New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Course Title: Data Communication & Networks	**Course Number:** CSCI-GA.2662-001
Instructor: Jean-Claude Franchitti	**Session:** 2

Lab #2 – Application Layer

I. Due

Thursday September 23, 2021 at the beginning of class.

II. Objectives

1. Understand the basic principles of computer networks and related protocols
2. Understand how to analyze application layer protocols
3. Deploy a simple web server to Kubernetes

III. References

1. Slides and handouts posted on the course Web site
2. Textbook chapters as applicable

IV. Software Required

1. Microsoft Word.
2. Win Zip as necessary.
3. Wireshark,
4. Docker.
5. IBM Cloud
6. Kubernetes (K8s).

V. Lab Instructions

1. Problem 1 – Protocols Analysis:

   (a) Use a packet sniffer (e.g., Wireshark) to capture the sequence of ASCII characters that are sent and received by a web browser as a result of a request of your choice to a web server. Provide a screen capture of these sequences and add carriage return and line feed characters as needed to improve readability.

(b) Identify the complete URL of the document requested, the HTTP protocol version for both the request and response, the operating system that the web browser is running on, and the kind of web server that answered the request.
(c) Is the HTTP connection persistent?
(d) Which web browser sent the request and why is it important for the server to know this information?
(e) Was the request successful and, if so, what type of document was received by the server?
(f) Experiment with a different application layer protocol of your choice and use Wireshark to analyze the corresponding requests.

2. Problem 2 – Network Application Deployment:

Deploy a simple application to K8s that returns the current time when visited using the path "/time".

Prerequisites:

i. Install Docker from https://www.docker.com/
ii. Create an account in Docker Hub: https://hub.docker.com/

(a) Clone the sample Python Flask application provided at the following link: https://github.com/metacomp/nyu-cs2262-001-fa20/tree/master/sample_time_app
   i. Refer to the sample Dockerfile provided at the above link.
   ii. To build an image: "docker build -t USERNAME/sample-time-app:latest" (USERNAME is your dockerhub username)
   iii. To run: "docker run --name sample-time-app -p 8080:8080 -it sample- time-app"
   iv. Make sure that your app is up and running.
   v. Make sure that your app returns the current time when visited using the "/time" path; make changes to the application accordingly.
(b) Once changes are done, push the image to Docker hub
   i. First login with: "docker login" (remember the username and password created earlier)
   ii. Push the image: docker push USERNAME/sample-time-app:latest
   iii. Check in dockerhub profile to see if you can see the image
(c) Now deploy your image to the IBM Cloud K8s instance.
   i. Create a deployment: "kubectl create deployment sample-time-app -- image=docker.io/USERNAME/sample-time-app:latest"
   ii. Expose your app's port: "kubectl expose deployment/sample-time-app -- type="NodePort" --port 8080"
   iii. Check the exposed port: kubectl get services
   iv. Check the IP of your worked node from IBM Cloud
   v. Access the http://IP:NodePort and confirm your app works
(d) Commit your code to Github (also make sure to submit the NYU Classes). Following should be included in the submission.

i. Dockerfile
ii. Any other file that your Dockerfile uses in order to successfully build the image.

VI. <u>Deliverables</u>

1. Electronic:

Your lab assignment file must be submitted via NYU Brightspace. Name the file "**firstname_lastname_lab_#.docx**" (e.g., "john_doe_lab_2.docx"). The file must be created and sent by the beginning of class. After the class period, the homework is late. The email clock is the official clock.

Also make sure to commit your programming question to Github. We will use the last commit time for grading.

- In Github create a folder for the programming exercise and name it "time_app". Make sure to upload your code in there.

2. Cover page and other formatting requirements:

The cover page supplied on the next page must be the first page of your lab assignment file.

Fill in the blank area for each field.

<u>NOTE</u>:

The sequence of the electronic submission is:

3. **Cover sheet**
4. **Lab Assignment Answer Sheet(s)**

3. Grading guidelines:

Assignment Layout (15%)

- o Lab Assignment is neatly assembled on 8 1/2 by 11 layout.
- o Cover page with your name (last name first followed by a comma then first name), username and section number with a signed statement of independent effort is included.
- o File name is correct.

Answers to Individual Questions (85%):

- o Answers to all questions are complete and correct.
- o Assumptions provided as required.

(100 points total, all questions weighted equally)

VII. Sample Cover Sheet

Name ________________________ Date: ____________
(last name, first name)
Section: ___________

Lab #2 – Application Layer

Total in points (100 points total): ___________________

Professor's Comments:

Affirmation of my Independent Effort: _____________________________
(Sign here)