Potential Experiments/areas to tackle

Scenario

- Existing Egeria cohort including metadata repository, UI components
- Kafka already deployed
- I want to add a connector to a new technology
    - build my **integration** connector / metadata access store
    - create my image
    - deploy it in k8s
    - must work as part of existing cohort

Proof points

#	Area	Objective	Current Status	Who	
1	Containers	Build customized containers with only the content required for a particular usage: • Start with integration connector • figure out approach to gathering correct runtime dependencies easily • figure out parameterization needed to make this a useful starting point for orgs • Exclude: Value: Can help in short term with existing model - already see need for this	Already being looked at for IBM/HMS	Nigel	
2a	Application	Create a streamlined application which can startup a single server (and shutdown cleanly - we have concern here with egeria) when application is launched, based off an existing configuration. Should be runnable from the command line, and just launch as a process. It can consume files/environment variables (in k8s these can be mapped from config maps, secrets etc). will not update configuration Figure out how much, if any, of existing platform/admin is needed/can be reused. Specifically needs to be able to run a server hosting an integration connector Value: Could be used even in short-medium term as a convenience to start single platform		Ljupcho ++ David	
3	Operator	Create custom resource definition for a 'server' (just 1 per 'platform' aka java process) & create an operator that can deploy/undeploy a pod containing running this server, as well as a service based on a pre-existing configuration: • probably Java (rather than go) - general team skills? • use autostart as an initial 'hack' Value: Allows for active management in k8s environment	see egeria-kubernetes-operator for a) platform operator in go b) initial port of above to java	Nigel	
4	Document	Document our design principles ie only use of ephemeral storage (create another wiki page?) Value: Information sharing, reviews, consensus	Design Principles for Cloud native Egeria	all	
5	Configuration /storage	Determine how egeria server has no dependency on persistent storage – ie which connectors/mechanisms are needed for storage ie potentially including • Configuration • Cohort registry • audit log • (metadata repositories) Consider use of existing kubernetes resources - directly or via mapping - config maps, secrets, custom resources Look at read/write, concurrent access from multiple servers or replicas (for example we know config documents get written to during startup) All must be accessible from a k8s operator in addition to other ways Develop appropriate connectors and/or techniques to support Excluded: security connector Value: Simplification and ultimately critically important in k8s environment	Existing operator uses configmaps	nigel taylan	

6	Configuration	Ensure configuration 'makes sense' at runtime. Review configuration For example endpoints may refer to other servers within a k8s cluster, or to 3rd party technologies externally. Late binding is exactly needed. Consider metadata collection id - may need to assert value Some information required to be kept secret - auth info, certs. these must be able to be pointed to - ie in a key store, or something like a k8s secret Develop proposal to handle. For example it may be sufficient to use hostnames which can be easily defined by an org in dns (static environments) or by k8s services	we don't think this is an issue - just preconfig the metadata colection id - host mapping should work for endpoints ▪ certs etc point to locations on filesystem - just need to map to mounted volumes from configmaps etc.,	n/a	
7	microprofile vs spring	• what benefits? • interoperable with spring services?	same 7 + 8	n/a	
8	footprint	Can we get a integration connector to deploy and work reliably in 1Gb or less? How low can we go Can we improve with spring? Does microprofile help?		n/a	
9	scenario	Combine above into a broader demo environment		n/a	
10		configuration authoring		n/a	
	readiness, liveness, probes /healthchecks	Service dependencies, monitoring. How is this exposed. application needs to expose . k8s monitoring. part of application		Taylan	