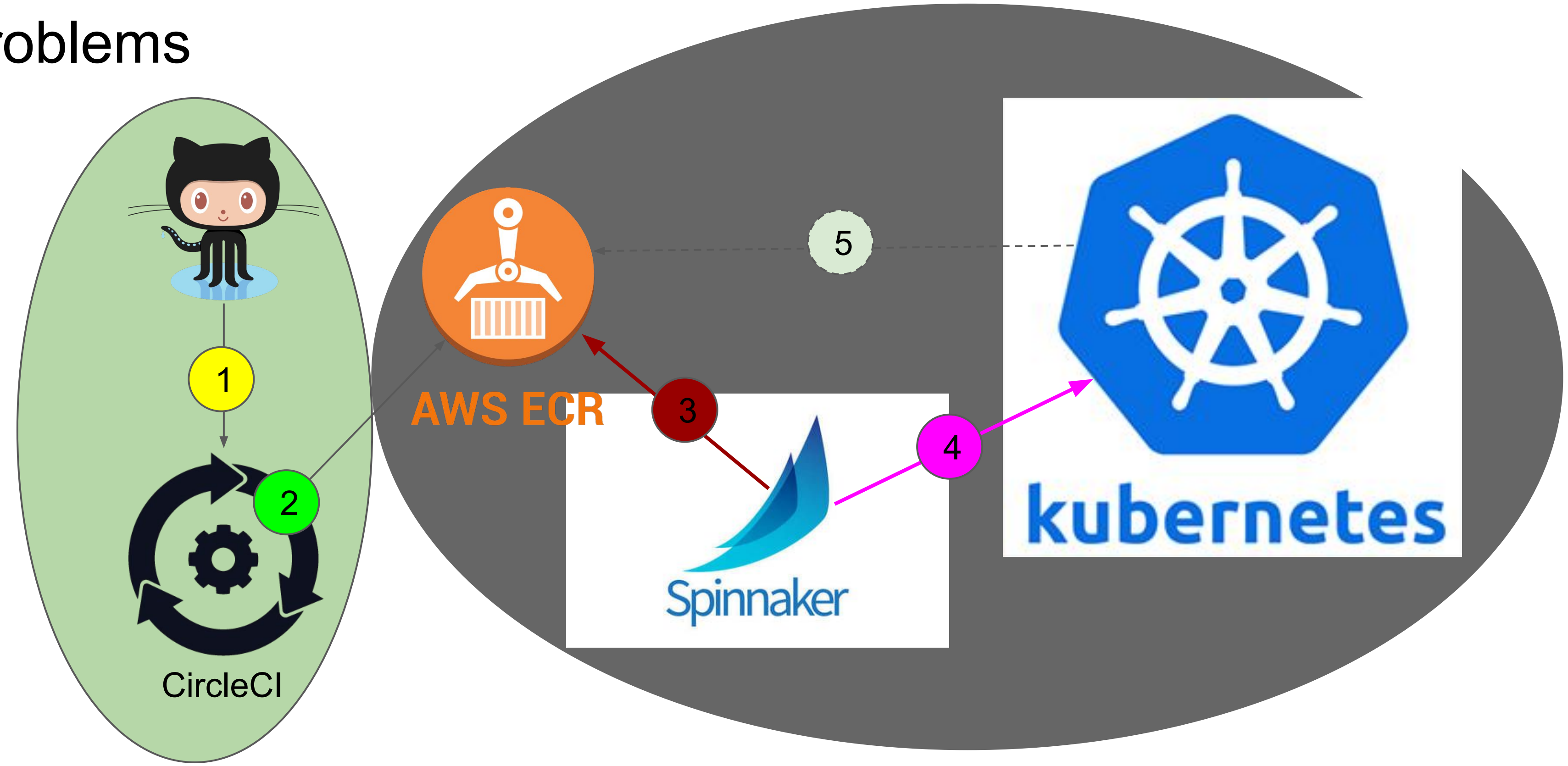
Release Engineering for EKS

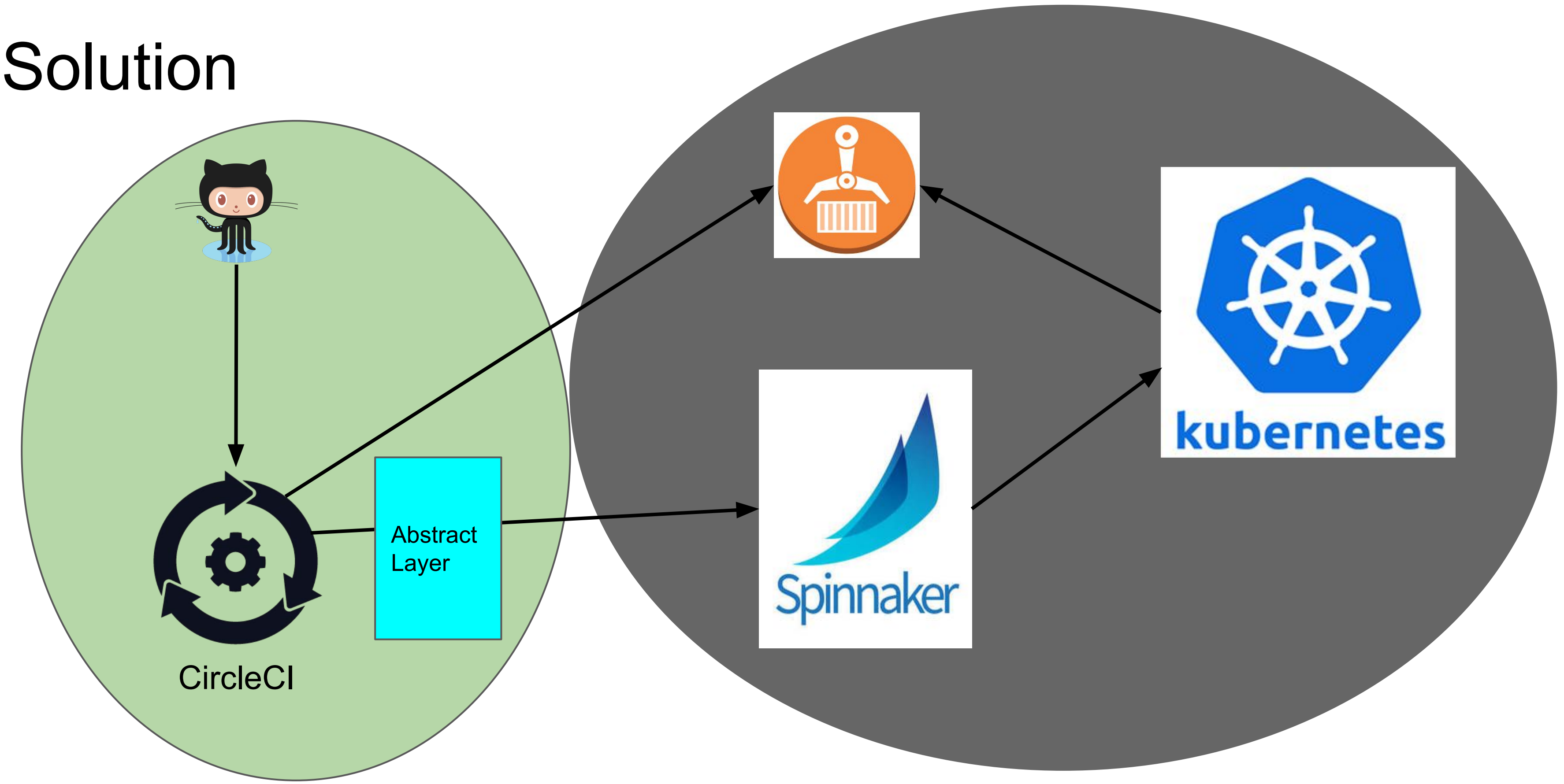
Problems



When creating a new application

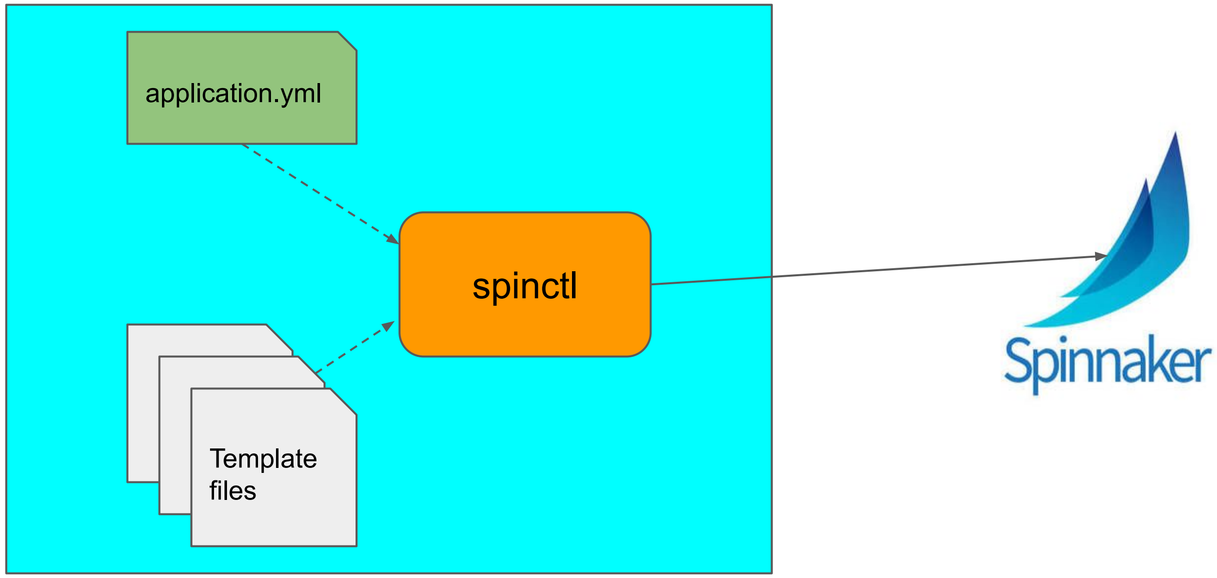
1. **Developers** cannot deliver the application to EKS by themselves because **they** don’t know how to create Spinnaker pipelines.
2. Spinnaker pipelines **are complicated and it takes time to get acquainted with.**
3. **Developers** cannot change Spinnaker pipelines, for example, **they** cannot add new async workers.
4. If the execution of Spinnaker pipelines is FAILED, developers need to open Spinnaker to check the errors ⇒ it’s not COOL!

