SPARK SUMMIT

RUNNING SPARK INSIDE CONTAINERS

Haohai Ma, IBM

Khalid Ahmed, IBM

Myself

- “How High”
- Software Architect
- IBM Spectrum Computing
- Toronto Canada

Agenda

- Why container?
- Migrate spark workload to container
- Spark instance on Kubernetes
  - Architecture
  - Workflow
  - Multi-tenancy
- Future work

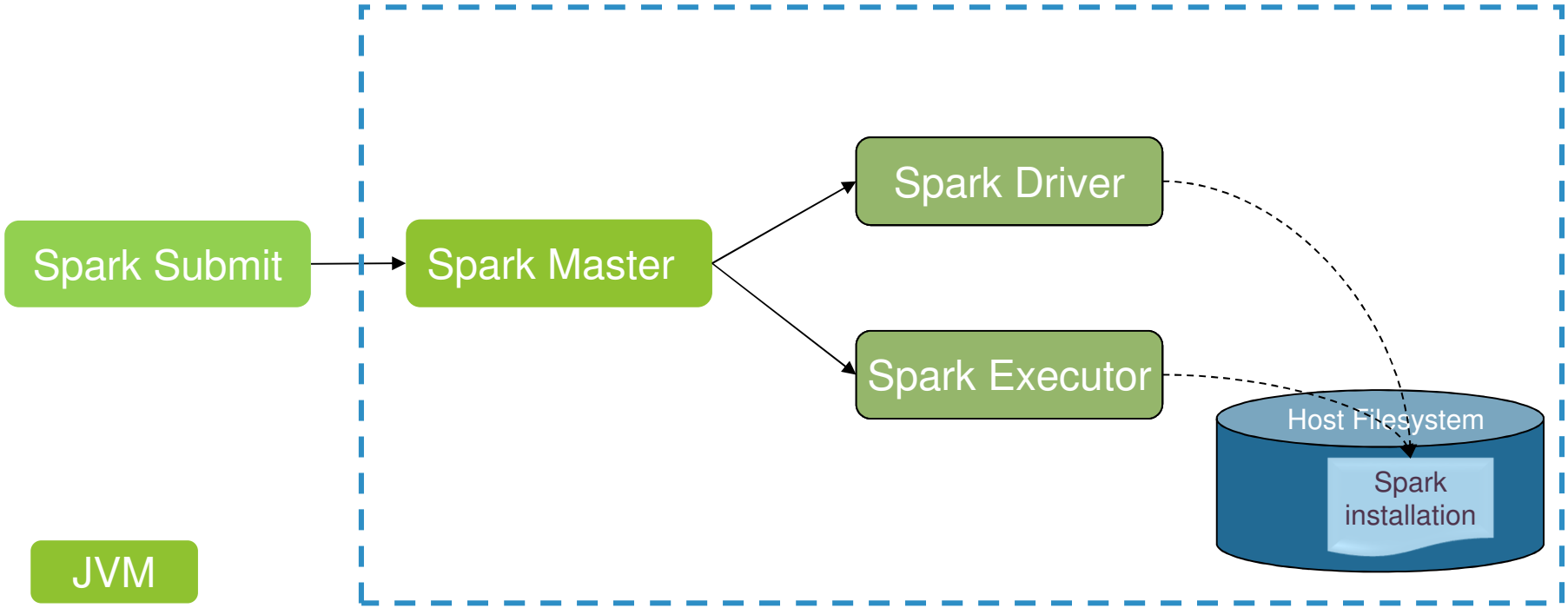
Why use containers?

- To enforce the CPU and memory bounds.
  - CPU shares are proportional to the allocated slots
  - spark.driver.memory & spark.executor.memroy
- To completely isolate the file system
  - Solve the dependency conflicts
- To create and ship images
  - Develop once and run everywhere

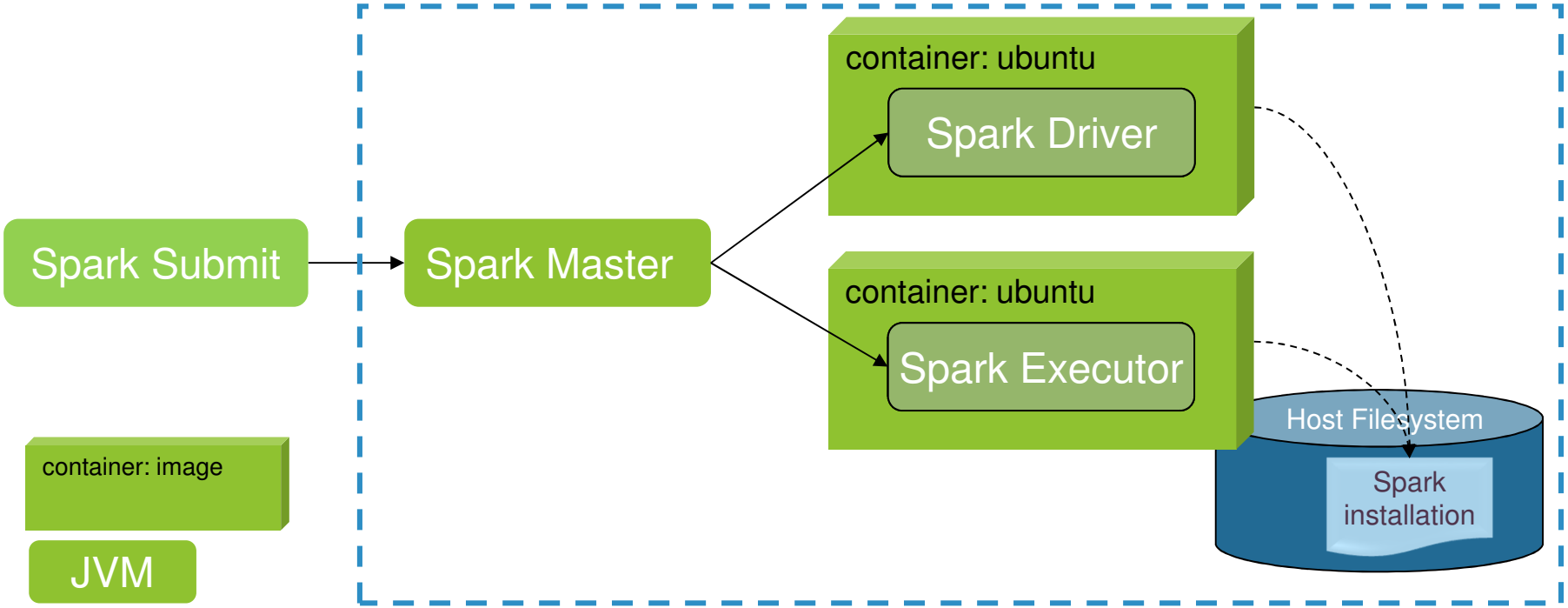
No prebuilt Spark image

- A running container needs an application image
  - Independent to Spark versions
- Seamlessly migrate Spark workloads to a container based environment
  - Assume: Spark is distributed onto the host file system

