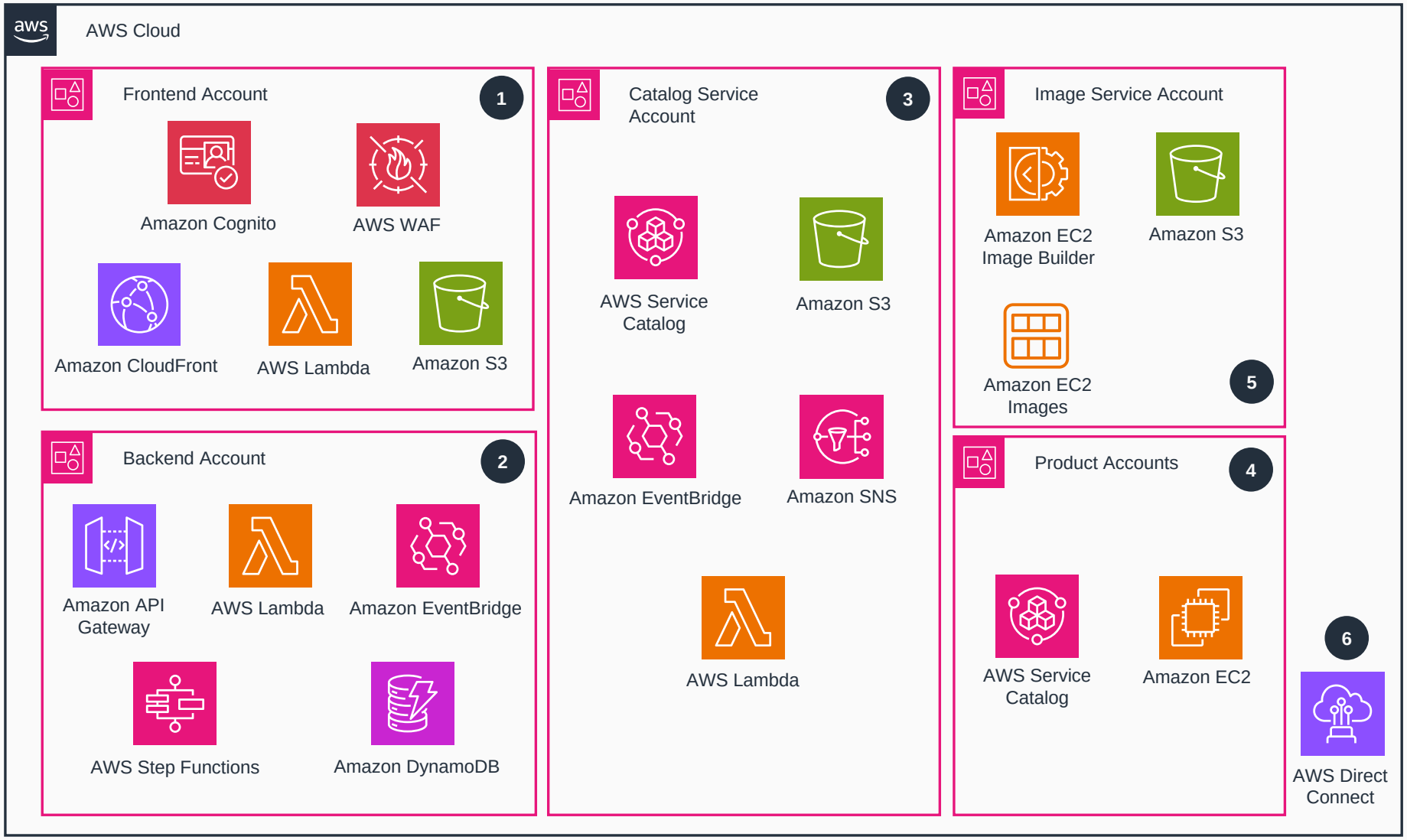
Guidance for Deploying Virtual Engineering Workbench on AWS

This architecture diagram illustrates the core AWS services and AWS account structure required to deploy VEW.



1. Connect to the VEW frontend through **Amazon CloudFront** backed by **Amazon Simple Storage Service (Amazon S3)**, and select the workbench and virtual targets with which you'll work. Manage user authentication with **Amazon Cognito** and secure access with **AWS WAF**.

2. **Amazon API Gateway** and **AWS Lambda** handle frontend requests and trigger workbench and virtual target provisioning through **Amazon EventBridge** and **AWS Step Functions**. Store product metadata in **Amazon DynamoDB**.

3. VEW manages its products (digital toolchains and virtual targets) in **AWS Service Catalog**. Store product dependencies like binaries or libraries in **Amazon S3**, and handle events using **EventBridge**, **Amazon Simple Notification Service (Amazon SNS)**, and **Lambda**.

4. VEW deploys workbenches and virtual targets as **Amazon Elastic Compute Cloud (Amazon EC2)** instances into product accounts using **Service Catalog**, from which they can be accessed by the user through **Amazon DCV**, remote desktop protocol (RDP), or SSH.

5. Deploy a fully automated VEW product lifecycle pipeline using **Amazon EC2** ImageBuilder, creating **Amazon EC2** AMIs and storing assets on **Amazon S3**.

6. With **AWS Direct Connect**, VEW can be integrated with customers' corporate resources like identity providers, license servers, or code repositories.

Reviewed for technical accuracy October 25, 2024

AWS Reference Architecture