



How to Test Your Fault Isolation Boundaries in the Cloud

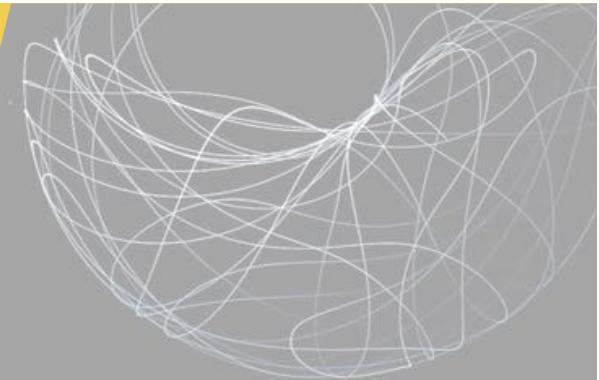


jason_barto

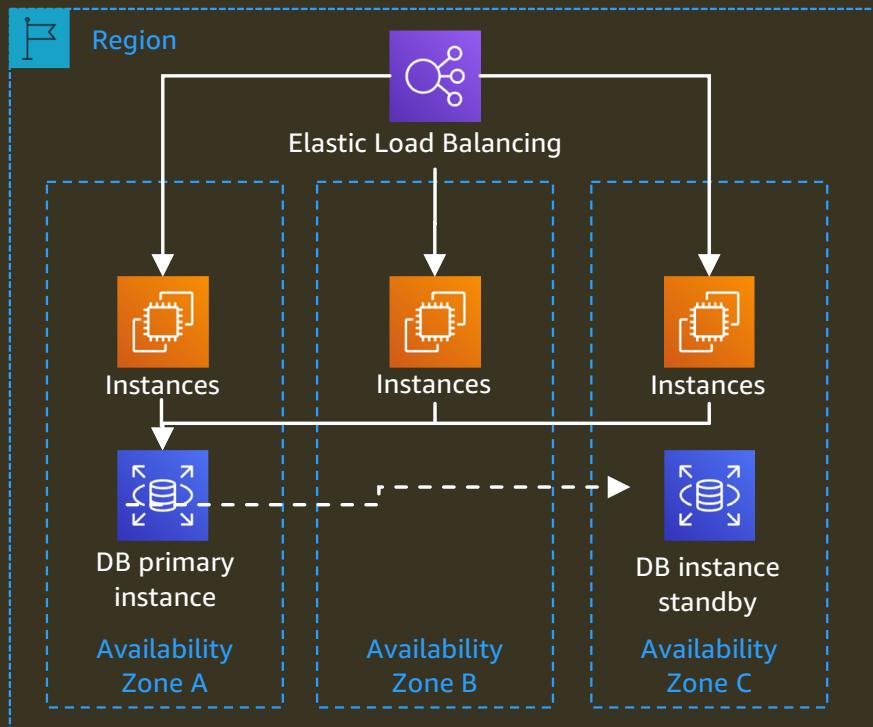


A Tale of Two Systems

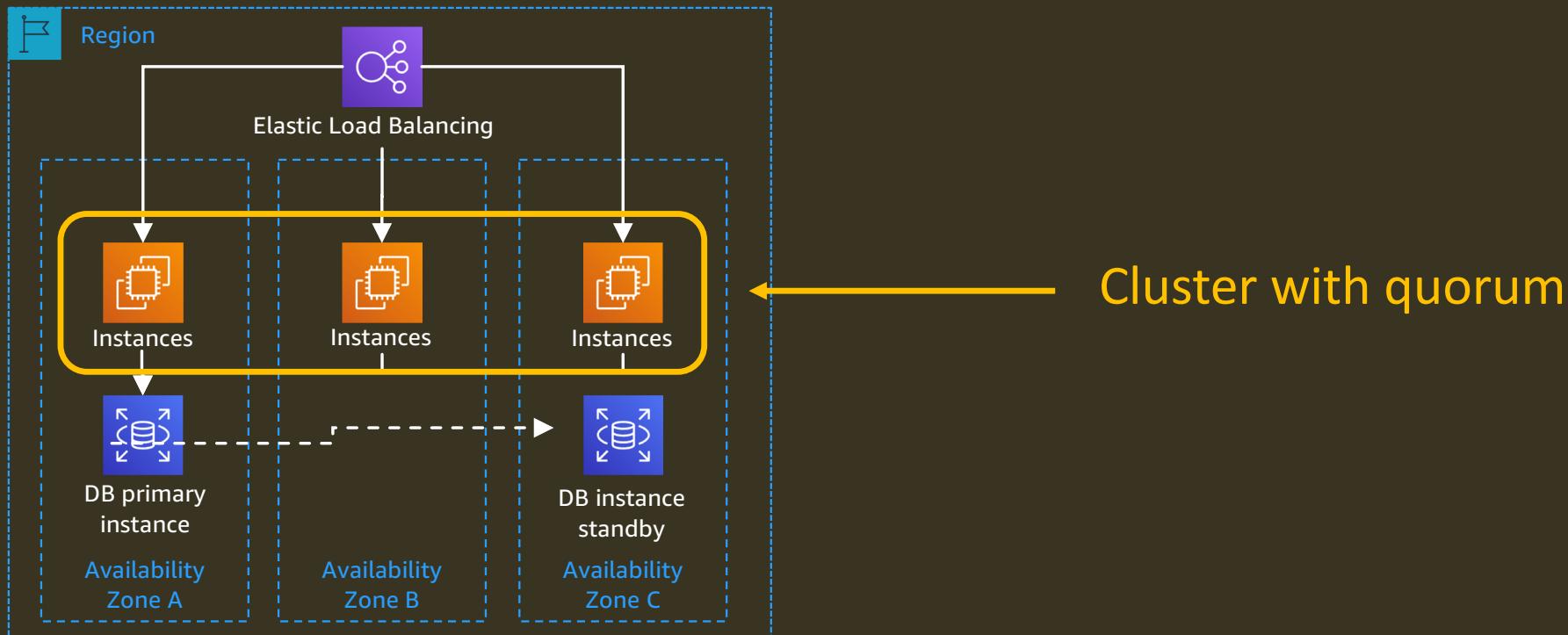
How a resilient banking system failed



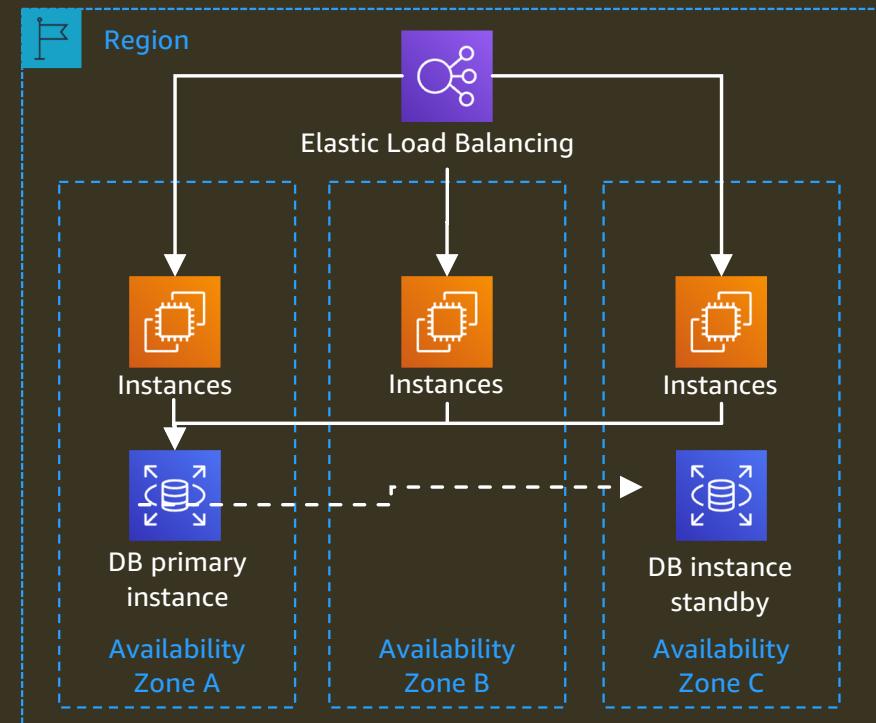
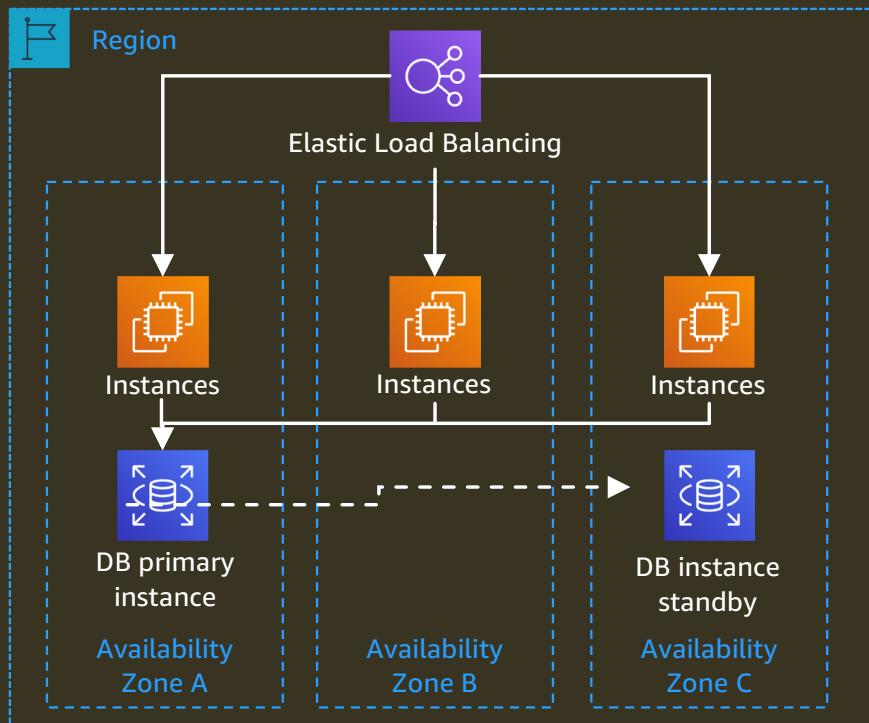
A Tale of Two Systems



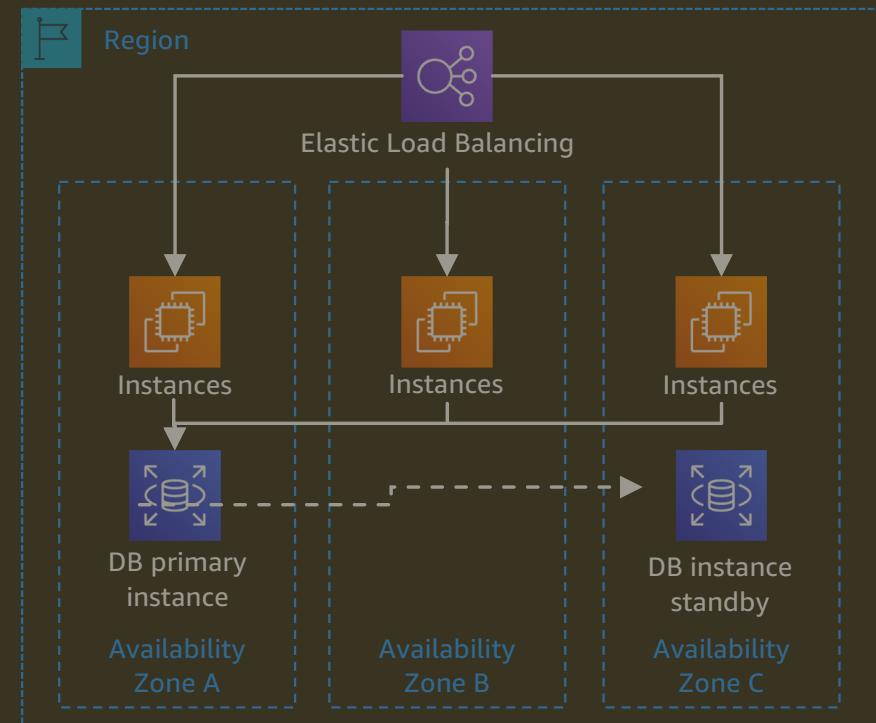
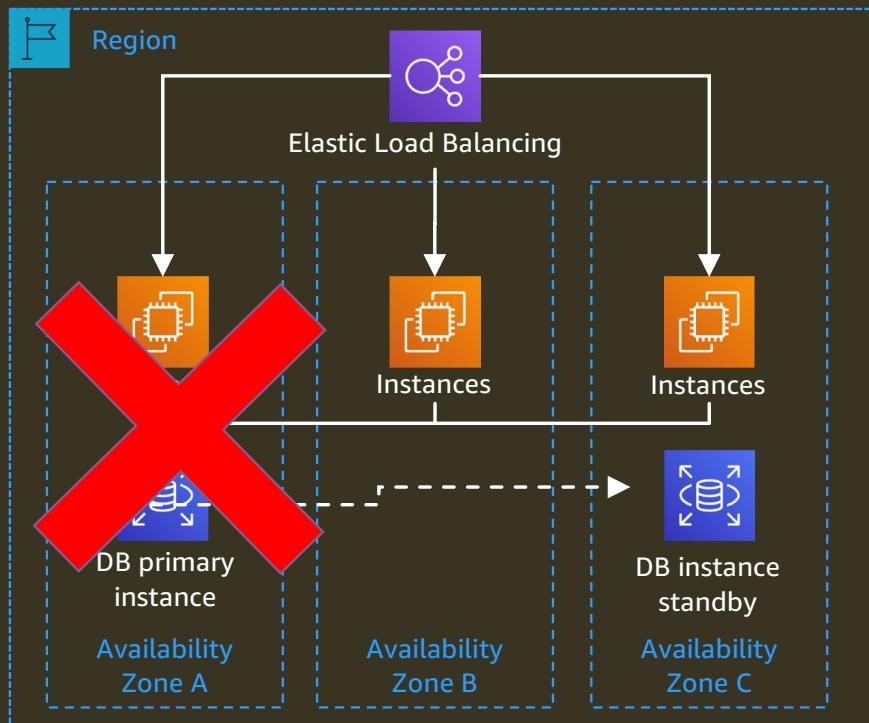
A Tale of Two Systems



