Harnessing Chaos Engineering with Azure Chaos Studio

Robert Cazacu - Microsoft

Agenda

1. Resilience and Chaos Engineering
2. Azure Chaos Studio
3. Scenarios, requirements and fault interests
4. Demo
5. Questions

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.”

("Principles of Chaos Engineering". principlesofchaos.org)

Where did it all start?

Steve Capps - "Monkey”
(Copilot generated)

Chaos Monkey Netflix

Resilience in the cloud

Cloud applications must be designed to handle failure.

When running applications in the cloud, resilience becomes a shared responsibility.

Resiliency vs Chaos Testing

Resiliency

- Designed into your service
- Deals with unexpected events in another service it calls
- Build multiple layers of resiliency when needed
- Unit tests, Mocks/proxies and fault injection can test your service in controlled environment

Chaos Testing

- Intentionally introduces faults into the system
- Monitoring how the system responds
- Identify potentially weaknesses
- Runs in production (or production-like) environments

Chaos use cases

Pre-release Validation

Shift left (Test, Stage)

Continuous Production Validation

Shift right (Canary, Production)

- Explore service dependencies in a **controlled environment**
- Gate production code flow with **CI/CD pipeline automation**
- Perform **incident fix validation**
- **Harden** release pipeline
- **Certify** new hardware

- Perform **BCDR Drills**
- Host **Game Days**
- Simulate Availability Zone or Region outages
- Use for **Error Budget testing**
- Past incident regression testing
- Validate on call and **livesite process**

Chaos experiments

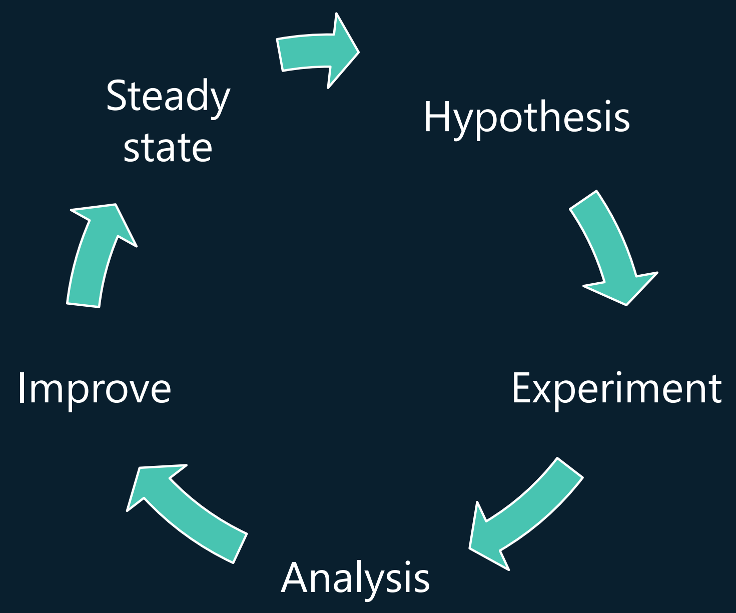
Hypothesis
What is being validated? What are possible outcomes?

Experiment
Orchestrated execution of workload + faults against subscription resource targets.

Analysis
Baseline performance, monitoring telemetry, recovery time.

Improvement
New code, code changes.
Process.

Steady state
Continuous production monitoring + validation.



Azure Chaos Studio

Fully managed service for building resilience into your Azure services

Deeply integrated into Azure, including Azure Resource Manager, Azure Monitor, and Azure Active Directory.

Expanding library of faults for common Azure service issues

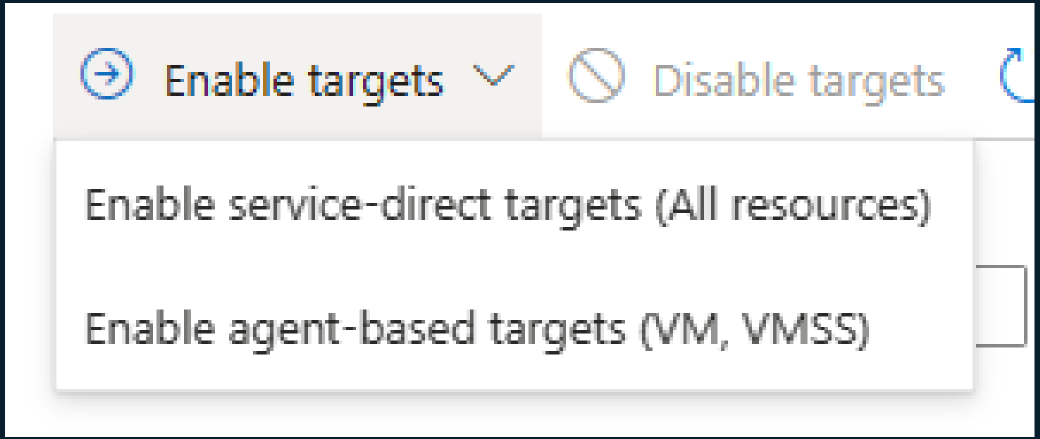
Simulation of real-world scenarios using orchestrated parallel and sequential fault injection

