Team 5: Integration

CS5604: Information Storage and Retrieval

Advised by Dr. Edward Fox

Blacksburg, VA 24061

Dec. 06, 2022

Aaron Travasso | Anmol Shukla | Harish Babu | Pallavi Sisodiya | Yuze Li
Subject Matter Expert: Dhanush Dinesh

Agenda

1. Workflow automation
2. Workflow CRUD interface
3. Developer Operations
4. Future work

01

Workflow Automation

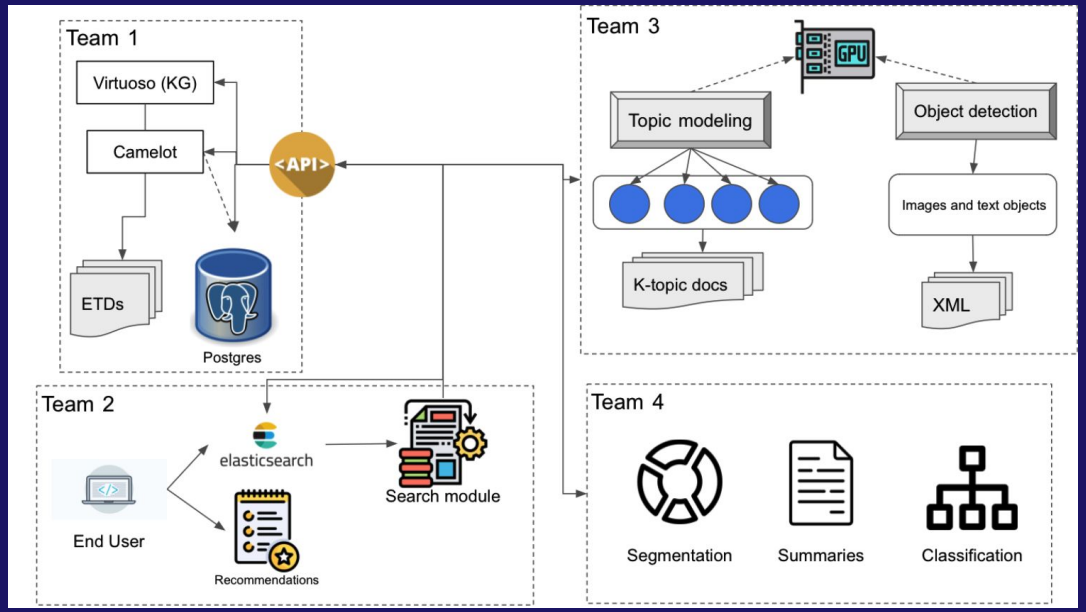
Requirements

Containerization of bunch of services

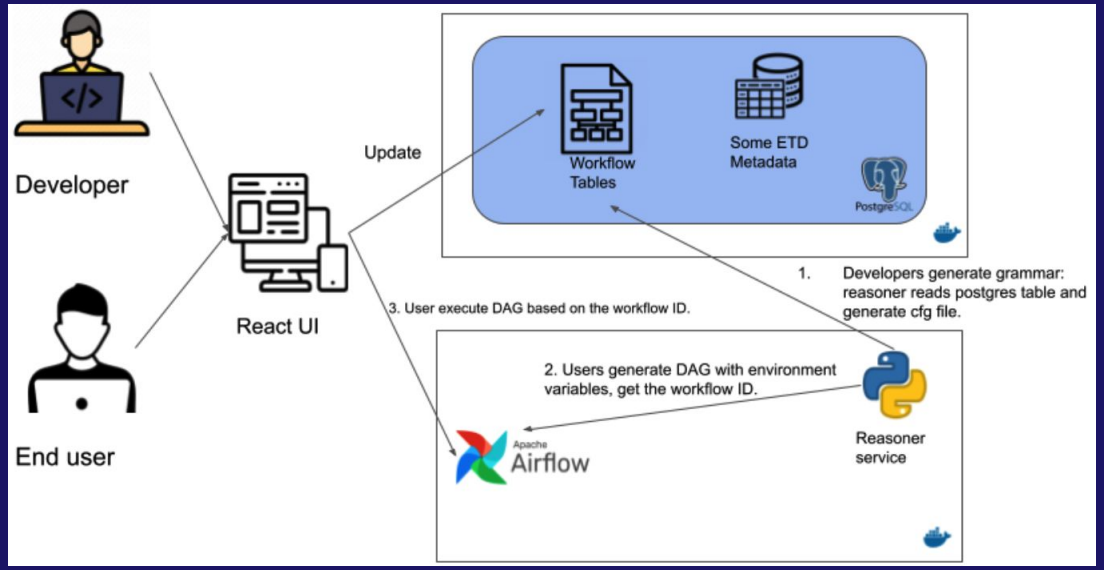
- For developers, hard to maintain dependencies
- For end-users, hard to know what they need

