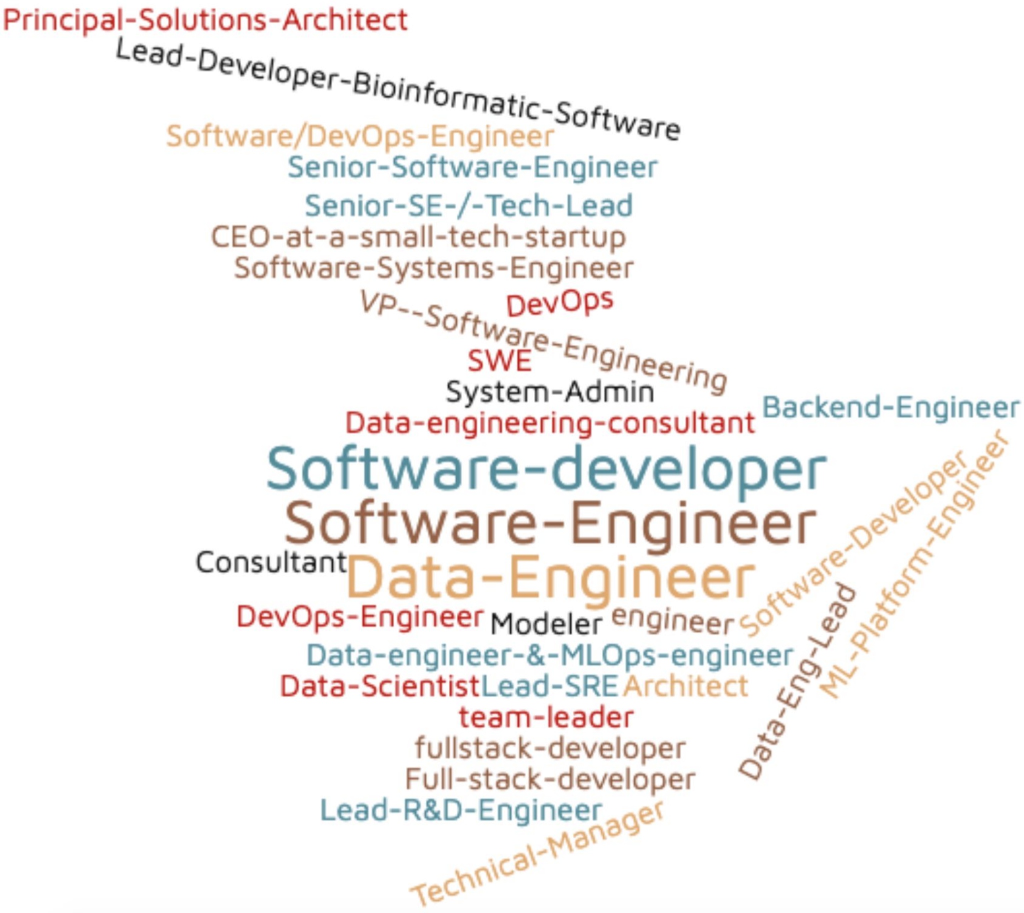
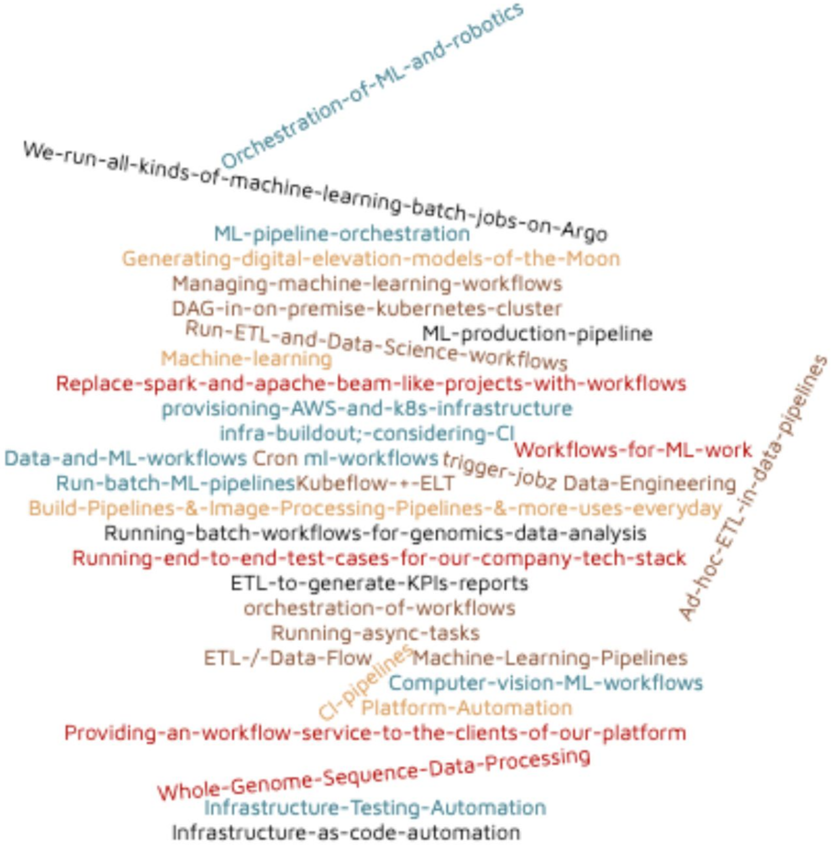