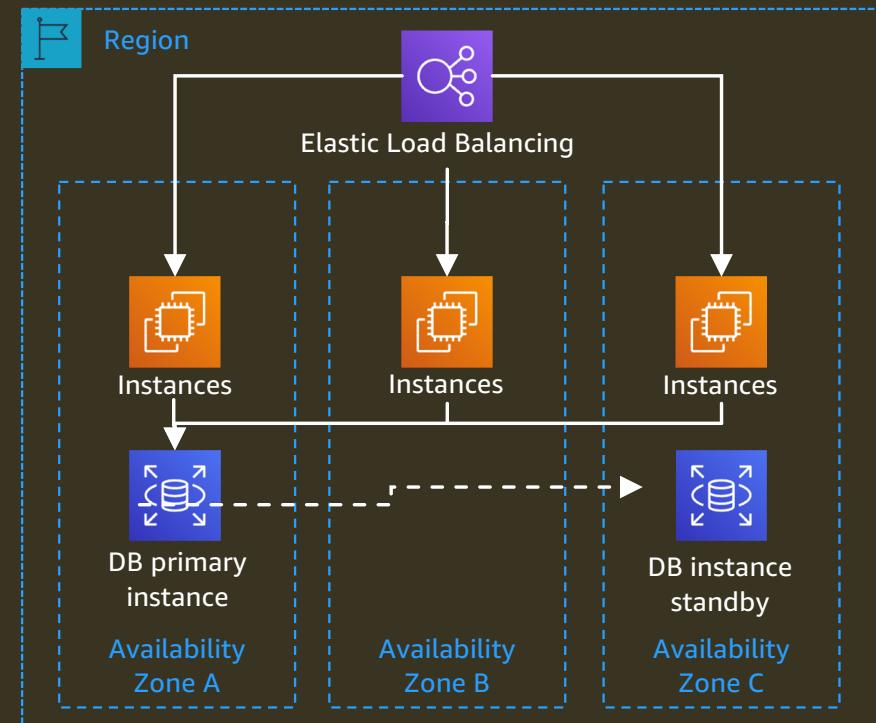
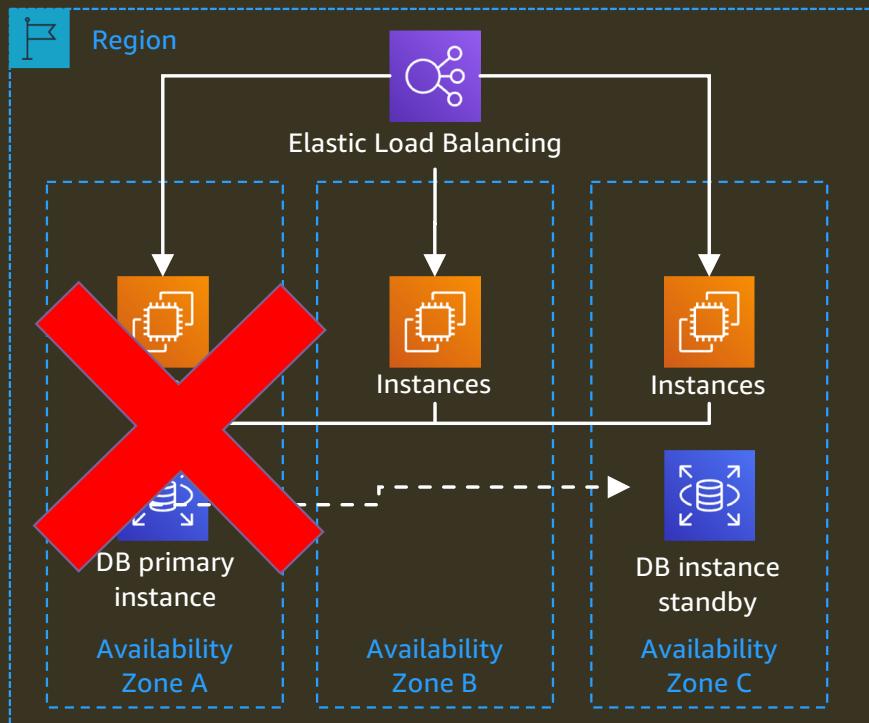
A Tale of Two Systems



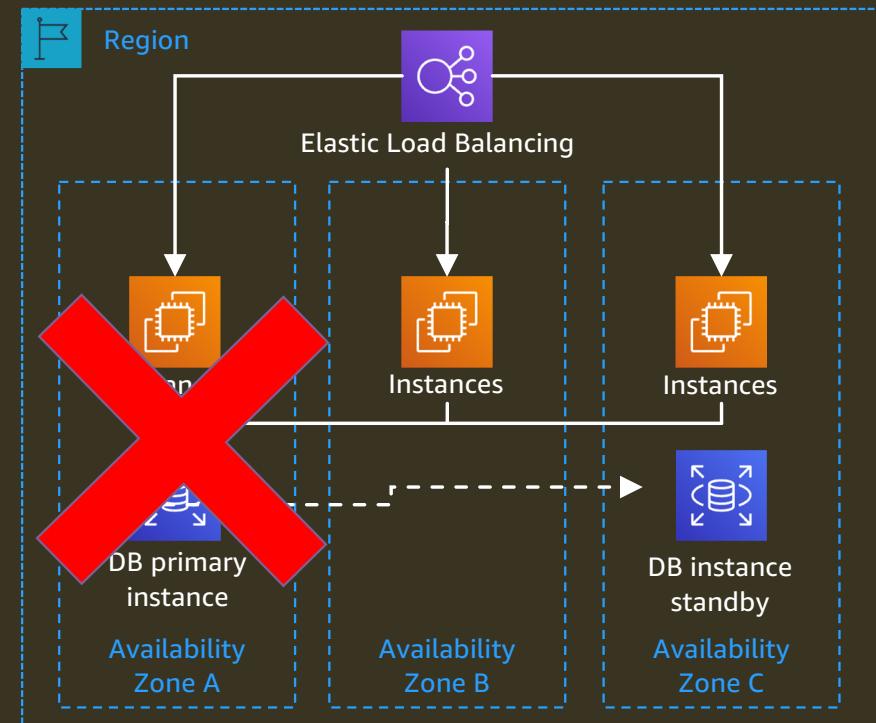
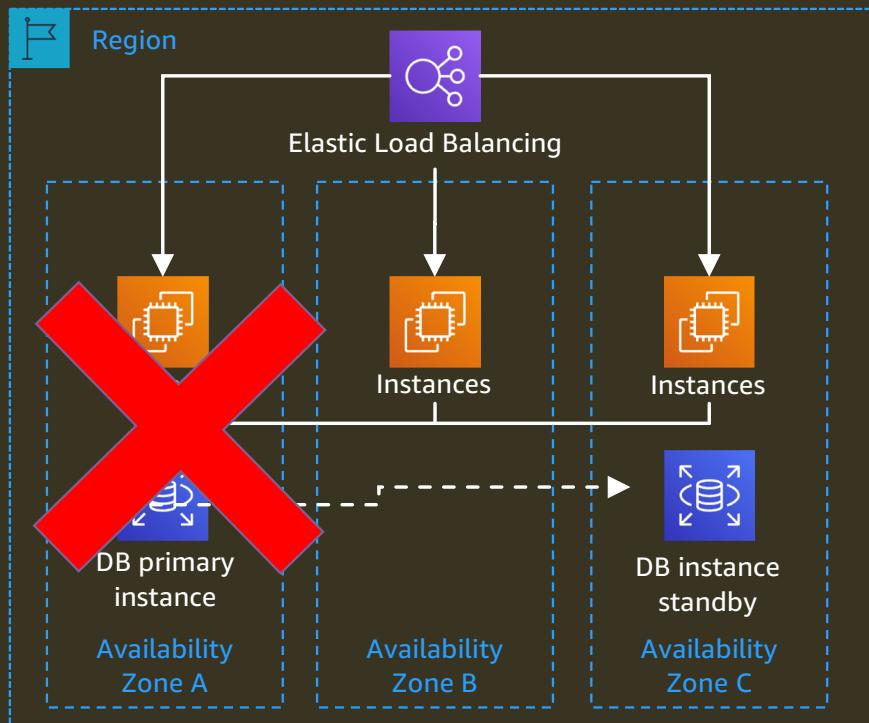
A Tale of Two Systems

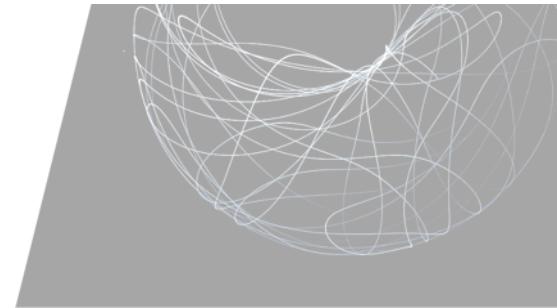


A Tale of Two Systems

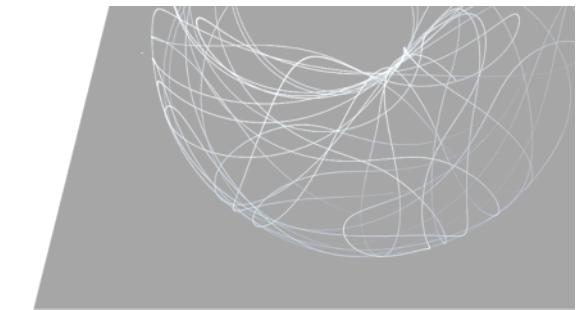
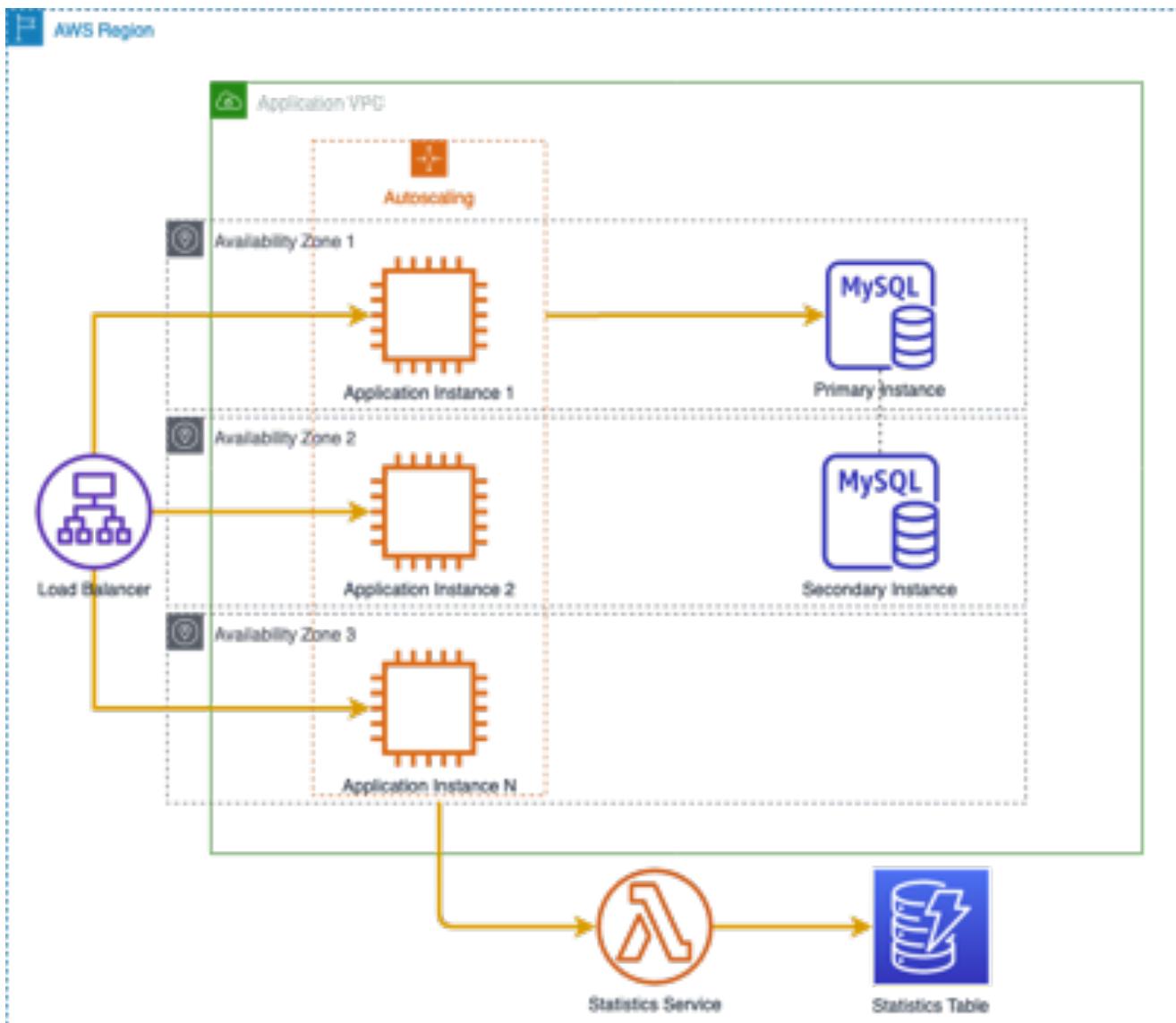


A Tale of Two Systems





So how do we confidently create resilient systems?

