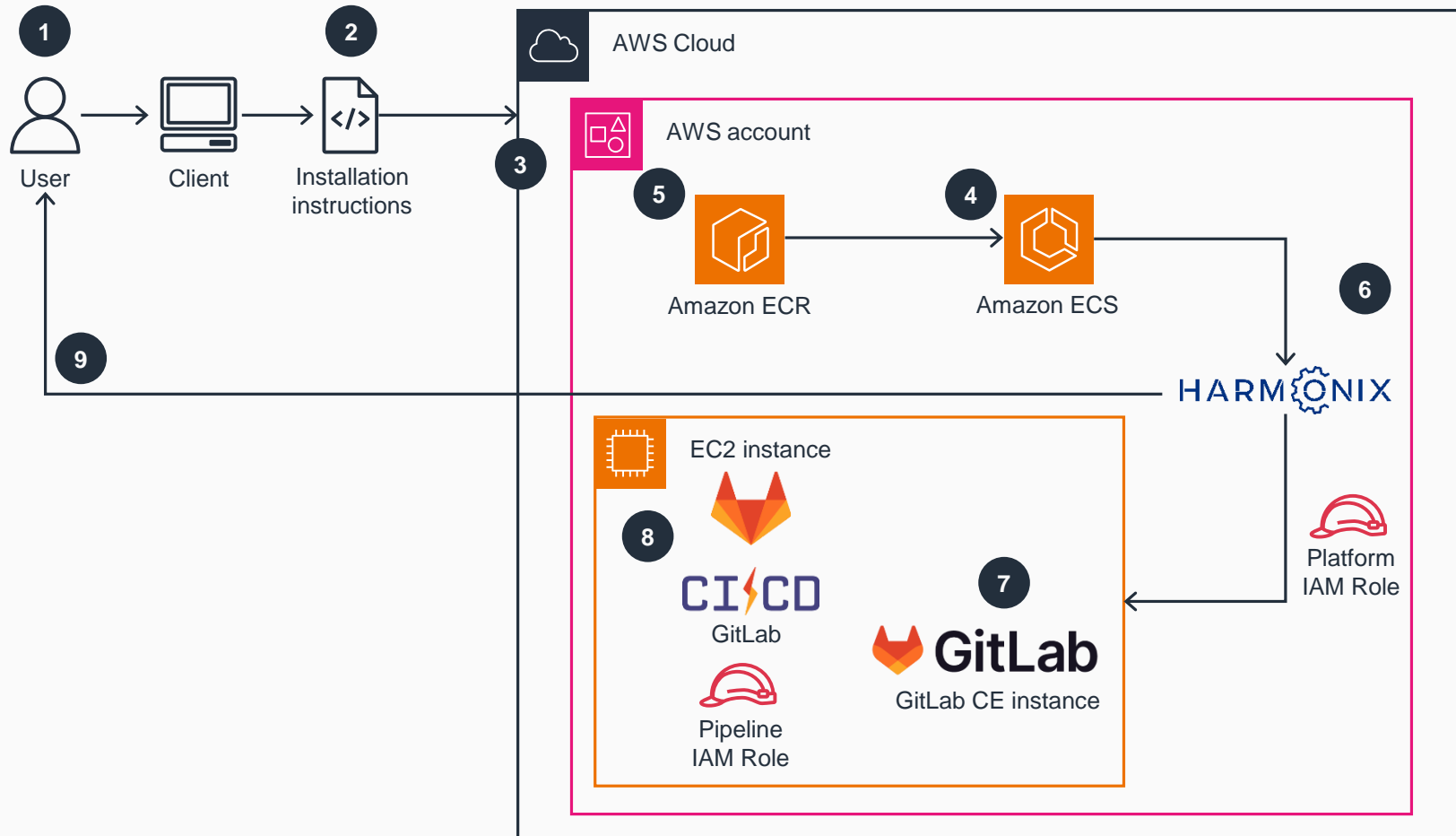


Guidance for Deploying Harmonix on AWS

Provisioning the AWS account

This architecture diagram illustrates how to provision an AWS account after cloning Harmonix from GitHub.



