Systematically Surviving Failures
In the Cloud

Manish Maheshwari, Principal Software Engineer
Expedia Group
Why?

Production Stability .... Cost of Incidents

<table>
<thead>
<tr>
<th>Uptime</th>
<th>Downtime</th>
<th>Revenue Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>99%</td>
<td>3.6 days</td>
<td>$100 million</td>
</tr>
<tr>
<td>99.9%</td>
<td>9 hours</td>
<td>$10 million</td>
</tr>
<tr>
<td>99.99%</td>
<td>53 minutes</td>
<td>$1 million</td>
</tr>
</tbody>
</table>
Why?

Poor User Experience ... Loss of Brand Value
Why?

Sleepless Nights... Spoiled Mornings
Why Now?

... circuit-breaking...
back-pressure....deadlines....retries....backoffs....
timeouts....bulk-heading....tracing...never-ending...
Why Now?

We have embraced velocity
Goals

“Build the fastest, most resilient global travel platform fully in the cloud. Modernize our automation to become cloud native, and switch from DR to active resiliency.”

– ad verbatim from our Cloud VP

Improve

– Resilience-at-large
– Resilience-at-small
How?
How?
How?
How?

- Incidents
  - CoE
    - Patterns
      - Resilience Maturity
        - What-Ifs
          - WIF-1
          - WIF-2
          - WIF-3
          - WIF-4
          - WIF-5
          - WIF-6
How?
THE SIX WHAT-IFS
What-if the VM is Unhealthy?

**isHealthy**

- Metrics
- Response Time
- Health Check Intervals
- Healthy Thresholds
What-if the VM is Unhealthy?

Anomaly detection for VM

Platform library for application health
What-if the Traffic Surges?

- Evaluate traffic pattern
  - Predictable Peaks
  - Unpredictable Peaks
- Figure out Min, Max, Desired
What-if the Traffic Surges?

Considerations
- Scale out & Scale in Policy
- Time to Launch and Run
- Right Metrics to Trigger Scaling
- Scaling Adjustment Types
- AZs to span
What-if the Traffic Surges?
What-if the Traffic Surges?
What-if the Traffic Surges?

Some Learnings -
• CPU Based Scaling works most of the time
• Operating Band Width – 10% to 15%
• Connection Draining
• Reduced bootstrap time goes long way
• Traffic jumps from 1x to 4x in a very short span, and for very short duration are signs of bots
What-if the Service in a Region is Unhealthy?
What-if the Service in a Region is Unhealthy?
What-if the Service in a Region is Unhealthy?
What-if the Service in a Region is Unhealthy?

Fault Domain and the Vegas Rule
What-if the Service in a Region is Unhealthy?

Considerations:

• Fault Domain and the Vegas Rule
• Minimum Failover Time
• Rule Type - Weighted RR, Primary/Failover
• State of CW Alarm
• Health Checks
• Failover Times
• Traffic Rules
What-if the Dependency goes Down?
What-if the Dependency goes Down?
What-if the Dependency goes Down?
What-if the Dependency goes Down?
What-if the Dependency goes Down?
What-if the Dependency goes Down?
What-if the Dependency goes Down?
What-if the Dependency goes Down?
What-if the Dependency goes Down?

Analyzing Dependencies
- Downstream
- Cross-cutting
What-if the Dependency goes Down?

Failover to Alternate

Use Offline Data

Degrade Gracefully

Think Creatively
What-if the Dependency goes Down?

### Circuit

<table>
<thead>
<tr>
<th>Sort: Error then Volume</th>
<th>Alphabetical</th>
<th>Volume</th>
<th>Error</th>
<th>Mean</th>
<th>Median</th>
<th>90</th>
<th>99</th>
<th>99.5</th>
<th>Success</th>
<th>Short-Circuited</th>
<th>Bad Request</th>
<th>Timeout</th>
<th>Rejected</th>
<th>Failure</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>6,261</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1 %</td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 2.5/s</td>
<td>Circuit Closed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 4.6/s</td>
<td>Circuit Closed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 1.8/s</td>
<td>Circuit Closed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 0.1/s</td>
<td>Circuit Closed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Hosts:**
  - Median: 220 90th: 249ms Mean: 103ms
  - Median: 220 90th: 268ms Mean: 103ms
  - Median: 220 90th: 320ms Mean: 103ms
  - Median: 220 90th: 94ms Mean: 103ms

- **Thread Pools**

<table>
<thead>
<tr>
<th>Sort: Alphabetical</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 1.5/s</td>
</tr>
<tr>
<td>Active: 133</td>
<td>Max Active: 674</td>
</tr>
<tr>
<td>Queued: 1</td>
<td>Executions: 3,940</td>
</tr>
<tr>
<td>Pool Size: 4425</td>
<td>Queue Size: 10</td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 1.5/s</td>
</tr>
<tr>
<td>Active: 37</td>
<td>Max Active: 443</td>
</tr>
<tr>
<td>Queued: 0</td>
<td>Executions: 3,401</td>
</tr>
<tr>
<td>Pool Size: 2200</td>
<td>Queue Size: 10</td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 0.0/s</td>
</tr>
<tr>
<td>Active: 0</td>
<td>Max Active: 0</td>
</tr>
<tr>
<td>Queued: 0</td>
<td>Executions: 0</td>
</tr>
<tr>
<td>Pool Size: 2190</td>
<td>Queue Size: 10</td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 1.5/s</td>
</tr>
<tr>
<td>Active: 18</td>
<td>Max Active: 18</td>
</tr>
<tr>
<td>Queued: 0</td>
<td>Executions: 0</td>
</tr>
<tr>
<td>Pool Size: 394</td>
<td>Queue Size: 10</td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 0.0/s</td>
</tr>
<tr>
<td>Active: 0</td>
<td>Max Active: 0</td>
</tr>
<tr>
<td>Queued: 0</td>
<td>Executions: 0</td>
</tr>
<tr>
<td>Pool Size: 56547</td>
<td>Queue Size: 11</td>
</tr>
<tr>
<td>Host: 0.0/s</td>
<td>Cluster: 2.5/s</td>
</tr>
<tr>
<td>Active: 32</td>
<td>Max Active: 1,283</td>
</tr>
<tr>
<td>Queued: 1</td>
<td>Executions: 5,531</td>
</tr>
<tr>
<td>Pool Size: 15317</td>
<td>Queue Size: 10</td>
</tr>
</tbody>
</table>

