Improving reliability using health checks and dependency management

Andrew Robinson

29th April 2021

Who am I?

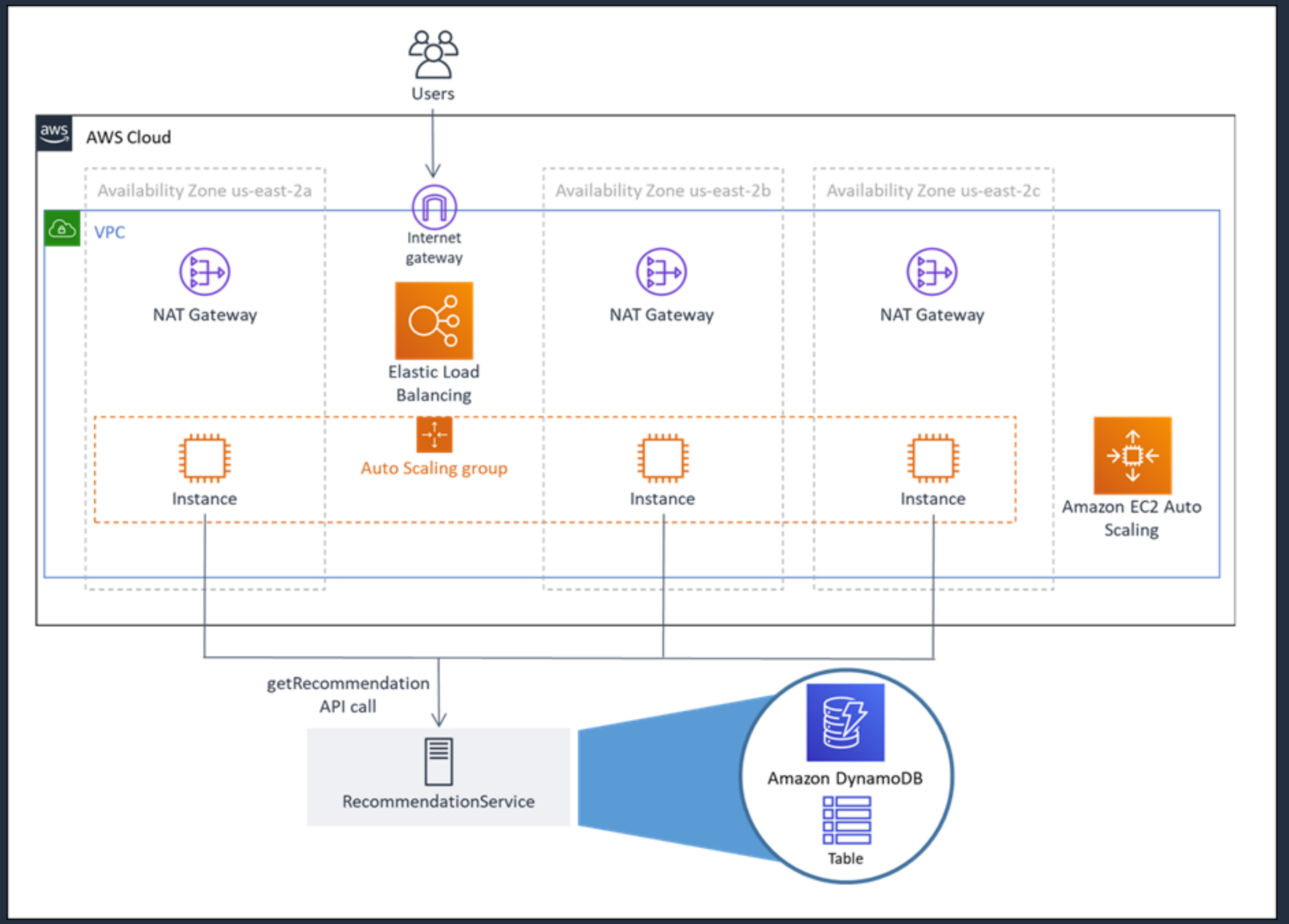
Andrew Robinson
Principal Solutions Architect AWS

AWS Well-Architected

Background in infrastructure and reliability for NetApp, Vision Express, Ametek, Inc.

