



The Cloud Native Stack - Docker & Kubernetes

The Problem

A major US logistics firm pursuing a mobile-first digital strategy needed a modern and scalable back-end to support its microservices based-architecture to be developed on a public cloud subscription and be deployable to both test and production on that cloud .

Among the challenges that needed to be addressed were:

- Automated deployment of cloud network resources, security controls, load balancing, compute and database resources that minimized costs upfront then would quickly scale to meet production workload and security requirements.
- A CI/CD pipeline integration into existing Jenkins and Bitbucket toolsets to support rapid-cycle, test-driven, application development.
- The development team needed high cost efficiency during early development without reworking the deployment when the application was ready to move to production.

The Solution

After careful analysis, a solution was designed that consisted of:

- Infrastructure as Code deployed with Terraform to stand up Kubernetes with a small number of compute nodes for early development and a PaaS PostgreSQL database service.
- Kubernetes deployment files that define the application containers, load balancer configuration, and quantity making it simple for developers to control application response times -- and balance performance against cost.
- All source code and configuration settings for the entire system are version-controlled; stored in a Git-compatible source-code repository and integrated into the company's CI/CD toolset.
- A common container management process enabled Developers to use Docker Compose for local development.

Notification	Pagerduty	Teams	xMatters	e-mail	Slack	NOC	Manage & Notify
Enterprise Platforms	Office 365	Teams	G-Suite	ServiceNow	Slack	Atlassian	
Observability	Splunk	Nagios	SCDM	AppDynamics	Datadog	Prometheus	Monitor
Security	Cisco	Nessus	Tripwire	Qualys	Carbon Black	CrowdStrike	
Operating Systems	Windows	Linux (CentOS)	Linux (Red Hat)	Linux (Debian)	FreeBSD	Unix	Run
Compute	HPE	Cisco UCS	Dell	IBM	Lenovo	White Box	
Storage	EMC	Pure	HPE	SPAR	NetApp	White Box	
Databases	Oracle	MS SQL	MySQL	MongoDB	IBM DB2		
Identity & Access	AD	Gemalto	Okta	Cyberark	Open LDAP	SiteMinder	
Cloud	AWS	Azure	IBM	Digital Ocean	On-Premise		
Orchestration	Ansible	Terraform	Saltstack	Puppet	OpenStack	Custom	
Testing	Locust	Selenium	HP Loadrunner	IBM Rational	JMeter	JUnit	
Source Control	Azure DevOps	Gitlab	Bitbucket	SVN	Artifactory		Build
Scripting / Programming	Python	Powershell	Java	Go	Ruby		



The Result

- Deployed and operated the development environment at 1/10th the cost of production with no additional investment required to scale to production workload.
- Reuse of deployment code to construct environments that were not originally planned.
- Production containers on workstations sped up development and decreased production defects by enabling functional tests prior to code commit.