Stop and roll back experiments to safeguard fault injection from causing outages

How Chaos Studio works

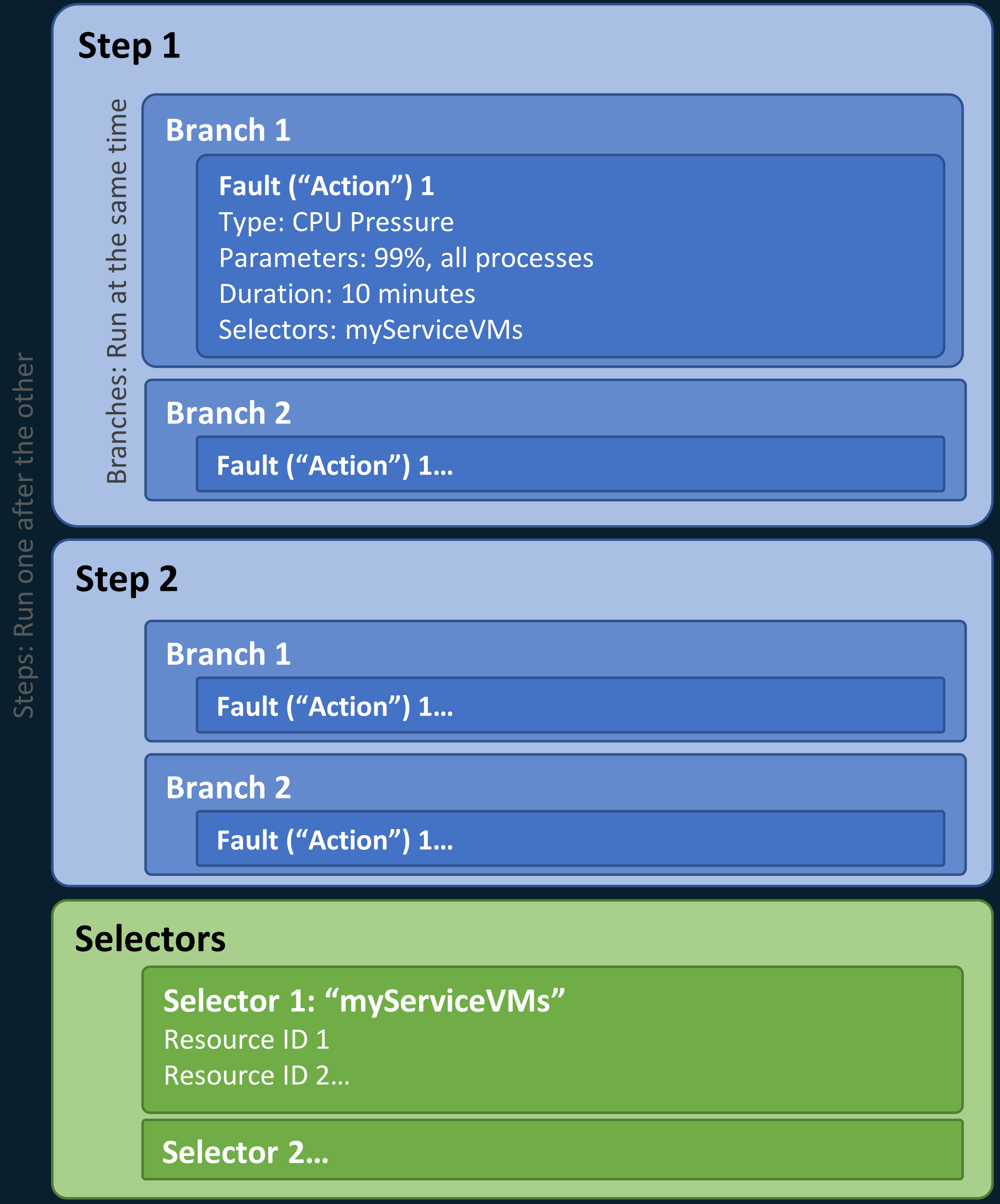
Service-direct: These faults run directly against an Azure resource

Agent-based: These faults run in VMs or virtual machine scale sets to do in-guest failures


Enable targets
Disable targets
Enable service-direct targets (All resources)
Enable agent-based targets (VM, VMSS)

Layout of Chaos Experiments

- Steps
- Branches
- Selectors



Steps: Run one after the other

Step 1

Branches: Run at the same time

Branch 1

Fault ("Action") 1
Type: CPU Pressure
Parameters: 99%, all processes
Duration: 10 minutes
Selectors: myServiceVMs

Branch 2

Fault ("Action") 1…

Step 2

Branch 1

Fault ("Action") 1…

Branch 2

Fault ("Action") 1…

Selectors

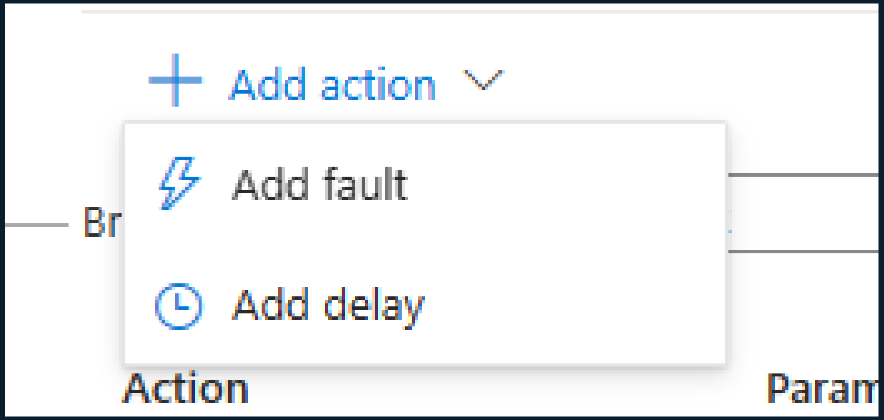
Selector 1: "myServiceVMs"
Resource ID 1
Resource ID 2…