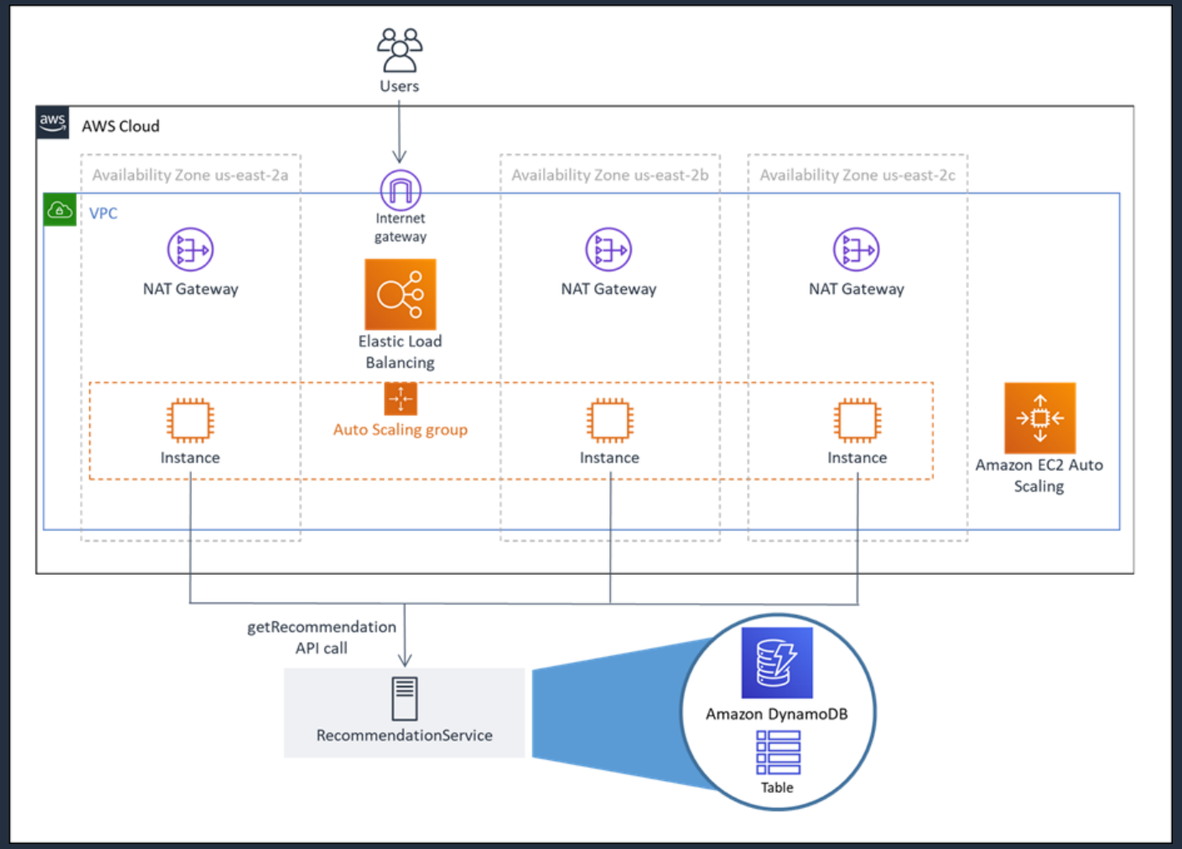
/in/andrew-m-robinson
@theonlyrobinson

aws



Infrastructure best practices

- Highly available network connectivity
- Deploy to multiple locations
- Loosely coupled dependencies



aws

Health checks

```
elif self.path == '/healthcheck’:

    self.send_response(200)
    self.send_header('Content-type', 'text/html’)
    self.end_headers()
```


aws

Deeper health checks

```
elif self.path == '/healthcheck’:

    is_healthy = False
    error_msg = ‘’
    TEST = 'test'

    try:
        user_id = str(0)
        response = call_getRecommendation(self.region, user_id)
        tv_show = response['Item']['Result']['S’]
        user_name = response['Item']['UserName']['S’]
        is_healthy = (tv_show == TEST) and (user_name == TEST)
    except Exception as e:
        error_msg += str(traceback.format_exception_only(e.__class__, e))
```