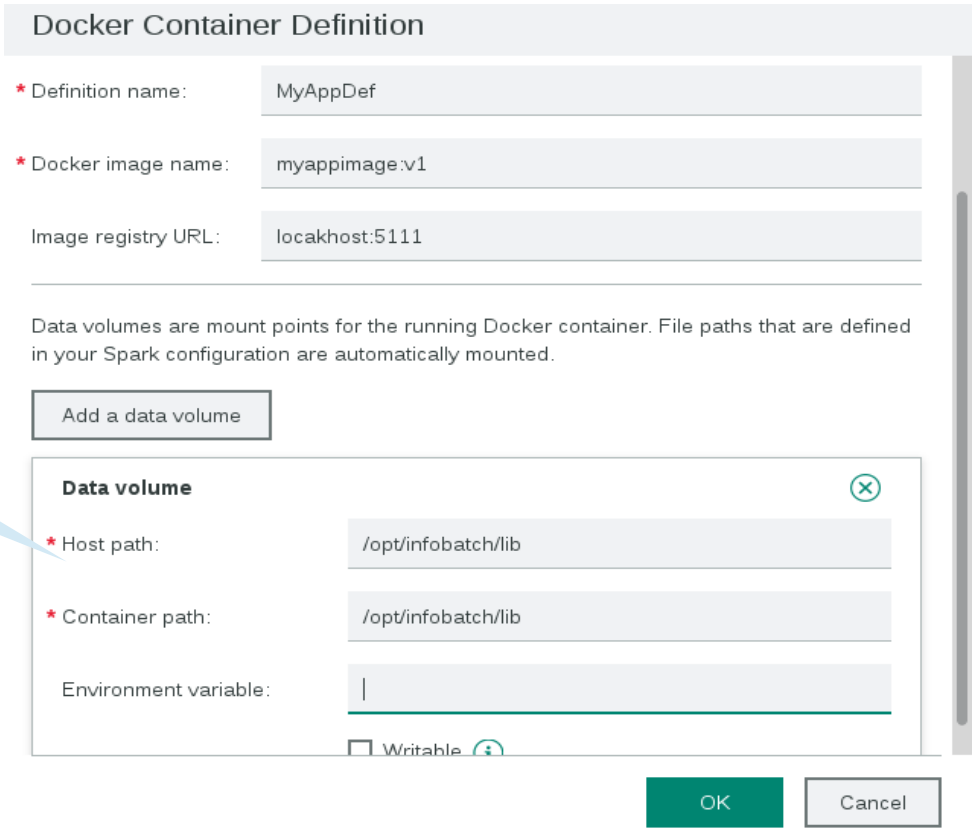
Regular Spark workload



Running in containers



Creating a container definition for an application image



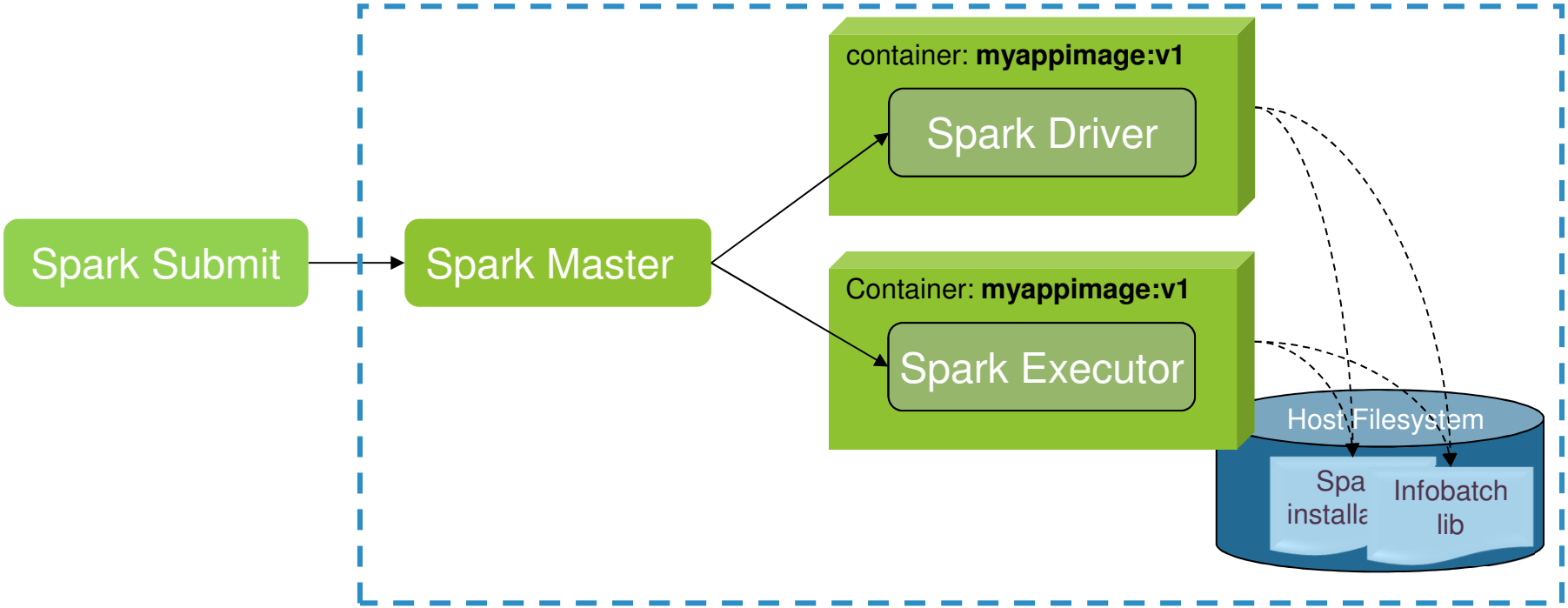
Extra dependency from host file system

Submitting workload with the container definition

```
spark-submit  --class<main-class> --master<master-url> --deploy-mode  cluster \
--conf spark.ego.driver.docker.definition= MyAppDef   \
--conf spark.ego.executor.docker.definition= MyAppDef \
<application-jar> \
[application-arguments]
```

