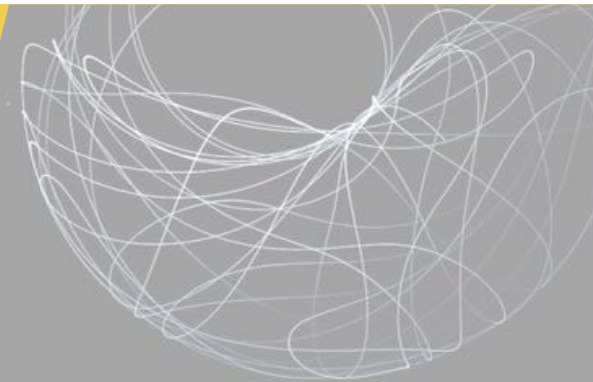


How to Test Your Fault Isolation Boundaries in the Cloud

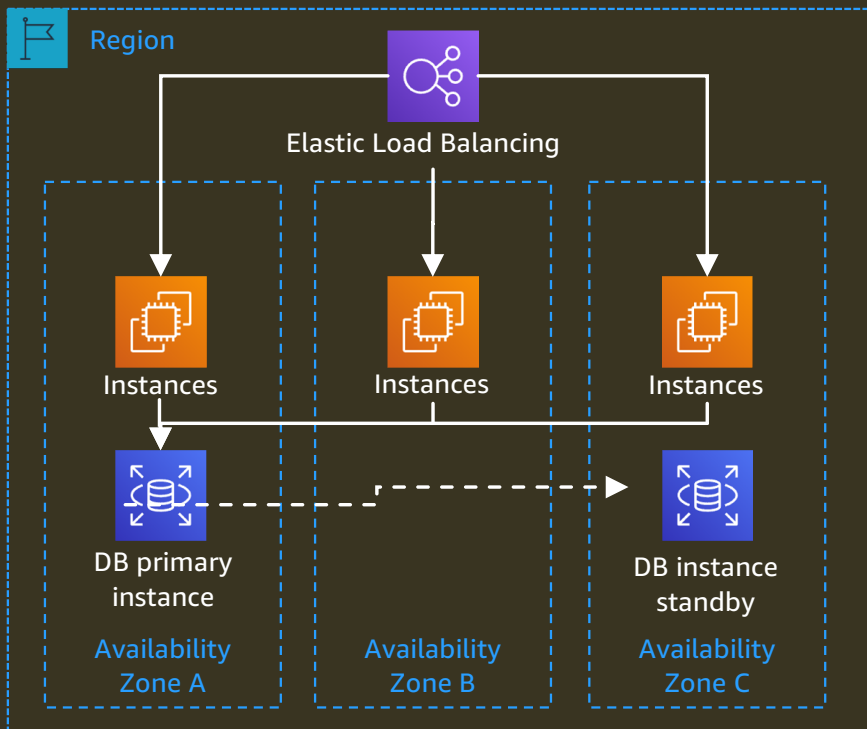
 [jason_barto](#)

A Tale of Two Systems

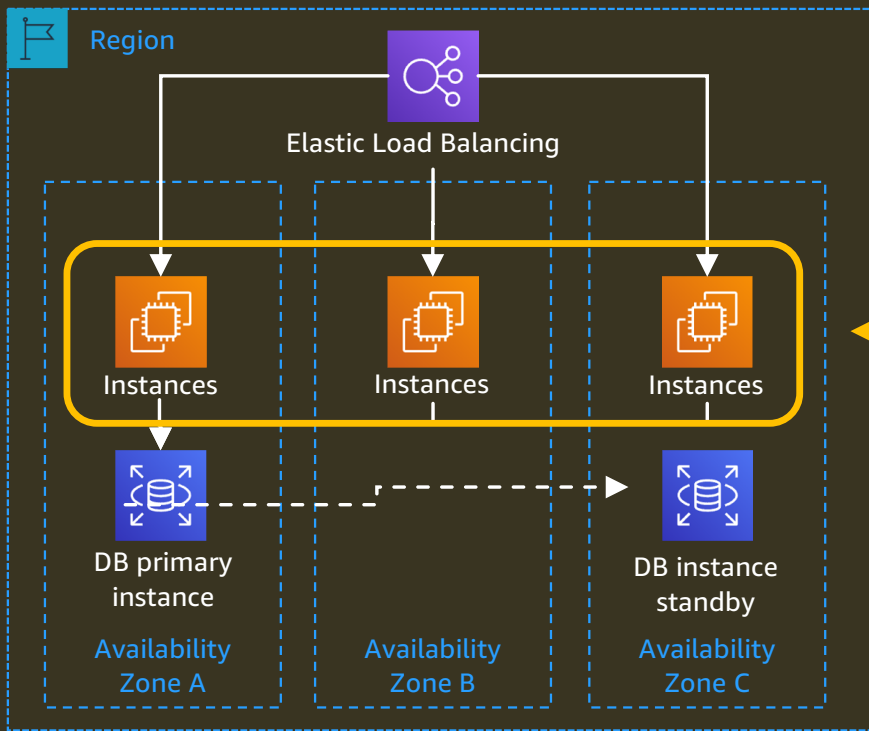
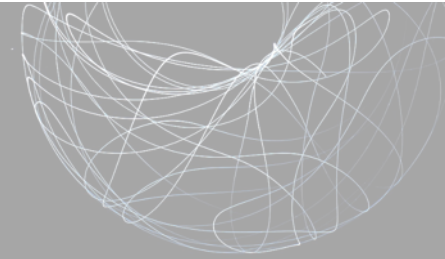
How a resilient banking system failed



A Tale of Two Systems

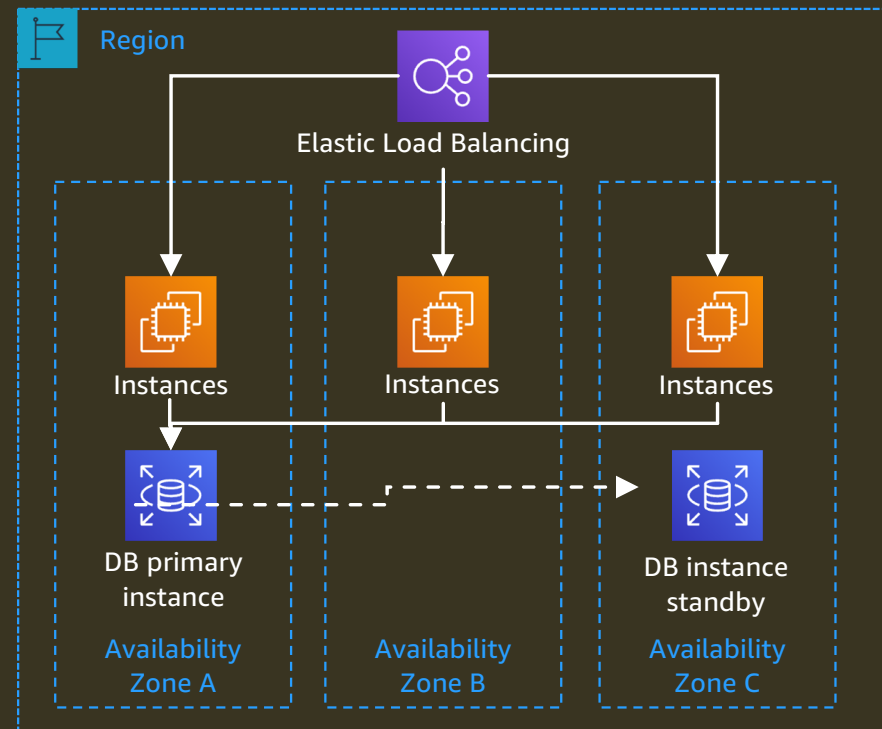
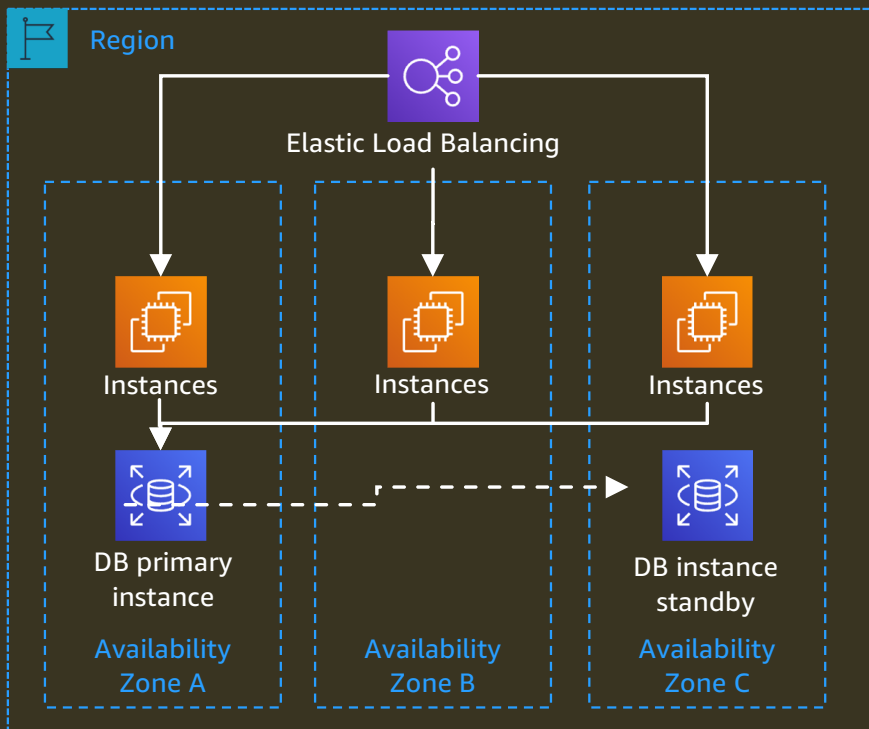
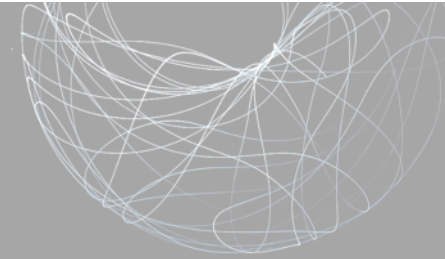


A Tale of Two Systems

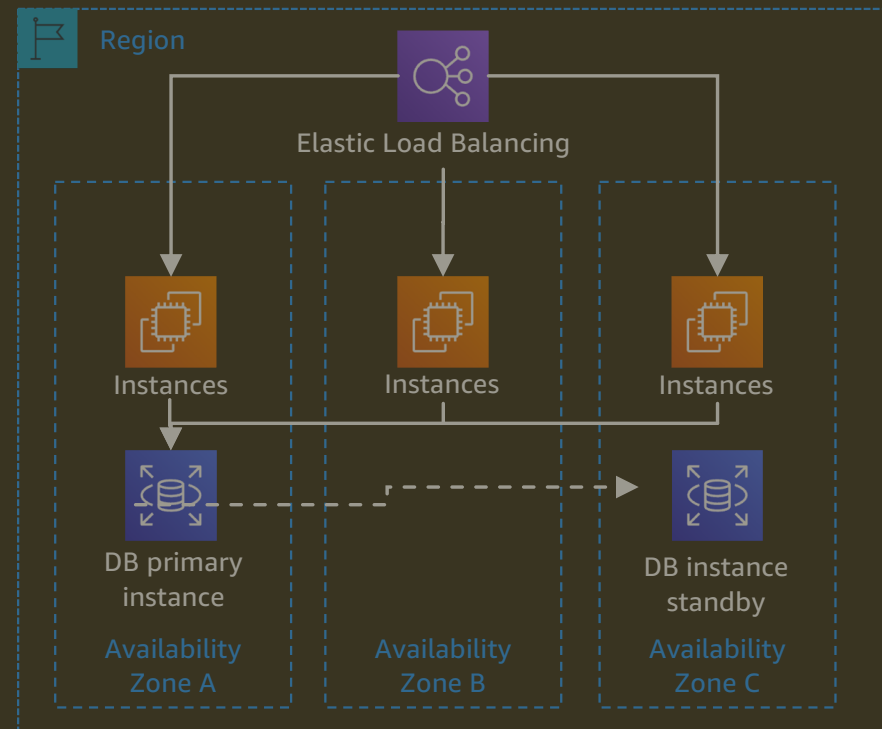
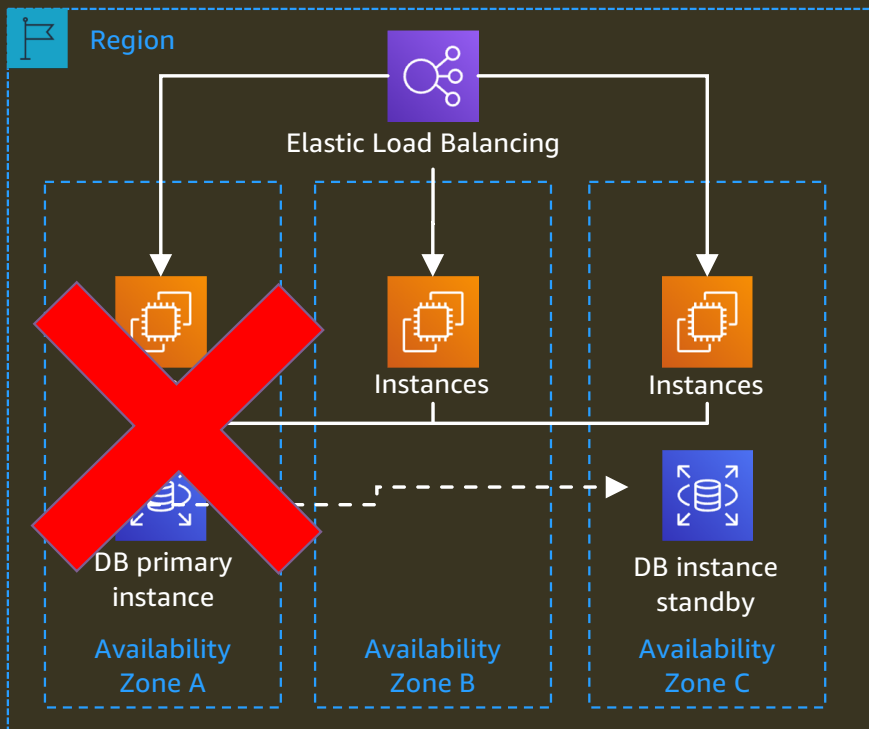
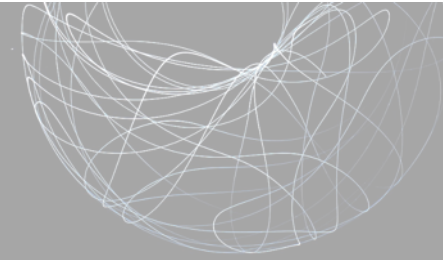