Selector 2…

Experiment actions

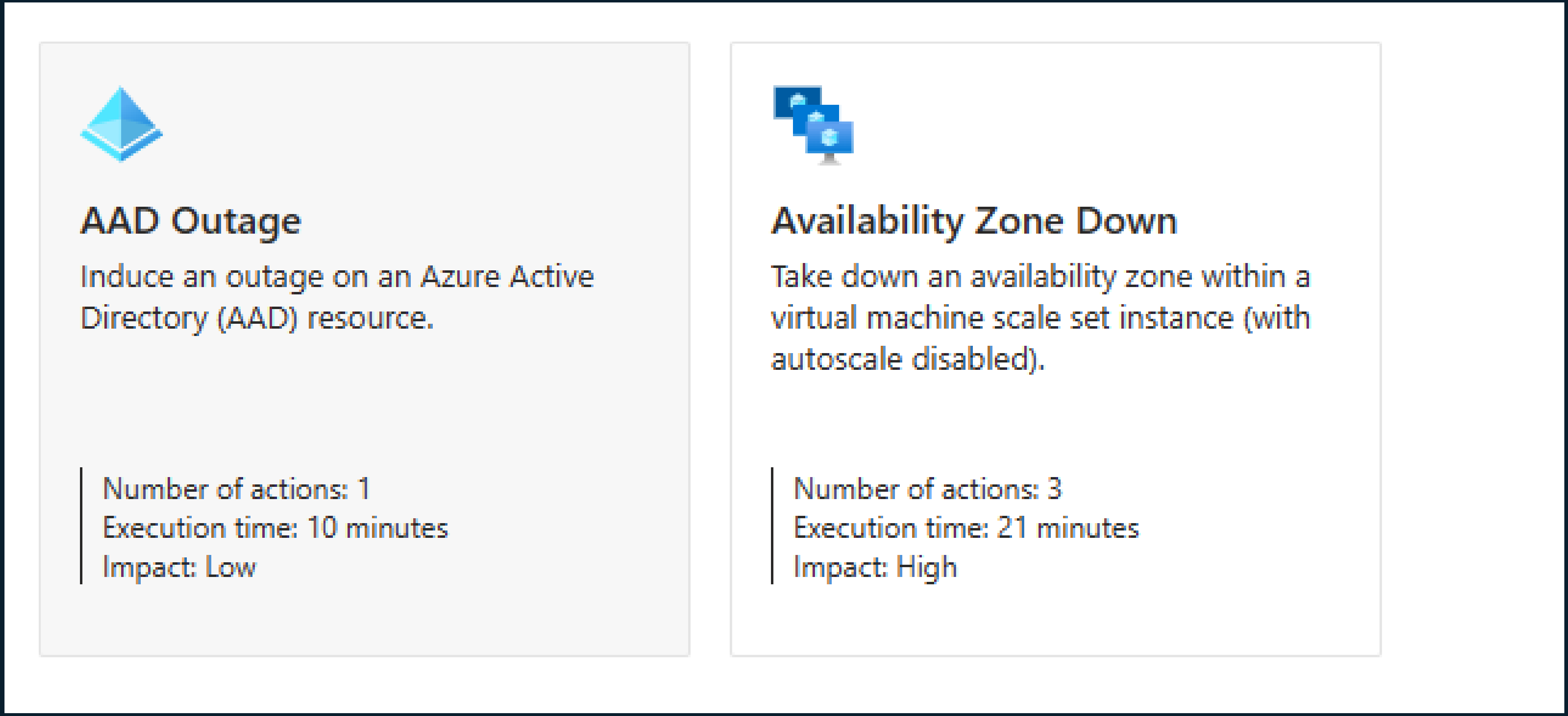
Faults: This action causes a disruption in one or more resources.

Time delays: This action "waits" without affecting any resource



Experiments

Experiment templates



AAD Outage

Induce an outage on an Azure Active Directory (AAD) resource.

