```yaml
# .gitlab-ci.yml, v1.0
# Alpha Nine Fabrication, Inc. - DevSecOps Pipeline
# Supports CMMC L2 / NIST 800-171-aligned controls
# Copyright Compliance Automation Engineering LLC
# All rights reserved. January 2, 2026

stages:
  - validate
  - build
  - test
  - security_static
  - security_software_supply_chain
  - security_dynamic
  - package
  - deploy_dev
  - deploy_stage
  - approve_prod
  - deploy_prod

# Include GitLab security scanners (adjust based on your license/edition)
include:
  - template: Security/SAST.gitlab-ci.yml
  - template: Security/Secret-Detection.gitlab-ci.yml
  - template: Security/Dependency-Scanning.gitlab-ci.yml
  - template: Security/Container-Scanning.gitlab-ci.yml
  - template: Security/DAST.gitlab-ci.yml
  - template: Security/IaC-Scanning.gitlab-ci.yml

variables:
  DOCKER_DRIVER: overlay2
  SECURE_IMAGE_REGISTRY: "$CI_REGISTRY_IMAGE"
  APP_IMAGE_TAG: "$CI_REGISTRY_IMAGE:$CI_COMMIT_SHA"

# ---------- VALIDATION ----------

lint:
  stage: validate
  image: python:3.12-slim
  script:
    - pip install ruff
    - ruff check .
  artifacts:
    when: always
    paths:
      - lint-report.txt
  rules:
    - if: '$CI_PIPELINE_SOURCE == "merge_request_event"'
    - if: '$CI_COMMIT_BRANCH == "main"'

# ---------- BUILD & UNIT TESTS ----------

build:
  stage: build
  image: docker:27
  services:
    - docker:27-dind
  script:
```

```
    - docker build -t "$APP_IMAGE_TAG" .
    - docker push "$APP_IMAGE_TAG"
  artifacts:
    when: always
    reports:
      dotenv: build.env
  rules:
    - if: '$CI_PIPELINE_SOURCE == "merge_request_event"'
    - if: '$CI_COMMIT_BRANCH == "main"'

unit_tests:
  stage: test
  image: python:3.12-slim
  script:
    - pip install -r requirements.txt
    - pytest --junitxml=tests-report.xml
  artifacts:
    when: always
    reports:
      junit: tests-report.xml
  needs: ["build"]
  rules:
    - if: '$CI_PIPELINE_SOURCE == "merge_request_event"'
    - if: '$CI_COMMIT_BRANCH == "main"'

# ---------- STATIC & REPO SECURITY ----------

# SAST, secret detection, dependency scanning, IaC scanning will be created
# automatically from the included templates. Here we ensure they block merges.

.sast:
  stage: security_static
  needs: ["build"]
  allow_failure: false

.secret_detection:
  stage: security_static
  allow_failure: false

.dependency_scanning:
  stage: security_software_supply_chain
  allow_failure: false

.iac_scanning:
  stage: security_software_supply_chain
  allow_failure: false

# ---------- CONTAINER SECURITY ----------

container_scanning:
  stage: security_software_supply_chain
  needs: ["build"]
  variables:
    CS_IMAGE: "$APP_IMAGE_TAG"
  allow_failure: false

# ---------- DYNAMIC / DAST ----------
```

```yaml
dast:
  stage: security_dynamic
  needs: ["deploy_dev"]
  variables:
    DAST_WEBSITE: "https://dev.example.anf.internal"
    DAST_FULL_SCAN_ENABLED: "true"
  allow_failure: false
  rules:
    - if: '$CI_COMMIT_BRANCH == "main"'
    - if: '$CI_COMMIT_TAG =~ /^release-.*$/'

# ---------- PACKAGING ----------

sbom:
  stage: package
  image: alpine:3.20
  script:
    - apk add syft
    - syft "$APP_IMAGE_TAG" -o cyclonedx-json > sbom.cdx.json
  artifacts:
    reports:
      cyclonedx: sbom.cdx.json
  needs: ["build", "dependency_scanning", "container_scanning"]

# ---------- DEPLOYMENTS ----------

deploy_dev:
  stage: deploy_dev
  image: alpine:3.20
  script:
    - ./deploy/dev_deploy.sh "$APP_IMAGE_TAG"
  environment:
    name: dev
    url: https://dev.example.anf.internal
  needs: ["build", "unit_tests"]
  rules:
    - if: '$CI_COMMIT_BRANCH != "main"'

deploy_stage:
  stage: deploy_stage
  image: alpine:3.20
  script:
    - ./deploy/stage_deploy.sh "$APP_IMAGE_TAG"
  environment:
    name: stage
    url: https://stage.example.anf.internal
  needs:
    - job: build
    - job: unit_tests
    - job: container_scanning
    - job: dependency_scanning
    - job: sast
    - job: secret_detection
  rules:
    - if: '$CI_COMMIT_BRANCH == "main"'
  when: on_success

# Manual approval gate for production (supports AC + CA + RA)
```

```yaml
approve_prod:
  stage: approve_prod
  script: echo "Security & change approvals recorded in GitLab."
  when: manual
  allow_failure: false
  rules:
    - if: '$CI_COMMIT_BRANCH == "main"'
  # In GitLab UI: require approvals from SECOPS + OIT via protected environment rules.

deploy_prod:
  stage: deploy_prod
  image: alpine:3.20
  script:
    - ./deploy/prod_deploy.sh "$APP_IMAGE_TAG"
  environment:
    name: prod
    url: https://prod.example.anf.internal
    deployment_tier: production
  needs:
    - job: deploy_stage
    - job: approve_prod
    - job: dast
  rules:
    - if: '$CI_COMMIT_BRANCH == "main"'
  when: on_success
```