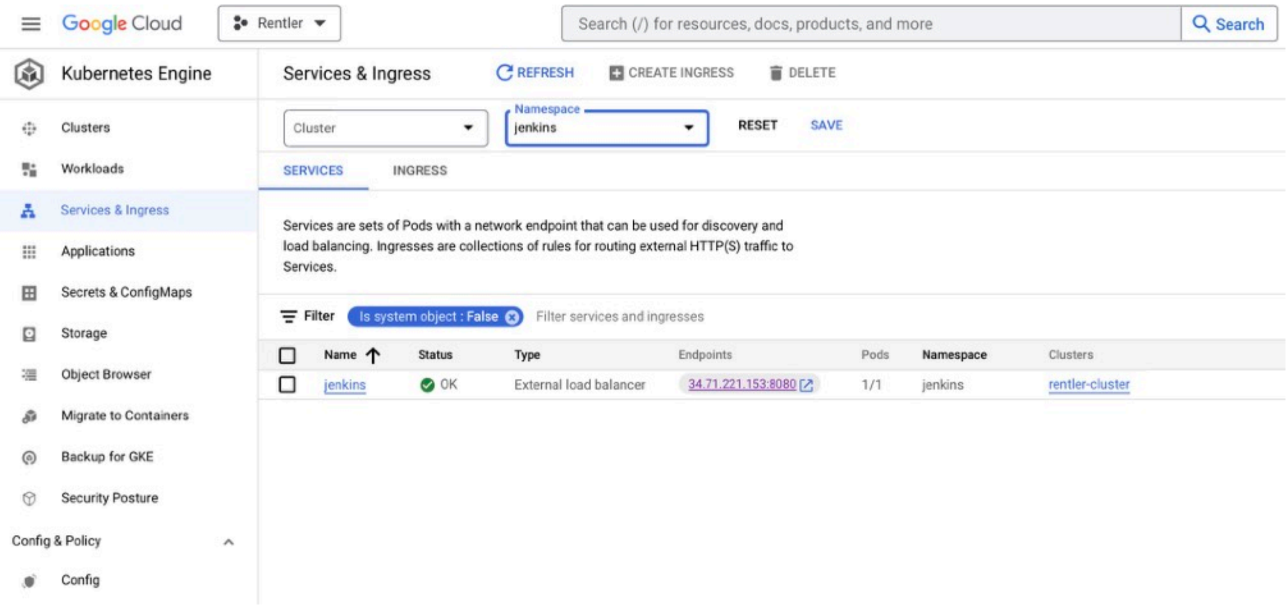
CD/CI Process Documentation using Jenkins and GKE

We implemented CD/CI process using Jenkins deployed as a service on Google Kubernetes Engine because it is considered one of the best tools due to such advantages as scalability, flexible plugin ecosystem, robust security features, support for Infrastructure as Code (IaC), and seamless integration with GKE's scalable infrastructure.



Namespace: jenkins

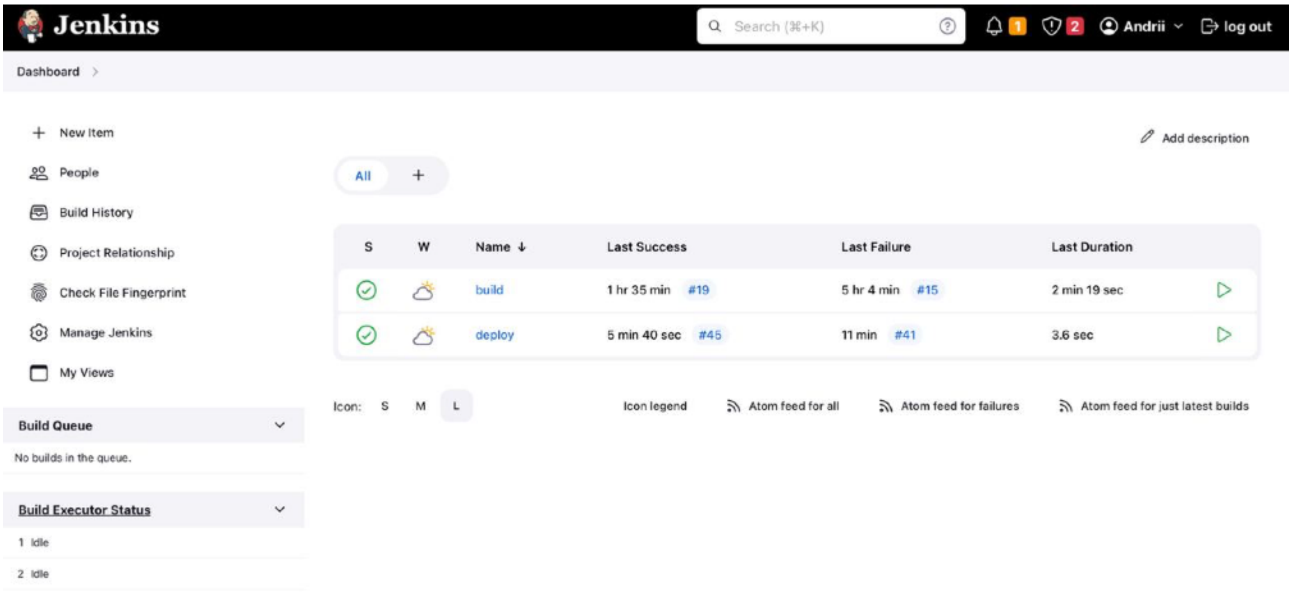
Services are sets of Pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to Services.

