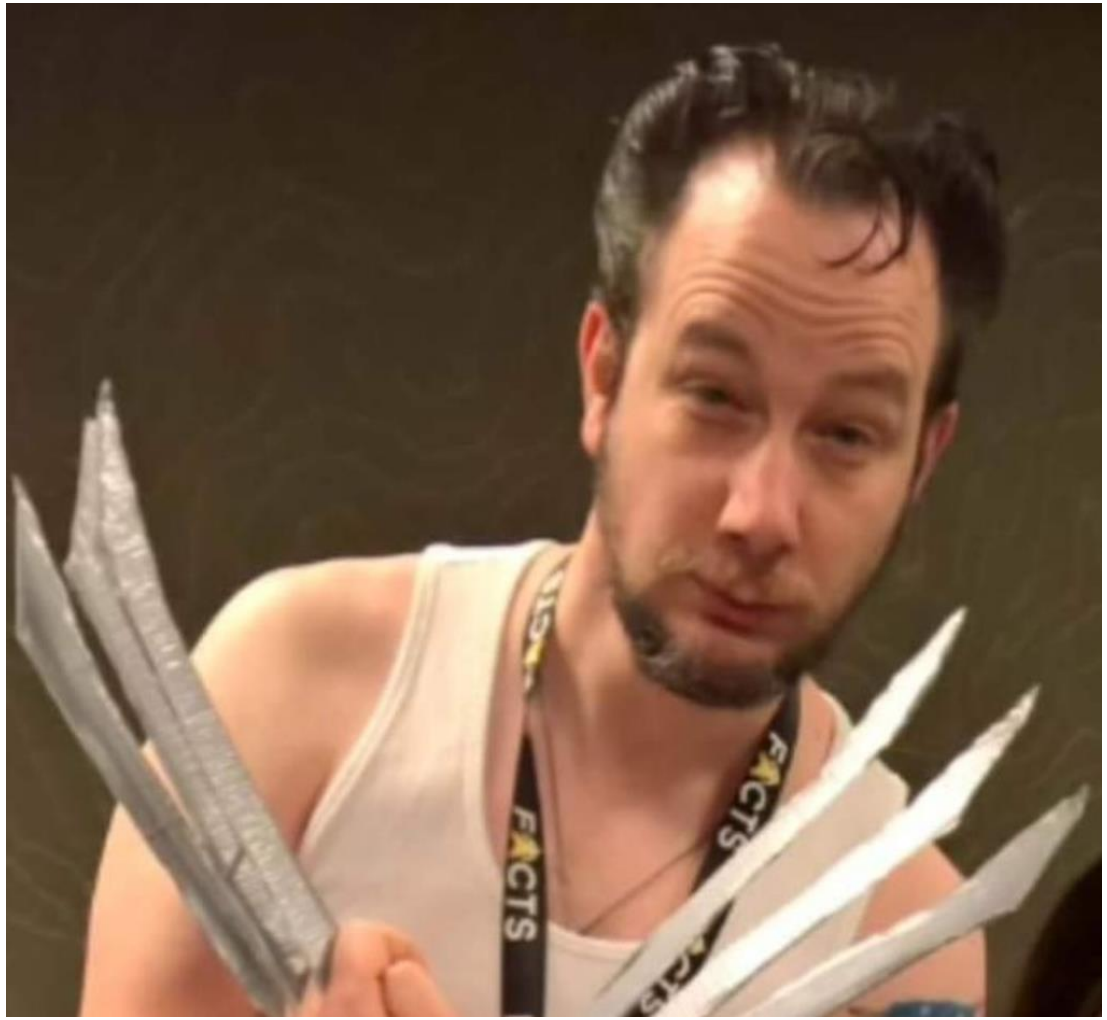
Blog
https://nicholaschangblog.com

Social Media:
https://linkfree.io/NickAzureDevops



Who is Peter De Tender

- **MARVEL** comics fan since mid 80's
- **Microsoft Technical Trainer**, recently relocated to Redmond, WA from Belgium
- **Azure Architecture and DevOps** focused, background in Microsoft Datacenter deployments, shifted to cloud since 2012, and loving it
- **Technical writer, book publisher, blogger,…**



Thank You

AZURE
BACK TO
SCHOOL

By: Derek Smith and Dwayne Natwick

Agenda

- What is chaos engineering
- Integrating Chaos Engineering with Azure OpenAI
- Implementing Chaos Experiments
- Demo
- Monitoring
- Conclusion

What is Chaos Engineering

Intentionally introduce faults to cause system components failure to improve resilience and availability. Compared to DevOps and SRE, Chaos Engineering helps obtain consistent reliability by hardening services

