Yonder DevOps project

By Raul-Alin Ienei

The first task of the project is to access two endpoints by running the following Docker image, binding to port 30000 through the standard Java-SpringBoot port - yondermakers/yonder-devops-tech-assessment:latest, as I have a intel cpu on my machine. As I have Docker already installed on my machine, I opened a terminal and I run the following command - docker run -p 30000:8080 yondermakers/yonder-devops-tech-assessment:latest.

The second task is to answer the questions located at "/". The first question is: Exemplify two data structures that you know and describe some situations where you would use them. The data structures that I use the most are arrays and dictionaries. An array is a collection of elements, each identified by an index. The elements of an array are stored at contiguous memory locations, being a linear data structure. The most relevant situations where I use arrays are when I want to store a fixed-size collection of elements of the same data type, for example, I would use an array if I want to store my bills for each month. A dictionary is a collection of key-value pairs, where each key is unique and maps to a corresponding value. Dictionaries allow for efficient retrieval, insertion, and deletion of key-value pairs. I would use a map if I have to store and retrieve key-value pairs where quick access to values based on keys is required. For example, storing user preferences with keys representing settings and values representing user preferences.

The second question is: You open a web browser and access http://www.tss-yonder.com. What is the IP address behind this website and how does the browser know how to get the correct IP? The IP addresses behind this website are 2606:4700:20::ac43:49b1, 2606:4700:20::681a:13e and 2606:4700:20::681a:3e. In general, the browser uses the Domain Name System (DNS) protocol, so when the browser is requested to access a website, it will perform a DNS lookup to resolve the domain name to its corresponding IP address.

The third question is: Exemplify two transport protocols and think of two applications that would use each of them. The most common transport protocols that I know are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data between applications. It ensures that data packets are delivered in the same order they were sent and handles retransmission of lost packets. One application that would use this protocol is a barber appointment app, where it is needed that each appointment is registered correctly so that two persons couldn’t get the same appointment at a specific time space. UDP is a connectionless protocol that provides a lightweight, best-effort delivery of data between applications. It does not guarantee delivery or order of packets, making it faster and more efficient but less reliable than TCP. One application that would use this protocol is a streaming app, like Twitch, where the content creators are watched live by a large amount of people.

The fourth question is: You wrote a chat web application in your favorite programming language. You need to host this somewhere and run it so that the entire world can start using it. Describe how you would do that and the tools you would use. The first step that I would take is to get a Hosting Provider. Here I would go with AWS, being familiar with the environment. After that I would set up a Virtual Private Server, in this case an EC2 instance. Then I would configure the server, for the web server I would use Apache Tomcat, for the application server Java Spring, and for the database MySQL, being my most familiar tech stack. After that I would deploy my application by building the app as JAR file, copying this file to the EC2 instance and running the app on this instance. Then I would set up a Domain Name, via Romarg, as I am based in Romania. After that I would secure my app by enabling SSL/TLS encryption via Digicert.

The fifth question is: Now your application is famous but unfortunately it has a lot of bugs. You want only you and a couple of your friends to be able to access it until you patch it. Describe two ways you can achieve this. To restrict the access to the app while I would patch some bugs, I would do an IP Whitelisting so only some specific Ips would be allowed to access the app, or I would do a HTTP Basic Authentication, allowing only the admin users to connect to my app.

The sixth question is: Your application is ready for the public once again. You realize that you forgot about security and any network administrator can see the messages that a user sends or receives. How would you improve your application to prevent this? Is there any way to do this so that not even the application owner (you) can see the messages between two random users? To prevent this from happening, I would implement an end-to-end encryption. This type of encryption ensure hat only the sender and the intended recipient can decrypt and read the messages, making it impossible for intermediaries such as network administrators or the application owner to access the message content.

The seventh question is: What are cookies and what are they used for? Find a cookie used by http://www.tss-yonder.com and copy its name and value. What do you think is its purpose? Cookies are small pieces of data stored on a user’s device by a web browser while browsing a website. They are commonly used to remember user preferences, track user interactions, enable personalized advertising, manage authentication and security. A cookie used by this website is “google-analytics\_v4\_60a4\_\_counter” that is a cookie used by Google Analytics for their analysis, to track user interactions on the website.

The eight question is: While writing your application you need to create more worker processes for processing some data. How can you create child processes in your favorite language? What are the possible states of a process? In Java, the language that I chose for this application and my favorite language, I can create a child process by extending the Thread class. The possible states of a process are running, ready, blocked or waiting, and terminated.

The ninth question is: Your application is running but it still has a few problems. Occasionally, it returns an error page. How can you find the PID of your application? What would you do to debug it? I can find the PID of my application by running in a terminal the “netstat” command as I have a Windows based machine. To debug it, I would look at the logs of the app, I would use the debugging tools of the IDE and I would do some exception handling.

The tenth question is: What DBMS would you use to store your application data and why? How would you store the passwords of each user? As we have a chat application that requires scalability, flexibility and real-time data processing, I would use a NoSQL Database, my choice being MongoDB, as the NoSQL databases are highly scalable and can handle large volumes of concurrent writes and reads, are well-suited for real-time data processing and can support low-latency communication. I would store the password for each user in the database hashed, using a an encryption mechanism as SHA-2 or Bcrypt.

The third task is to create a Python OOP console application that consumes 150 data points from the API and create methods for the following requirements of the authority:

1. List suspended licenses by the authority.

2. Extract valid licenses issued until today's date.

3. Find licenses based on category and their count (e.g., A1 => 4, B

=> 6 ...).

So I created a new Python project with a Python file inside and I written the necessary code. When I tried to run the code, the program thrown the following error: ModuleNotFoundError: No module named 'requests', so I checked my environment. I had already installed Python with version 3.9. I tried to reinstall the requests library but I got the following error: ERROR: Could not install packages due to an OSError: [WinError 2] The system cannot find the file specified: 'c:\\python39\\Scripts\\normalizer.exe'>'c:\\python39\\Scripts\\normalizer.exe.deleteme', so I reinstalled Python with the newest version. After that I installed the requests library and everything worked as planned. The rest of the imports didn’t raise any problems.