Optimising K6 Test Execution in VPE

This presentation outlines enhancements to our K6 performance testing strategy within the VPE environment, focusing on improved efficiency and resource management.

R **by Rahul Jain**

Current K6 Test Execution Flow

1 Post-Install Job Execution

Tests are executed within the post-install job of the primary harness.

2 Sidecar Deployment

K6 tests leverage a sidecar container for execution resources.

3 No Canary Support

Current setup does not support K6 testing for canary deployments.

4 Developer-Defined Durations

Test durations are fixed based on developer input due to job step time constraints.

Challenges with Current Approach

Resource Contention

Sidecar usage competes for resources with the main application.

Time Limitations

Post-install job timeouts restrict comprehensive test runs.

The existing K6 setup introduces resource bottlenecks and time limitations, hindering effective performance validation.

Introducing the New K6 Strategy

Independent Job Execution

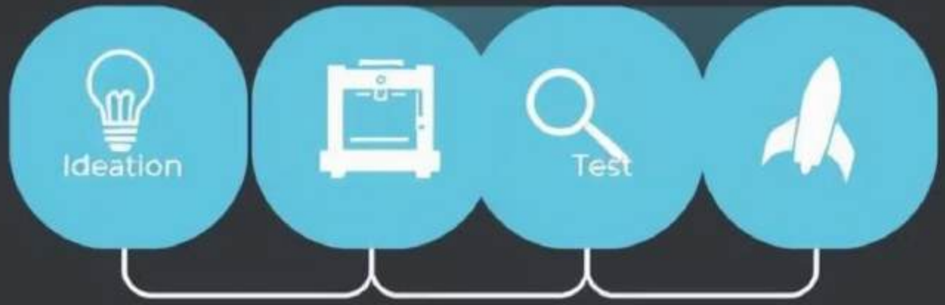
K6 tests run as a secondary, independent Kubernetes job.

Sidecar Elimination

The sidecar is removed, freeing up primary application resources.

Unrestricted Duration

Independent jobs overcome post-install time limits for longer tests.



Enhanced Deployment Process

'oc' Logic Integration

Custom 'oc' logic is added to orchestrate job creation.

Dynamic Job Generation

`Job.yaml` and `ConfigMap` are dynamically created.

Environment Variable Utilisation

Configurations leverage post-install job environment and Helm values.

The new process automates K6 job creation and configuration, leveraging existing environment variables for flexibility.

Monitoring and Logging

Background Job Monitoring

Primary harness monitors the secondary K6 job's completion status.

Live Log Streaming

Secondary job logs are streamed to the primary job until timeout or completion.

Graceful Exit

K6 job gracefully exits, preventing harness job step failures.

Real-time monitoring and seamless log integration ensure visibility without disrupting the main deployment.

Cleanup and Finalisation

1 Job Completion Wait

Primary job waits for secondary K6 job to fully complete.

2 Resource Deletion

Completed K6 jobs and associated ConfigMaps are cleaned up.

3 Ensuring Idempotency

Cleanup ensures a clean state for subsequent deployments.

A structured cleanup routine ensures no residual artifacts remain, maintaining system hygiene.

Key Benefits and Next Steps

40%

Resource Optimisation

Reduced resource contention for main applications.

100%

Testing Flexibility

Unrestricted test durations for comprehensive analysis.

15%

Deployment Reliability

Improved stability of post-install job execution.

This new approach promises significant improvements in efficiency, flexibility, and reliability of our K6 performance testing in VPE. We anticipate a smoother deployment pipeline and more accurate performance insights.