aws

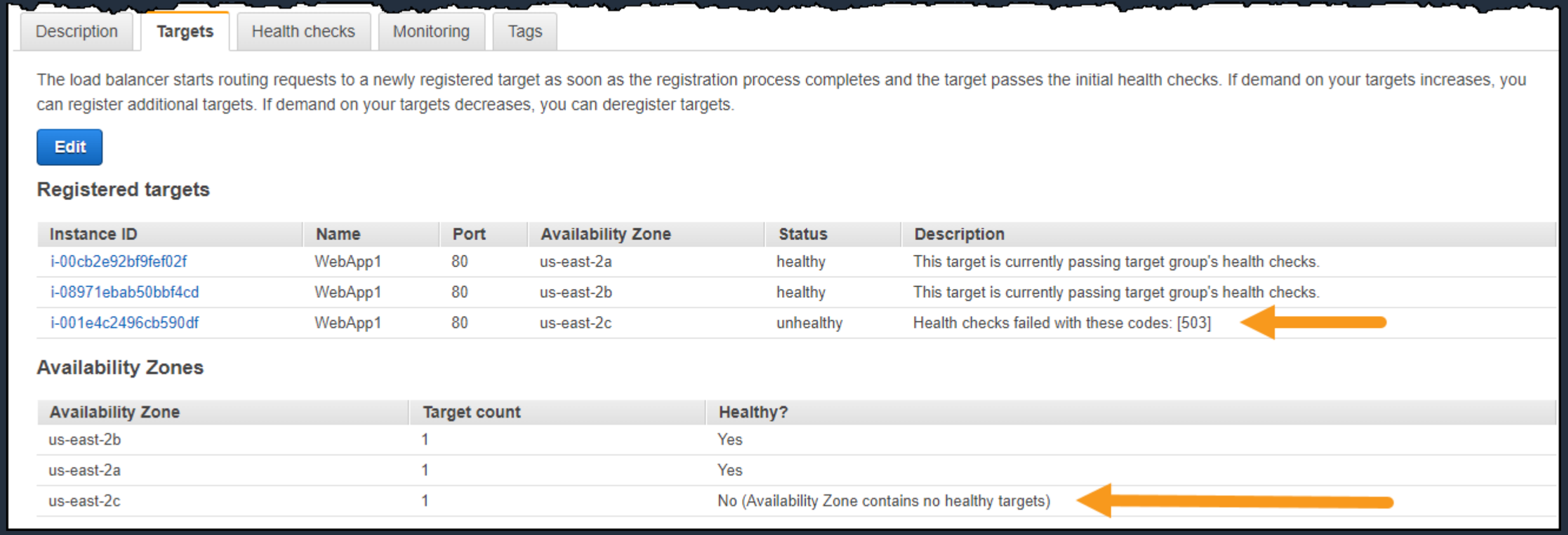
Deeper health checks

```
message = ""
if (is_healthy):
    self.send_response(200)
    self.send_header('Content-type', 'text/html’)
    self.end_headers()
    message += "<h1>Success</h1>"
     message += get_metadata()

 else:
    self.send_response(503)
    self.send_header('Content-type', 'text/html’)
    self.end_headers()
    message += "<h1>Fail</h1>"
    message += "<h3>Error message:</h3>"
    message += error_msg
```


aws

Fail closed vs fail open



Description | **Targets** | Health checks | Monitoring | Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