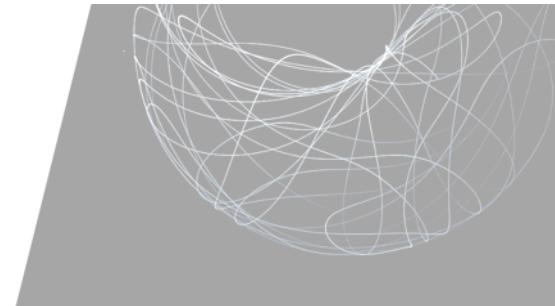


1. Failure domains
2. Fault isolation boundaries
3. Testing with chaos experiments

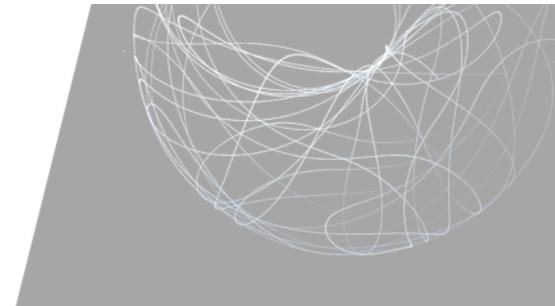
Failure Domains



Failure Domain



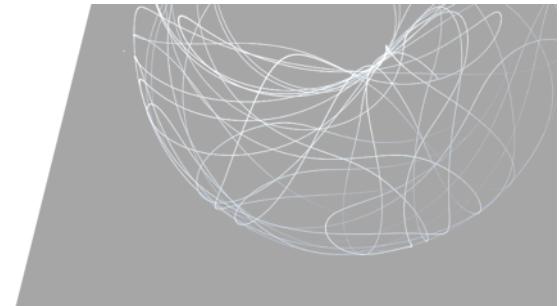
Failure Domain



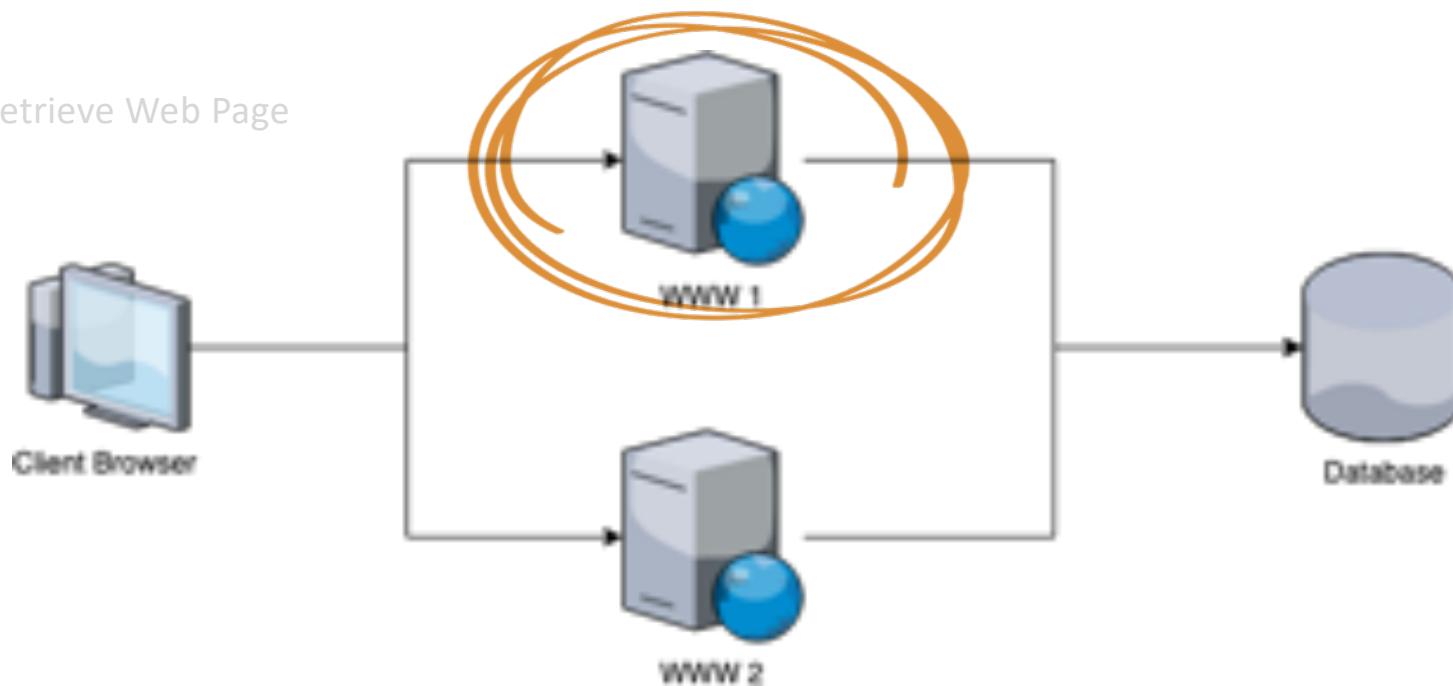
Scenario: Retrieve Web Page



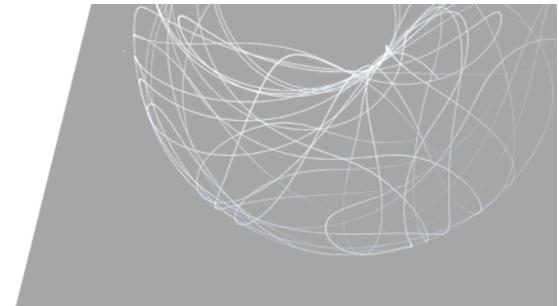
Failure Domain



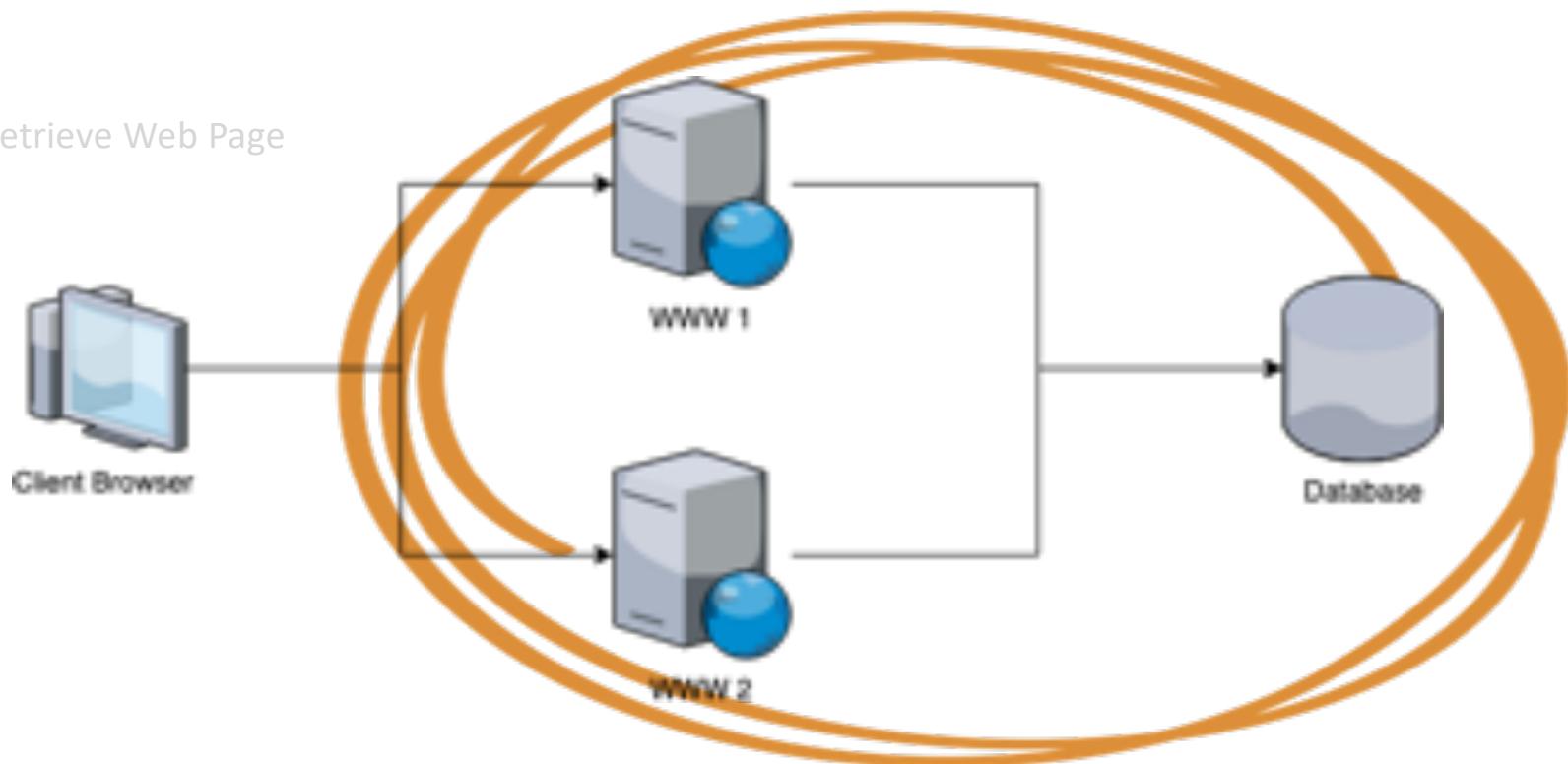
Scenario: Retrieve Web Page



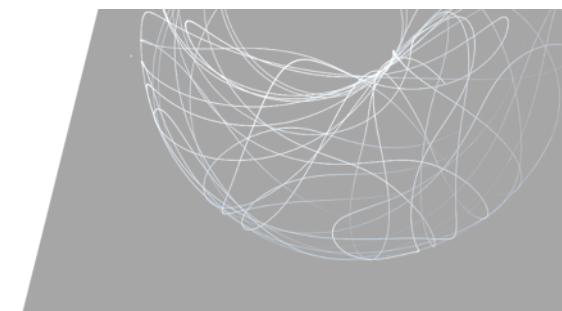
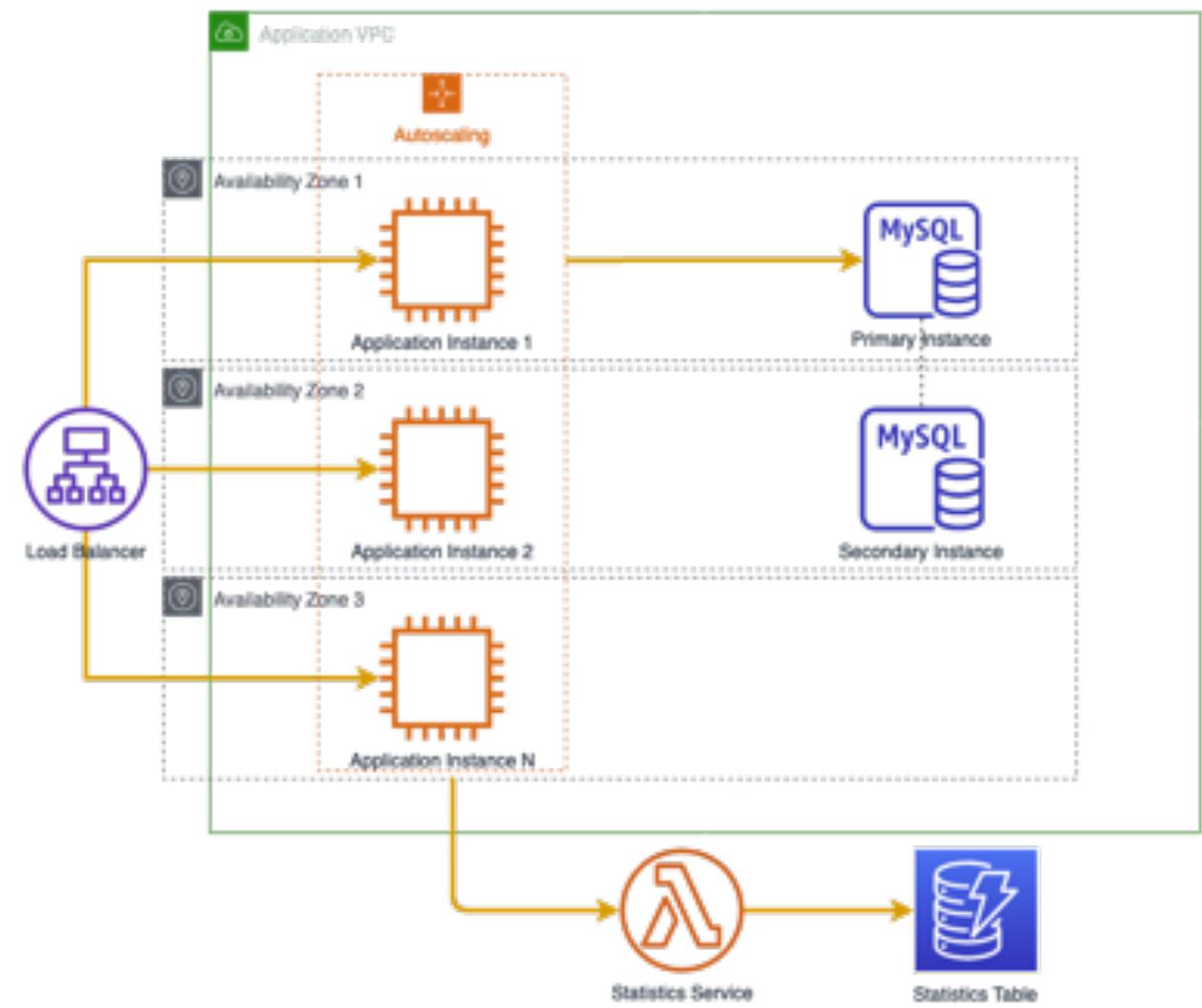
Failure Domain



