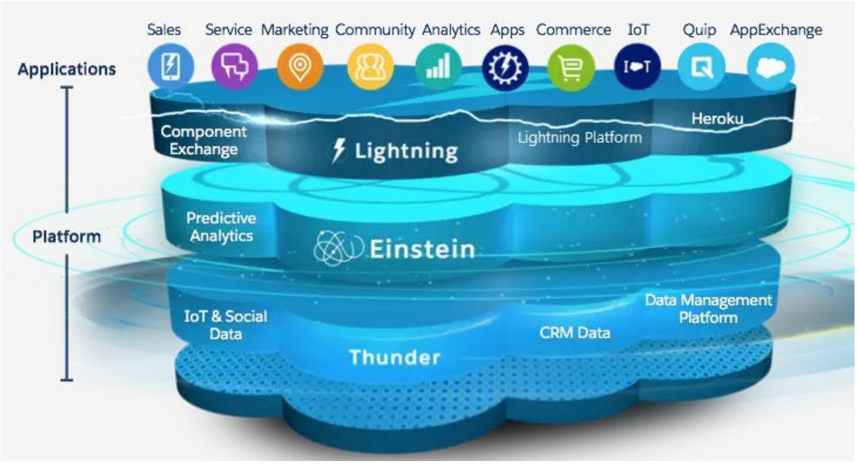
Salesforce Platform Basics-

- More than a CRM (Customer Relationship Manager) , it is an entire platform that has many built-in features.
- The platform allows you to modify or build entire apps on top of what Salesforce already provides.
- Org = business's instance of Salesforce. Apps, Objects = tables, Records = rows on table, Fields = columns in tables
- TP) Trailhead Playground = Salesforce's Dev Edition, (safe dev env)
  - To perform challenges:
    - Read all requirements
    - Watch your spelling exactly since testing is automated
- Declarative development = developing without code.
- Programmatic development = the use of components, code, and pages
- Initial steps to make edits seems to be clicking the gear and selecting Setup. Though you can also edit an object or page from the same menu.
- Within Setup, Object(Table) Manager tab allows you to edit your data entry fields. Aka- view and customize your org's objects.
- Every custom object created gets **Chatter feed tracking** = allows multiple ppl to comment on a record (entry/row).
  - Ideal use- bring all comms on a record to one place

Flags for potential high impact, low effort opportunities:

- Heavy email collab
- Reliance on spreadsheets
- Shared local docs
- Time-intensive repetitive steps
- !!Impact on only a few departments (this one is not for resulting in high impact, but while in learning stages it means limit stakeholders and be more likely to occur)

Architecture:



Architecture cont'd:

*Salesforce is a cloud company. All their offerings/apps are referred to as 'clouds' (sales 'cloud', commerce 'cloud', etc). They all integrate with each other out of the box via the platform

- Platform = foundation of all applications. Powered by metadata (Thunder) and AI (Einstein). The apps integrate with these and each other via the (Lightning) framework that includes APIs, and shared platforms.

Key terms of Architecture: Trust, MetaData, API, and Multi-tenancy

Trust- Salesforce is responsible for your org's data and functionality of what you build on top of their platform. *Trust.salesforce.com = performance data, scheduled maintenance.

Multi-tenancy - Shared resources between salesforce customers. Computing power, data storage, core features can be accessed regardless the business size

Metadata - Data about data, or more simply the details of how your org is handling its record data. User accounts, page layouts and customizations, etc.

API - communicating data between software. Anytime you are accessing anything on Salesforce, you are using APIs. This is powerful as it lets changes/additions to your org's processes be universal across sites, mobile apps, etc.

Navigating Setup:

- Tabs across the top- Home: displays the setup Menu which has 3 main categories
    - Administration: manage users and data (permissions, import/export data, create email templates)
    - Platform Tools: most of customization. view/manage data MODELS, create apps. This is where coding will occur also.
    - Settings: manage company info and org security.

AppExchange:

This is Salesforce's application store. Partners build apps on the SF platform and can make them available to others for free or a fee.

**AppExchange Strategy- you want to get high value without duplicating functions or buying unused functions when deciding on apps from AppExchange.

Steps to successful strategy:

1. Identify departments that would/could be affected. These are your stakeholders.
2. Research available solutions that meet requirements. Discuss business use cases- ask these questions:
    a. What business problem are we trying to solve?
    b. What are the main pain points currently?
    c. How many users need this app?
    d. What's the budget? e. What's the timeline?
3. Find an app that could work, download into a test environment. Ensure that it doesn't interfere with any other apps or customizations.
4. If considering multiple apps, take time to compare
5. When deploying- inform users about what's changing. Provide training and documentation!

How to properly install an app from AppExchange without bringing about the fall of Rome (or your org):

1. Find an app on AppExchange and click ‘Get It Now’ bottom right corner,
2. Key Questions before or during install:
    a. Where do I install, production or dev env (hint: good practice to dev first!)
    b. Who needs permissions to this new app?
3. Install! Where is it though?
    a. Apps are installed as packages (java style)
    b. From Setup, search ‘installed packages’ in quick Find
    c. Click the name of the package, click ‘View Components’ for more info. You can see all the objects and fields the package will include to determine conflicts.

*Salesforce Labs is the creator of free and open source apps from the internal team.

Getting Started:
Platform building blocks: what’s available?
Building from scratch:

- Core platform
- Heroku platform- highly scalable apps and back-end services. Database tools to sync with Salesforce data.
- Salesforce APIs - integrating and connecting everything.
- Mobile SDK - suite of tech to build native or hybrid app

Developing without Code:
Salesforce isn’t just an interface on top of a relational DB, it is an abstraction of the DB driven by the metadata-aware architecture.
The objects become your tables in this abstraction.
It is the metadata that forms the structure of your org (definition of the fields, the settings, etc). When you use it, it pulls in the actual data belonging to the metadata to populate the fields and control actions.

Schema Builder: under ‘Setup’, Schema Builder is a roadmap of your data relationships. A blueprint, if you will, for how all the metadata effects, or not, the rest of the metadata.