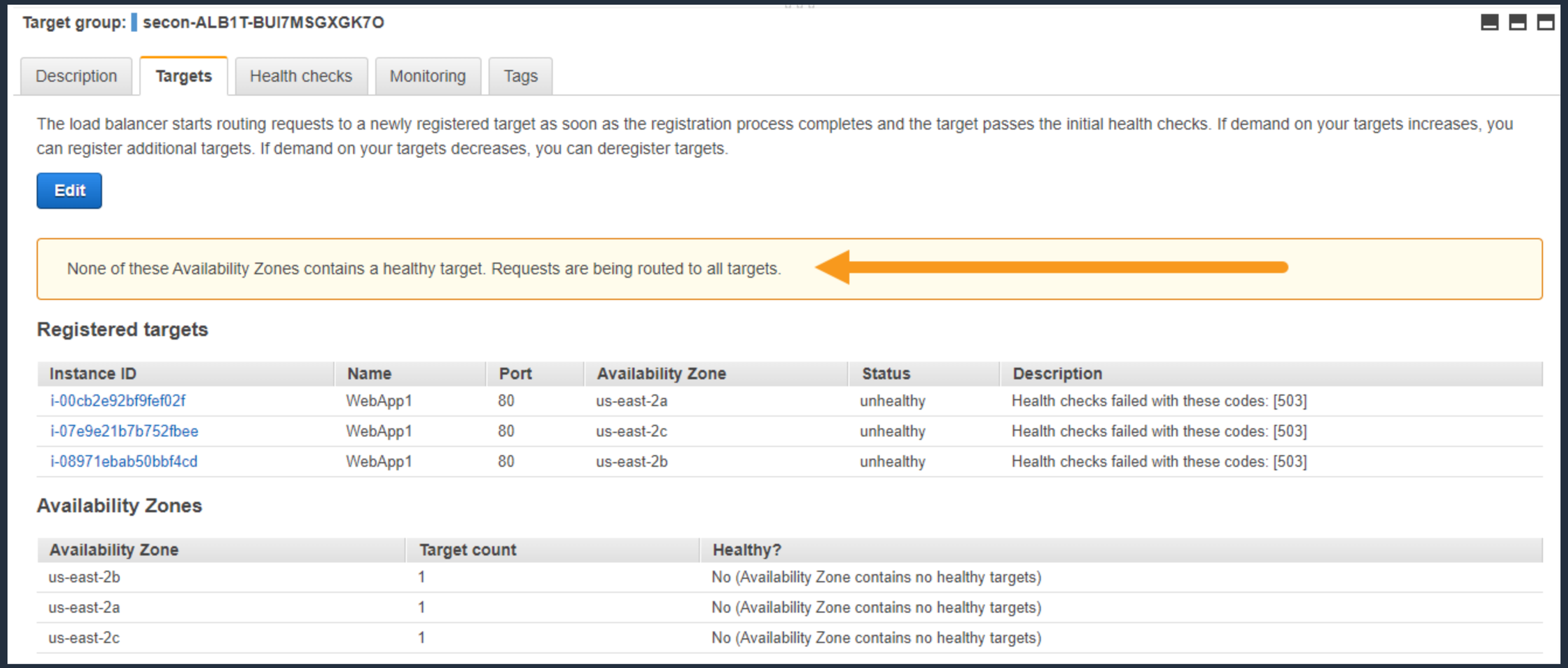
Instance ID	Name	Port	Availability Zone	Status	Description
i-00cb2e92bf9fef02f	WebApp1	80	us-east-2a	healthy	This target is currently passing target group's health checks.
i-08971ebab50bbf4cd	WebApp1	80	us-east-2b	healthy	This target is currently passing target group's health checks.
i-001e4c2496cb590df	WebApp1	80	us-east-2c	unhealthy	Health checks failed with these codes: [503]

Availability Zones

Availability Zone	Target count	Healthy?
us-east-2b	1	Yes
us-east-2a	1	Yes
us-east-2c	1	No (Availability Zone contains no healthy targets)

aws

Fail closed vs fail open



aws

Hard dependency

```
# Call the getRecommendation API on the RecommendationService
response = call_getRecommendation(self.region, user_id)

# Parses value of recommendation from DynamoDB JSON return value
tv_show = response['Item']['Result']['S']
user_name = response['Item']['UserName']['S']
message += recommendation_message (user_name, tv_show, True)
```


aws

Soft dependency

```
try:
.....

except Exception as e:
message += recommendation_message ('Valued Customer', 'I Love Lucy', False)
message += '<br><br><br><h2>Diagnostic Info:</h2>'
message += '<br>We are unable to provide personalized recommendations'
message += '<br>If this persists, please report the following info to us:'
message += str(traceback.format_exception_only(e.__class__, e))
```


aws

Best Practices

- Graceful degradation
- Transform hard dependencies into soft dependencies
- Monitor all components
- Failover to healthy resources
- Implement emergency levers
- Fail open vs fail closed

aws

Conclusion

- Servers and software fails for weird reasons
- Multiple layers of checks to catch failures
- Detect failure, take affected server out of service quickly
- Fail open gives benefits of deep health with rate limited safety

aws

Call to action

- Learn
- https://aws.amazon.com/builders-library/implementing-health-checks/

- Hands on
- https://wellarchitectedlabs.com/reliability/300_labs/300_health_checks_and_dependencies/

aws