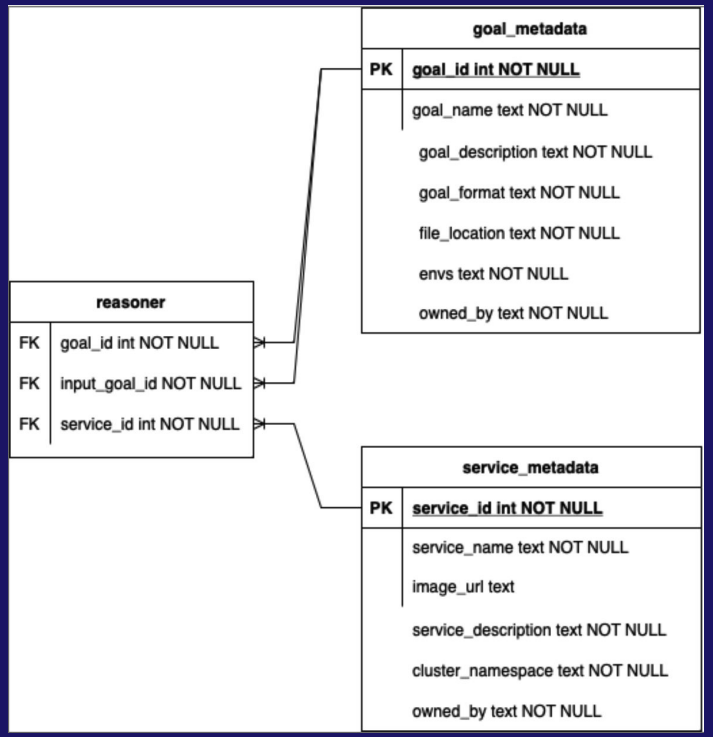
Workflow automation helps end-users by abstracting/hiding complex low-level services.

- For developers, each service in workflow is containerized, easy to maintain
- For end-users, they only see the predefined workflow, don't care about what's inside the black box



Design


Developer
End user
React UI
Update
Workflow Tables
Some ETD Metadata
PostgreSQL
1. Developers generate grammar: reasoner reads postgres table and generate cfg file.
3. User execute DAG based on the workflow ID.
2. Users generate DAG with environment variables, get the workflow ID.
Apache Airflow
Reasoner service


goal_metadata
PK goal_id int NOT NULL
goal_name text NOT NULL
goal_description text NOT NULL
goal_format text NOT NULL
file_location text NOT NULL
envs text NOT NULL
owned_by text NOT NULL
reasoner
FK goal_id int NOT NULL
FK input_goal_id NOT NULL
FK service_id int NOT NULL
service_metadata
PK service_id int NOT NULL
service_name text NOT NULL
image_url text
service_description text NOT NULL
cluster_namespace text NOT NULL
owned_by text NOT NULL

Implementation

- Context-Free Grammar (CFG)
  - Recursively generate patterns
  - Represents goals and services
- Reasoner APIs
  - Generate grammar (CFG file)
  - Generate workflow
    - Required input: <goal_id>
    - Optional input: <environment (ETD_ID)>
    - Output: <workflow_id>
  - Execute workflow
    - Required input: <workflow_id>
    - Optional input: <service_id(s)>
    - Output: <log_url>, <status_key>
  - Check workflow status
    - Required input: <status_key>
    - Output: <status_metadata>
    - G3->S1S2 epsilon

