AWS Certified DevOps Engineer - Professional Cheat Sheet

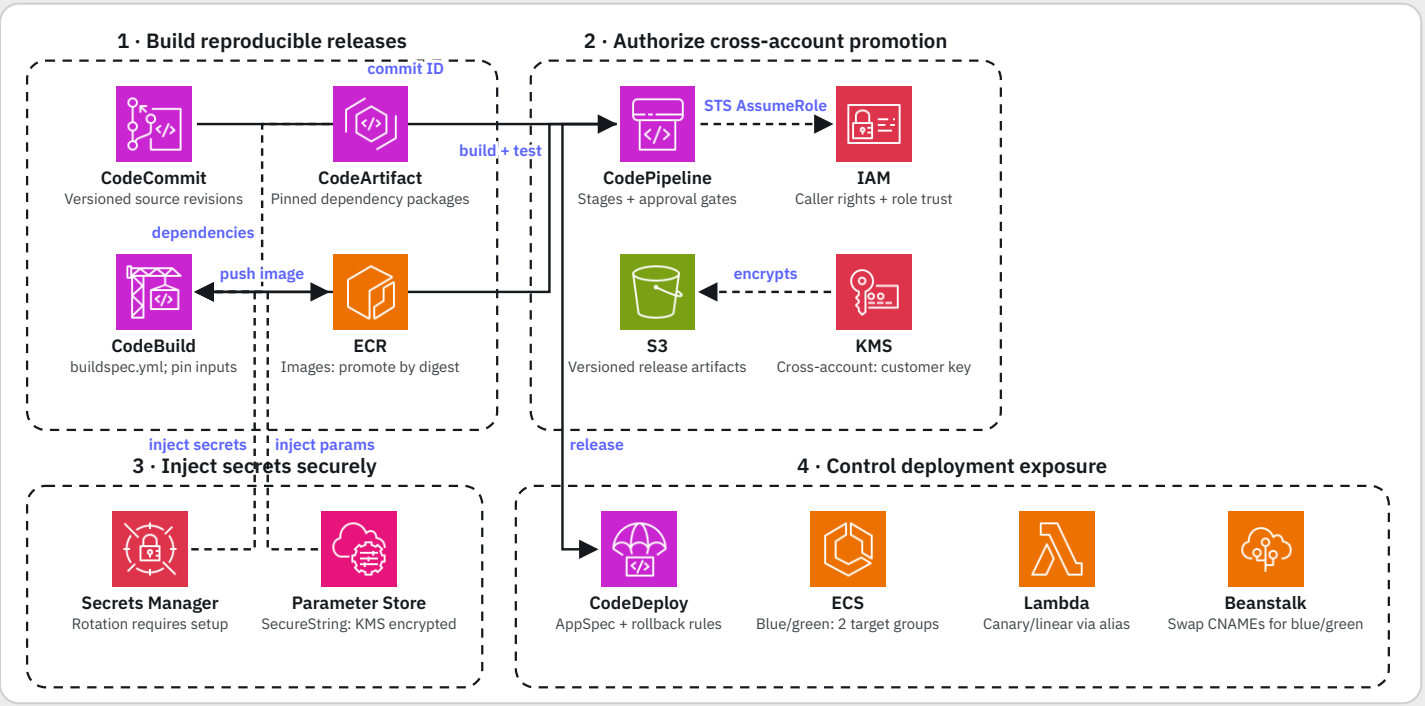
DOP-C02 · Domain 1 of 6

BlowStack

DOMAIN 1 · SDLC AUTOMATION · 22% OF THE EXAM · TASK STATEMENT 1.1

Implement CI/CD pipelines

Build, authorize, inject, deploy



1 Build reproducible releases

Reproducible builds pin toolchains and dependencies; buildspec commands run builds/tests and select output artifacts. Build once, then promote the same artifact through environments.