Number of actions: 1
Execution time: 10 minutes
Impact: Low

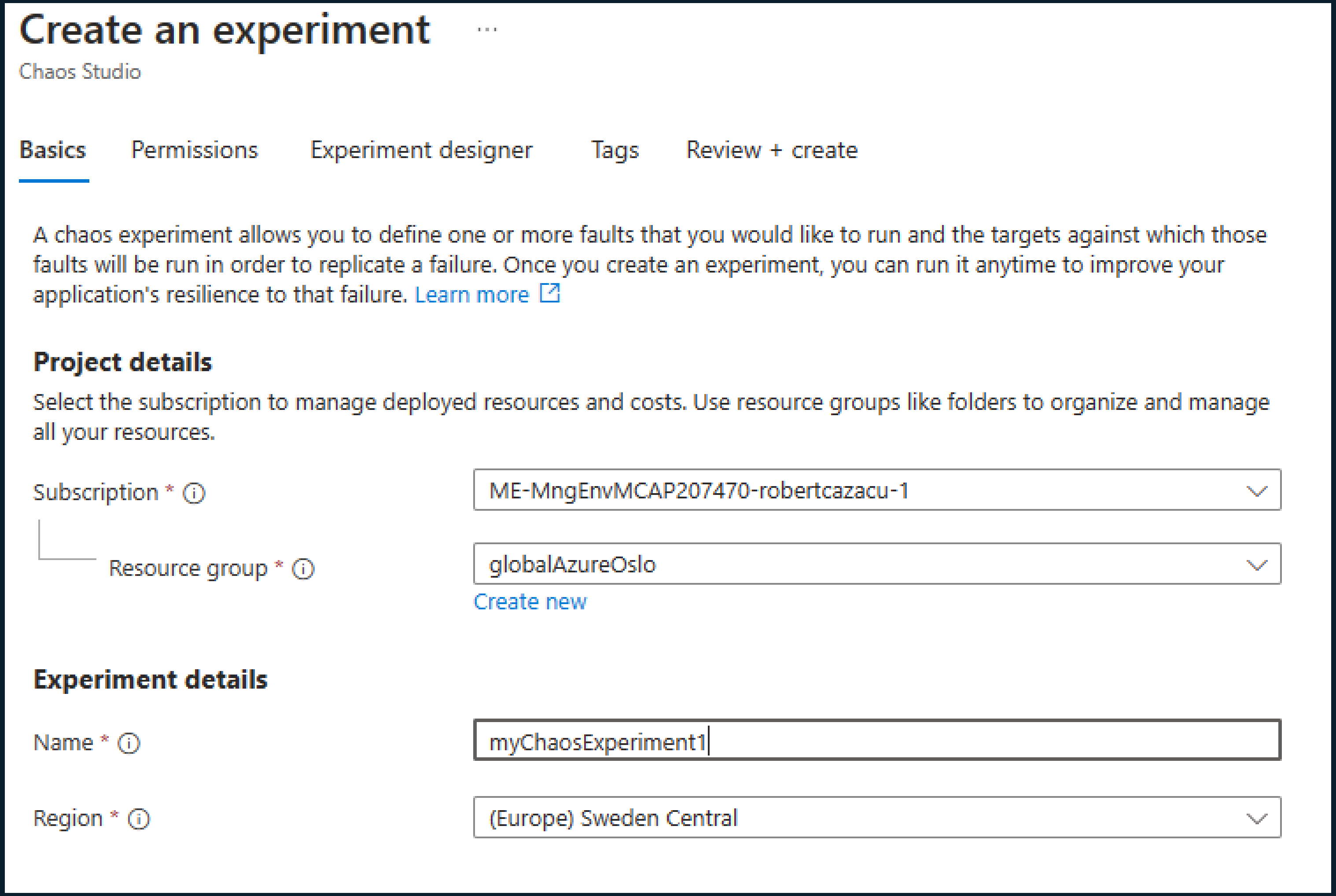
Availability Zone Down

Take down an availability zone within a virtual machine scale set instance (with autoscale disabled).

Number of actions: 3
Execution time: 21 minutes
Impact: High

Experiments

Create experiments from scratch



A chaos experiment allows you to define one or more faults that you would like to run and the targets against which those faults will be run in order to replicate a failure. Once you create an experiment, you can run it anytime to improve your application's resilience to that failure. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ: ME-MngEnvMCAP207470-robertcazacu-1