Cluster Mode:
Define container specifications for the drivers and executors

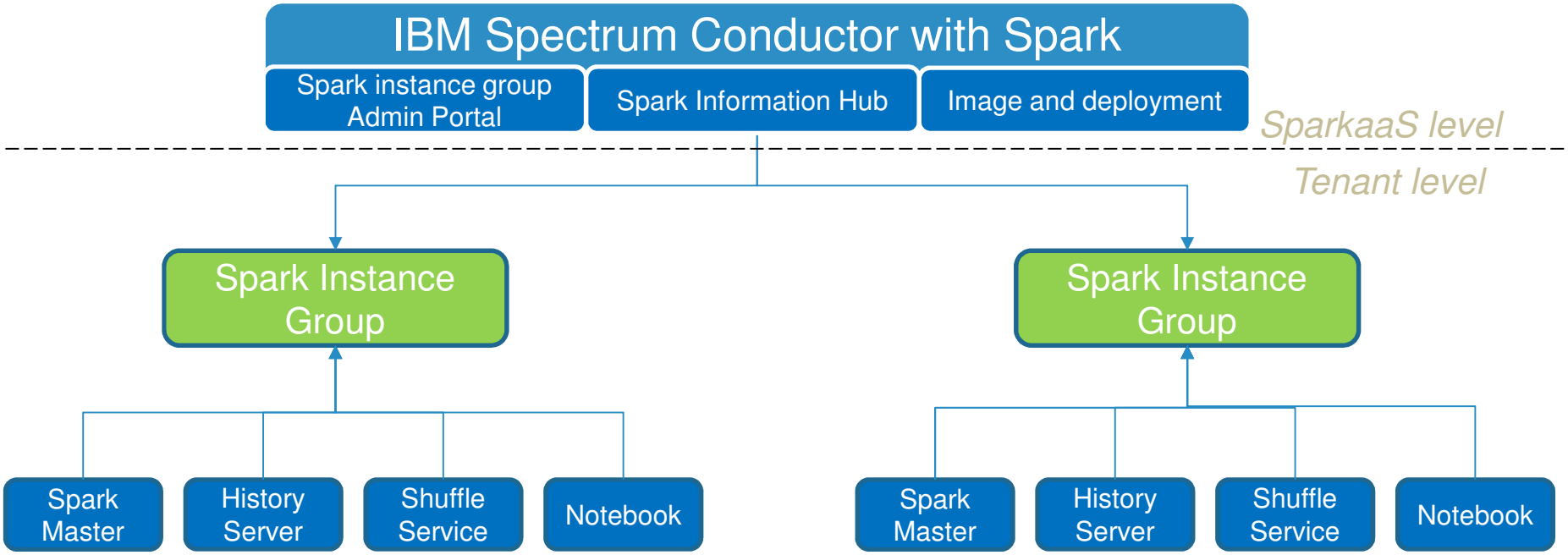
Running in containers



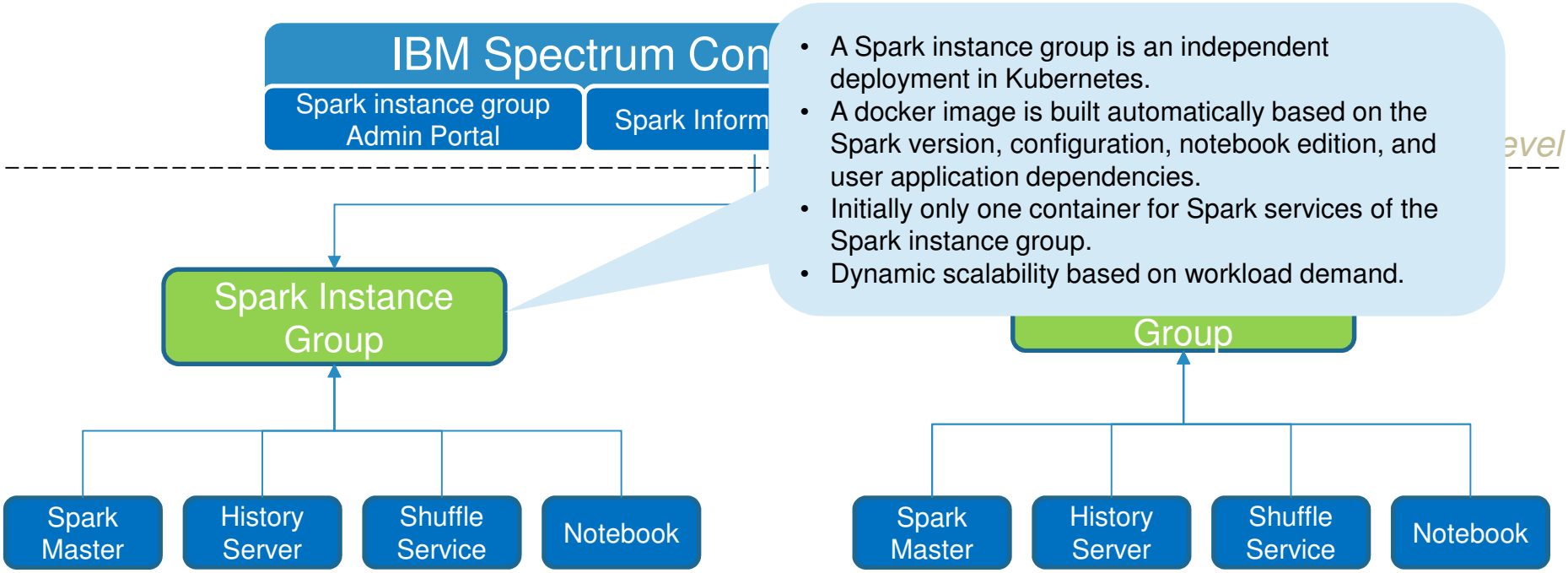
Spark Instance on Kubernetes

- Increase resource utilization
  - Share nodes between Spark and surrounding ecosystem
- Isolation between tenants and apply resource enforcement
  - Each tenant gets a dedicated Spark working instance
  - Tenant price plan can directly map to its resource quota
- Simplify deployment and roll out

