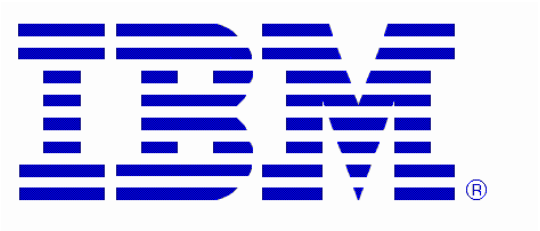


WebSphere Application Server トラブルシューティングおよびパフォーマンス ワークショップ（Docker 版）準備手順

著者

- ケビン グリゴレンコ <Kevin Grigorenko> (kevin.grigorenko@us.ibm.com)

目次

1　ワークショップの準備

このワークショップは、Docker 環境で行うことを前提としています。Windows や Mac をお使いの場合は Docker Desktop をインストールしてください。Linux をお使いの場合は Docker をインストールしてください。

ワークショップでは、Traditional WAS と WAS Liberty のトラブルシューティングとパフォーマンス・チューニングの概念および解析に必要なツールの使い方を学びます。

トラブルシューティングとパフォーマンス・チューニングは、多くの場合、オペレーティングシステムや Java のレベルで行われるため、このワークショップで取得する技術は WAS だけでなく広い範囲で応用できます。

このワークショップの Docker イメージには、Traditional WAS と WAS Liberty があらかじめインストール、構成されているので、課題にすぐにとりかかれるようになっています。

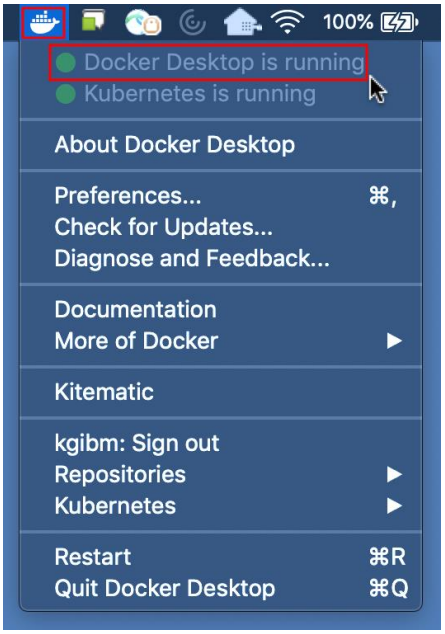
注）ワークショップの Docker イメージには、ワークショップ用に多くのサービスやツールがインストールされています。本番環境で使われる Docker には、必要なサービスのみインストールするよう推奨されています。

1. Docker Desktop をインストールしてください。
    a. Mac ("Apple Mac OS Sierra 10.12"以降のバージョン)
        - ダウンロード: https://hub.docker.com/editions/community/docker-ce-desktop-mac
        - 詳細は次のリンクをご参照ください。 https://docs.docker.com/docker-for-mac/install/
    b. Windows ("Microsoft Windows 10 Professionl 64-bit”または “Microsoft Windows 10 Enterprise 64-bit”)
        - ダウンロード: https://hub.docker.com/editions/community/docker-ce-desktop-windows
        - 詳細は次のリンクをご参照ください。 https://docs.docker.com/docker-for-windows/install/
    c. Linux をお使いになる場合は、下記の Docker をインストールして、下記のコマンドで実行できます。 (sudo systemctl start docker):
        - Fedora Linux の例です。 https://docs.docker.com/install/linux/docker-

