Winston: Helping Netflix Engineers Sleep at Night

Our journey… assisting engineers reduce operational load and MTTR
On-Call!
Sayli Karmarkar
Senior Software Engineer
Diagnostics and Remediation Engineering (DaRE)

skarmarkar@netflix.com
@HikerTechy
https://www.linkedin.com/in/saylikarmarkar

DaRE Team’s Focus
Build platforms, tools and libraries
to help teams reduce MTTR for operational issues.
Traditional On-Call Timeline

2:00 AM: PagerDuty Alert
2:02 AM: Engineer Wakes up
2:07 AM: Logs in and ACK
2:10 AM: Studies the alert
2:15 AM: Checks runbook
2:20 AM: Runs diagnostics
2:30 AM: Fixes/Mitigates the problem
Works, but does it scale?!
Netflix goes down. Twitter blows up

SUNNI
@SunniAndTheCity

Netflix is down on a rainy Saturday afternoon.
12:42 PM - 1 Oct 2016

105 replies 105 likes
Traditional On-Call Pain Points

MTTR

Productivity
Solution?

Automate

- Removing False Positives
- Collecting Diagnostic Information
- Mitigating the problem to reduce impact on the customers

Hands-free

Feed the runbooks to an event-driven automation platform and have them executed in response to operational events
Unique Problem? Not really ..
Define

- Business Goals
- Use-cases
- Customers
- Constraints
- Interactions with other services
Winston’s Goals

- Assist engineers in reducing MTTR and pager fatigue by providing a platform to automate their runbooks
- Provide an easy way to connect automated runbooks to an event
- Let engineers focus on the business logic of runbooks rather than infrastructure aka PaaS.
- Provide appropriate wrappers and libraries to interact with other services
- Ensure best practices for automations and runbook lifecycle management
What is Winston?

Winston is an event driven runbook automation platform. It is designed to host and execute runbooks in response to operational events.
Traditional On-Call Timeline

2:00 AM
- PagerDuty Alert

2:02 AM
- Engineer Wakes up

2:07 AM
- Logs in and ACK

2:10 AM
- Studies the alert

2:15 AM
- Checks runbook

2:20 AM
- Runs diagnostics

2:30 AM
- Fixes the problem
On-Call With Winston

2:00 AM

False Positive

Mitigates the problem

Assisted Diagnostics

2:05 AM

Winston

2:05 AM

2:15 AM
Evaluation - Build / Reuse / Buy
Stackstorm

+  
  - A generic pluggable Event-Driven Automation Platform  
  - Designed with availability and reliability in mind  
  - Open source + Code following good design practices  
  - Good alignment with respect to goals and future direction

-  
  - High availability and reliability not exercised a lot  
  - Dependency on MongoDB and RabbitMQ  
  - No easy way of adding and updating automation
Good Starting Point..

As a Service (High Availability and Reliability)

Iterate and Evaluate Regularly

Inbound Integrations (through SQS)

Outbound Integrations

StackStorm +

Spinnaker

Bolt ...
A closer look at a Winston Instance
Challenges

- Added cognitive load resulting in less adoption
- How to help engineers choose operational efficiency over new features?
- Recommended and safe automation and lifecycle management practices are often not followed
- Simple use-cases are not trivial to on-board
Winston Studio

- One stop portal for all things Winston
- Supports Create, Read, Update, Delete, Execute and Diagnose functionality
- Implements best practices
  - Compliance/Auditing
  - Persistence
  - Security (Authentication/Authorization)
- Self serve & scalable
Select a Pack to get started

A Pack is a group of related Actions, typically organized around the needs of a specific service, product, or team.

<table>
<thead>
<tr>
<th>Pack Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>netflix_cass</td>
<td>Netflix Cassandra pack</td>
</tr>
<tr>
<td>netflix_cloudsec</td>
<td>Netflix cloudsec Pack</td>
</tr>
<tr>
<td>netflix_core</td>
<td>Netflix Core Pack - Contains re-usable building blocks to access Netflix ecosystem</td>
</tr>
<tr>
<td>netflix_coreeng</td>
<td>Netflix coreeng Pack</td>
</tr>
<tr>
<td>netflix_coresre</td>
<td>Netflix coresre Pack</td>
</tr>
<tr>
<td>netflix_datapipeline</td>
<td>Winston Pack for Platform Data Pipeline</td>
</tr>
</tbody>
</table>
Runbook View
### Executions

**Search Criteria**

- **Criteria**: `broker_offline_process_alert`

#### Details

<table>
<thead>
<tr>
<th>Status</th>
<th>Started at</th>
<th>Duration (sec)</th>
<th>alert_env</th>
<th>alert_matchset</th>
<th>alert_name</th>
<th>alert_region</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Jul 18, 2018 12:03 AM</td>
<td>12</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 18, 2018 12:06 AM</td>
<td>22</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 17, 2018 11:47 AM</td>
<td>19</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 17, 2018 11:44 AM</td>
<td>22</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 17, 2018 11:42 AM</td>
<td>22</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 17, 2018 11:04 AM</td>
<td>23</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 16, 2018 02:23 AM</td>
<td>23</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 16, 2018 02:03 AM</td>
<td>22</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 16, 2018 02:01 AM</td>
<td>22</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 15, 2018 12:00 AM</td>
<td>14</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 12, 2018 09:30 AM</td>
<td>23</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 10:16 AM</td>
<td>20</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 09:15 AM</td>
<td>19</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 08:14 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 07:15 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 04:14 AM</td>
<td>19</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 05:14 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 10, 2018 04:14 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 03:14 AM</td>
<td>19</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 02:11 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 01:13 AM</td>
<td>19</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 11, 2018 12:12 AM</td>
<td>19</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 10, 2018 11:11 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 10, 2018 10:11 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>Jul 10, 2018 09:11 AM</td>
<td>18</td>
<td>prod</td>
<td></td>
<td>BrokerOffline</td>
<td>vs-east-1</td>
<td>Details</td>
</tr>
</tbody>
</table>
Execution Details

