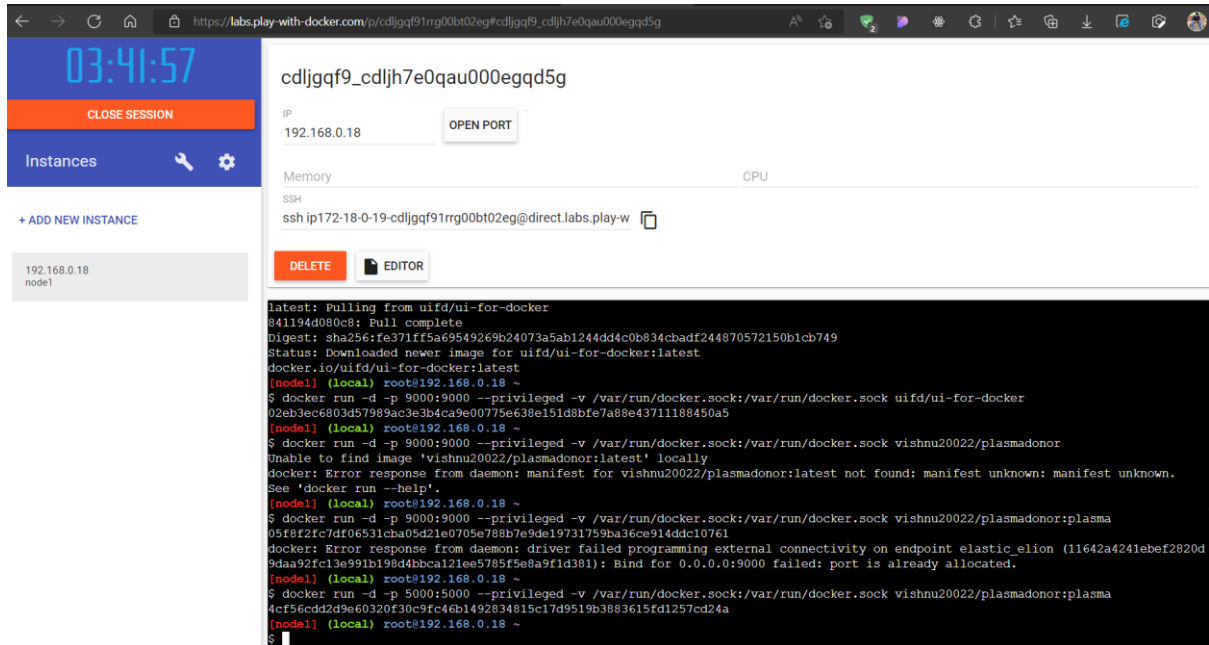


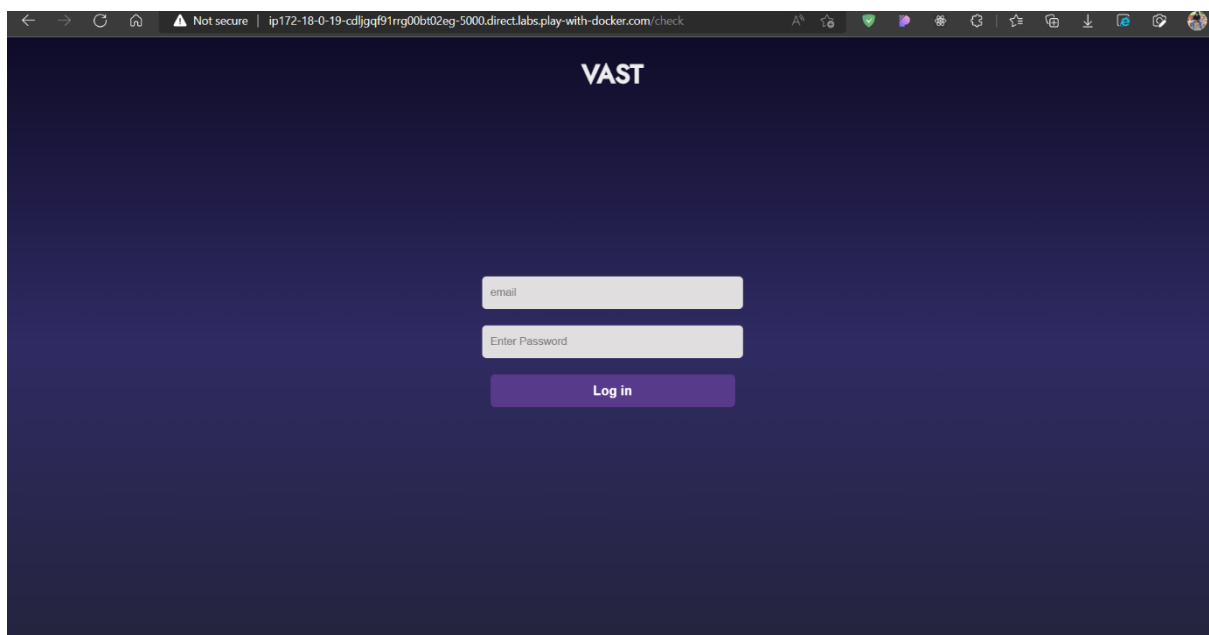
# Assignment 4

Pull an Image from docker hub and run it in docker playground



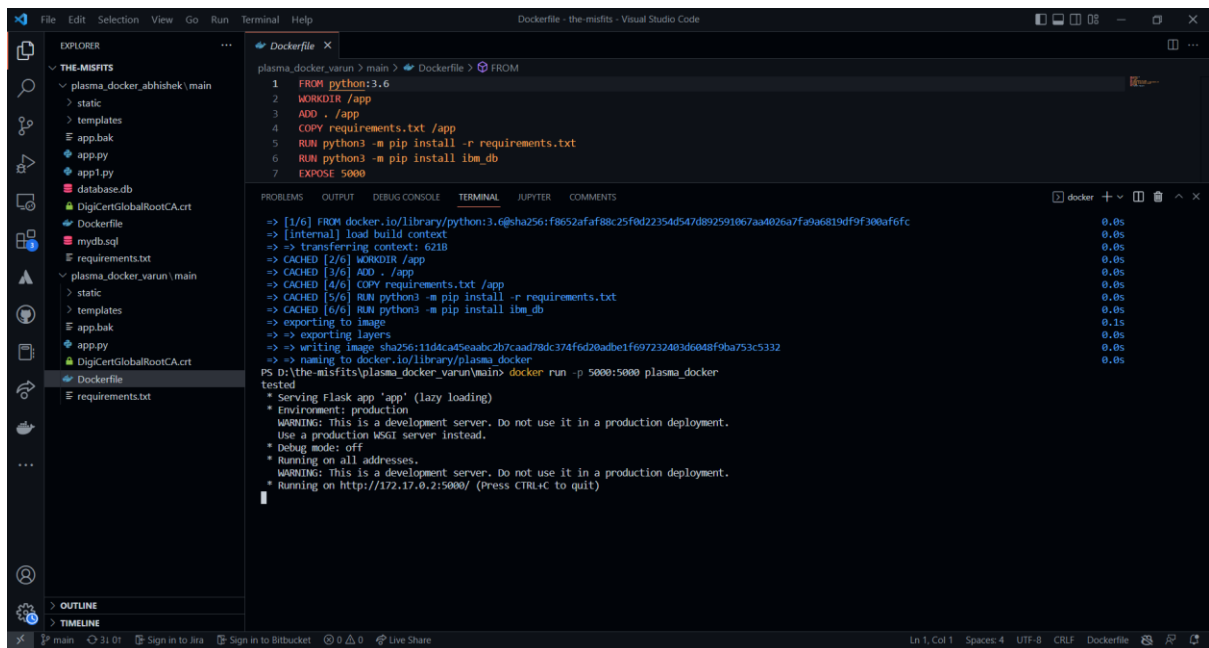
The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:41:57, a 'CLOSE SESSION' button, and a list of instances. The main area displays the instance details for 'cdljgqf9\_cdljh7e0qau000egqd5g'. It shows the IP address 192.168.0.18, an 'OPEN PORT' button, and an SSH command: `ssh ip172-18-0-19-cdljgqf91rrg00bt02eg@direct.labs.play-w`. Below this, there's a terminal window showing the following commands and output:

```
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
02eb3ec6803d57989ac3e3b4ca9e00775e638e151d8bfe7a88e43711188450a5
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock vishnu20022/plasmadonor
Unable to find image 'vishnu20022/plasmadonor:latest' locally
docker: Error response from daemon: manifest for vishnu20022/plasmadonor:latest not found: manifest unknown: manifest unknown.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock vishnu20022/plasmadonor:plasma
05f8f2fc7df06531cba05d21e0705e788b7e9de19731759ba36ce914ddc10761
docker: Error response from daemon: driver failed programming external connectivity on endpoint elastic_elion (11642a4241ebef2820d
9daa92fc13e991b1b98d4bbca121ee5785f5e8a9f1d381): Bind for 0.0.0.0:9000 failed: port is already allocated.
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 5000:5000 --privileged -v /var/run/docker.sock:/var/run/docker.sock vishnu20022/plasmadonor:plasma
4cf56cdd2d9e60320f30c9fc46b1492834815c17d9519b3883615fd1257cd24a
[node1] (local) root@192.168.0.18 ~
$
```