Is system object : False

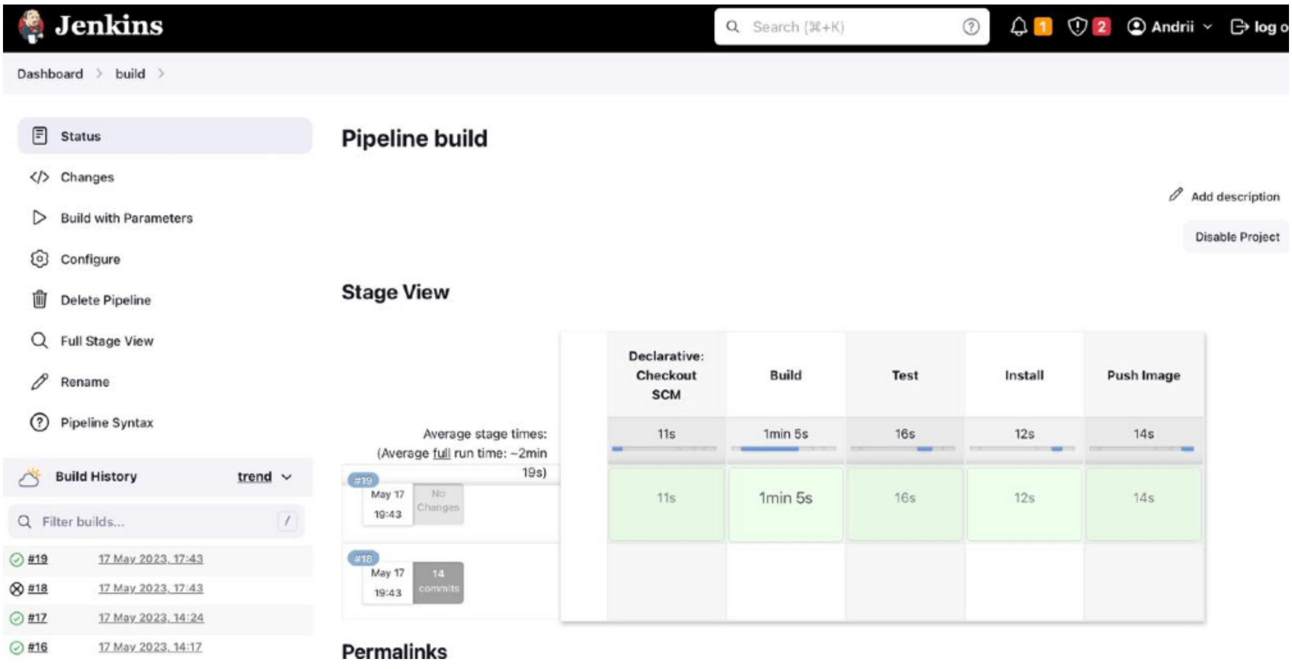
□	Name ↑	Status	Type	Endpoints	Pods	Namespace	Clusters
□	jenkins	✓ OK	External load balancer	34.71.221.153:8080	1/1	jenkins	rentler-cluster

The CD/CI process implemented using Jenkins and GKE consists of two jobs: **build** and **deploy**. The process is as follows:

1. Build Job:
    - This job is automatically triggered on a commit to the GitHub repository.
    - It starts by pulling the code from the GitHub repository.
    - Automated tests are executed to ensure the quality of the code.
    - If the test coverage is insufficient, the build job fails.
    - The Maven artifacts generated during the build are installed to the Artifact Registry.
    - A Docker image is built using the Maven artefacts from Artifact Registry.
    - The Docker image is then pushed to the Container Registry.
    - Kaniko is utilized to build Docker images because it enables secure and efficient image building within the Jenkins agent container without requiring privileged access or the use of a Docker daemon.