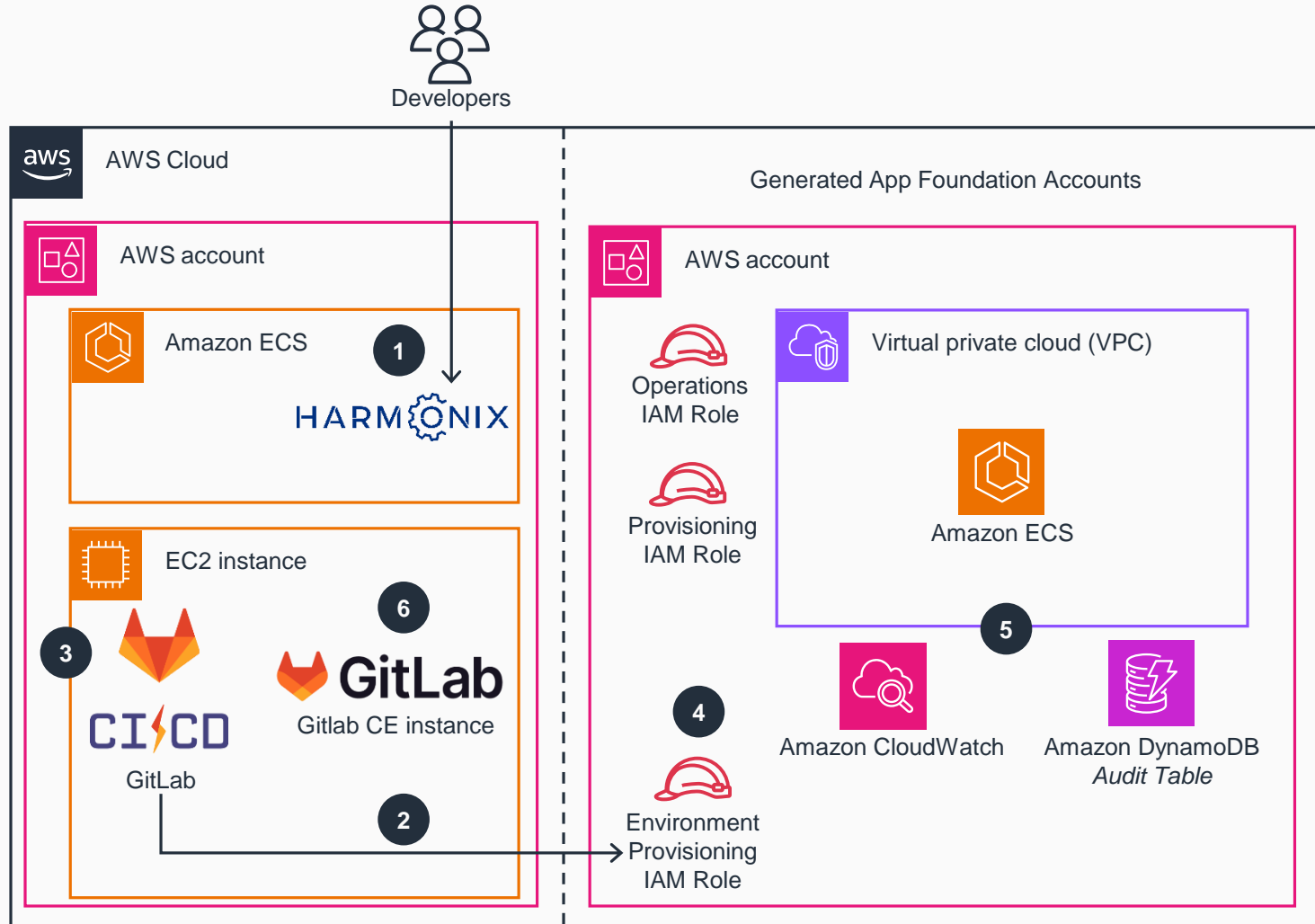
- 1 Clone the Harmonix on AWS repository from GitHub.
- 2 Follow the Installation instructions from the Harmonix website, and configure your `.env` file.
- 3 Deploy Harmonix to your AWS account.
- 4 **Amazon Elastic Container Service (Amazon ECS)** creates a service to run the Harmonix application.
- 5 **Amazon Elastic Container Registry (Amazon ECR)** provides the latest Harmonix custom container image to **Amazon ECS**.
- 6 The Backstage application starts running on an **Amazon ECS** task, connecting to a GitLab Community Edition (CE) instance to pull all the platform templates. The task has a reserved **AWS Identity and Access Management (IAM)** identity role, *Platform Role*.
- 7 GitLab CE runs on **Amazon Elastic Compute Cloud (Amazon EC2)** with pre-backed templates and an identity token for Harmonix on AWS.
- 8 A GitLab runner instance registers itself with a GitLab instance and is ready to execute pipelines and jobs. The instance also has a reserved **IAM** identity, *Pipeline Role*.
- 9 The Harmonix on AWS environment is now accessible through the browser. Follow the Application Environment Tutorial from the Harmonix website to start provisioning resources on AWS.