The screenshot shows the VAST login page. The page has a dark blue background with the word 'VAST' in white at the top. Below it, there are two input fields: 'email' and 'Enter Password'. A purple 'Log in' button is positioned below the password field. The browser's address bar shows the URL: `ip172-18-0-19-cdljgqf91rrg00bt02eg-5000.direct.labs.play-with-docker.com/check`.

## Create a docker file for the application and deploy it in Docker desktop application

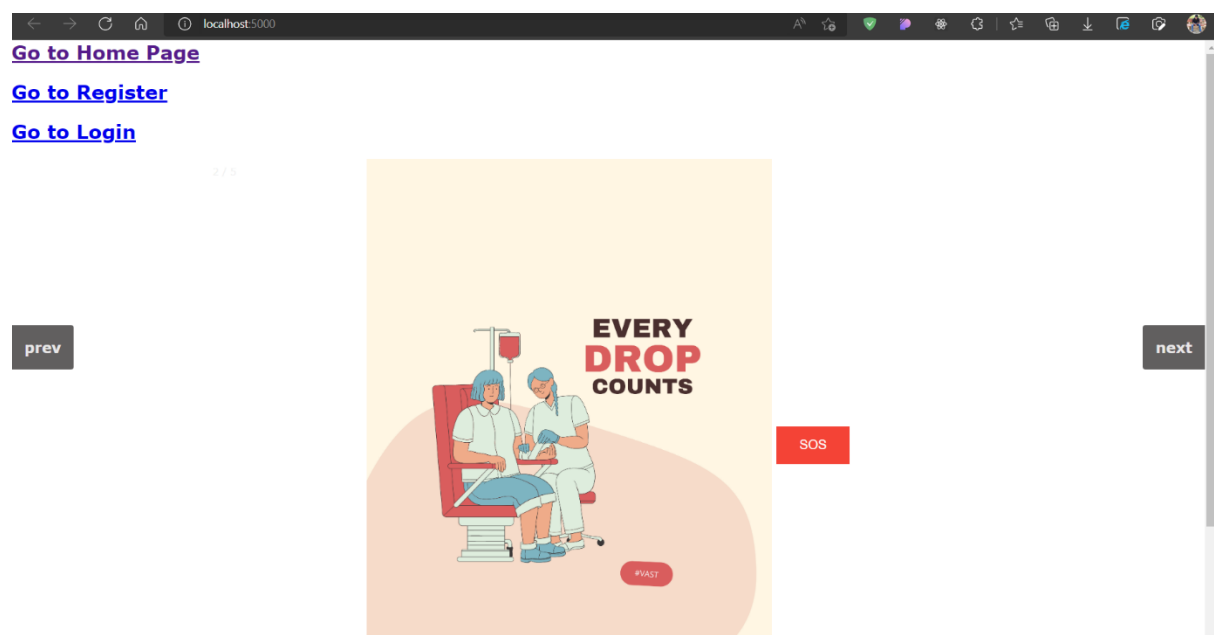
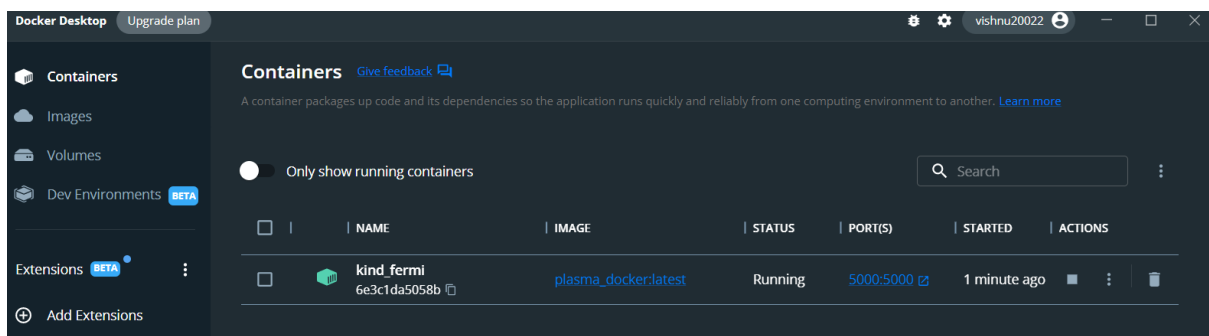


