

Spark on Kubernetes (for Data Science)

Jordan Volz

Senior Data Scientist @ Dataiku



About This Talk

About Me

- Senior Data Scientist @ Dataiku
- Previously ... Software Engineer, Technology Consultant, Sales Engineer, Systems Engineer
- Particular interest in:
 - Big Data
 - Data Science
 - Production Systems
- Hopefully smarter than average in:
 - Spark,
 - Kubernetes, and/or
 - Spark on Kubernetes

About You

- Tired after a long conference
- Curious about, but not assumed to be an expert in:
 - Spark,
 - Kubernetes, and/or
 - Spark on Kubernetes
- Hopefully interested in Data Science
- Not offended by hastily constructed slides with little flair



Quick Recap:

Spark ~~on Kubernetes~~
(for Data Science)

~60 Second Overview of Apache Spark

- Open-source distributed data processing framework
- Bridge to “Big Data”
- Flexible programming paradigm
- Broad integrations
- Wide Adoption



Things Data Scientists **Do** Like About Apache Spark

- Data Processing on large data sets
- Python/R APIs
- Higher-level APIs for non-experts
- Parallel processing can be nice for:
 - Hyperparameter Tuning
 - Scoring large datasets
 - Random distributed frameworks of non-distributed frameworks
- Built-in Distributed Algos (?)

Things Data Scientists **Do Not** Like About Apache Spark

- Managing a Hadoop Cluster
- Libraries/Dependencies
- Performance Tuning/Debugging
- Using GPUs
- UI

Spark: What We've Learned

- Spark can be a useful tool for an advanced data scientist, but ...
- It's very likely that you should **not** be using it for *everything*
- Complexity of System and Ease of Use have prevented wider standardization and local python/R is still very much the standard for small/medium data
- We can do better!

Quick Recap:

~~Spark on~~ **Kubernetes**
(for Data Science)

~60 Second Overview of Kubernetes

- Open-source system for deploying + managing containerized applications
- Crux of many “cloud-native” architectures
- Available from your cloud vendor of choice
- Broad integrations
- Wide Adoption



Things Data Scientists **Do** Like About Kubernetes

- Containers are like magic for libraries + dependencies
- Someone else is likely managing it (it's doing more than just DS)
- Compute on demand!
- Flexible resource profiles (i.e. GPUs), relatively easy autoscaling
- A great place to deploy models

Things Data Scientists **Do Not** Like About Kubernetes

- Building Docker Images
- Troubleshooting when things don't work
- Many DS tools for k8s are more devOps-oriented (or assume you are)
- Distributed tooling is still very experimental
- Half-arsed integrations

