Bridging the Safety Gap from Scripts to Full Auto-Remediation

David Mah
Dropbox
Scary Tasks
Life of Tooling
Naoru Workflow
Dropbox Storage-team
On-call Maintenance
### Dropbox Storage-team

#### On-call Maintenance

<table>
<thead>
<tr>
<th>Service</th>
<th>Status</th>
<th>Duration</th>
<th>3/3</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapacityCheck-mp-team</td>
<td>CRITICAL</td>
<td>06:13:51</td>
<td>35d 5h 7m 41s</td>
<td>CRITICAL: Capacity Check Failed:</td>
</tr>
<tr>
<td>mp_version_inconsistent</td>
<td>CRITICAL</td>
<td>06:39:30</td>
<td>22d 12h 27m 14s</td>
<td>CRITICAL: Deployments are inconsistent or autoheal is disabled</td>
</tr>
<tr>
<td>puppet_agent_check</td>
<td>CRITICAL</td>
<td>06:30:51</td>
<td>19d 2h 27m 36s</td>
<td>CRITICAL: Puppet had 5 failed resources, last ran 30 minutes and 5 seconds ago</td>
</tr>
<tr>
<td>MpDisksNeedAction</td>
<td>CRITICAL</td>
<td>06:24:32</td>
<td>15d 6h 20m 55s</td>
<td>CRITICAL: 1 disks need actioning</td>
</tr>
<tr>
<td>DbxinitConfigChanges</td>
<td>CRITICAL</td>
<td>06:37:38</td>
<td>13d 11h 5m 48s</td>
<td>CRITICAL: dbxinit has pending changes: (added: mp_volmgr)</td>
</tr>
</tbody>
</table>

...
Scary Tasks
Scary Tasks

Rebooting Servers
Scary Tasks
Rebooting Servers
Hardware Replacements
Scary Tasks

Rebooting Servers

Hardware Replacements

Reformatting Hard Drives
We need to **safely** automate these problems away
Scary Tasks
Life of Tooling
Naoru Workflow
Life of Tooling
Life of Tooling
Manual Debug + Manual Fix
Life of Tooling

Manual Debug + Manual Fix
Alert + Manual Fix
Life of Tooling

Manual Debug + Manual Fix
Alert + Manual Fix
Alert + Script to Fix
Life of Tooling

Manual Debug + Manual Fix
Alert + Manual Fix
Alert + Script to Fix

Alert + Auto Run Script
Life of Tooling

Manual Debug + Manual Fix
Alert + Manual Fix
Alert + Script to Fix
Alert + Human Auth Script
Alert + Auto Run Script
A kernel upgrade is pending.

We must reboot the host to get the upgrade

```
reboot_host ash-ra9-6a
```

May this run? [y/n]
Why should I NOT run this?
Why should I NOT run this?

We can automate that query
Life of Tooling

Manual Debug + Manual Fix
Alert + Manual Fix
Alert + Script to Fix
Alert + Human Auth Script
Alert + Auto Run Script
Life of Tooling

Manual Debug + Manual Fix
Alert + Manual Fix
Alert + Script to Fix
Alert + Human Auth Script
Alert + Auto Run Script

Automate verification of safety
Why should I NOT run this?
Scary Tasks
Life of Tooling
Naoru Workflow
Alerting:

- Automate away ‘finding problems’
Alerting:
- Automate away ‘finding problems’

DiagnosePlugins:
- Find **root cause**
- Decide action to take
DiagnosePlugin Example

SSH Unreachable Alert?

1. If SSH actually works → do nothing
2. If repeated reboot history → deallocate
3. Check out-of-band IPMI console
   i. If no response → reboot
   ii. If (initramfs) shell → deallocate
   iii. If cpu soft lockup → reboot
DiagnosePlugins:
- Find root cause
- Decide action to take

Alerting:
- Automate away ‘finding problems’

Human Authorization Required
Human Authorized Execution

A kernel upgrade is pending.

This remediation will reboot the host (zookeeper participant)

    reboot_host ash-ra9-6a

This is safe because
* We have a rate limit of 1 participant per 5 minutes
* We check for 4/5 participant availability

May this run? [y/n]
Alerting:
- Automate away 'finding problems'

DiagnosePlugins:
- Find root cause
- Decide action to take

Human Authorization Required

pre_remediate hooks:
- Possibly POSTPONE the remediate plugin
  - Rate limits
  - Sanity checks
Example of a Sanity Check

class ZookeeperEnsembleAllUp(Hook):
    """
    If a prescription is destined for a Zookeeper host
    then make sure that all members of its
    ensemble are currently up
    """

def pre_remediate(self, issue_database, remediate_plugin, prescription):
    hostname = prescription.issue.attributes["hostname"]
    if zookeeper_ensemble_all_up(hostname):
        return Hook.EXECUTE_PROCEED
    else:
        return Hook.EXECUTE_POSTPONE
Example of a Rate Limit

Reboot:
matches:
    remediate_plugin:
        - RebootSSH
        - RebootIPMI

group_by:
    - service
limit: 1
Time_window: 300
comment: Reboots are 1 per 5 minutes per-service
**Alerting:**
- Automate away 'finding problems'

**Diagnose Plugins:**
- Find root cause
- Decide action to take

**Remediate Plugin:**
- Actually fix the problem

**Human Authorization Required**

**pre_remediate hooks:**
- Possibly POSTPONE the remediate plugin
  - Rate limits
  - Sanity checks
Alerting

Diagnose Plugins:
- Find root cause
- Decide action to take

Human Authorization Required

Remediate Plugin:
- Actually fix the problem

pre_remediate hooks:
- Possibly POSTPONE the remediate plugin
  - Rate limits
  - Sanity checks

Alerting:
- Automate away 'finding problems'
Alerting:
- Automate away 'finding problems'

DiagnosePlugins:
- Find root cause
- Decide action to take

Human Authorization Required

RemediatePlugin:
- Actually fix the problem

pre_remediate hooks:
- Possibly POSTPONE the remediate plugin
  - Rate limits
  - Sanity checks
Thanks for stopping by!

David Mah
mah@dropbox.com