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

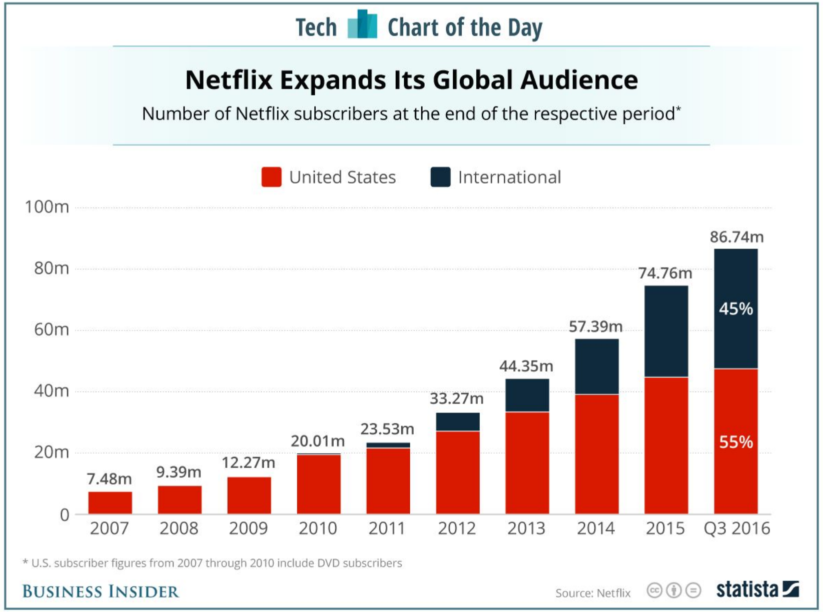
DaRE Team's Focus

Build platforms, tools and libraries
to help teams reduce MTTR for operational issues.

Traditional On-Call Timeline

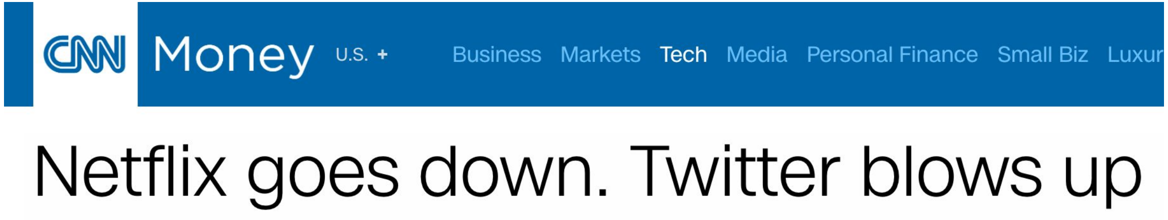
2:00 AM	2:02 AM	2:07 AM	2:10 AM	2:15 AM	2:20 AM	**2:30 AM**
ALERT						✓
PagerDuty Alert	Engineer Wakes up	Logs in and ACK	Studies the alert	Checks runbook	Runs diagnostics	Fixes/Mitigates the problem

Works, but does it scale?!

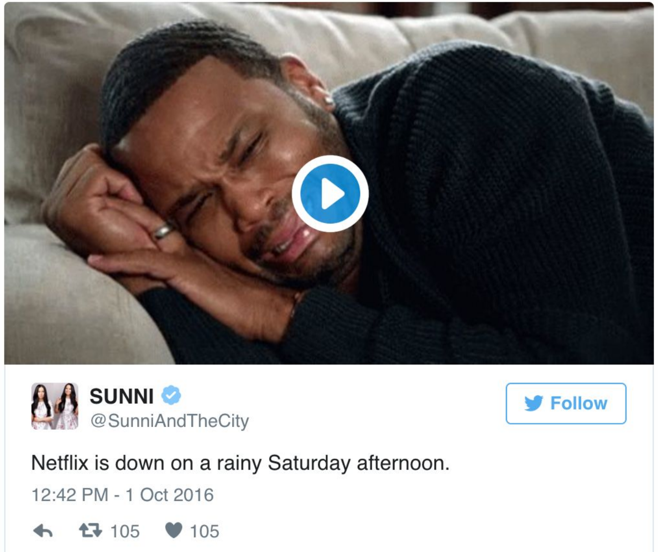


Scale and Growth

Availability



Netflix goes down. Twitter blows up



SUNNI
@SunniAndTheCity

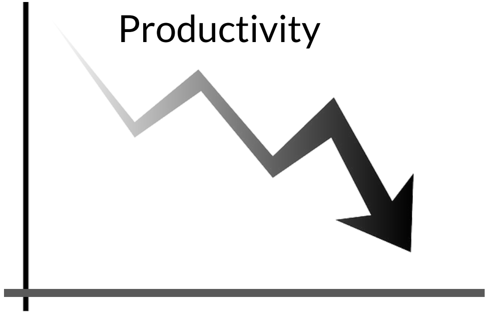
Netflix is down on a rainy Saturday afternoon.

12:42 PM - 1 Oct 2016

105 105

Traditional On-Call Pain Points

MTTR



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 ..

StackStorm

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

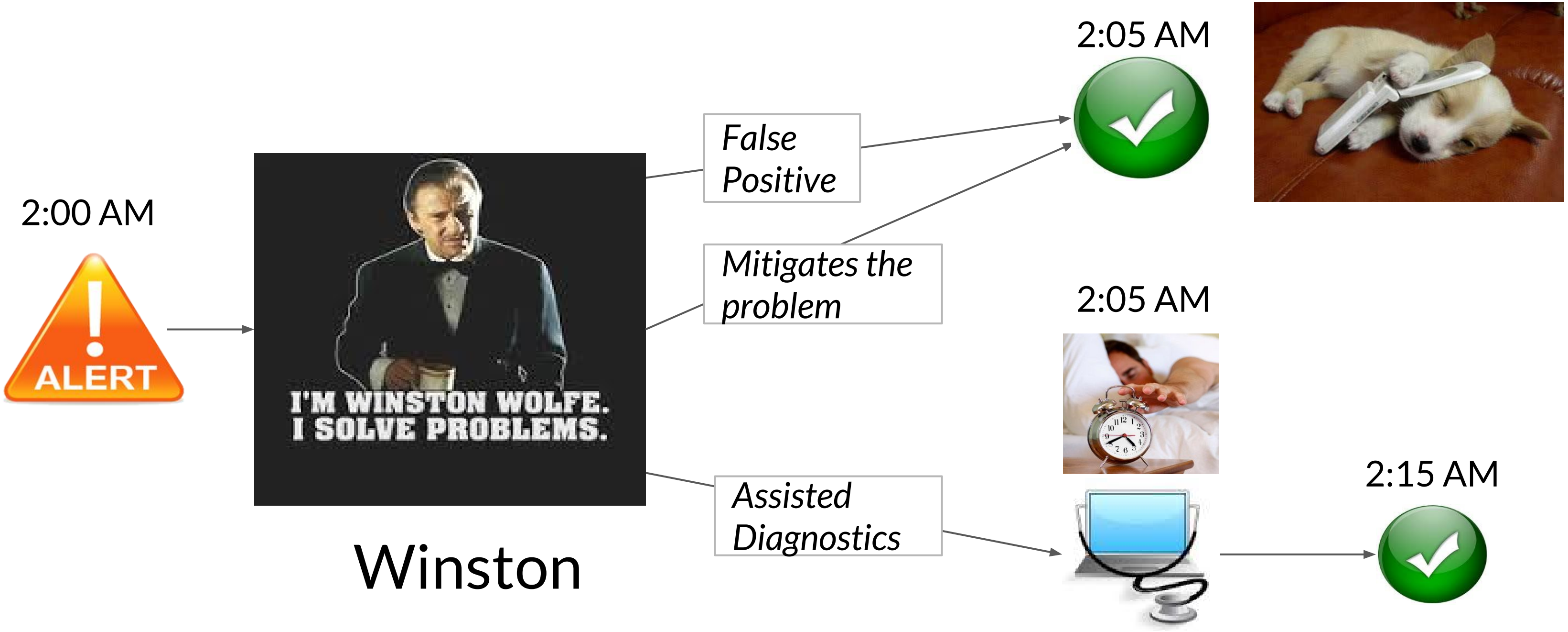
What is Winston?