The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following instructions:

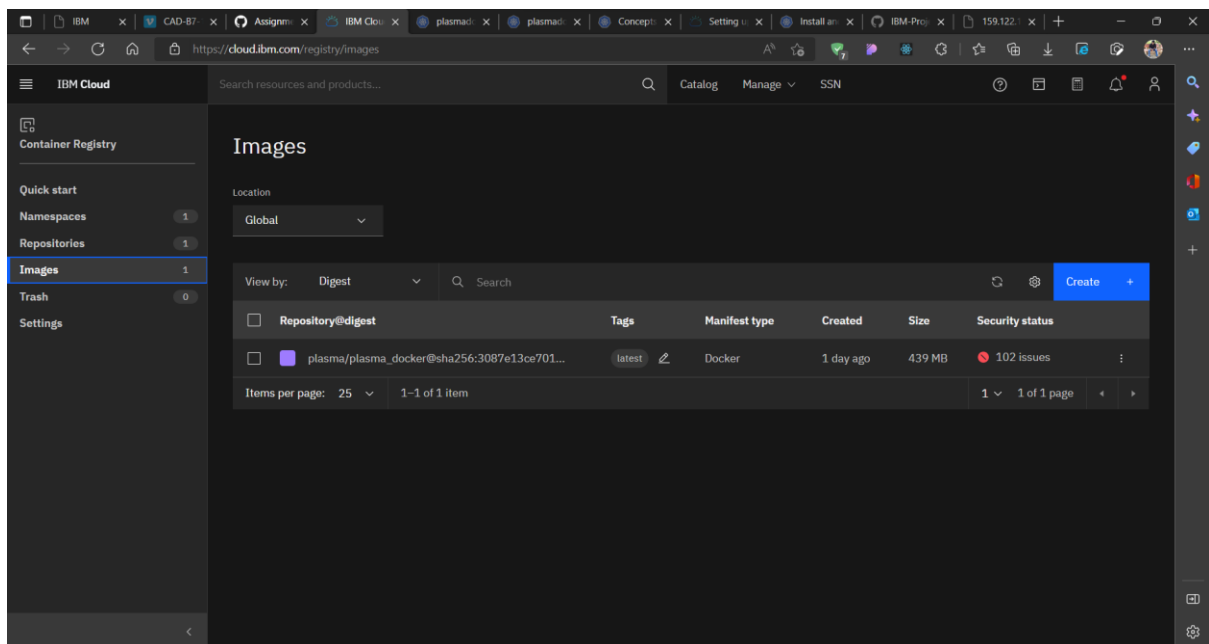
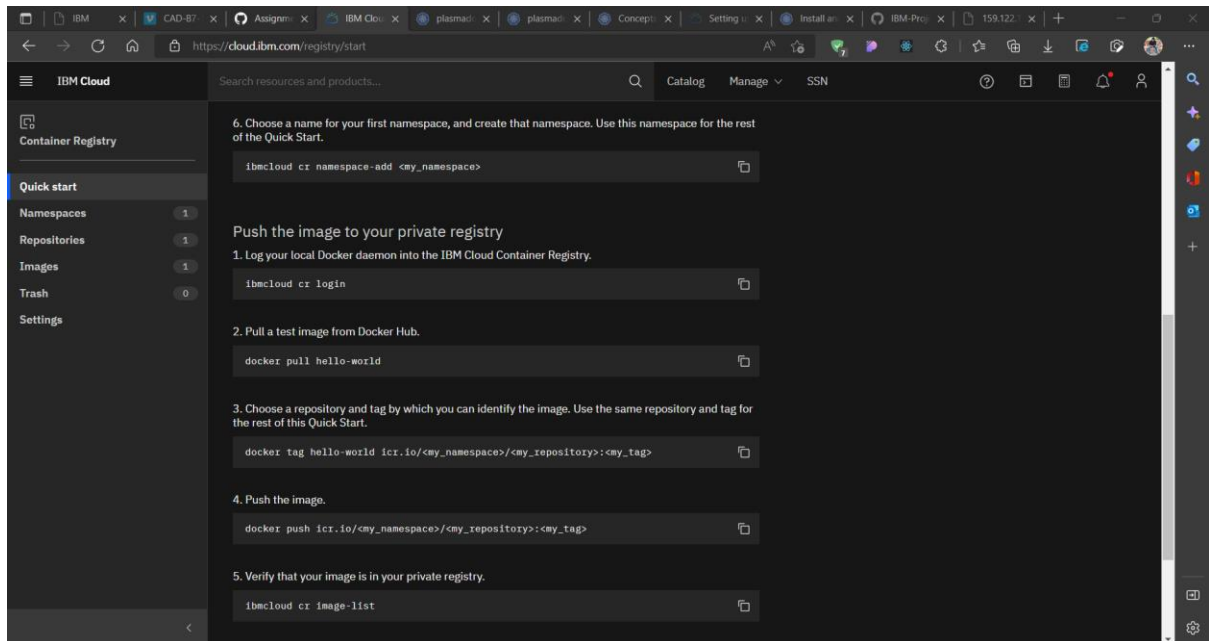
```
1 FROM python:3.6
2 WORKDIR /app
3 ADD . /app
4 COPY requirements.txt /app
5 RUN python3 -m pip install -r requirements.txt
6 RUN python3 -m pip install ibm_db
7 EXPOSE 5000
```