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		build	1 hr 35 min #19	5 hr 4 min #15	2 min 19 sec
✓		deploy	5 min 40 sec #45	11 min #41	3.6 sec



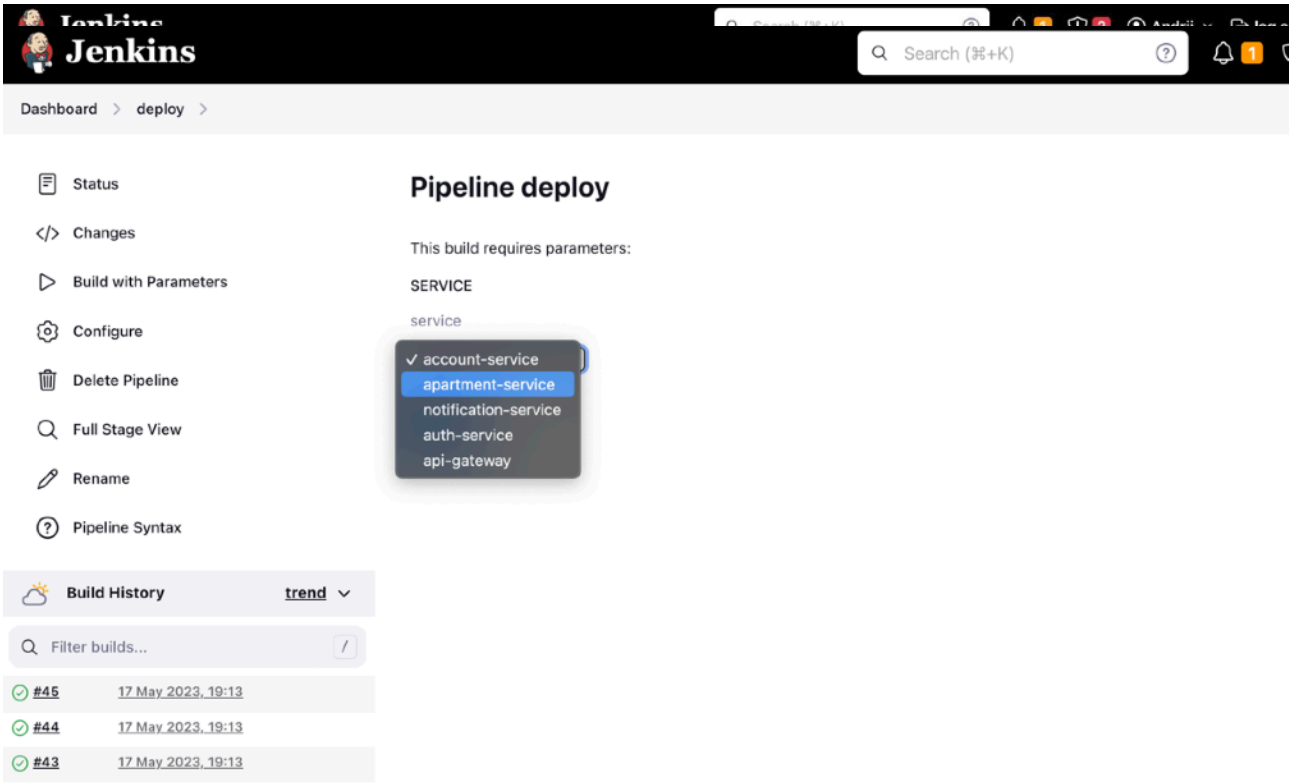
Pipeline build

Stage View

	Declarative: Checkout SCM	Build	Test	Install	Push Image
Average stage times: (Average full run time: ~2min 19s)	11s	1min 5s	16s	12s	14s
#19 May 17 19:43 No Changes	11s	1min 5s	16s	12s	14s
#18 May 17 19:43 14 commits					

✓ #19	17 May 2023, 17:43
⊗ #18	17 May 2023, 17:43
✓ #17	17 May 2023, 14:24
✓ #16	17 May 2023, 14:17

2. Deploy Job:
   - This job is responsible for deploying the Docker image to GKE.
   - It pulls the Docker image from the Container Registry.
   - The image is deployed to the Google Kubernetes Engine (GKE) cluster.
   - If the build job fails, the deploy job uses the previously built Docker image to ensure a fallback option.



