Managing Failure Modes in Microservice Architectures

Adrian Cockcroft
@adrianco
AWS VP Cloud Architecture Strategy
Datacenter to cloud migrations are under-way for the most business and safety critical workloads

AWS and our partners are developing patterns, solutions and services for customers in all industries including travel, finance, healthcare, manufacturing...
Resilience

Past  Present  Future

Disaster recovery

Chaos engineering

Continuous Resilience
“If we change the name from chaos engineering to continuous resilience, will you let us do it all the time in production?”
You can only be as strong as your **weakest link**

Dedicated teams are needed to find weaknesses before they take you out!
Availability, Safety and Security have similar characteristics

Hard to measure near misses

Hard to model complex dependencies

Catastrophic failure modes
Availability, Safety and Security have similar mitigations

Layered defense in depth

Bulkheads to contain blast radius

Minimize dependencies/privilege
Availability, Safety and Security Break Each Other

Security breaks availability

Availability breaks safety

Etc.
What should your system do when something fails?

- Stop?
- Carry on with reduced functionality?
- Collapse horribly?
What should your system do when something fails? If a permissions look up fails, should you stop or continue?

Permissive failure, what’s the real cost of continuing?

See *Memories, Guesses, and Apologies* by Pat Helland
Do you have a backup datacenter?

How often do you failover apps to it?

How often do you failover the **whole datacenter** at once?

“Availability Theater”
A fairy tale...

Once upon a time, in theory, if everything works perfectly, we have a plan to survive the disasters we thought of in advance.

How did that work out?
Forgot to renew domain name...

Didn’t update security certificate and it expired...

Datacenter flooded in hurricane Sandy...

Whoops!

YOU, tomorrow
Drift into Failure

Sydney Dekker

Everyone can locally optimize for the right outcome at every step, and you may still get a catastrophic failure as a result...

We need to capture and learn from near misses, test and measure the safety margins, before things go wrong.
Chaos architecture

Four layers
Two teams
An attitude—
Find the weakest link
Defense In Depth

Experienced staff

Robust applications

Dependable switching fabric

Redundant service foundation
“You can’t legislate against failure, focus on fast detection and response.”

—Chris Pinkham
What’s wrong with minutes?

Takes too long to see a problem

- 1st high metric arrives at monitoring system
- 1st high metric processed (maybe)
- 3rd datapoints on user graph so looks bad at 8m00.

- Something broke at 2m20
- 40s of failure didn’t trigger
- 1st high metric seen at agent on instance
- 1st high metric seen on graph

Minute 1  Minute 2  Minute 3  Minute 4  Minute 5  Minute 6  Minute 7

Monitorama Keynote by @adriancockcroft May 2014
https://www.slideshare.net/adriancockcroft/monitorama-please-no-more
Try that again by the second

More confidence more quickly

Threshold

Something broke at 2m20

1st high metric arrives at monitoring system

1st high metric processed

Three datapoints on user graph so looks bad at 2m25.

Measurable in 1s

1st high metric seen at agent on instance

1st high metric seen on graph

Minute 1 Minute 2 Minute 3 Minute 4 Minute 5 Minute 6 Minute 7

Monitorama Keynote by @adrianco May 2014
https://www.slideshare.net/adriancockcroft/monitorama-please-no-more
Kalman, 1961 paper

*On the general theory of control systems*

A system is observable if the behavior of the entire system can be determined by only looking at its inputs and outputs.

Physical and software control systems are based on models, remember all models are wrong, but some models are useful...
Engineering a Safer World

Systems Thinking Applied to Safety

Nancy G. Leveson

STPA – Systems Theoretic Process Analysis

STAMP – Systems Theoretic Accident Model & Processes

http://psas.scripts.mit.edu for handbook and talks
Observability

STPA Model
(System Theoretic Process Analysis)
Observability

STPA Model
Understand Hazards that could disrupt successful application processing
STPA Hazards

Human Control Action:
- Not provided
- Unsafe action
- Safe but too early
- Safe but too late
- Wrong sequence
- Stopped too soon
- Applied too long

Conflicts
- Coordination problems
- Degradation over time
STPA Hazards

Sensor Metrics:
- Missing updates
- Zeroed
- Overflowed
- Corrupted
- Out of order
- Updates too rapid
- Updates infrequent
- Updates delayed
- Coordination problems
- Degradation over time

Throughput

Customer requests

Completed actions

Financial Services App
STPA Hazards

Model problems:
- Model mismatch
- Missing inputs
- Missing updates
- Updates too rapid
- Updates infrequent
- Updates delayed

Coordination problems

Degradation over time

Diagram:
- Control Plane
- Data Plane
- Human Controller
- Throughput
- Financial Services App
- Customer requests
- Completed actions
STPA Hazards
Model problems:
Model mismatch
Missing inputs
Coordination problems

Pilots were not trained on the new anti-stall algorithm
How do we usually calculate risk?