The terminal output shows the build process and the command to run the container:

```
plasma_docker_varun > main > Dockerfile > FROM
-> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 0.0s
-> [internal] load build context 0.0s
-> -> transferring context: 621B 0.0s
-> CACHED [2/6] WORKDIR /app 0.0s
-> CACHED [3/6] ADD . /app 0.0s
-> CACHED [4/6] COPY requirements.txt /app 0.0s
-> CACHED [5/6] RUN python3 -m pip install -r requirements.txt 0.0s
-> CACHED [6/6] RUN python3 -m pip install ibm_db 0.0s
-> exporting image 0.1s
-> writing image sha256:11d4ca45eaabc2b7caad78dc374fd20adbe1f697232403d6048f9ba753c5332 0.0s
-> naming to docker.io/library/plasma_docker 0.0s
PS D:\the-misfits\plasma_docker_varun\main> docker run -p 5000:5000 plasma_docker
tested
* Serving Flask app 'app' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug modes: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
```



## Create an IBM container registry and deploy the app



Create a Kubernetes cluster in IBM cloud and deploy image and expose the same app to run in nodeport

```
Windows PowerShell
Session Affinity: None
Events: <none>
PS C:\Users\srivi> ibmcloud ks cluster ls
OK
Name ID cdlui2gf0lvv2pci4kg State Created 19 hours ago Workers 1 Location ams Version 1.24.7_1542 Resource Group Name Default Provider classic
PS C:\Users\srivi> kubectl get pods
No resources found in default namespace.
PS C:\Users\srivi> kubectl create deployment plasmadonor-deploy --image=icr.io/plasma/plasma_docker@sha256:3687e13ce7010ae2cbd5255ef61641fbf16e9da865a4b097a
b6926277a621996
deployment.apps/plasmadonor-deploy created
PS C:\Users\srivi> kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
plasmadonor-deploy 1/1 1 1 13s
PS C:\Users\srivi> kubectl expose deployment/plasmadonor-deploy --type="NodePort" --port 5000
service/plasmadonor-deploy exposed
PS C:\Users\srivi> kubectl describe service plasmadonor-deploy
Name: plasmadonor-deploy
Namespace: default
Labels: app=plasmadonor-deploy
Annotations: <none>
Selector: app=plasmadonor-deploy
Type: NodePort
IP Family Policy: SingleStack
IP Families: IPv4
IP: 172.21.208.227
IPs: 172.21.208.227
Port: <unset> 5000/TCP
TargetPort: 5000/TCP
NodePort: <unset> 30145/TCP
Endpoints: 172.30.101.141:5000
Session Affinity: None
External Traffic Policy: Cluster
Events: <none>