2 Authorize cross-account promotion

For **cross-account** actions, assume a target role. Caller permissions, role trust, S3 bucket policy, and customer-managed KMS key permissions must allow access. Separate approvers from deployers.

3 Inject secrets securely

Credential rotation must be configured; Parameter Store has no built-in automatic rotation. CodeBuild retrieves secrets using its service role; scope read and KMS decrypt permissions to required resources.

4 Control deployment exposure

Blue-green switches between environments for quick traffic rollback; canary tests a small traffic share first. Rolling replaces batches with mixed versions. CodeDeploy alarm-based rollback requires configuration.

Exam traps

- **Continuous delivery** can retain manual production approval; continuous deployment releases automatically. Agile and Waterfall govern lifecycle organization, not whether stages can be automated.
- **Gitflow** uses long-lived main/develop and release/hotfix branches; trunk-based development favors small, frequent merges. Branch separation does not provide account isolation.
- **Secret injection is not log protection**: printing secrets or packaging them into artifacts exposes them. Rotation does not update credentials already cached by a running consumer.

Get practice exams and lessons for AWS Certified DevOps Engineer - Professional

AWS Certified DevOps Engineer - Professional Cheat Sheet

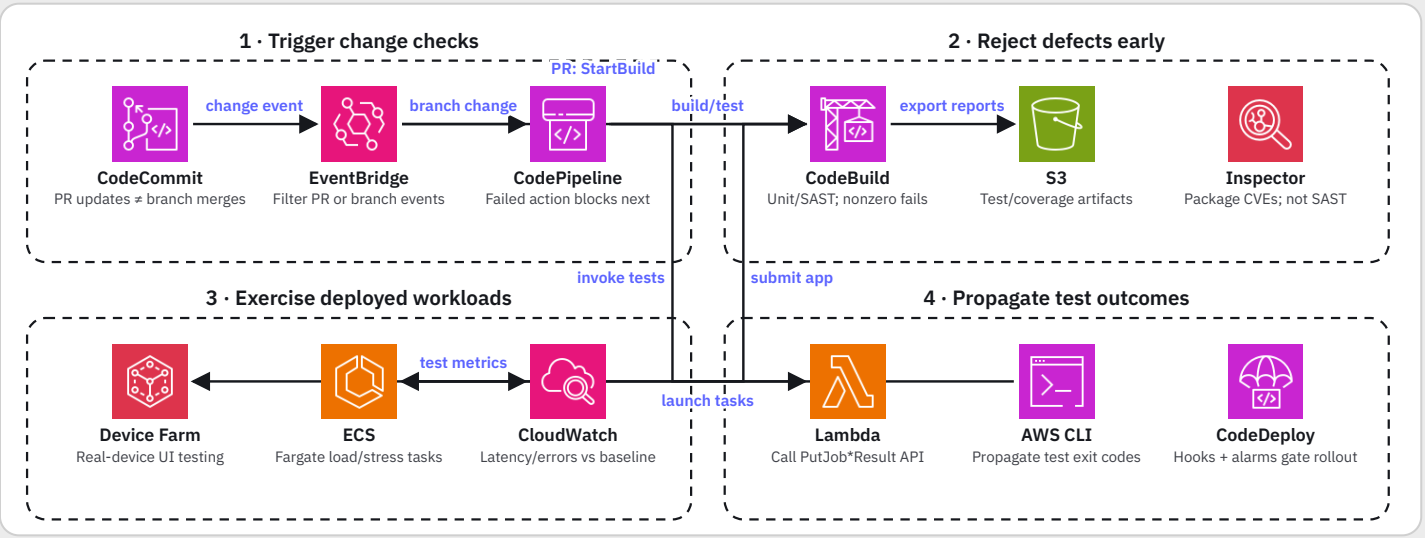
DOP-C02 · Domain 1 of 6

BlowStack

DOMAIN 1 · SDLC AUTOMATION · TASK STATEMENT 1.2

Integrate automated testing into CI/CD pipelines

Trigger, test, measure, gate



1 Trigger change checks

For **pull request checks**, match CodeCommit PR events in EventBridge and invoke CodeBuild. Merged branch changes can start CodePipeline. Supported external repositories can use webhook-triggered builds.