Kubernetes: What We've Learned

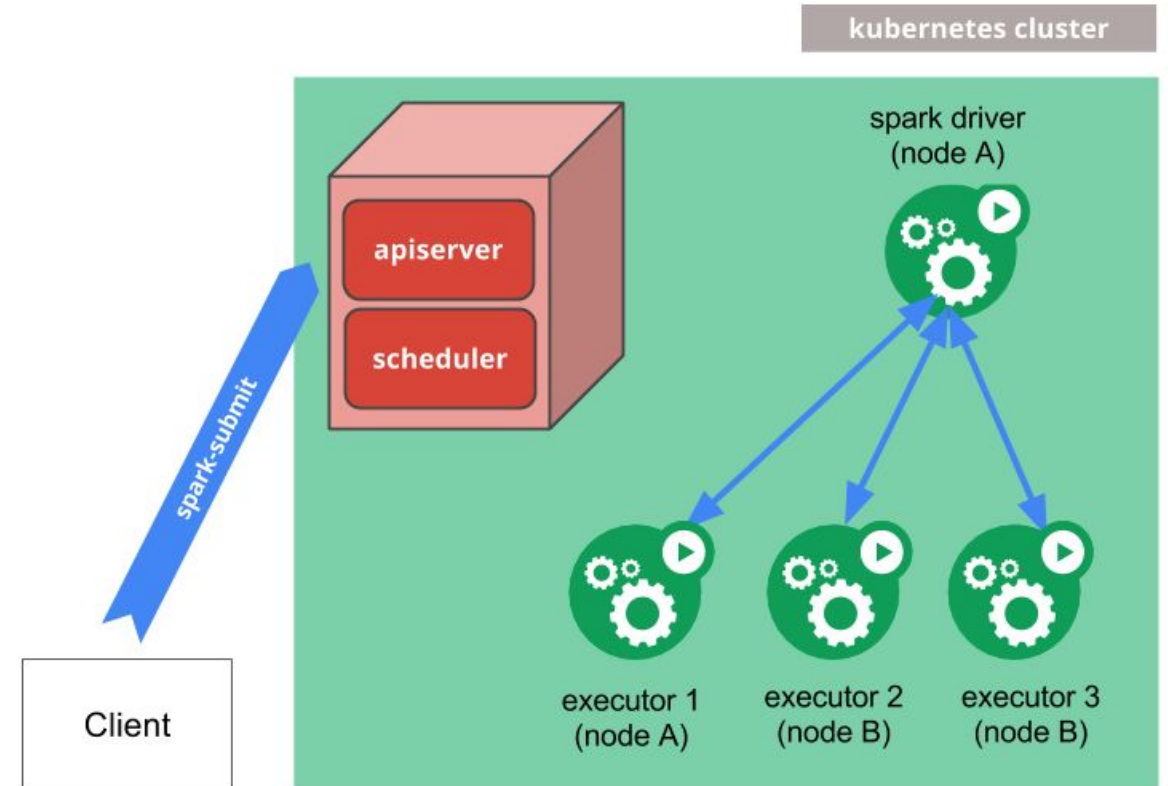
- Kubernetes can be useful for several different types of data science workloads, but ...
- It's pretty complicated!
- Luckily you probably won't have to maintain the system.
- Interesting frameworks are being developed for k8s, but it's still early for many of them.
- Make things as easy as possible to ensure adoption from data scientists.

Quick Intro:

Spark on Kubernetes

Spark on k8s: The Gist

- Yarn \Rightarrow Kubernetes
- Executors and Driver in k8s pods
- k8s handles resource scheduling + mgmt
- Spark works as normal (*more or less*)
- Many standard k8s-specific features are available, i.e. namespaces, RBAC, etc.



A Quick Timeline

- < **Spark 2.3** - Not part of mainstream Spark. Limited use.
- Q1/2018: **Spark 2.3** - Spark on k8s brought into main project.
Experimental with some significant limitations:
 - Java/Scala Support only
 - Cluster mode only
 - Not data science friendly :(
- Q4/2018: **Spark 2.4** - Now data science friendly!
 - pyspark/sparkR support!
 - Client mode supported! (i.e. notebooks!)
 - Base docker images for pyspark/sparkR
- 2019: No Spark Release???
- 2020: **Spark 3.0**, new stuff for Spark on K8s.

What Data Scientists **Should** Like about Spark on K8s

- Specify container with Spark job. Adios to lib/deps nightmares?
- “Just works” on the cloud. (Someone else is managing the k8s cluster.)
- Get resources when you need them, like lots of compute, or GPUs.
- The spark you already wrote still works. The spark you know how to write is still good.
- Easy to get started

What Data Scientists **Should Not** Like about Spark on K8s

- May need to build lots of docker containers
- Can be very buggy at times
- Most all spark tuning info is for Spark on Yarn. May have to think more intelligently about optimizing workflows.
- Lacking some key features of Spark on Yarn:
 - No External Shuffle Service,
 - Can't use files from client
 - No Kerberos, etc.
- May not be super fun trying to connect to exotic data sources.
- External Metastore???