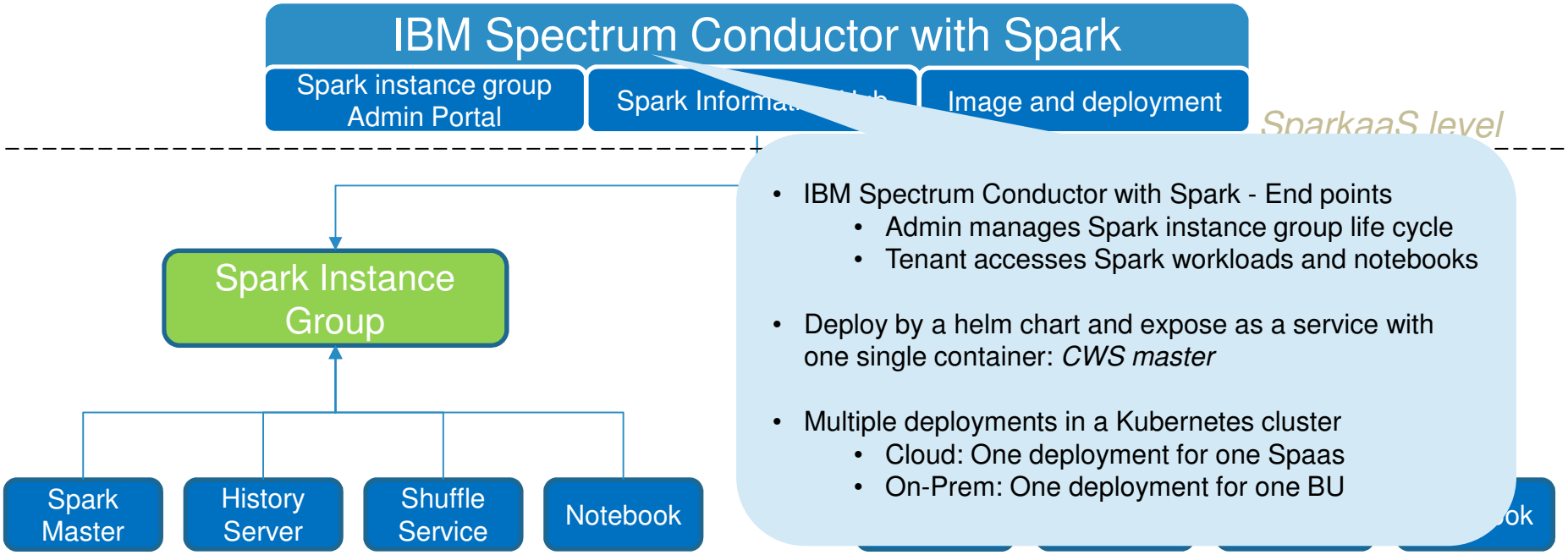
Architecture



Architecture



Architecture



Creating a master container

Kubernetes

*helm install conductor-spark --name **spaas4bu1** --namespace **ns4bu1***

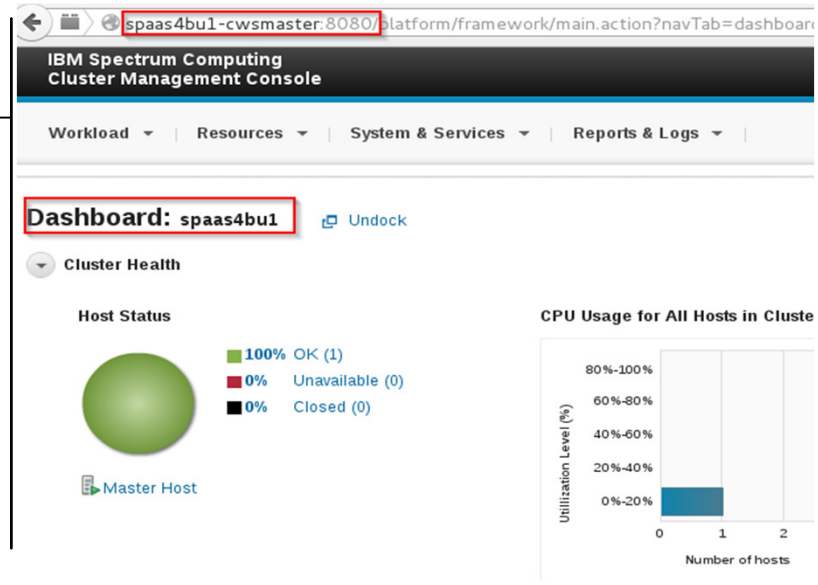
Creating a master container

Kubernetes

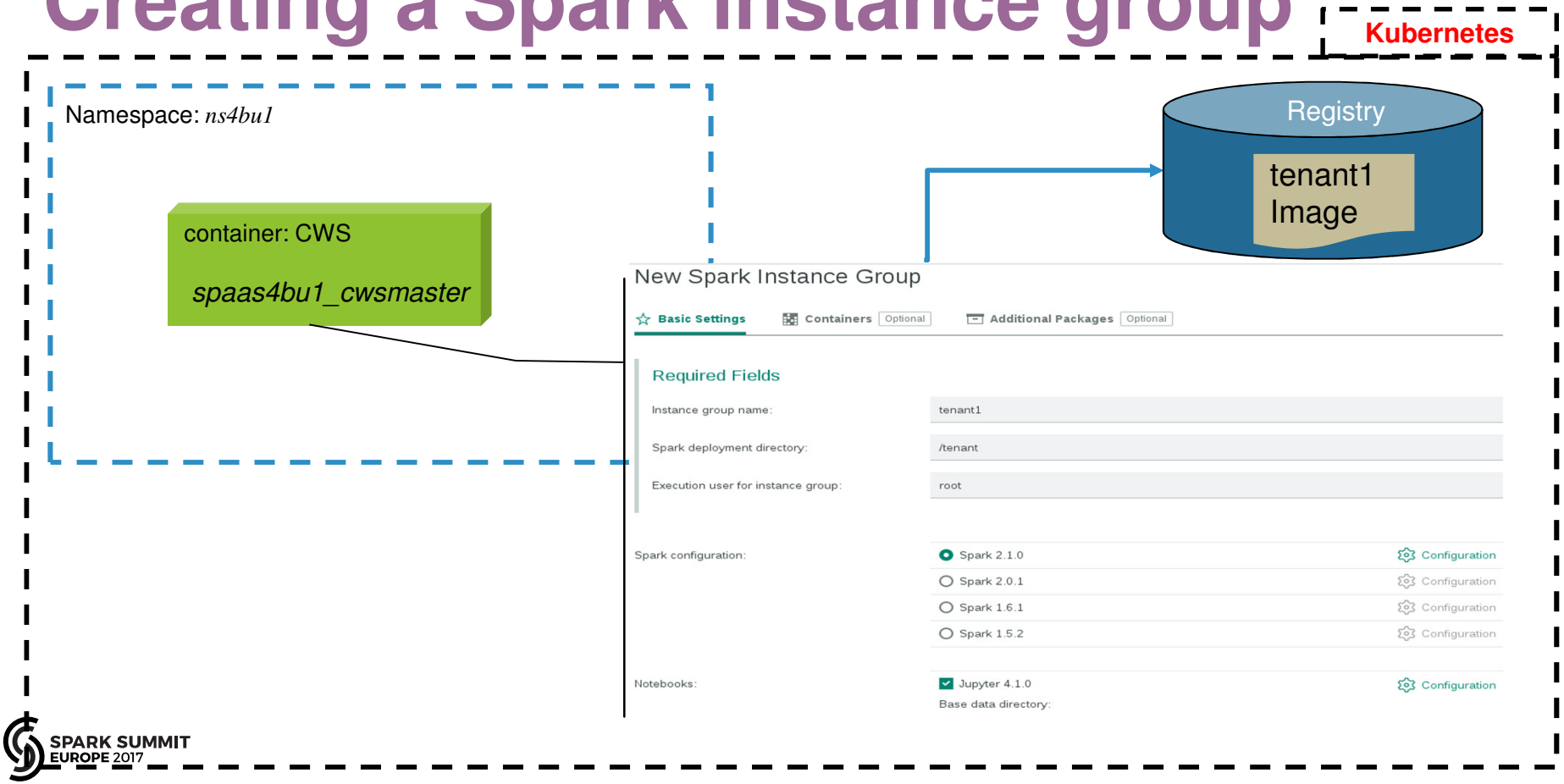
Namespace: *ns4bu1*

