

# Spark Job Submission Methods



Uploaded image

summarise the spark job submit methods

The image outlines **four methods** to submit a Spark job on Kubernetes (K8s), summarized as follows:

## ✓ Method 1: CLI-based (spark-submit)

- **Tool:** `spark-submit`
- **How it works:**
  - User generates and executes the command using the CLI.
  - `spark-submit` interacts with the Kubernetes API to create a **driver pod**.
  - The driver pod then requests creation of **executor pods**.
- **Typical Use Case:** Manual job submission, testing, ad-hoc analysis.

## ✓ Method 2: Airflow DAG with SparkSubmitOperator

- **Tool:** Apache Airflow's `SparkSubmitOperator`
- **How it works:**
  - Airflow DAG task uses the operator to trigger a `spark-submit` job.
  - Follows the same path as Method 1 (driver → executors).
- **Typical Use Case:** Automated and scheduled Spark jobs in production pipelines.

## ✓ Method 3: Direct Kubernetes API or kubectl

- **Tool:** `kubectl` or Kubernetes client SDK (Go, Java, Python)
- **How it works:**
  - A driver pod is manually created using Kubernetes tools or client APIs.
  - The driver pod then requests executor pods.
- **Typical Use Case:** Custom integrations or direct Kubernetes-based workflows.

## ✓ Method 4: Spark Operator with CRD (Recommended for Production)

- **Tool:** Spark Operator + CustomResourceDefinition (CRD) `SparkApplication`
- **How it works:**

- User creates a `SparkApplication` CRD object using `kubectl` or client SDK.
- The Spark Operator watches the CRD and creates the driver and executor pods.
- **Typical Use Case:** Declarative Spark job management, GitOps, production deployments.

## Common Flow in All Methods

- Driver pod is created first.
- Driver communicates with K8s API server to request executor pods.
- Executors are created and managed by K8s scheduler.

Let me know if you'd like a tabular comparison (e.g., pros/cons, use cases, automation level).



is there any new method ?

Yes — in addition to the **four traditional methods** shown in your diagram, **newer or alternative methods** have emerged as Spark-on-Kubernetes matures and ecosystem tools evolve. Here are **new or emerging methods** worth knowing:

## ✓ Method 5: Argo Workflows + SparkSubmit or Spark CRD

- **Description:** Use [Argo Workflows](#) to orchestrate Spark jobs declaratively.
- **How:**
  - Each Argo step can either:
    - Run `spark-submit` (like Method 1), **or**
    - Submit a `SparkApplication` CRD (like Method 4).
- **Benefits:**
  - Fully declarative, GitOps-friendly.
  - Native K8s CRD support.
  - Integrates easily with other containerized ML/ETL steps.

## ✓ Method 6: Serverless Spark on K8s via Ray or Volcano

- **Tools:** [Ray](#), [Volcano](#), or similar batch/ML job schedulers.
- **How:**
  - These frameworks offer **serverless Spark job execution**, handling resource allocation and scheduling.
  - You submit a Spark job spec; the backend takes care of launching driver + executors.
- **Use Case:**

- ML pipelines or mixed workloads (Spark + PyTorch/TensorFlow).
- Better auto-scaling and multi-tenant job isolation.

## ✓ Method 7: Custom REST API Gateway (Enterprise)

- **Description:** Build a lightweight REST API that receives job parameters and:
  - Uses the K8s API to create `SparkApplication` CRDs or
  - Invokes `spark-submit` internally.
- **Tools:** Flask/FastAPI (Python), Spring Boot (Java), or Golang.
- **Benefits:**
  - Easy job submission for non-engineers.
  - Abstracts Spark/K8s complexity.
  - Integrates with CI/CD or approval workflows.

## ✓ Method 8: Data Engineering Platforms (e.g., Databricks, Google Cloud Dataproc on K8s)

- **Cloud Vendors** now offer Spark-on-K8s via managed platforms.
- Example:
  - [Dataproc on GKE](#)
  - [Databricks on Kubernetes](#)
- **You submit via their UI/API**, and they handle the Spark-on-K8s setup under the hood.

### Summary Table:

Method	Description	Tools	Best For
5	Argo + Spark	Argo Workflows	Declarative, GitOps pipelines
6	Serverless Spark	Ray, Volcano	ML & batch workloads
7	Custom REST API	Flask/FastAPI, Spring, Go	Enterprise self-service portals
8	Managed Platforms	Dataproc, Databricks	Simplified SparkOps in cloud

Let me know if you want diagrams or code examples for any of the new methods.