**Execution Region**
us-east-1

**Parameters**

- **alert_env**
  - prod

- **alert_matchset**
  - [set]

- **alert_metadata**
  - 

- **alert_name**
  - BrokerOffline

- **alert_region**
  - us-east-1

- **team_email**
  - [email]

- **timeout**
  - 7200

**Standard Output (stdout)**

```plaintext
1 InstanceIds: [{'id': 'i-01234567890abcde'}]
2 active_instance states for app kafka: []
3 active_instance states for app kafkacluster: []
4 active_instance states for app x: []
5 active_instances: ['kafkacluster': [], 'kafka': [], 'x': []]
```

**Log Output (includes stdout)**

```plaintext
2016-07-18 06:13:07,482 INFO Loading/Refreshing App kafka
2016-07-18 06:13:07,483 DEBUG App kafka loaded/refreshed
2016-07-18 06:13:07,483 DEBUG App kafkacluster loaded/refreshed
2016-07-18 06:13:07,483 DEBUG App x loaded/refreshed
2016-07-18 06:13:07,483 DEBUG Loading/Refreshing App kafka
2016-07-18 06:13:07,483 DEBUG App kafka loaded/refreshed
2016-07-18 06:13:07,483 DEBUG App kafkacluster loaded/refreshed
2016-07-18 06:13:07,483 DEBUG App x loaded/refreshed
2016-07-18 06:13:07,483 DEBUG Log sent to [email] with subject=[prod] netsite
```
Current Winston Deployment
Sample Use-cases

False Positives
- Broker reporting offline when AWS maintenance takes down an instance
- Cassandra ring health

Diagnostics - Correlation could point towards the root cause
- Checking current maintenance jobs running on a cluster when an issue occurs
- Querying dependencies upstream and downstream for anomalous behavior
- Capture current system state and logs to analyze failures and reach the root cause quicker

Mitigation
- Restart kafka process
- Clean up disk space
Alert: test us-east-1 BrokerOffline

winston via PAE winston alerts <pae-winston-alerts> 3:28 PM (0 minutes ago)  

to data-pipeline-

This is an alert generated by Winston - Automated Troubleshooting and Remediation Platform. 
Winston execution ID: 56314b959287d930566e34a2

PROBLEM: Following kafka broker instances were reported to be offline:

Instances terminated by AWS --
  i-cd939c7c: kafkabroker-logtrace-us-east-1d

Alert Snapshot: http://alert-history.us-east-1.test.netflix.net/history/snapshot/kafka/BrokerOffline/us-east-1/1446071188127?checkTime=1446071188127&sourceInstance=i-c5e8ce65

If you need to, you can look at the execution details of the winston workflow at http://winstoncode-useast1c.test.netflix.net:8080/#/history/56314b959287d930566e34a2/general
Alert: test eu-west-1 cass_pay_1-disk_space_critical

winston@ubuntu.netflix.com
to cde-team, pae-winston-al.

Sep 23

Action Needed: True

This alert is generated by Winston - Automated Troubleshooting and Remediation Platform.
Winston execution ID: 5602b71accfde20c9742919b

PROBLEM: Instance: [i-a362390e] for App: [cass_pay_1] is reporting high disk space usage
File system with high usage: md0
File system percent used: 88%
/data/cassandra070/data size in kb: 1533394580
Internal Compaction running: False
Repair/Compaction job running: False
Attempted removal of old snapshots: True

Alert Snapshot: http://alert-history.eu-west-1.test.netflix.net/history/snapshot/cde-cass-disk_space_critical-test/cass_pay_1-disk_space_critical/eu-west-1/14430182180000?checkTime=1443018521473
This is an alert generated by Winston - Automated Troubleshooting and Remediation Platform. Winston execution ID: 56101e8b8fd4d808436c840a

PROBLEM: Following kafka broker instances were reported to be offline.

kafkabroker i-901b5067 - Instance is in 'running' state.

No disk failure detected. Kafka broker restarted Successfully.


If you need to, you can look at the execution details of the winston workflow at http://winstoncde-uswest2a.prod.netflix.net:8080/#/history/56101e8b8fd4d808436c840a/general

If any of the above troubleshooting information has errors in it or if you have suggestions for how it can be improved, please file a JIRA ticket for PAE team using https://jira.netflix.com/secure/CreateIssueDetails!init.jspa?pid=17043&issuetype=4&components=23786&priority=4
The Road Ahead

- **Adoption / Usability**
  - Find common operational use-cases and allow them to be re-used
  - Improve discoverability of Winston by integrating into existing alerting systems
  - Polyglot support (Groovy based runbooks)

- **Safety**
  - Resource isolation using containers
  - Rate limiting capability

- **Stronger analytics**
  - Provide aggregate visualization of runbook executions
Key Takeaways

- Don’t re-invent the wheel
- Start simple and iterate. Have some room for experimentation.
- Start with use-cases where there is more pain and less control over the source of the problem
- Pay special attention to usability of your product
- Push for changing the culture -- Usage will follow
- Find sponsors for features that are much more involved
- Implement best practices through carefully designed user interface instead of documentation.
- Discourage anti-patterns that focus on long term mitigation rather than fixing the root-cause
- Talk to us and others to share insights and learnings
Resources

● Winston Tech Blog link

● Stackstorm documentation
  https://docs.stackstorm.com/

● Reach out
  skarmarkar@netflix.com
  @HikerTechy

We are hiring
Senior Software Engineer - https://jobs.netflix.com/jobs/860752