$$S \rightarrow G_1$$

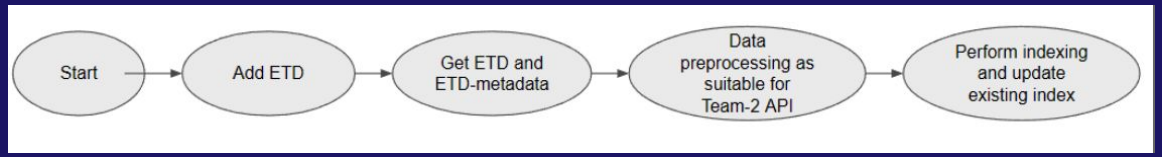
$$G_3 \rightarrow S_2 G_2$$

$$G_2 \rightarrow S_1 G_1$$

$$G_1 \rightarrow \epsilon$$

Implementation

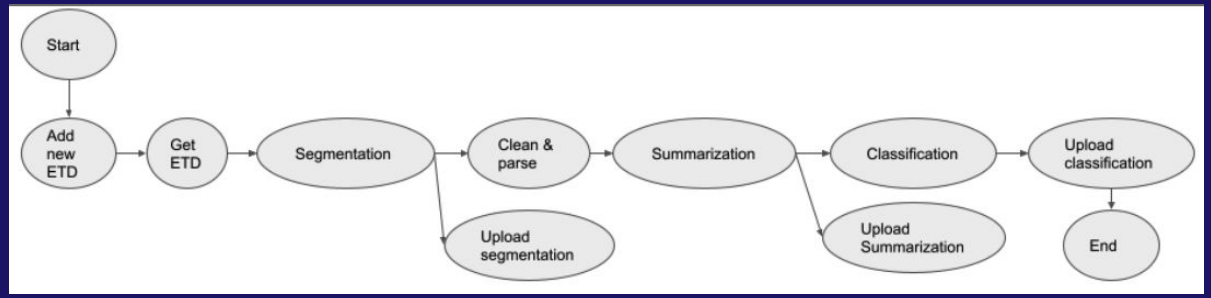
- Workflows implemented
  - Indexing



  - Object detection

Implementation

- Workflows implemented
  - Segmentation, summarization, classification



Demonstration

Team-4's segmentation

- File Team5INTdemo.mp4 - Team 4's Segmentation Demo Video

How do I integrate this?

- Helper code to generate reasoner [6]
  - generateWorkflow()
  - runWorkflow()
  - getWorkflowStatus()
- List of container registries [2]

```javascript
import axios from "axios";
import { baseURL } from "../constants";

const REASONER_BASE_URL = baseURL.team5.reasoner,
  AIRFLOW_BASE_URL = baseURL.team5.airflow;

/**
 * Generates a Workflow run instance with the given environment variables.
 * @param {String|Number} workflowId - The workflow that you want to get generated
 * @param {Object} payload - The environment variables that you will give to your workflow run instance
 * payload structure:
 * {
 *    "env": [
 *        {
 *            "service_id":"fetch_metadata_and_update_index",
 *            "service_env":[
 *                "ETDID=1"
 *            ]
 *        }
 *    ]
 * }
 * @returns Response
 */
export function generateWorkflow(workflowId, payload) {
  let config = {
    method: "post",
    url: `${REASONER_BASE_URL}/generateWorkflow/${workflowId}`,
    headers: {},
  };

  if (payload && typeof payload === "object") {
    config.data = payload;
  } else {
    config.data = {
      env: [],
    };
  }

  return axios(config)
    .then(function (response) {
      if (response.status >= 400) {
        throw new Error(response);
      }

      console.log(response);
      return {
        workflowId: response.data.workflowId[0],
        services: response.data.serviceMetadata[0].map((service) => service.service_name),
```

02

Workflow CRUD Interface

- Close
- Search
- Curate
- Experiment
- Workflow Automation
- Profile
- Settings
- About
- Search Experiments
- Logout

Workflow Automation

Workflow automation definition page

utomation, we need to visualize in terms of services, inputs and outputs. Each service is a unitof computation, which takes some inputs and generates outputs. For that extracts URLs from a text file, the service is a file written in Python, Java, etc. that does this computation. But, we still need to provide an input to this unit the inputs and the outputs come in.

t. Please don't think of goals in the traditional sense here, but as what we can pass as inputs or outputs.

goal_name	goal_description	goal_format	file_location	enironment_variable	owned_by
clean_and_parse_output	CLEAN_PDF	<Directory>	/mnt/data/team5/cleaned_chapters	CLEANED_ENV	INT
classification_output	CLASSIFY_CHAP	<Directory>	/mnt/data/team5/classified_chapters	CLASSIFICATION_ENV	INT
fetch_etd_output	Stores fetched ETD for segmentation, summ,	<Directory>	/mnt/data/team5/segme	ORIGINAL_DATASET	INT

+

Workflow Automation

Workflow automation definition page

When we think of workflow automation, we need to visualize in terms of services, inputs and outputs. Each service is a unitof computation, which takes some inputs and generates outputs. For example, if we have a service that extracts URLs from a text file, the service is a file written in Python, Java, etc. that does this computation. But, we still need to provide an input to this unit ofcomputation. This is where the inputs and the outputs come in.

A goal is an output or an input. Please don't think of goals in the traditional sense here, but as what we can pass as inputs or outputs.

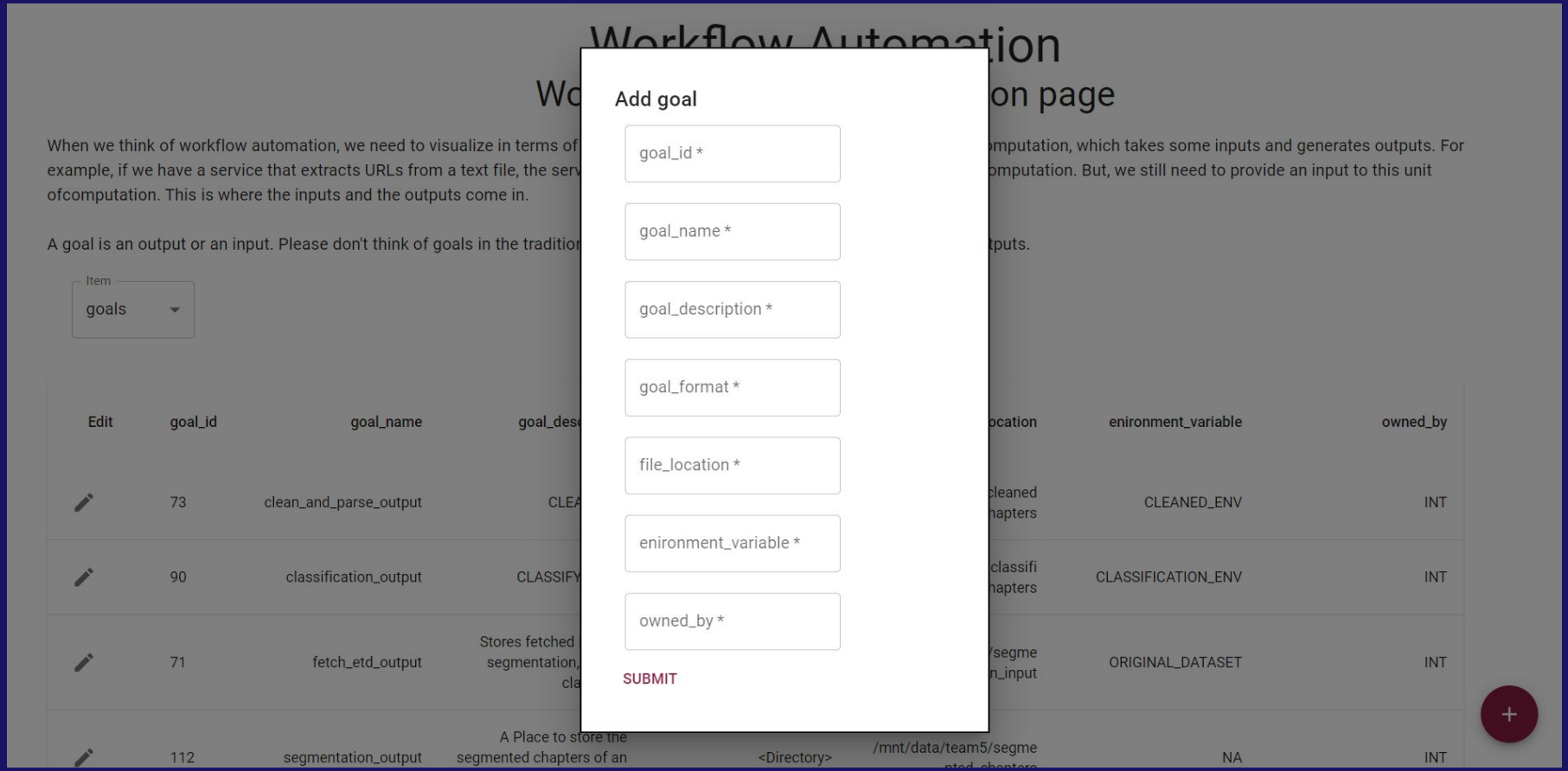
Item

goals

- goals
- services
- reasoner

	l_id	goal_name	goal_description	goal_format	file_location	enironment_variable	owned_by
	73	clean_and_parse_output	CLEAN_PDF	<Directory>	/mnt/data/team5/cleaned_chapters	CLEANED_ENV	INT
	90	classification_output	CLASSIFY_CHAP	<Directory>	/mnt/data/team5/classified_chapters	CLASSIFICATION_ENV	INT
	71	fetch_etd_output	Stores fetched ETD for segmentation, summ, class, etc.	<Directory>	/mnt/data/team5/segmentation_input	ORIGINAL_DATASET	INT
	112	segmentation_output	A Place to store the segmented chapters of an	<Directory>	/mnt/data/team5/segmented_chapters	NA	INT

+



For the complete frontend, please check here [4]