Severity * Probability = Risk

Assumes that we can determine severity and probability

Assumes we always detect the failure when it occurs

Basic model for financial and economic risk analysis
Failure Modes and Effects Analysis (FMEA)

Engineering oriented risk analysis

Severity * Probability * Detectability = Risk

Add observability to mitigate silent failures

Discuss and record component level failure modes

Prioritize mitigation work where it will do most good
FMEA for Web Services - Layered Responsibility

- Product Managers and Developers – unique business logic
- Software Platform Team – standard components and services
- Infrastructure Platform Team – resources, regions and networks
- Resilience Engineering – observability and incident management
- FMEA Spreadsheets: github.com/adrianco/slides
<table>
<thead>
<tr>
<th>Effect</th>
<th>SEVERITY of Effect</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hazardous without warning</td>
<td>Earthquake or meteorite destroys datacenter building, no warning, people injured</td>
<td>10</td>
</tr>
<tr>
<td>Hazardous with warning</td>
<td>Hurricane or tornado destroys datacenter building, several days warning, people injured</td>
<td>9</td>
</tr>
<tr>
<td>Very High</td>
<td>Datacenter flooded, compute and storage systems destroyed, building ok</td>
<td>8</td>
</tr>
<tr>
<td>High</td>
<td>Fire in datacenter, suppression system saves building, partial permanent compute and storage loss</td>
<td>7</td>
</tr>
<tr>
<td>Moderate</td>
<td>Hardware failure, CPU, disk, or power supply needs replacement. Often occurs after power or cooling failures.</td>
<td>6</td>
</tr>
<tr>
<td>Low</td>
<td>Power cut, cooling failure or network partition. Compute and storage returns when power, cooling and network are restored</td>
<td>5</td>
</tr>
<tr>
<td>Very Low</td>
<td>System operable with significant degradation of performance</td>
<td>4</td>
</tr>
<tr>
<td>Minor</td>
<td>System operable with some degradation of performance</td>
<td>3</td>
</tr>
<tr>
<td>Very Minor</td>
<td>System operable with minimal interference</td>
<td>2</td>
</tr>
<tr>
<td>None</td>
<td>No effect</td>
<td>1</td>
</tr>
</tbody>
</table>
## FMEA Probability Per Service Request

Guess to start with, then measure in production

<table>
<thead>
<tr>
<th>PROBABILITY of Failure</th>
<th>Failure Prob</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High: Failure is almost inevitable</td>
<td>&gt;1 in 2</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>1 in 3</td>
<td>9</td>
</tr>
<tr>
<td>High: Repeated failures</td>
<td>1 in 8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1 in 20</td>
<td>7</td>
</tr>
<tr>
<td>Moderate: Occasional failures</td>
<td>1 in 80</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1 in 400</td>
<td>5</td>
</tr>
<tr>
<td>Low: Relatively few failures</td>
<td>1 in 15,000</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1 in 150,000</td>
<td>2</td>
</tr>
<tr>
<td>Remote: Failure is unlikely</td>
<td>&lt;1 in 1,500,000</td>
<td>1</td>
</tr>
</tbody>
</table>
FMEA Detectability

Needs an observable monitoring alert to detect a failure

<table>
<thead>
<tr>
<th>Detection</th>
<th>Likelihood of DETECTION by Design Control</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Uncertainty</td>
<td>Design control cannot detect potential cause/mechanism and subsequent failure mode</td>
<td>10</td>
</tr>
<tr>
<td>Very Remote</td>
<td>Very remote chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>9</td>
</tr>
<tr>
<td>Remote</td>
<td>Remote chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>8</td>
</tr>
<tr>
<td>Very Low</td>
<td>Very low chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>7</td>
</tr>
<tr>
<td>Low</td>
<td>Low chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>6</td>
</tr>
<tr>
<td>Moderate</td>
<td>Moderate chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>5</td>
</tr>
<tr>
<td>Moderately High</td>
<td>Moderately High chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>4</td>
</tr>
<tr>
<td>High</td>
<td>High chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>3</td>
</tr>
<tr>
<td>Very High</td>
<td>Very high chance the design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>2</td>
</tr>
<tr>
<td>Almost Certain</td>
<td>Design control will detect potential cause/mechanism and subsequent failure mode</td>
<td>1</td>
</tr>
</tbody>
</table>
FMEA Example – Application Level

Customer is trying to obtain an IP address for a service

what could go wrong?

