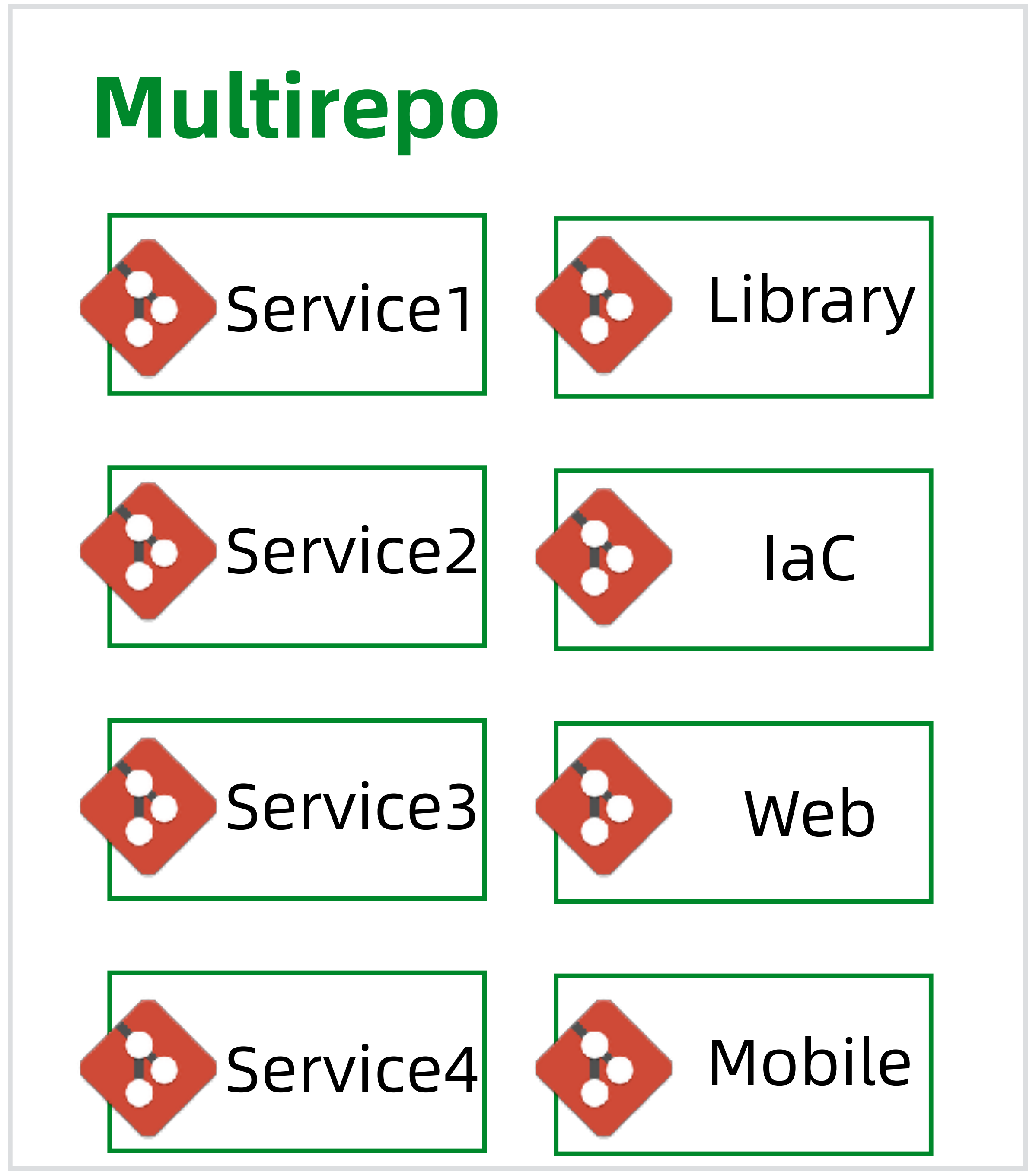
第 2 节

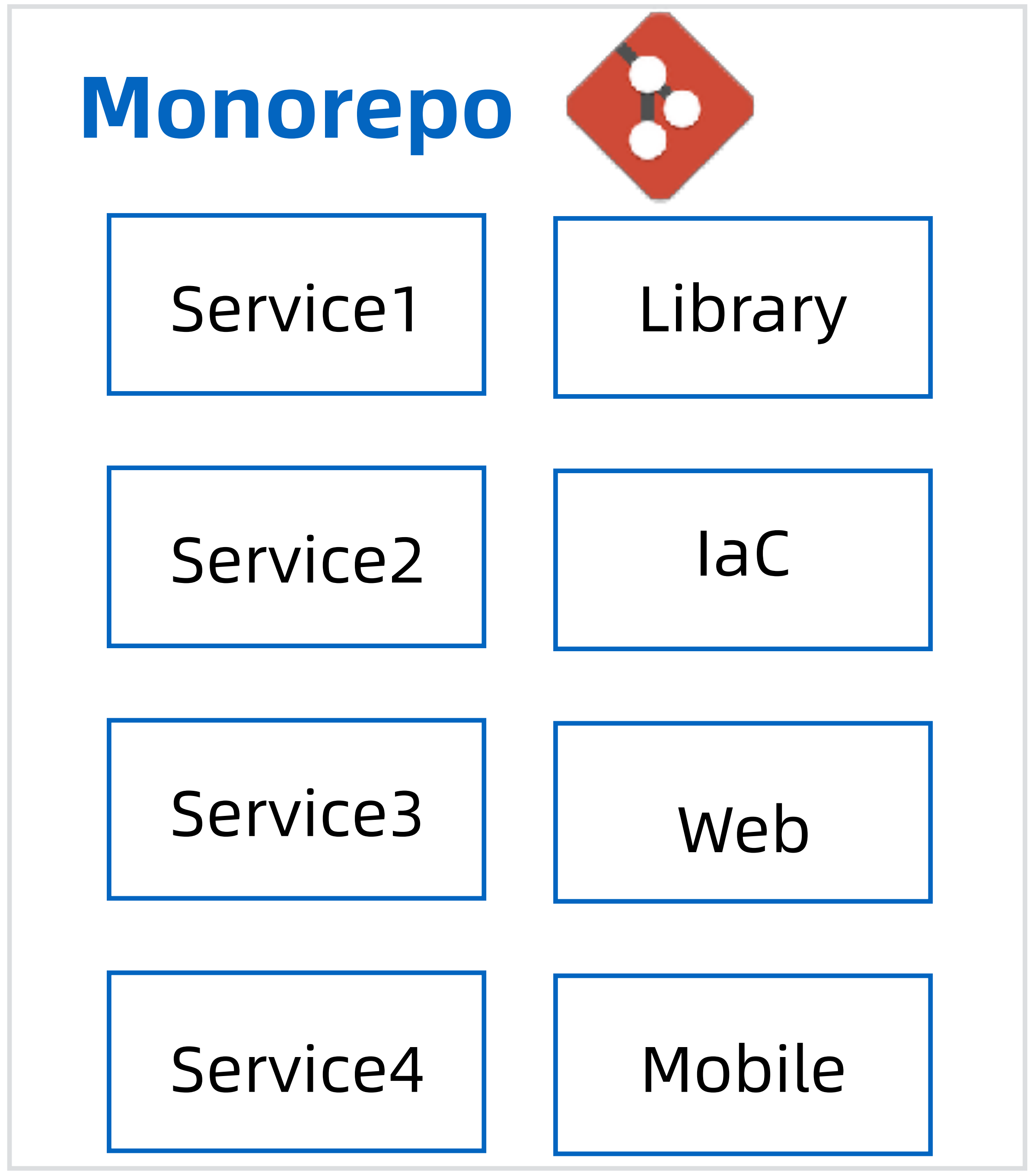
基于单体仓库的CI方案

本课内容

- 单体仓库(monorepo) vs 多仓库(multirepo)
- 一种基于单体仓库和CircleCI的CI方案

单体仓库(monorepo) vs 多仓库(multirepo)