Guidance for Deploying Harmonix on AWS

Provisioning an Amazon ECS environment

This architecture diagram shows how to provision an Amazon ECS environment using Harmonix on AWS.



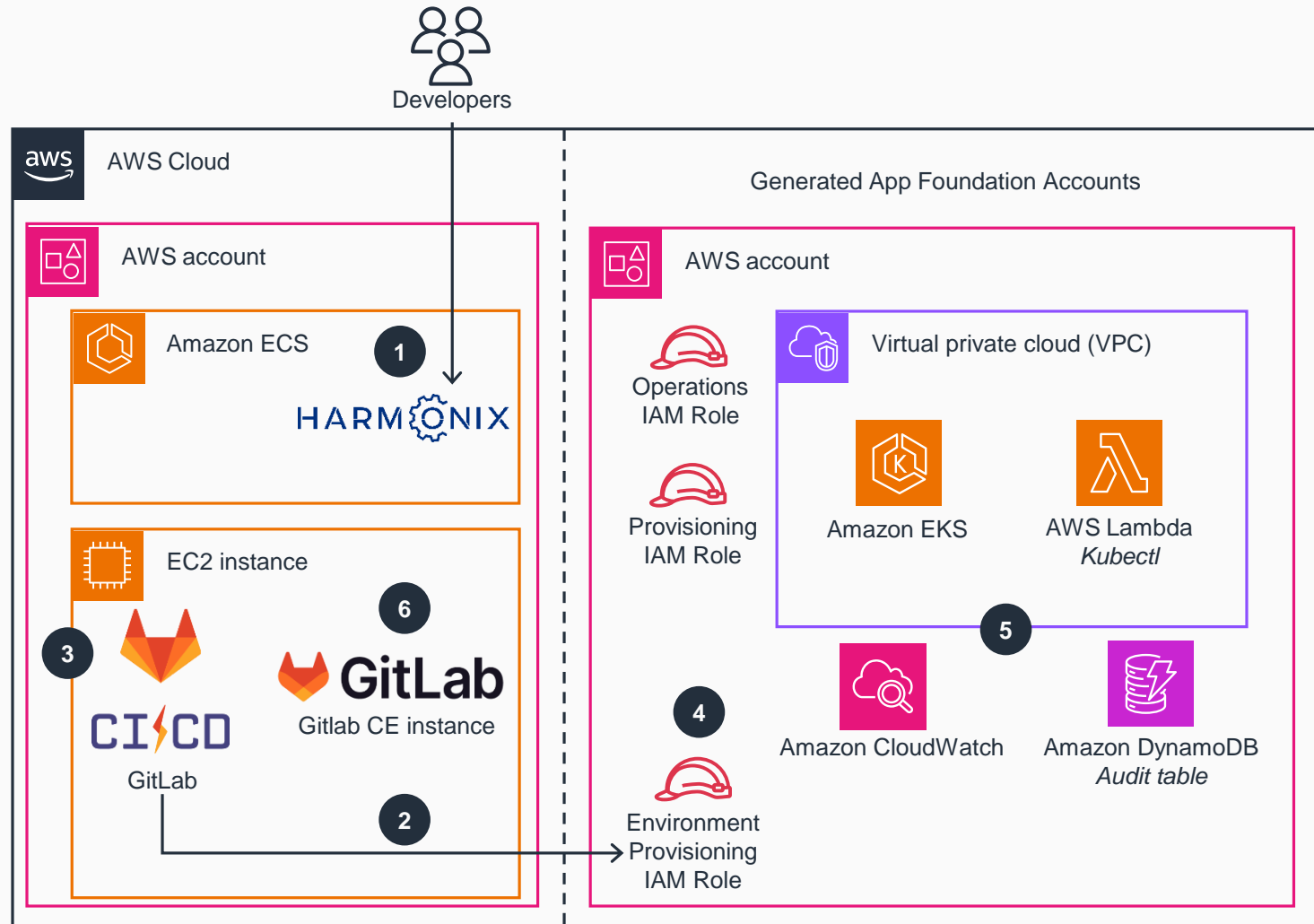
- 1 The developer uses Harmonix on AWS to create an **Amazon ECS** environment using the *ECS Environment provider template*.
- 2 Once submitted, Harmonix on AWS creates a new repository on GitLab with the desired environment provider information, including the AWS account, Region, and additional configuration.
- 3 The commit to GitLab also includes a continuous integration/continuous development (CI/CD) pipeline, which automatically triggers and invokes GitLab runner to start executing the deployment of the request environment to the target account.
- 4 The GitLab runner first assumes a target role with sufficient credentials to deploy the desired environment. Using these credentials, the pipeline deploys all the environment resources using infrastructure as code (IaC).
- 5 The pipeline provisions resources, including **Amazon Virtual Private Cloud (Amazon VPC)**, an **Amazon Elastic Kubernetes Service (Amazon EKS)** cluster, **IAM** roles for operations and provisioning, an **Amazon CloudWatch** log group, and an **Amazon DynamoDB** audit table.
- 6 The last step of the pipeline is to update the GitLab repository with the created resource in the catalog-info.yaml file. This file is used by Backstage.io to visualize the environment entity and provide additional actions. Developers can now create apps and resources on the newly created environment. Review the Create Apps tutorial from the Harmonix website for additional information.



