Shift-Left SRE - Self-healing on OpenShift with Ansible

Jürgen Etzlstorfer, Technology Strategist
@jetzlstorfer

#RedHatOSD
A relentless pursuit of software perfection

1600 Employees
5000 Enterprise Customers
79 of the Global 100

9/10 Cloud providers
8/10 Banks
8/10 Retailers
9/10 Manufacturers
9/10 Tech brands
If you write applications, they will break eventually

~ Murphy’s law
Applications are getting more complex!

On average, a single transaction uses 82 different types of technology.
What if you had something similar to a self-healing robot?
But how?
Building auto-remediation into your pipeline

commit  build  test  stage  prod
✓       ✓      ✓      ✓      ✓
Building auto-remediation into your pipeline

commit ✓ build ✓ test ✓ stage ✓ prod ✗
Building auto-remediation into your pipeline

commit | build | test | stage | prod
✓      | ✓     | ✗    |       |   

Break the pipeline early
Building auto-remediation into your pipeline

Include remediation-as-code
Building auto-remediation into your pipeline

Include remediation-as-code
Self-healing applications

OpenShift Container Platform
Dynatrace Software Intelligence
Ansible Automation
Self-healing building blocks

- **Monitoring**: know what’s going on in your applications
  - End-to-end
  - Full-stack – fully integrated in production

- **Automation/Execution**: perform mitigation/remediation actions
  - Access to all systems

#RedHatOSD
Self-healing with Ansible Tower and Dynatrace

- APIs are key to enable automation
- Ansible Tower provides rich API for managing Ansible jobs
- Playbooks can be orchestrated in workflows and job templates
How to enable auto-remediation

- Full-stack environment is monitored
- Anomalies are detected automatically
- Root cause analysis is performed
- Problem notification is sent
- Event is received
- Job is triggered
- Playbook is executed
- Problem is remediated

#RedHatOSD
How to enable auto-remediation

- Full-stack environment is monitored
- Anomalies are detected automatically
- Root cause analysis is performed
- Problem notification is sent
- Event is received
- Job is triggered
- Playbook is executed
- Problem is remediated

dynatrace + ANSIBLE

#RedHatOSD
Ansible Tower integration in Dynatrace
Ansible Tower integration in Dynatrace
Ansible Tower integration in Dynatrace
What we will see in the demo

- TicketMonster application running on OpenShift
- Full-stack, end-to-end monitoring by Dynatrace
- Feature release via Ansible Tower
- Auto-remediation as code (Ansible playbooks)
DEMO TIME :)
What we have seen in the demo – short recap

- Release of a new feature
- Dynatrace detects increase of failure rate
- Dynatrace fires a problem notification to Ansible Tower
- Ansible Tower kicks off a playbook
- Check for latest deployment with remediation scripts
- Remediation script is executed
- Problem is remediated
Self-healing in an enterprise environment

Escalate at 2AM?

Auto Mitigate!

CPU Exhausted? Add a new service instance!
Self-healing in an enterprise environment

Auto Mitigate!

1. **CPU Exhausted?** Add a new service instance!
2. **High Garbage Collection?** Adjust/Revert Memory Settings!
Self-healing in an enterprise environment

Escalate at 2AM?

Auto Mitigate!

1. **CPU Exhausted?** Add a new service instance!
2. **High Garbage Collection?** Adjust/Revert Memory Settings!
3. **Issue with BLUE only?** Switch back to GREEN!
Self-healing in an enterprise environment

1. **CPU Exhausted?** Add a new service instance!
2. **High Garbage Collection?** Adjust/Revert Memory Settings!
3. **Issue with BLUE only?** Switch back to GREEN!
4. **Hung threads?** Restart Service!
Self-healing in an enterprise environment

1. **CPU Exhausted?** Add a new service instance!
2. **High Garbage Collection?** Adjust/Revert Memory Settings!
3. **Issue with BLUE only?** Switch back to GREEN!
4. **Hung threads?** Restart Service!
5. **Impact Mitigated?**

**Update Dev Tickets**
Self-healing in an enterprise environment

1. CPU Exhausted? Add a new service instance!
2. High Garbage Collection? Adjust/Revert Memory Settings!
3. Issue with BLUE only? Switch back to GREEN!
4. Hung threads? Restart Service!
5. Still ongoing? Initiate Rollback!

Impact Mitigated?

Update Dev Tickets

Mark Bad Commits
Self-healing in an enterprise environment

1. **CPU Exhausted?** Add a new service instance!
2. **High Garbage Collection?** Adjust/Revert Memory Settings!
3. **Issue with BLUE only?** Switch back to GREEN!
4. **Hung threads?** Restart Service!
5. **Still ongoing?** Initiate Rollback!

Mark Bad Commits
Update Dev Tickets
Escalate
Embed auto-remediation in your CI/CD pipeline
Embed auto-remediation in your CI/CD pipeline

Shift-Right: Tags, Deploys, Events

Path to NoOps: Self-Healing, ...
Embed auto-remediation in your CI/CD pipeline

Shift-Left: Break Pipeline Earlier

Shift-Right: Tags, Deploys, Events

Actionable Feedback Loops

Path to NoOps: Self-Healing, ...
Red Hat® Ansible® Tower makes it easy to scale automation, manage complex deployments and speed productivity. Extend the power of Ansible with workflows to streamline jobs and simple tools to share solutions with your team.

Beyond years of industry knowledge in the APM space, Dynatrace offers one of the best solutions I’ve seen for monitoring applications running on OpenShift. What really distinguishes them from others is the use of artificial intelligence based root-cause analysis. OpenShift is a platform to allow you to run decoupled services and applications, which can be a monitoring nightmare, but Dynatrace’s insights makes it less scary.

Chris Morgan, Technical Director – Red Hat OpenShift Ecosystem
GRAZIE PER L’ATTENZIONE

Jürgen Etzlstorfer, Technology Strategist