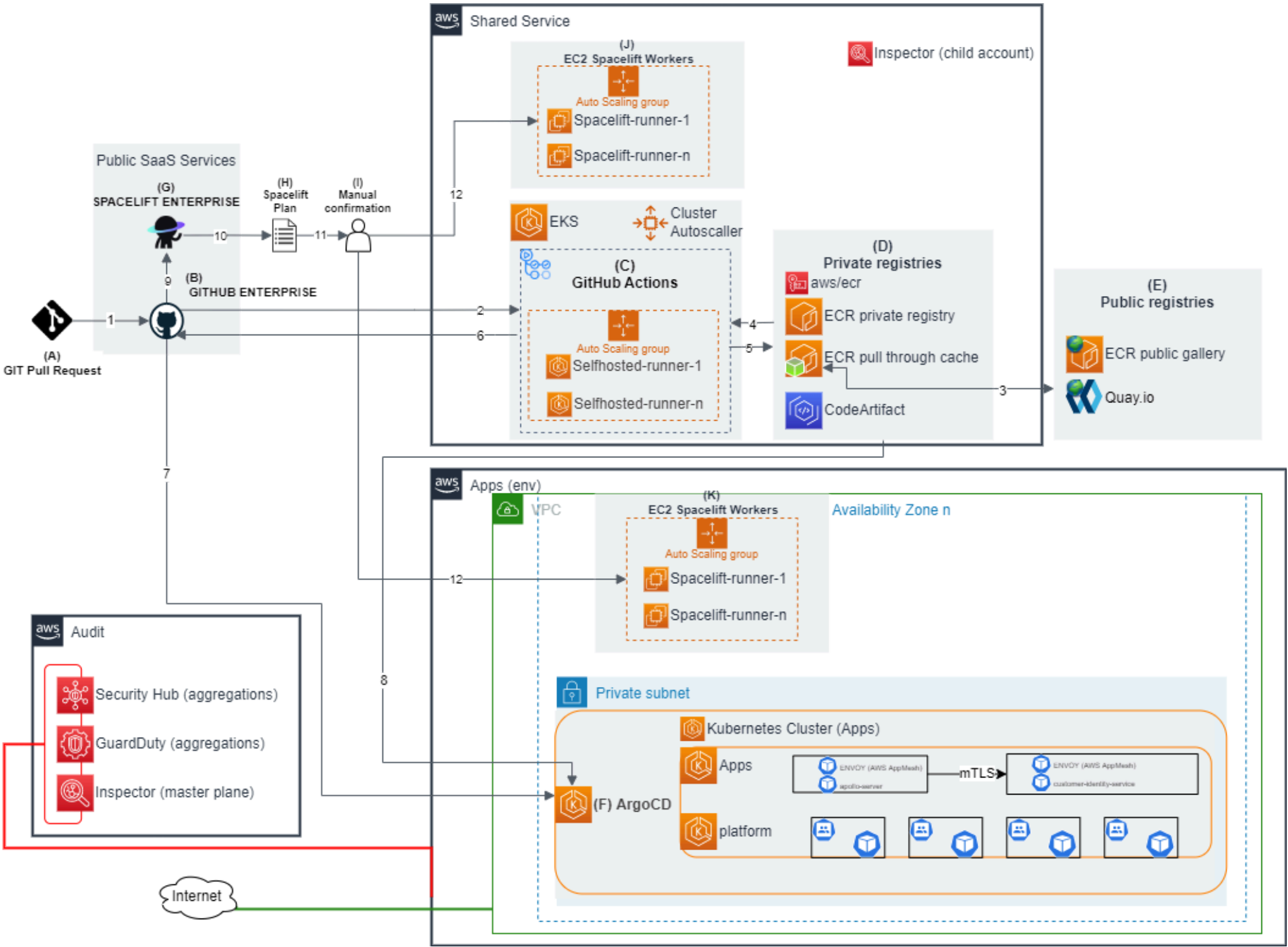


Architecture_for_CI_and_CD

Last edited by [czerniga-gft](#) 8 months ago

Architecture for CI and CD

- [Diagram of Architecture for CI and CD](#)
- [Relevant elements of architecture](#)
- [Description of architecture/flow for CI libraries](#)
- [Description of architecture/flow for CI dockers](#)
- [Description of architecture/flow for CD](#)
- [Description of architecture for Spacelift Delivery](#)
- Diagram of Architecture for CI and CD**



Relevant elements of architecture

- A** – Pull Request that was reviewed by team members
- B** – GitHub Enterprise remote repository
- C** – Continuous Integration pipeline (GitHub Actions)
- D** – Private Registers ((**1**) ECR private registry, (**2**) ECR pull through cache, (**3**) CodeArtifact). ECR performs scanning images triggered by push
- E** – Public Registers ((**1**) ECR public gallery, (**2**) [Quay.io](#))
- F** – ArgoCD Continuous Delivery
- G** – Spacelift Enterprise Terraform Management CI/CD
- **H ****– Spacelift Plan
- I** – Manual confirmation
- J** – Spacelift Workers in Shared Services
- K** – Spacelift Workers in Apps

• Description of architecture/flow for CI libraries

- 1 – Pushing reviewed commit **(A)** to GitHub Enterprise remote repository **(B)**
- 2 – Changes on the repository **(B)** trigger Continuous Integration Pipeline **(C)**
- 4 – Fetching app dependencies from Code Artifact **(D3)** to CI Pipeline **(C)**
- 5 – Storing Built in CI Pipeline **(C)** Software Packages in CodeArtifact **(D3)**
- 6 – Sending information about new builds back to the GitHub enterprise repository **(B)**

• Description of architecture/flow for CI dockers

- 1 – Pushing reviewed commit **(A)** to GitHub Enterprise remote repository **(B)**
- 2 – Changes on the repository **(B)** trigger Continuous Integration Pipeline **(C)**
- 3 – Caching images from public registries **(E1, E2)** to ECR pull through cashe **(D2)**
- 5 – Storing Built in CI Pipeline **(C)** Docker images in ECR private registry **(D1)**
- 6 – Sending information about new builds back to the GitHub enterprise repository **(B)**

• Description of architecture/flow for CD

- 7 – Triggering Argo CD **(F)** Continuous Delivery (which is watching for information about new builds in linked GitHub repository **(B)**)
- 8 – Fetching new built docker images stored in ECR private registry **(D1)** to EKS by Argo CD **(F)**

• Description of architecture for Spacelift Delivery

- 1 – Pushing reviewed commit **(A)** to GitHub Enterprise remote repository **(B)**
- 9 – Triggerering Spacelift Enterprise **(G)** by GitHub Enterprise **(B)**
- 10 – Creating Spacelift Plan **(H)**
- 11 – Automatic plan **(H)** calcutation and waiting for human manual confirmation **(I)**
- 12,13 - Applying changes to Spacelift workers in Shared services **(J)** and Spacelift workers in Apps **(K)**

Attachments:

■ [image2022-10-21_10-47-23.png](#) (image/png) ■ [image2022-10-21_10-51-22.png](#) (image/png) ■ [Capture2.PNG](#) (image/png) ■ [image2022-10-25_14-25-Z.png](#) (image/png) ■ [Architecture for CI-CD.drawio](#) (application/octet-stream) ■ [Architecture for CICD.PNG](#) (image/png) ■ [Architecture for CI-CD.drawio.png](#) (image/png)

Comments:

Wiechno, Dawid please attach here source of draw.io (as page attachment)

🗨️ Posted by jwsz at Oct 28, 2022 09:47