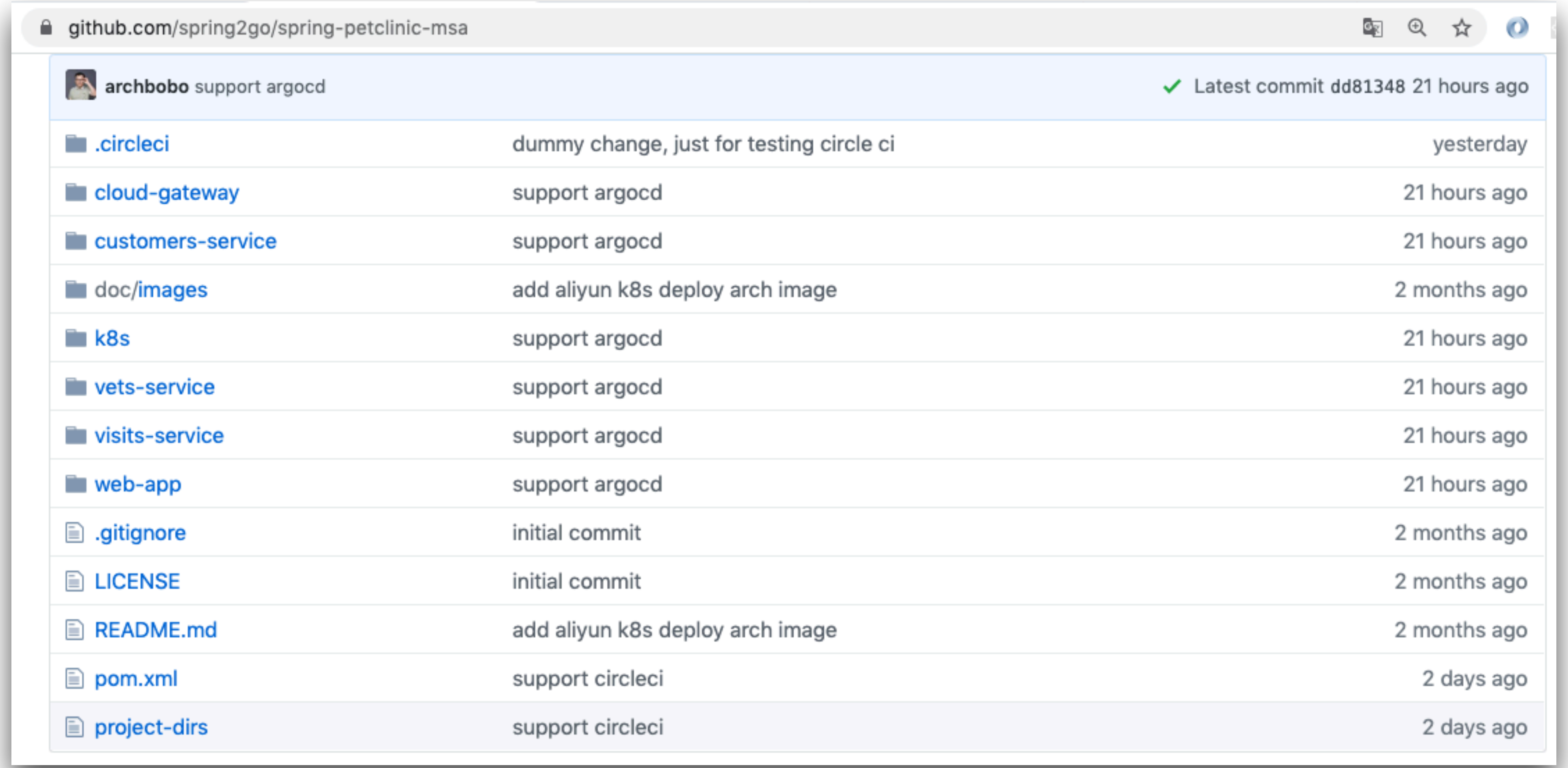




1. 职责归属清晰
2. 易于扩展
3. 限制clone范围

1. 易于开发者测试
2. 易于标准化代码
3. 易于开展Code Reivew
4. 易于共享公共组件
5. 易于重构

Petclinic微服务单体仓库

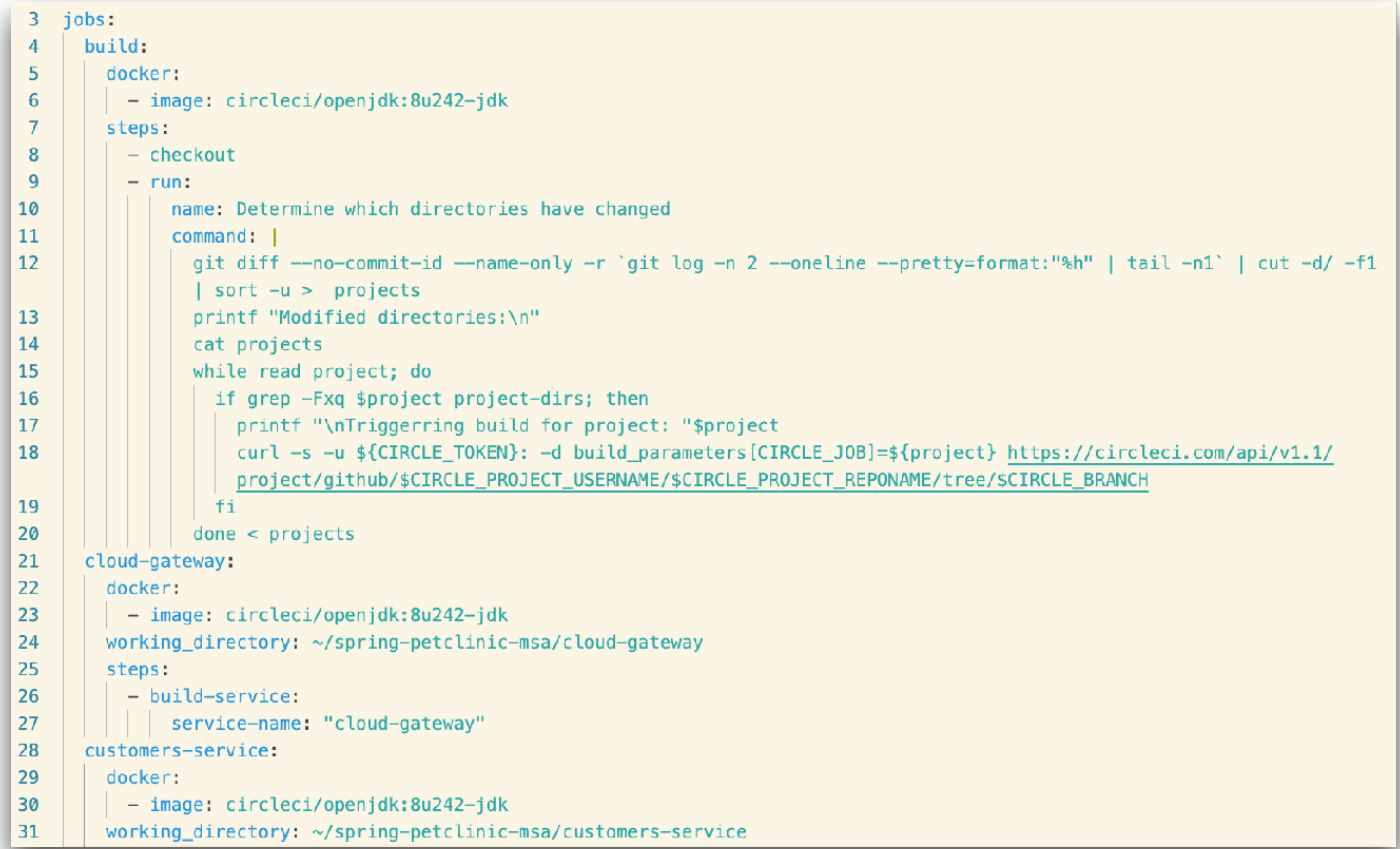


github.com/spring2go/spring-petclinic-msa

archbobo support argocd ✓ Latest commit dd81348 21 hours ago