2 Reject defects early

Use **fail-fast gates**: many unit tests, fewer integration tests, few UI tests. CodeBuild runs test and SAST tools; commands must enforce coverage thresholds. Reports and vulnerability findings alone do not block promotion.

3 Exercise deployed workloads

Use **deployed tests** for integration and acceptance; Device Farm exercises device UI. ECS with Fargate runs load/stress generators. Compare CloudWatch measurements with benchmark limits through an explicit gate.

4 Propagate test outcomes

Explicit status gates connect tests to promotion: Lambda actions report success/failure to CodePipeline; build commands propagate nonzero exits. CodeDeploy validates lifecycle hooks and can stop or roll back on alarms.

Exam traps

- **Coverage is not correctness**: high coverage does not prove assertions are meaningful. Publishing coverage does not enforce a minimum; a threshold check must fail the build.
- **Inspector is not SAST**: vulnerability findings identify affected packages and supported workloads; source-code security analysis requires a SAST tool, often run in CodeBuild.
- **Invocation success is not test success**: Lambda must report the pipeline job result. An asynchronous test launch or zero exit from a launch command does not prove the tests passed.

Get practice exams and lessons for AWS Certified DevOps Engineer - Professional

AWS Certified DevOps Engineer - Professional Cheat Sheet

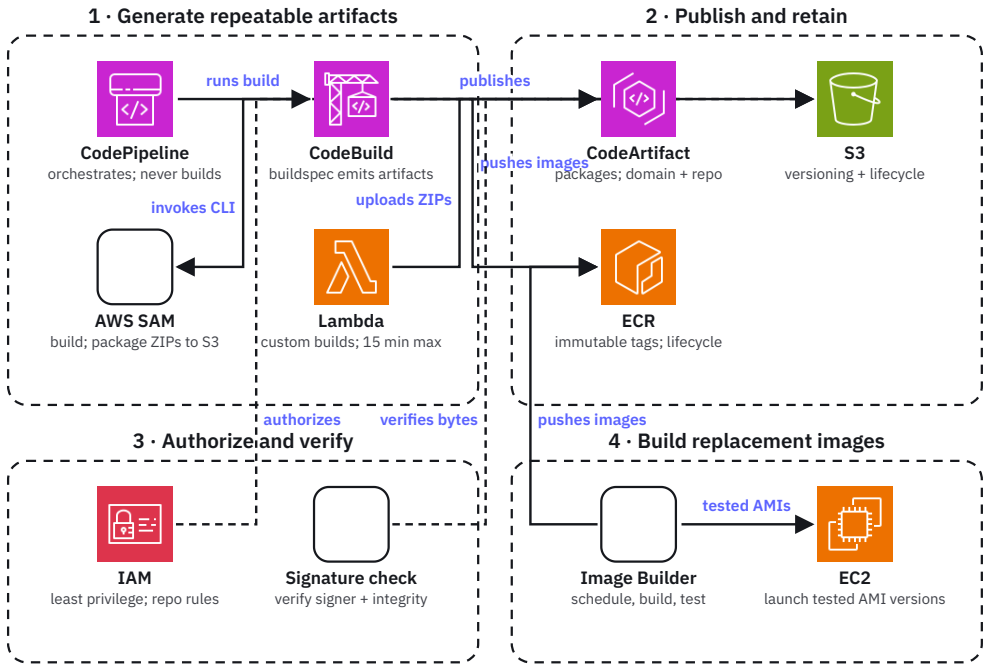
DOP-C02 · Domain 1 of 6

BlowStack

DOMAIN 1 · SDLC AUTOMATION · TASK STATEMENT 1.3

Build and manage artifacts

Build, publish, protect, replace



1 **Generate repeatable artifacts**

Use CodeBuild for builds; Lambda suits short custom packaging tasks. **Reproducible builds** require pinned dependencies, toolchains, and deterministic packaging. Version build definitions; promote the same tested bytes.

