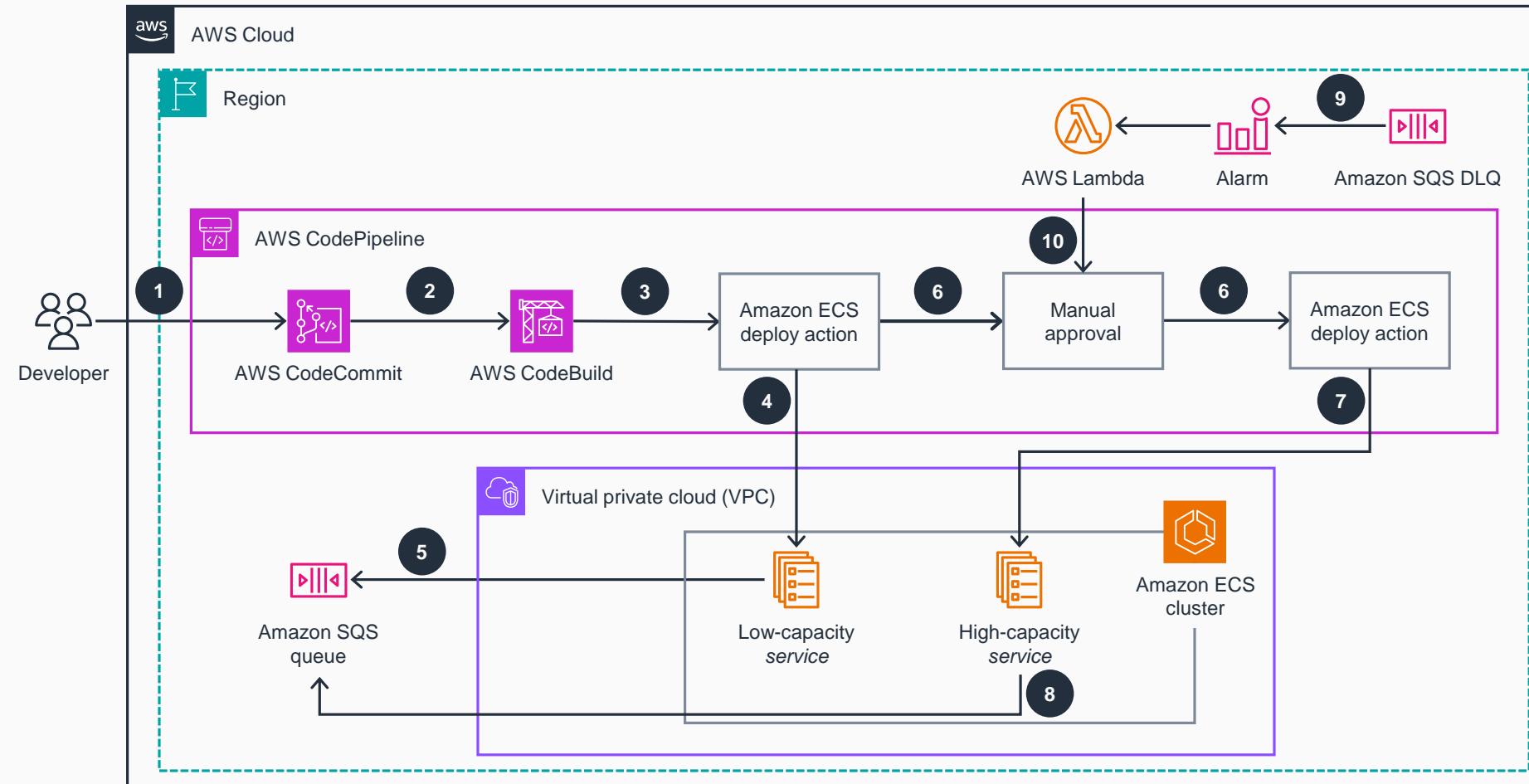


Guidance for Canary Deployments for Queue Processing Workloads in Amazon ECS

This architecture diagram shows how to implement canary deployments for backend or queue processing workloads in Amazon Elastic Container Service (Amazon ECS) without a load balancer.



- 1 Submit any code changes to software configuration management tools like **AWS CodeCommit**.
- 2 **AWS CodePipeline** watches for new code changes and initiates the continuous integration and continuous delivery (CI/CD) pipeline to build the new container image using **AWS CodeBuild**.
- 3 After the image is built, **CodePipeline** initiates the **Amazon Elastic Container Service (Amazon ECS)** deploy action.
- 4 The **Amazon ECS** deploy action deploys the change to a low-capacity **Amazon ECS** service instance to start the canary process and waits for manual approval.
- 5 The low-capacity **Amazon ECS** service instance will start processing messages from the **Amazon Simple Queue Service (Amazon SQS)** queue using the new version, while the high-capacity service instance is still using the existing version.
- 6 Once the changes are validated, your team can manually approve the canary release to propagate the change to the high-capacity **Amazon ECS** service instance.
- 7 The **Amazon ECS** deploy action deploys the change to the high-capacity **Amazon ECS** service instance to complete the deployment process.
- 8 The new version is deployed to both the low-capacity and the high-capacity **Amazon ECS** service instance to process messages from the **Amazon SQS** queue.
- 9 Failures in processing **Amazon SQS** messages are sent to the **Amazon SQS** dead letter queue (DLQ) so that you can monitor issues from the new code version.
- 10 An **Amazon CloudWatch** alarm monitors the **Amazon SQS** DLQ depth and can invoke an **AWS Lambda** function to stop and roll back the canary release.