Cluster with quorum

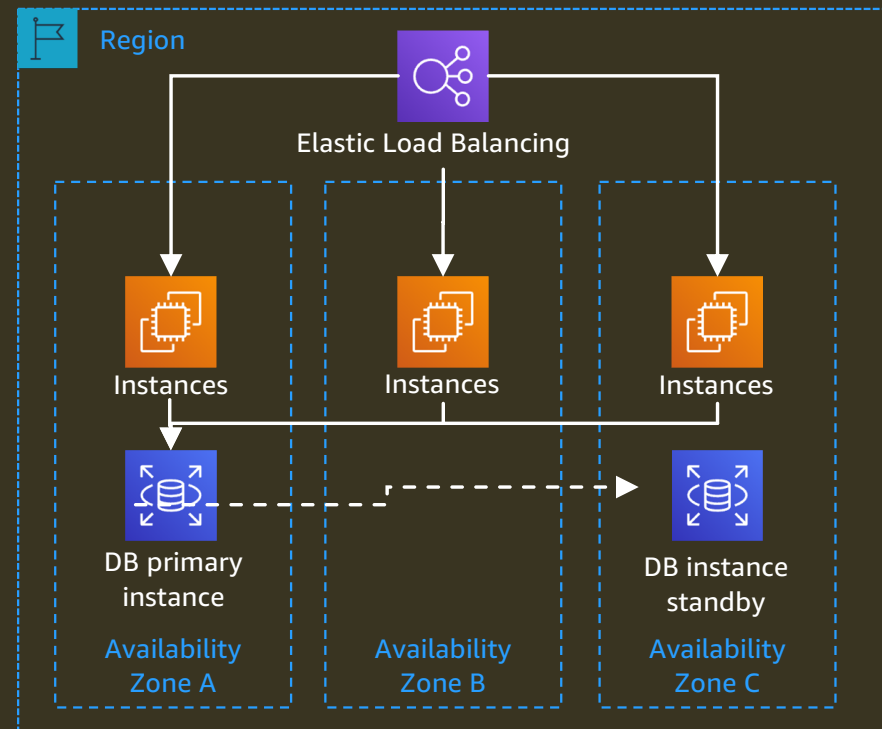
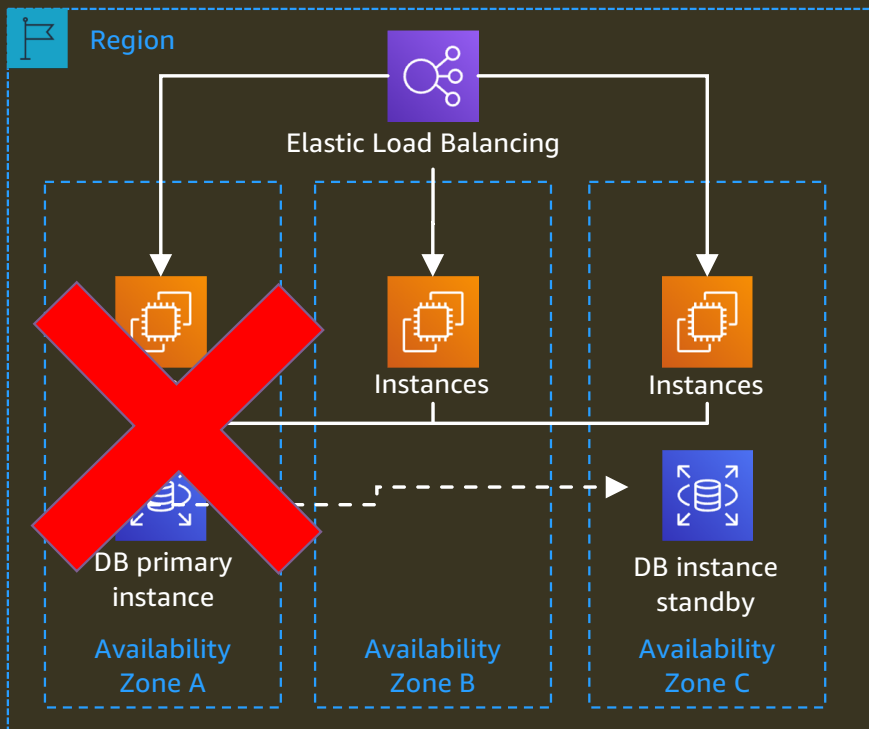
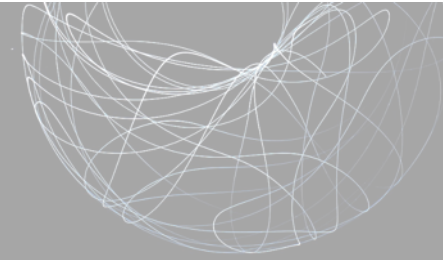
A Tale of Two Systems



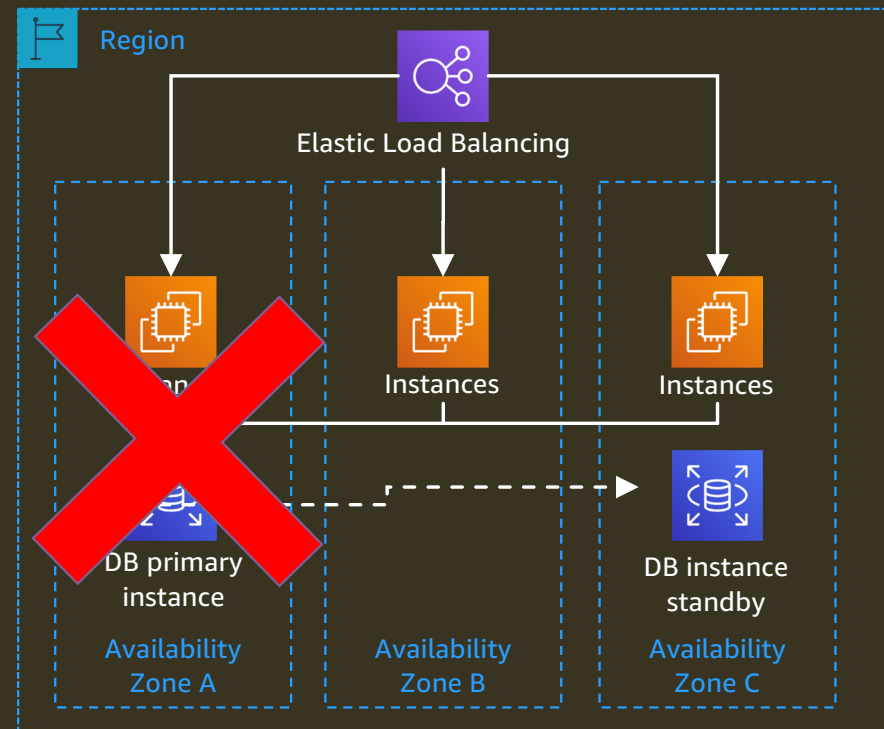
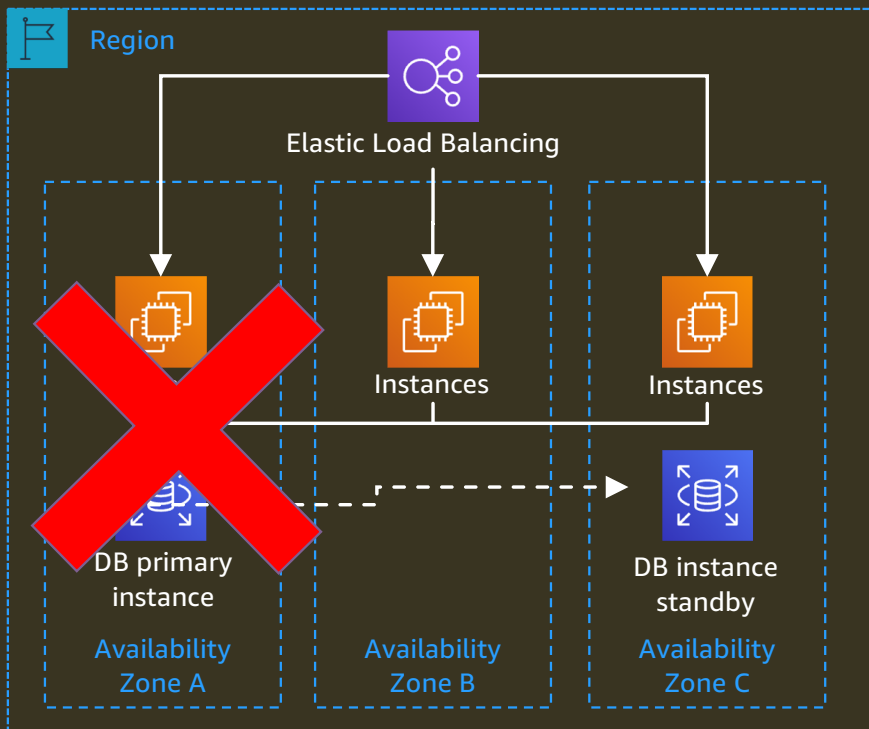
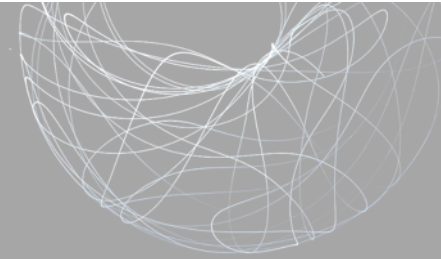
A Tale of Two Systems

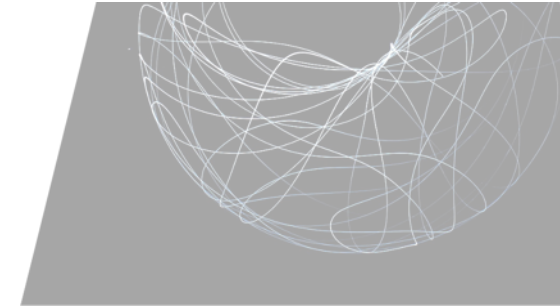


A Tale of Two Systems

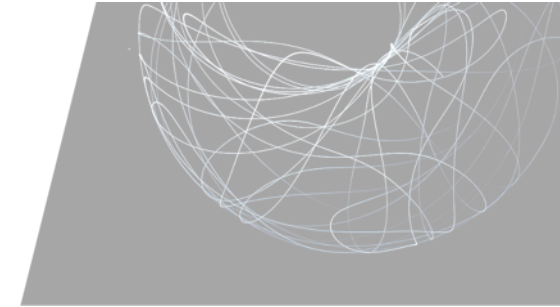
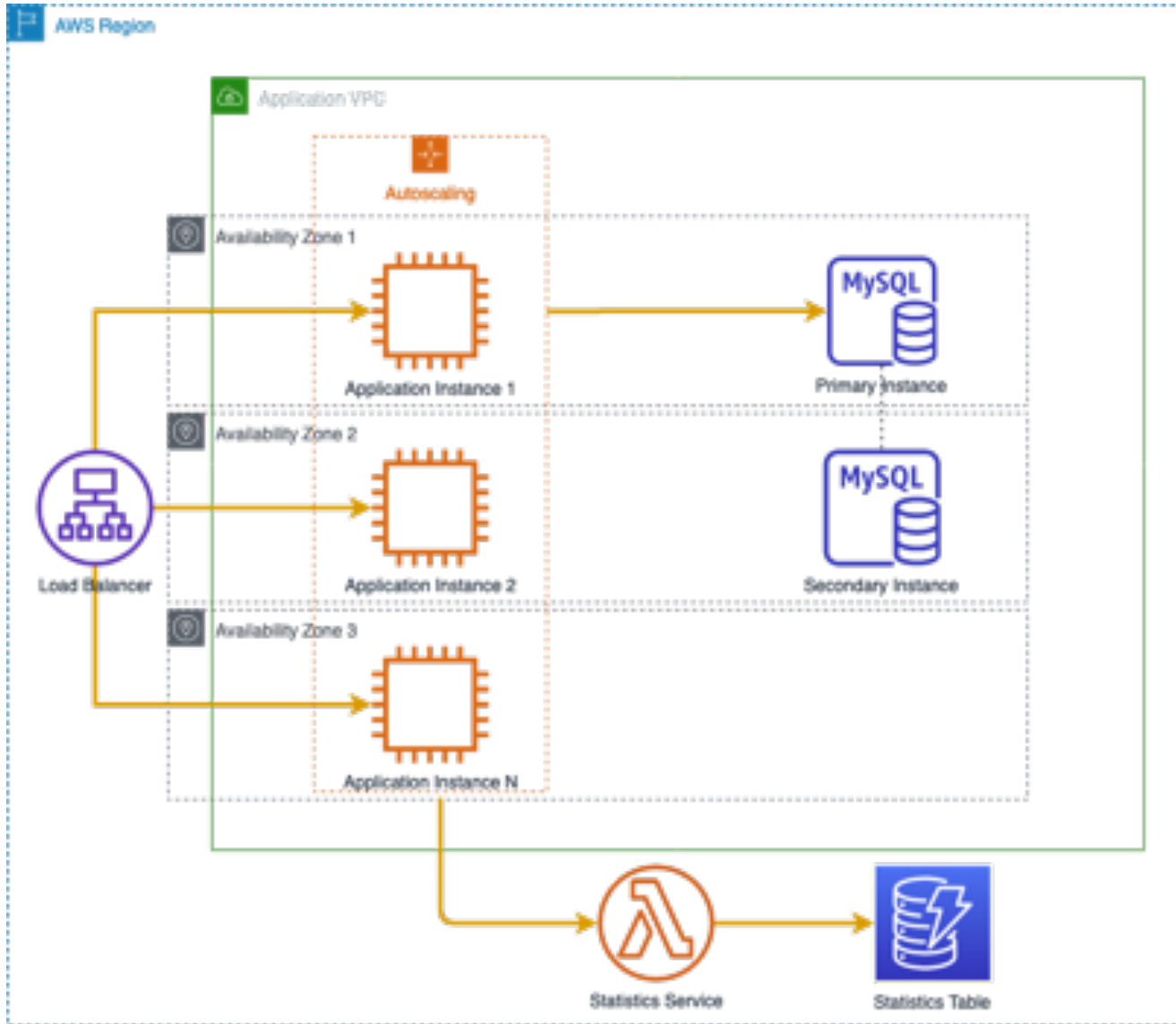


A Tale of Two Systems



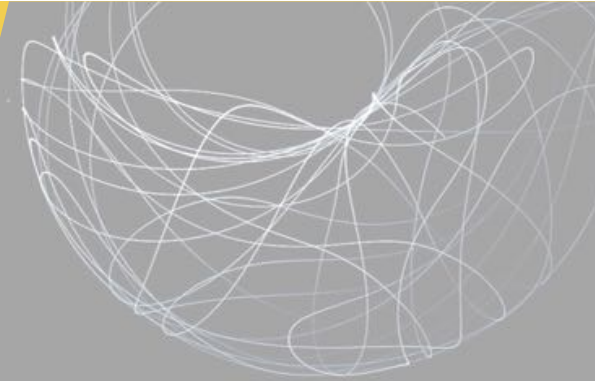


So how do we confidently create resilient systems?