Lookup service?

DNS

No response...
FMEA Example – Application Level

Customer is trying to make a request to a service

what could go wrong?

Connect to host

No route
FMEA Example – Application Level

Customer is trying to make a request to a service

what could go wrong?

- Connect to host
- Undeliverable
FMEA Example – Application Level

Customer is trying to make a request to a service

what could go wrong?

Connect to host
Connect to host
Connected

Retry Timeout 100ms
### FMEA Example – Application Level

See full spreadsheets github.com/adrianco/slides for more failure modes

<table>
<thead>
<tr>
<th>Client Request to API Endpoint</th>
<th>Service unknown, address un-resolvable</th>
<th>Delay while discovery or DNS times out, slow fallback response</th>
<th>5</th>
<th>DNS configuration error, denial of service attack, or provider failure</th>
<th>1</th>
<th>Customer eventually complains via call center</th>
<th>10</th>
<th>50</th>
<th>Dual redundant DNS, fallback to local cache, hardcoded IP addresses. Endpoint monitoring and alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service unreachable, request undeliverable</td>
<td>Fast fail, no response</td>
<td>4</td>
<td>Network route down or no service instances running</td>
<td>1</td>
<td>Autoscaler maintains a number of healthy instances</td>
<td>1</td>
<td>4</td>
<td>Endpoint monitoring and alerts</td>
<td></td>
</tr>
<tr>
<td>Service reachable, request undeliverable</td>
<td>Connect timeout, slow fail, no response</td>
<td>4</td>
<td>Service frozen/not accepting connection</td>
<td>1</td>
<td>Retry request on different instance. Healthcheck failed instances removed. Log and alert.</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request delivered, no response - stall</td>
<td>Application request timeout, slow fail, no response</td>
<td>4</td>
<td>Broken service code, overloaded CPU or slow dependencies</td>
<td>1</td>
<td>Retry request on different instance. Healthcheck failed instances removed. Log and alert.</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FMEA Example – Application Level

Customer is trying to make a request to a service

what could go wrong?

Hi, I’m user123

Auth failure

Log: 25ms user123 Auth failure
## FMEA Example – Application Level

### Authentication Failures

<table>
<thead>
<tr>
<th>Item / Function</th>
<th>Potential Failure Mode(s)</th>
<th>Potential Effect(s) of Failure</th>
<th>Sever</th>
<th>Potential Cause(s)/Mechanism(s) of Failure</th>
<th>Prob</th>
<th>Current Design Controls</th>
<th>Det</th>
<th>RPN</th>
<th>Recommended Action(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Client can’t authenticate</td>
<td>Can’t connect application</td>
<td>5</td>
<td>Certificate timeout, version mismatch, account not setup, credential changed</td>
<td>3</td>
<td>Log and alert on authentication failures</td>
<td>3</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Slow or unreliable authentication</td>
<td>Slow start for application</td>
<td>4</td>
<td></td>
<td>Auth service overloaded, high error and retry rate</td>
<td>3</td>
<td>Log and alert on high authentication latency and errors</td>
<td>4</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>
FMEA Example – Application Level

Customer is trying to make a request to a service

what else could go wrong?