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	2:02 AM	2:07 AM	2:10 AM	2:15 AM	2:20 AM	**2:30 AM**
PagerDuty Alert	Engineer Wakes up	Logs in and ACK	Studies the alert	Checks runbook	Runs diagnostics	Fixes the problem

On-Call With Winston



2:00 AM

Winston

False Positive

Mitigates the problem

Assisted Diagnostics

2:05 AM

2:05 AM

2:15 AM

Evaluation - Build / Reuse / Buy

Stackstorm

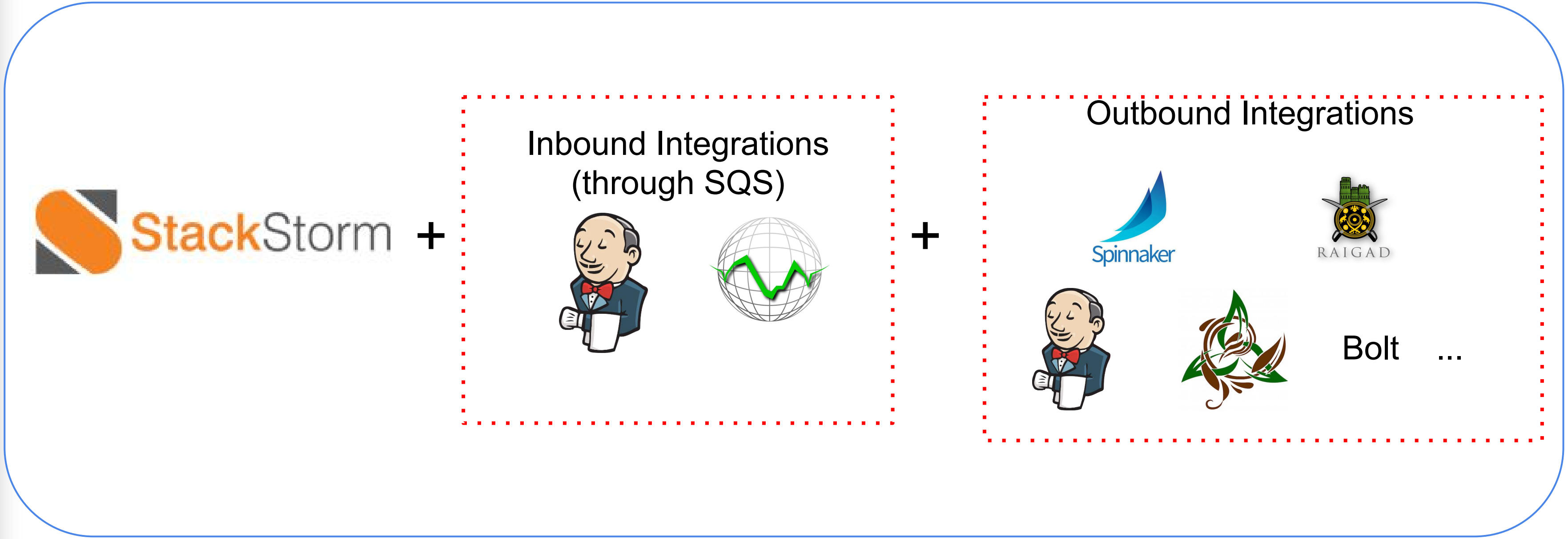
+

- 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 ..



+

Inbound Integrations (through SQS)