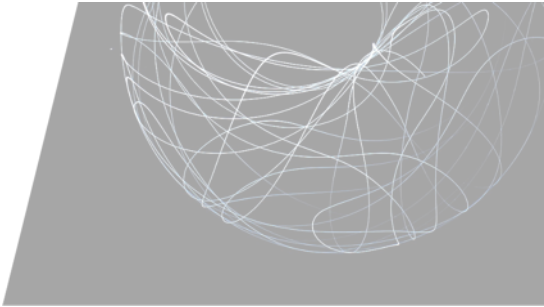


1. Failure domains
2. Fault isolation boundaries
3. Testing with chaos experiments

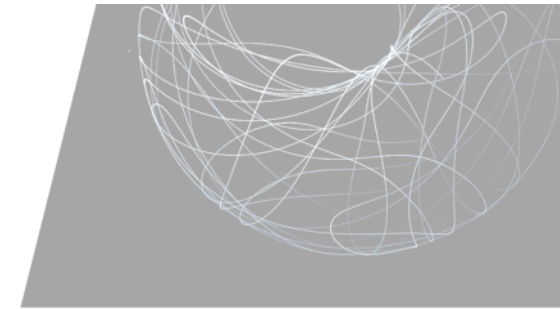
Failure Domains



Failure Domain



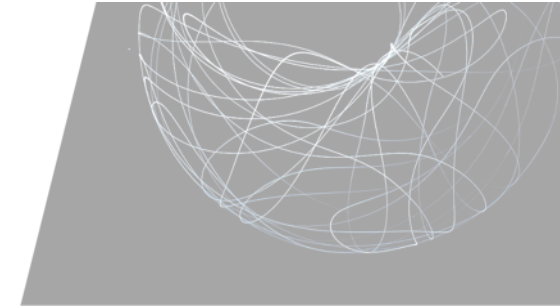
Failure Domain



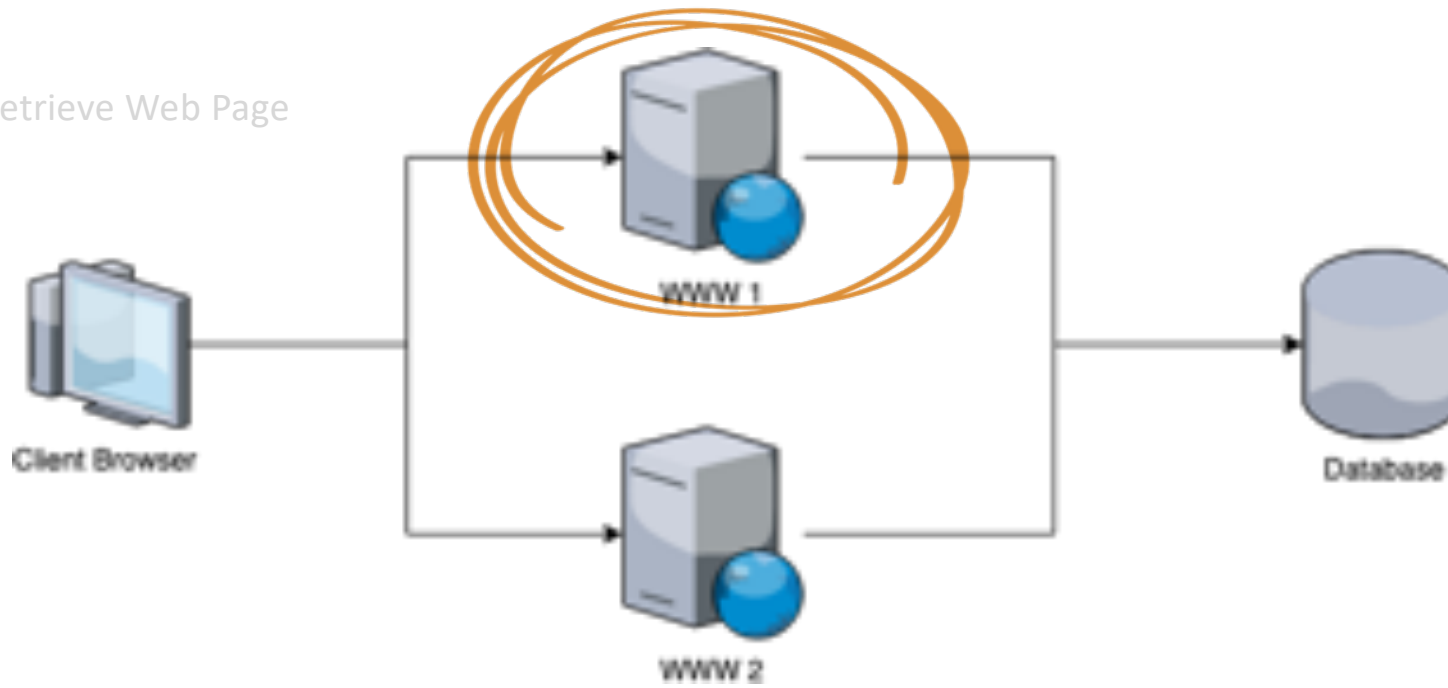
Scenario: Retrieve Web Page



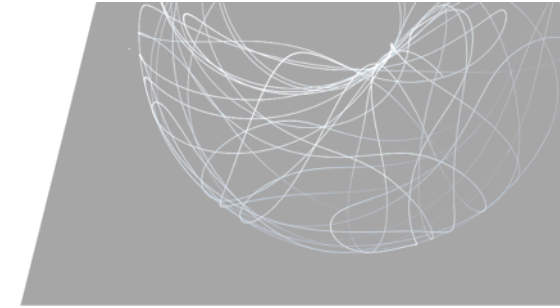
Failure Domain



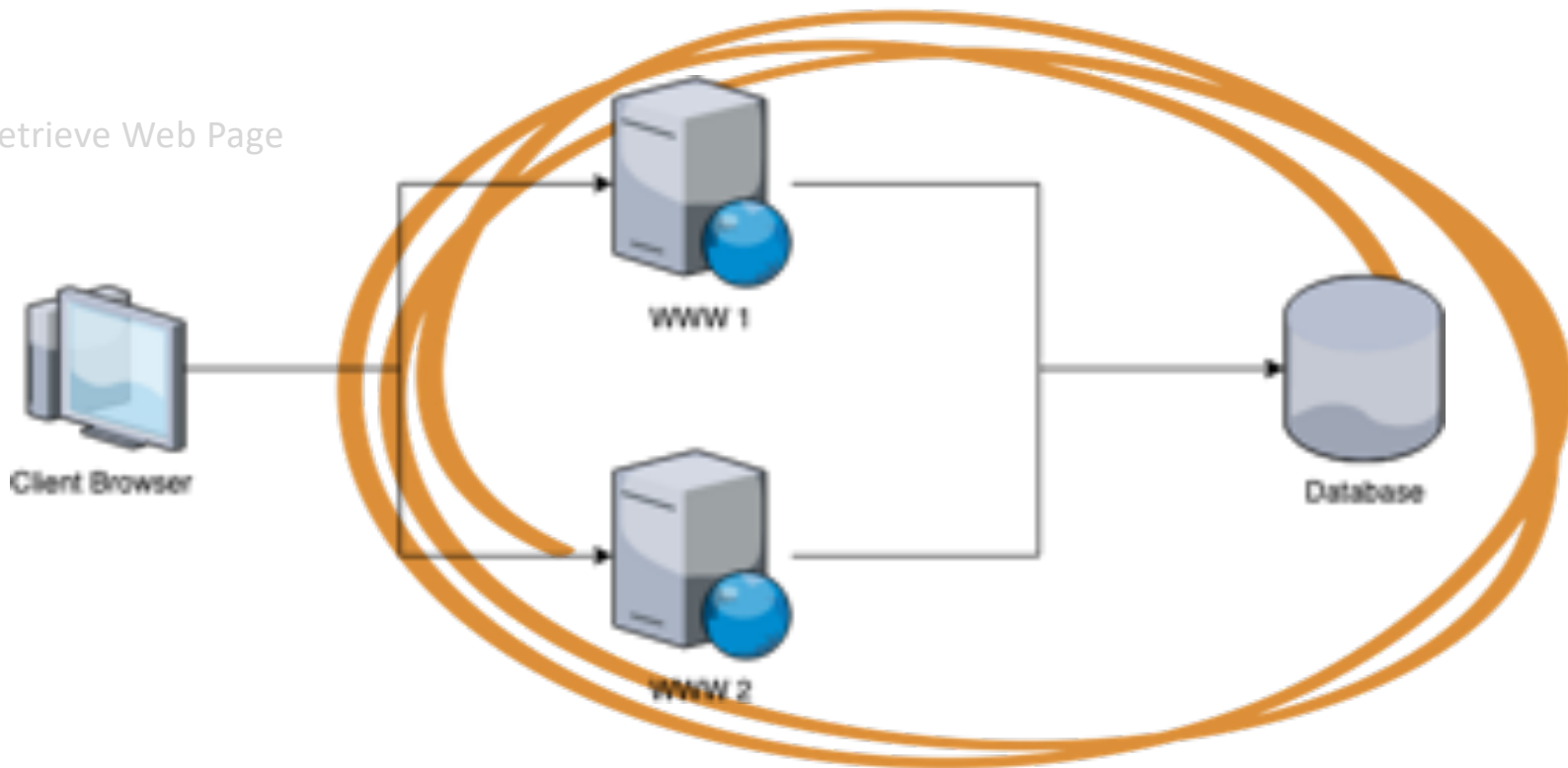
Scenario: Retrieve Web Page

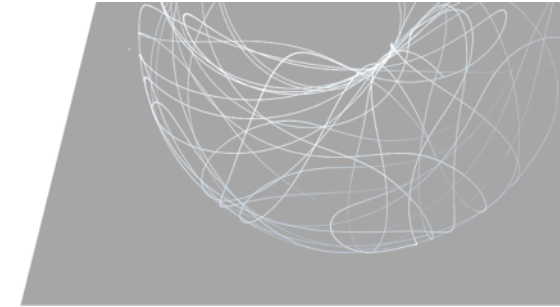
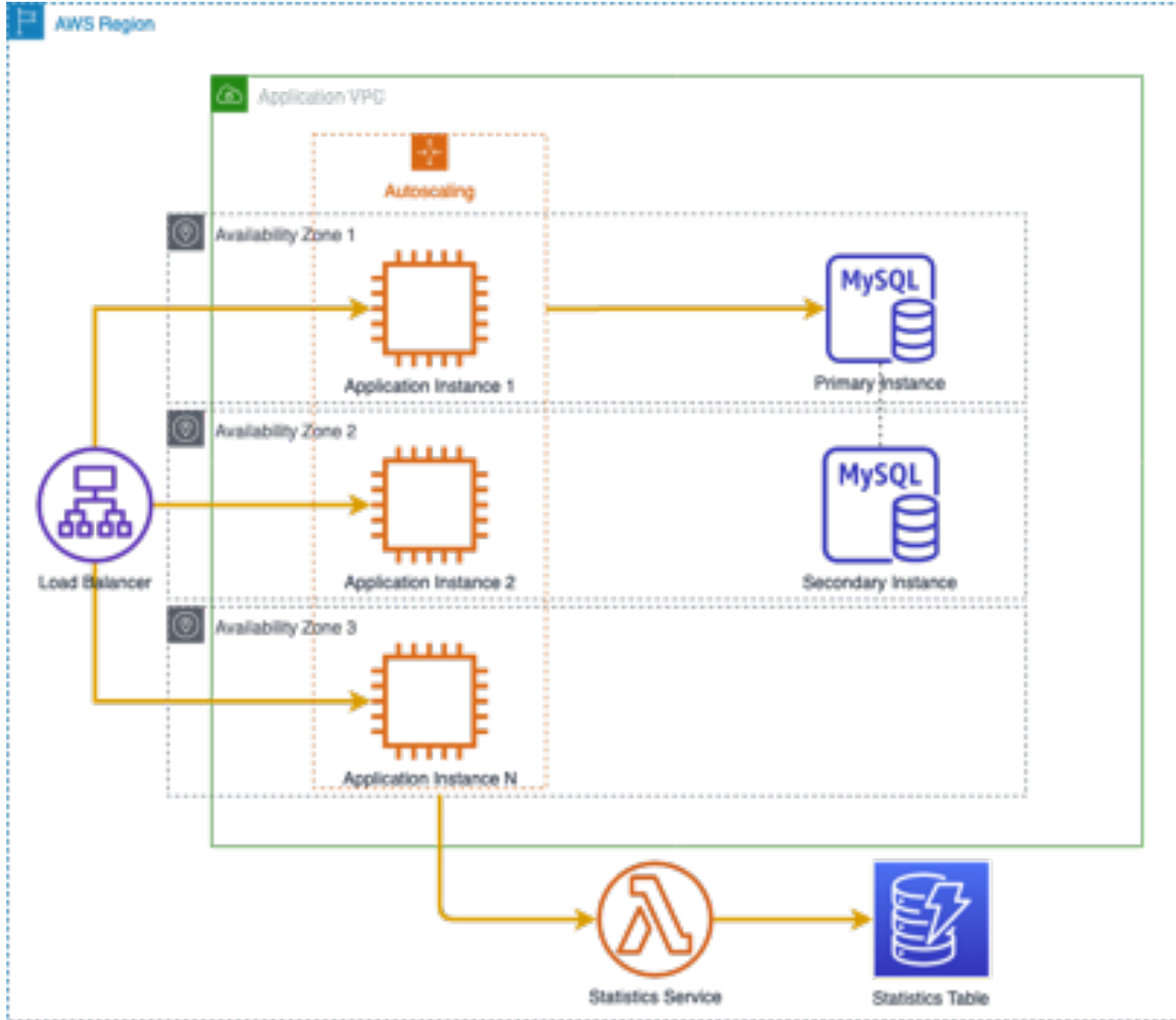


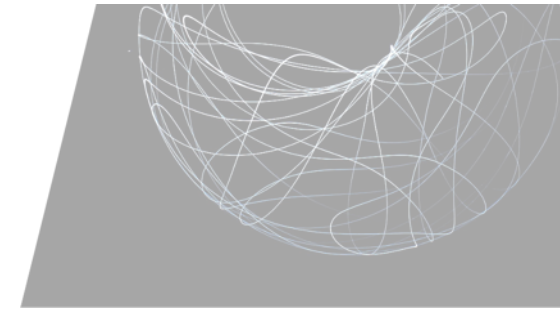
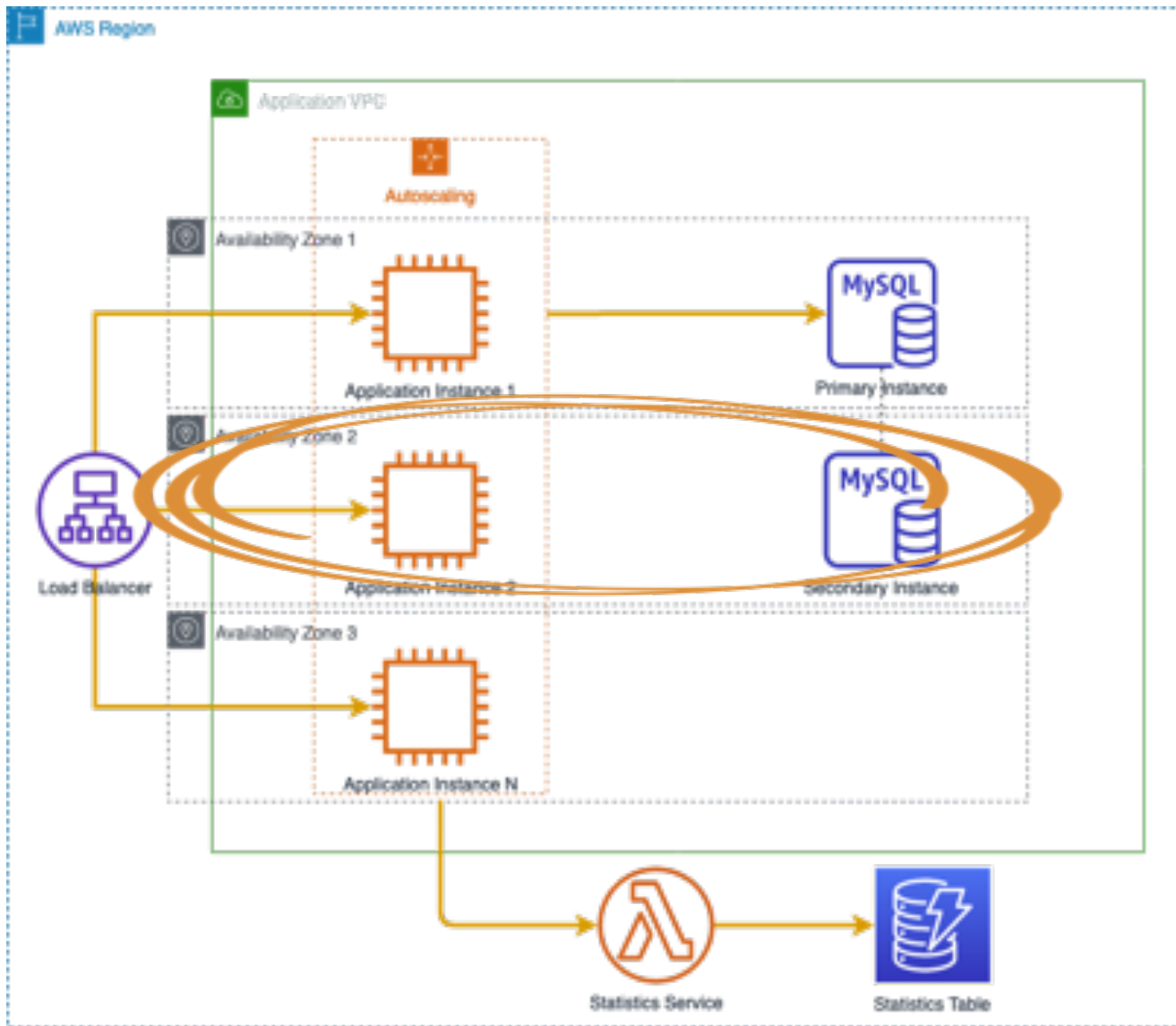
Failure Domain