container: CWS

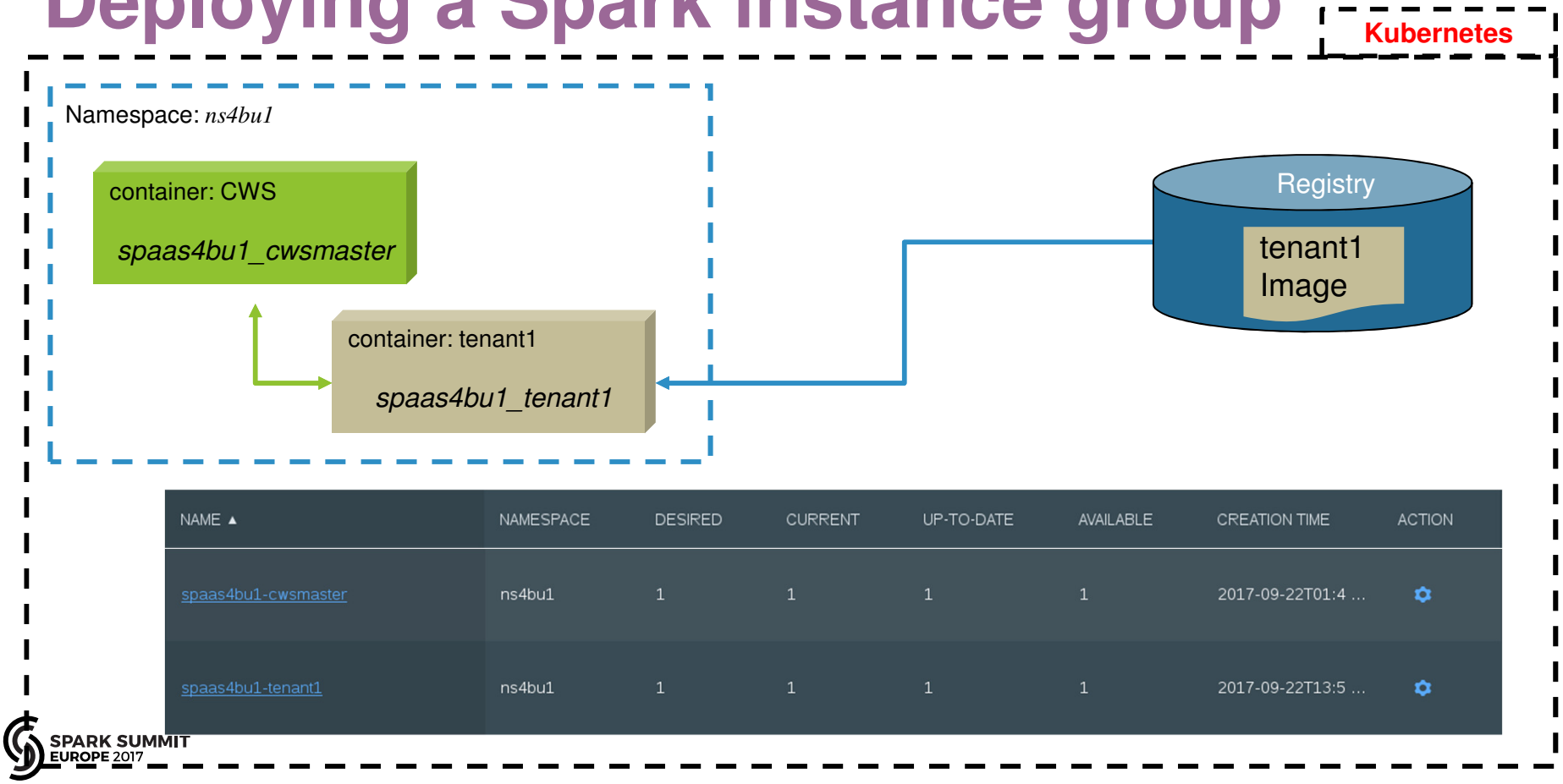
spaas4bu1_cwsmaster



Creating a Spark instance group

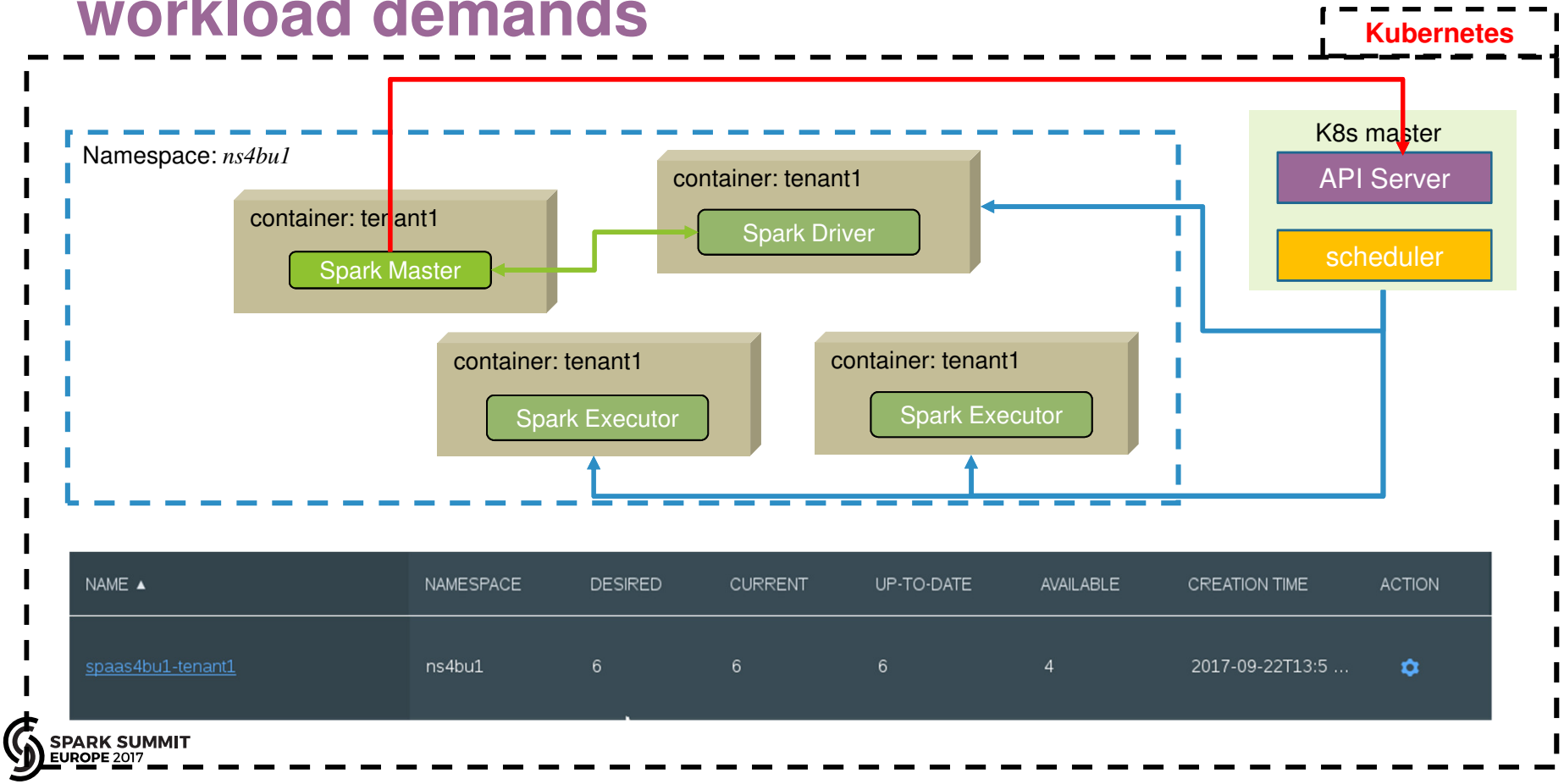


Deploying a Spark instance group



NAME ▲	NAMESPACE	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	CREATION TIME	ACTION
spaas4bu1-cwsmaster	ns4bu1	1	1	1	1	2017-09-22T01:4 ...	