Resource group * ⓘ: globalAzureOslo

Create new

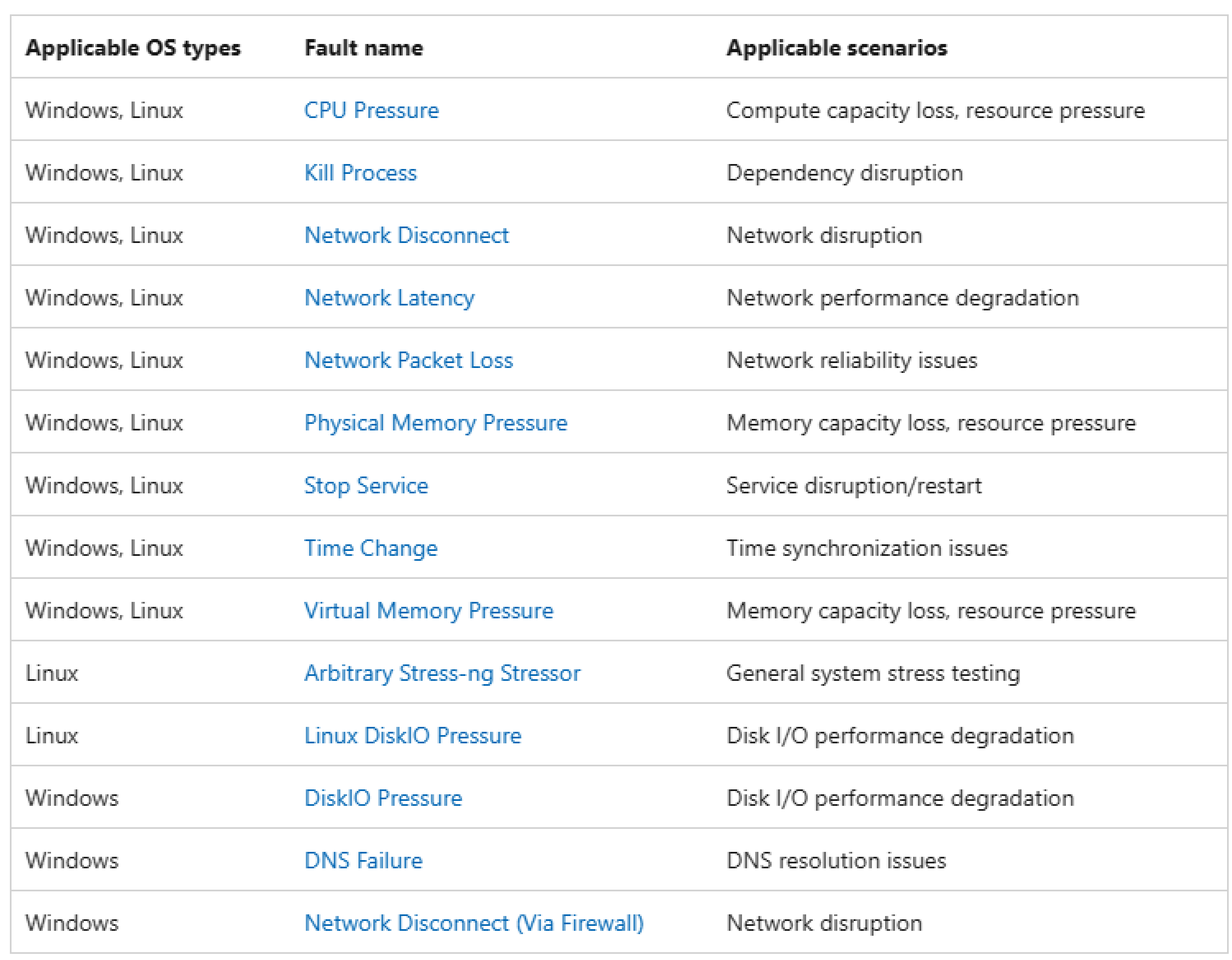
Experiment details

Name * ⓘ: myChaosExperiment1

Region * ⓘ: (Europe) Sweden Central

Fault library

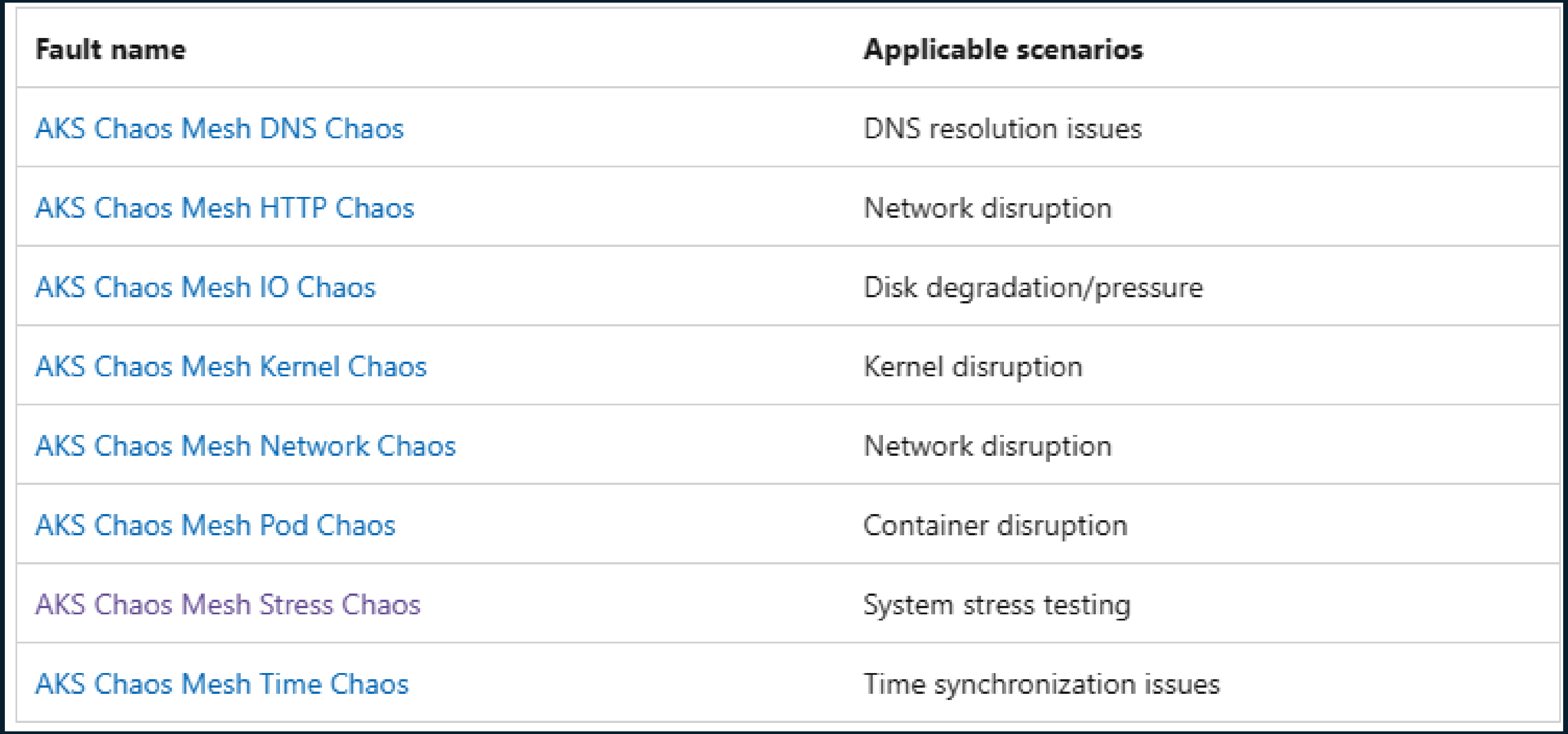
Agent based faults **VMs and VMSS**



Applicable OS types	Fault name	Applicable scenarios
Windows, Linux	CPU Pressure	Compute capacity loss, resource pressure
Windows, Linux	Kill Process	Dependency disruption
Windows, Linux	Network Disconnect	Network disruption
Windows, Linux	Network Latency	Network performance degradation
Windows, Linux	Network Packet Loss	Network reliability issues
Windows, Linux	Physical Memory Pressure	Memory capacity loss, resource pressure
Windows, Linux	Stop Service	Service disruption/restart
Windows, Linux	Time Change	Time synchronization issues
Windows, Linux	Virtual Memory Pressure	Memory capacity loss, resource pressure
Linux	Arbitrary Stress-ng Stressor	General system stress testing
Linux	Linux DiskIO Pressure	Disk I/O performance degradation
Windows	DiskIO Pressure	Disk I/O performance degradation
Windows	DNS Failure	DNS resolution issues
Windows	Network Disconnect (Via Firewall)	Network disruption

Fault library

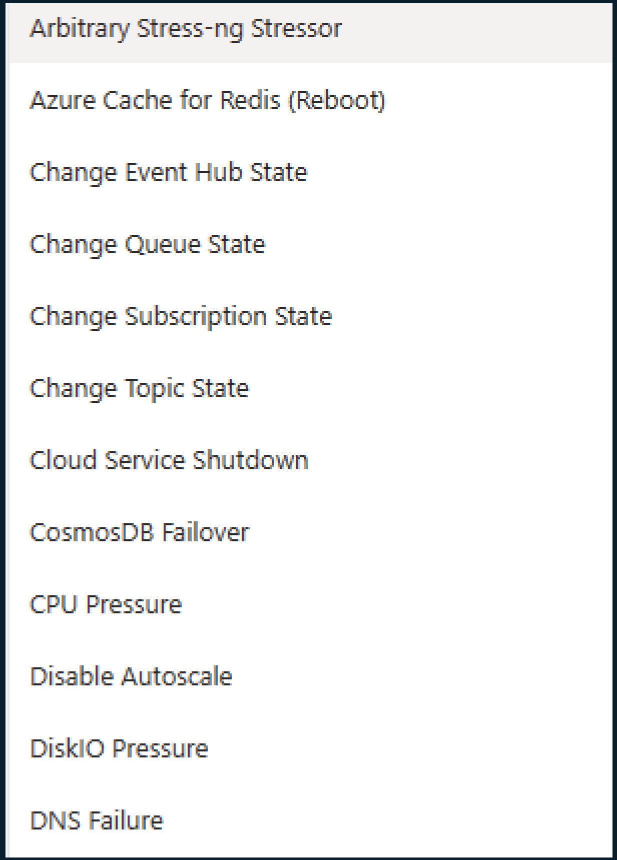
Azure Kubernetes Service: uses **Chaos Mesh**