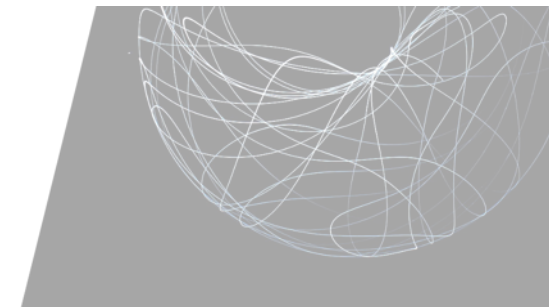
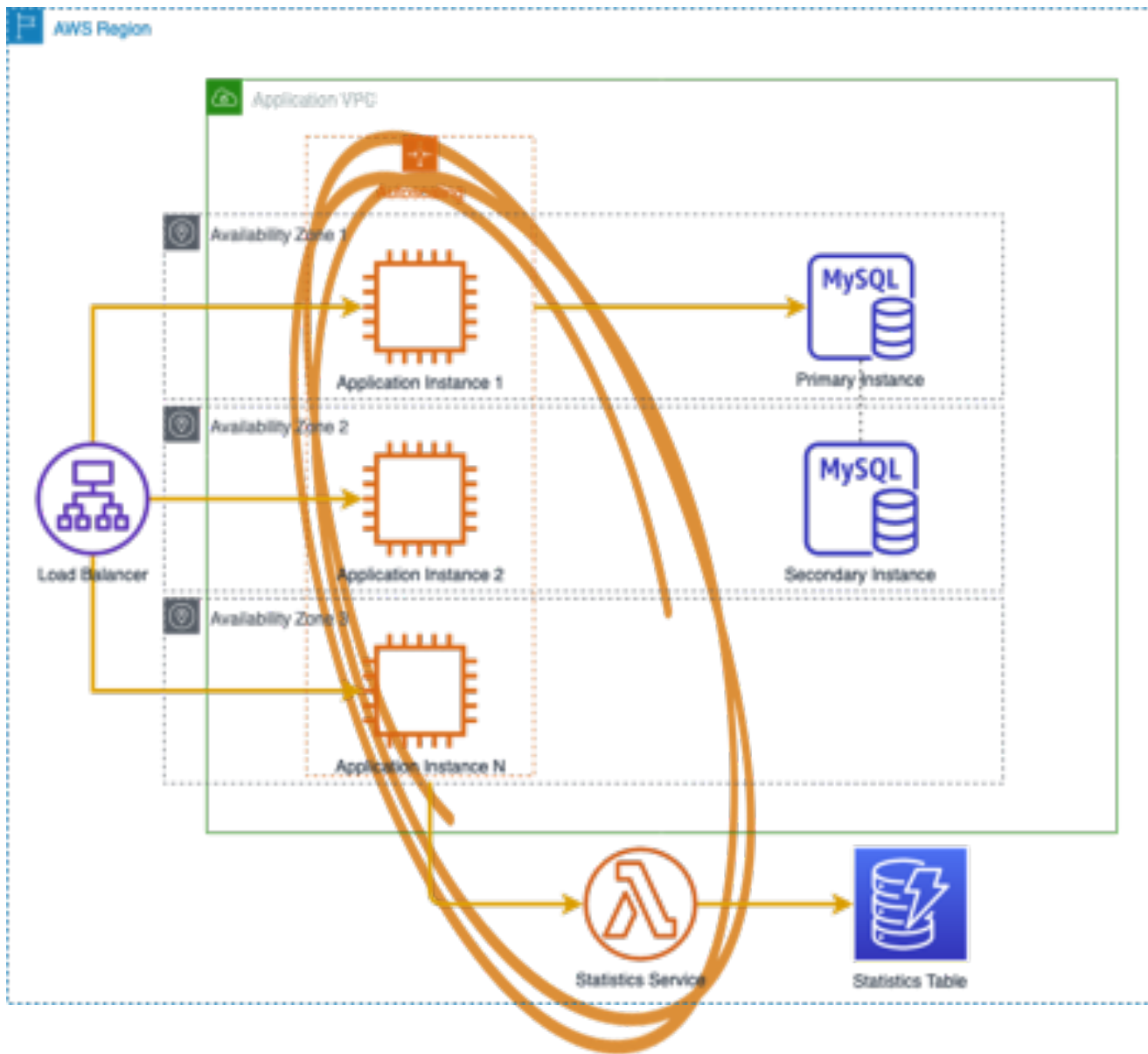
Scenario: Retrieve Web Page





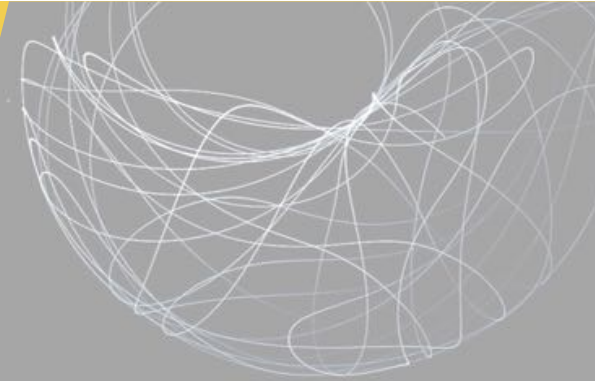


Failure domain of an
AWS Availability
Zone

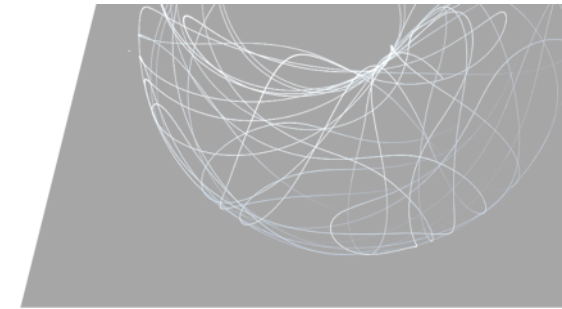


Failure domain of an
AWS Service

Fault Isolation Boundaries



Failure-oriented Patterns

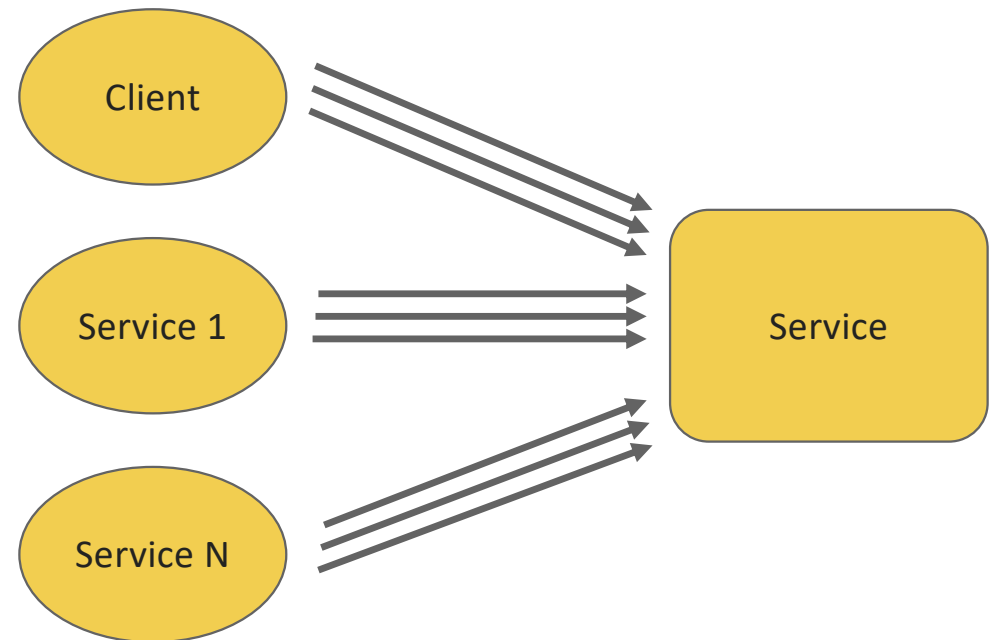


Client

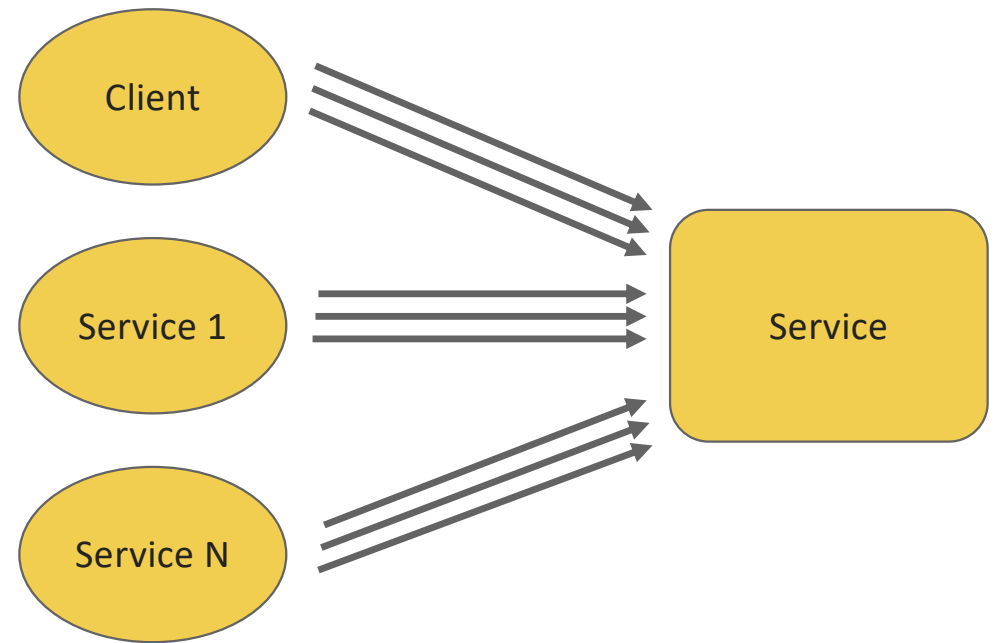
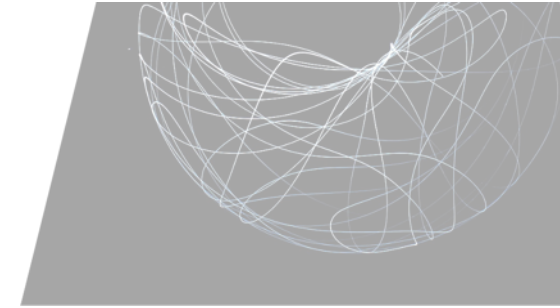
- Set timeouts
- Retries with backoff
 - Add jitter
 - Add retry limit
- Circuit breakers

Service

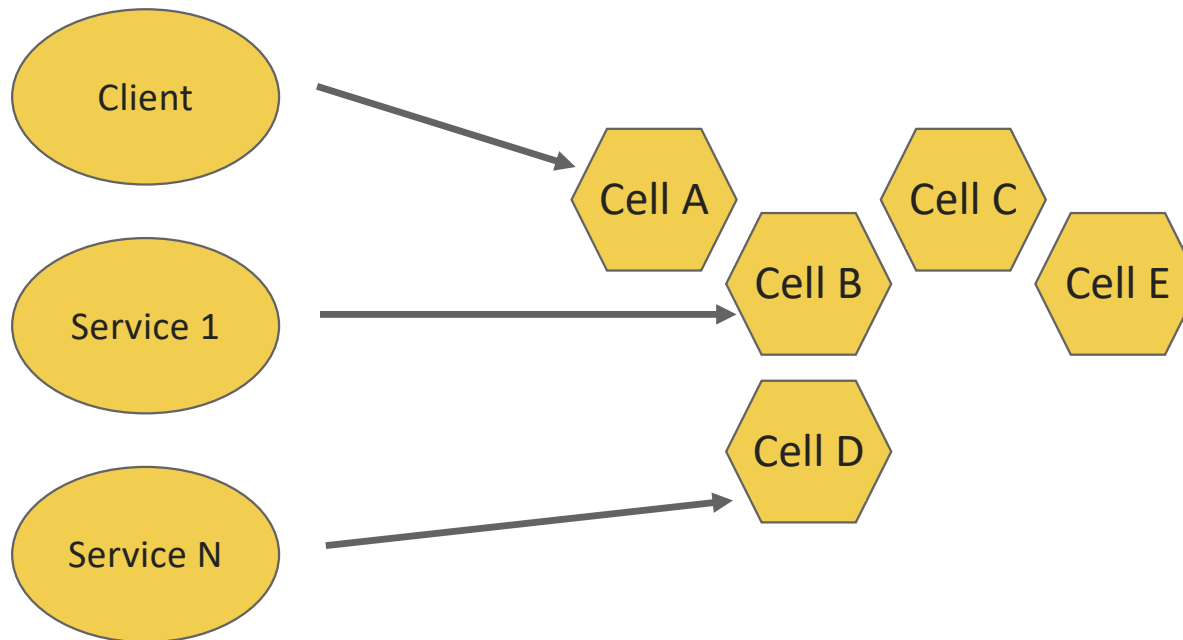
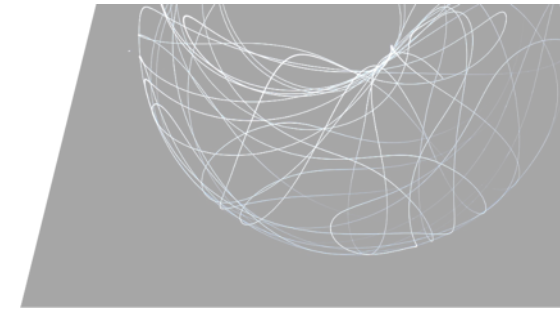
- Rate limit
- Load shedding