Solution



Create an abstract layer between (Developers + CircleCI) and Spinnaker.
This layer will solve all problems.

Zoom in Abstract Layer

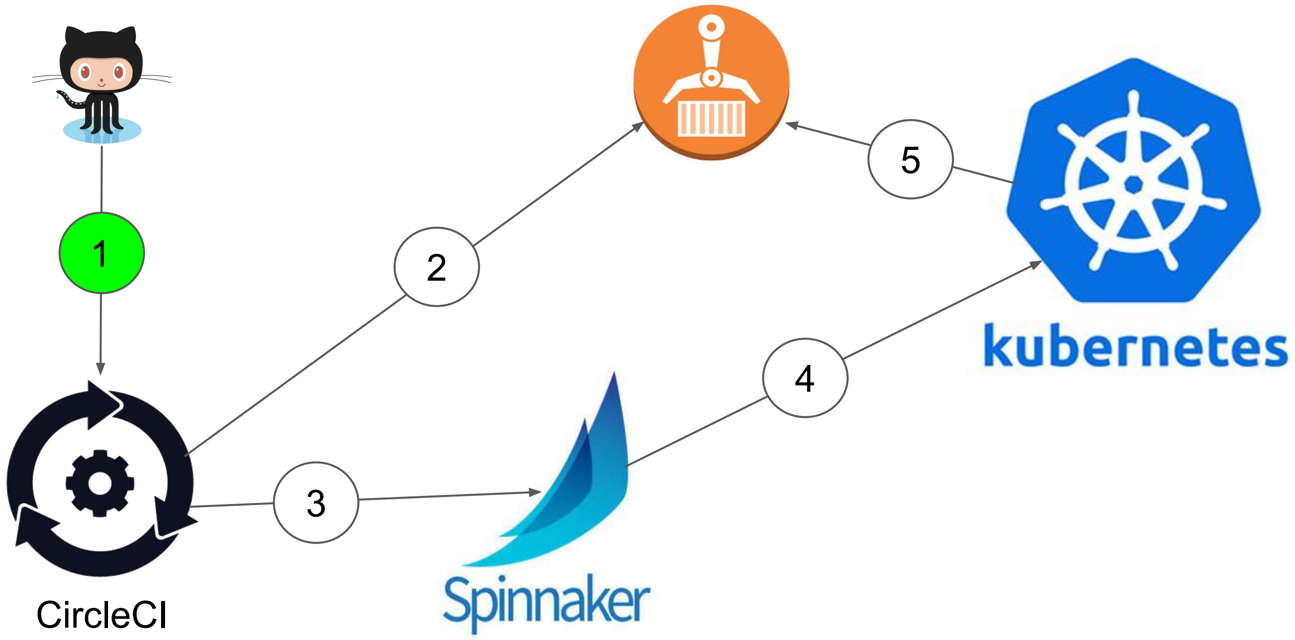


- **spinctl** is a customized version of the **spin** program, it reads **application.yml** and **template files**
- **application.yml** and **template files** are generated from **rails_application_template**
- **application.yml** is an abstract of Spinnaker pipelines
- Developers don’t need to understand template files
- Developers can easily understand and edit **application.yml** by themselves

