**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **31/07/2020** | **Name:** | **Gaganashree P** |
| **Course:** | **Coursera** | **USN:** | **4AL15EC024** |
| **Topic:** | Industrial IoT on Google Cloud Platform | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Gaganashree-P** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |
| **REPORT**  In this lesson you learn about Cloud Dataflow. Cloud Dataflow is a fully managed service for transforming and enriching data in stream (real time) or batch (historical) modes. This means you don't have to do complex workarounds or compromise your pipeline design. Cloud Dataflow uses a serverless approach to resource provisioning and management. You can access virtually limitless capacity to solve your biggest data processing challenges, while paying only for what you use.    Dataflow pipelines are either batch (processing bounded input like a file or database table) or streaming (processing unbounded input from a source like Cloud Pub/Sub). The lab you do in this lesson is a batch pipeline. In a later module you create a streaming pipeline.  **Ingest, manage and optimize your IoT device data securely**    [Cloud IoT Core](https://cloud.google.com/iot-core/) is a fully managed service designed to:   * Help connect, manage, and ingest data from globally dispersed devices. * Easily and securely ingest event streams from anywhere, at any scale, for simple, reliable, real-time stream analytics. * Seamlessly move IoT data across Google Cloud services. * Ingest data with Cloud IoT Core and distribute data with [Cloud Pub/Sub](https://cloud.google.com/pubsub/docs/overview).   [Cloud IoT Edge](https://cloud.google.com/iot-core/) extends Google Cloud’s powerful data processing and machine learning to billions of edge devices, such as robotic arms, wind turbines, and oil rigs, so they can act on the data from their sensors in real time and predict outcomes locally. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **31/07/2020** | **Name:** | **Gaganashree P** | |
| **Course:** | **Salesforce** | **USN:** | **4AL15EC024** | |
| **Topic:** | **Salesforce platform basics** | **Semester& Section:** | **8th A** | |
| **AFTERNOON SESSION DETAILS**  **s2.PNG** | | | |
| **REPORT**  **Overview of Objects**  DreamHouse is a realty company that provides a way for customers to shop for homes and contact real estate agents online. DreamHouse brokers use some of Salesforce’s standard functionality, like contacts and leads, to track home buyers. But when it comes to selling houses, there are a lot more things they want to track. For example, Salesforce doesn’t include a standard way to track properties. How is DreamHouse supposed to know which homes they have for sale or how much each home costs?  **Create a Custom Object**  Let’s work alongside D’Angelo to see how he builds the Property object. We need this object later, so don’t skip these steps!   1. Scroll to the bottom of this page. 2. Click the arrow next to Launch and select **Create a Trailhead Playground**. Don’t skip this step! You need to use a fresh and clean Trailhead Playground for this module. 3. Once your playground is created (it takes a minute!), press **Launch**. 4. Click the gear icon The setup gear. at the top of the page and launch setup. 5. Click the **Object Manager** tab. 6. Click **Create** | **Custom Object** in the top-right corner. 7. For Label, enter Property. Notice that the Object Name and Record Name fields auto-fill. 8. For Plural Label, enter Properties. 9. Check the box for **Launch New Custom Tab Wizard after saving this custom object**. 10. Leave the rest of the values as default and click **Save**. 11. On the New Custom Object Tab page, click the Tab Style field and select a style you like. The style sets the icon to display in the UI for the object. 12. Click **Next**, **Next**, and **Save**.  **Lightning Platform** Lightning Platform (also known as Force.com) is a [platform as a service](https://en.wikipedia.org/wiki/Platform_as_a_service) (PaaS) that allows developers to create add-on applications that integrate into the main Salesforce.com application. These third-party applications are hosted on Salesforce.com's infrastructure. Force.com applications are built using declarative tools, backed by Lightning and Apex, a proprietary [Java](https://en.wikipedia.org/wiki/Java_(programming_language))-like programming language for Force.com,  as well as and Visualforce, a framework including an XML syntax typically used to generate [HTML](https://en.wikipedia.org/wiki/HTML). The Force.com platform typically receives three complete releases a year. As the platform is provided as a service to its developers, every single development instance also receives all these updates.  In the Spring 2015 release a new framework for building user interfaces – Lightning Components – was introduced in beta. Lightning components are built using the open-source Aura Framework but with support for Apex as the server-side language instead of Aura's [Javascript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript) dependency. This has been described as an alternative to, not necessarily a replacement for, Visualforce pages. According to a September 2009 [Gartner Group](https://en.wikipedia.org/wiki/Gartner) report, Force.com had over 1,000 customer accounts. As of 2013, the Force.com platform has 1.4 million registered developers. Lightning Base Components is the component library built on top of Lightning Web Components. **Community Cloud** Community Cloud provides Salesforce customers the ability to create online web properties for external collaboration, customer service, channel sales, and other custom portals in their instance of Salesforce. Tightly integrated to Sales Cloud, Service Cloud, and App Cloud, Community Cloud can be quickly customized to provide a wide variety of web properties. Community Cloud combines the functionality of the former Salesforce Customer and Partner Portals with some additional features. | | | |