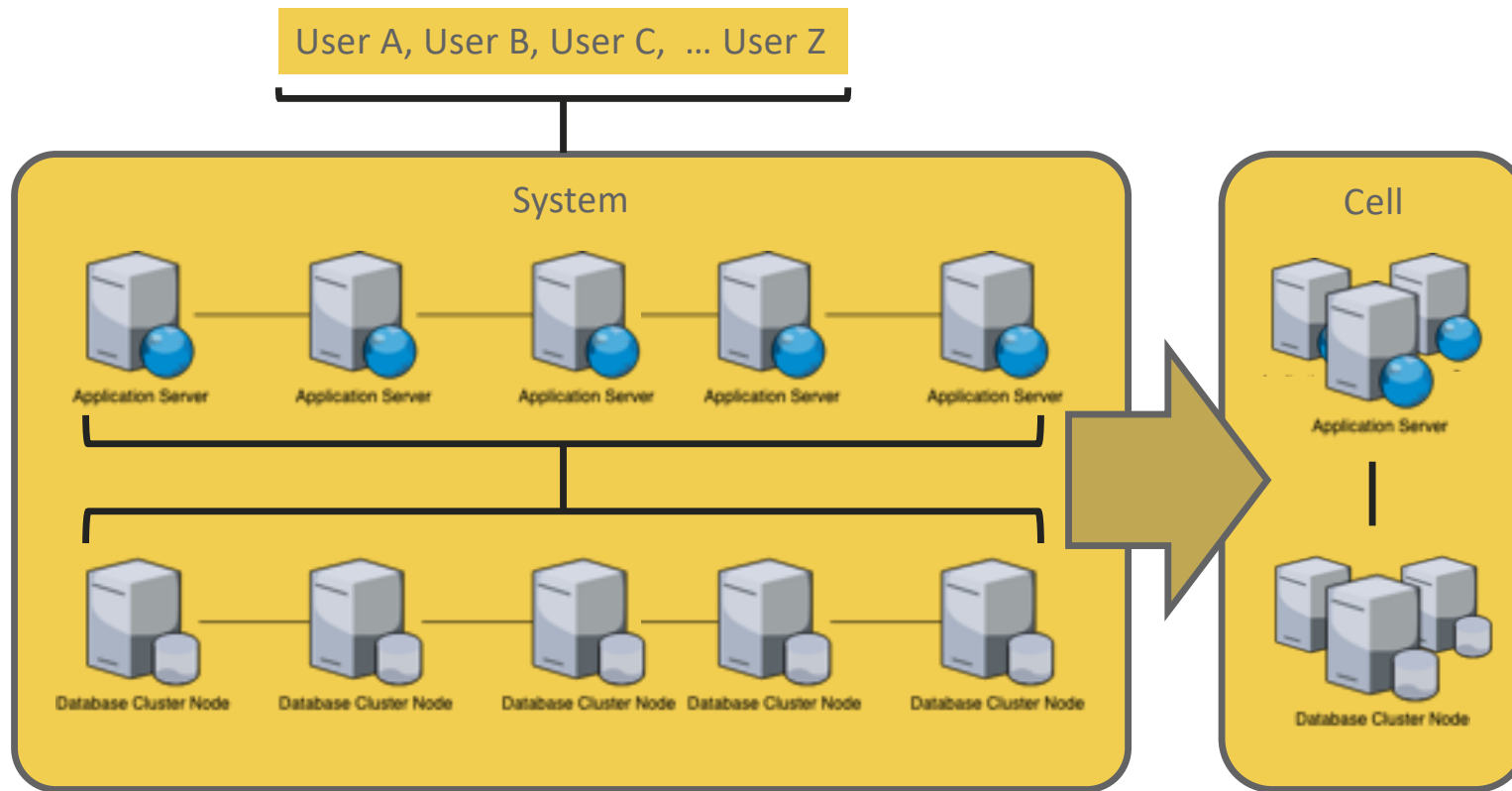
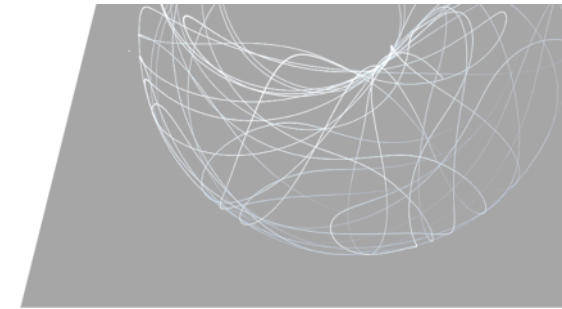
Failure-oriented Patterns



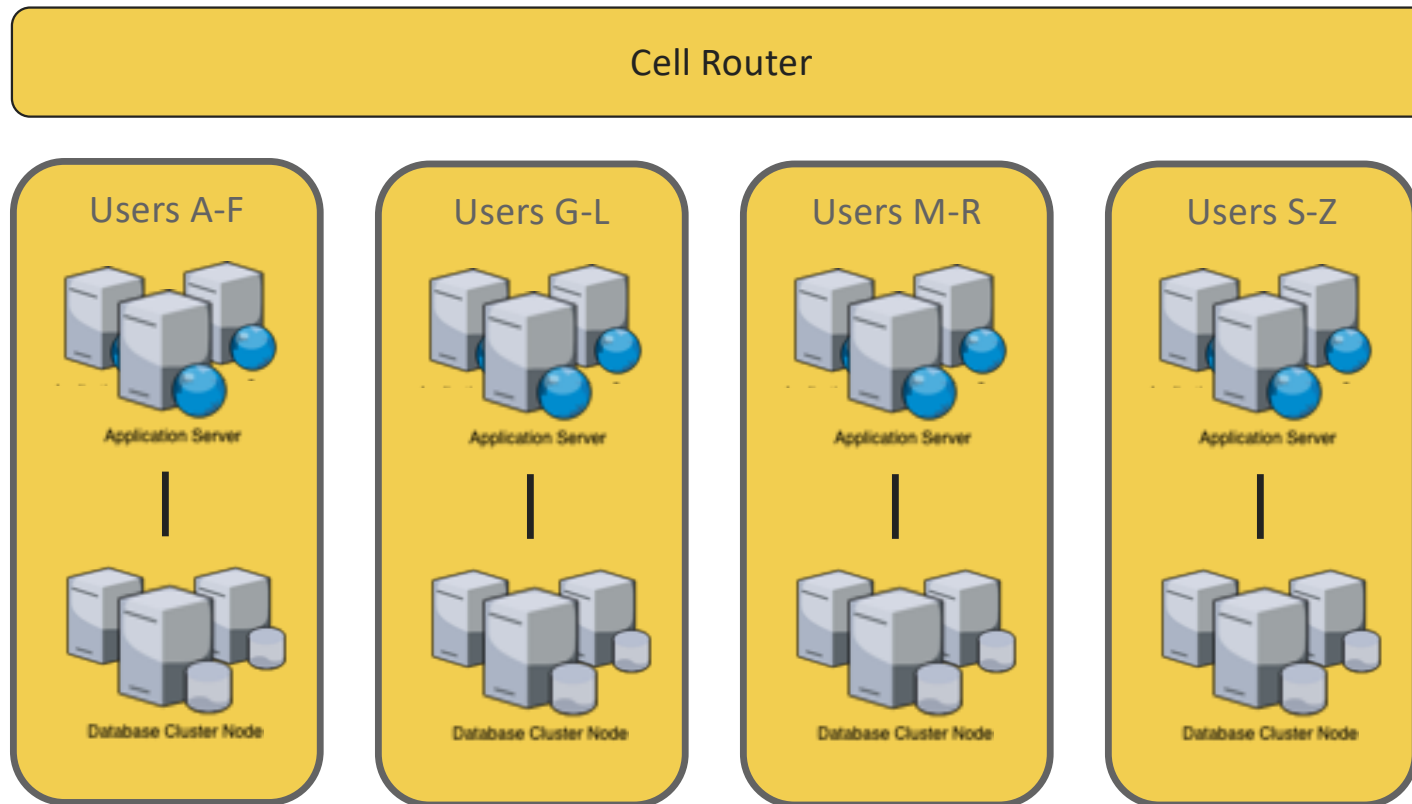
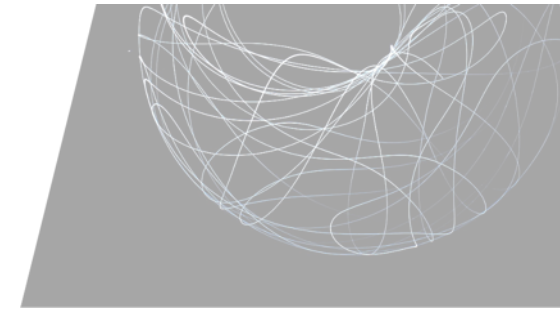
Limiting impact of failures with cells



From System to Cell

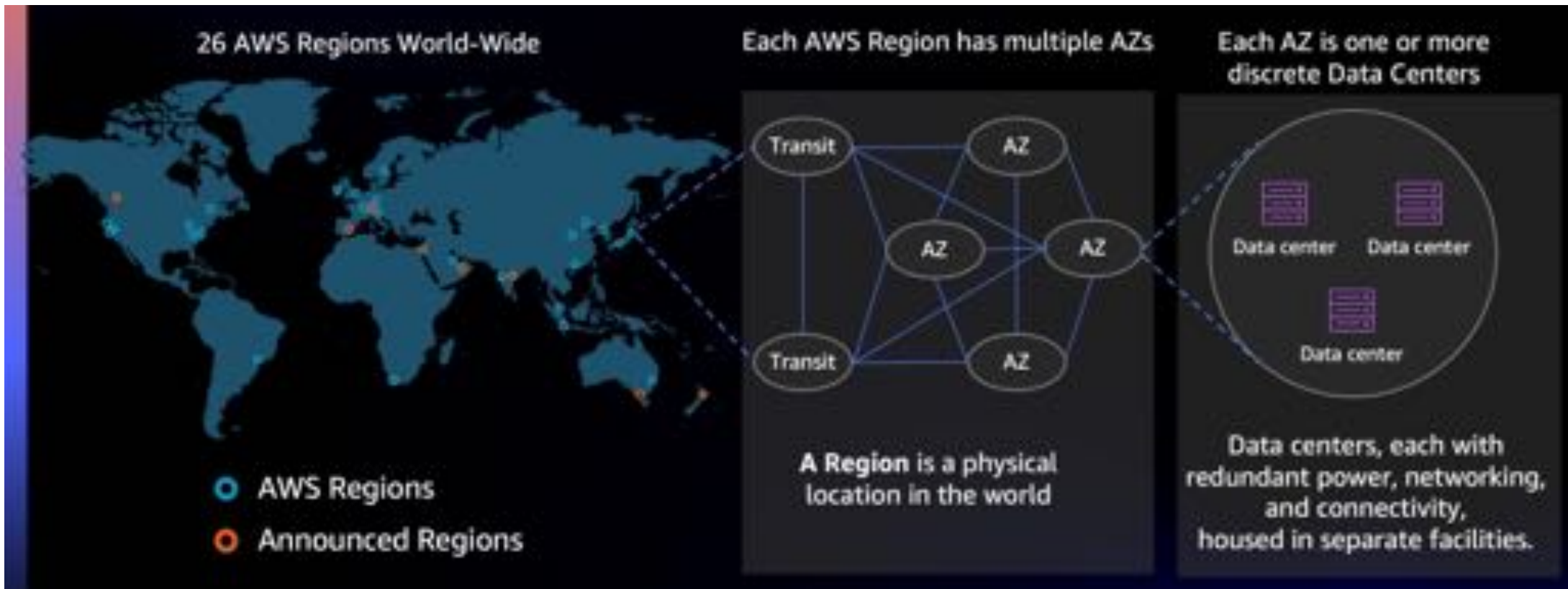
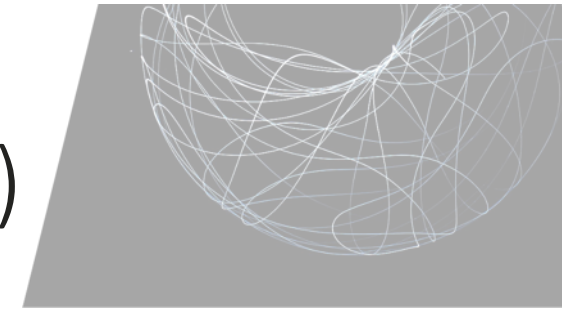


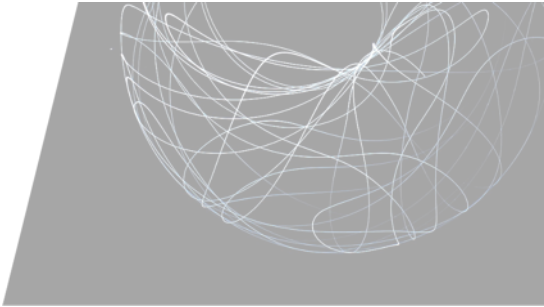
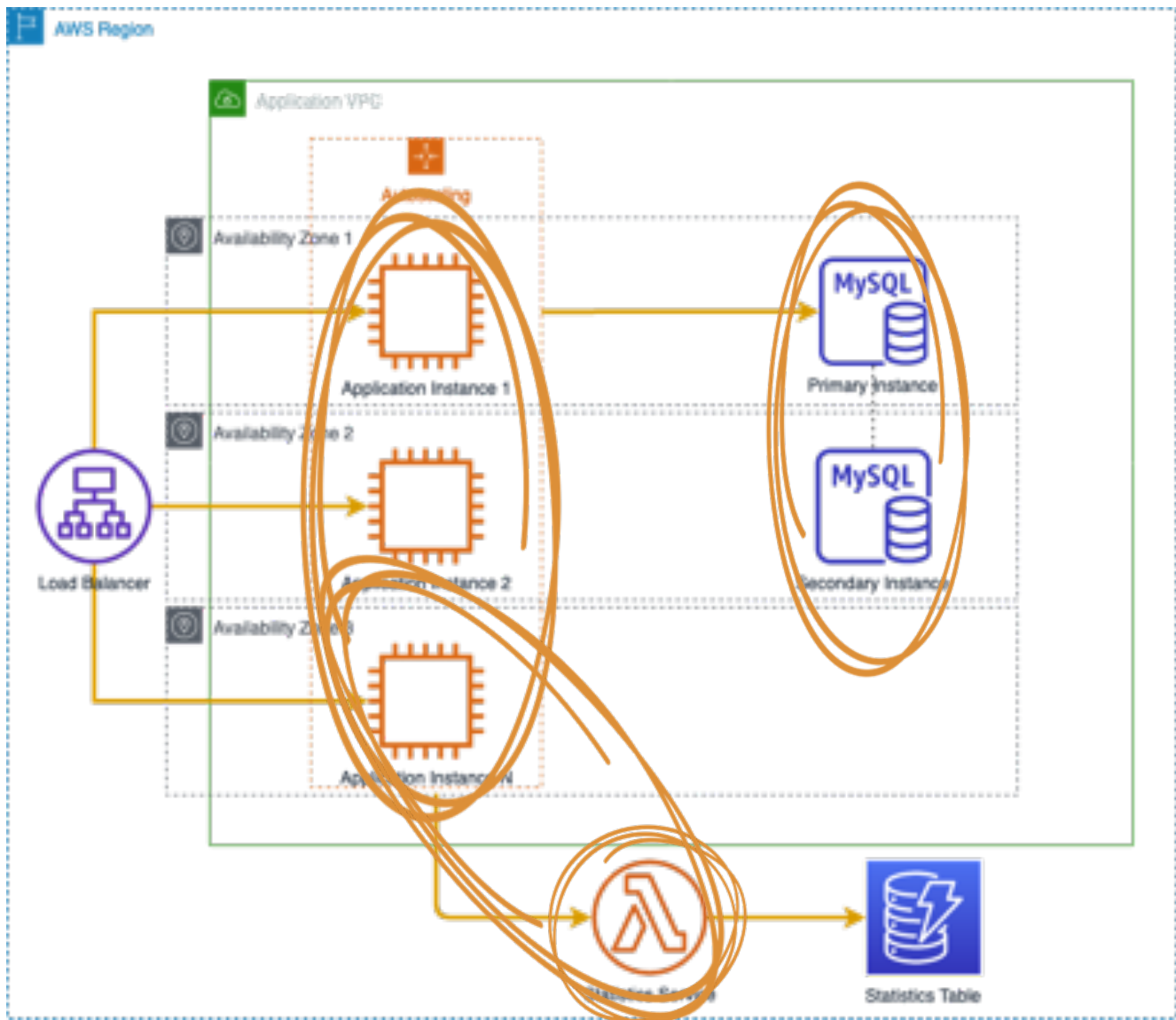
Cell-based Architecture