+

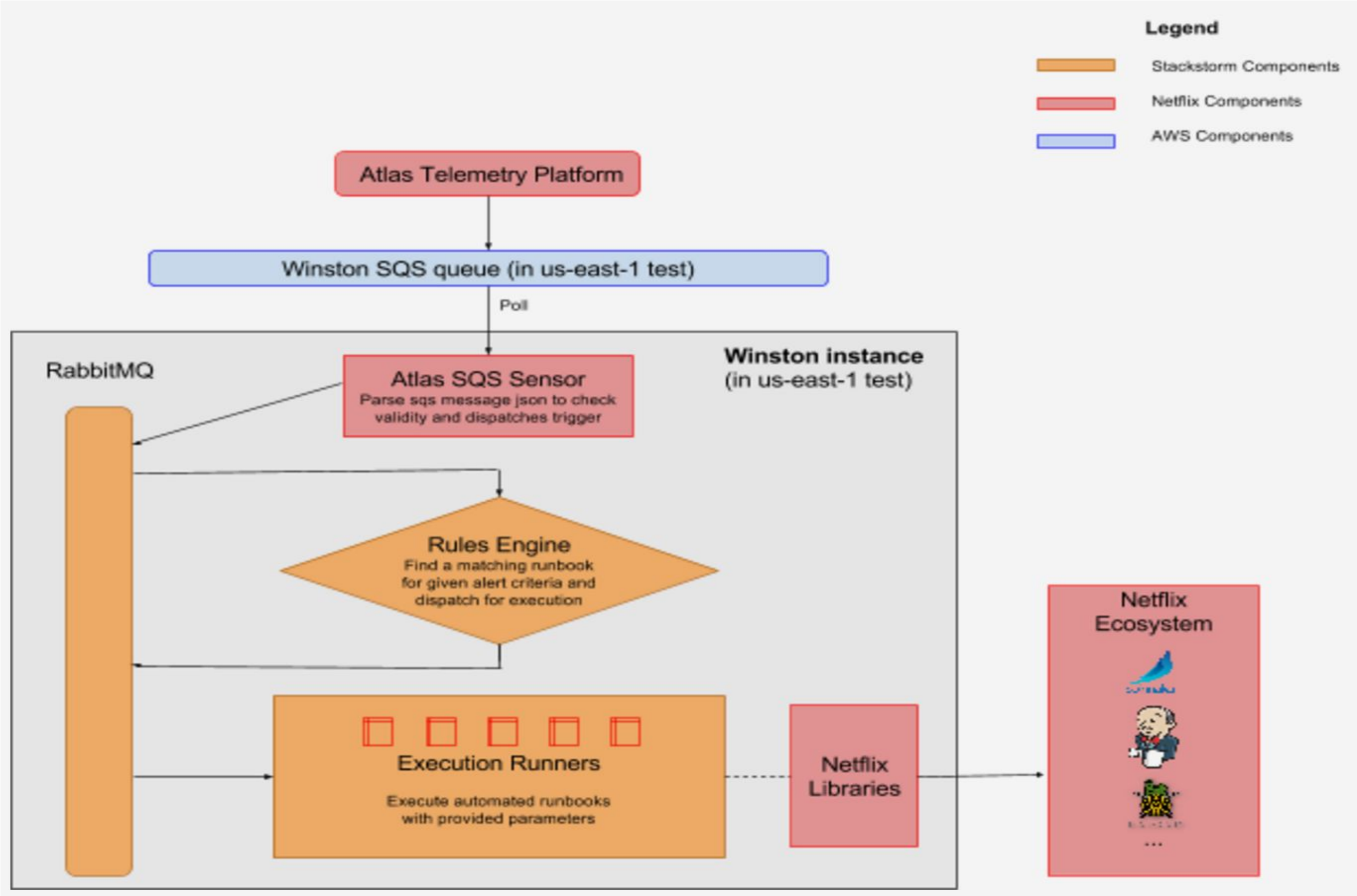
Outbound Integrations

Bolt ...

As a Service (High Availability and Reliability)

Iterate and Evaluate Regularly

A closer look at a Winston Instance



Legend

🟧 Stackstorm Components

🟥 Netflix Components

🟦 AWS Components

Atlas Telemetry Platform

Winston SQS queue (in us-east-1 test)

Poll

Winston instance
(in us-east-1 test)

RabbitMQ

Atlas SQS Sensor
Parse sqs message json to check validity and dispatches trigger

Rules Engine
Find a matching runbook for given alert criteria and dispatch for execution

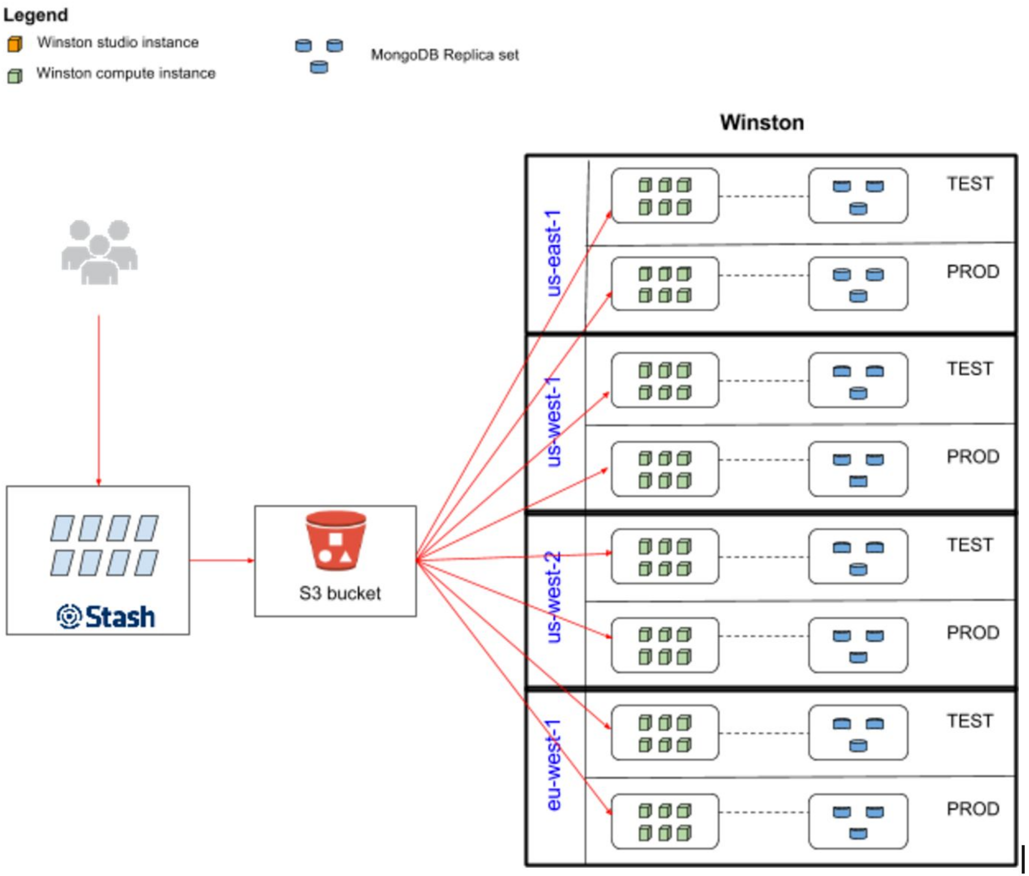
Execution Runners
Execute automated runbooks with provided parameters

Netflix Libraries

Netflix Ecosystem

...