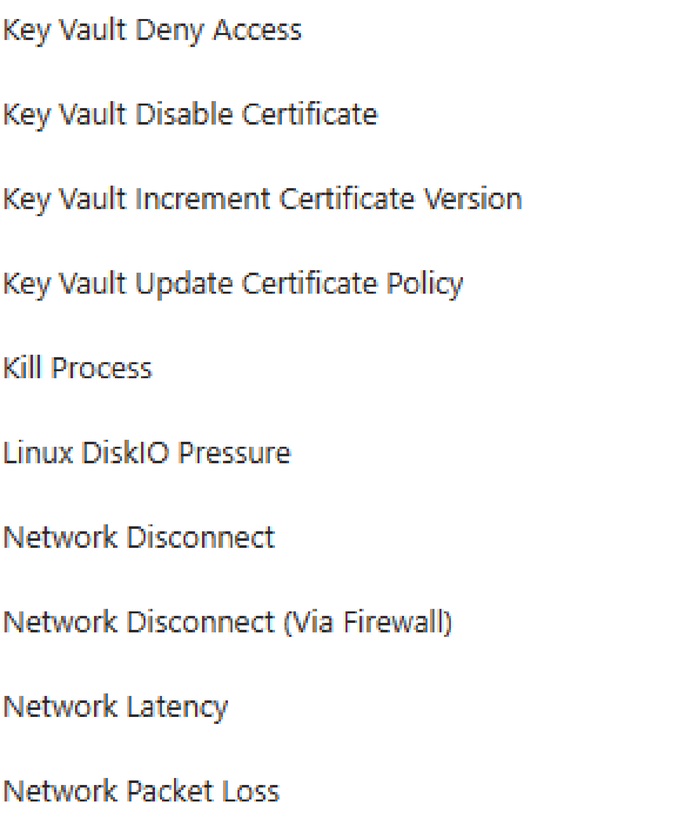
Fault name	**Applicable scenarios**
AKS Chaos Mesh DNS Chaos	DNS resolution issues
AKS Chaos Mesh HTTP Chaos	Network disruption
AKS Chaos Mesh IO Chaos	Disk degradation/pressure
AKS Chaos Mesh Kernel Chaos	Kernel disruption
AKS Chaos Mesh Network Chaos	Network disruption
AKS Chaos Mesh Pod Chaos	Container disruption
AKS Chaos Mesh Stress Chaos	System stress testing
AKS Chaos Mesh Time Chaos	Time synchronization issues

Fault library

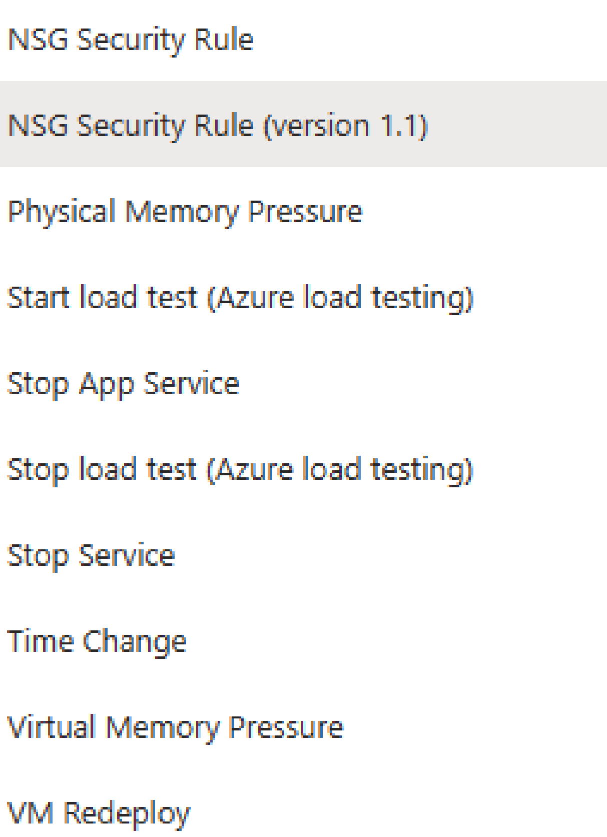
And much more



- Arbitrary Stress-ng Stressor
- Azure Cache for Redis (Reboot)
- Change Event Hub State
- Change Queue State
- Change Subscription State
- Change Topic State
- Cloud Service Shutdown
- CosmosDB Failover
- CPU Pressure
- Disable Autoscale
- DiskIO Pressure
- DNS Failure