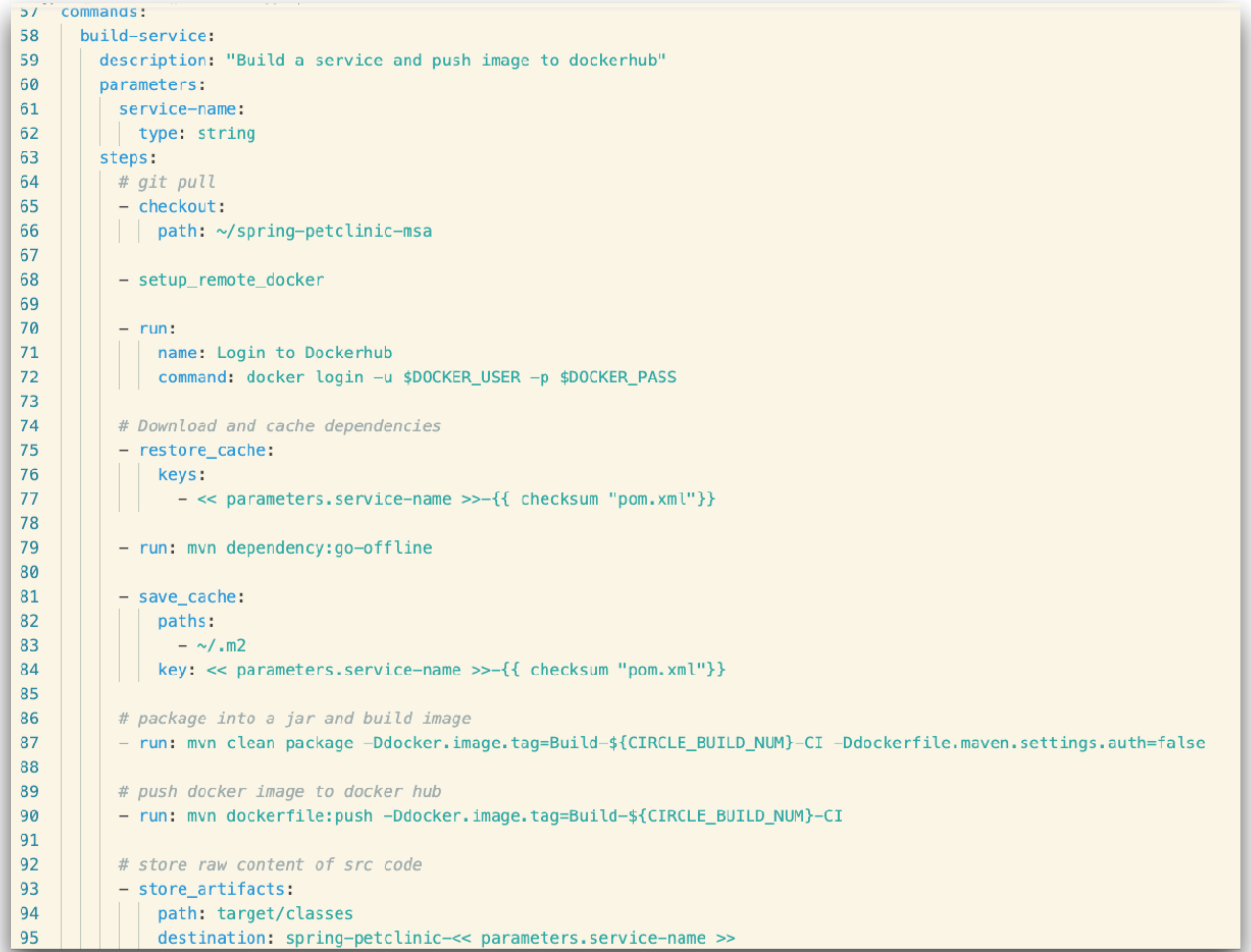
.circleci	dummy change, just for testing circle ci	yesterday
cloud-gateway	support argocd	21 hours ago
customers-service	support argocd	21 hours ago
doc/images	add aliyun k8s deploy arch image	2 months ago
k8s	support argocd	21 hours ago
vets-service	support argocd	21 hours ago
visits-service	support argocd	21 hours ago
web-app	support argocd	21 hours ago
.gitignore	initial commit	2 months ago
LICENSE	initial commit	2 months ago
README.md	add aliyun k8s deploy arch image	2 months ago
pom.xml	support circleci	2 days ago
project-dirs	support circleci	2 days ago