V1.0 Winston HA Deployment



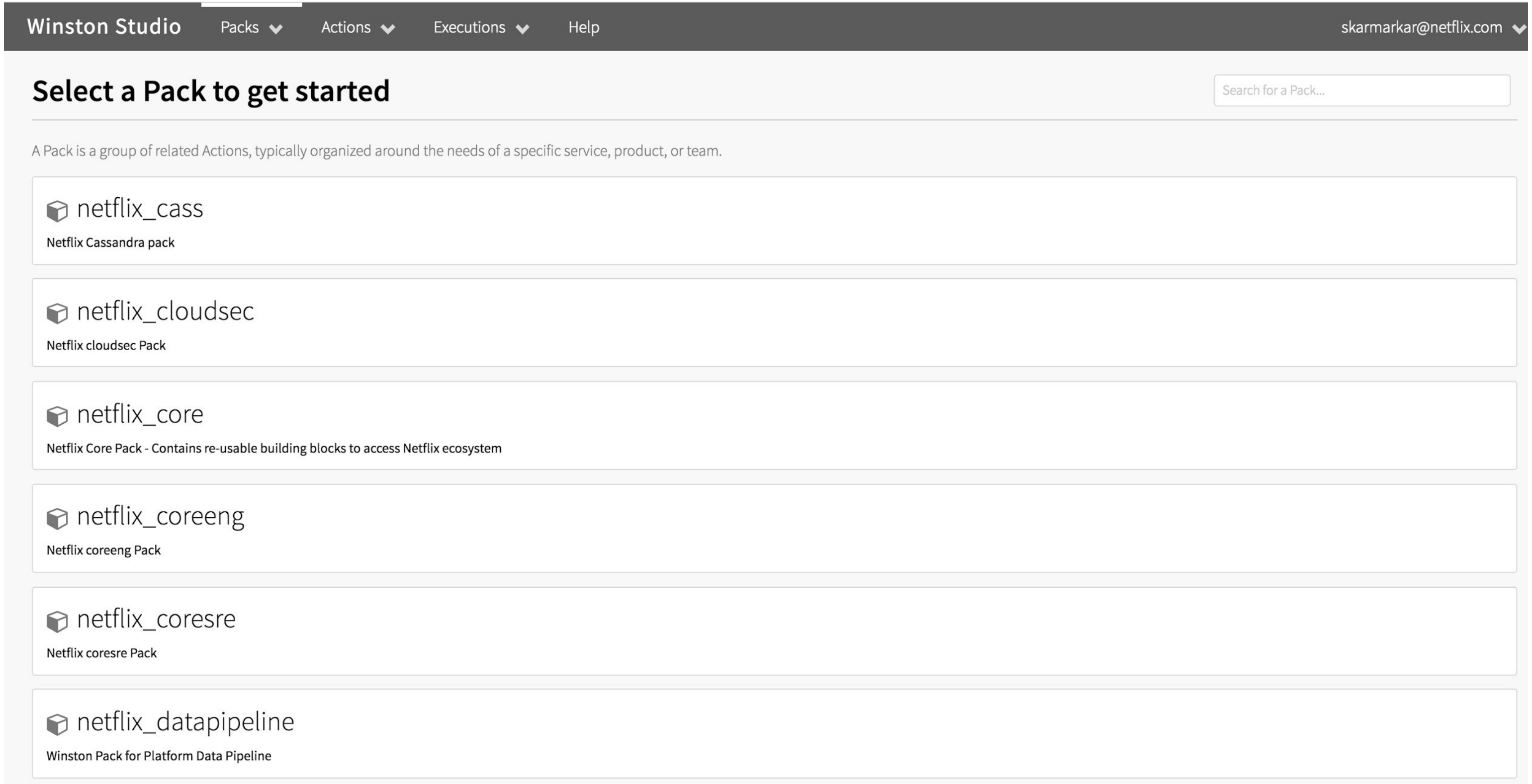
Challenges

- 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 **C**reate, **R**ead, **U**pdate, **D**elete, **E**xecute and **D**iagnose functionality
- Implements best practises
  - Compliance/Auditing
  - Persistence
  - Security (Authentication/Authorization)
- Self serve & scalable

Winston Studio



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.

netflix_cass

Netflix Cassandra pack

netflix_cloudsec

Netflix cloudsec Pack

netflix_core

Netflix Core Pack - Contains re-usable building blocks to access Netflix ecosystem

netflix_coreeng

Netflix coreeng Pack

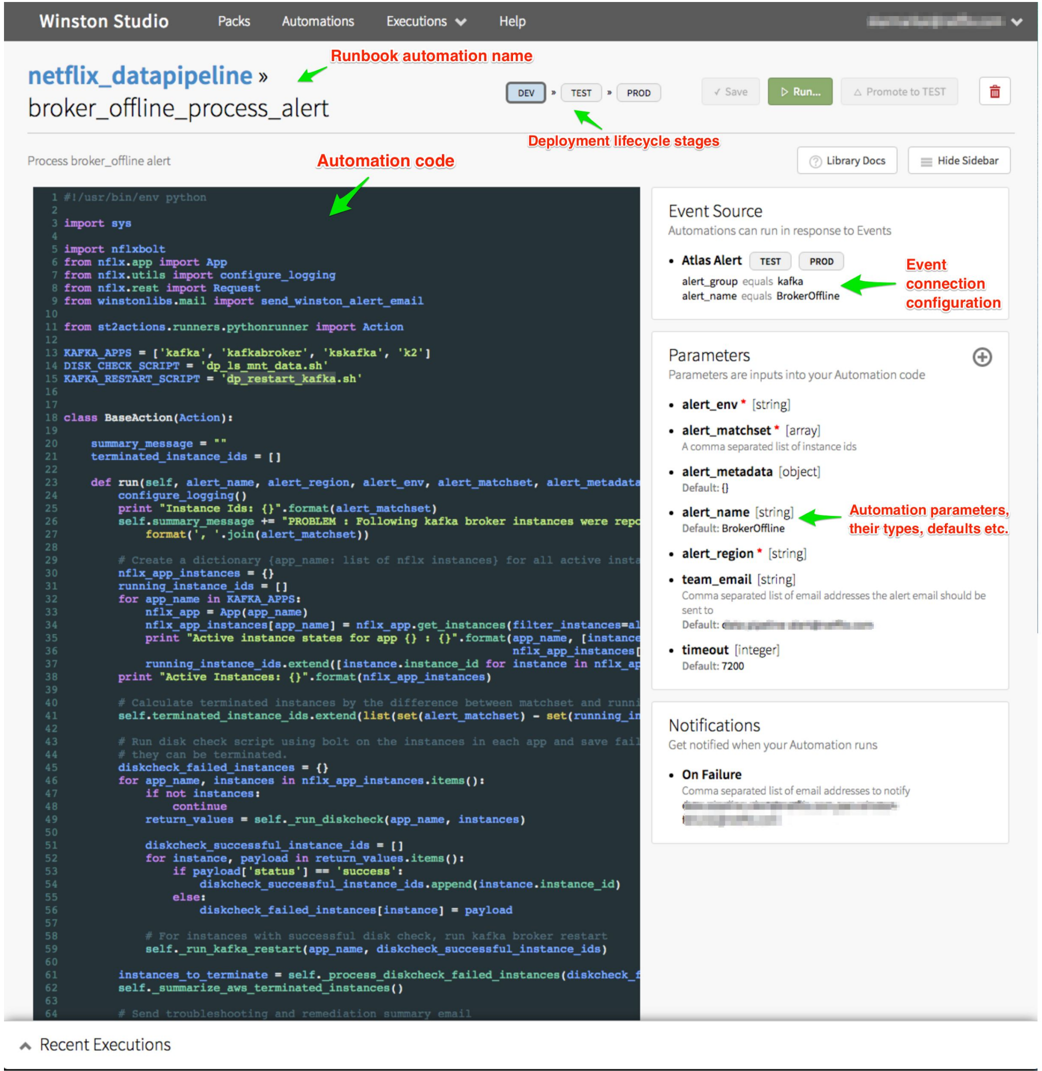
netflix_coresre

Netflix coresre Pack

netflix_datapipeline

Winston Pack for Platform Data Pipeline

Runbook View



netflix_datapipeline »
broker_offline_process_alert

Process broker_offline alert

```
#!/usr/bin/env python

import sys

import nflxbolt
from nflx.app import App
from nflx.utils import configure_logging
from nflx.rest import Request
from winstonlibs.mail import send_winston_alert_email

from st2actions.runners.pythonrunner import Action

KAFKA_APPS = ['kafka', 'kafkabroker', 'kskafka', 'k2']
DISK_CHECK_SCRIPT = 'dp_ls_mnt_data.sh'
KAFKA_RESTART_SCRIPT = 'dp_restart_kafka.sh'


class BaseAction(Action):

    summary_message = ""
    terminated_instance_ids = []

    def run(self, alert_name, alert_region, alert_env, alert_matchset, alert_metadata
        configure_logging()
        print "Instance Ids: {}".format(alert_matchset)
        self.summary_message += "PROBLEM : Following kafka broker instances were repo
            format(', '.join(alert_matchset))

        # Create a dictionary {app_name: list of nflx instances} for all active insta
        nflx_app_instances = {}
        running_instance_ids = []
        for app_name in KAFKA_APPS:
            nflx_app = App(app_name)
            nflx_app_instances[app_name] = nflx_app.get_instances(filter_instances=al
            print "Active instance states for app {} : {}".format(app_name, [instance
                                                                   nflx_app_instances[
            running_instance_ids.extend([instance.instance_id for instance in nflx_ap
        print "Active Instances: {}".format(nflx_app_instances)

        # Calculate terminated instances by the difference between matchset and runni
        self.terminated_instance_ids.extend(list(set(alert_matchset) - set(running_in

        # Run disk check script using bolt on the instances in each app and save fail
        # they can be terminated.
        diskcheck_failed_instances = {}
        for app_name, instances in nflx_app_instances.items():
            if not instances:
                continue
            return_values = self._run_diskcheck(app_name, instances)

            diskcheck_successful_instance_ids = []
            for instance, payload in return_values.items():
                if payload['status'] == 'success':
                    diskcheck_successful_instance_ids.append(instance.instance_id)
                else:
                    diskcheck_failed_instances[instance] = payload

            # For instances with successful disk check, run kafka broker restart
            self._run_kafka_restart(app_name, diskcheck_successful_instance_ids)

        instances_to_terminate = self._process_diskcheck_failed_instances(diskcheck_f
        self._summarize_aws_terminated_instances()

        # Send troubleshooting and remediation summary email
```