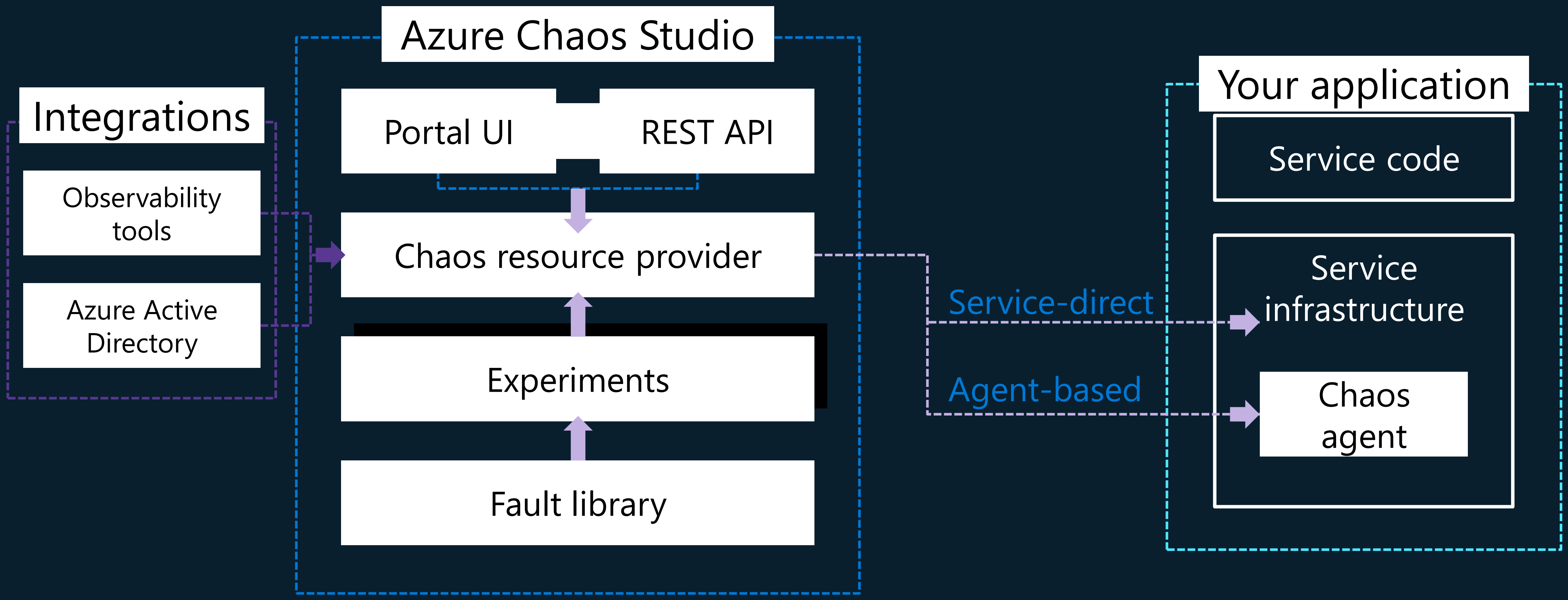


- Key Vault Deny Access
- Key Vault Disable Certificate
- Key Vault Increment Certificate Version
- Key Vault Update Certificate Policy
- Kill Process
- Linux DiskIO Pressure
- Network Disconnect
- Network Disconnect (Via Firewall)
- Network Latency
- Network Packet Loss

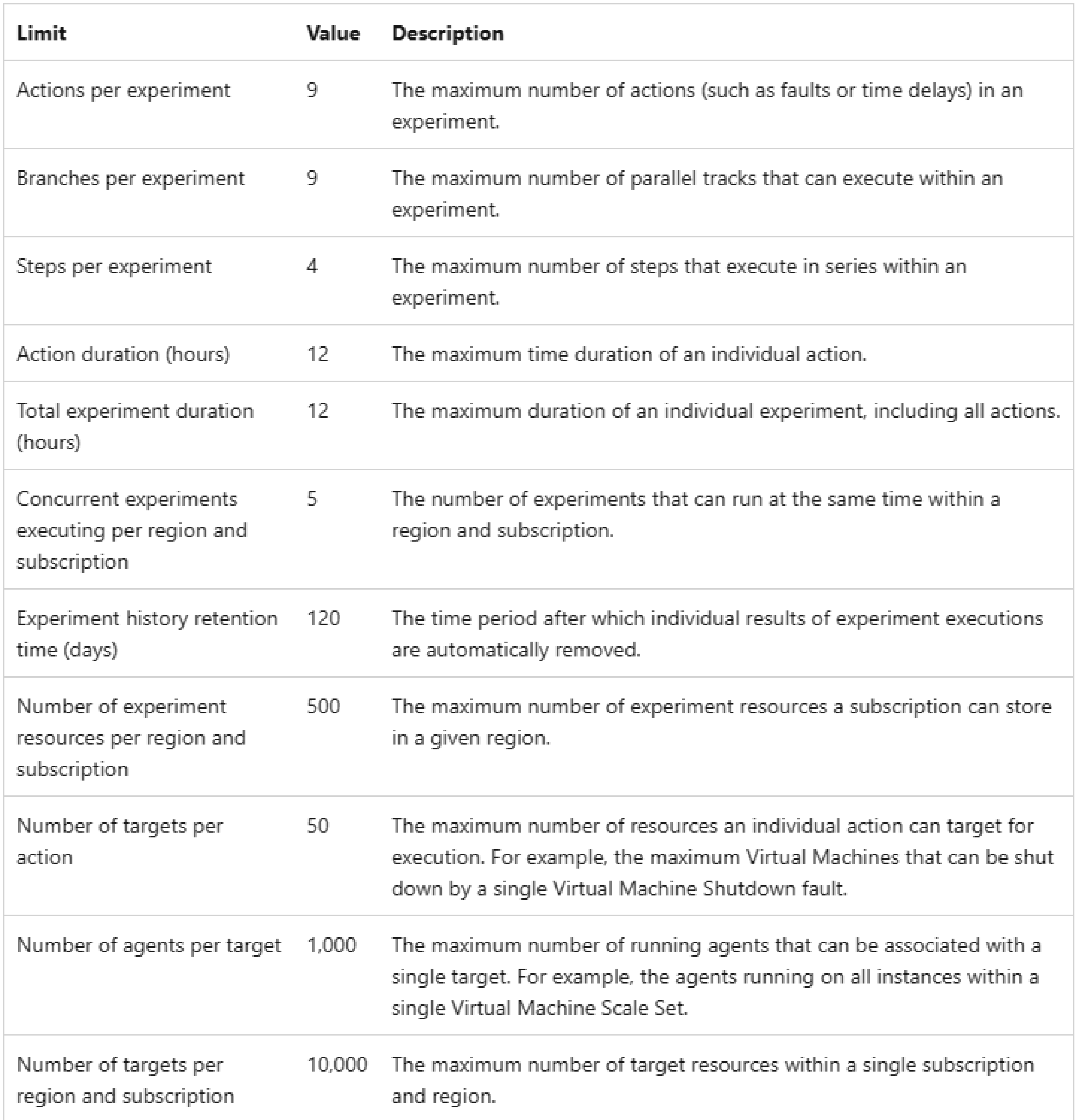


- NSG Security Rule
- NSG Security Rule (version 1.1)
- Physical Memory Pressure
- Start load test (Azure load testing)
- Stop App Service
- Stop load test (Azure load testing)
- Stop Service
- Time Change
- Virtual Memory Pressure
- VM Redeploy