AWS Regions & Availability Zones (AZs)

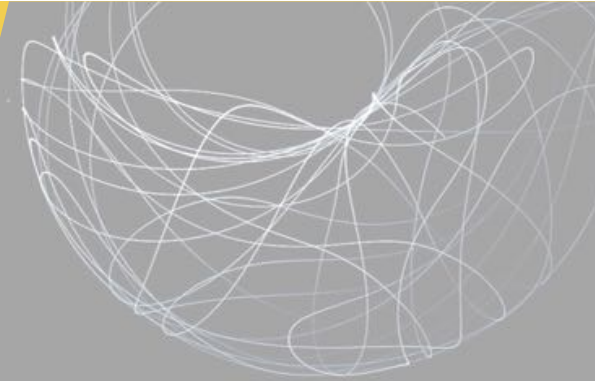
AWS Regions are physical locations around the world with data center clusters



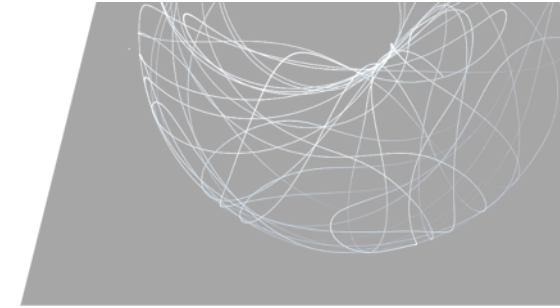


Fault isolation for our system

Initial Tests for Chaos Engineering

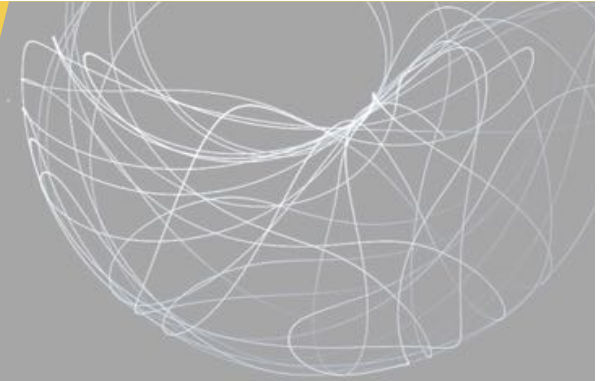


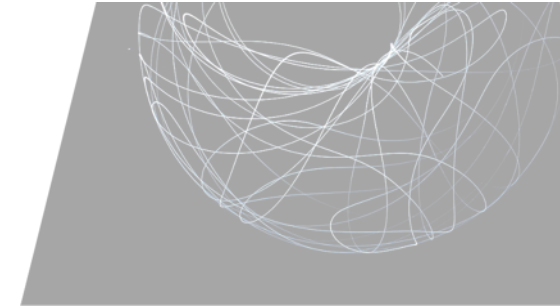
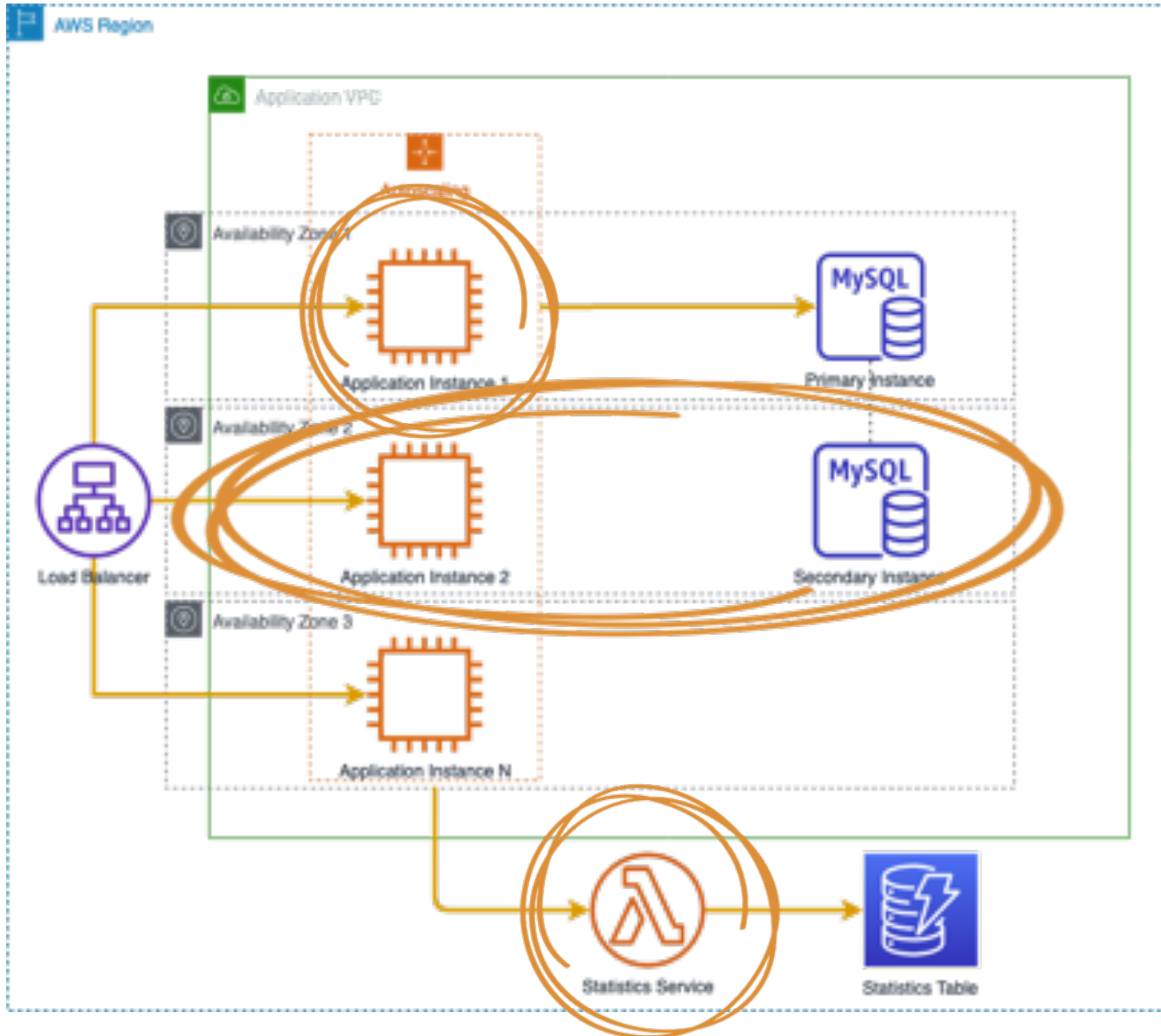
Disclaimer



Do not try this in Production

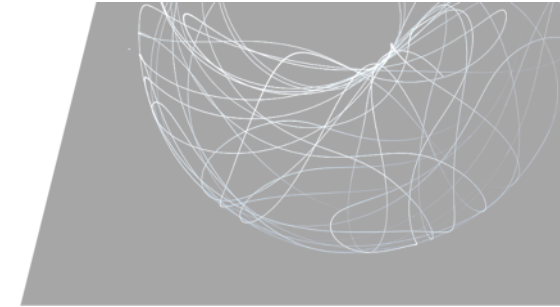
Initial Tests





- Zonal failure
- Centralized dependency
- Service dependency

Initial Tests: Zonal Resources



- Singular resource in an architecture (virtual machine, block device)

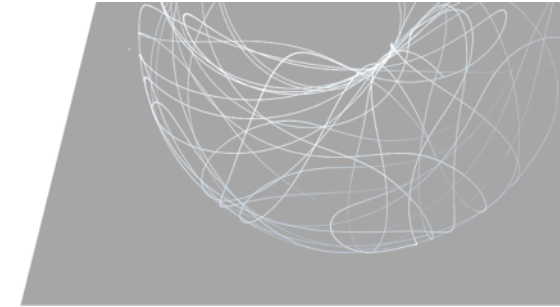
Terminate a virtual machine

```
$ aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

Initiate a failover

```
$ aws rds reboot-db-instance \  
    --db-instance-identifier my-db-identifier \  
    --force-failover
```

Initial Tests: Availability Zone



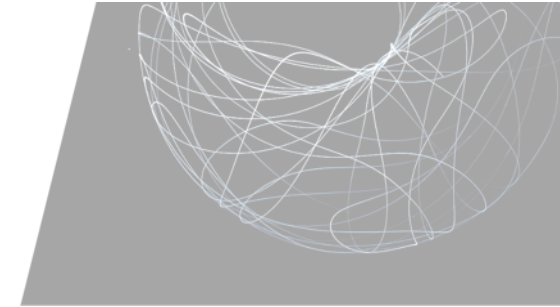
- Block all network traffic into / out of an availability zone

```
# Create a new NACL, storing the ID for the new NACL
$ NEW_NACL_ID=$(aws ec2 create-network-acl --vpc-id $VPC_ID \
  --query NetworkAcl.NetworkAclId --output text)

# Attach 2 rules to the new NACL that deny ingress and egress traffic
$ aws ec2 create-network-acl-entry --network-acl-id $NEW_NACL_ID \
  --rule-number 100 --cidr-block "0.0.0.0/0" --egress --protocol all \
  --port-range From=0,To=65535 --rule-action deny

$ aws ec2 create-network-acl-entry --network-acl-id $NEW_NACL_ID \
  --rule-number 101 --cidr-block "0.0.0.0/0" --ingress --protocol all \
  --port-range From=0,To=65535 --rule-action deny
```


Initial Tests: Regional Service



- Deny access to the service

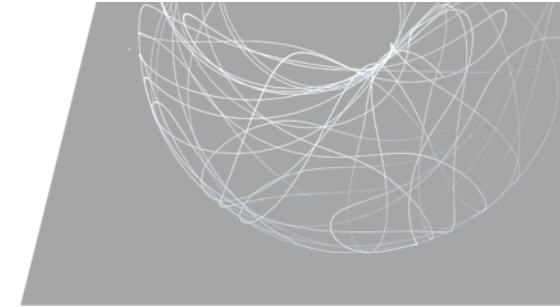
```
# Define the policy document
$ cat >/tmp/deny-lambda.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "lambda:*"
      ],
      "Resource": "*"
    }
  ]
}
EOF
```