GET /index.html

Log: 25ms user123 GET /index.html
## FMEA Example – Application Level

### Application Failures

<table>
<thead>
<tr>
<th>Time Bombs</th>
<th>Internal application counter wraparound</th>
<th>Test long running operations of code base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory leak</td>
<td></td>
<td>Monitor process sizes and garbage collection intervals over time</td>
</tr>
<tr>
<td>Date Bombs</td>
<td>Leap year, leap second, epoch wrap around, &quot;Y2K&quot;</td>
<td>Test across date boundaries</td>
</tr>
<tr>
<td>Content Bombs</td>
<td>Incoming data that crashes the app</td>
<td>Fuzz the input with generated random and structured data to show it doesn't crash.</td>
</tr>
<tr>
<td>Configuration Errors</td>
<td>Configuration file syntax errors or incorrect values</td>
<td>Canary test deployments incrementally. Chaos testing.</td>
</tr>
<tr>
<td>Versioning Errors</td>
<td>Incompatible interface versions</td>
<td>Canary test deployments incrementally</td>
</tr>
<tr>
<td>Retry Storms</td>
<td>Too many retries, too large timeout values</td>
<td>Chaos testing applications under stress</td>
</tr>
<tr>
<td>Excessive Logging</td>
<td>Cascading overload</td>
<td>Chaos testing applications under stress</td>
</tr>
</tbody>
</table>
### FMEA Example – Software Stack Level

#### Service Control Plane Failures – part of a long list...

<table>
<thead>
<tr>
<th>Service Control Plane</th>
<th>Failure Cause</th>
<th>Mitigation Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC2 Control Plane</td>
<td>Instance request refused, direct or via autoscaler</td>
<td>Service call for increased limit. Try a different instance type, different zone, or different region</td>
</tr>
<tr>
<td></td>
<td>Capacity limited or control plane failure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Limit reached, or Insufficient Capacity Exception</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instance created but fails to start</td>
<td>0 Retry via autoscaler</td>
</tr>
<tr>
<td></td>
<td>Bad instance hardware</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instance slow to start</td>
<td>0</td>
</tr>
<tr>
<td>EC2 Network Control Plane</td>
<td>Configuration request refused</td>
<td>Service call for increased limit. Try a different zone, or different region</td>
</tr>
<tr>
<td></td>
<td>Capacity limited or control plane failure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Limit reached, or Insufficient Capacity Exception</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Network creation started but operation fails</td>
<td>0 Pre-allocate all network structures in all regions</td>
</tr>
<tr>
<td>Database Service (Aurora Postgres)</td>
<td>Control plane configuration request refused</td>
<td>Service call for increased limit. Try a different zone, or different region</td>
</tr>
<tr>
<td></td>
<td>Database table creation started but operation fails</td>
<td>0 Pre-allocate all database tables in all regions</td>
</tr>
<tr>
<td></td>
<td>Master Instance failure</td>
<td></td>
</tr>
</tbody>
</table>
### FMEA Example – Infrastructure Level

#### Availability Zone Durability

<table>
<thead>
<tr>
<th>Item / Function</th>
<th>Potential Failure Mode(s)</th>
<th>Potential Effect(s) of Failure</th>
<th>Sev</th>
<th>Potential Cause(s)/ Mechanism(s) of Failure</th>
<th>Prob</th>
<th>Current Design Controls</th>
<th>Det</th>
<th>RPN</th>
<th>Recommended Action(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability Zone Durability</td>
<td>Permanent destruction of zone</td>
<td>Total data loss in zone</td>
<td>8</td>
<td>Fire or flood inside building or destruction of datacenter building</td>
<td>2</td>
<td>Cross zone synchronous replication to over 10Km away</td>
<td>1</td>
<td>16</td>
<td>Ensure that system can run on two out of three zones</td>
</tr>
<tr>
<td>Temporary loss of zone</td>
<td></td>
<td>Loss of compute capacity and nondurable state in zone</td>
<td>5</td>
<td>Power or cooling outage causes reboot of part or all of a datacenter building</td>
<td>3</td>
<td>Cross zone synchronous replication to over 10Km away</td>
<td>1</td>
<td>15</td>
<td>Ensure that system can run on two out of three zones</td>
</tr>
</tbody>
</table>
### FMEA Example – Infrastructure Level