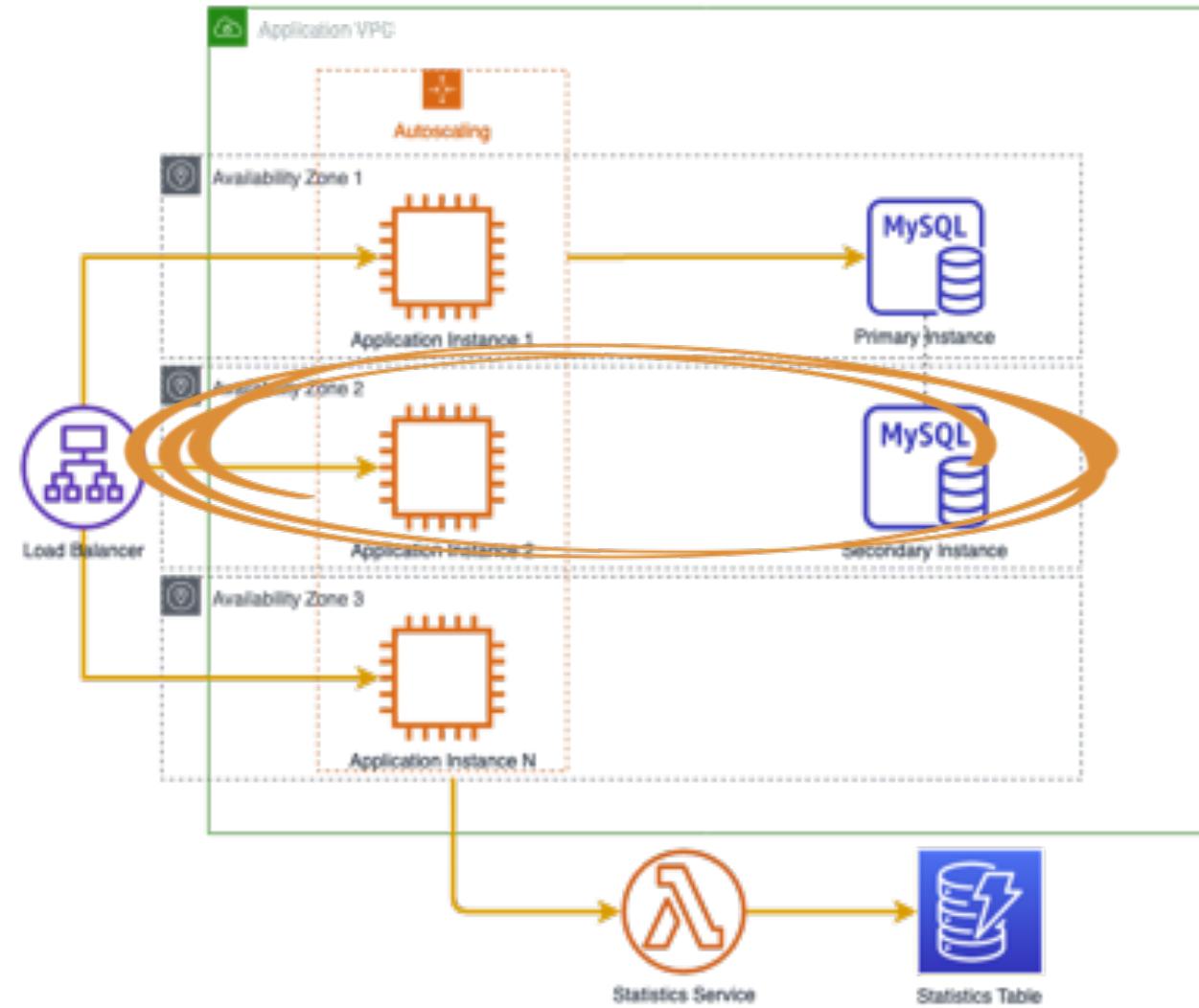
Scenario: Retrieve Web Page



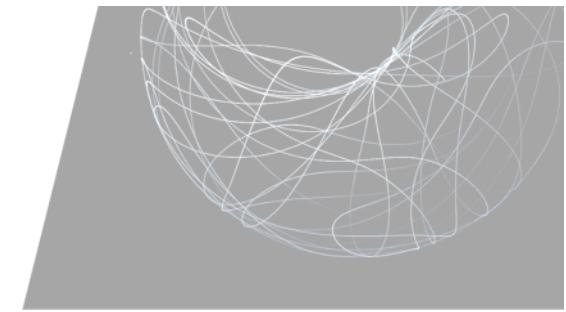
AWS Region



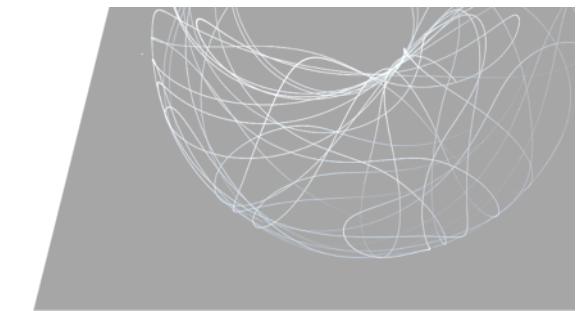
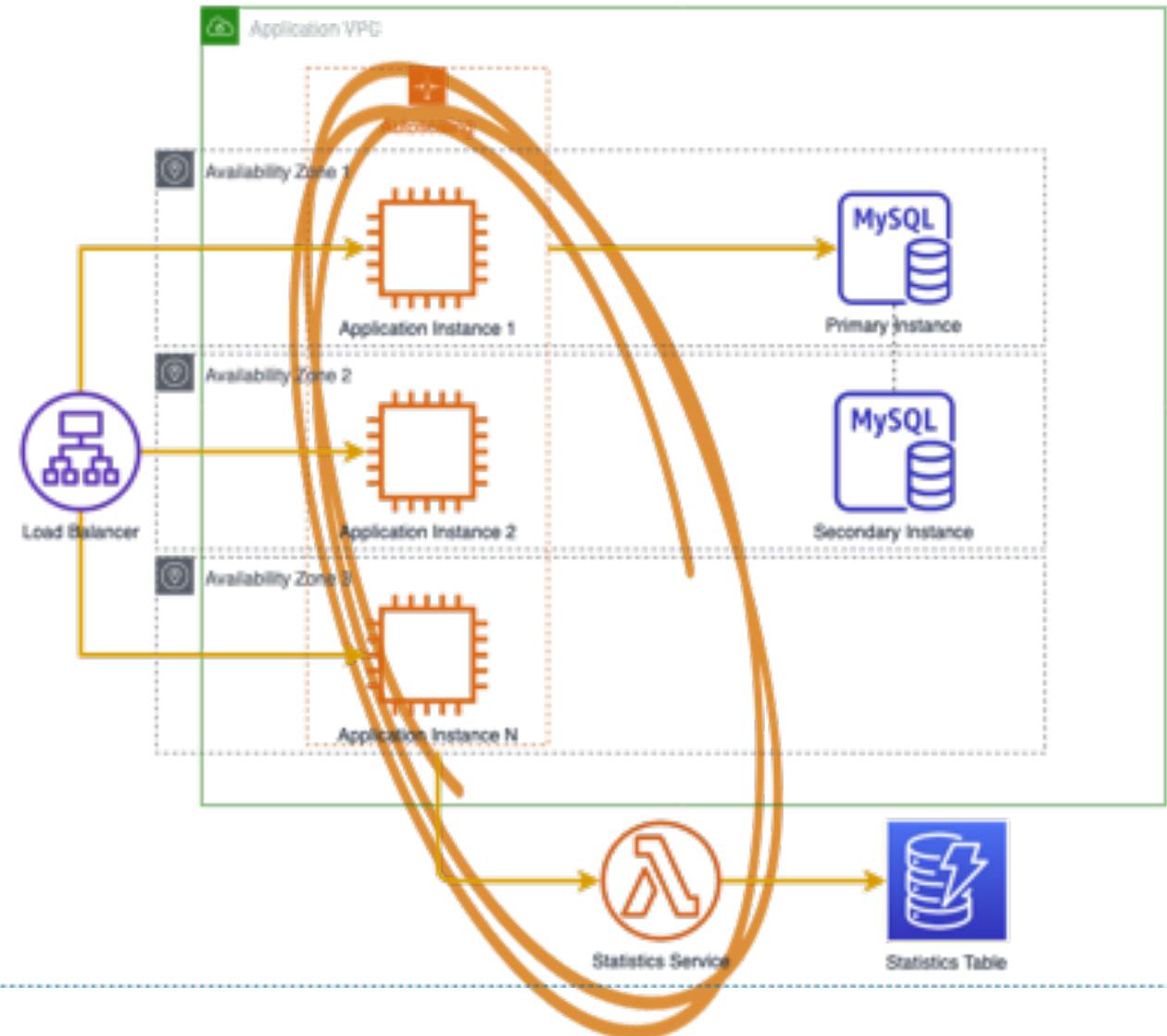
AWS Region



Failure domain of an AWS Availability Zone

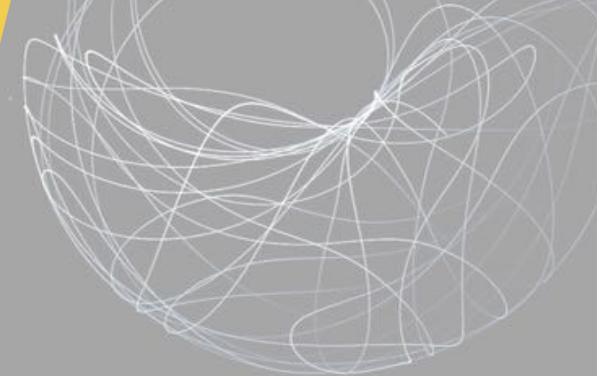


AWS Region

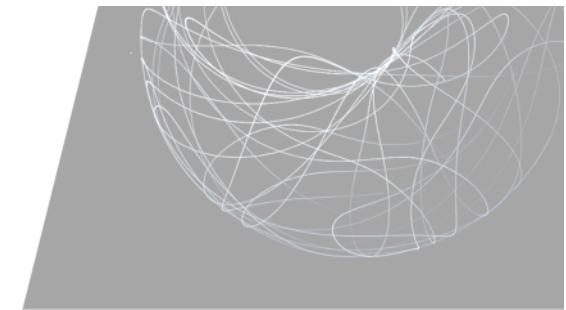


Failure domain of an AWS Service

Fault Isolation Boundaries



Failure-oriented Patterns

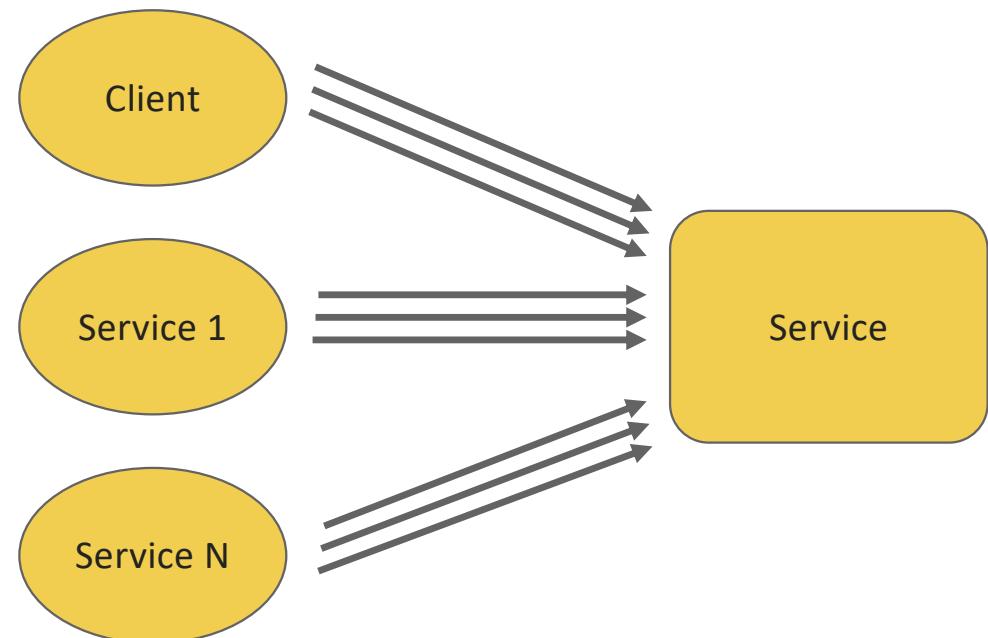


Client

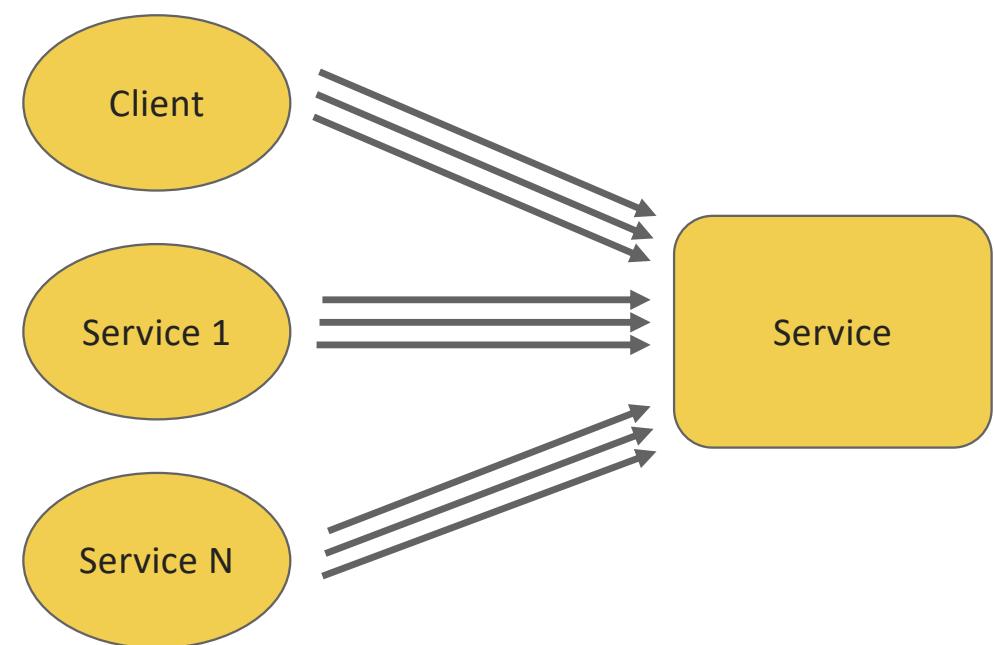
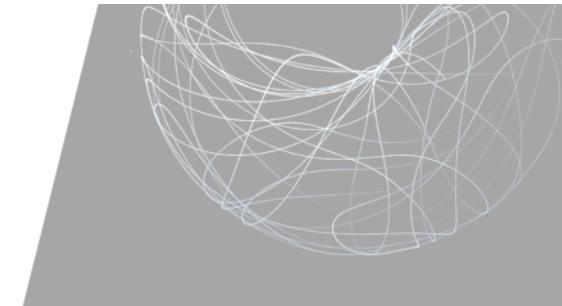
- Set timeouts
- Retries with backoff
 - Add jitter
 - Add retry limit
- Circuit breakers

Service

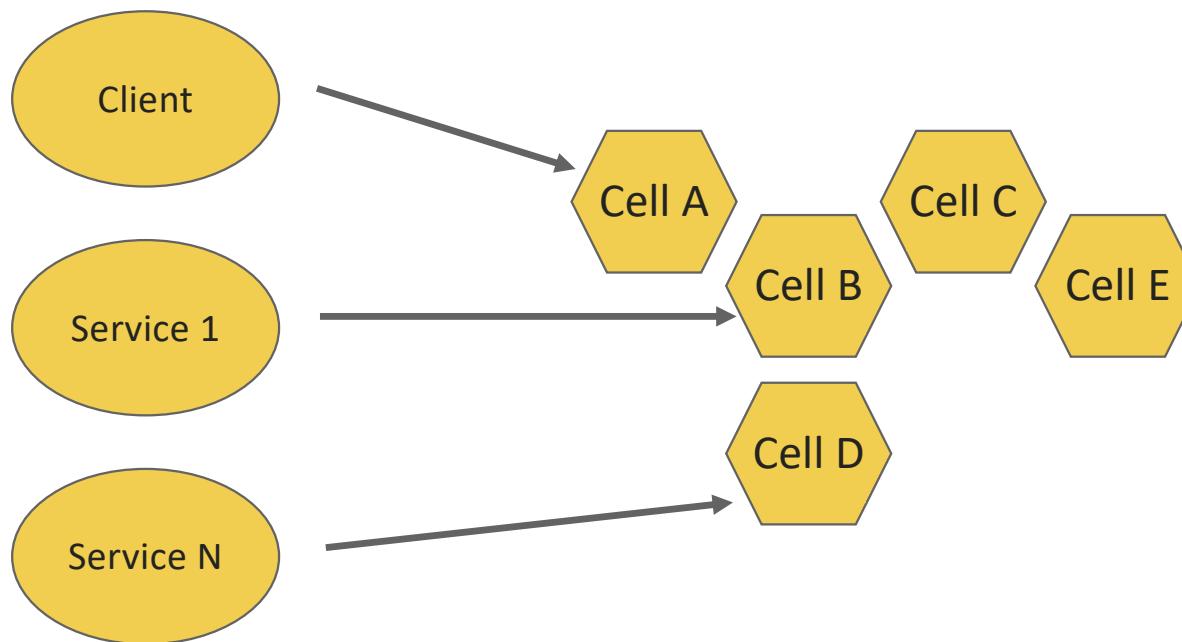
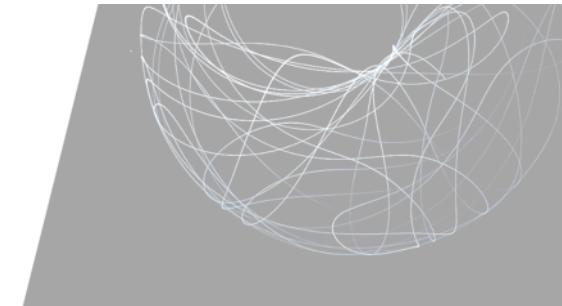
- Rate limit
- Load shedding



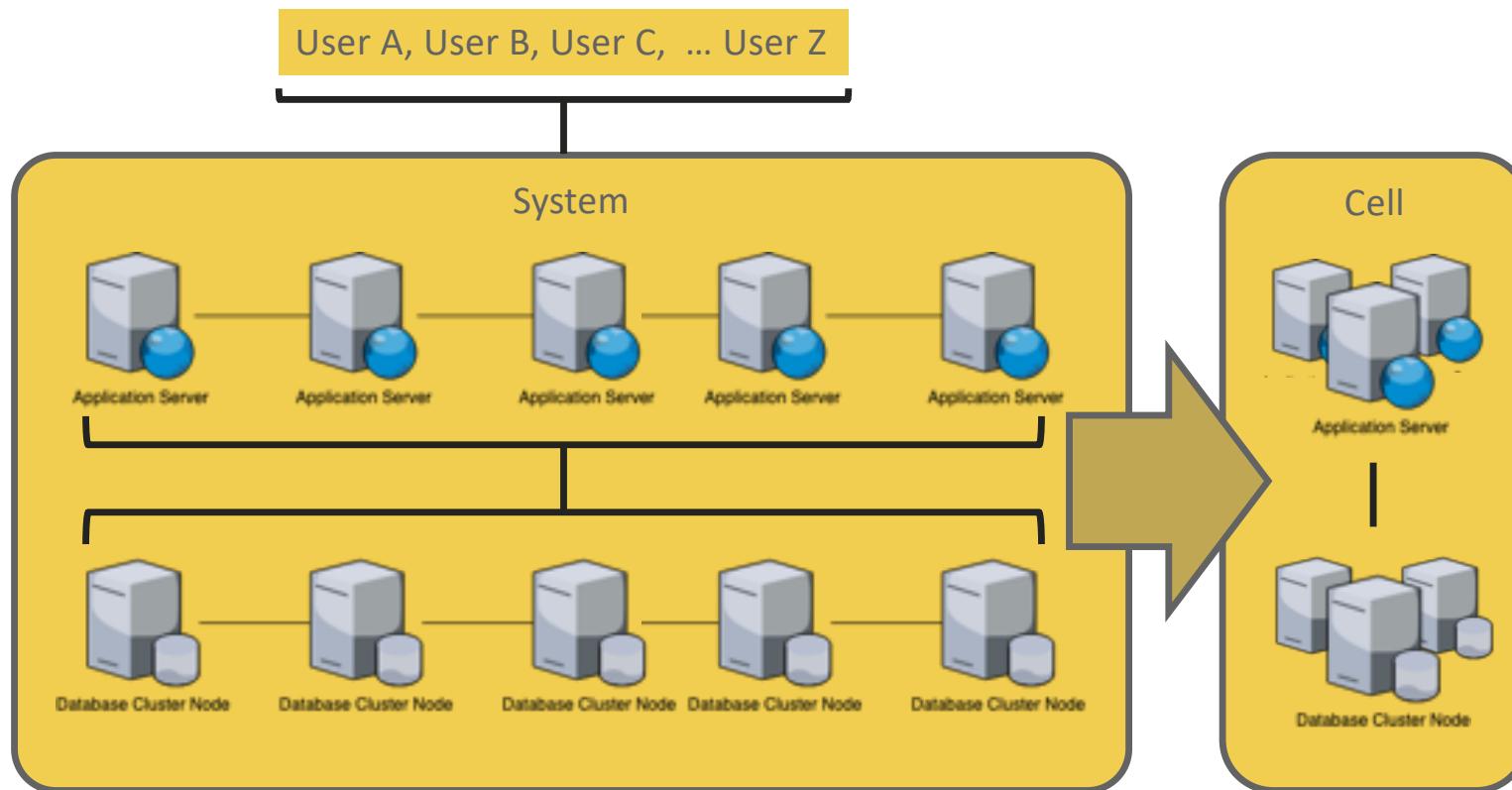
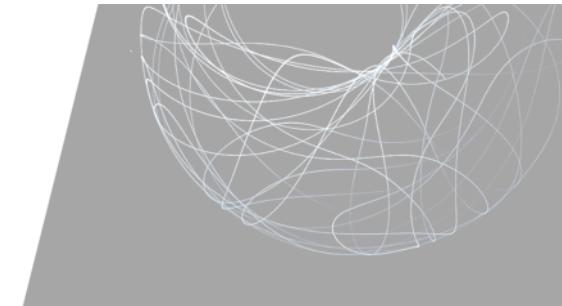
Failure-oriented Patterns