```
$ POLICY_ARN=$(aws iam \
  create-policy \
  --policy-name DenyLambda \
  --policy-document \
  file:///tmp/deny-lambda.json \
  --query 'Policy.Arn')
```

Demonstration



Chaos Engineering Tooling



AWS Fault Injection Simulator

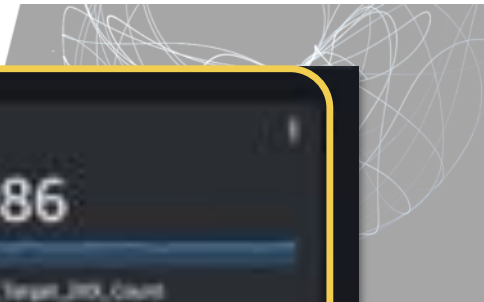
- Serverless
- Experiment templates
- Fault documents
- Integrated with Amazon CloudWatch



The Chaos Toolkit

- An open API to chaos engineering
- Open source extensions for
 - Infrastructure/platform fault injections
 - Application fault injections
 - Observability
- Integrates easily into CI/CD pipelines





Simulate AZ Failure



EXT38H2X1canPBn / Initial Test: Availability Zone Fault Isolation [info](#)

Actions ▾

Details

Experiment template ID

EXT38H2X1canPBn

Stop conditions

[TemplateSystemAlert](#) [🔗](#)

Log group log destination

[%](#) [🔗](#)

Description

Initial Test: Availability Zone Fault Isolation

Creation time

March 20, 2022, 20:19:51 (UTC+08:00)

IAM role

[FISIAMRole](#) [🔗](#)

Last update time

March 21, 2022, 11:59:48 (UTC+08:00)

Actions

Targets

Expert

Tags

Timeline

Actions (2)

View your experiment template actions, action duration, and action sequence.

• **Apply-NACLs / aws:ssm:start-automation-execution (15 min)**

Start: After Assert-Steady-State

• **Assert-Steady-State / aws:cloudwatch:assert-alarm-state**

Start: At beginning of experiment

AZ-Network-Disruption-NACL

[Delete](#)[Actions](#)[View all automation](#)[Description](#)[Content](#)[Versions](#)[Details](#)

Document version

1 (Default)

Document description

Platform	Created	Owner	Target type
Windows, Linux, MacOS	Sat, 19 Mar 2022 21:51:20 GMT	6470063196A	-

Status

 Active

Document Name - NACL-FIS-Automation

What does this document do?

This document modifies the subnets of a particular VPC to deny traffic in the Subnets associated with a particular AZ. Rollback on Cancel or Failure.

Security Risk

Low: This fault does not change the security posture of the VPC since by default the NACL created denies all traffic in the subnet associated with an AZ. It can also easily manually rolled back if necessary since the created NACL is tagged for easy identification.

Input Parameters

- AutomationAssumeRole: (Optional) The ARN of the role that allows Automation to perform the actions on your behalf.
- VPCId: (Required) The ID of the VPC where the subnet resides.
- AvailabilityZone: (Required) The Availability Zone to impact
- Duration: (Optional) Default 1 minute Maximum duration the fault can exist for

Supports Rollback

Yes. The NACL association is reverted.

Cancellation behaviour

The NACL association is reverted.

Output Parameters

EXP355fDfzdCjJMC6e Info

[Refresh](#) [Actions](#) ▼

Details

Experiment ID EXP355fDfzdCjJMC6e	Start time March 28, 2022, 09:50:30 (UTC+01:00)	State Running	Experiment template ID EXT5B42Gk1uaP6r
Creation time March 28, 2022, 09:50:30 (UTC+01:00)	End time -	IAM role FISIAMRole ↗	Stop conditions SampleSystemStart ↗
Log group log destination % ↗			

Actions | [Targets](#) | [Tags](#) | [Timeline](#) | [Stop conditions](#)

Actions (2)

View your experiment template actions, action duration, and action response.

- Apply-NACLs / aws:iam:start-automation-execution (15 min)** Running
Start: After Assert-Steady-State
- Assert-Steady-State / aws:cloudwatch:assert-alarm-state** Completed
Start: At beginning of experiment

sampl-ALBta-1HOGT9TX9X8UW

Actions

arn:aws:elasticloadbalancing:us-west-1:867001831946:targetgroup/sampl-ALBta-1HOGT9TX9X8UW/796254552:cb2130

Details

Target type	Protocol / Port	Protocol version	VPC
Instance	HTTP / 80	HTTP1	vpc-0f2840b0526xxxxx
IP address type	Load balancer		
IPv4	sampl-App-151150C9WL20W		

Total targets	Healthy	Unhealthy	Unused	Initial	DRAINING
3	2	1	0	0	0

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (3)

Unregister Register targets

Filter resources by property or value

< 1 >

Instance ID	Name	Port	Zone	Health status	Health status details
i-0e284869ac88b6c29	sample-app-server	80	us-west-1c	Healthy	
i-0883f2175a042775e	sample-app-server	80	us-west-1a	Unhealthy	Request timed out
i-05a4b46c4f6a02b4ed	sample-app-server	80	us-west-1b	Healthy	

sample-ALBta-1HOGT9TX9X8UW

Actions ▾

arn:aws:elasticloadbalancing:eu-west-1:867001811966:targetgroup/sample-ALBta-1HOGT9TX9X8UW/796854513:tb2130

Details

Target type	Protocol / Port	Protocol version	VPC
Instance	HTTP / 80	HTTP1	vpc-f28402b2696cc0d6d
IP address type	Load balancer		
IPv4	sample-App5-15115604WL20w		

Total targets	Healthy	Unhealthy	Unread	Initial	Draining
4	2	0	0	1	1

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (4)

Refresh | Deregister | Register targets

Filter (required for property or value)

1

Instance ID	Name	Port	Zone	Health status	Health status details
i-02784855wcc0d6d	sample-app-server	80	eu-west-1c	healthy	
i-0682f011wcc0d6d	sample-app-server	80	eu-west-1a	draining	Target deregistration is in progress
i-0a05e4015cc0d6d	sample-app-server	80	eu-west-1a	initial	Target registration is in progress
i-05a0b4d4wcc0d6d	sample-app-server	80	eu-west-1b	healthy	

sample-ALBta-1HOGT9TX9X8UW

Actions ▾

arn:aws:elasticloadbalancing:eu-west-1:867003811966:targetgroup/sample-ALBta-1HOGT9TX9X8UW/796854532:tb2130

Details

Target type	Protocol / Port	Protocol version	VPC
Instance	HTTP / 80	HTTP1	vpc-f28402b2:tkccxatbt
IP address type	Load balancer		
IPv4	sample-App5-15115604wL20w		

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
4	2	1	0	0	1

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (4)

Refresh | Deregister | Register targets

Filter: (optional) for property or value

< 1 > ⌵

Instance ID	Name	Port	Zone	Health status	Health status details
i-0278485f:ec2:tkccxatbt	sample-app-server	80	eu-west-1a	healthy	
i-0662011:ec2:tkccxatbt	sample-app-server	80	eu-west-1a	draining	Target deregistration is in progress
i-0905e4031:ec2:tkccxatbt	sample-app-server	80	eu-west-1a	unhealthy	Request timed out
i-05e403d4:ec2:tkccxatbt	sample-app-server	80	eu-west-1b	healthy	

EXP355fDfzdCjJMC6e [Info](#)

[Refresh](#) [Actions](#) ▼

Details

Experiment ID EXP355fDfzdCjJMC6e	Start time March 28, 2022, 09:50:30 (UTC+01:00)	State Completed	Experiment template ID EXT384CXv1uazP6e
Creation time March 28, 2022, 09:50:30 (UTC+01:00)	End time March 28, 2022, 09:56:15 (UTC+01:00)	IAM role FISIAMRole Info	Stop conditions SampleSystemStart Info
Log group log destination Info			

[Actions](#) | [Targets](#) | [Tags](#) | [Timeline](#) | [Stop conditions](#)

Actions (2)

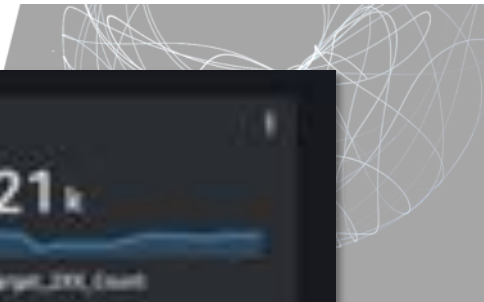
View your experiment template actions, action duration, and action sequence.

• **Apply-NACLs / aws:ssm:start-automation-execution (15 min)** Completed

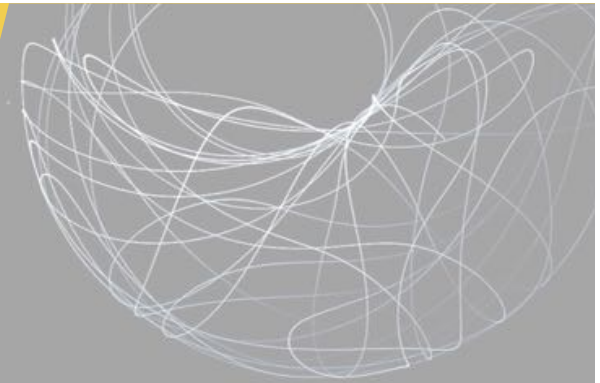
Start: After Assert-Steady-State

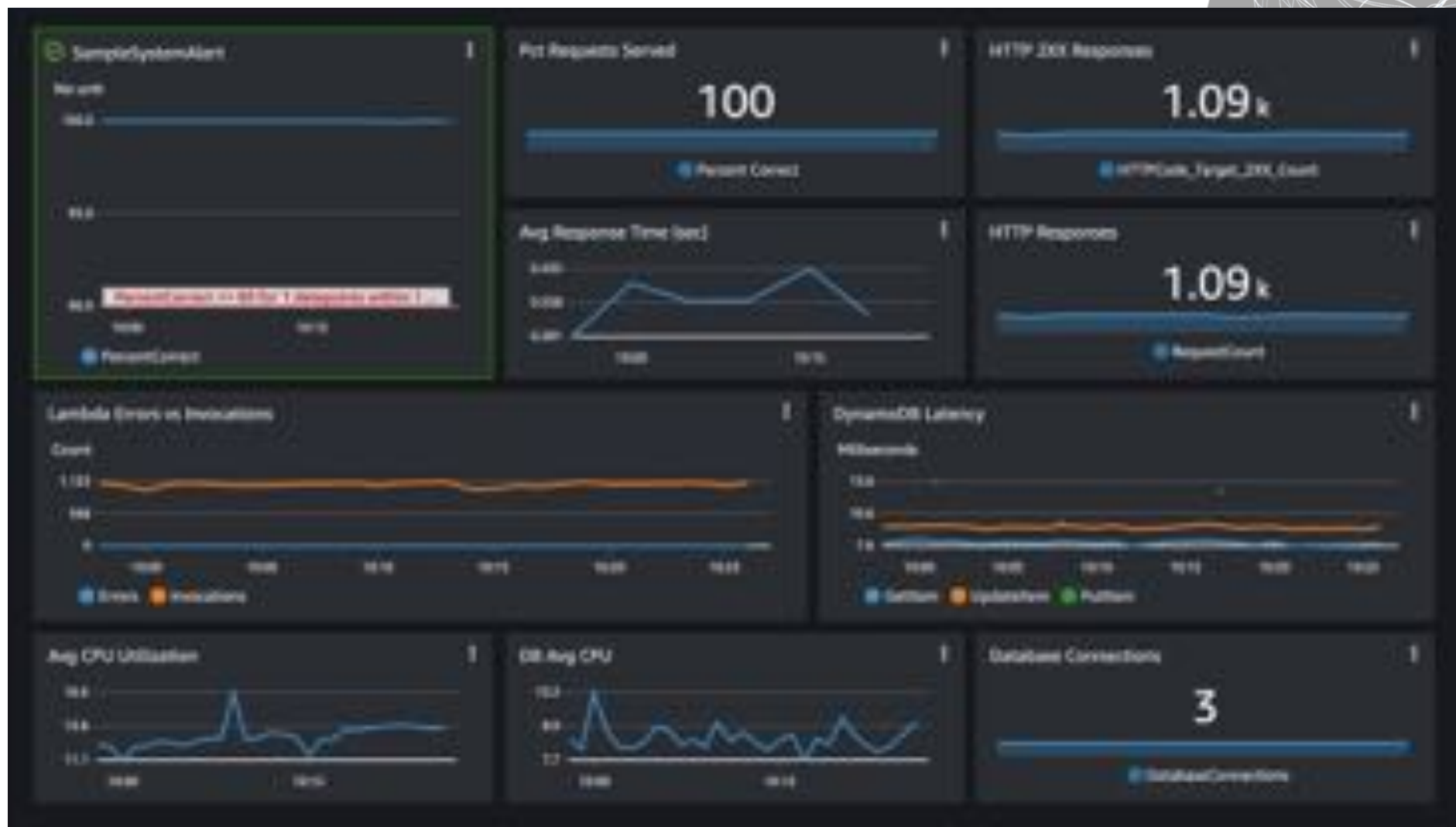
• **Assert-Steady-State / aws:cloudwatch:assert-alarm-state** Completed

Start: At beginning of experiment



Simulate Region Failure





sampl-ALBta-1HOGT9TX9X8UW

Actions

arn:aws:elasticloadbalancing:us-west-1:867005831968:targetgroup/sampl-ALBta-1HOGT9TX9X8UW/796854553dc2130

Details

Target type Instance	Protocol / Port HTTP / 80	Protocol version HTTP1	VPC vpc-0f2940b8264cc6d8
IP address type IPv4	Load balancer sampl-App-1101195C96L20W		

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
3	3	0	0	0	0

- Targets
- Monitoring
- Health checks
- Attributes
- Tags

Registered targets (3)

Registered Register targets

Filter registered by instance or alias

< 1 >

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0c35uab5f6f6e763f	sample-app-server	80	us-west-1a	Healthy	
<input type="checkbox"/>	i-0579f8a00e9844d11	sample-app-server	80	us-west-1b	Healthy	
<input type="checkbox"/>	i-08a25007f09348c	sample-app-server	80	us-west-1c	Healthy	


```

version: 1.0
title: Deny access to AWS Service
description: |
  To simulate service failure this department will apply an IAM policy with a DEN statement to restrict the service control plane.
  We respond with a non-2xx response.

capabilities:
  reliability: high
  availability: high

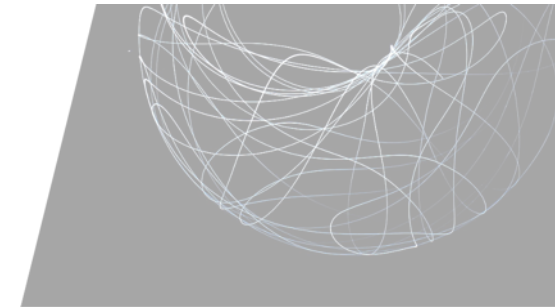
configurations:
  iam_policy_arn: arn:aws:iam::107468812969:policy/deny-access-to-aws-service
  iam_role_name: "arn:aws:iam::107468812969:role/iam-policy-test"

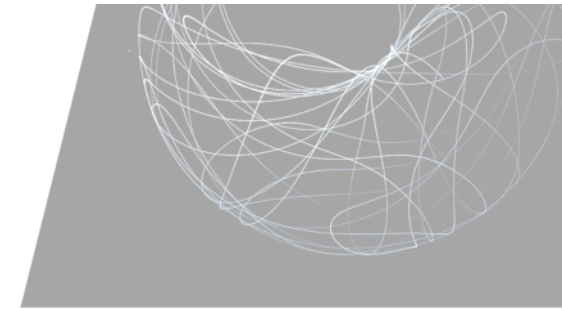
cloud-formation-templates:
  title: Service will success alarm is within tolerance
  problem:
    - type: probe
      name: check-system-error-rate
      tolerance: "0"
      providers:
        type: python
        module: awscloud_formation_probes
        func: get_alarm_state_value
        arguments:
          cloud_name: awscloud_formation

methods:
  - type: action
    name: attach-deny-policy
    providers:
      type: python
      module: awscloud_formation_actions
      func: attach_iam_policy
      arguments:
        iam_policy_arn:
          ref: iam_policy_arn
        role_name: [iam_role_name]
    status: OK
  - type: check-system-error-rate

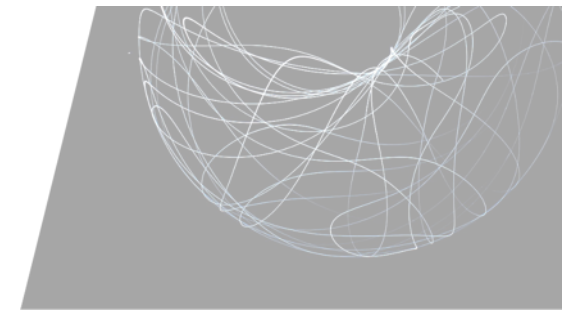
cloudwatch:
  - type: action
    name: detach-deny-policy
    providers:
      type: python
      module: awscloud_formation_actions
      func: detach_iam_policy
      arguments:
        iam_policy_arn:
          ref: iam_policy_arn
        role_name: [iam_role_name]

```

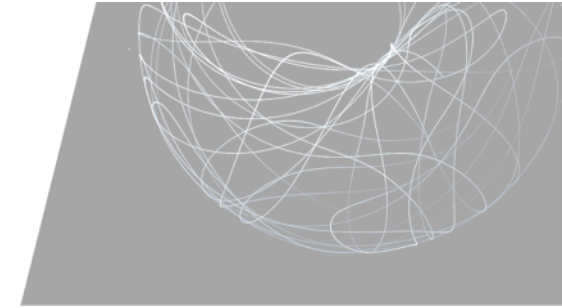




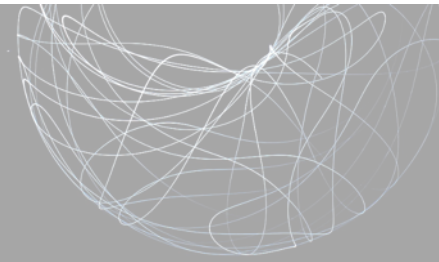
```
[[{"name": "check-system-error-call", "type": "python", "args": [{"name": "check_system_error_call", "type": "python", "args": [{"name": "get_alarm_status_value", "type": "python", "args": [{"name": "alarm_value", "type": "boolean"}]}]}]}]]
```



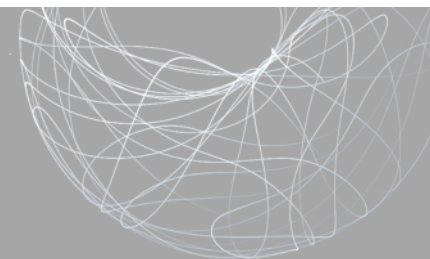
```
def main():
    # Type: ACTION
    name: attack_policy
    providers:
        type: python
        module: ansible_collections.ansible.netconf.plugins.action_attack_policy
    arguments:
        url: {{url_policy_url}}
        url_name: {{url_name}}
    return:
        # Type: CHECK_SYSTEM_ERROR
        - name: check_system_error
```