Getting Started

1. Get access to k8s cluster. BYO or vendor-managed.
2. Download spark binary onto client host.
3. Build base spark image using the provided `docker-image-tool.sh`. Make customizations to build as needed
4. Push image to registry
5. Run your spark job/spin up spark shell

Configurations to Know

- **spark.master** = `k8s://http(s)://<k8s-apiserver-host>:<k8s-apiserver-port>` : location of k8s cluster
- **spark.kubernetes.container.image**: Image to use for executor pods.
(Note, can do `spark.kubernetes.<driver/executor>.container.image` separately if they need to be different)
- **spark.kubernetes.namespace** : Namespace to use, if applicable
- **spark.kubernetes.pyspark.pythonVersion**: 2 or 3
- **spark.executor.instances**, **spark.executor.cores**, **spark.executor.memory**: as normal
- **spark.driver.host** + **spark.driver.port**: if running in client mode, can be useful
- **spark.kubernetes.container.image.pullPolicy**: behavior of image pulls for spark executors/driver (default = `IfNotPresent`)

Some Warnings

- When running in client mode, your k8s cluster must be able to communicate back to the host the driver is on. In many cloud setups, this may require explicitly setting `spark.driver.host/port`.
- There is no distributed cache mechanism, so we can't simply do something like `spark-submit my_spark_job.py`
In particular, dependencies must be local to the k8s driver (client) or k8s pod (cluster), or reachable from there.
- The default spark image runs as root. You may want to change this.
- After a job finishes, executor pods are immediately cleaned up. Driver pods will stay in “completed” status until gc'ed or manually purged. I.E. they don't live forever.
- Expect Bugs.

Some Tips

- Access logs via: `kubectl logs <driver-pod-name> -n=<namespace>`. The Spark Web UI is available on the driver host on port 4040. `kubectl describe pod <driver-pod-name>` may be useful when troubleshooting as well
- Security: Use Namespaces + RBAC to isolate security groups in your cluster and limit resources (well, not you, but the k8s admin). Use `spark.kubernetes.authenticate.serviceAccountName` to specify the account to use.
- If using private repos, you'll likely need to set `spark.kubernetes.container.image.pullSecrets`
- `spark.kubernetes.container.image.pullPolicy=Always` is good for debugging image issues (default = `IfNotPresent`, which can be frustrating)
- You can mount k8s volumes into your executors.
`spark.kubernetes.<driver/executor>.volumes.<VolumeType>.<VolumeName>.mount.path=
 <mount path>`

Potentially Exciting Future Work

- External Shuffle Service + Dynamic Allocation
- Submission of dependencies from driver host
- Better Authentication Mechanisms (Kerberos?)
- Easier custom image building
- Pod Templates

A Very Very Quick Demonstration

Spark on Kubernetes: What We've Learned

- Spark on Kubernetes offers to make some major headaches for DS much easier ...
- But it's still pretty new. YMMV.
- Spark 3.0 should make it a less of a hassle to work with and fill out some current gaps.
- Despite current limitations, your DS platform of choice may integrate with it, making it fairly seamless to transition to w/o sacrificing functionality in your existing workflows.

Rate today's session

Cyberconflict: A new era of war, sabotage, and fear

See passes & pricing

David Sanger (The New York Times)
9:55am-10:10am Wednesday, March 27, 2019
Location: Ballroom
Secondary topics: Security and Privacy

Rate This Session

We're living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you're often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we're uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David E. Sanger is the national security correspondent for the *New York Times* as well as a national security and political contributor for CNN and a frequent guest on *CBS This Morning*, *Face the Nation*, and many PBS shows.

Keynotes

Session page on conference website

Attending Notes Remove

Cyberconflict: A new era of war, sabotage, and fear

9:55 AM - 10:10 AM, Wed, Mar 27, 2019

Speakers

David Sanger
National Security Correspondent
The New York Times

Ballroom

Keynotes

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

SESSION EVALUATION

O'Reilly Events App

The End!
Questions???



data
iku

Jordan Volz

[linkedin.com/in/jordanvolz/](https://www.linkedin.com/in/jordanvolz/)