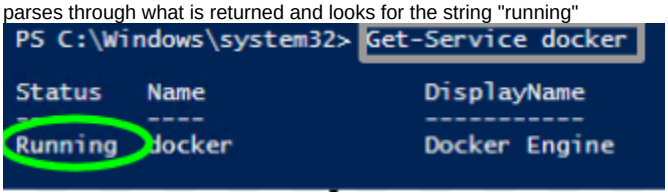
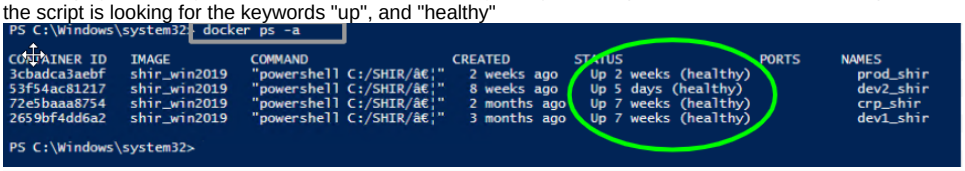
Docker Health Check Process

What does it do?

The docker health check automatically checks to see if the Docker Service is up and running. Then, checks to see if each of the containers are running and healthy, sending emails along the way notifying the DB ops team of any failures.

Checking the health

1. RDP into D1VAZINT3000
2. The health check is run through the script: C:\scripts\healthcheck.ps1
    a. The first thing that the script checks is whether or not the docker service is even running, and if not it sends an email to the DB Ops team.
        i. checks if running or not using the PowerShell command "get-service docker'
        ii. parses through what is returned and looks for the string "running"
        iii. 

    b. If the Docker service is running, then the script moves on to checking the health of each individual container.
        i. A variable is created for each container name
        ii. A variable is made for when the container is healthy ($healthy) and another for when a container is unhealthy ($unhealthy) (used when emailing)
        iii. $errorcode makes sure that when one of them doesn't meet the requirements that it will send an email
        iv. $statusoutput is used to construct the body of the email using the $healthy and $unhealthy variables incase one needs to be sent out
        v. The script goes through all of the containers individually using "docker ps -f $container" in a for loop
        vi. the script is looking for the keywords "up", and "healthy"
        vii. 

        viii. **Container Health Check**

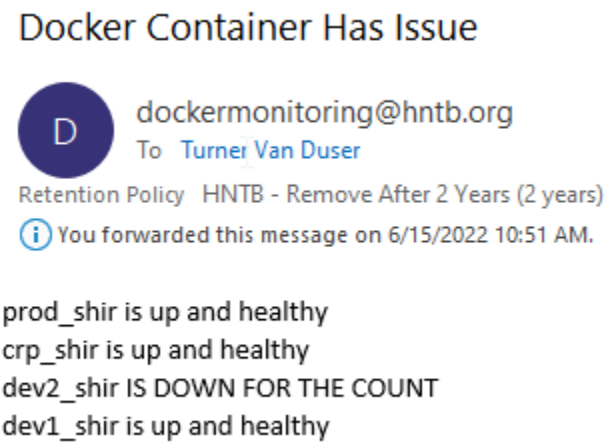
```
$containername = ""
$containername = @(
    'prod_shir',
    'crp_shir',
    'dev2_shir',
    'dev1_shir'
 )

#loop through and assign health to each container
$healthy = "is up and healthy`n"
$unhealthy = "IS DOWN FOR THE COUNT`n"
$containerstatus = ""
$statusoutput = ""
$errorcode = 0
foreach ($container in $containername)
{
    if(docker ps -f "name=$container" | Out-String | Select-String $container | Select-
String  'up.*healthy')
    {
       $containerstatus =  $container + " " + $healthy
    }
    else
    {
       $containerstatus = $container + " " + $unhealthy
       $errorcode = 1
    }
    $statusoutput = $statusoutput + $containerstatus
}
```

Emailing using Send-MailMessage

1. Email if docker is not running
    a. If the docker service isn't running then an email is sent to the DB Ops team alerting them of this issue.
    b. Script checks for the string "Running" in the "get-service docker" command.
    c. **Docker Not Running**

```
if(get-service docker | out-string | Select-String Running -NotMatch)
{
    Send-MailMessage -From "dockermonitoring@hntb.org" -To
"tHNTBITDatabaseOperations@hntb.com" -Subject "Docker Not Running" -Body "when
the service was checked, the service 'docker' was not running. " -SmtpServer
"smtp.hntb.org"
    break
}
```

2. Docker Container not healthy email
    a. When looking through the containers individually, if one of the containers isn't up and healthy it raises the variable $errorcode by one, and if $errorcode > 0 then it knows to send the email



    b.
    c. **Container Not healthy**

```
if($errorcode -ne 0)
{
    Send-MailMessage -From "dockermonitoring@hntb.org" -To
"HNTBITDatabaseOperations@hntb.com" -Subject "Docker Container Has Issue" -
Body $statusoutput -SmtpServer "smtp.hntb.org"
    break
}
```

Scheduling the task to run

1. The automation of the DockerHealthCheck is handled through the windows task scheduler.
2. Check out this article on how to run PowerShell through the task scheduler
    a. This task uses AppOpsAutomation as the user running this task
3. The task runs every 15 min. starting at 12:10 a.m.