Event Source

Automations can run in response to Events

- **Atlas Alert** TEST PROD
  alert_group equals kafka
  alert_name equals BrokerOffline

Parameters

Parameters are inputs into your Automation code

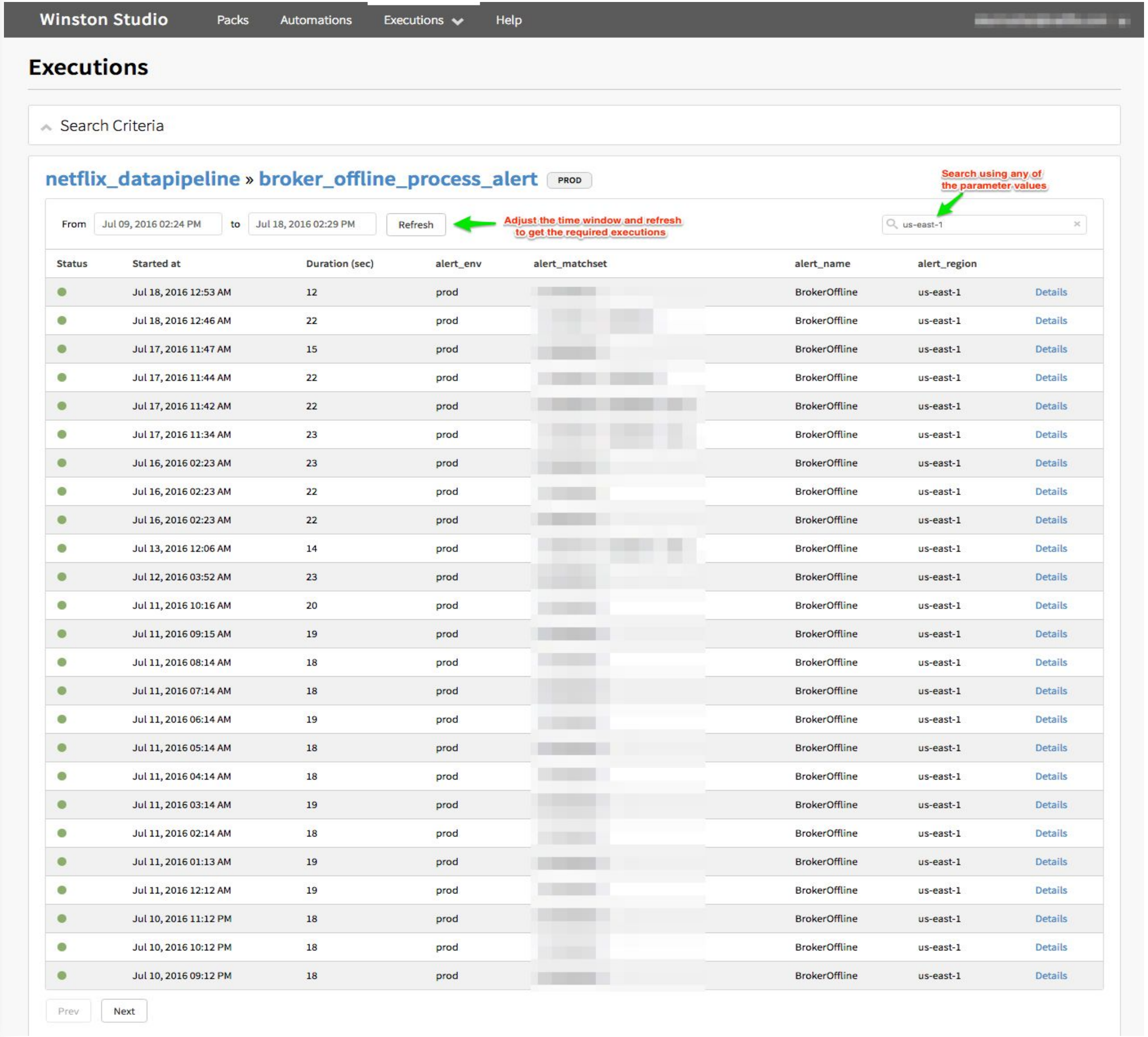
- **alert_env** * [string]
- **alert_matchset** * [array]
  A comma separated list of instance ids
- **alert_metadata** [object]
  Default: {}
- **alert_name** [string]
  Default: BrokerOffline
- **alert_region** * [string]
- **team_email** [string]
  Comma separated list of email addresses the alert email should be sent to
  Default: [redacted]
- **timeout** [integer]
  Default: 7200

Notifications

Get notified when your Automation runs

- **On Failure**
  Comma separated list of email addresses to notify
  [redacted]

Executions



[redacted]