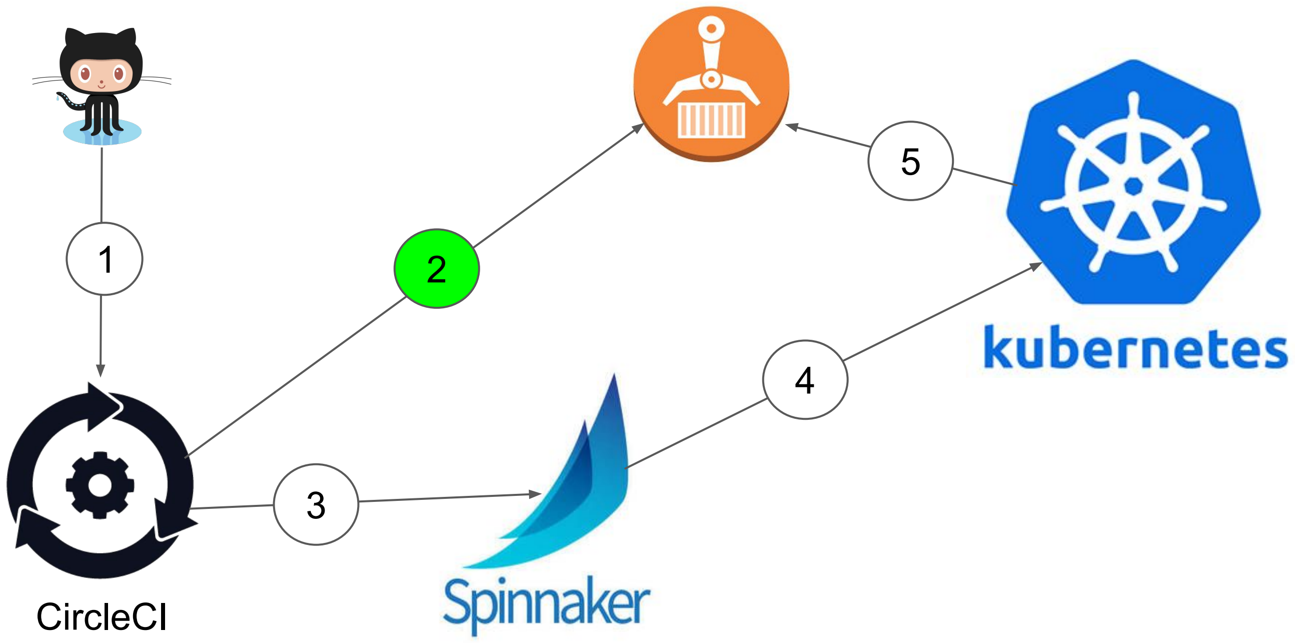
A sample **application.yml**

https://github.com/ngocson2vn/spinctl/blob/master/example/spinnaker/application.yml

With the abstract layer, the CI/CD flow becomes like this

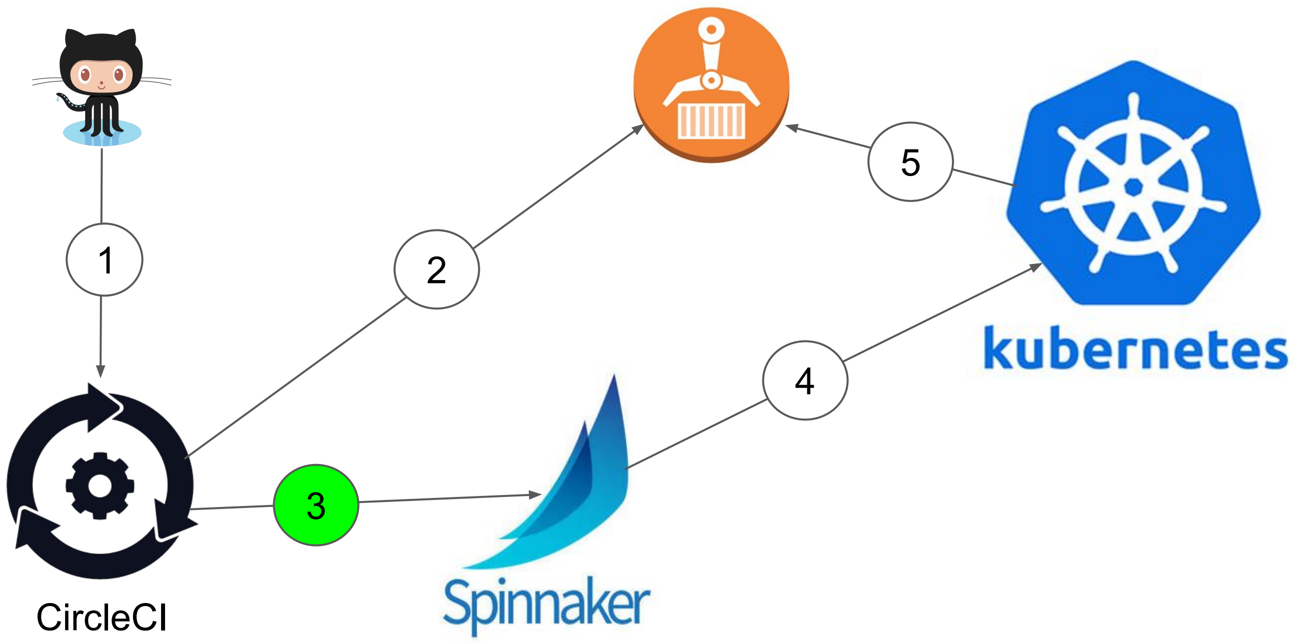


Step (1)
- Circle CI fetches source code, executes unit tests and then builds docker images



