Making the most of your SRE Toolbox

Bootstrapping your SRE team through reuse

Mark Duquette
SRE Guild Leader, Monitoring and Metrics, IBM Bluemix DevOps Services

Tom Schmidt
SRE Guild Leader, Test Frameworks, Security and Compliance Focal, IBM Bluemix DevOps Services
Introductions

Mark Duquette
SRE Guild Leader
Monitoring and Logging
IBM Cloud DevOps Services

Mark currently works as a Site Reliability Engineer supporting IBM Bluemix DevOps services where he is responsible for the monitoring and metrics infrastructure.

Mark’s expertise and knowledge of designing reusable automation has been used to facilitate teams as they explore SRE as well as embrace DevOps practices.

Tom Schmidt
SRE Guild Leader
Test Frameworks, Security and Compliance Focal
IBM Cloud DevOps Services

Tom currently works as a Site Reliability Engineer in support of DevOps Services at the IBM Canada Lab in Markham, Ontario, Canada.

With a diverse background developing common infrastructure and test frameworks, and a passion for automation, Tom has transformed IBM Cloud development organization's perspective on security.

Tom leverages recent real world experience applying SRE concepts to develop security and compliance solutions within a Continuous Delivery offering.
Our journey begins …

We’ve spent many hours building out test and deployment infrastructure.

We’ve been moved to a team responsible for Site Reliability Engineering.

We wonder how much of that infrastructure will help get the SRE effort off the ground.

Here is what we did …
Overview

Identifying potential candidates for re-use
  – Taking inventory of your toolbox
  – Characteristics to look for in multi-purpose tools

Implementing for re-use
  – Reliance on existing infrastructure (i.e. Jenkins, GIT)
  – Building effective dashboards

Putting it all together
  – Re-purposing and re-use

Our experiences
  – Basic availability monitoring using test frameworks and Jenkins
  – Dashboard to track deployment readiness and system health
  – Building synthetics monitors from test cases
Concepts to keep in mind

Identify
- What areas are the easiest to address with the largest immediate impact
- Which tools are currently used in those areas

Explore
- Go deep. Think about what your existing tools do
- Go wide. Look at re-purposing the functionality

Iterate
- V1 is almost never THE solution; Re-work is not wasted

Evaluate
- There’s always room for improvement and innovation
Identify : Which area should be the first to tackle

Visibility
– Show measurable progress fast and frequently
– Ability to pivot

Manual processes and/or repetitive processes
– Configuring environments
– Applying patches
– Deploying services

Complex and time consuming tasks
– System / software inventories
– Metrics reconciliation
– Service monitoring
## Explore: What’s in your toolbox

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Possibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensible</td>
<td>Test frameworks</td>
</tr>
<tr>
<td>– Is it easy to combine with other tools?</td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>Functional test cases</td>
</tr>
<tr>
<td>– Works or it doesn’t</td>
<td>CI Systems</td>
</tr>
<tr>
<td>Configurable</td>
<td>Security Systems</td>
</tr>
<tr>
<td>– Is everything hard coded or data-driven?</td>
<td>Single-use scripts</td>
</tr>
<tr>
<td>Specific</td>
<td>Deployment automation</td>
</tr>
<tr>
<td>– Performs a single action</td>
<td>“Helper code”</td>
</tr>
<tr>
<td>Complexity</td>
<td>System administration tools</td>
</tr>
<tr>
<td>– More complex == longer to adapt</td>
<td></td>
</tr>
</tbody>
</table>
Tracking availability

**Objective**
Implement a process to track basic service availability

**Tools used**
- Jenkins
- TestNG
- Functional test framework

**Initial approach**
- Monitored site functionality by reusing framework code
- Used the test status as the monitor status
- Scheduled tests to run every minute