```
#!/usr/bin/perl
# Types action
name default-mysql-policy
provider
  type python
  module ansible_collections.ansible.netconf.plugins.action_policy
  arguments:
    name: ansible_action_policy
    url: https://raw.githubusercontent.com/ansible-collections/ansible-netconf/master/plugins/action_policy.py
```



```
1 "Year": "2017-08-17",
2 "Category": [
3   "Effect": "Day",
4   "Action": [
5     "action:1:read:read:read",
6     "action:1:read:read:read"
7   ],
8   "Message": "action:1:read:read:read:read:read:read:read:read:read:read"
9 ]
```



```
~/d/sample-aws-workload/e/fail-lambda on main 11:12:19 chaos run experiment.yaml \  
--rollback-strategy always \  
--journal-path exp-fail-lambda-run-003.log
```

```
[2022-03-28 23:50:39 INFO] Validating the experiment's syntax  
[2022-03-28 23:50:39 INFO] Experiment looks valid  
[2022-03-28 23:50:39 INFO] Running experiment: Deny access to AWS Service  
[2022-03-28 23:50:39 INFO] Steady-state strategy: default  
[2022-03-28 23:50:39 INFO] Rollbacks strategy: always  
[2022-03-28 23:50:39 INFO] Steady state hypothesis: Service call success alarm is within tolerance  
[2022-03-28 23:50:39 INFO] Probe: check-system-error-rate  
[2022-03-28 23:50:40 INFO] Steady state hypothesis is met!  
[2022-03-28 23:50:40 INFO] Playing your experiment's method now...  
[2022-03-28 23:50:40 INFO] Action: attach-deny-policy  
[2022-03-28 23:50:40 INFO] Pausing after activity for 900s...
```



Details

Target type
Instance

Protocol / Port
HTTP / 80

Protocol version
HTTP1

VPC

vpc-4f26402d:af85cc02b1 [↗](#)

IP address type
IPv4

Load balancer
sample-App-15V15004WLE23W [↗](#)

Total targets

5

Healthy



Unhealthy



Unused



Initial



Draining



Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (5)



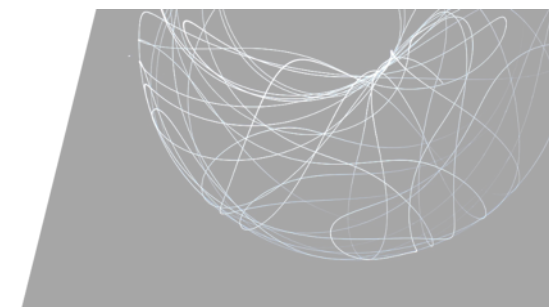
Deregister

Register targets

Filter resources by property or value

< 1 >

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-08626fa7d8f9166d	sample-app-server	80	eu-west-1c	unhealthy	Health checks failed with these codes [500]
<input checked="" type="checkbox"/>	i-0868a6d72217a9ca	sample-app-server	80	eu-west-1a	draining	Target deregistration is in progress
<input type="checkbox"/>	i-08626fa7d8f9166d	sample-app-server	80	eu-west-1b	unhealthy	Health checks failed with these codes [500]
<input checked="" type="checkbox"/>	i-0579f5a1040864a7	sample-app-server	80	eu-west-1b	draining	Target deregistration is in progress
<input checked="" type="checkbox"/>	i-08626fa7d8f9166d	sample-app-server	80	eu-west-1c	draining	Target deregistration is in progress



```
[2022-03-28 23:58:30 INFO] Validating the experiment's syntax
[2022-03-28 23:58:39 INFO] Experiment looks valid
[2022-03-28 23:58:39 INFO] Running experiment: Deny access to AWS Service
[2022-03-28 23:58:39 INFO] Steady-state strategy: default
[2022-03-28 23:58:39 INFO] Rollbacks strategy: always
[2022-03-28 23:58:39 INFO] Steady state hypothesis: Service call success alarm is within tolerance
[2022-03-28 23:58:39 INFO] Probe: check-system-error-rate
[2022-03-28 23:58:40 INFO] Steady state hypothesis is met!
[2022-03-28 23:58:40 INFO] Playing your experiment's method now...
[2022-03-28 23:58:40 INFO] Action: attach-deny-policy
[2022-03-28 23:58:40 INFO] Pausing after activity for 900s...
[2022-03-29 00:06:07 INFO] Probe: check-system-error-rate
[2022-03-29 00:06:07 INFO] Steady state hypothesis: Service call success alarm is within tolerance
[2022-03-29 00:06:07 INFO] Probe: check-system-error-rate
[2022-03-29 00:06:08 CRITICAL] Steady state probe 'check-system-error-rate' is not in the given tolerance so failing this experiment
[2022-03-29 00:06:08 WARNING] Rollbacks were explicitly requested to be played
[2022-03-29 00:06:08 INFO] Let's rollback...
[2022-03-29 00:06:08 INFO] Rollback: detach-deny-policy
[2022-03-29 00:06:08 INFO] Action: detach-deny-policy
[2022-03-29 00:06:08 INFO] Experiment ended with status: deviated
[2022-03-29 00:06:08 INFO] The steady-state has deviated, a weakness may have been discovered
```

Details

Target type
Instance

Protocol / Port
HTTP / 80

Protocol version
HTTP1

VPC
[vpc-0f26402b6816xxxxx](#)

IP address type
IPv4

Load balancer
[sample-App-15N15004WJ2CM](#)

Total targets

5

Healthy

1

Unhealthy

1

Unused

0

Initial

1

Draining

2

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (5)



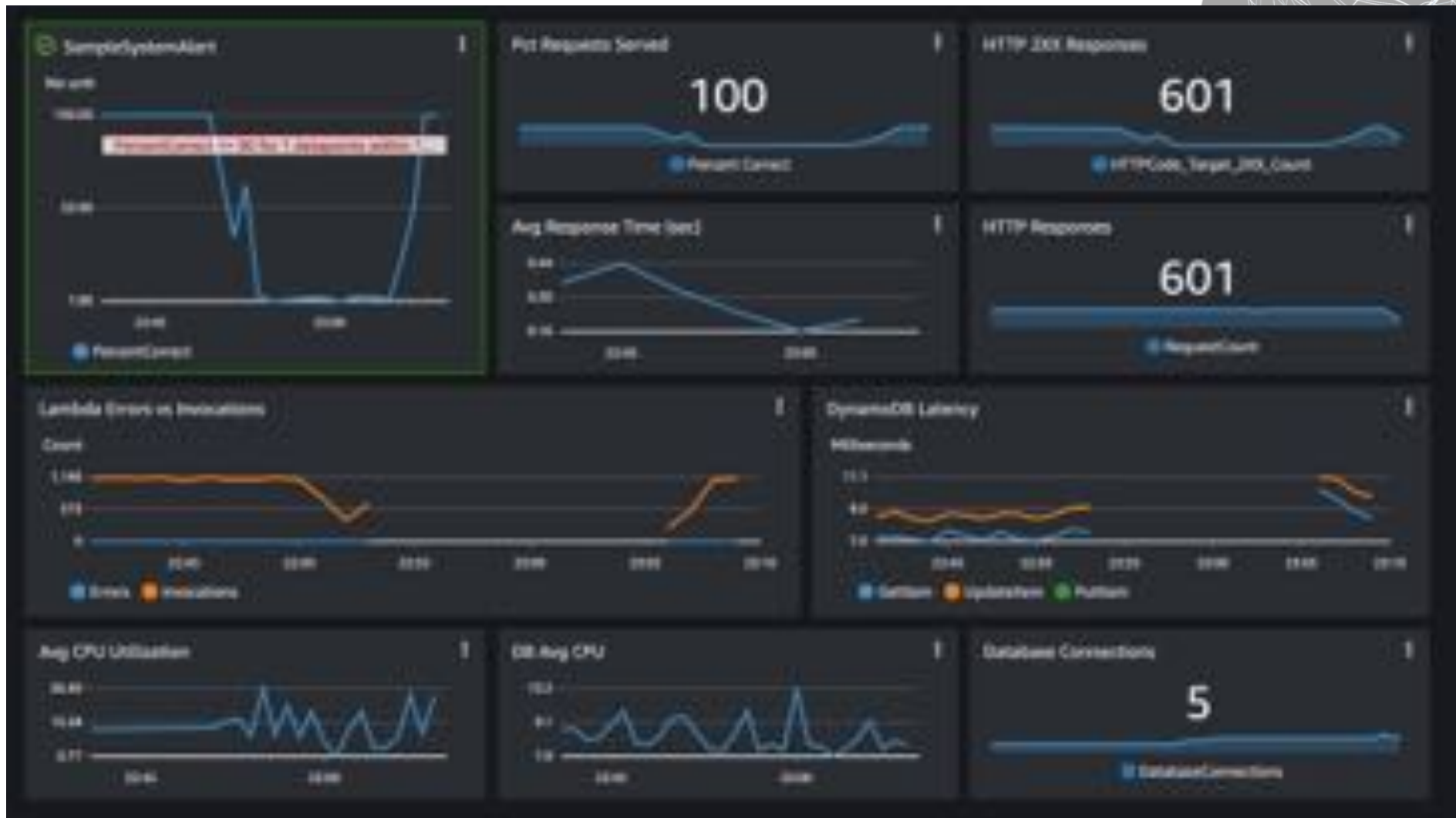
De-register

Register targets

Filter results by property or value

1

Instance ID	Name	Port	Zone	Health status	Health status details
i-007c80xx51819678	sample-app-server	80	eu-west-1c	Initial	Target registration is in progress
i-0360f06d12e90c964d	sample-app-server	80	eu-west-1a	draining	Target deregistration is in progress
i-3a05239f9c6d6d4e	sample-app-server	80	eu-west-1c	Unhealthy	Health checks failed with these codes [500]
i-06721e029e58668f	sample-app-server	80	eu-west-1b	draining	Target deregistration is in progress
i-04143476d6c8f721f	sample-app-server	80	eu-west-1a	Healthy	



Details

Target type
Instance

Protocol / Port
HTTP / 80

Protocol version
HTTP1

VPC

vpc-0f26402b6816xxxxx01d [↗](#)

IP address type
IPv4

Load balancer

sample-AppB-15V15G04WJ23W [↗](#)

Total targets

5

Healthy

3

Unhealthy

0

Unread

0

Initial

0

Draining

2

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (5)



De-register

Register targets

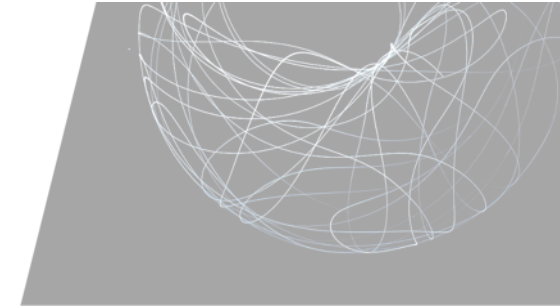
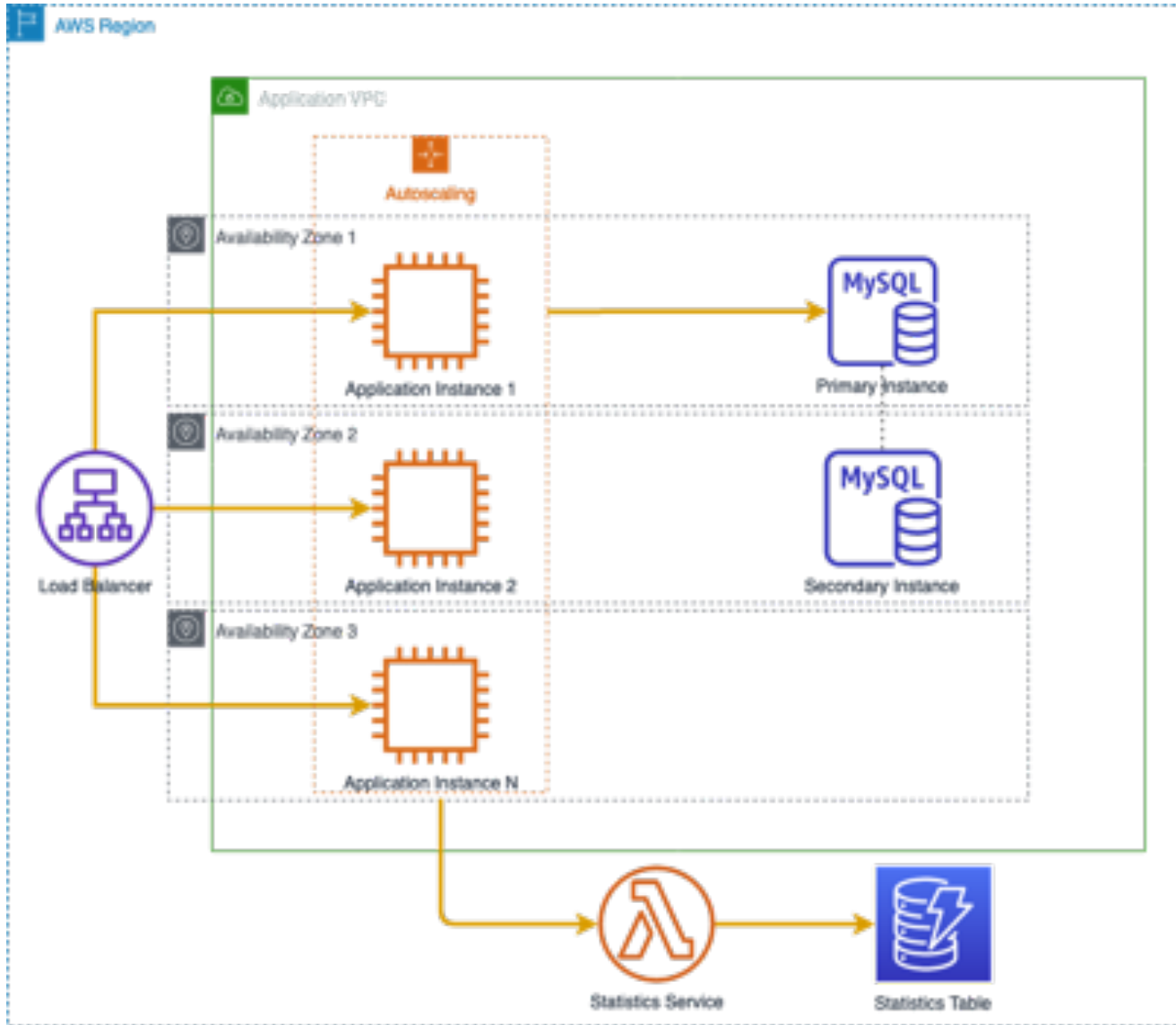
Filter (restricted by expand/collapse)

< 1 >

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-007c80xxv53359878	sample-app-server	80	eu-west-1b	Healthy	
<input checked="" type="checkbox"/>	i-0605230f6c6a6846	sample-app-server	80	eu-west-1c	Draining	Target deregistration is in progress
<input checked="" type="checkbox"/>	i-06725f025458668f	sample-app-server	80	eu-west-1b	Draining	Target deregistration is in progress
<input type="checkbox"/>	i-0416546b6c08f2a1	sample-app-server	80	eu-west-1a	Healthy	
<input type="checkbox"/>	i-0a94830a07940246a	sample-app-server	80	eu-west-1c	Healthy	

Wrapping Up





Learnings:

- Centralized dependency
- Service dependency

Wrap Up

- Nature of failure (binary vs gray failure)
- This is only an initial set of tests
- Identify your failure domains and fault isolation boundaries
- Test your fault isolation boundaries

 jason_barto

