

# 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

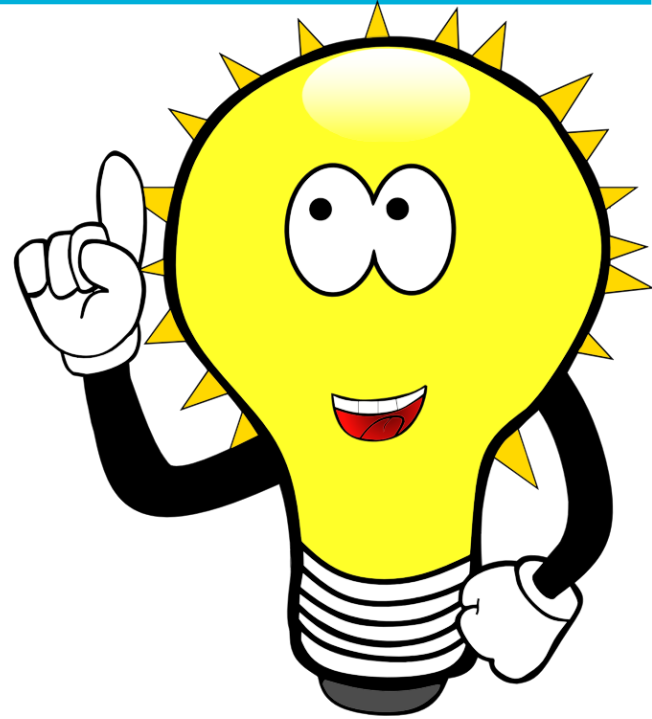
- Go deep. Think about what your existing tools do
- Go wide. Look at re-purposing the functionality

## 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

---

## Characteristics

### Extensible

- Is it easy to combine with other tools?

### Deterministic

- Works or it doesn't

### Configurable

- Is everything hard coded or data-driven?

### Specific

- Performs a single action

### Complexity

- More complex == longer to adapt

## Possibilities

Test frameworks

Functional test cases

CI Systems

Security Systems

Single-use scripts

Deployment automation

“Helper code”

System administration tools



# 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

S	W	Name ↓	Last Duration
		<a href="#">monitor.addGit-US</a>	9 sec
		<a href="#">monitor.addGit.UK</a>	13 sec
		<a href="#">monitor.bluemix</a>	1 min 10 sec
		<a href="#">monitor.ids</a>	29 sec
		<a href="#">monitor.v2</a>	1.8 sec
		<a href="#">v1.pipeline.consumption.monitor</a>	3 min 25 sec
		<a href="#">v2.yp.pipeline.consumption.monitor</a>	2 min 0 sec
		<a href="#">v2.y1DEV.pipeline.consumption.monitor</a>	2 min 5 sec
		<a href="#">v2.y1INT.pipeline.consumption.monitor</a>	2 min 11 sec

*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

Monthly Availability Last 12 months							
Month	Uptime(m)	Planned Outages		Unplanned Outages		Extraordinary Events	
		# Outages	Duration(m)	# Outages	Duration(m)	# Outages	Duration(m)
Mar 2017	11174	0	0	0	0	0	0
Feb 2017	40320	0	0	0	0	0	0
Jan 2017	44640	0	0	0	0	0	0
Dec 2016	44640	0	0	0	0	0	0
Nov 2016	43200	0	0	0	0	0	0
Oct 2016	44640	0	0	0	0	0	0
Sep 2016	43200	0	0	0	0	0	0
Aug 2016	44640	0	0	0	0	0	0

Available Maintenance Budget as of Mar 08					
Time Period	Allocated(m)	Planned Outage(m)	Unplanned Outage(m)	Budget Remaining(m)	Availability
Current Week	50	0	0	50	100.00 %
Current Month	215	0	0	215	100.00 %
Current Quarter	648	0	0	648	100.00 %
Current Year	2600	0	0	2600	100.00 %

Note: Graphic is for presentation and example purposes only



# Explore: What goes where?

---

SRE Area	Potential tools
Basic health monitoring	Test frameworks Functional test cases
System management	“User” scripts ( .profile, etc. ) Administration utilities Runbooks
Deployment automation	Test frameworks Administration utilities

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

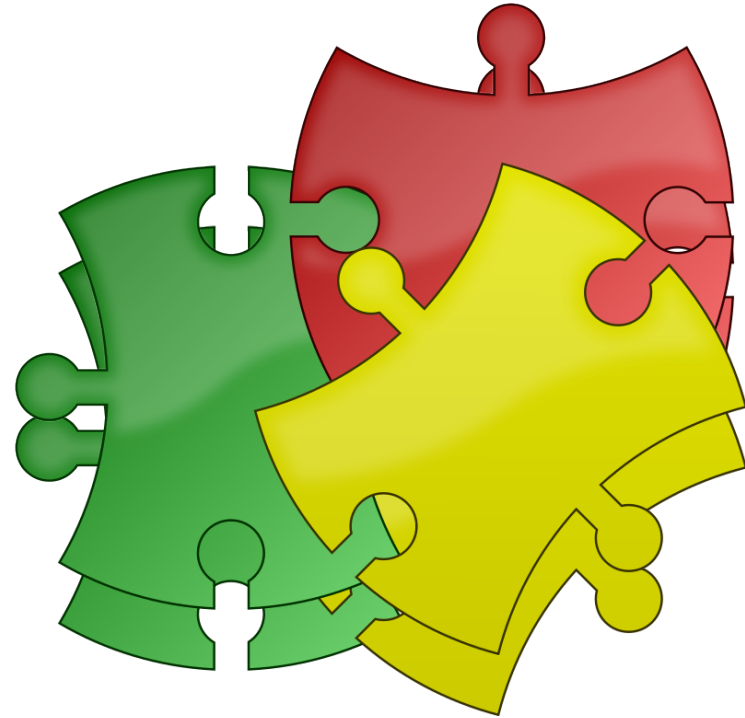
- What is the minimum needed to complete the task?
- Think of what the tool currently does and how it applies