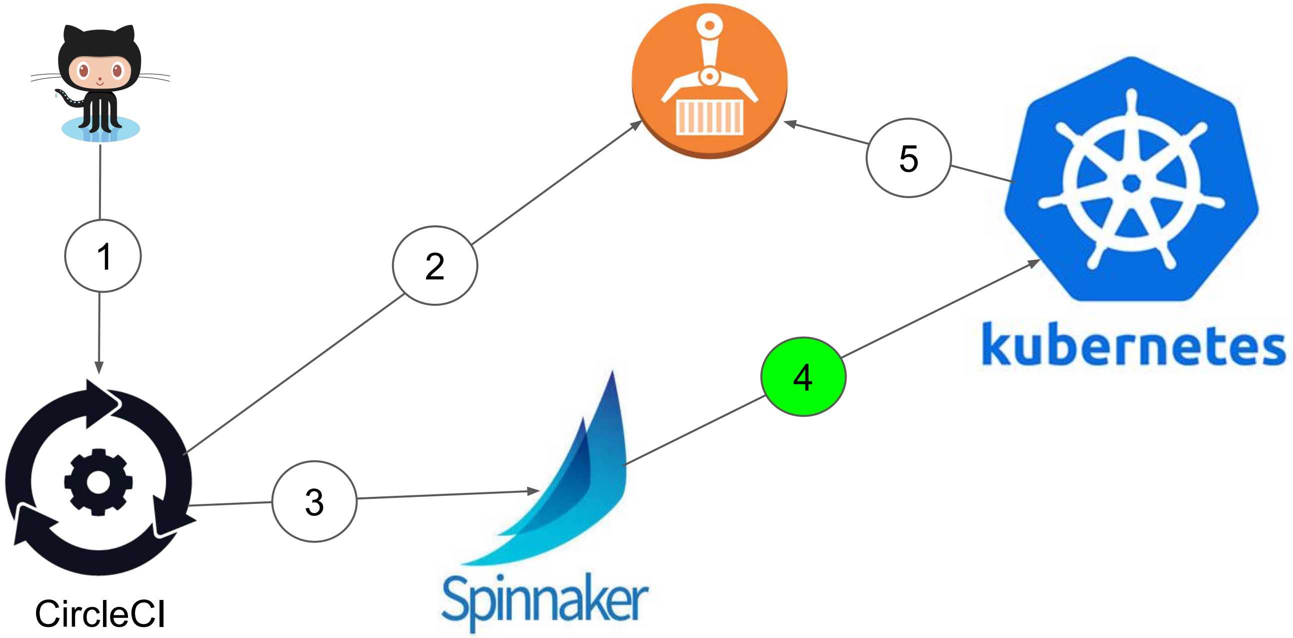
Step (2)

- Circle CI pushes docker images to AWS ECR



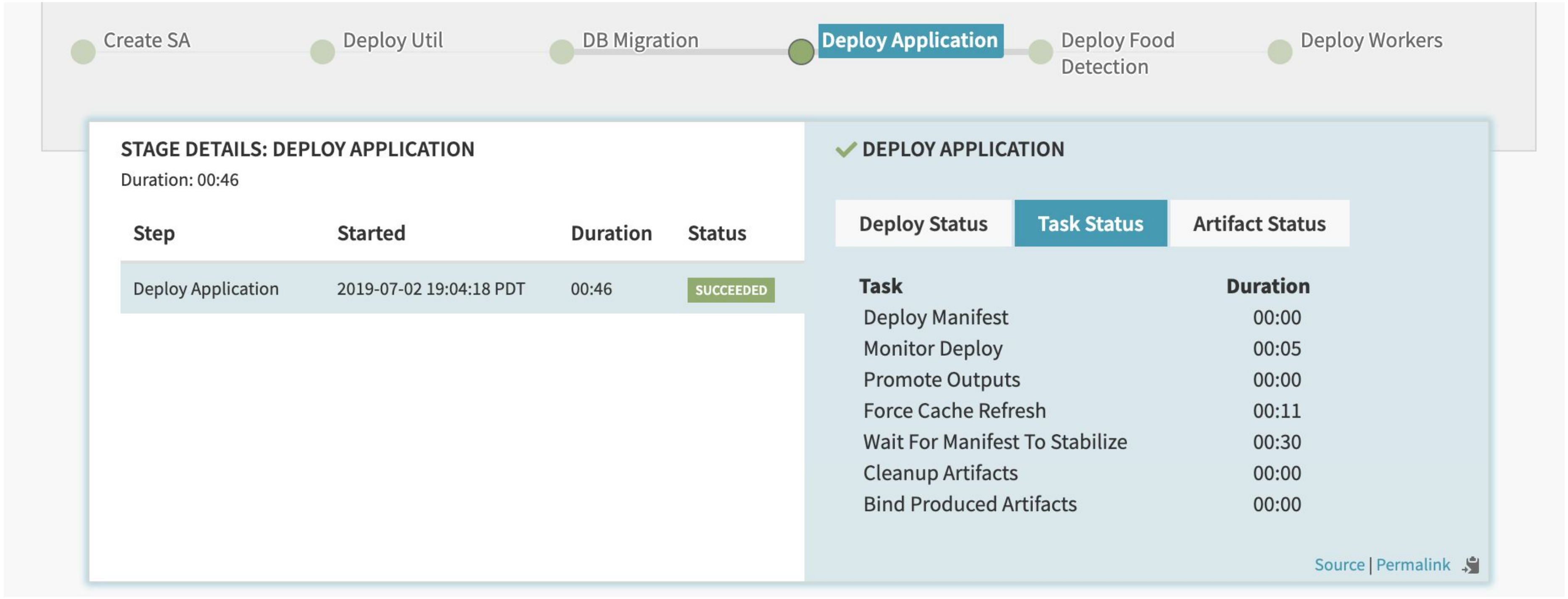
Step (3)