---

**HYSTRIX**

Defend Your App

www.developersummit.com
You need to restart your computer. Hold down the Power button until it turns off, then press the Power button again.

Redémarrez l’ordinateur. Enfoncez le bouton de démarrage jusqu'à l’extinction, puis appuyez dessus une nouvelle fois.

Debe reiniciar el ordenador. Mantenga pulsado el botón de arranque hasta que se apague y luego vuelva a pulsarlo.

Sie müssen den Computer neu starten. Halten Sie den Ein-/Ausschalter gedrückt bis das Gerät ausgeschaltet ist und drücken Sie ihn dann erneut.

コンピュータの再起動が必要です。電源が切れるまでパワーボタンを押し続けてから、もう一度パワーボタンを押します。
What-if the Dependency Behaves Bad?

“Be Conservative in what we send to a service and Liberal in what we accept”

• Bad Behaviors
  – packet-drops...connection-resets...slow-readers...
  – time-outs...increased-latency...

• Considerations
  – Software Resiliency Patterns - ... circuit-breaking...
    retries....backoffs....
  – Test the Resiliency – Simian Army, Gremlin, et al
What-if the Dependency Behaves Bad?
What-if the Dependency Behaves Bad?

**Feature:** Resilience Verification of Search Service

**Background:** Ensure that the service is up and running
- **Given** There are no proxies
  - And Proxy named `pricingServiceProxy` is created from `<localhost:21001>` to `<fareservice.internal.expedia.com:443>`
  - And Proxy named `collectorProxy` is created from `<localhost:31001>` to `<collector.internal.expedia.com:443>`
  - And Proxy named `databaseProxy` is created from `<localhost:31002>` to `<offermaster.expedia.com:3306>`
  - And Proxy named `fareSearchProxy` is created from `<localhost:31003>` to `<fareservice.internal.expedia.com:443>`

**Scenario:** Pricing service downstream latency of 2+3 seconds
- **Given** The `pricingServiceProxy` has downstream latency of 6000 ms with jitter of 1000 ms
  - And The `pricingServiceProxy` has upstream latency of 5000 ms with jitter of 1000 ms
- **When** The `/restservice` endpoint is hit with POST request from resource `xmls/carssearch1.xml` and below headers
  - *accept* application/xml
  - *content-type* application/xml
  - *cache-control* no-cache
- **Then** Response code is 200
  - And Response received has `<ns4:CarProduct>` text
  - And Response received has `vendor` text
  - And Delay in response is more than 5000 ms
  - And Delay in response is less than 90000 ms

**Scenario:** Pricing service downstream slice
- **Given** The `pricingServiceProxy` has downstream slicing of 3000 byte packets delayed by 10000 microseconds
- **Given** The `pricingServiceProxy` has upstream slicing of 3000 byte packets delayed by 10000 microseconds
- **When** The `/restservice` endpoint is hit with POST request from resource `xmls/carssearch1.xml` and below headers
  - *accept* application/xml
  - *content-type* application/xml
  - *cache-control* no-cache
- **Then** Response code is 200
  - And Response received has `<ns4:CarProduct>` text
  - And Responses received have `vendor` text
What-if the Dependency Behaves Bad?

Chaos Engineering
• VM Termination
• Health Check integration
• New attacks
What-if the Service gives up too early?

Time Grading
- SLA Definition
- Right Alarm Thresholds
- Deadlines
What-if the Service gives up too early?

Classifying HTTP Responses

- Non-Retryable Failures
- Retryable Failures
TACTICAL LEARNINGS
What Helps

1. Translating the Black Book into Small, Actionable Items
2. Simplest things give the Largest Gains *(just take the first step)*
3. State the Problem (what-ifs) *(but do not prescribe a solution)*
4. Unblock by filling in Tech Gaps
5. A good Program Manager
Tracking
Acknowledgements

Abhayjit Kharbanda *(innovative deviation-based VM terminations)*
Abhinav Garg *(building and on-boarding numerous services with right step scaling)*
Ankit Goyal *(creating image for Hystrix Turbine Dashboards for EC2/DC)*
Geetika Arora *(several fixes in the CICD)*
Nitin Narang *(program managing 50+ services on What-Ifs)*
Nitish Sabharwal *(building scaling using non-traditional metrics)*
Sunil Singhal *(platform library for health checks)*
Willie Wheeler *(building confidence and running Chaos Monkey on Production)*
Conclusion

1. What-if the VM is Unhealthy?
2. What-if the Traffic Surges?
3. What-if the Region is Unhealthy?
4. What-if the Dependency is Down?
5. What-if the Dependency Behaves Bad?
6. What-if the Service gives up too early?
Thank you
GREAT INDIAN DEVELOPER SUMMIT 2019

Conference: April 23-26, Bangalore

Register early and get the best discounts!

www.developersummit.com  @greatindiandev  bit.ly/gidslinkedin