# Autonomous deployments

## An inside out approach to automated releases



#### The Agenda

- Bio
- Team / Context
- The challenge
  - Defining our why
- Problem statement 1 ES Rally
- Problem statement 2 E2E performance testing
- Demo
  - Universal Local
  - Build pipelines
- K8s vs blue green
- New toys
- Review / Summary
- Conclusion

# About Me:



# A history brief

- Core platforms team
- Woolies as a business
- Project approach
- Talk outline
- Test Areas



# The Challenge

"The challenge of the unknown future is so much more exciting than the stories of the accomplished past."

- Simon Sinek

## **Defining our 'Why'**

- App's to prod faster
- Low maintenance
- Attempt to reduce risk
- Manageable + scalable
- Consistency
- Accountability



# Problem 1: ElasticSearch Upgrade

#### **Rudy's rutabaga rule:**

Once you eliminate your number one problem, number two gets a promotion

- Jerry Weinberg



#### Sort by: Relevance ~





**Banana Organic each** 

Woolworths Banana Bread Slice



Add to cart 📜

=

# Issue

- Version upgrade 2.4 -> 5.6
- New indexes
- Document structure changes
- Perf testing was done through the frontend



#### The Baseline







# Problem 2: E2E performance testing

"Life is essentially an endless series of problems. The solution to one problem is merely the creation of another."

- Mark Manson

# Issue

- Testing was at the end of the Dev process
- Lack of transparency
- Lack of accountability
- Expensive process

# Solution

## **Component based performance test framework**

- Run locally and at each CD stage
- Uses docker and K8s
- Use of Helm (run as a job)
- reporting , metrics and alerting

# Helm



A package manager : Homebrew, APT, Chocolatey
Simplified deployment and testing
Yaml config (chart + templates)
Release naming versioning

#### Helm Template

wow-loa	d-test / templates / job.yaml 🛞 Fork 🖞 Clone
Conte	ents History Compare Blame   🖉 Edit 🛋 Rename ៣ Delete 🚽 Download 🗸 🧷
1 {·	{- \$deploymentName := .Release.Name -}}
2 a	piVersion: batch/v1
	ind: Job
4 m	etadata:
	<pre>name: {{ \$deploymentName }}</pre>
	labels:
	<pre>app: {{ \$deploymentName }}</pre>
	<pre>namespace: {{ .Release.Namespace }}</pre>
9 s	pec:
10	template:
11	metadata:
12	labels:
13	app: {{ \$deploymentName }}
14	spec:
15	imagePullSecrets:
16	- name: wowdevcontainers
17	containers:
18	<pre>- name: {{ \$deploymentName }}</pre>
19	<pre>image: "{{ .values.image.repository }}:{{ .values.image.tag }}"</pre>
20	<pre>imagerullrollcy: {{ .values.image.pullrollcy }}</pre>
21	
22	
23	
25	value: {{ Values arureDevOnsPat }}
25	
20	value: {{ Values.azureDevOnsFeed }}
28	- name: PACKAGE NAME
29	value: {{ ,Values.packageName }}
30	- name: PACKAGE VERSION
	value: {{ .Values.packageVersion }}
32	volumeMounts:
	- name: config-secret
24	mountDath, "/ucr/local/imater config"

#### Helm Values file

w-load-test / values.yaml	🚸 Fork 🖞	Clone
Contents History Compare Blame 🖉 Edit 🛋 Rename 🛅 Delete 🛓 Download		
1 # Default values for wow-load-test.		
2 # This is a YAML-formatted file.		
3 # Declare variables to be passed into your templates.		
5 alerts:		
6   slackChannel: "#test-alerts"		
8 image:		
9 repository: wowdevcontainers.azurecr.io/platform-engineering/load-testing		
10 tag: 0.1-alpha./		
11 Command. []		
13 pullPolicy: Always		
14		
15 azureDevOpsOrgUrl: https://wowonline.visualstudio.com/		
16 azureDevOpsFeed: LoadTests		
17		
18 nameOverride: ""		
19 fullnameOverride: ""		
20		
21 environment: {}		
22		
23 secrets: {}		
24		
25 resources:		
20 requests:		
27 Cpu. 1999m 28 memory: 1 5Ci		
20 ] limits:		
30 cou: 1000m		
31 memory: 1.5Gi		
32		
33 config: {}		
34		

#### The Demo

#### START WHERE YOU ARE.

USE WHAT You have.

### DO WHAT YOU CAN.

**ARTHUR ASHE** 

SCRATCH PAPER STUDIO



# Universal packages

A named and versioned collection of files

vsts package universal download \

--instance

- "https://wowonline.visualstudio.com/" \
- --feed "LoadTests" \
- --name "samples-dotnet-webapp" \
- --version "1.0.62-0.1-demo-1" \
- --path .

vsts package universal publish \
 --instance
"https://wowonline.visualstudio.com/" \
 --feed "LoadTests" \
 --name "samples-dotnet-webapp" \
 --version "1.0.62-0.1-demo-1" \
 --description "Sample web app Load test" \
 --path .

#### **Demo - Execution**

Steps:

- Create directory
- Pull down the package
- Inspect the package/image
- Run the tests

# Stitching it all together

"Everything is created twice, first in the mind and then in reality."

— Robin Sharma

#### Each build pipelines



#### Demo 2 - Reject Canary

- Deploy the app to UAT and run the load/functional tests
- Poll the webapp and load test pods
- Review the docker entrypoint
- Look at the config.yaml
  - Cancel the Canary

#### Demo 2 - Accept Canary

- Build to UAT
- Trigger a Proceed
- New version gets pushed out
- Canary cleaned up



# Where we would like to be:



# New Toys

- Docker
- Kubernetes (AKS)
- Helm
- Universal packages
- Azure-pipelines
- Blazemeter Taurus
- IaC
- ML and AI
- Canary

## The Review

"You don't have to be great to start, but you have to start to be great."

~ Zig Ziglar

#### Review

- We defined our 'why' early on
- We built the solutions then applied partial support
- Detailed wiki and readme's helped
- We used tools available and being analysed by the team
- We started with real world issues

## The Summary

### Summary

- Make use of the internal resource skills
- Using a 'core' team for designing and building tools internally has its benefits
- Use of the 'infinite mindset'
- Implementing process that supports team culture
- Building tools that were consistent across the business
- There's a lot of benefits by using a 'throw-away' mentality

## Thank you

Contact: <u>https://linkedin.com/in/scgriffiths/</u>