My answers shown below.

1. list our products:

2. Customer registration

3. Order placements

4. Sending transactional email depending upon the triggers like registration, forget password, order placement, order fulfilment etc.

**ANS**: It is clearly revealing that above use cases follow single responsibility principle,

I suggest you to create 4 micro services for above 4

For example: Product micro service will have multiple APIs related to product.

There is big chunk of work where our windows service kicks in for:

1. Loading the inventories from suppliers from either endpoints or csv been sent to us This can be achieved by AZ **Function which could be http trigger based**
2. Update the stock information if required This can be achieved by **AZ Function**
3. Process the products images and upload to our file server This can be achieved by **AZ Function**

**ANS** For above Create three Azure Functions (for first you need http trigger enabled for your function). Azure functions are advanced now a days, it supports all capabilities of web jobs.

If above three jobs are long running tasks (>10 mins or continuous load), you may get benefit of price saving if you go with web jobs. It’s like if you stay in hotel for 1 hour(Functions pay per use) , vs 1 day check-in (fixed price in case of web jobs), below links may help you in choosing Web jobs over functions. But My preference is Azure functions.

<https://build5nines.com/azure-functions-vs-web-jobs-how-to-choose/>

<https://stackoverflow.com/questions/36610952/azure-webjobs-vs-azure-functions-how-to-choose>



1.  image storage part as well? How would you organize the products images and where?

Azure blobs in storage account blobs can identify by keys tagged to product.

2. What option I should go for if some of the supplier provides the Feed in .CSV instead real time end point

Real time api always better, create api and post that CSV to api by azure function listening to folder or post CSV to blob and create a trigger in azure function to process it.