Micro-Services Walkthrough

Here are the key points from the walkthrough:

- Every microservice deployment in the company has a file
  - One discussed was the one that controls the CI/CD process through GitLab.
    - The CI/CD pipeline consists of stages and jobs:
      - Jobs within stages typically progress in a procedural, linear manner.
      - Some stages run in parallel, while others have dependencies.
- Most services include templates for various components of the pipeline.
  - Most will include the below

```
include:
  - project: "vector.ai/deploy"
    file: */gitlab-ci/.template.yml
```

This shows the location of the templates file and the deploy

- Build jobs:
  - Often don't differ significantly between microservices.
    The definitions are located in the templates folder in the deploy repo.
- These templates can be overridden if needed, with the override taking precedence over the template.
- Usually only override rules and make minor changes to the build staging job.
- The company uses Kaniko, a tool created by Google, which allows running Docker builds without actually running Docker.
- GitLab automations run inside the staging cluster.
- Service deployments:
  - Are moving towards creating their own namespace for each deployment.
  - This approach helps with organization and easier resource management.
- Helm charts are used for packaging deployments and applying templates.
- Deployment jobs:
  - Often use environment variables pulled from secrets.
  - May include review jobs that create temporary environments.
  - Can have automatic cleanup processes, such as deleting inactive environments after a week.
- The CI/CD configuration file (.gitlab-ci.yml) is the central control point for the CI/CD process.
- Additional components like charts and templates play crucial roles in defining how microservices are built and deployed.
- charts folder is of interest