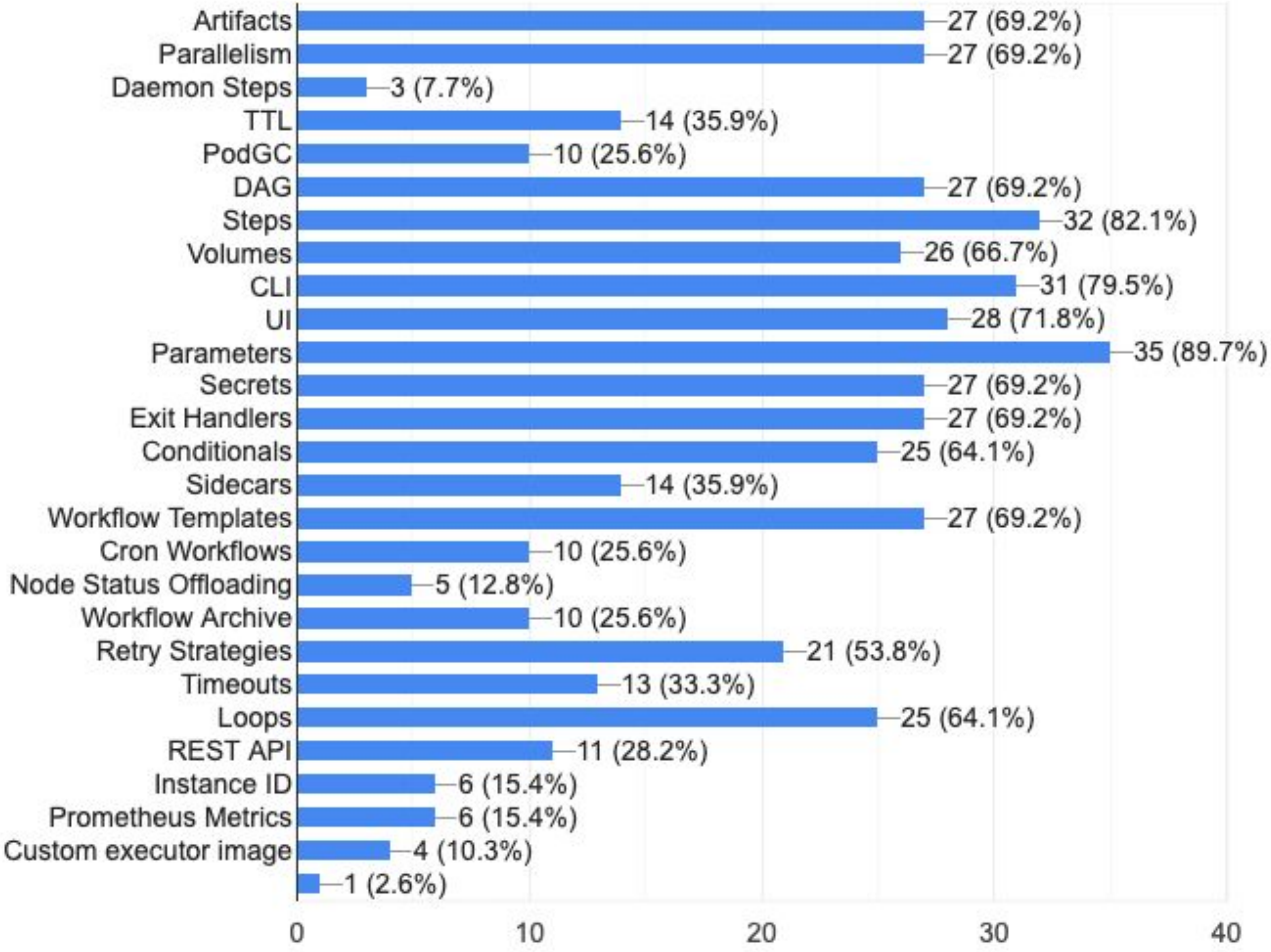
What is your role?