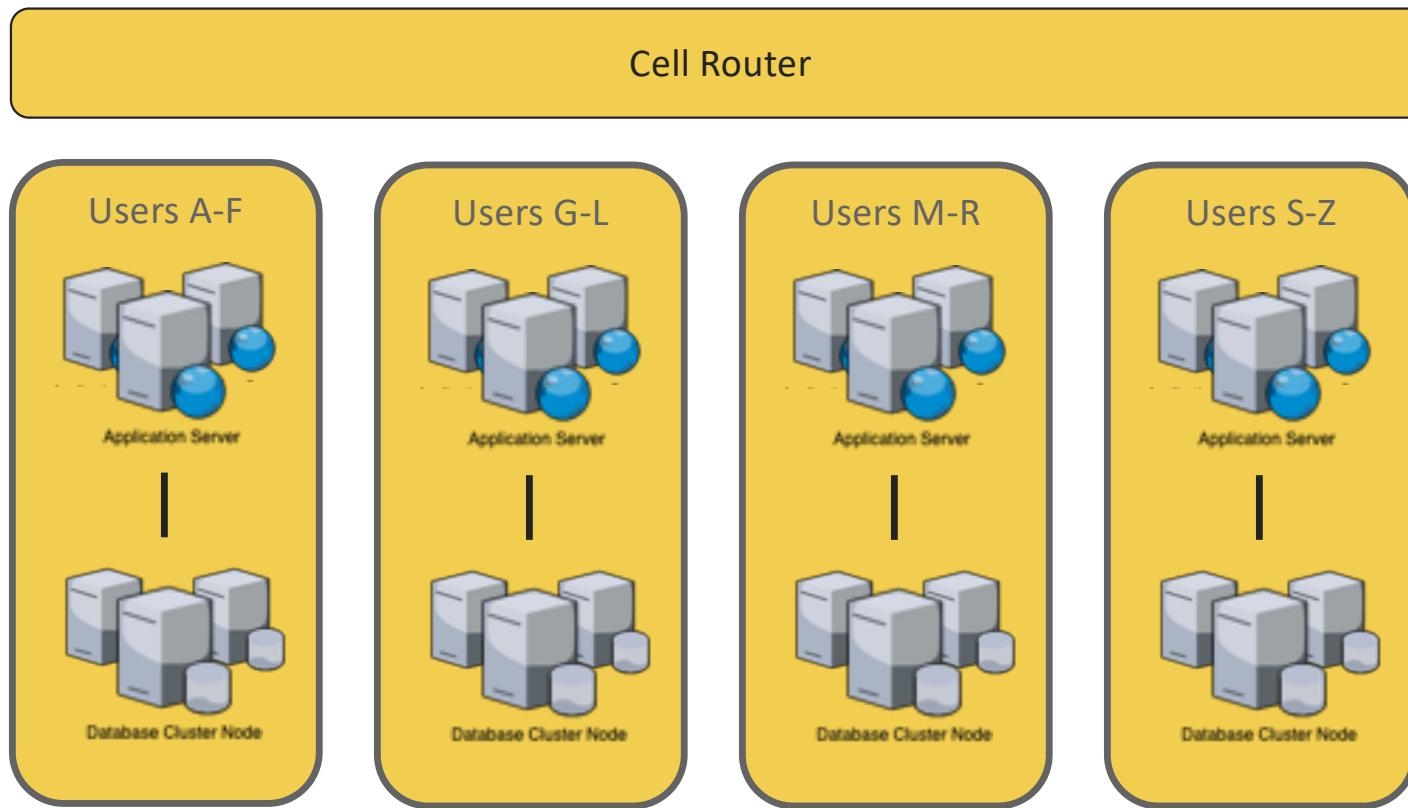
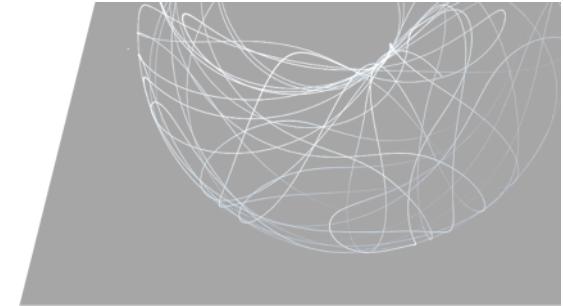
Limiting impact of failures with cells



From System to Cell

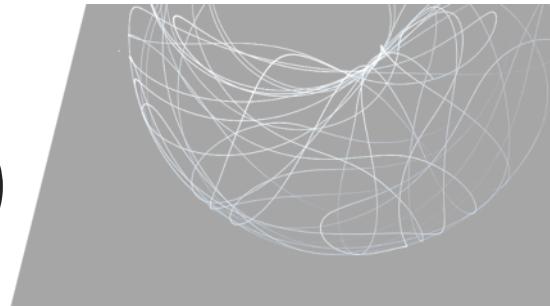


Cell-based Architecture



AWS Regions & Availability Zones (AZs)

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



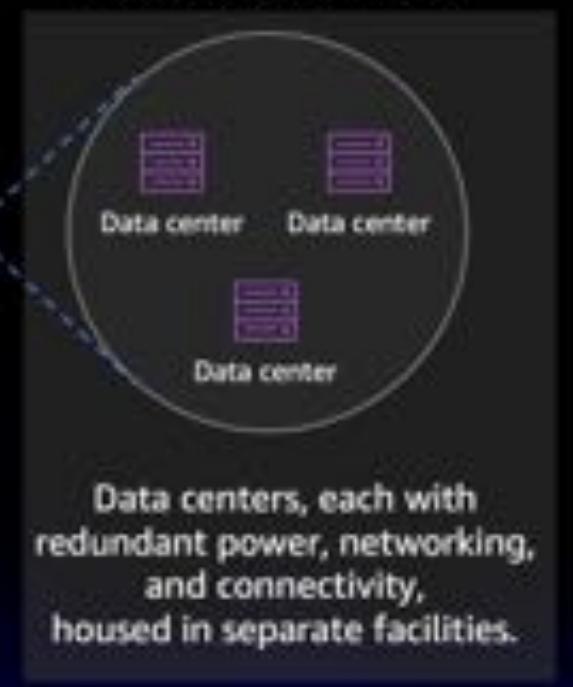
26 AWS Regions World-Wide



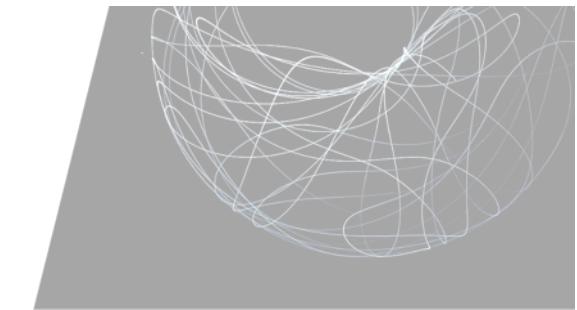
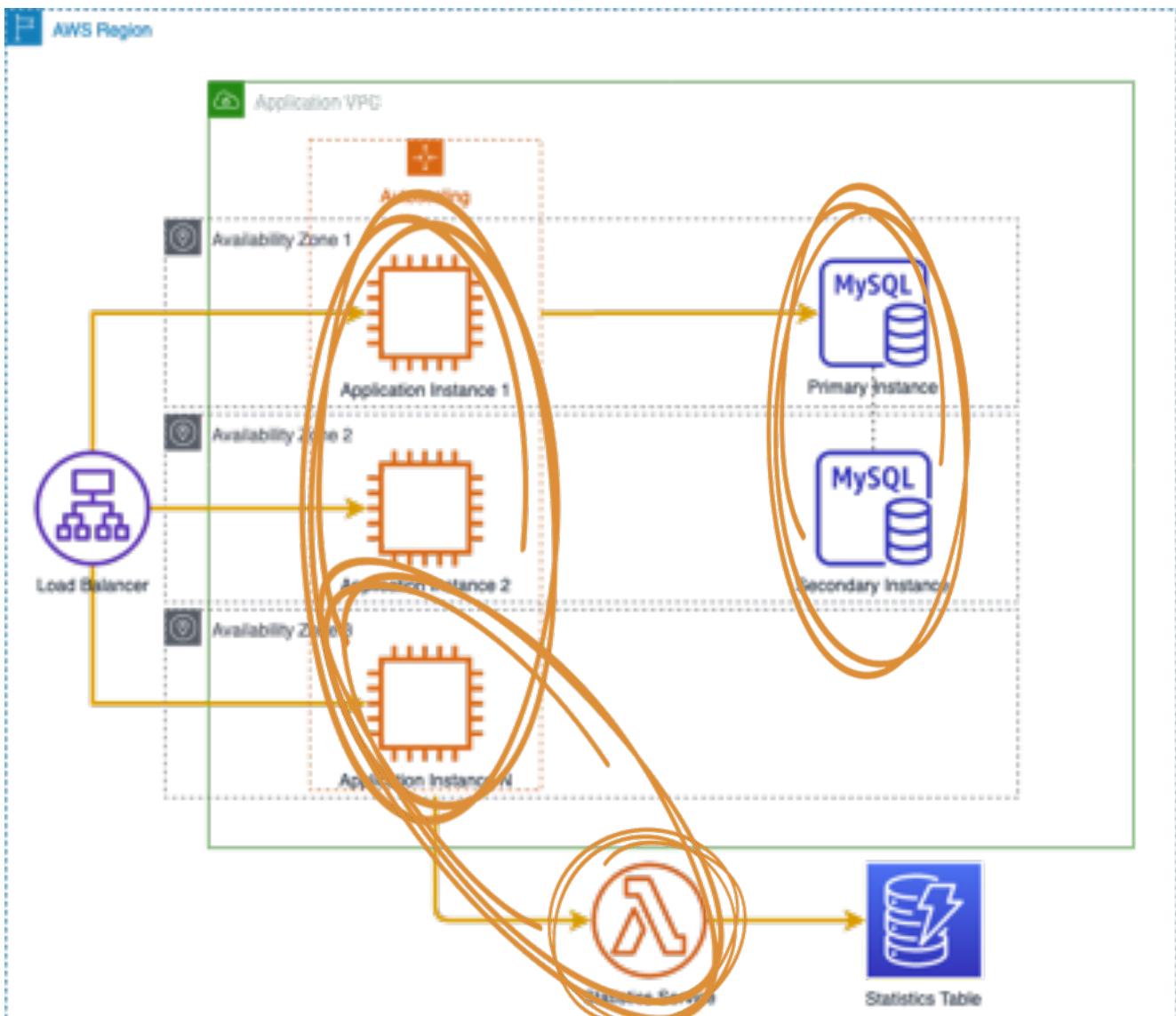
Each AWS Region has multiple AZs



Each AZ is one or more discrete Data Centers



AWS Regions Image copyright Amazon Web Services



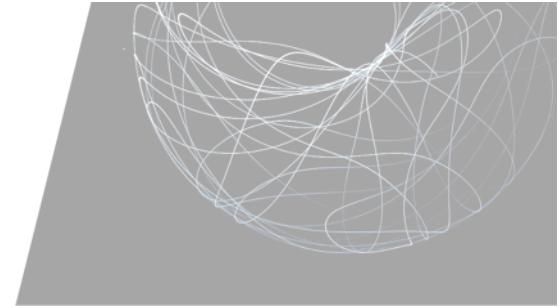
Fault isolation for
our system

Initial Tests for Chaos Engineering



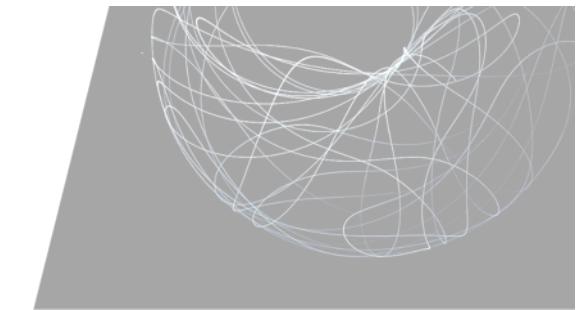
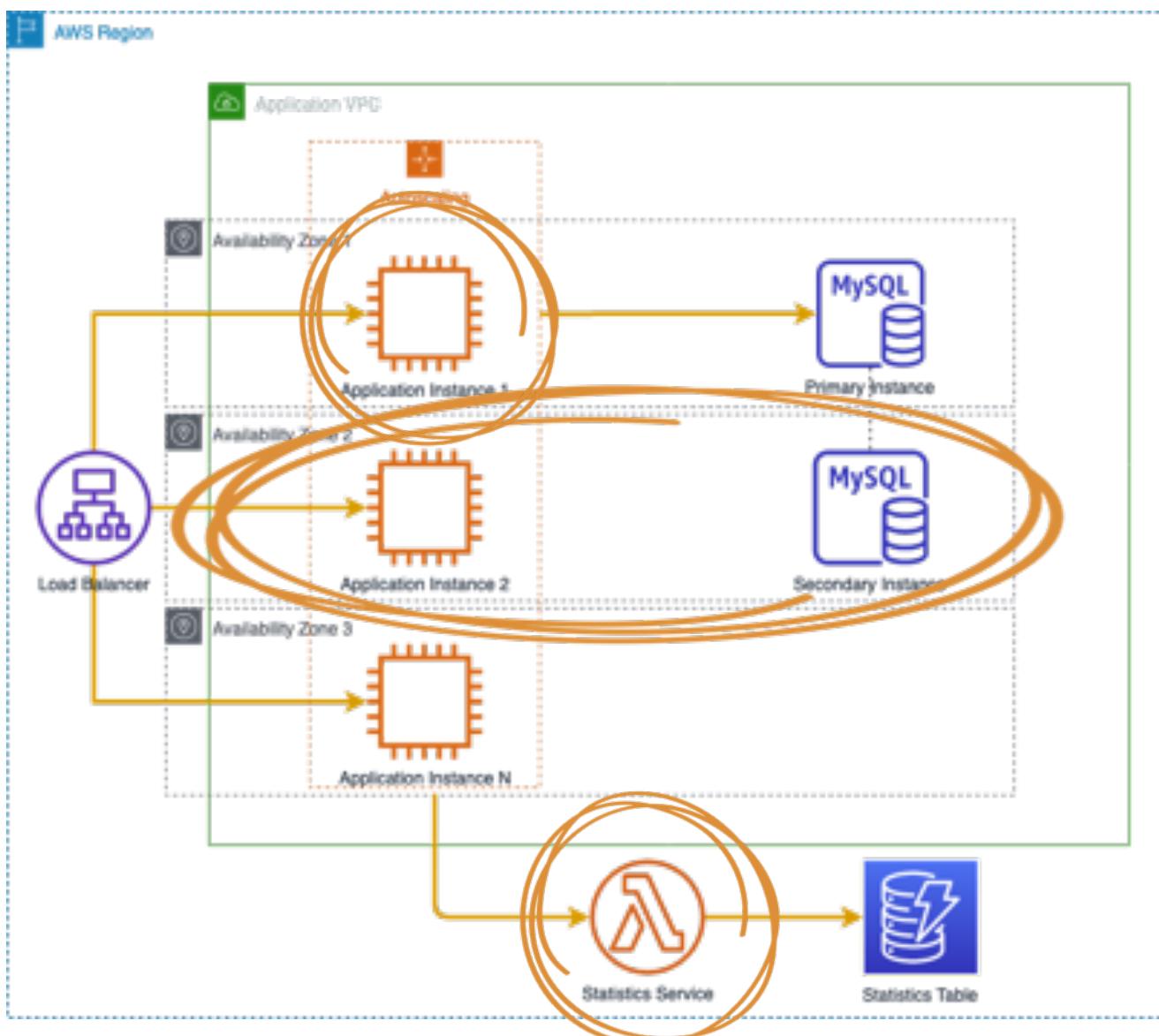
Disclaimer