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>Name</th>
<th>Last Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>⬜️</td>
<td>⬜️</td>
<td>monitor.addGit-US</td>
<td>9 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>⬜️</td>
<td>monitor.addGit.UK</td>
<td>13 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>☀️</td>
<td>monitor.bluemix</td>
<td>1 min 10 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>☀️</td>
<td>monitor.ids</td>
<td>29 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>☀️</td>
<td>monitor.v2</td>
<td>1.8 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>☀️</td>
<td>v1.pipeline.consumption.monitor</td>
<td>3 min 25 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>☀️</td>
<td>v2.yp.pipeline.consumption.monitor</td>
<td>2 min 0 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>☀️</td>
<td>v2.ys1DEV.pipeline.consumption.monitor</td>
<td>2 min 5 sec</td>
</tr>
<tr>
<td>⬜️</td>
<td>☀️</td>
<td>v2.ys1INT.pipeline.consumption.monitor</td>
<td>2 min 11 sec</td>
</tr>
</tbody>
</table>

*Note: Graphic is for presentation and example purposes only*
Tracking availability

After a few iterations, our tool
- Logged test results to a database
- Provided data on overall availability
- Integrated with incident management
- Allowed us to track deployment budgets
- Has been replaced

### Available Maintenance Budget as of Mar 08

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Allocated(m)</th>
<th>Planned Outage(m)</th>
<th>Unplanned Outage(m)</th>
<th>Budget Remaining(m)</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Week</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>100.00 %</td>
</tr>
<tr>
<td>Current Month</td>
<td>215</td>
<td>0</td>
<td>0</td>
<td>215</td>
<td>100.00 %</td>
</tr>
<tr>
<td>Current Quarter</td>
<td>648</td>
<td>0</td>
<td>0</td>
<td>648</td>
<td>100.00 %</td>
</tr>
<tr>
<td>Current Year</td>
<td>2600</td>
<td>0</td>
<td>0</td>
<td>2600</td>
<td>100.00 %</td>
</tr>
</tbody>
</table>

**Note:** Graphic is for presentation and example purposes only
## Explore: What goes where?

<table>
<thead>
<tr>
<th>SRE Area</th>
<th>Potential tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic health monitoring</td>
<td>Test frameworks</td>
</tr>
<tr>
<td></td>
<td>Functional test cases</td>
</tr>
<tr>
<td>System management</td>
<td>“User” scripts (.profile, etc.)</td>
</tr>
<tr>
<td></td>
<td>Administration utilities</td>
</tr>
<tr>
<td></td>
<td>Runbooks</td>
</tr>
<tr>
<td>Deployment automation</td>
<td>Test frameworks</td>
</tr>
<tr>
<td></td>
<td>Administration utilities</td>
</tr>
</tbody>
</table>

CI servers and dashboards figure into each of the areas above
Iterate: Adapting a tool for SRE activities

Identify the challenge
- What do you want to do?
- How can the tool help achieve the goal?

Brainstorm
- What is the minimum needed to complete the task?
- Think of what the tool currently does and how it applies

Configure and execute
- Try the tool out
- Does it need to be updated?

Rinse and Repeat
- Fine tune and adapt
- Continue to evolve and increase visibility
Deployment readiness and system health

Objective
- Provide information on deployment readiness and system health

Tools used
- Jenkins
- Slack
- Saucelabs
- Delivery pipelines
- Internal web app

Initial approach
- Schedule automated test run intermittently via Jenkins
- Home grown tool targeted test infrastructure team as users
- Change verification requires manual intervention
- Deployment requires infrastructure team
Deployment readiness and system health

A dashboard is born!
- Integrated system test coverage with deployment pipelines for change verification
- Jenkins automated schedule synced with deployment schedule
- Improved usability of dashboard
- Leveraged Jenkins plugins to isolate failures
- Generated Slack notifications to alert component squads
- Compared versions across environments
Building synthetics monitors from functional tests

Objective
– Provide consumption monitoring of services

Initial tools used
– Jenkins
– TestNG
– System integration tests
– Docker

Initial approach
– Build containers containing our test cases
– Make the results accessible through an API for reporting
Building synthetics monitors from functional tests

Our journey continues …
  – We are still iterating on a solution …
  – This is the next step on our journey of re-use
Lessons Learned

Evolve to meet the team’s needs
- Find a solution and tweak it to your needs
- Remember the first solution is never the final one
- Start with the basics and move from there
- We were amazed at how quickly the toolbox expanded

Visibility
- Present solutions, gather feedback

The SRE toolbox is always evolving as the journey continues …
Thank you!