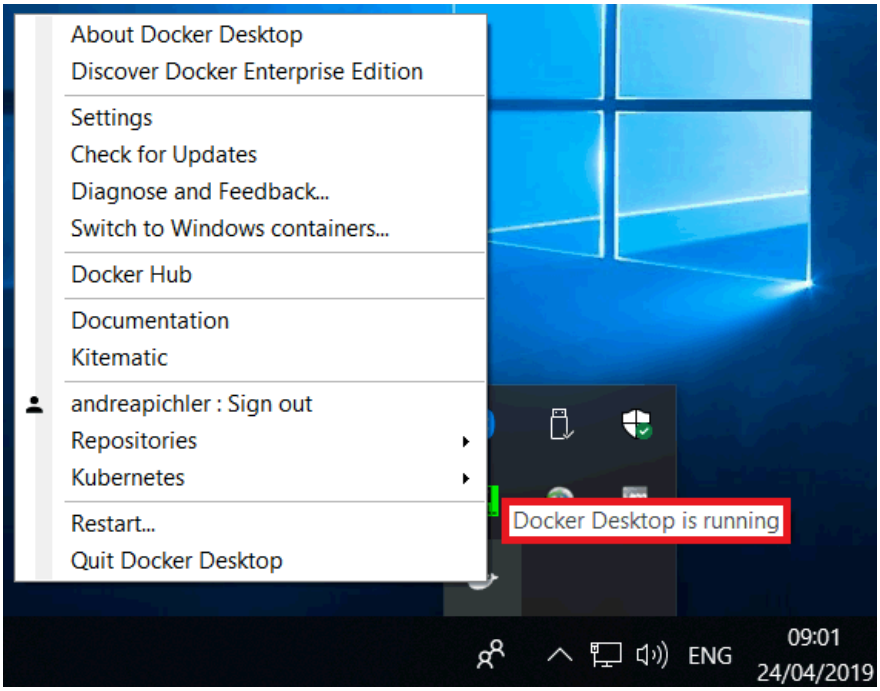
ce/fedora/

2. Docker をスタートします。Docker Desktop を見ると Docker が実行されていることが確認できます。

macOS:



Windows:



3. Docker の実行環境に、十分なリソースを与えてください。特にメモリー使用量を十分に割り当ててください。

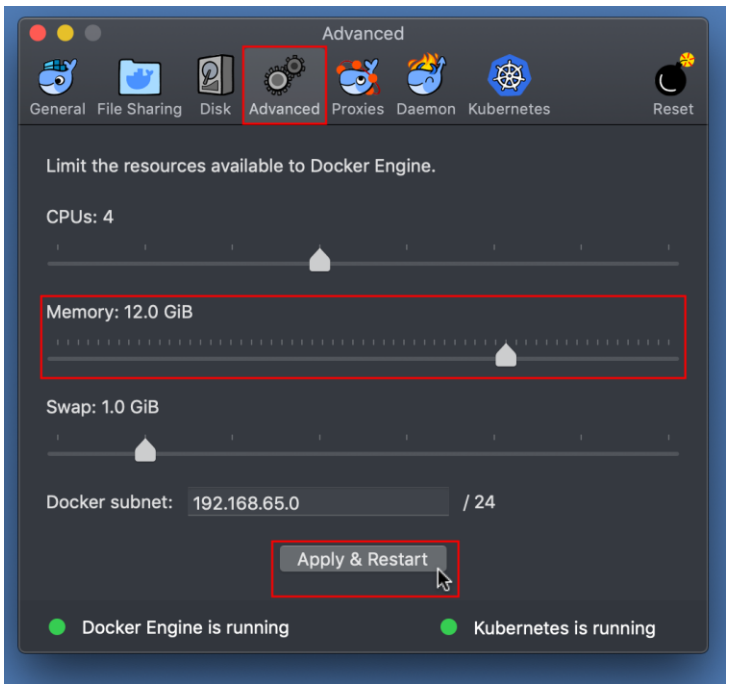
    a. Docker Desktop のアイコンをクリックし、 “Preferences…” (macOS) または “Settings” (Windows) を選びます。

    b. Advanced のタブを選択します。

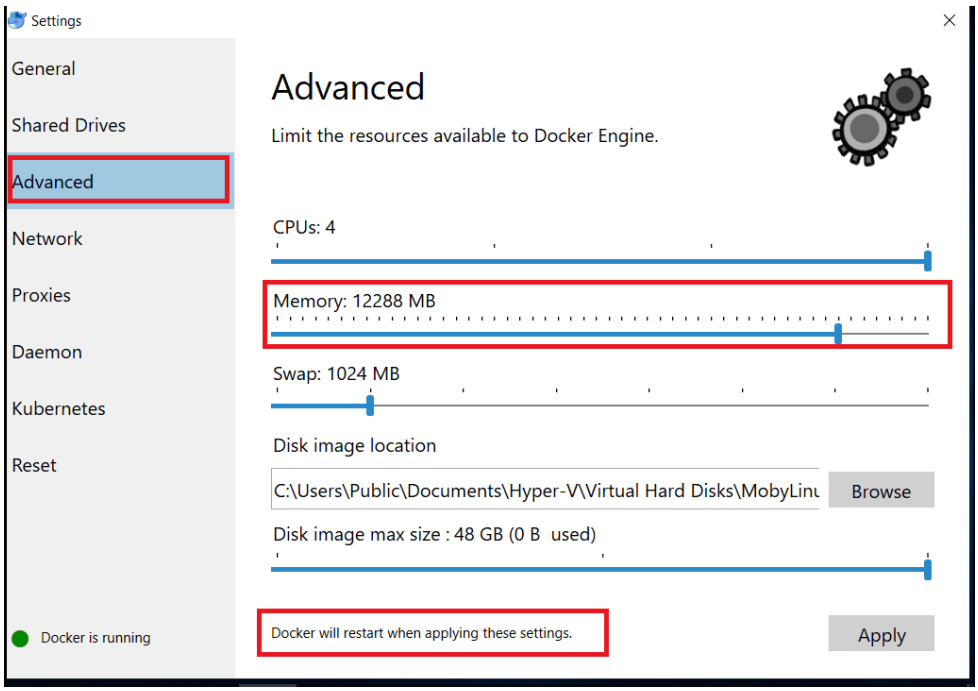
    c. メモリー量を調節します。ワークショップを行うには、8GB 以上のメモリーが必要です。

