Cloud :

I worked By the past on AWS, and the first argument sold by AWS is “we are handling many topics like security scalability and you can focus on the development itself”. The inconvenient is that some default properties provided by AWS is depending on the region. The issue it was that we have to manage security feature by ourself for some region. Which is disadvantage because we have to update different configuration for each region.

API Framework :

In this question we can translate grow up by scalability and since we are talking about machine learning we need big data to train the data so we need the fastest algorithm. So I will focus on scalability and speed features.

Scale

It seems we would like to use microservice application. So my approach is to:

* check (if not existing create) the sequence diagram to know how to split correctly each piece in the good microservice.
* Check if we can we do asynchronous call in the api
* Can we duplicate the Db server?
* Can we add a load balancer?
* Is my API supporting the scalability?

New Tech

I recently worked with FastApi and this is responding perfectly to my topic which is link web3 application with web2 application.  
 The interesting point is this api has been created thorough a study of several other frameworks like Django, Sanic or Nodejs. So to summarize FastApi is

• FastAPI is properly fast when we compare it to Flask and Django

• Support for asynchronous code

• Very short development time

• Excellent documentation

• Big Community

Python Specific Tooling

The best way is to do uml class but I am short with time 😊

* Create a class Prospect which is taking as argument Name and id
* Define Name as string and id as UUID
* The output should be a collection
* Create weekly\_target class with 2 methods method\_1 and method\_2 to weekly\_target
* In weekly\_target create instance of class Prospect
* Each time you call Prospect you save the output in ordered collection