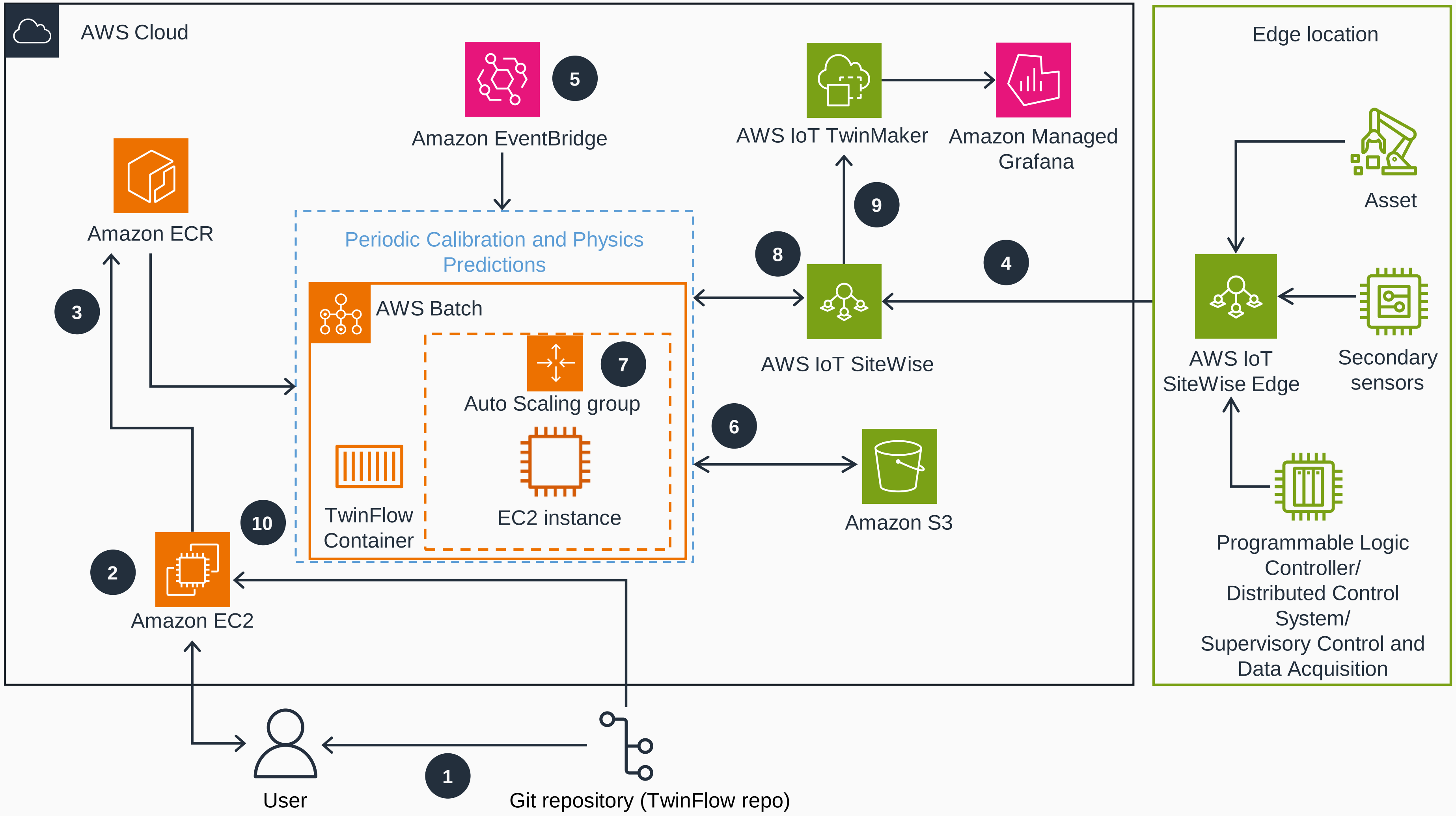
Guidance for Self-Calibrating Level 4 Digital Twins on AWS

This architecture diagram demonstrates how to probabilistically calibrate a physics-based digital twin with IoT data to improve predictions and enable a digital twin to adapt to changing environmental conditions.



1 Download **TwinFlow from GitHub**, and install it on your **Amazon Elastic Compute Cloud (Amazon EC2)** instance.

2 TwinFlow can orchestrate any number of containers. For this example, modify the example containers for their specific application, and embed a digital twin inside the container. The example **TwinFlow** containers use probabilistic methods to calibrate the digital twin.

3 Build and upload the container to **Amazon Elastic Container Registry (Amazon ECR)** to enable using the container in the AWS Cloud**.**

4 Ingest the sensor telemetry timeseries data from an edge location to **AWS IoT SiteWise** using **AWS IoT SiteWise Edge**.

5 Using **Amazon EventBridge** scheduler, periodically deploy an **Amazon EC2** instance in **AWS Batch**, which loads the **TwinFlow** container and application customized code.

6 **TwinFlow** container reads the sensor telemetry timeseries data from **AWS IoT SiteWise**, calibrates the digital twin, and stores the calibration results in an **Amazon Simple Storage Service (Amazon S3)** bucket.

7 Using an autoscaling **EC2** instance in an **AWS Batch** compute environment, use the calibrated digital twin to make physics predictions.

8 Upload the physics predictions into **AWS IoT SiteWise**, enabling downstream consumption.

9 Monitor both physical and virtual sensor data in **Amazon Managed Grafana** with **AWS IoT TwinMaker.**

10 Stop your initial **EC2** instance once it is no longer needed.

Reviewed for technical accuracy November 16, 2023

AWS Reference Architecture