spaas4bu1-tenant1	ns4bu1	1	1	1	1	2017-09-22T13:5 ...	

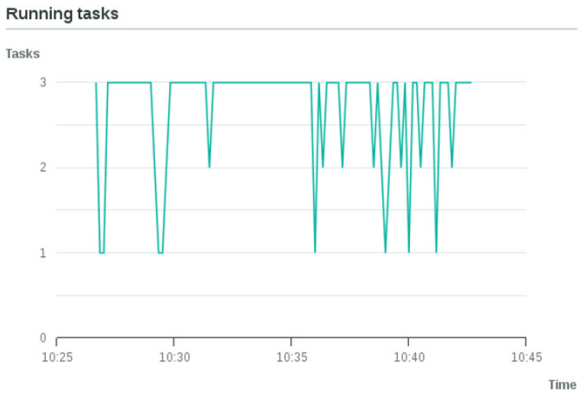
Scaling the Spark instance group based on workload demands

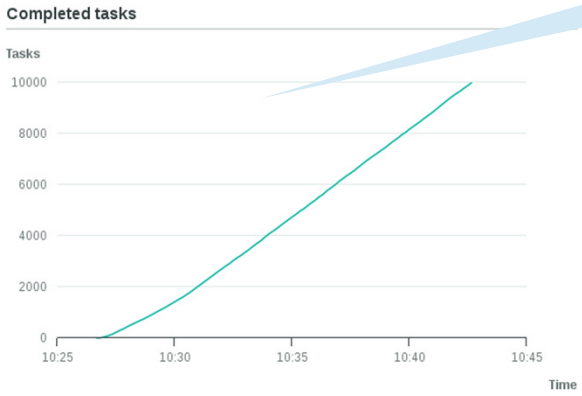


NAME ▲	NAMESPACE	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	CREATION TIME	ACTION
spaas4bu1-tenant1	ns4bu1	6	6	6	4	2017-09-22T13:5 ...	