Pipeline deploy

This build requires parameters:

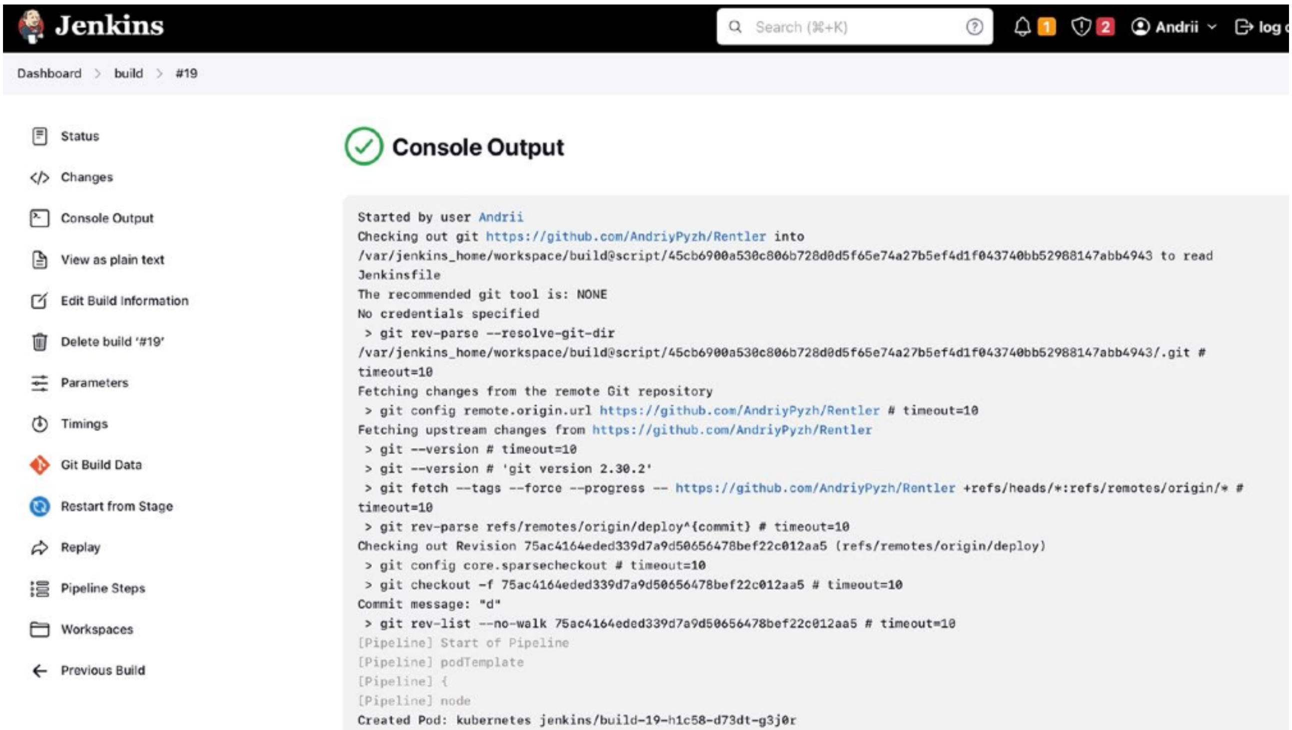
SERVICE

service

✓ account-service
apartment-service
notification-service
auth-service
api-gateway

Key Features of the CD/CI Process:

- **Automatic Triggering:** The build job is automatically triggered on each commit to the GitHub repository, ensuring a continuous integration process.
- **Test Coverage Check:** If the test coverage is found to be insufficient during the build job, the process fails, indicating the need for improvement in test coverage.
- **Parameterization:** Both the build and deploy jobs are parameterized, allowing flexibility to build and deploy any service to GKE as required.
- **Dynamic Agent Creation:** Jenkins creates agents such as Maven and Docker using GKE resources, ensuring efficient resource utilization and scalability.