03

Developer Operations

CONTAINERIZATION

TEAM-1

PostgreSQL

Python Jupyter

Virtuoso

TEAM-2

Jupyter Notebook

Elasticsearch

TEAM-3

PyTorch with CUDA

PyTorch with CUDA

TEAM-4

PyTorch & other dependencies - Segmentation

PyTorch & other dependencies - Summaries

PyTorch & other dependencies - Classification

Workloads | Load Balancing | Service Discovery | Volumes

Import YAML | Deploy

Redeploy | Pause Orchestration | Download YAML | Delete

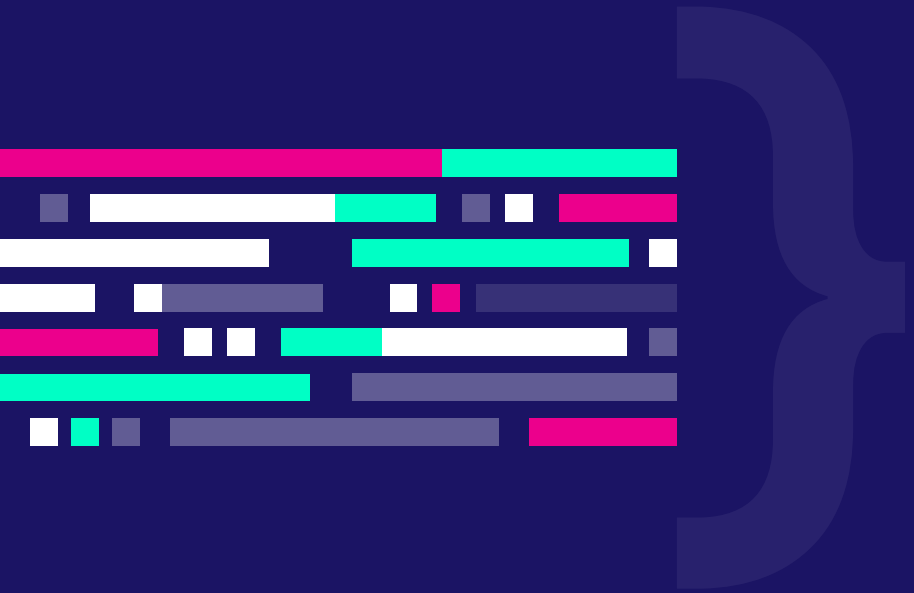
Search

State	Name	Image	Scale
Namespace: etd			
Active	airflow-team2020 80/http, 31596/tcp	ano2202/cs5604-team-2020-int-airflow:0.1 1 Pod / Created a month ago / Pod Restarts: 0	1
Active	ano-test-container 30936/tcp	ano2202/cs5604-segmentation:0.3 1 Pod / Created 2 months ago / Pod Restarts: 0	1
Active	api-team2020 80/http, 30019/tcp	ano2202/cs5604-team-2020-int-api:0.5 1 Pod / Created 17 days ago / Pod Restarts: 0	1
Active	bigrampod	dharneeshkar/segmentation:0.3 1 Pod / Created 21 days ago / Pod Restarts: 0	1
Active	etdweb 80/http, 32739/tcp	sampyash/yolov5:airflow_integration 0 Pods / Created 3 months ago / Pod Restarts: 0	0
Active	frontend-team 80/http, 30493/tcp	code.vt.edu:5005/aaron2000/cs5604-front-end:93d82aaf 1 Pod / Created 2 months ago / Pod Restarts: 0	1