    d. “Apply & Restart”(macOS) または“Apply”(Windows) を押して設定を保存します。

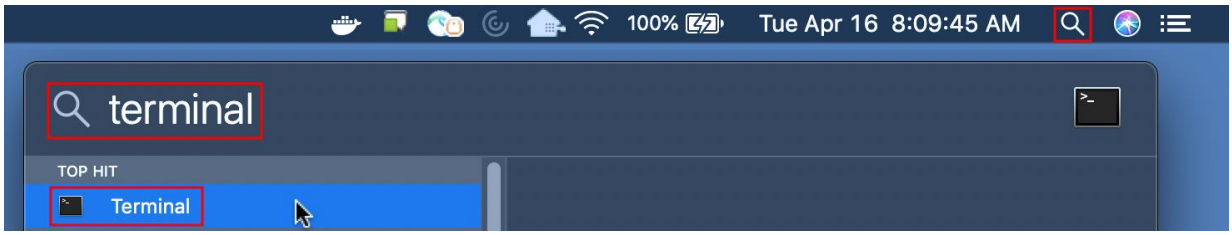
    macOS:



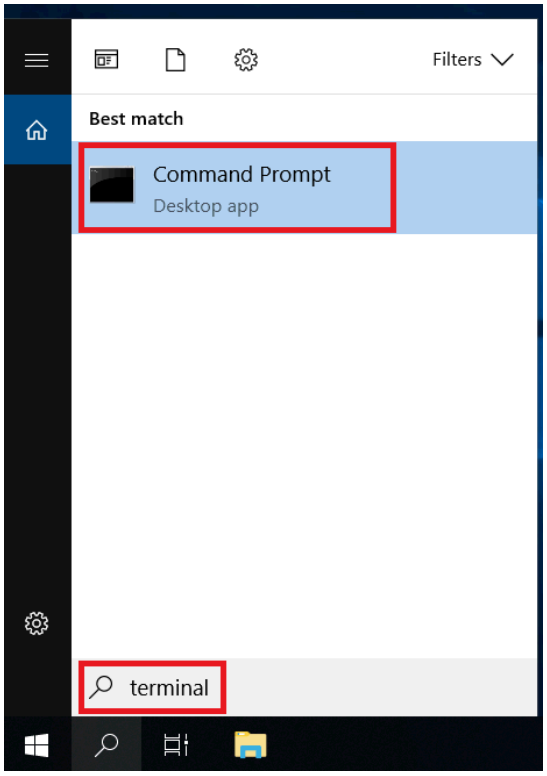
    Windows:


Settings
General
Shared Drives
Advanced
Network
Proxies
Daemon
Kubernetes
Reset
Docker is running
Advanced
Limit the resources available to Docker Engine.
CPUs: 4
Memory: 12288 MB
Swap: 1024 MB
Disk image location
C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\MobyLinu
Browse
Disk image max size : 48 GB (0 B used)
Docker will restart when applying these settings.
Apply

4. macOS や Linux のターミナル、または Windows のコマンドプロンプトを開けます。
   macOS:



   Windows:



5. Docker のイメージを下記のコマンドでダウンロードします。

```
docker pull kgibm/fedorawasdebug
```

   注）Docker のイメージは20GB ほどのサイズです。この Docker のイメージをワークショップの前に予めダウンロードされることをお勧めします。

6. ワークショップ用の Docker コンテナをスタートします。コマンドで使用されているポート(9080, 9443, 9043, 9081, 9444, 5901, 5902, 3389, 22, 9082, 9445)が、他のプログラムに使用されていないようにしてください。