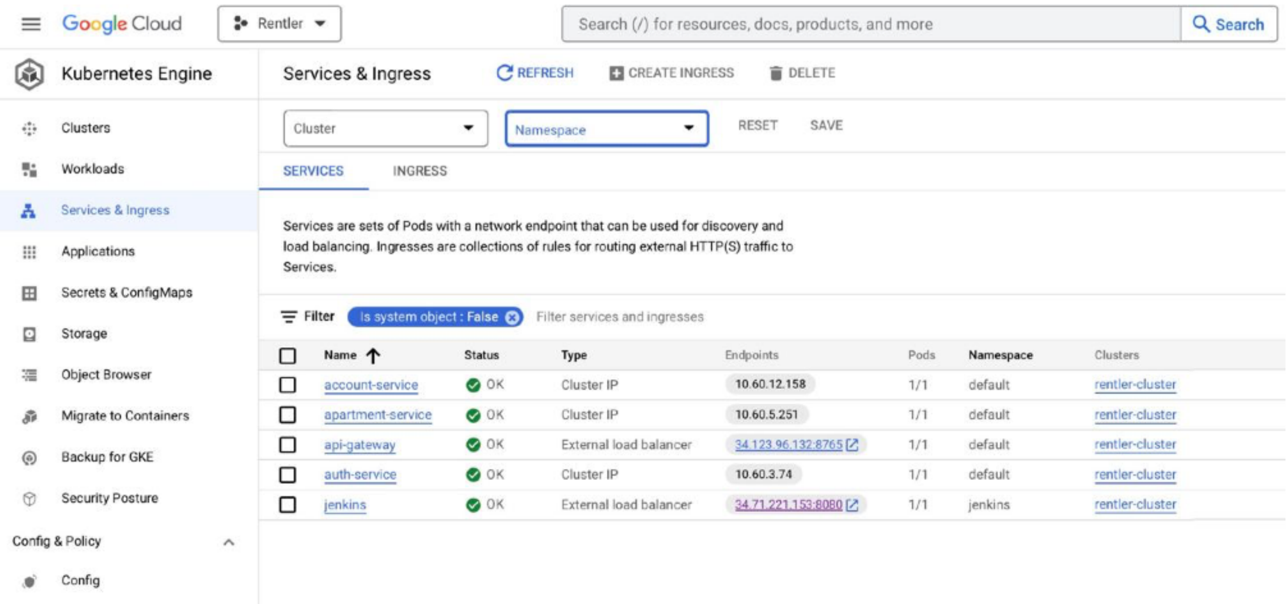
Console Output

```
Started by user Andrii
Checking out git https://github.com/AndriyPyzh/Rentler into
/var/jenkins_home/workspace/build@script/45cb6900a530c806b728d0d5f65e74a27b5ef4d1f043740bb52988147abb4943 to read
Jenkinsfile
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir
/var/jenkins_home/workspace/build@script/45cb6900a530c806b728d0d5f65e74a27b5ef4d1f043740bb52988147abb4943/.git #
timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/AndriyPyzh/Rentler # timeout=10
Fetching upstream changes from https://github.com/AndriyPyzh/Rentler
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
 > git fetch --tags --force --progress -- https://github.com/AndriyPyzh/Rentler +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > git rev-parse refs/remotes/origin/deploy^{commit} # timeout=10
Checking out Revision 75ac4164eded339d7a9d50656478bef22c012aa5 (refs/remotes/origin/deploy)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 75ac4164eded339d7a9d50656478bef22c012aa5 # timeout=10
Commit message: "d"
 > git rev-list --no-walk 75ac4164eded339d7a9d50656478bef22c012aa5 # timeout=10
[Pipeline] Start of Pipeline
[Pipeline] podTemplate
[Pipeline] {
[Pipeline] node
Created Pod: kubernetes jenkins/build-19-h1c58-d73dt-g3j0r
```

Both the **build** and **deploy** Jenkins jobs are created from a Jenkinsfile, which provides several benefits:

- using a Jenkinsfile allows for version control, ensuring that the pipeline definition evolves with the codebase and making it easier to track changes and collaborate.
- it enables reproducibility and consistency by defining the exact steps and configurations required for the CD/CI process. This ensures that every execution follows the defined pipeline, reducing errors and inconsistencies.
- Jenkinsfile promotes automation by codifying the pipeline, eliminating the need for manual configuration and reducing human error.

Overall, the use of Jenkinsfiles for defining Jenkins jobs enhances maintainability, repeatability, and scalability of the CD/CI process.



Services are sets of Pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to Services.

□	Name ↑	Status	Type	Endpoints	Pods	Namespace	Clusters
□	account-service	✓ OK	Cluster IP	10.60.12.158	1/1	default	rentler-cluster
□	apartment-service	✓ OK	Cluster IP	10.60.5.251	1/1	default	rentler-cluster
□	api-gateway	✓ OK	External load balancer	34.123.96.132:8765	1/1	default	rentler-cluster
□	auth-service	✓ OK	Cluster IP	10.60.3.74	1/1	default	rentler-cluster
□	jenkins	✓ OK	External load balancer	34.71.221.153:8080	1/1	jenkins	rentler-cluster

As a result, we implemented CD/CI process with Jenkins and GKE that provides a streamlined and automated workflow for building, testing, and deploying services, promoting faster development cycles and improved software quality.