Do not try this in Production



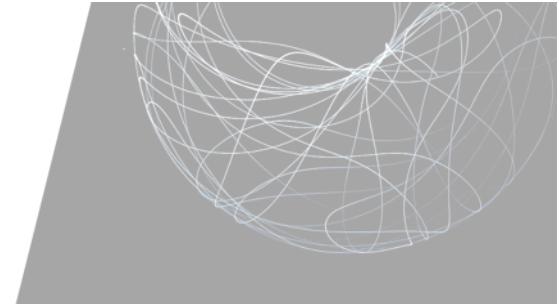
Initial Tests





- Zonal failure
- Centralized dependency
- Service dependency

Initial Tests: Zonal Resources



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

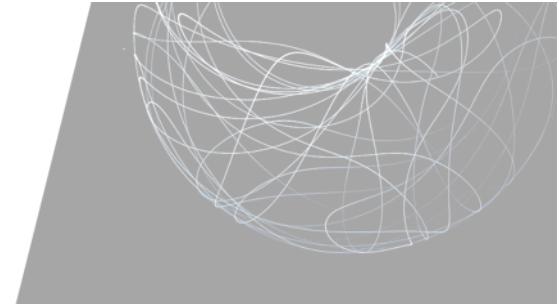
Terminate a virtual machine

```
$ 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



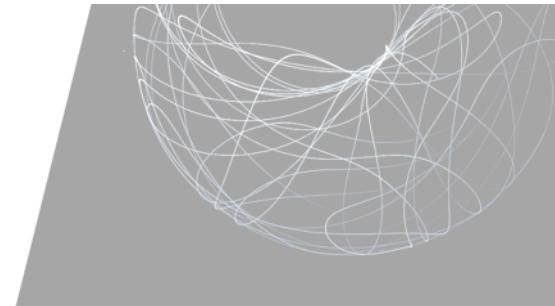
- Block all network traffic into / out of an availability zone

```
# 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

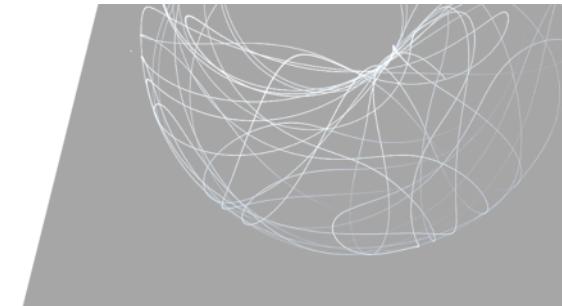
```
# Define the policy document
$ cat >/tmp/deny-lambda.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "lambda:*"
      ],
      "Resource": "*"
    }
  ]
}
EOF
```

```
$ 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

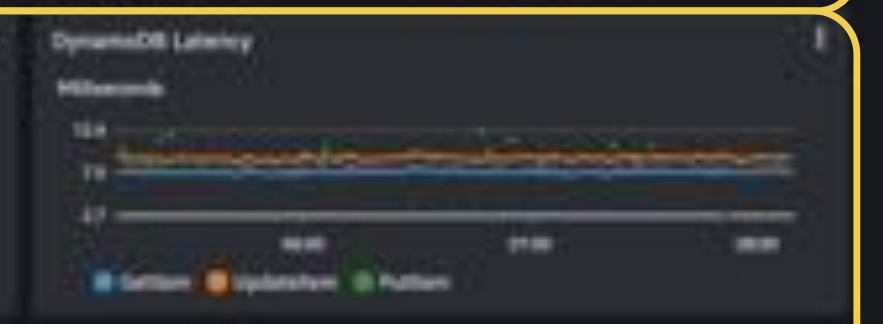
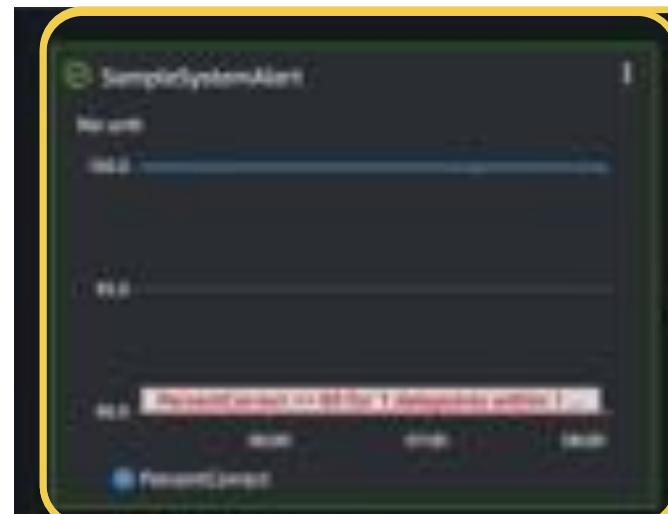
- Serverless
- Experiment templates
- Fault documents
- Integrated with Amazon CloudWatch



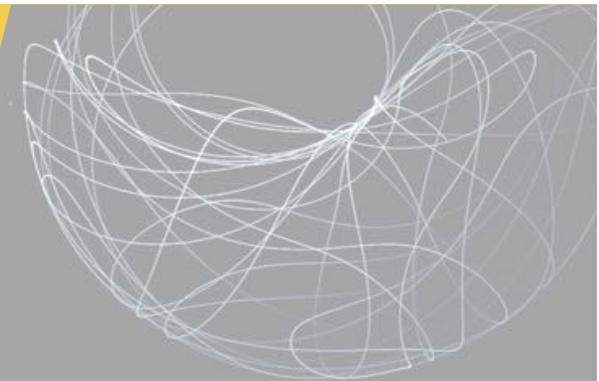
The Chaos Toolkit

- 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



EXT3BH2Xx1canPBn / Initial Test: Availability Zone Fault Isolation

Details

Experiment template ID:
EXT3BH2Xx1canPBnDescription:
Initial Test: Availability Zone Fault IsolationSAM role:
[FT33334Role](#)Stop conditions:
[SampledByEventAlert](#)Creation time:
March 20, 2022, 20:19:03 (UTC+00:00)Last update time:
March 21, 2022, 11:09:48 (UTC+00:00)Log group log destination:
[fl](#)[Actions](#) [Targets](#) [Export](#) [Tags](#) [Timeline](#)

Actions (2)

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

+ [Apply-NACLs / awsicom:start-automation-execution \(15 min\)](#)Start after [Assert-Steady-State](#)+ [Assert-Steady-State / awscloudwatch:assert-alarm-state](#)

Start at beginning of experiment

AZ-Network-Disruption-NACL

[Delete](#) [Actions](#) [View details >](#)

[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 | 867060631964 | |

Status: [Active](#)

Document Name - NACL-F15-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: 5 minutes Maximum duration the fault can exist for.

Supports Rollback

No. The NACL association is reverted.

Cancellation behaviour

The NACL association is reverted.

Output Parameters

EXP3S5fDfzdCjJMC6e

Actions

Details

Experiment ID
EXP3S5fDfzdCjJMC6eStart time
March 28, 2022, 09:50:30 (UTC +01:00)State
 RunningExperiment template ID
EX1584061499fbCreation time
March 28, 2022, 09:50:30 (UTC +01:00)End time
-MM role
FIFOQueue [?]Stop conditions
SimpleTermination [?]Log group log destination:
No [?]

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)

Start: After Assert-Steady-State

Running

+ Assert-Steady-State / aws:cloudwatchassert-alarm-state

Start: At beginning of experiment

Completed

EC2 > Target groups > sampl-ALBTa-1HOGT9TX9X8UW

sampl-ALBTa-1HOGT9TX9X8UW

Amazon CloudWatch Load Balancing (us-west-1:807005631794) target group sampl-ALBTa-1HOGT9TX9X8UW (arn:aws:elasticloadbalancing:us-west-1:807005631794:targetgroup/sampl-ALBTa-1HOGT9TX9X8UW/79485452c213)

Actions

Details

| | | | | | |
|-----------------|---------------------------|------------------|----------------------|-------|----------|
| Target type | Protocol / Port | Protocol version | VPC | | |
| instance | HTTP:80 | HTTP1 | vpc-ef294030f12e0e1e | | |
| IP address type | Load balancer | | | | |
| IPv4 | sample-app-1-192-168-1-10 | | | | |
| Total targets | Healthy | Unhealthy | Unused | Total | Draining |
| 3 | 2 | 1 | 0 | 0 | 0 |

Targets Monitoring Health checks Attributes Tags

Registered targets (3)

| Instance ID | Name | Port | Zone | Health status | Health status details |
|--------------|-------------------|------|------------|---------------|-----------------------|
| 10.0.2.15:80 | sample-app-server | 80 | eu-west-1a | healthy | |
| 10.0.2.17:80 | sample-app-server | 80 | eu-west-1a | unhealthy | Request timed out |
| 10.0.2.18:80 | sample-app-server | 80 | eu-west-1b | healthy | |

Deregister Register targets

« 1 »

sample-ALBTa-1H0GT9TX9X8UW

Actions ▾

arn:aws:elasticloadbalancing:eu-west-1:96700381196:targetgroups/sample-ALBTa-1H0GT9TX9X8UW/716854532/uv2130

Details

Target type:
Instance

Protocol / Port:
HTTP: 80

Protocol version:
HTTP1

VPC:
vpc-0f384010915e01e12

IP address type:
IPv4

Load balancer:
sample-AppS-1515604ewL2ow [0]

Total targets:
4

Healthy
2

Unhealthy
0

Unused
0

Initial
1

Draining
1

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (4)



Unregister

Register targets

Filter results by property or value

| Instance ID | Name | Port | Zone | Health status | Health status details |
|---------------------|-------------------|------|------------|------------------------------|--------------------------------------|
| 106784859ac0bb0d | sample-app-server | 80 | eu-west-1a | Green healthy | |
| 106800119ad42775e | sample-app-server | 80 | eu-west-1a | Yellow draining | Target deregistration is in progress |
| 1068140140315caea7f | sample-app-server | 80 | eu-west-1a | Yellow initial | Target registration is in progress |
| 1068160140315caea7d | sample-app-server | 80 | eu-west-1b | Green healthy | |

sample-ALBTa-1H0GT9TX9X8UW

Actions ▾

arn:aws:elasticloadbalancing:eu-west-1:987003811966:targetgroups/sample-ALBTa-1H0GT9TX9X8UW/796854532/uv2130

Details

Target type:
Instance

Protocol / Port:
HTTP: 80

Protocol version:
HTTP1

VPC:
vpc-0f384c02d52cc0be19

IP address type:
IPv4

Load balancer:
sample-App-1315604wLzw [0]

Total targets:
4

Healthy
 2

Unhealthy
 1

Unused
 0

Initial
 0

Draining
 1

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (4)

Q Filter results by property or value



Sort targets

Register targets

◀ 1 ▶ ⏪ ⏩

| Instance ID | Name | Port | Zone | Health status | Health status details |
|---------------------|-------------------|------|------------|------------------------|--------------------------------------|
| 1-0d784965ac0e6ed | sample-app-server | 80 | eu-west-1a | Healthy | |
| 1-06602011ead42775e | sample-app-server | 80 | eu-west-1a | Draining | Target deregistration is in progress |
| 1-032f6e40315caef7 | sample-app-server | 80 | eu-west-1a | Unhealthy | Request timed out |
| 1-05a49444a055a44d | sample-app-server | 80 | eu-west-1b | Healthy | |

Refresh Actions

Details

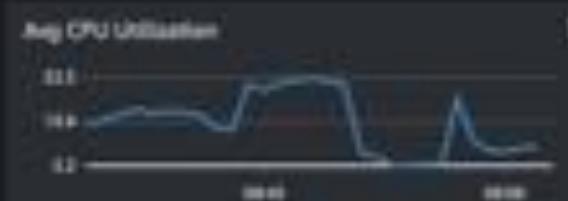
| | | | | | | | |
|--------------------------------|--------------------------------------|------------|--------------------------------------|----------|--------------|------------------------|------------------------|
| Experiment ID | EXP355fDfzdCjJMC6e | Start time | March 28, 2022, 09:50:30 (UTC+01:00) | Status | Completed | Experiment template ID | ext1benchmarks |
| Creation time | March 28, 2022, 09:50:30 (UTC+01:00) | End time | March 28, 2022, 09:58:35 (UTC+01:00) | Last run | F5DD9B8e [8] | Stop conditions | sampledtermination [2] |
| Ci/cd pipeline log destination | None [2] | | | | | | |

Actions Targets Tags Timeline Stop conditions

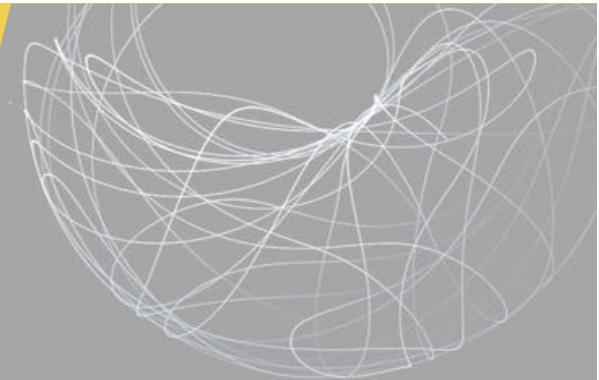
Actions (2)

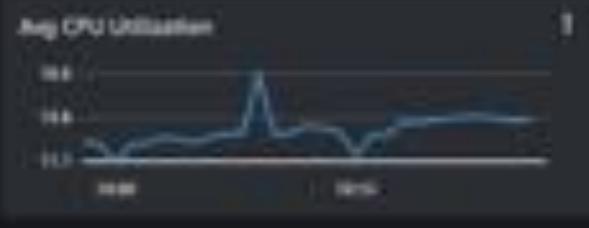
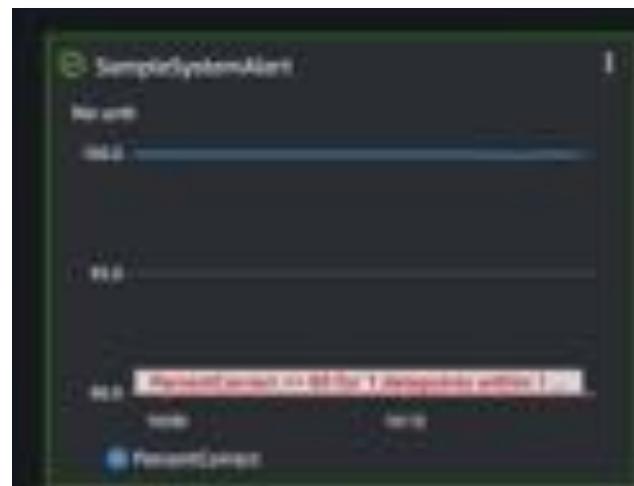
View your experiment template actions, action duration, and action response.

- **Apply-NACLs / awscloudstart-automation-execution (15 min)** Completed
Start: After Assert-Steady-State
- **Assert-Steady-State / awscloudwatchassert-alarm-state** Completed
Start: At beginning of experiment



Simulate Region Failure





EC2 > Target groups / sampl-ALBTa-1HOGT9TX9X8UW

sampl-ALBTa-1HOGT9TX9X8UW

arn:aws:elasticloadbalancing:us-west-1:867005837968:targetgroup/sampl-ALBTa-1HOGT9TX9X8UW/794854532c2150

Actions

Details

| | | | | | |
|-----------------|--|------------------|--|---------|----------|
| Target type | Protocol : Port | Protocol version | VPC | | |
| instance | HTTP:80 | HTTP1 | arn:aws:vpc:us-west-1:867005837968:subnet/123456789012345678 | | |
| IP address type | Load balancer | | | | |
| IPv4 | sample-app-1.vvtttqchewl20m.us-west-1.elasticbeanstalk.com | | | | |
| Total targets | Healthy | Unhealthy | Unused | Initial | Draining |
| 3 | 3 | 0 | 0 | 0 | 0 |

Targets Monitoring Health checks Attributes Tags

Registered targets (3)

Filter resources by property or value

| Instance ID | Name | Port | Zone | Health status | Health status details |
|---------------------|-------------------|------|------------|---------------|-----------------------|
| i-0d35eaa5f0f46efc9 | sample-app-server | 80 | eu-west-1a | healthy | |
| i-05f7ff0d9e4a4477 | sample-app-server | 80 | eu-west-1b | healthy | |
| i-08ca25d09fd793460 | sample-app-server | 80 | eu-west-1c | healthy | |

Actions

Targets **Monitoring** **Health checks** **Attributes** **Tags**

Registered targets (3)

Filter resources by property or value

| Instance ID | Name | Port | Zone | Health status | Health status details |
|---------------------|-------------------|------|------------|---------------|-----------------------|
| i-0d35eaa5f0f46efc9 | sample-app-server | 80 | eu-west-1a | healthy | |
| i-05f7ff0d9e4a4477 | sample-app-server | 80 | eu-west-1b | healthy | |
| i-08ca25d09fd793460 | sample-app-server | 80 | eu-west-1c | healthy | |