CircleCI config.yml



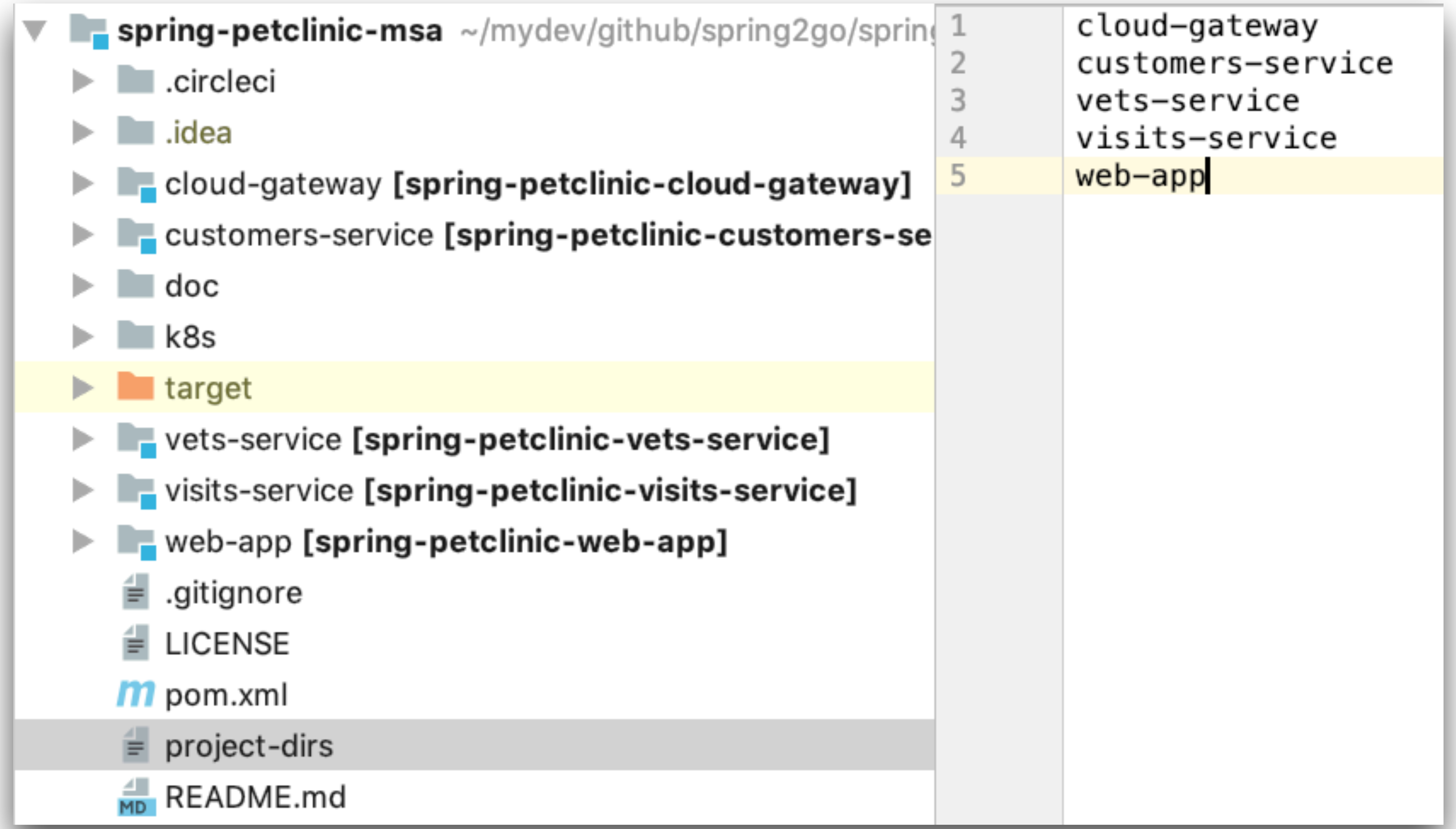
```yaml
jobs:
  build:
    docker:
      - image: circleci/openjdk:8u242-jdk
    steps:
      - checkout
      - run:
          name: Determine which directories have changed
          command: |
            git diff --no-commit-id --name-only -r `git log -n 2 --oneline --pretty=format:"%h" | tail -n1` | cut -d/ -f1 | sort -u >  projects
            printf "Modified directories:\n"
            cat projects
            while read project; do
              if grep -Fxq $project project-dirs; then
                printf "\nTriggerring build for project: "$project
                curl -s -u ${CIRCLE_TOKEN}: -d build_parameters[CIRCLE_JOB]=${project} https://circleci.com/api/v1.1/project/github/$CIRCLE_PROJECT_USERNAME/$CIRCLE_PROJECT_REPONAME/tree/$CIRCLE_BRANCH
              fi
            done < projects
  cloud-gateway:
    docker:
      - image: circleci/openjdk:8u242-jdk
    working_directory: ~/spring-petclinic-msa/cloud-gateway
    steps:
      - build-service:
          service-name: "cloud-gateway"
  customers-service:
    docker:
      - image: circleci/openjdk:8u242-jdk
    working_directory: ~/spring-petclinic-msa/customers-service
```