- CircleCI executes **spinctl**
- **spinctl** reads **application.yml and template files** and then
  - Creates a Spinnaker application if the application **does not exist**
  - Creates new Spinnaker pipelines with the latest docker images
  - Executes Spinnaker pipelines having **executable** attribute
  - Monitors the execution of the triggered Spinnaker pipelines
    - Gets the triggered pipeline from Spinnaker API server
    - Prints the status of pipeline and the status of stages
    - If the pipeline **SUCCEEDS**, exit with code 0
    - If the pipeline **FAILS**, print logs of the failed stage and exit with code 1



Step (4)

- Spinnaker executes pipelines
  - Each pipeline is composed of many stages
    - Each stage performs a specific task such as executing db:migrate, rolling update an application



STAGE DETAILS: DEPLOY APPLICATION

Duration: 00:46

Step	Started	Duration	Status
Deploy Application	2019-07-02 19:04:18 PDT	00:46	SUCCEEDED

✓ DEPLOY APPLICATION

Task	Duration
Deploy Manifest	00:00
Monitor Deploy	00:05
Promote Outputs	00:00
Force Cache Refresh	00:11
Wait For Manifest To Stabilize	00:30
Cleanup Artifacts	00:00
Bind Produced Artifacts	00:00

More about spinctl

spin: https://github.com/spinnaker/spin

spinctl: https://github.com/ngocson2vn/spinctl

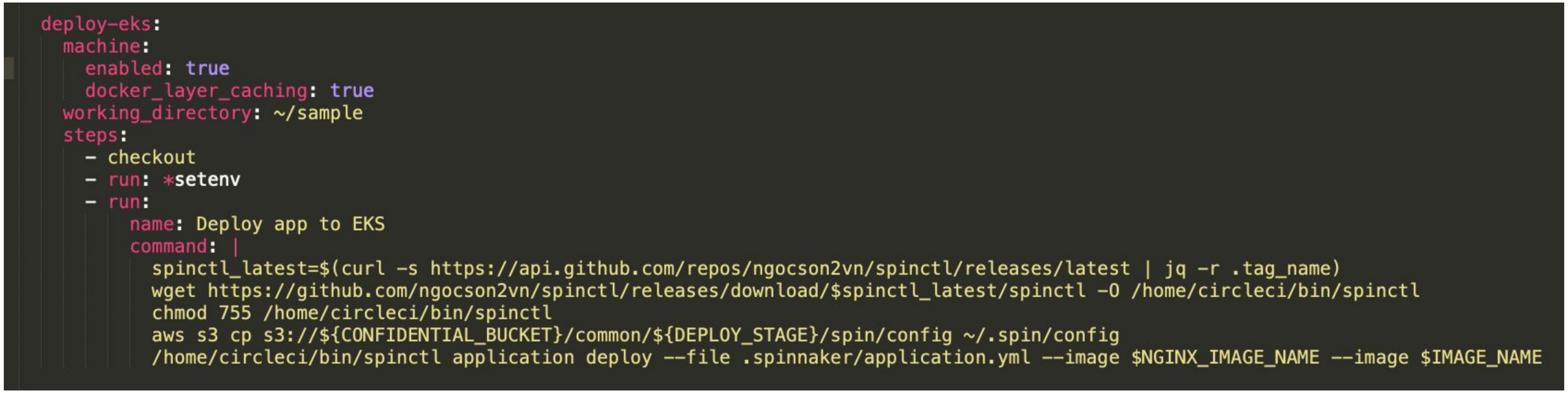
I have customized **spin** to become **spinctl**.