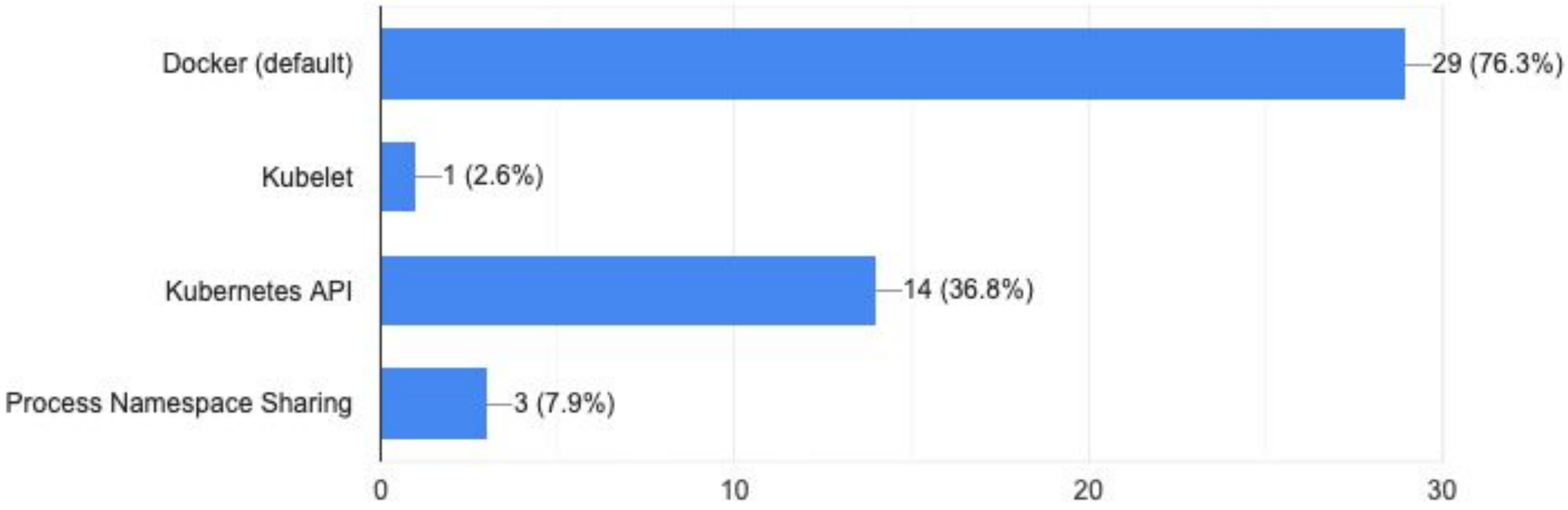
What are your main use cases?