Guidance for Deploying Harmonix on AWS

Provisioning an Amazon EKS environment

This architecture diagram shows how to provision an Amazon EKS environment using Harmonix on AWS.



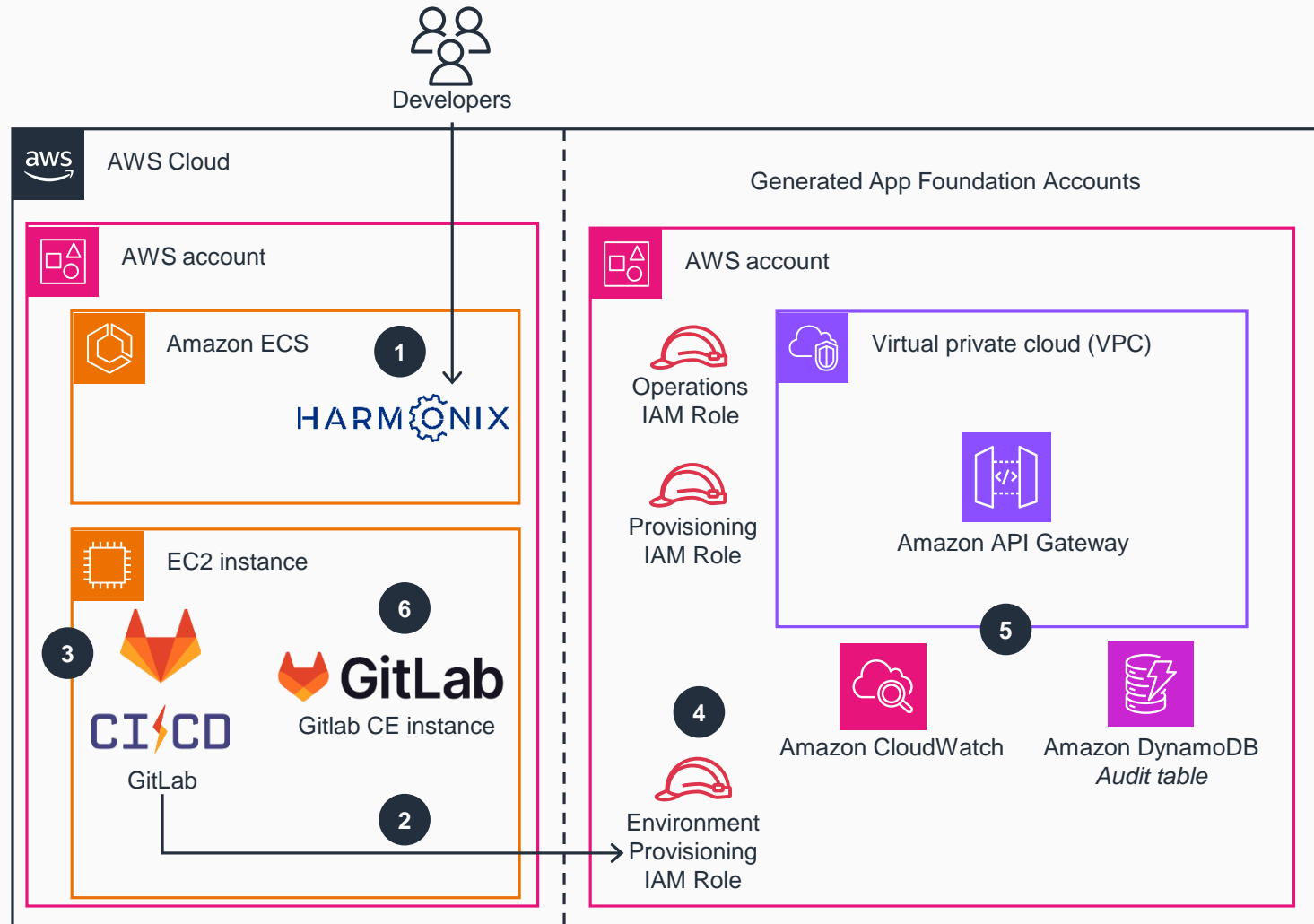
- 1 The developer uses Harmonix on AWS to create an **Amazon EKS** environment using the *EKS Environment provider template*.
- 2 Once submitted, Harmonix on AWS creates a new repository on GitLab with the desired environment provider information, including the AWS account, Region, and additional configuration.
- 3 The commit to GitLab also includes a continuous integration/continuous development (CI/CD) pipeline, which automatically triggers and invokes GitLab runner to start executing the deployment of the request environment to the target account.
- 4 The GitLab runner first assumes a target role with sufficient credentials to deploy the desired environment. Using these credentials, the pipeline deploys all the environment resources using infrastructure as code (IaC).
- 5 The pipeline provisions resources, including **Amazon VPC**, an **EKS** cluster, **IAM** roles for operations and provisioning, a **CloudWatch** log group, and an **DynamoDB** audit table. Additionally, the pipeline provisions a kubectrl **AWS Lambda** function to access private **EKS** clusters which do not have an accessible control plane.
- 6 The last step of the pipeline is to update the GitLab repository with the created resource in the catalog-info.yaml file. This file is used by Backstage.io to visualize the environment entity and provide additional actions. Developers can now create apps and resources on the newly created environment. Review the Create Apps tutorial from the Harmonix website for additional information.



Guidance for Deploying Harmonix on AWS

Provisioning a serverless environment

This architecture diagram shows how to provision a serverless environment with Amazon API Gateway using Harmonix on AWS.



- 1 The developer uses Harmonix on AWS to create a serverless environment using the *serverless provider template*.
- 2 Once submitted, Harmonix on AWS creates a new repository on GitLab with the desired environment provider information, including the AWS account, Region, and additional configuration.
- 3 The commit to GitLab also includes a continuous integration/continuous development (CI/CD) pipeline, which automatically triggers and invokes GitLab runner to start executing the deployment of the request environment to the target account.
- 4 The GitLab runner first assumes a target role with sufficient credentials to deploy the desired environment. Using these credentials, the pipeline deploys all the environment resources using infrastructure as code (IaC).
- 5 The pipeline provisions resources, including **Amazon VPC**, **Amazon API Gateway**, **IAM** roles for operations and provisioning, a **CloudWatch** log group, and an **DynamoDB** audit table.
- 6 The last step of the pipeline is to update the GitLab repository with the created resource in the `catalog-info.yaml` file. This file is used by Backstage.io to visualize the environment entity and provide additional actions. Developers can now create apps and resources on the newly created environment. Review the Create Apps tutorial from the Harmonix website for additional information.