Performance

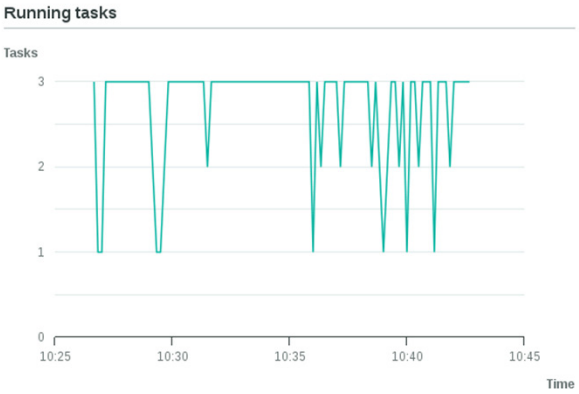
- Without Dynamic Scaling

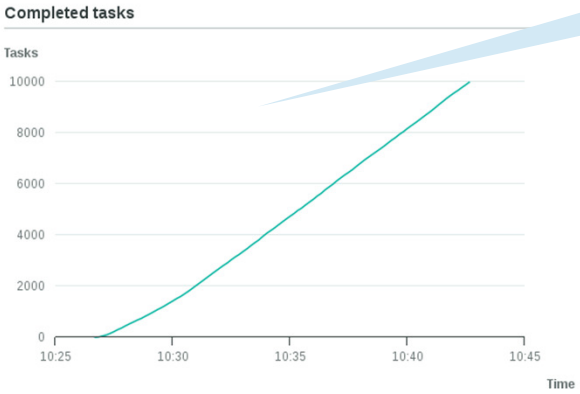




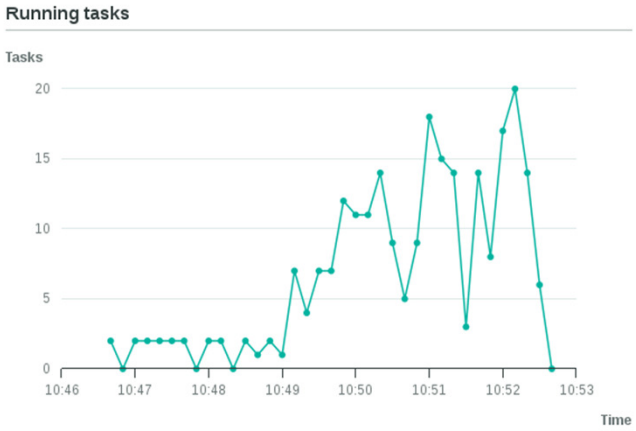
Performance

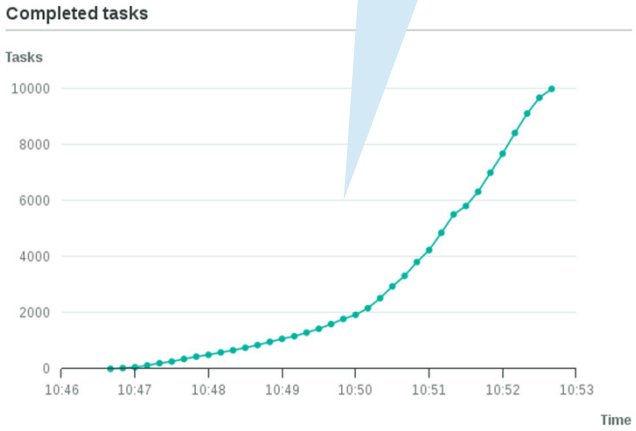
- Without Dynamic Scaling



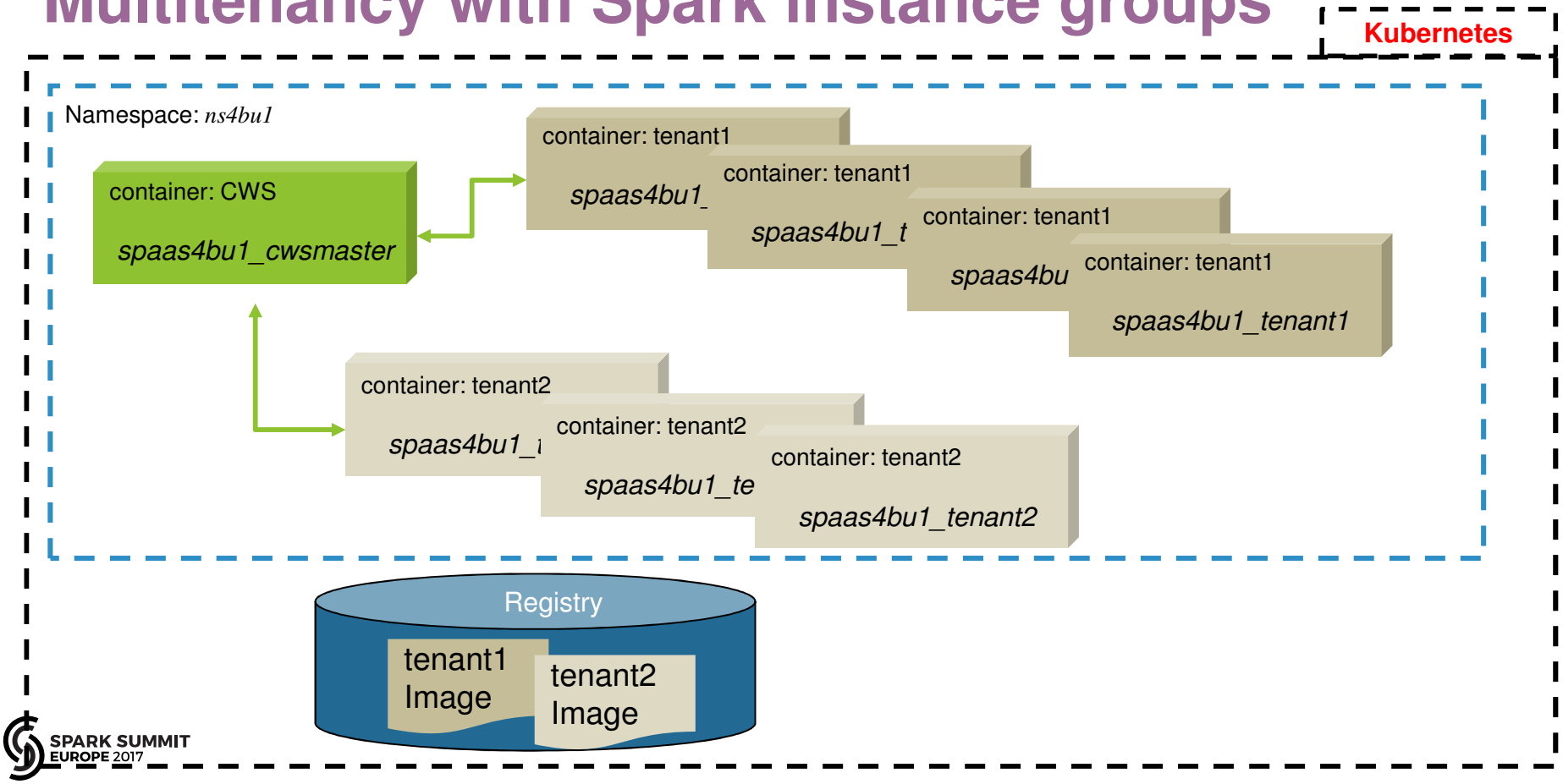


- With Dynamic Scaling





Multitenancy with Spark instance groups

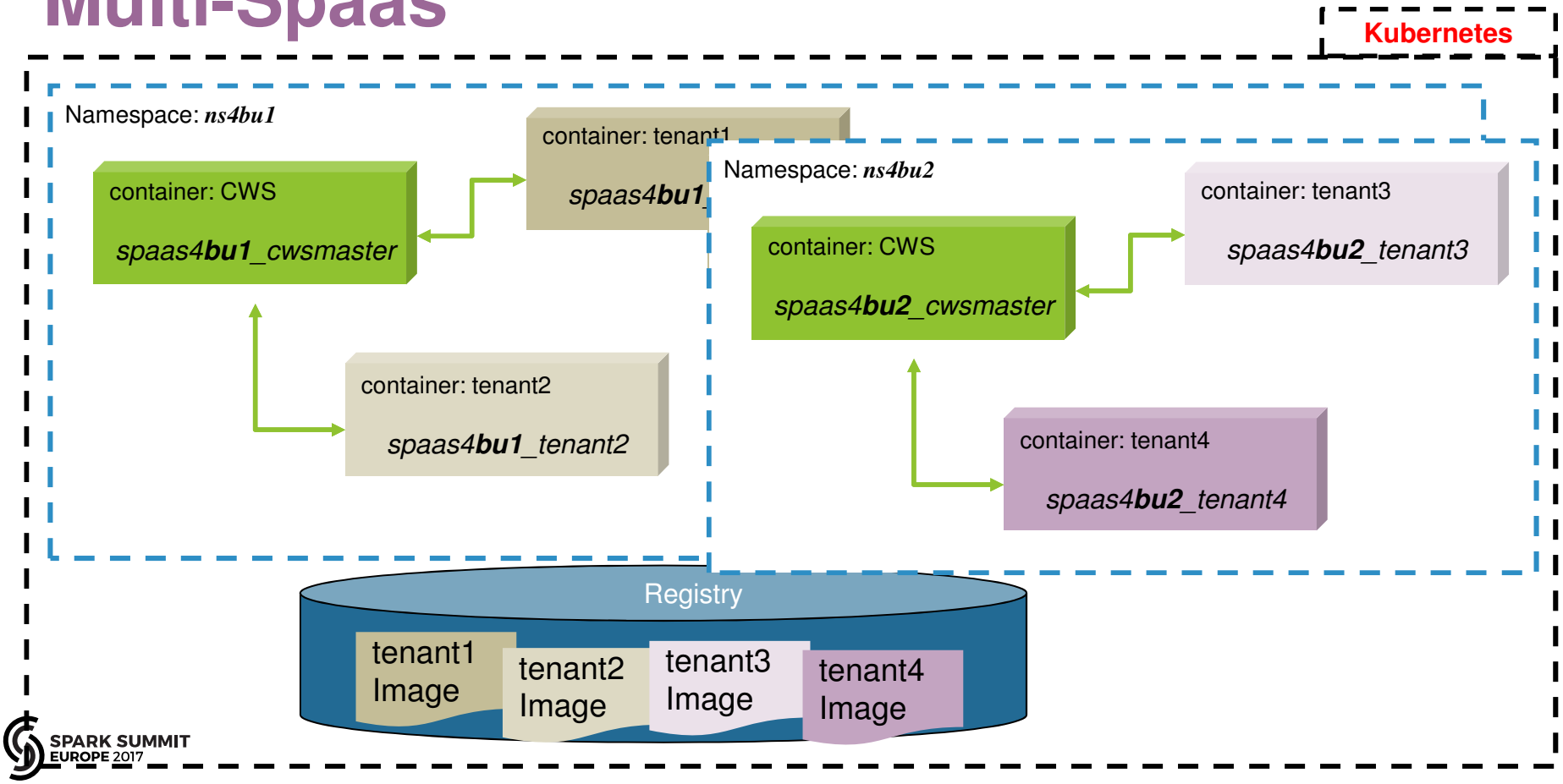


Multi-Spaas

Kubernetes

*helm install conductor-spark --name spaas4**bu1** --namespace ns4**bu1***
*helm install conductor-spark --name spaas4**bu2** --namespace ns4**bu2***

Multi-Spaas


Kubernetes
Namespace: ns4bu1
container: tenant1
spaas4bu1_
container: CWS
spaas4bu1_cwsmaster
container: tenant2
spaas4bu1_tenant2
Namespace: ns4bu2
container: CWS
spaas4bu2_cwsmaster
container: tenant3
spaas4bu2_tenant3
container: tenant4
spaas4bu2_tenant4
Registry
tenant1 Image
tenant2 Image
tenant3 Image
tenant4 Image

Survey: Spark on Kubernetes

	SPARK-18278	Standalone	IBM Spectrum Conductor with Spark on Kubernetes
Dynamic allocation on demand	Yes	Static	Yes
K8s interaction granularity	Job level	Instance level – static	Instance level – dynamic
Deployment Automation • Simple deploy by helm charts	No	Yes	Yes
Spark instance per tenant • Multi-job/workflow/user • Image with user applications • Security	No	limited	Yes

Future work

- Integration with Kubernetes batch workload scheduler
  - Kube-arbitrator (https://github.com/kubernetes-incubator/kube-arbitrator)
- Performance comparation with other Spark on Kubernetes solutions

SPARK SUMMIT

Thank You

www.ibm.com/spectrum-conductor

hma@ca.ibm.com