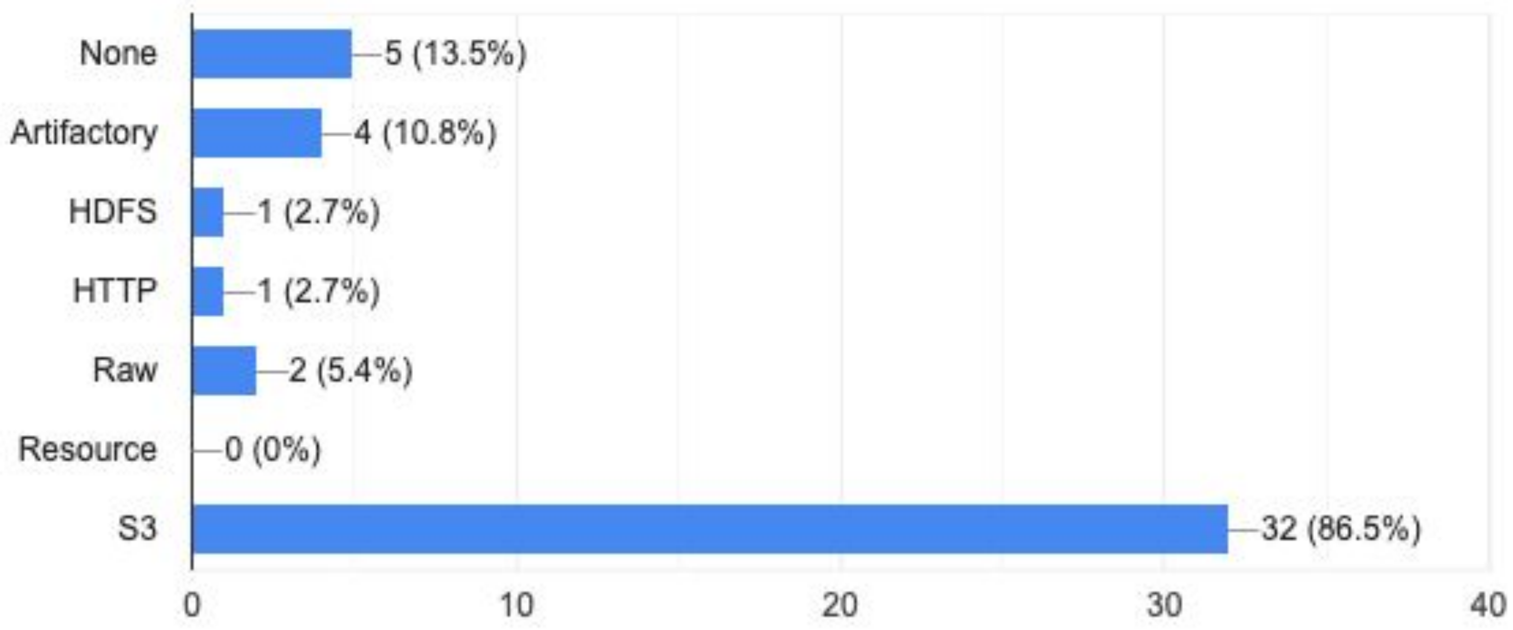
What features are you using?



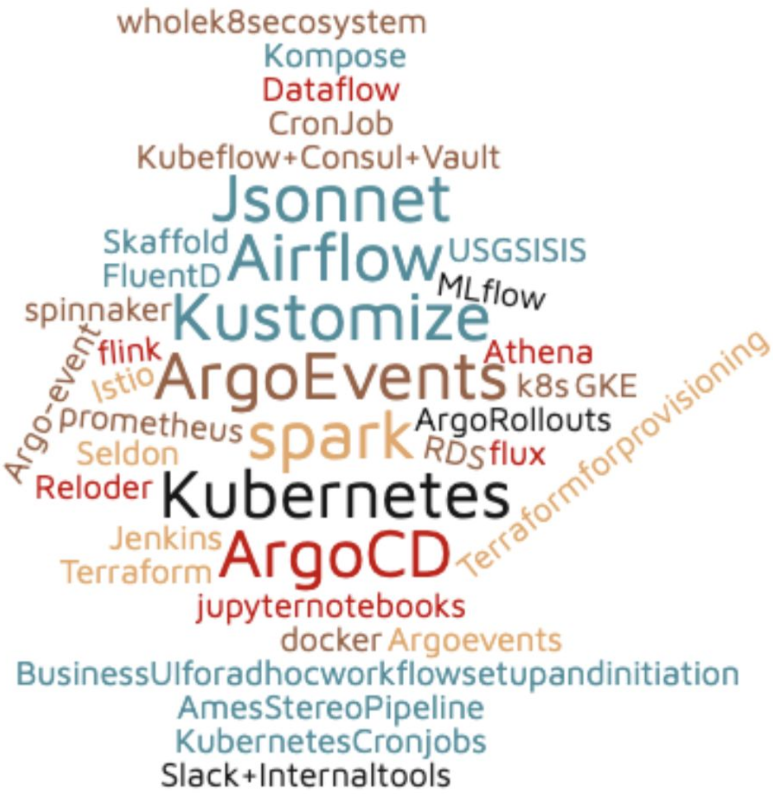
What executors are you using?



