

## Week 4 - ~~Langchain~~ Langgraph

(2)

### The Langchain Ecosystem

→ langchain is one of the initial framework that was introduced.

→ Langchain is used for RAG implementation

→ Build your app with Langchain

• Build context-aware, reasoning applications with Langchain's flexible framework that leverages your company's data and APIs

<sup>"Langchain website"</sup>

• Langchain allows us to build apps as well

→ we can use Langchain workflow to build agentic platforms

→ Langgraph →

→ Langgraph is an open-source AI agent framework developed by Langchain, designed for building, maintaining and deploying complex generative AI workflows

→ It provides tools and libraries to create, run and optimize large language models (LLMs) efficiently

→ Langgraph is a low-level orchestration framework that allows developers to build controllable agents and offers features like state management and debugging capabilities.

→ It is an abstraction layer that allows to organise your thinking of workflow of different activities that would have feedback loops and can involve human interference

→ Langchain has a third product called Langsmith that does monitoring.

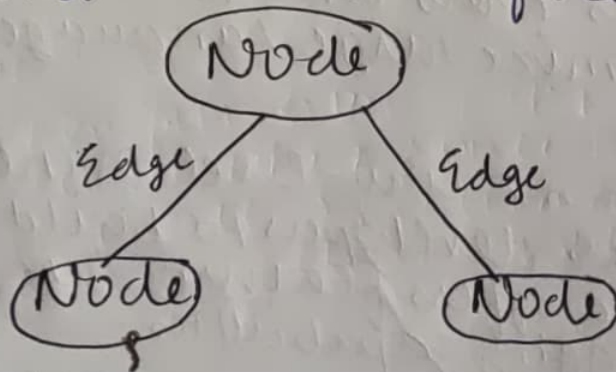
→ We can connect Langgraph via Langsmith so that it can monitor



- Langgraph has three things →
- ① Langgraph studio → visual builder
  - ② Langgraph platform → it is a place used for deploying & running your agents at scale.
  - ③ Langgraph

## Terminologies →

- **Graphs** →
- Agents workflows are represented as graph.
- **State** →
- state represents the current snapshot of the application.
- **Nodes** → Nodes are python fn that represent agent logic.
- These nodes receives current state as input, do something, and return an updated state.
- **Edges** → Edges are python fn that determine which node to execute based on the state. They can be conditional or fixed.



## Five steps to the first graph →

- ① Define the state class
- ② Start the graph builder
- ③ Create a node
- ④ Create edges } repeat to create a lot of nodes and edges
- ⑤ Compile the graph → for running.



State →

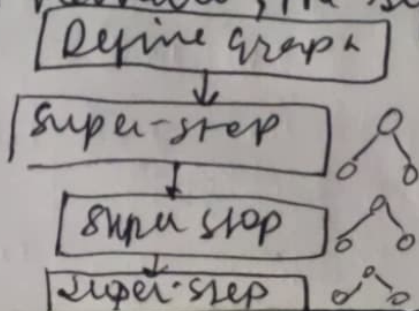
(13)

- State is immutable
- a function takes a state and returns a state as output.
- For each field in your state, you specify a special function called a reducer
- Langraph uses the reducer fn to combine this field with existing state, when you return state.
- This enables Langraph to run multiple nodes concurrently and combine states without overwriting
- A graph defines one set of interactions b/w like agents and their use of tools and perhaps delegating.

### The Super-step →

- A super-step can be considered a single iteration over the graph nodes.
- Nodes that run in parallel are part of the same super step, while nodes that run sequentially before belong to separate super step.
- A graph describes one super-step: one interaction between agents and tools to achieve an outcome.
- Every user interaction is a fresh graph. invoke (state) call.
- The reducer handles updating state during a super-state but not between super steps
- Reducer doesn't handle, the separate super steps

Checkpointing →



→ Project 6 → sidekick on main

goals -

- introducing powerful tool
- structured outputs
- Multi-agent workflow

→ Playwright -

→ Playwright is an open-source automation library for browser.

→ It is used for web scraping

→ It was developed by Microsoft

→ Next version of Selenium.

→ Problem of async -

→ It only supports one event loop.

→ And we can't run any other event loop.

→ for overcoming that -

we can use `next-async` for doing this that is running event loop within another event loop

Worker now -

It is an python function that we will use in edge and decide which route to go in.  
if it is a cool url → return tools  
otherwise → return evaluator