Executions

Search Criteria

netflix_datapipeline » broker_offline_process_alert PROD

Search using any of the parameter values

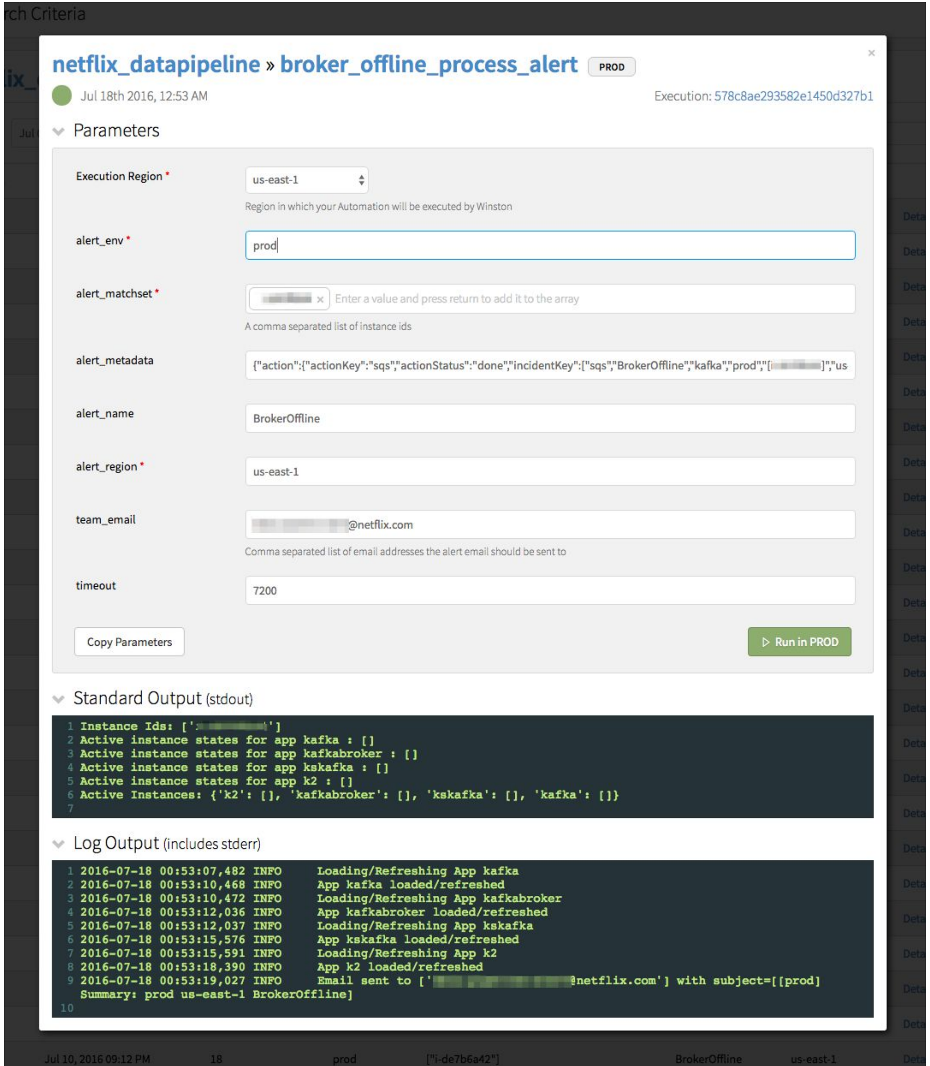
From Jul 09, 2016 02:24 PM to Jul 18, 2016 02:29 PM Refresh

Adjust the time window and refresh to get the required executions

us-east-1

Status	Started at	Duration (sec)	alert_env	alert_matchset	alert_name	alert_region	
●	Jul 18, 2016 12:53 AM	12	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 18, 2016 12:46 AM	22	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 17, 2016 11:47 AM	15	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 17, 2016 11:44 AM	22	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 17, 2016 11:42 AM	22	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 17, 2016 11:34 AM	23	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 16, 2016 02:23 AM	23	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 16, 2016 02:23 AM	22	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 16, 2016 02:23 AM	22	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 13, 2016 12:06 AM	14	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 12, 2016 03:52 AM	23	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 10:16 AM	20	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 09:15 AM	19	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 08:14 AM	18	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 07:14 AM	18	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 06:14 AM	19	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 05:14 AM	18	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 04:14 AM	18	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 03:14 AM	19	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 02:14 AM	18	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 01:13 AM	19	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 11, 2016 12:12 AM	19	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 10, 2016 11:12 PM	18	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 10, 2016 10:12 PM	18	prod	[redacted]	BrokerOffline	us-east-1	Details
●	Jul 10, 2016 09:12 PM	18	prod	[redacted]	BrokerOffline	us-east-1	Details

Execution Details



netflix_datapipeline » broker_offline_process_alert PROD

Jul 18th 2016, 12:53 AM

Execution: 578c8ae293582e1450d327b1

Parameters

Execution Region *	us-east-1 Region in which your Automation will be executed by Winston
alert_env *	prod
alert_matchset *	[redacted] × Enter a value and press return to add it to the array A comma separated list of instance ids
alert_metadata	{"action":{"actionKey":"sqs","actionStatus":"done","incidentKey":["sqs","BrokerOffline","kafka","prod","[[redacted]]","us
alert_name	BrokerOffline
alert_region *	us-east-1
team_email	[redacted]@netflix.com Comma separated list of email addresses the alert email should be sent to
timeout	7200

Copy Parameters   ▷ Run in PROD

Standard Output (stdout)

```
Instance Ids: ['[redacted]']
Active instance states for app kafka : []
Active instance states for app kafkabroker : []
Active instance states for app kskafka : []
Active instance states for app k2 : []
Active Instances: {'k2': [], 'kafkabroker': [], 'kskafka': [], 'kafka': []}

```

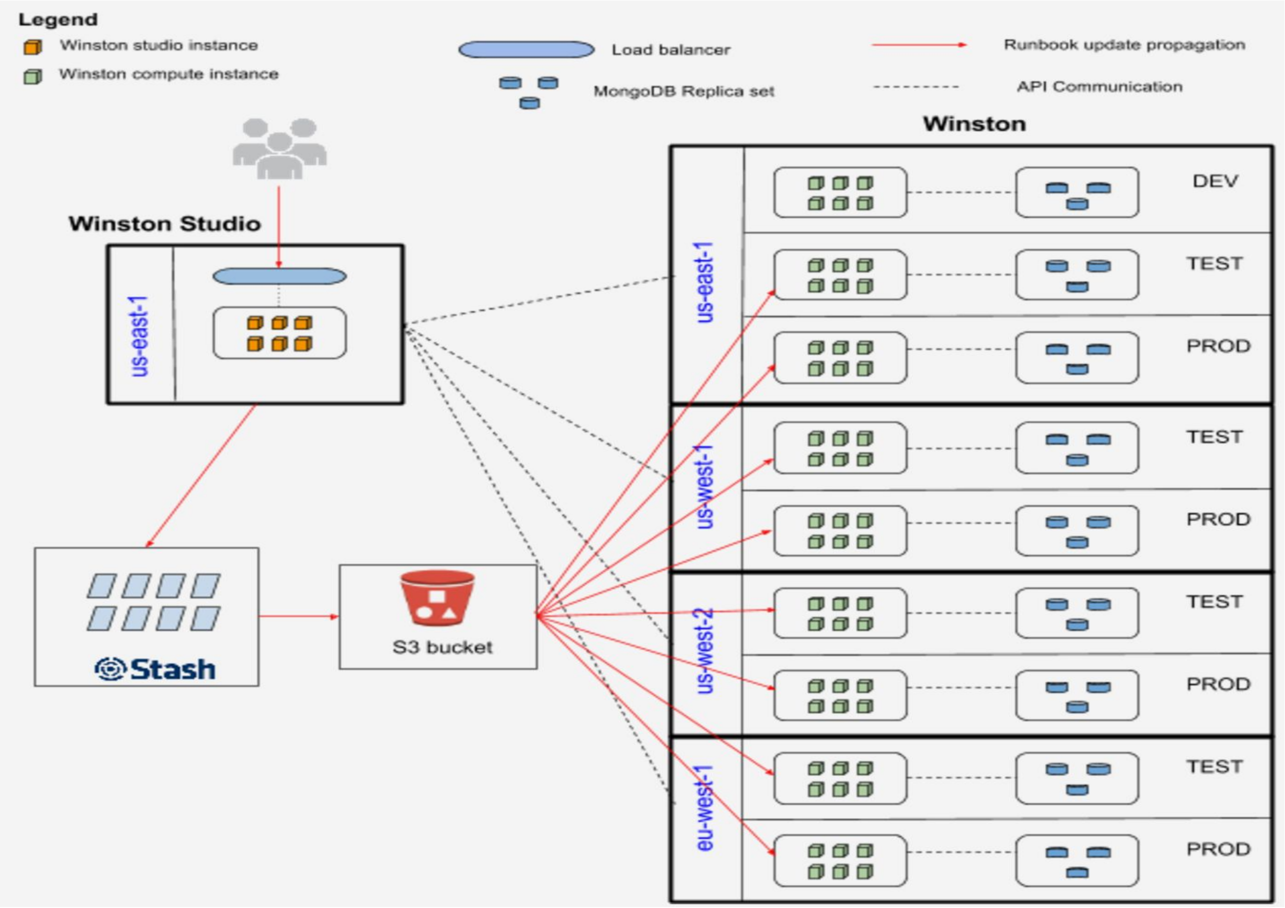
Log Output (includes stderr)

```
2016-07-18 00:53:07,482 INFO     Loading/Refreshing App kafka
2016-07-18 00:53:10,468 INFO     App kafka loaded/refreshed
2016-07-18 00:53:10,472 INFO     Loading/Refreshing App kafkabroker
2016-07-18 00:53:12,036 INFO     App kafkabroker loaded/refreshed
2016-07-18 00:53:12,037 INFO     Loading/Refreshing App kskafka
2016-07-18 00:53:15,576 INFO     App kskafka loaded/refreshed
2016-07-18 00:53:15,591 INFO     Loading/Refreshing App k2
2016-07-18 00:53:18,390 INFO     App k2 loaded/refreshed
2016-07-18 00:53:19,027 INFO     Email sent to ['[redacted]@netflix.com'] with subject=[[prod]
Summary: prod us-east-1 BrokerOffline]