2 **Publish and retain**

Choose by **artifact type**: CodeArtifact for packages and upstream dependencies, S3 for ZIPs/templates, ECR for images. Configure S3 lifecycle and ECR expiration; automate CodeArtifact cleanup through APIs.

3 **Authorize and verify**

Use roles and repository policies for access; CodeArtifact requires authorization tokens. Encryption protects stored/transmitted bytes. **Digital signatures** require a trusted signer and explicit verification before use.

4 **Build replacement images**

Use **EC2 Image Builder** for scheduled AMI/container builds with versioned recipes and components. Distribute AMIs across accounts/Regions or push images to ECR. Deploy replacements rather than modifying running hosts.

Exam traps

- **Pinning is not reproducibility**: lockfiles alone do not control toolchains, timestamps, or packaging order. Floating image tags can change; deploy by digest for an exact image.
- **Versioning is not retention**: S3 current-version expiration can leave noncurrent versions stored. Configure noncurrent-version expiration separately; preserve releases required for rollback.
- **Immutable tags do not prevent deletion**: ECR lifecycle rules can expire images. Signatures prove integrity and signer identity, not vulnerability-free software; encryption does not verify provenance.

Get practice exams and lessons for AWS Certified DevOps Engineer - Professional

AWS Certified DevOps Engineer - Professional Cheat Sheet

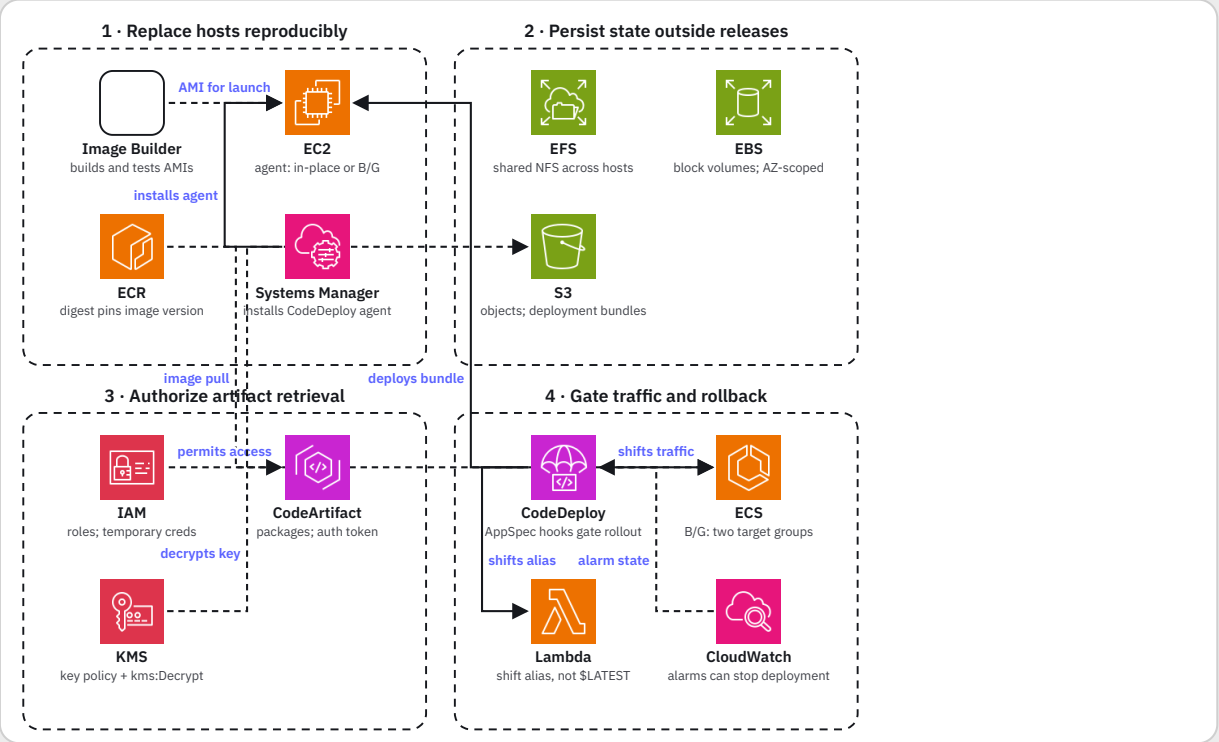
DOP-C02 · Domain 1 of 6

BlowStack

DOMAIN 1 · SDLC AUTOMATION · TASK STATEMENT 1.4

Implement deployment strategies for instance, container, and serverless environments

Build, authorize, persist, deploy



1 Replace hosts reproducibly

Immutable replaces hosts from tested images, reducing configuration drift; Image Builder does not deploy fleets. In-place deployment mutates hosts; idempotent hooks make retries safe.

2 Persist state outside releases

Shared files favor EFS; object access favors S3; disk-like access favors EBS. Externalized state survives replacement only when lifecycle settings retain it; an EBS volume cannot attach across AZs.

3 Authorize artifact retrieval

Cross-account repository access needs identity permissions plus repository policy, or an assumed role with trust and permissions. Least privilege scopes reads; SSE-KMS artifacts also require key authorization.

4 Gate traffic and rollback