Active	reasoner-team2020 80/http	outerspace1920/team2020-reasoner:1.6 1 Pod / Created 25 days ago / Pod Restarts: 0	1
Active	registry-backend 80/http, 32060/tcp	ano2202/registry-backend:0.0.6-linux 1 Pod / Created a month ago / Pod Restarts: 0	1
Active	sampanna	sampyash/yolov5:airflow_integration 0 Pods / Created 3 months ago / Pod Restarts: 0	0
Active	team-1-container-2 80/http, 30111/tcp	rahulavt/jupyter:first 1 Pod / Created 2 months ago / Pod Restarts: 0	1
Active	team-1-container-3-db	postgres:14-alpine 1 Pod / Created 2 months ago / Pod Restarts: 0	1

Cluster statistics

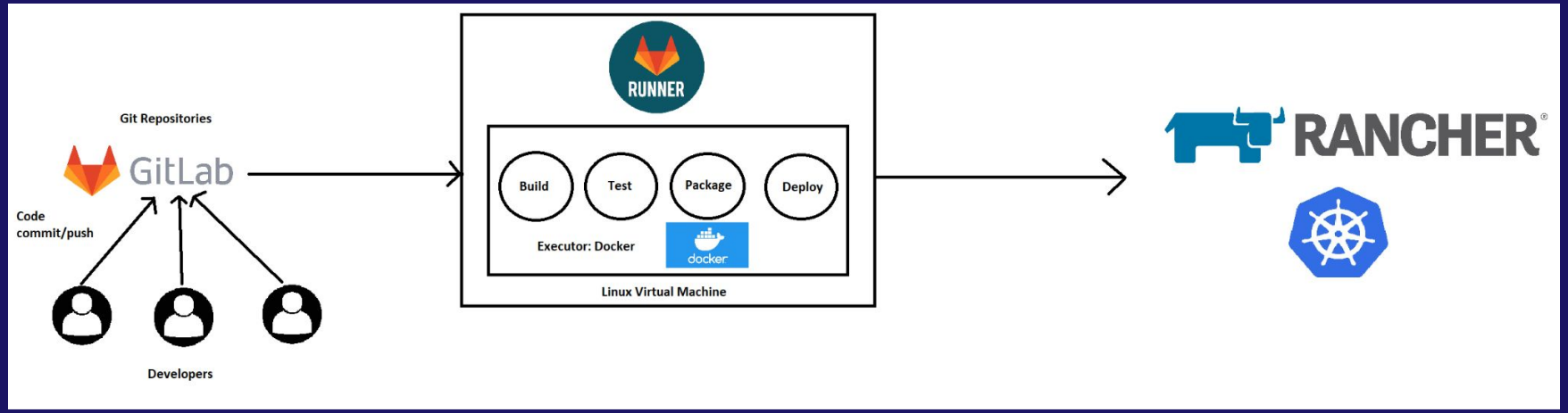
- 45 Deployments
- 40 Ingresses
- 39 Pods
- 180 Services
- 13 Secrets
- Volumes
  - CEPH[1]: etd-cephfs-cs5604-pv:
    - Total: 19TB, Used: 13T, Available: 6.3T, Used: 66%

1. Although the total capacity is 19TB, due to it being a shared space, it was recommended to use no more than 4TB.



CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT

Implementation



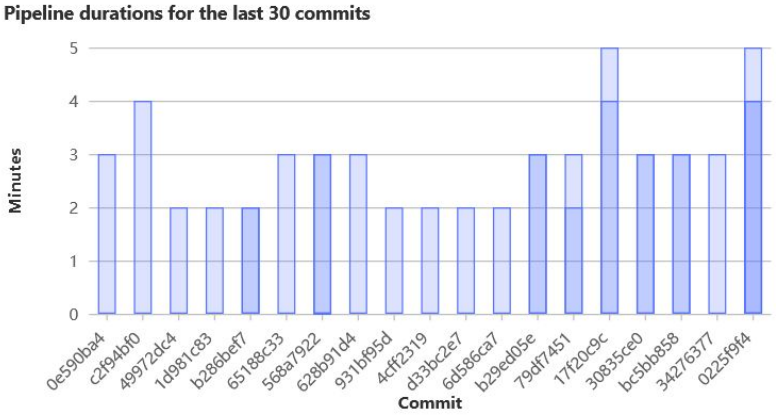
- CI/CD has been implemented for Teams 1, 2, 3 and frontend.
- GitLab Runner is hosted on containers.cs.vt.edu
- All Docker Images are located in code.vt.edu Container Registry.

