Roke Part of the Chemring Group

How Roke moved an on-premise CI-based project to AWS

...and the lessons we learned along the way

7th March 2019

aws TEST 🕄 DEPLOY O -**O**-422cbad6 -**O-**2ac4de -0--0--0--0-4f52ed2a c6d045cf d9f8e0aa 722ac4de $\odot \odot$ $\odot \odot \odot$ $\odot \odot \odot$ CI/CD 0 0 0 0

Charles Woolf, Senior Software Engineer

About Roke

- Up the road at Romsey
- •~350 engineers
- Predominantly project-based organisation
- Work primarily in the defence and national security domains
- Majority of projects use agile/scrum/kanban
- Big focus on automated testing and CI/CD





About the team

- Scrum team
- Sprints are two weeks long
- Customer is product owner
- Engineers estimate tasks
- Devs use Linux or Windows (their choice)
- Gitlab as our Git hosting, CI, and collaboration platform





About the project

- Software hub
- JavaScript frontend & backend
- Python microservice add-ons
- MongoDB Data store
- Elasticsearch-backed search
- RabbitMQ for IPC
- Active Directory for auth
- Requirements: 16GB RAM, 4+ cores per instance





Typical nightly pipeline

Lint		 Build Packages		Build Containers	Test	Package		Terraform	Instance Testing	Upload Artefacts
 Dockerfiles 	0	Addon SDK	0	Addon: Analyse 0	Auto uploader 0	Addons	0	Terraform Branch	Test deployme 0	O Upload Archives
O Linux line endi	0	O Addon template	0	Addon: Analyse Q	Backend Unit T 0	(⊂) Core	0	Terraform Prod		O Upload Docum
O Python	0	O Auto uploader	0	Addon: Analyse 0	⊖ Frontend unit O					⊖ Upload Execut
Shell scripts	0			Addon: Analyse 0	⊘ Integration tests ©					⊖ Upload Misc.
⊘YAML	0			O Addon: Auto u 0	Nightwatch Sa 0					O Upload Sources
				Addon: Checks	⊘Nightwatch St 0			1 Lint		
				Addon: Extract	Nightwatch St			T.FILL		
				Addon: Extract 0	Nightwatch Str			2.Build pa	ckages	
				⊘ Addon: Labeller Q	⊘ Python Stress 0			3.Build co	ntainers	
				Addon: Metada Q	⊘ Python System Q			4 Test (th	longost stags	.)
				Addon: Summa O	⊘ Python System Q			4.1651 (116	e iongest stage	-)
				⊘ Addon: Summa O	OPython Unit Te O			5.Package		
				Addon: Summa 0	⊘ Python Whole O			6.Deploy	review/prod ir	frastructure
				Core: Backend				7 Teet year		
				Core: Frontend				7. lest rev	iew infrastruct	ure
				⊘ Core: Migratio Q				8.Populate	e with represe	ntative data
				⊘ Core: Migratio 0				Stens 1-1	run on every nuch	
				Core: Nginx				Steps 1-41	un on every push	



- Infrastructure
 - Three servers (ESXi VSAN cluster)
 - 15 Gitlab runner VMs
 - Handful of test deployment environments (master, nightly, current release)
 - Internet access shared by the rest of the company





• Cost

- New servers are expensive!
- Engineer time is valuable
- Waiting for infrastructure is wasteful
- Engineer time to set up and maintain servers/runners
 - Even though the majority of work is automated
- Power, rack space, cooling, networking...





- Scaling up
 - Adding a new runner VM takes about 30 mins of engineer time
 - Adding a new server takes 1-2 weeks depending on data center staff availability and procurement
- Scaling down
 - Can't return servers to Dell
 - Servers are per-project capex so can't easily be transferred between projects







- Infrastructure
 - One managed Kubernetes cluster (EKS)
 - 0-50 cluster nodes, depending on usage
 - Internet access through AWS
 - Much closer to lots of mirrors
 - Uncontested Internet connection
 - Representative environments for every branch, and a production/reference instance





• Cost

- \sim £100/month for managed Kubernetes
- Nodes charged on a PAYG basis
 - £0.07 per hour per node
- Easy to estimate the cost of CI
 - 1 job running for 1 hour = 1 job hour
 - 2 jobs running for 2 hours = 4 job hours
 - Total job hours * EC2 hourly cost = cost of a pipeline
 - ~10p per full pipeline run
- Scaling completely automated (no cost)





Scaling

- ~30 seconds to add a new node to the cluster
 - Completely automated zero engineer time
- Scaling down automated and transparent
 - Unused/underutilised nodes are removed after 5 minutes
- Can scale across environments
 - CI runners could be run at site for things that interface with hardware







SENSORS & INFORMATION | COUNTERMEASURES & ENERGETICS

- Kubernetes
 - Big and complex
 - Lots to learn! A little at a time...
 - Completely different way of thinking
 - Experiment with Minikube
 - Security considerations in a CI environment





• AWS

- Different infrastructure paradigm
- Focus on reproducibility and automation
- Learn by doing
- Some concepts transferable/adaptable
- Security in the cloud





- Migration
 - Expect some things to break!
 - Modify CI jobs to make the most of the cloud
 - Parallellise your workloads
 - Automation of key infrastructure (DNS, VMs, VPN)
 - Storage volumes and objects
 - Automation of backups





Summary

Thorough, fast testing can quickly outgrow traditional infrastructure, or grow at a pace that can't be matched by buying and integrating more servers.

Buying hardware costs time and money – and you can only really scale up.

Moving to a more dynamically scalable (elastic) infrastructure can speed up the development and test cycle.

It's a big undertaking to move everything to the cloud – the value is paid back over time rather than immediately.