#### Region Connectivity

<table>
<thead>
<tr>
<th>Region Connectivity</th>
<th>Address unresolved</th>
<th>Delay while DNS times out, slow fallback response</th>
<th>DNS configuration error, denial of service attack, or provider failure</th>
<th></th>
<th></th>
<th>Dual redundant DNS, failback to local cache, hardcoded IP addresses, Endpoint monitoring and alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reachable, request undeliverable</td>
<td>Fast fail, no response</td>
<td></td>
<td>Network route down</td>
<td>1</td>
<td></td>
<td>Failover to secondary region</td>
</tr>
<tr>
<td>Request undeliverable</td>
<td>Connect timeout, slow fail, no response</td>
<td></td>
<td>Router frozen/not accepting connection</td>
<td>1</td>
<td></td>
<td>Failover to secondary region</td>
</tr>
<tr>
<td>Request delivered, no response - stall</td>
<td>Application request timeout, slow fail, no response</td>
<td></td>
<td>Broken router, overloaded network or slow dependencies</td>
<td>1</td>
<td></td>
<td>Failover to secondary region</td>
</tr>
<tr>
<td>Response undeliverable</td>
<td>Application request timeout, slow fail, no response</td>
<td></td>
<td>Network return route failure, dropped packets</td>
<td>1</td>
<td></td>
<td>Failover to secondary region</td>
</tr>
<tr>
<td>Response received in time but empty or unintelligible</td>
<td>Fast fail, no response</td>
<td></td>
<td>Network response failure</td>
<td>2</td>
<td></td>
<td>Failover to secondary region</td>
</tr>
<tr>
<td>Request delivered, response delayed beyond spec</td>
<td>Degraded response arrives too late, slow fallback response</td>
<td></td>
<td>Network overloaded dependent services responding slowly</td>
<td>2</td>
<td></td>
<td>Failover to secondary region</td>
</tr>
<tr>
<td>Request delivered, degraded response delivered in time</td>
<td>Degraded timely response</td>
<td></td>
<td>Service overloaded, dependent services responding slowly</td>
<td>2</td>
<td></td>
<td>Alert operators</td>
</tr>
</tbody>
</table>
### FMEA Example – Operations Level

#### Monitoring System Connectivity

<table>
<thead>
<tr>
<th>Item / Function</th>
<th>Potential Failure Mode(s)</th>
<th>Potential Effect(s)</th>
<th>Severity</th>
<th>Potential Cause(s)/Mechanism(s) of Failure</th>
<th>Probability</th>
<th>Current Design Controls</th>
<th>Detection</th>
<th>RPN</th>
<th>Recommended Action(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Monitoring agent can't authenticate</td>
<td>Can't monitor application</td>
<td>5</td>
<td>Certificate timeout, version mismatch, account not setup, credential changed</td>
<td>3</td>
<td>Log and alert on authentication failures</td>
<td>3</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monitoring tool end user operator can't authenticate</td>
<td>Can't monitor system, increased MTTR</td>
<td>5</td>
<td>Certificate timeout, version mismatch, account not setup, credential changed</td>
<td>3</td>
<td>Log and alert on authentication failures</td>
<td>3</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Slow or unreliable authentication</td>
<td>Errors and delays in observability and alerts</td>
<td></td>
<td>4</td>
<td>Auth service overloaded, high error and retry rate</td>
<td>3</td>
<td>Log and alert on high authentication latency and errors</td>
<td>4</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>
STPA – Top down focus on control hazards

FMEA – Bottom up focus on prioritizing failure modes

STPA tends to have better failure coverage than FMEA, especially for human controller/user experience issues

Both are useful...
Cloud provides the automation that leads to chaos engineering
Good Cloud Resilience Practices

Rule of 3 – three ways for critical operations to succeed

- Synchronous data replication over three zones in a region
- DR failover from primary region to either of two secondary regions
- Active-Active-Active workloads across three regions
Good Cloud Resilience Practices

Fail up - DR failover between regions

- From smaller capacity region to larger capacity region
- From distant region to closer (lower latency) region
Good Cloud Resilience Practices

Chaos first

Build your resilience environment *before* introducing apps to it

Automated continuous zone and region failover testing

Make it a “badge of honor” to have an app pass the chaos test
Good Cloud Resilience Practices

Continuous Resilience

Continuous Delivery needs Test Driven Development and Canaries

Continuous Resilience needs automation in both test and production

Make failure mitigation into a well tested code path and process

Call it Chaos Engineering if you like, it’s the same thing...
As datacenters migrate to cloud, fragile and manual disaster recovery processes can be standardized and automated.
Testing failure mitigation will move from a scary annual experience to automated continuous resilience
Continuous Resilience

Paper: Building Mission Critical Financial Services Applications on AWS
By Pawan Agnihotri with contributions by Adrian Cockcroft

Blog post: Failure Modes and Continuous Resilience - medium.com/@adrianco

Adrian Cockcroft
@adrianco
AWS VP Cloud Architecture Strategy