Canary shifts a small share, then the remainder; linear shifts equal increments. ECS blue/green uses ELB health checks; Lambda uses published versions and an alias. Automatic rollback must be configured.

Exam traps

- **CodeDeploy agents** run on EC2, not ECS or Lambda. EKS uses Kubernetes deployment tooling and readiness probes; it is not a CodeDeploy target.
- **Rolling is not canary**: rolling replaces batches; canary weights traffic. CodeDeploy offers canary/linear for ECS and Lambda, not EC2; EC2 blue/green reroutes traffic to replacement instances.
- **Rollback is not data undo**: restoring code or redirecting traffic does not reverse schema changes. Retained artifacts, compatible data, and a healthy previous release are still required.

Get practice exams and lessons for AWS Certified DevOps Engineer - Professional

AWS Certified DevOps Engineer - Professional Cheat Sheet

DOP-C02 · Domain 1 of 6

BlowStack

DOMAIN 1 · SDLC AUTOMATION · WHAT TO REMEMBER

Concepts and services of SDLC Automation

Concepts to know

CI/CD

CI validates integrated changes through builds and tests; continuous delivery keeps releases deployable. **Promote the same tested artifact** across environments; continuous deployment also automates production release without manual approval.

Blue-Green Deployment

Blue/green validates a replacement environment before switching production traffic. Choose it for **fast traffic-based rollback**; rolling updates replace capacity incrementally, while canary releases expose a small traffic share first.

Cross-Account Access

Cross-account pipelines use assumed roles to cross account boundaries. **Check every authorization layer**: role trust, identity permissions, artifact bucket policies, and customer managed KMS key policies for encrypted artifacts.

Core services



AWS CodePipeline

Choose it to **orchestrate release stages**, including source, build, test, approval, and deployment. Cross-account actions use target-account roles and authorized access to encrypted artifacts.



AWS CodeBuild

Choose it to **run builds and automated tests** using buildspec.yml. Nonzero command exits can fail builds; test reports and coverage reports expose results but do not enforce every quality threshold.



AWS CodeDeploy

Choose it to **automate deployments and rollback**: EC2/on-premises uses an agent and lifecycle hooks; ECS uses blue/green; Lambda shifts alias traffic between published versions.

Get practice exams and lessons for AWS Certified DevOps Engineer - Professional