```
version: 1.0
service: http-service-as-waf-service
description: A
  To simulate service that has been experiment with apply an http policy with a http placement to intercept the service control plane
  by respond with a non-2xx response.

http://127.0.0.1:8080
  https://127.0.0.1:8081
  https://127.0.0.1:8082

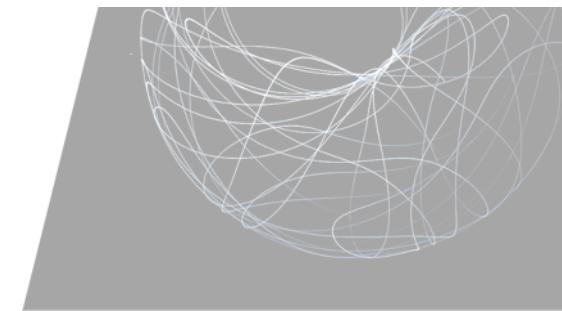
port 8080
  http://127.0.0.1:8080/service/executePolicy?serviceId=1&path=/&method=GET
  https://127.0.0.1:8080/service/executePolicy?serviceId=1&path=/&method=POST

port 8081
  https://127.0.0.1:8081/service/executePolicy?serviceId=1&path=/&method=PUT
  https://127.0.0.1:8081/service/executePolicy?serviceId=1&path=/&method=DELETE

port 8082
  https://127.0.0.1:8082/service/executePolicy?serviceId=1&path=/&method=PATCH
  https://127.0.0.1:8082/service/executePolicy?serviceId=1&path=/&method=HEAD

actions:
  - type: action
    name: attack-policy
    arguments:
      - type: python
        module: database.ban.actions
        name: attack_rule_policy
        arguments:
          - name: attack_policy_and
            value: attack_and
          - name: attack_and_value
            value: attack_and_value
    enabled: true
  - type: action
    name: attack-policy-or
    arguments:
      - type: python
        module: database.ban.actions
        name: attack_rule_policy
        arguments:
          - name: attack_policy_or
            value: attack_or
          - name: attack_or_value
            value: attack_or_value

  - type: action
    name: attack-policy-not
    arguments:
      - type: python
        module: database.ban.actions
        name: attack_rule_policy
        arguments:
          - name: attack_policy_not
            value: attack_not
          - name: attack_not_value
            value: attack_not_value
```



```
#!/bin/bash -e
# This script will run tests for all supported platforms
# and architectures. It will also build the project
# and generate documentation.

# Set the project root directory
PROJECT_ROOT=$(cd $(dirname $0) && cd .. && pwd)

# Set the build directory
BUILD_DIR=$PROJECT_ROOT/build

# Set the source directory
SOURCE_DIR=$PROJECT_ROOT/src

# Set the test directory
TEST_DIR=$PROJECT_ROOT/test

# Set the documentation directory
DOC_DIR=$PROJECT_ROOT/doc

# Set the configuration file
CONFIG_FILE=$PROJECT_ROOT/config.json

# Set the build type
BUILD_TYPE="Release"

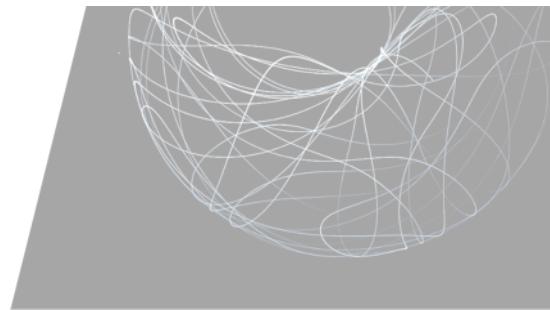
# Set the compiler
COMPILER="gcc"

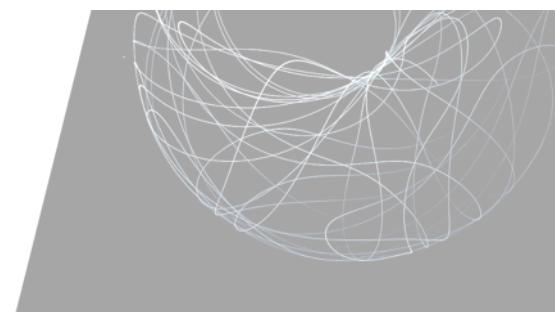
# Set the architecture
ARCHITECTURE="x86_64"

# Set the platform
PLATFORM="Linux"

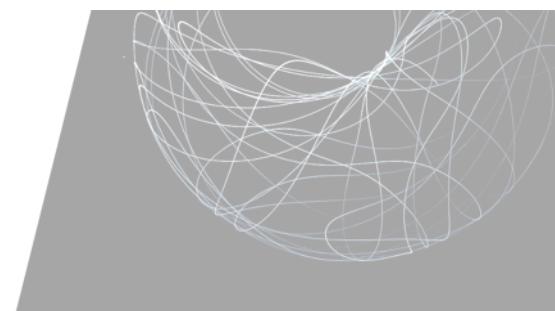
# Set the test suite
TEST_SUITE="unit"

# Set the test timeout
TEST_TIMEOUT=300
```





```
graph TD
    A[Attack] --> B[Attack every policy]
    B --> C[Attack]
    C --> D[Attack every policy]
    D --> E[Attack]
    E --> F[Attack every policy]
    F --> G[Attack]
    G --> H[Attack]
    H --> I[Attack]
    I --> J[Attack]
    J --> K[Attack]
    K --> L[Attack]
    L --> M[Attack]
    M --> N[Attack]
    N --> O[Attack]
    O --> P[Attack]
    P --> Q[Attack]
    Q --> R[Attack]
    R --> S[Attack]
    S --> T[Attack]
    T --> U[Attack]
    U --> V[Attack]
    V --> W[Attack]
    W --> X[Attack]
    X --> Y[Attack]
    Y --> Z[Attack]
    Z --> AA[Attack]
    AA --> BB[Attack]
    BB --> CC[Attack]
    CC --> DD[Attack]
    DD --> EE[Attack]
    EE --> FF[Attack]
    FF --> GG[Attack]
    GG --> HH[Attack]
    HH --> II[Attack]
    II --> JJ[Attack]
    JJ --> KK[Attack]
    KK --> LL[Attack]
    LL --> MM[Attack]
    MM --> NN[Attack]
    NN --> OO[Attack]
    OO --> PP[Attack]
    PP --> QQ[Attack]
    QQ --> RR[Attack]
    RR --> SS[Attack]
    SS --> TT[Attack]
    TT --> UU[Attack]
    UU --> VV[Attack]
    VV --> WW[Attack]
    WW --> XX[Attack]
    XX --> YY[Attack]
    YY --> ZZ[Attack]
    ZZ --> AAA[Attack]
    AAA --> BBB[Attack]
    BBB --> CCC[Attack]
    CCC --> DDD[Attack]
    DDD --> EEE[Attack]
    EEE --> FFF[Attack]
    FFF --> GGG[Attack]
    GGG --> HHH[Attack]
    HHH --> III[Attack]
    III --> JJJ[Attack]
    JJJ --> KKK[Attack]
    KKK --> LLL[Attack]
    LLL --> MLL[Attack]
    MLL --> NLL[Attack]
    NLL --> OLL[Attack]
    OLL --> PLL[Attack]
    PLL --> QLL[Attack]
    QLL --> RLL[Attack]
    RLL --> SLL[Attack]
    SLL --> TLL[Attack]
    TLL --> ULL[Attack]
    ULL --> VLL[Attack]
    VLL --> WLL[Attack]
    WLL --> XLL[Attack]
    XLL --> YLL[Attack]
    YLL --> ZLL[Attack]
    ZLL --> AAAA[Attack]
    AAAA --> BBBB[Attack]
    BBBB --> CCCC[Attack]
    CCCC --> DDDD[Attack]
    DDDD --> EEEE[Attack]
    EEEE --> FFFF[Attack]
    FFFF --> GGGG[Attack]
    GGGG --> HHHH[Attack]
    HHHH --> IIII[Attack]
    IIII --> JJJJ[Attack]
    JJJJ --> KKKK[Attack]
    KKKK --> LLLL[Attack]
    LLLL --> MLLL[Attack]
    MLLL --> NLLL[Attack]
    NLLL --> OLLL[Attack]
    OLLL --> PLLL[Attack]
    PLLL --> QLLL[Attack]
    QLLL --> RLLL[Attack]
    RLLL --> SLLL[Attack]
    SLLL --> TLLL[Attack]
    TLLL --> ULLL[Attack]
    ULLL --> VLLL[Attack]
    VLLL --> WLLL[Attack]
    WLLL --> XLLL[Attack]
    XLLL --> YLLL[Attack]
    YLLL --> ZLLL[Attack]
    ZLLL --> AAAAA[Attack]
    AAAAA --> BBBBB[Attack]
    BBBBB --> CCCCC[Attack]
    CCCCC --> DDDDD[Attack]
    DDDDD --> EEEEE[Attack]
    EEEEE --> FFFFF[Attack]
    FFFFF --> GGGGG[Attack]
    GGGGG --> HHHHH[Attack]
    HHHHH --> IIIII[Attack]
    IIIII --> JJJJJ[Attack]
    JJJJJ --> KKKKK[Attack]
    KKKKK --> LLLLL[Attack]
    LLLLL --> MLLLL[Attack]
    MLLLL --> NLLLL[Attack]
    NLLLL --> OLLLL[Attack]
    OLLLL --> PLLLL[Attack]
    PLLLL --> QLLLL[Attack]
    QLLLL --> RLLLL[Attack]
    RLLLL --> SLLLL[Attack]
    SLLLL --> TLLLL[Attack]
    TLLLL --> ULLLL[Attack]
    ULLLL --> VLLLL[Attack]
    VLLLL --> WLLLL[Attack]
    WLLLL --> XLLLL[Attack]
    XLLLL --> YLLLL[Attack]
    YLLLL --> ZLLLL[Attack]
    ZLLLL --> AAAAAA[Attack]
    AAAAAA --> BBBBBB[Attack]
    BBBBBB --> CCCCCC[Attack]
    CCCCCC --> DDDDDD[Attack]
    DDDDDD --> EEEEEEE[Attack]
    EEEEEEE --> FFFFFF[Attack]
    FFFFFF --> GGGGGG[Attack]
    GGGGGG --> HHHHHH[Attack]
    HHHHHH --> IIIIII[Attack]
    IIIIII --> JJJJJJ[Attack]
    JJJJJJ --> KKKKKK[Attack]
    KKKKKK --> LLLLLL[Attack]
    LLLLLL --> MLLLLL[Attack]
    MLLLLL --> NLLLLL[Attack]
    NLLLLL --> OLLLLL[Attack]
    OLLLLL --> PLLLLL[Attack]
    PLLLLL --> QLLLLL[Attack]
    QLLLLL --> RLLLLL[Attack]
    RLLLLL --> SLLLLL[Attack]
    SLLLLL --> TLLLLL[Attack]
    TLLLLL --> ULLLLL[Attack]
    ULLLLL --> VLLLLL[Attack]
    VLLLLL --> WLLLLL[Attack]
    WLLLLL --> XLLLLL[Attack]
    XLLLLL --> YLLLLL[Attack]
    YLLLLL --> ZLLLLL[Attack]
    ZLLLLL --> AAAAAA
    AAAAAA --> BBBBBB
    BBBBBB --> CCCCCC
    CCCCCC --> DDDDDD
    DDDDDD --> EEEEEEE
    EEEEEEE --> FFFFFF
    FFFFFF --> GGGGGG
    GGGGGG --> HHHHHH
    HHHHHH --> IIIIII
    IIIIII --> JJJJJJ
    JJJJJJ --> KKKKKK
    KKKKKK --> LLLLLL
    LLLLLL --> MLLLLL
    MLLLLL --> NLLLLL
    NLLLLL --> OLLLLL
    OLLLLL --> PLLLLL
    PLLLLL --> QLLLLL
    QLLLLL --> RLLLLL
    RLLLLL --> SLLLLL
    SLLLLL --> TLLLLL
    TLLLLL --> ULLLLL
    ULLLLL --> VLLLLL
    VLLLLL --> WLLLLL
    WLLLLL --> XLLLLL
    XLLLLL --> YLLLLL
    YLLLLL --> ZLLLLL
```



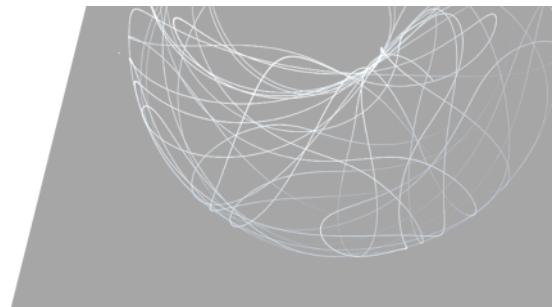
```
def main():
    # Load the dataset
    dataset = load_breast_cancer()
    X, y = dataset.data, dataset.target
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