build-service command



```
commands:
  build-service:
    description: "Build a service and push image to dockerhub"
    parameters:
      service-name:
        type: string
    steps:
      # git pull
      - checkout:
          path: ~/spring-petclinic-msa

      - setup_remote_docker

      - run:
          name: Login to Dockerhub
          command: docker login -u $DOCKER_USER -p $DOCKER_PASS

      # Download and cache dependencies
      - restore_cache:
          keys:
            - << parameters.service-name >>-{{ checksum "pom.xml"}}

      - run: mvn dependency:go-offline

      - save_cache:
          paths:
            - ~/.m2
          key: << parameters.service-name >>-{{ checksum "pom.xml"}}

      # package into a jar and build image
      - run: mvn clean package -Ddocker.image.tag=Build-${CIRCLE_BUILD_NUM}-CI -Ddockerfile.maven.settings.auth=false

      # push docker image to docker hub
      - run: mvn dockerfile:push -Ddocker.image.tag=Build-${CIRCLE_BUILD_NUM}-CI

      # store raw content of src code
      - store_artifacts:
          path: target/classes
          destination: spring-petclinic-<< parameters.service-name >>
```

project-dirs文件



```
spring-petclinic-msa ~/mydev/github/spring2go/spring
    .circleci
    .idea
    cloud-gateway [spring-petclinic-cloud-gateway]
    customers-service [spring-petclinic-customers-se
    doc
    k8s
    target
    vets-service [spring-petclinic-vets-service]
    visits-service [spring-petclinic-visits-service]
    web-app [spring-petclinic-web-app]
    .gitignore
    LICENSE
    pom.xml
    project-dirs
    README.md
```

```
cloud-gateway
customers-service
vets-service
visits-service
web-app
```

本课小结

- 单体 vs 多仓库
  - 微服务可以采用单体仓库
  - 单体仓库优点：易于代码维护，标准化，组件共享，重构等，不足：代码复杂性和规模化问题
- 基于单体仓库的CI
  - 先检测哪个微服务发生了变更，再单独构建变更的微服务
  - 适用于其它CI系统如Jenkins