## 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

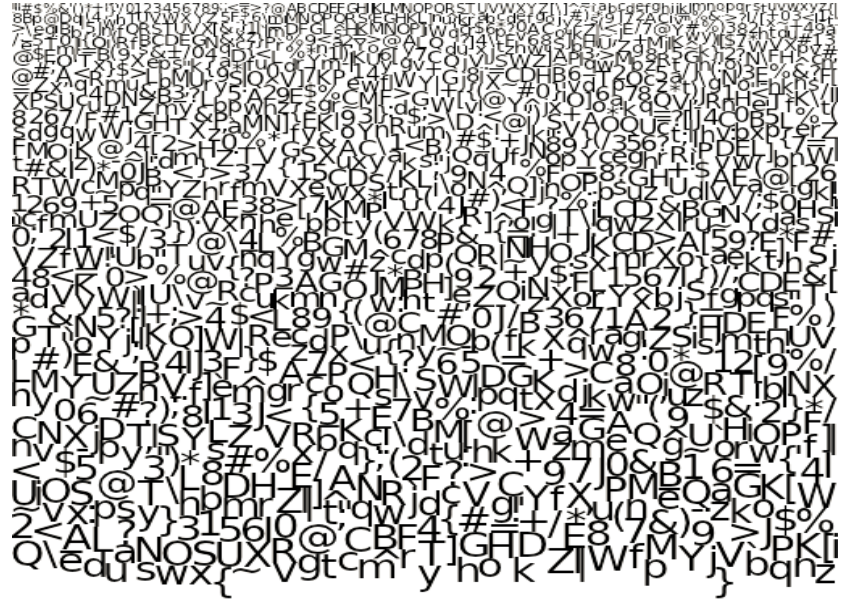
- Provide information on deployment readiness and system health

## Tools used

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

## Initial approach

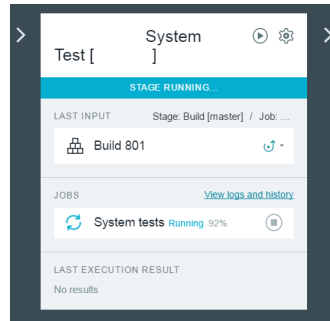
- Schedule automated test run intermittently via Jenkins
- Home grown tool targeted test infrastructure team as users
- Change verification requires manual intervention
- Deployment requires infrastructure team



# 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



```
functional-test APP 11:51 PM
added and commented on this Plain Text snippet
1 [TestCase 23:51:30,184] ERROR Unable to find javascript redirect in response
2 [TestCase 23:51:30,185] ERROR STATUS: HTTP/1.1 503 Service Unavailable
3 [TestCase 23:51:30,902] ERROR login failed!
4 [TestCase 23:51:32,829] ERROR
5 Failed to retrieve version.

User: <username@emailAddress.com>
Source: <Component> : <Pipeline Stage> : <Job Number>
Environment: <dev environment>
Test: <link to descriptive java doc>
Details: <Component> service was unavailable due to <reason>
```

SL-PROD	SL-QA	SL-DEV
20161201-1704	20161201-1704	20161223-1456-Revert-the-revert
20160420-161e-af2ec0c152817045c71adcd9d0e9ec2c31c3fc69	20160420-161e-af2ec0c152817045c71adcd9d0e9ec2c31c3fc69	20160420-161e-af2ec0c152817045c71adcd9d0e9ec2c31c3fc69
20151120-1157	20151120-1157	20151120-1157
20170203-migration-items	20170203-migration-items	20170217-1045
PSLoginDevOneRing_1.0.0-I20170213-1542	PSLoginDevOneRing_1.0.0-I20170213-1542	PSLoginDevOneRing_1.0.0-I20170213-1542
PSLoginDevOneRing_1.0.0-I20170213-1542	PSLoginDevOneRing_1.0.0-I20170213-1542	PSLoginDevOneRing_1.0.0-I20170213-1542
2017-03-01_18-33-09_3589_code_dev-minify	2017-03-01_18-33-09_3589_code_dev-minify	2017-03-10_03-54-34_3638_code_dev-minify
20150831-1144	20150831-1144	20161025-1112
20160608-1345	20160608-1345	20161025-0952



# 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!