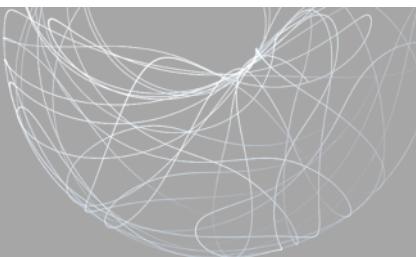
    # Standardize the features
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)

    # Train a SVC model
    model = SVC()
    model.fit(X_train, y_train)

    # Evaluate the model
    accuracy = model.score(X_test, y_test)
    print(f"Model accuracy: {accuracy:.2f}")

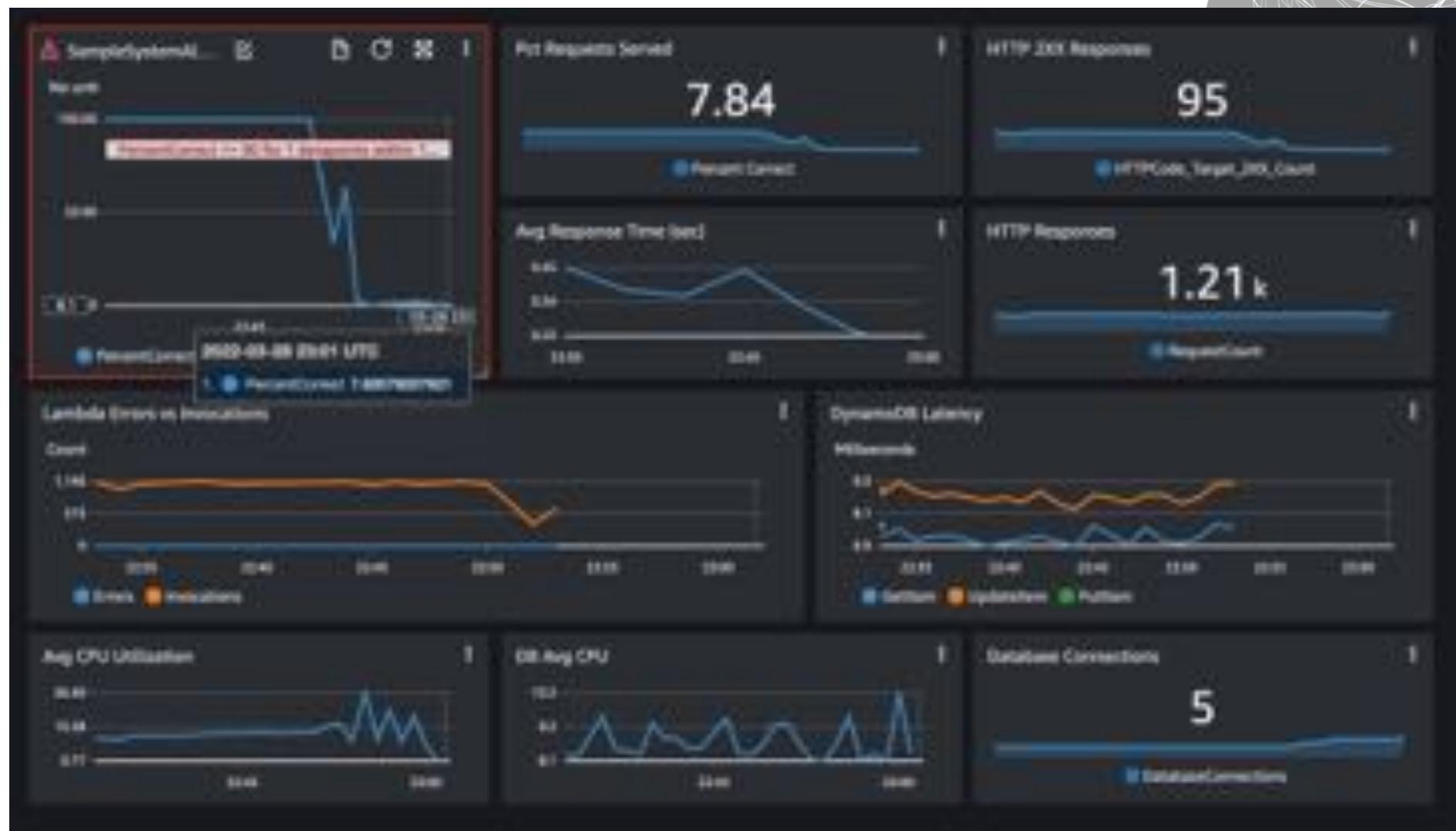
if __name__ == "__main__":
    main()
```





```
git: ~ / c / sample - aws - workload / fail - lambda [on master] 1: 12: 29 ] chaos run experiment.yaml -- rollback-strategy always --journal-path exp-fail-lambda-run-001.log
```

```
[2022-03-28 23:58:39 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] Probes: 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...
```



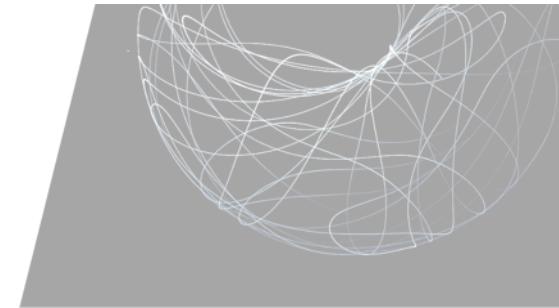
Details

| | | | | | |
|-------------------------|---|---|---|--|---|
| Target type Instance | Protocol : Port HTTP:80 | Protocol version HTTP1 | VPC vpc-0f25402ef8cc0e12 | | |
| IP address type IPv4 | Load balancer: sample-App-15V15804WLJWU [5] | | | | |
| Total targets: 5 | Healthy 0 | Unhealthy 2 | Unused 0 | Initial 0 | Draining 3 |

Targets Monitoring Health checks Attributes Tags

Registered targets (5)

| Instance ID | Name | Port | Zone | Health status | Health status details |
|-------------------|-------------------|------|------------|--|--|
| AQ8t8d4f7d8f8166d | sample-app-server | 80 | eu-west-1c | Unhealthy | Health checks failed with these codes: [500] |
| 4-0818a473217aee | sample-app-server | 80 | eu-west-1a | Draining | Target deregistration is in progress |
| 4-0818a474d130460 | sample-app-server | 80 | eu-west-1b | Unhealthy | Health checks failed with these codes: [500] |
| 4-05795a0e94a9f7 | sample-app-server | 80 | eu-west-1b | Draining | Target deregistration is in progress |
| 4-0818a47505156d | sample-app-server | 80 | eu-west-1c | Draining | Target deregistration is in progress |



```
2022-03-28 23:58:36 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 23:58:48 INFO] Steady state hypothesis is met!
2022-03-28 23:58:48 INFO] Playing your experiment's method now...
2022-03-28 23:58:48 INFO] Action: attach-deny-policy
2022-03-28 23:58:48 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 00:06:09 [CRITICAL] Steady state probe 'check-system-error-rate' is not in the given tolerance so failing this experiment
2022-03-29 00:06:09 WARNING] Rollbacks were explicitly requested to be played
2022-03-29 00:06:09 INFO] Let's rollback...
2022-03-29 00:06:09 INFO] Rollback: detach-deny-policy
2022-03-29 00:06:09 INFO] Action: detach-deny-policy
2022-03-29 00:06:09 INFO] Experiment ended with status: deviated
2022-03-29 00:06:09 INFO] The steady-state has deviated, a weakness may have been discovered
```

Chaitin

| | | | |
|-----------------|--------------------------------|------------------|----------------------------|
| Target type | Protocol Port | Protocol version | VPC |
| Instance | HTTP-80 | HTTP1 | sgp-0f236402d4f86cc0fe [5] |
| IP address type | Load balancer | | |
| IPv4 | sample-Appl-15V1500446LJ0M [5] | | |

Total targets: Healthy: Unhealthy: Unused: Total: Draining:

| | | | | | |
|---|-----|-----|-----|-----|-----|
| 5 | ◎ 1 | ◎ 1 | ◎ 0 | ◎ 1 | ◎ 2 |
|---|-----|-----|-----|-----|-----|

Targets Monitoring Health checks Activities Targets

Registered targets (5)

6

Benzidine

Register targets

Filter comments by similarity or relatedness

← → ←

| Instance ID | Name | Port | Zone | Health status | Health status details |
|----------------------------------|-------------------|------|------------|------------------------|--|
| 1Q07rx86xa55555678 | sample-app-server | 80 | us-west-1a | Initial | Target registration is in progress |
| 4Q340000000000000000000000000000 | sample-app-server | 80 | us-west-1a | draining | Target deregistration is in progress |
| 4Q805299999999999999999999999999 | sample-app-server | 80 | us-west-1c | unhealthy | Health checks failed with these codes: [500] |
| 4Q8721a829e511111 | sample-app-server | 80 | us-west-1b | draining | Target deregistration is in progress |
| 4Q874011111111111111111111111111 | sample-app-server | 80 | us-west-1a | healthy | |



Details

| | | | |
|-----------------|-------------------------------|------------------|---------------------------|
| Target type | Protocol / Port | Protocol version | VPC |
| Instance | HTTP:80 | HTTP1 | vpc-0f26402ef8f6cc0fe [5] |
| IP address type | Load balancer | | |
| IPv4 | sample-AppL-15N15004WLZEW [5] | | |
| Total targets | Healthy | Unhealthy | Unused |
| 5 | 3 | 0 | 0 |
| Initial | | | Draining |
| 0 | | | 2 |

[Targets](#) | [Monitoring](#) | [Health checks](#) | [Attributes](#) | [Tags](#)

Registered targets (5)

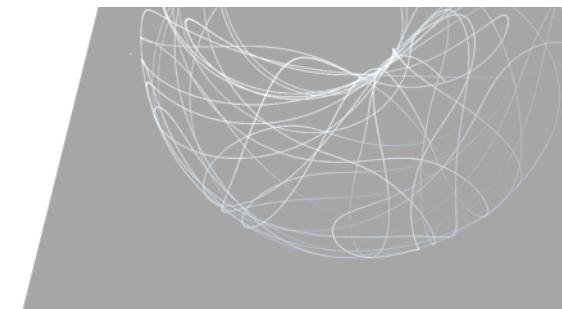
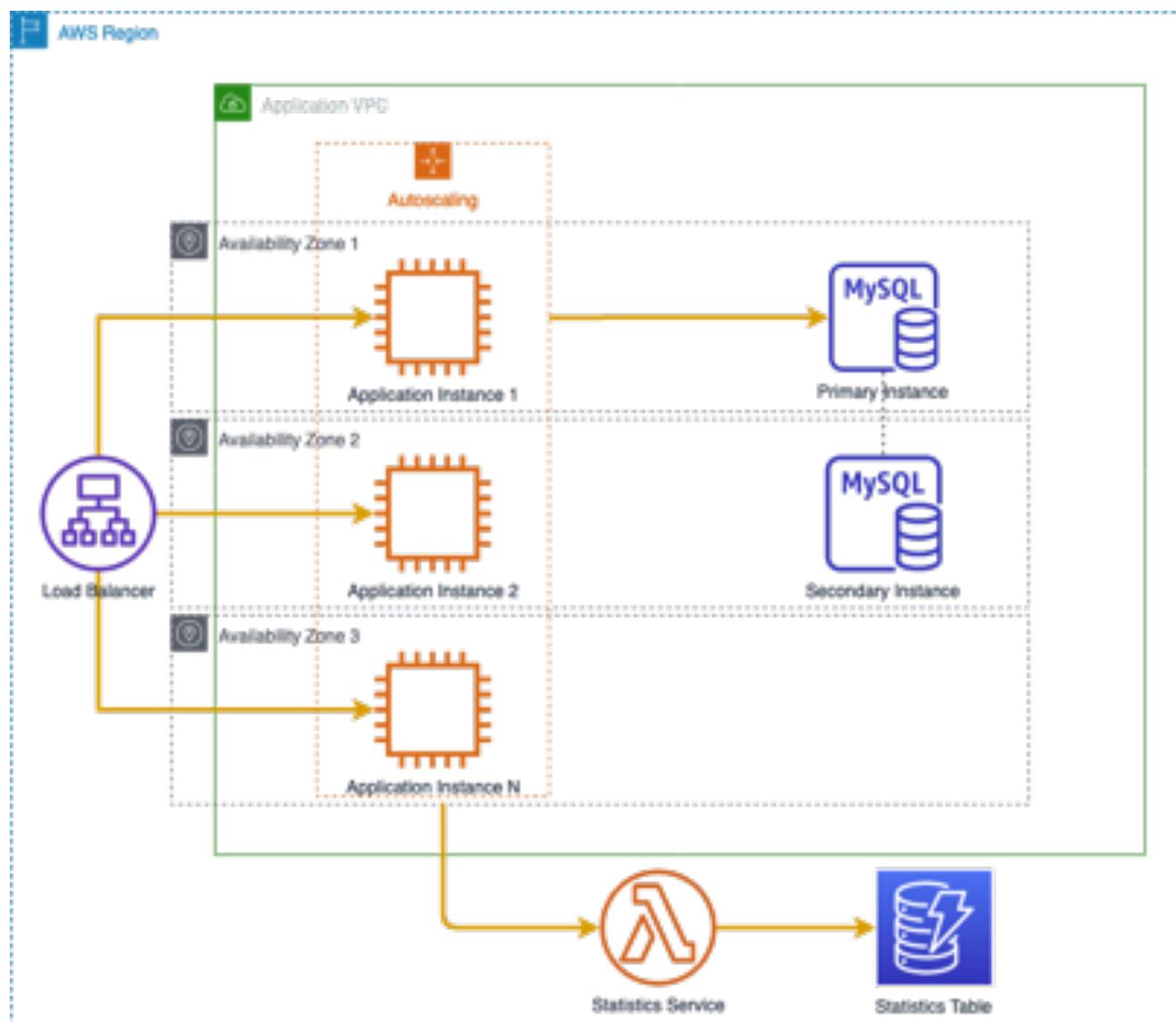
Filter resources by name, ID or value

| Instance ID | Name | Port | Zone | Health status | Health status details |
|--------------------|-------------------|------|------------|---------------|--------------------------------------|
| 1-007fb6ca55958673 | sample-app-server | 80 | eu-west-1a | healthy | |
| 1-00d525bfedab4e | sample-app-server | 80 | eu-west-1c | draining | Target deregistration is in progress |
| 1-06725992545f0aaf | sample-app-server | 80 | eu-west-1b | draining | Target deregistration is in progress |
| 1-0416549a0cf02a1 | sample-app-server | 80 | eu-west-1a | healthy | |
| 1-0a04330a97402e6a | sample-app-server | 80 | eu-west-1c | healthy | |

[Deregister](#) | [Register targets](#)

Wrapping Up

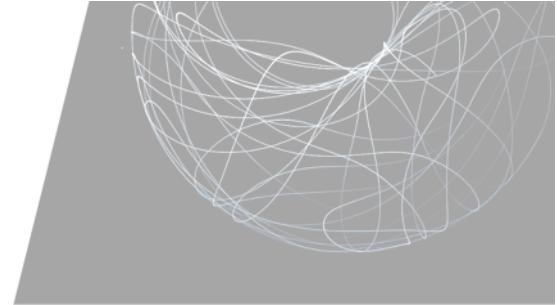




Learnings:

- 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