```

Jul 10, 2016 09:12 PM   18   prod   ["i-de7b6a42"]   BrokerOffline   us-east-1

Current Winston Deployment



Use-case Patterns

Sample Use-cases

False Positive

False Positives

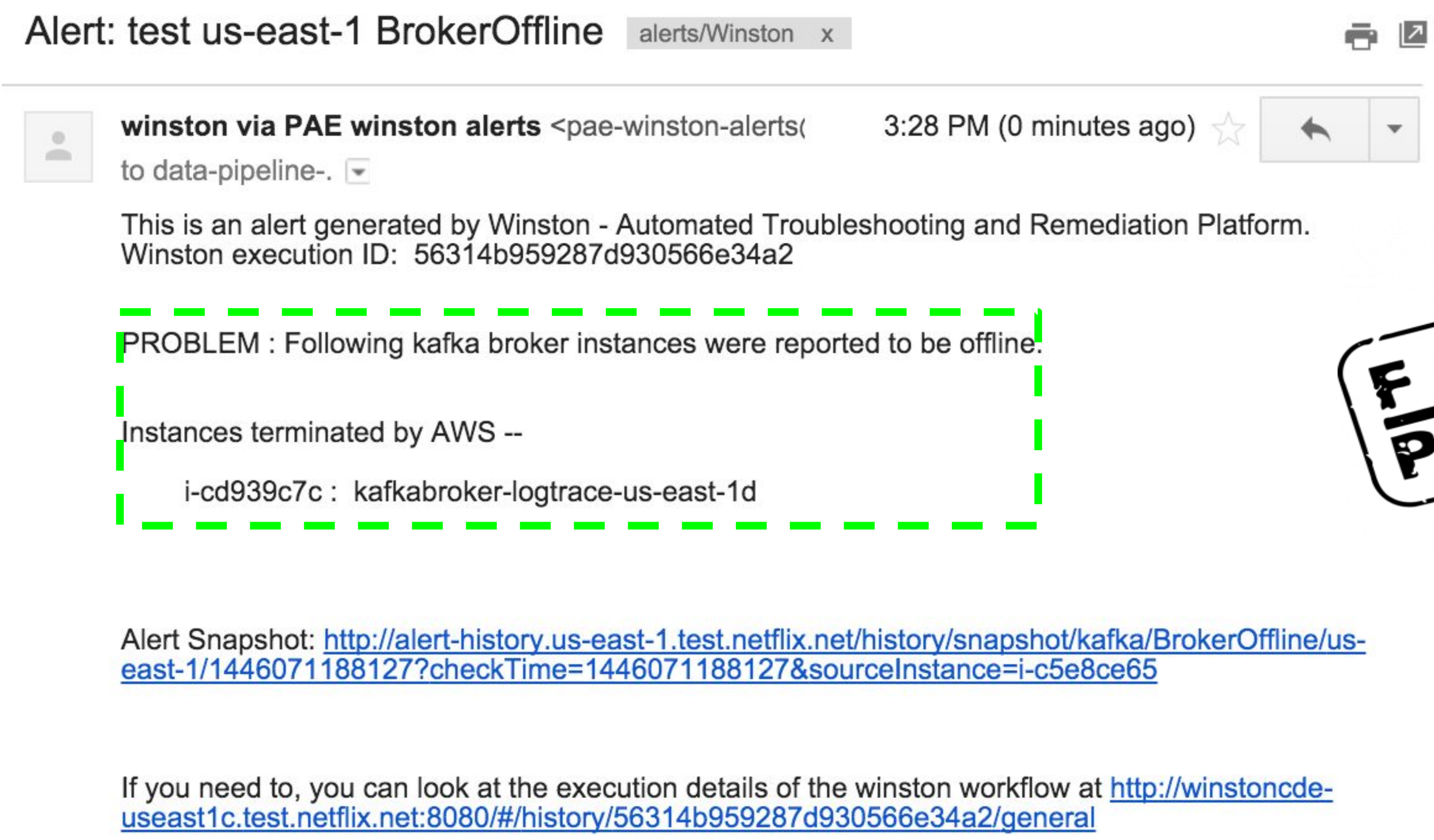
- Broker reporting offline when AWS maintenance takes down an instance
- Cassandra ring health

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.