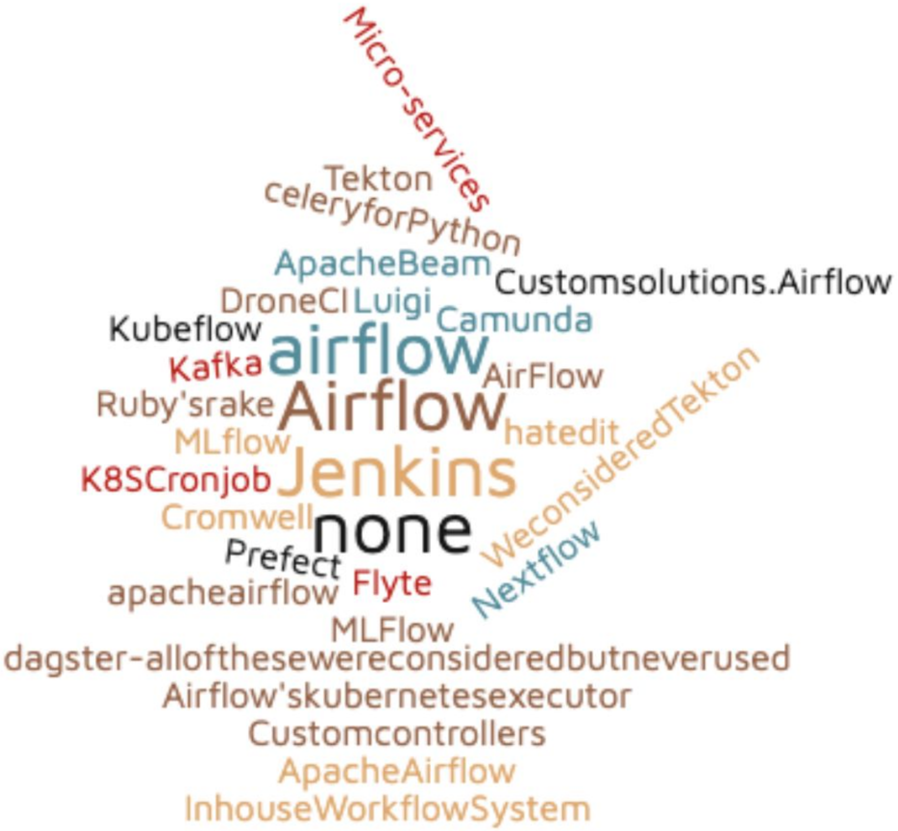
What artifact storage are you using?



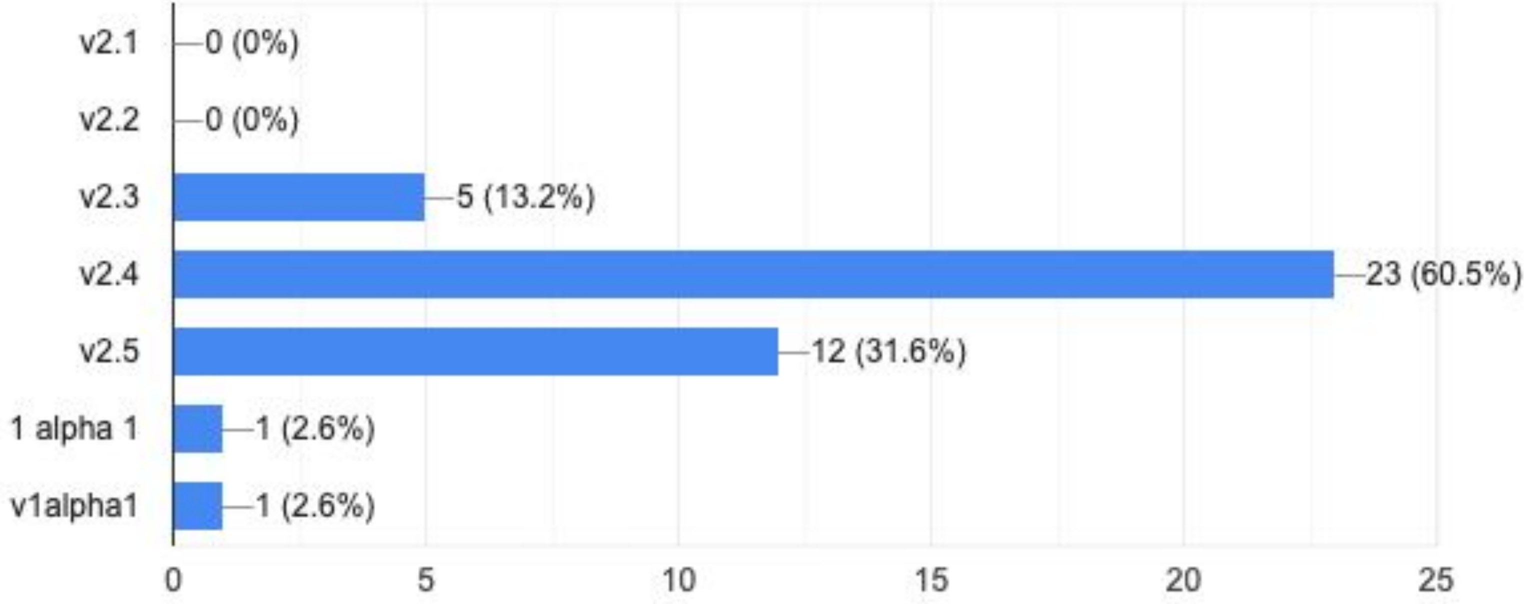
What tools are you using alongside Argo?



What alternative to Argo Workflows have you considered or switched from?



What versions are you using?



What is the one feature we should build in that will make your experience better?

- Non-leaf retry
- I have not yet run into a missing feature that I needed, but https://github.com/argoproj/argo/issues/2185 would be nice.
- Job history UI (pass, fail runtime), like Airflow
- Storage for history and analytics (not k8s' etcd!)
- Map / Reduce approach with mounted volumes to process data in parallel and merge the results into a single file without the need of copying data back to S3.
- json output from argo list (so we can manipulate the list using tools like jq), i.e. list all failed workflow executions
- Turn artifact management as more of a plugin so that we can easily contribute/expand the set of data sources and sinks for argo workflows. Eg: I'm dying to set up authenticated HTTP calls so that my workflows can call an API to access or push workflow data.
- more support for argo events
- How about developing CRD's for Monitoring the Kubernetes cluster I/O, cores and Memory usage .
- better logs, without use kubectl, on pod initializing
- Enhanced UI - Can see all runs for a specific "Pipeline", similar to kubeflow pipelines; Automic HTML form for submitting workflows using existing templates, or maybe executing a Cron workflow run immediately
- Template- / step-level volume claims