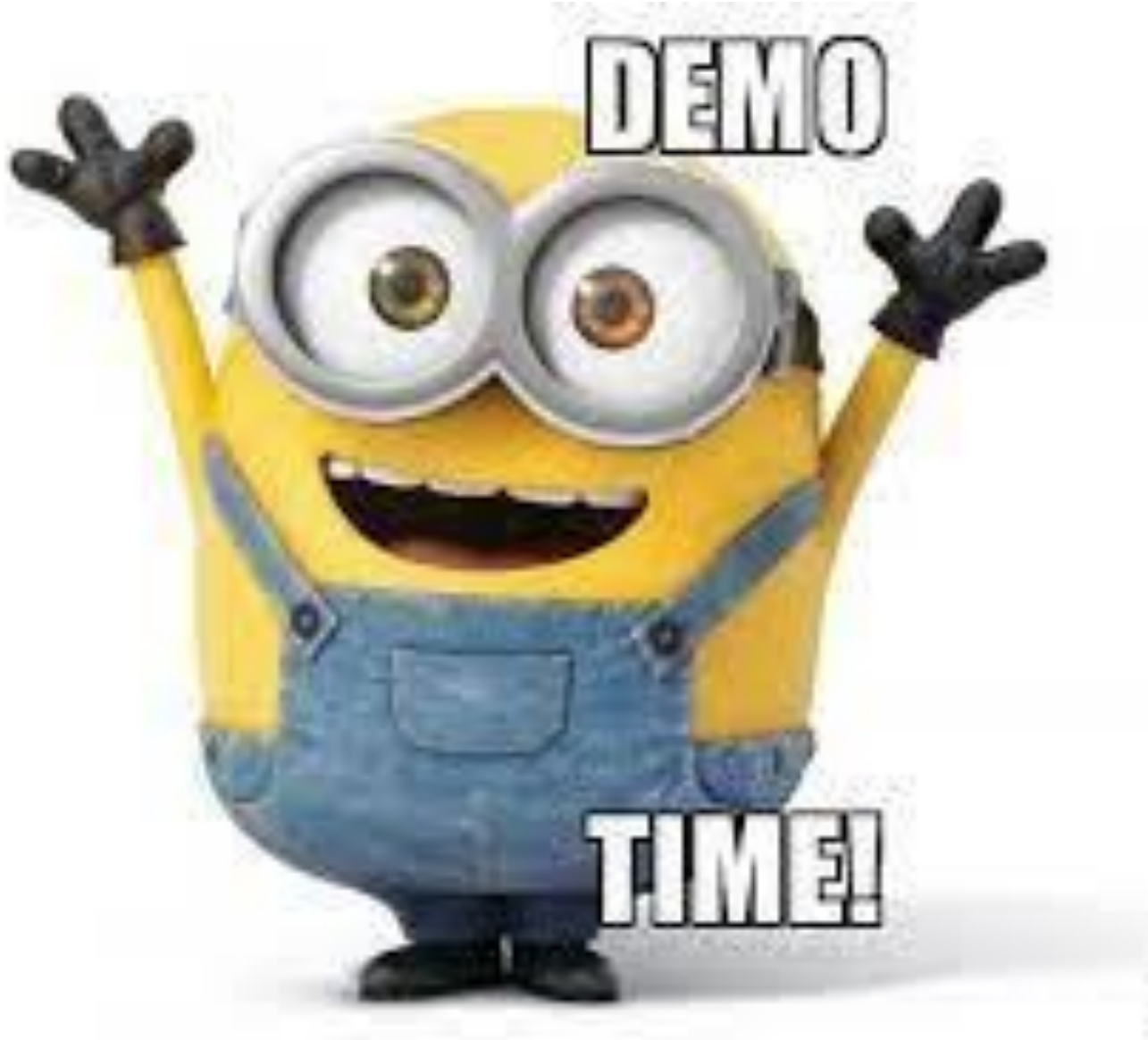
- Improve system resilience to failure and outage.
- Reduces downtime
- Identify any “What if’s questions.
- Using faulting injection
- Bombing Production to make them more reliable

Integrating Chaos Engineering with Azure OpenAI

- **Design**
  - Azure OpenAI can assist in generating realistic scenarios for failure injection.
  - It can create complex and varied failure patterns based on historical data and system dependencies.

- **Execution**
  - Using Azure OpenAI's automation features, engineers can automatically initiate chaos experiments.
  - Engineers can control the scope and severity of failures.

- **Analysis**
  - Azure OpenAI can analyze the results of chaos experiments.
  - It provides insights into system behavior and identifies areas for improvement.


DEMO
TIME!