Demo

application.yml and template files

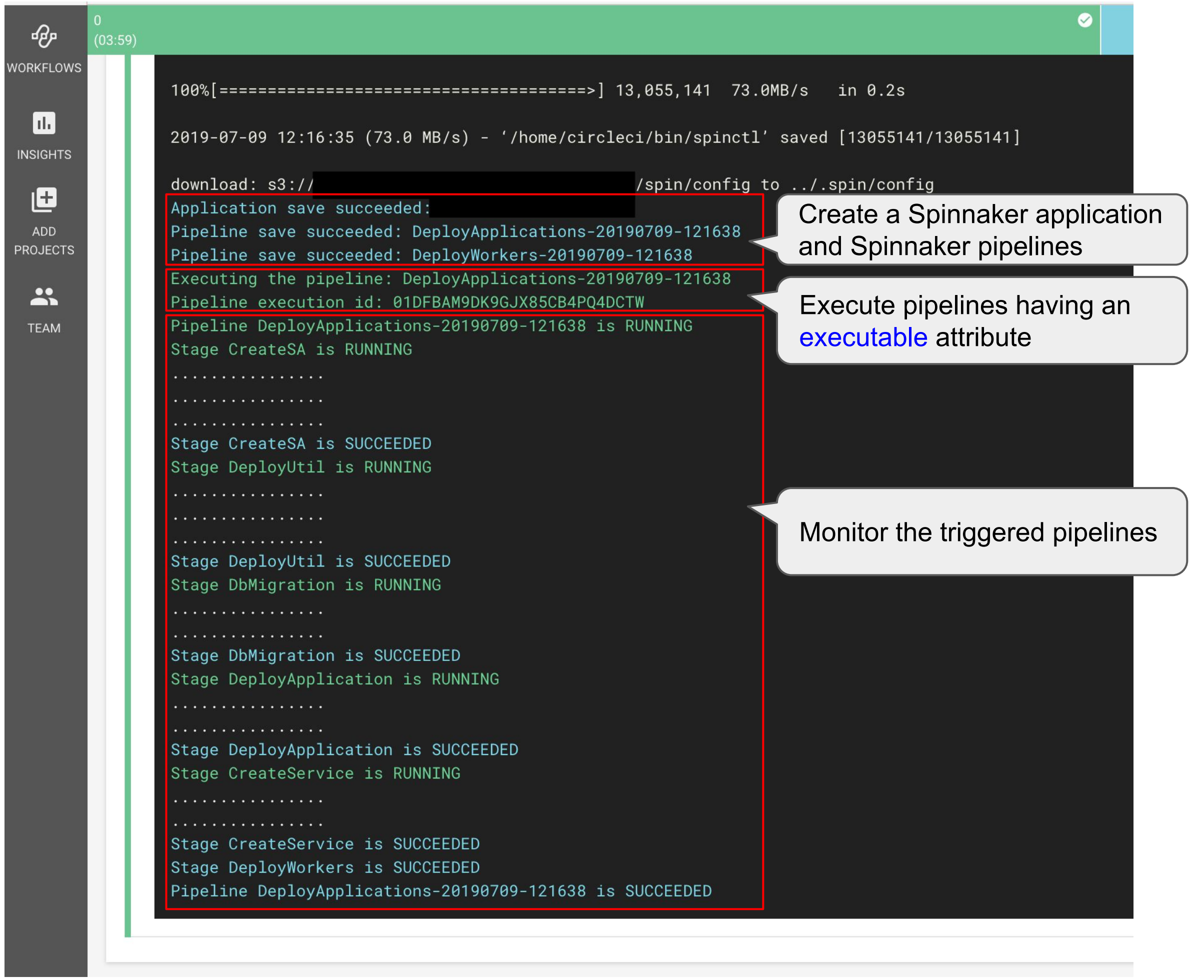
https://github.com/ngocson2vn/spinctl/tree/master/example/spinnaker

CircleCI config



```yaml
deploy-eks:
  machine:
    enabled: true
    docker_layer_caching: true
  working_directory: ~/sample
  steps:
    - checkout
    - run: *setenv
    - run:
        name: Deploy app to EKS
        command: |
          spinctl_latest=$(curl -s https://api.github.com/repos/ngocson2vn/spinctl/releases/latest | jq -r .tag_name)
          wget https://github.com/ngocson2vn/spinctl/releases/download/$spinctl_latest/spinctl -O /home/circleci/bin/spinctl
          chmod 755 /home/circleci/bin/spinctl
          aws s3 cp s3://${CONFIDENTIAL_BUCKET}/common/${DEPLOY_STAGE}/spin/config ~/.spin/config
          /home/circleci/bin/spinctl application deploy --file .spinnaker/application.yml --image $NGINX_IMAGE_NAME --image $IMAGE_NAME
```