PS C:\Users\srivi> ibmcloud cs workers --cluster cdlui2gf0lvv2pci4kg
OK
ID kube-cdlui2gf0lvv2pci4kg-plasmadonor-default-0000003c Public IP 159.122.183.185 Private IP 10.144.214.235 Flavor free State normal Status Ready Zone mil01 Version 1.24.7_1543
PS C:\Users\srivi>
```

Go to Home Page

Go to Register

Go to Login

prev

next

SOS

## PLASMA DONORS ARE ESSENTIAL

**WHAT IS PLASMA?**  
Plasma is the clear-colored liquid portion of blood composed of water, salts, and proteins which contains numerous proteins essential for proper functioning of the body. INSUFFICIENT LEVELS OF ABO+ PLASMA PROTEIN prevents the body from carrying out vital functions, causing a VARIETY OF ENDORGE AND LIFE-THREATENING MEDICAL CONDITIONS.

**WHAT IS COVID-19 PLASMA?**  
COVID-19 plasma is plasma collected from PATIENTS WHO HAVE RECOVERED FROM COVID-19. In addition to the proteins found in plasma, COVID-19 PLASMA CONTAINS PATIENT-SPECIFIC ANTIBODIES.

**WHO CAN DONATE SOURCE PLASMA?**  
Source plasma donors undergo a RIGOROUS DONATION SCREENING PROCESS, which determines donor eligibility, and must maintain a healthy lifestyle to remain a qualified donor.

**WHO CAN DONATE COVID-19 PLASMA?**  
In addition to meeting source plasma donor eligibility criteria, individuals must have had a POSITIVE DIAGNOSIS OF COVID-19 by a laboratory test and must have COMPLETE RESOLUTION OF SYMPTOMS FOR AT LEAST 10 DAYS prior to donation.

**PLASMA DONOR ELIGIBILITY CRITERIA MUST BE:**  
✓ 18-55 YEARS OLD  
✓ 175-185 CM (5'8"-6'1")  
✓ MEDICALLY SOUND  
✓ MUST HAVE NO SPECIFIC MEDICAL CONDITIONS

**THERAPIES MADE FROM SOURCE PLASMA TREAT:**  
• Acute & Antigenic Deficiency  
• Bleeding disorders, such as Hemophilia  
• Chronic Inflammatory Demyelinating Polyneuropathy  
• Hematologic Dysplasias  
• Primary and Secondary Immune Deficiencies  
• Acute conditions, such as shock, trauma, and burns

**THERAPIES MADE FROM COVID-19 PLASMA COULD TREAT:**  
• Patients with COVID-19  
• Pre- and post-exposure prophylaxis  
• Those at high risk for contracting COVID-19, such as:  
• Individuals with underlying conditions or chronic lung disease  
• Health care workers  
• Public health service staff

To find where you can donate Source and COVID-19 plasma, visit: [www.DonatingPlasma.org](http://www.DonatingPlasma.org)

EVERY DONATION 130: 900: 1200: