**Commands to build and deploy agents using ADK**   
Agent Development Tool kit:

* Flexible and modular framework for developing and deploying AI agents.
* Compatibility with other frameworks
* Capable of complex tasks and complex workflows

Command to install adk: **pip install google-adk**

Agent ability to understand the user request and generate response is powered

By the LLM. here are two ways of llm calling

1. By using google ai studio api key( we can provide api key in the environment file and we can call any of the gemini model by specifying the model name)
2. By using the google cloud vertex ai

( for this first we need to create the google cloud project, setup the google cli, and enabling the vertex ai api when we are using this vertex ai in environment file we need to specify the project name, location, and keep google\_geani\_use\_vertexai as true)

For running this agent we have different ways

1. Dev UI ( command : **adk web)**

**a)** We will get an url when we run the above command in our browser,

b) In top left corner we can select the respective agent in dropdown( if we do not find the respective agent name in the dropdown menu we need to make sure that we are adk web in parent folder are not).

c) By looking at the events tab in top left we can inspect individual function calls, responses, model responses by clicking on that actions .

d) By clicking on the trace button we can trace logs for each event where we can see latency for each function call.

1. Terminal ( command : **adk run**)

Just by running adk run and agent \_name we can chat with out agent

1. API server( command : **adk api\_server**)

The above command will enable us to create a local fast api server

For Deploying agent some ways are:

1. Deploying to Agent Engine (this is an service managed by the vertex Ai in google cloud which is the easiest way to deploy our adk agents)
2. Deploying to cloud run( we can have full control over scaling and managing our agent by deploying it to the serverless architecture on google cloud)