A SUCCESS case



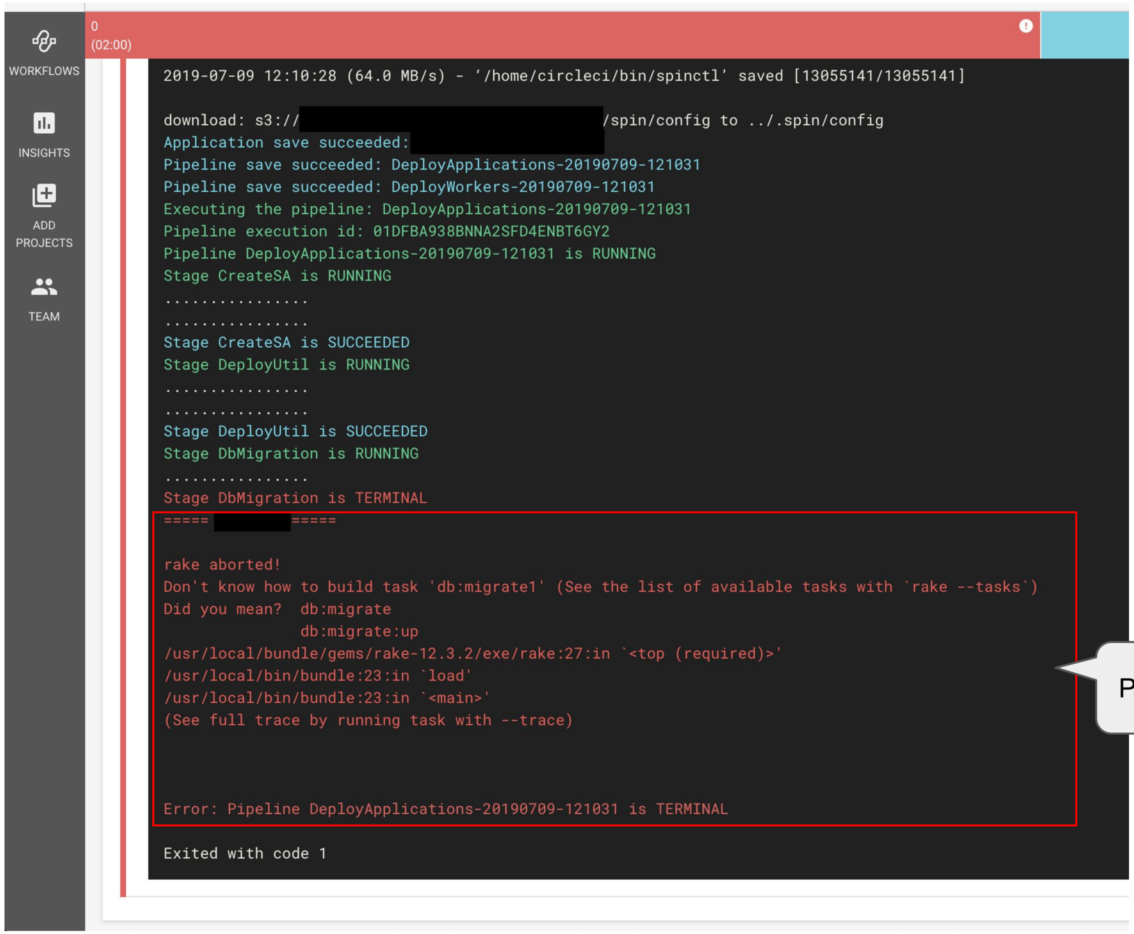
```
100%[====================================>] 13,055,141  73.0MB/s   in 0.2s

2019-07-09 12:16:35 (73.0 MB/s) - ‘/home/circleci/bin/spinctl’ saved [13055141/13055141]

download: s3://[redacted]/spin/config to ../.spin/config
Application save succeeded: [redacted]
Pipeline save succeeded: DeployApplications-20190709-121638
Pipeline save succeeded: DeployWorkers-20190709-121638
Executing the pipeline: DeployApplications-20190709-121638
Pipeline execution id: 01DFBAM9DK9GJX85CB4PQ4DCTW
Pipeline DeployApplications-20190709-121638 is RUNNING
Stage CreateSA is RUNNING
...............
...............
...............
Stage CreateSA is SUCCEEDED
Stage DeployUtil is RUNNING
...............
...............
...............
Stage DeployUtil is SUCCEEDED
Stage DbMigration is RUNNING
...............
...............
Stage DbMigration is SUCCEEDED
Stage DeployApplication is RUNNING
...............
...............
Stage DeployApplication is SUCCEEDED
Stage CreateService is RUNNING
...............
...............
Stage CreateService is SUCCEEDED
Stage DeployWorkers is SUCCEEDED
Pipeline DeployApplications-20190709-121638 is SUCCEEDED
```

Create a Spinnaker application and Spinnaker pipelines

Execute pipelines having an executable attribute

Monitor the triggered pipelines

A FAILED case



```
2019-07-09 12:10:28 (64.0 MB/s) - '/home/circleci/bin/spinctl' saved [13055141/13055141]

download: s3://[redacted]/spin/config to ../.spin/config
Application save succeeded: [redacted]
Pipeline save succeeded: DeployApplications-20190709-121031
Pipeline save succeeded: DeployWorkers-20190709-121031
Executing the pipeline: DeployApplications-20190709-121031
Pipeline execution id: 01DFBA938BNNA2SFD4ENBT6GY2
Pipeline DeployApplications-20190709-121031 is RUNNING
Stage CreateSA is RUNNING
................
................
Stage CreateSA is SUCCEEDED
Stage DeployUtil is RUNNING
................
................
Stage DeployUtil is SUCCEEDED
Stage DbMigration is RUNNING
................
Stage DbMigration is TERMINAL
===== [redacted] =====

rake aborted!
Don't know how to build task 'db:migrate1' (See the list of available tasks with `rake --tasks`)
Did you mean?  db:migrate
               db:migrate:up
/usr/local/bundle/gems/rake-12.3.2/exe/rake:27:in `<top (required)>'
/usr/local/bin/bundle:23:in `load'
/usr/local/bin/bundle:23:in `<main>'
(See full trace by running task with --trace)



Error: Pipeline DeployApplications-20190709-121031 is TERMINAL

Exited with code 1
```

Print the error logs