Frontend CI/CD Analytics

CI/CD Analytics

Overall statistics

- Total: **224 pipelines**
- Successful: **191 pipelines**
- Failed: 12 pipelines
- Success ratio: **94.09%**



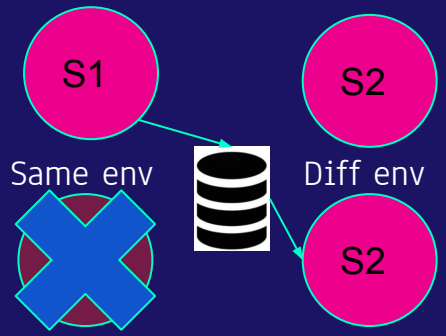
04

Future Work

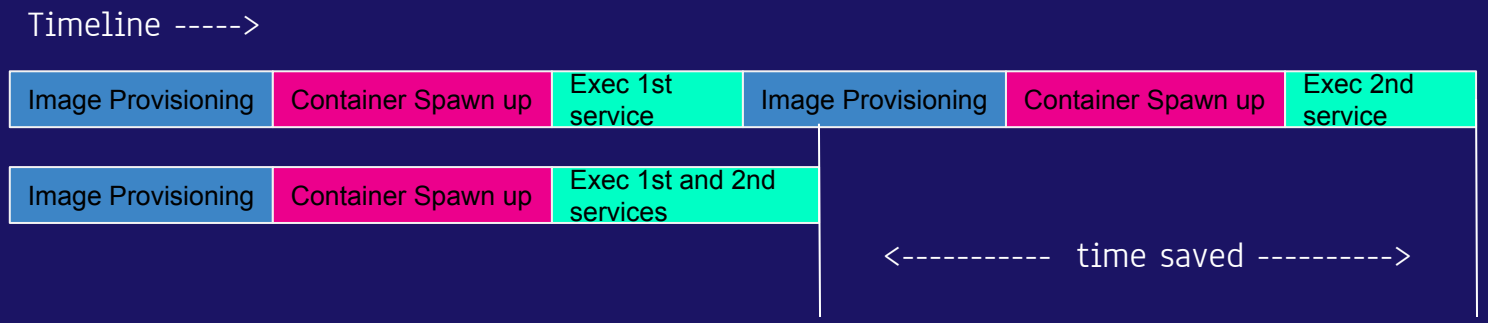
- Reasoner Optimization
  - Saving/loading intermediate data. For example, if two workflows overlap in some steps, the succeeding workflow can directly read the intermediate output produced by the former workflow, instead of executing the workflow end-to-end.

1st time executing:

2nd time executing:



- System Performance Trade-off
  - Observation: Image provisioning & container spawning up take constant time, service execution time varies.
  - Optimization: Merging several insignificant/lightweight services into one monolithic service.
  - Trade-off: Decoupled services VS. user response time.



Future work

Gitlab issue board for issues for future teams [3]

References

[1] Wiki, https://code.vt.edu/aaron2000/cs5604-f22-team-5-repo-2/-/wikis/home

[2] List of container registries: https://hub.docker.com/u/ano2202, https://hub.docker.com/u/outerspace1920, https://hub.docker.com/u/lyuze

[3] Future work, https://code.vt.edu/aaron2000/cs5604-f22-team-5-repo-2/-/issues

[4] Front-end website: https://frontend.discovery.cs.vt.edu/workflows

[5] Postman collection to trigger Workflow API, https://documenter.getpostman.com/view/4087380/2s8YzL2kT9

[6] Helper code for integrating reasoner, https://code.vt.edu/aaron2000/cs5604-front-end/-/blob/team4-experimenter-updates/src/api/AirflowReasoner.js

THANK YOU!

Questions?