Product Architecture

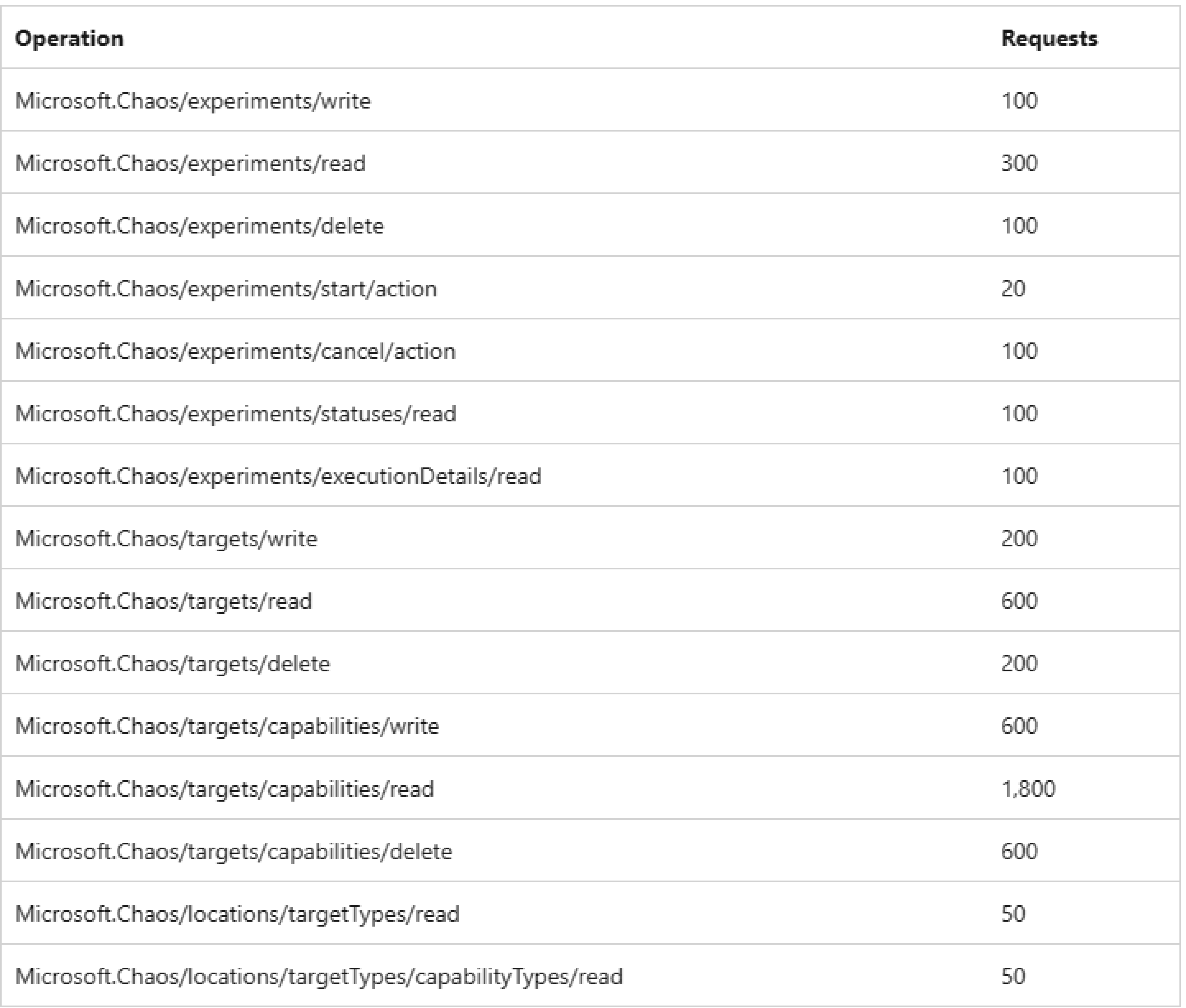


Service Limits



Limit	Value	Description
Actions per experiment	9	The maximum number of actions (such as faults or time delays) in an experiment.
Branches per experiment	9	The maximum number of parallel tracks that can execute within an experiment.
Steps per experiment	4	The maximum number of steps that execute in series within an experiment.
Action duration (hours)	12	The maximum time duration of an individual action.
Total experiment duration (hours)	12	The maximum duration of an individual experiment, including all actions.
Concurrent experiments executing per region and subscription	5	The number of experiments that can run at the same time within a region and subscription.
Experiment history retention time (days)	120	The time period after which individual results of experiment executions are automatically removed.
Number of experiment resources per region and subscription	500	The maximum number of experiment resources a subscription can store in a given region.
Number of targets per action	50	The maximum number of resources an individual action can target for execution. For example, the maximum Virtual Machines that can be shut down by a single Virtual Machine Shutdown fault.
Number of agents per target	1,000	The maximum number of running agents that can be associated with a single target. For example, the agents running on all instances within a single Virtual Machine Scale Set.
Number of targets per region and subscription	10,000	The maximum number of target resources within a single subscription and region.

API throttling limits



Operation	Requests
Microsoft.Chaos/experiments/write	100
Microsoft.Chaos/experiments/read	300
Microsoft.Chaos/experiments/delete	100
Microsoft.Chaos/experiments/start/action	20
Microsoft.Chaos/experiments/cancel/action	100
Microsoft.Chaos/experiments/statuses/read	100
Microsoft.Chaos/experiments/executionDetails/read	100
Microsoft.Chaos/targets/write	200
Microsoft.Chaos/targets/read	600
Microsoft.Chaos/targets/delete	200
Microsoft.Chaos/targets/capabilities/write	600
Microsoft.Chaos/targets/capabilities/read	1,800
Microsoft.Chaos/targets/capabilities/delete	600
Microsoft.Chaos/locations/targetTypes/read	50
Microsoft.Chaos/locations/targetTypes/capabilityTypes/read	50

Demo

Microsoft

Thank you!