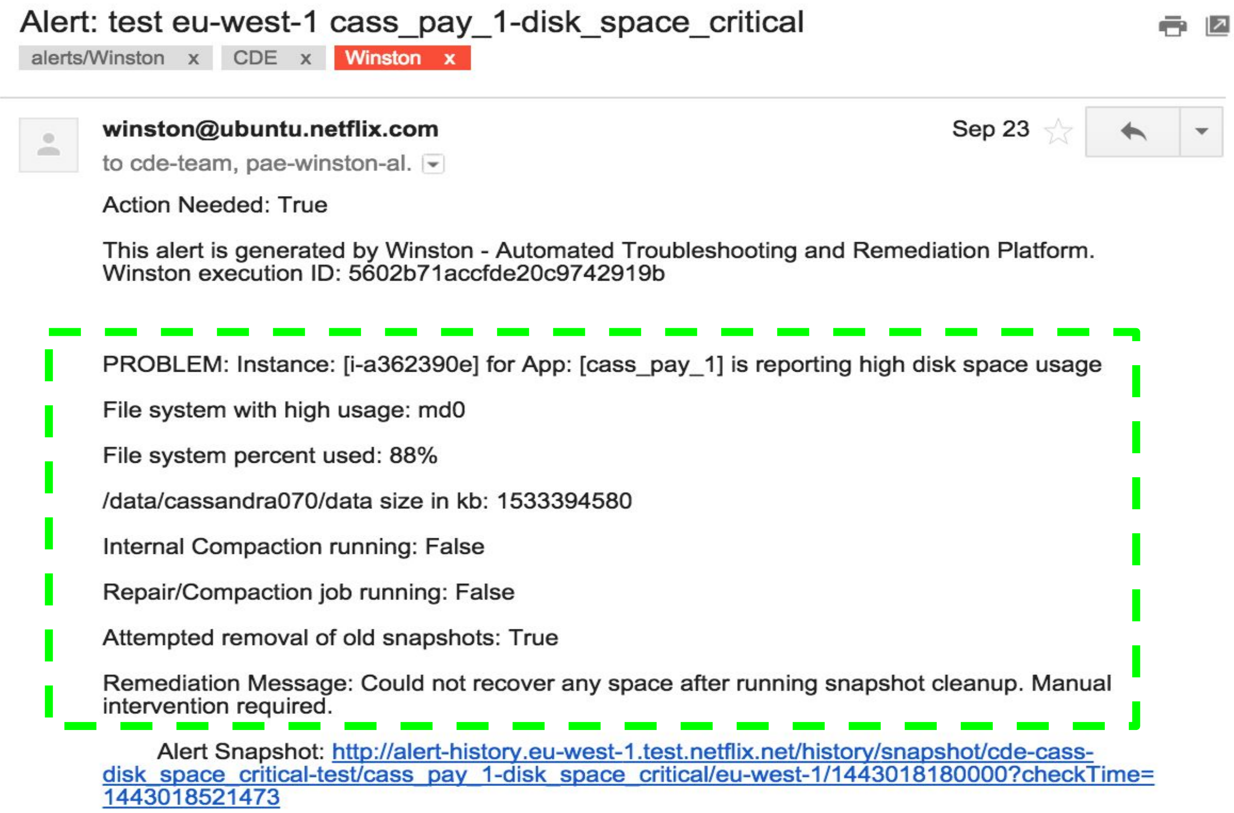
Instances terminated by AWS --

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://winstoncde-useast1c.test.netflix.net:8080/#/history/56314b959287d930566e34a2/general

False Positive



Alert: test eu-west-1 cass_pay_1-disk_space_critical

winston@ubuntu.netflix.com Sep 23

to cde-team, pae-winston-al.

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

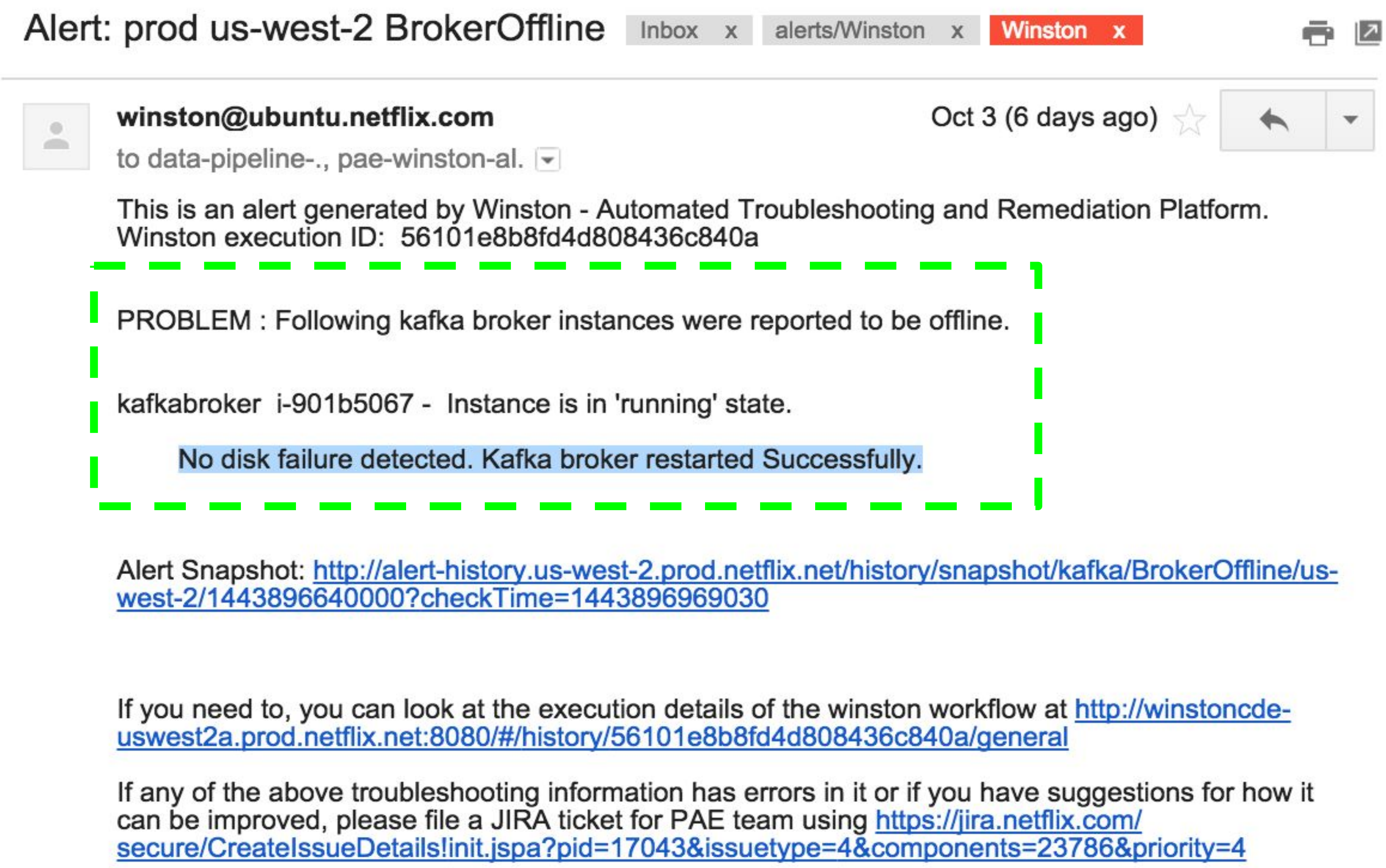
Internal Compaction running: False

Repair/Compaction job running: False

Attempted removal of old snapshots: True

Remediation Message: Could not recover any space after running snapshot cleanup. Manual intervention required.

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-1/1443018180000?checkTime=1443018521473



Alert: prod us-west-2 BrokerOffline

winston@ubuntu.netflix.com

to data-pipeline-., pae-winston-al.

Oct 3 (6 days ago)

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.

Alert Snapshot: http://alert-history.us-west-2.prod.netflix.net/history/snapshot/kafka/BrokerOffline/us-west-2/1443896640000?checkTime=1443896969030

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
  http://techblog.netflix.com/2016/08/introducing-winston-event-driven.html
- Stackstorm documentation
  https://docs.stackstorm.com/
- Reach out
  skarmarkar@netflix.com

  @HikerTechy

We are hiring

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