- Getting https://github.com/argoproj/argo/pull/1067 merged is at the moment most important item for us.
- A mechanism for exposing step progress - see issue #1658
- input-output pipes between dag tasks/steps for hassle free transfer of data between them. Right now we can do that by writing files to persistent storage but handling file paths by hand adds a lot of clutter. Ex. channel logic in nextflow was a great tool when we were using that.
- Workflow DSL in code instead of YAML. Python or Jsonnet or other.
- another option for template ({{, }}), it has conflict with helm.
- Support for Spark in workflows
- First class support for Python wrapper instead of YAML
- Composable workflows, workflow registry, integration with events
- Accessory file transclusion (#2060)
- Would like to see custom workflow metadata in become more accessible in in the database schema ie easy access to labels and annotations in both workflow and workflow archive tables.
- easy logging (with scripts)
- documentation could be better but was able to get some answer from slack channel
- cost metrics, progress
- Better UI/Better Argo Events UI
- OIDC on the dashboard
- On demand retry from ui/api
- Better library to create workflow.
- More metadata about workflows stored in postgresql archive (ie. custom metrics, cost/utilization, other custom labels/annotations). Looks like a lot of this is already in development, which is great.
- Continue workflow from UI
- add the ability to restart workflow from the dashboard
- Per Step Event Handlers. Exposing logs to event handlers.
- `argo submit --from-workflow-template`