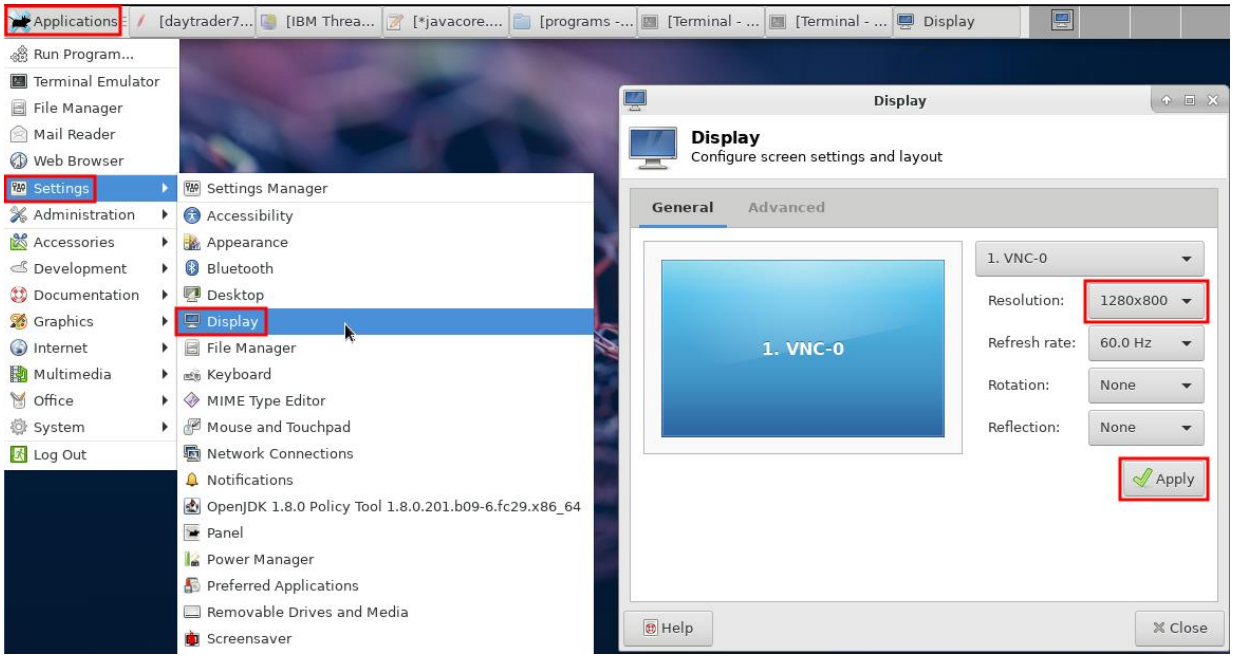
```
docker run --cap-add SYS_PTRACE --ulimit core=-1 --rm -p 9080:9080 -p 9443:9443 -p
9043:9043 -p 9081:9081 -p 9444:9444 -p 5901:5901 -p 5902:5902 -p 3390:3389 -p 22:22
-p 9082:9082 -p 9445:9445 -it kgibm/fedorawasdebug
```

7. コンテナの開始から30秒後くらいで VNC や Remote Desktop のサービスが立ち上がり

ます。下記の要領で、Docker コンテナにログインできることを確認してください。

a. macOS 付属の VNC クライアント
    i. 新しいターミナルをあけて次のコマンドを実行します。
        1. open vnc://localhost:5902
        2. パスワードを入力してください: **websphere**
b. Linux の VNC クライアント
    i. 新しいターミナルをあけて次のコマンドを実行します。
        1. vncviewer localhost:5902
        2. パスワードを入力してください: **websphere**
c. Windows に VNC クライアントがインストールされている場合（Windows 付属のソフトウェアではありません）
    i. 上記と同様に **localhost** のポート **5902** に接続し、パスワードとして **websphere** を入力してください。
d. Windows リモートデスクトップ
    i. Windows のリモートデスクトップを使用する場合は、いくつかのステップが必要です。”2 補足”の2.1　Windows のリモートデスクトップから Docker コンテナにアクセスする を参照してください。
e. SSH:
    i. SSH や Putty などを使うと本番環境に近い状態で Docker コンテナにアクセスできます。ほとんどのワークショップのツールは GUI を使うので、VNC クライアント、またはリモートデスクトップで接続することが必要です。下記は、ターミナルから ssh でログインするコマンド例です。
        1. ssh was@localhost
        2. Password: **websphere**

8. VNC クライアントから、ディスプレイの解像度を Docker コンテナの中で変更できます。VNC クライアントは自動的に新しい解像度に合わせて表示します。下記の例では Setting から Display 表示を1200 x 800に変えています。



9. 端末のブラウザ、またはリモートデスクトップや VNC のブラウザから http://localhost:9080/ を指定して WAS Liberty にアクセスしてみましょう。

10. 同様に http://localhost:9081/swat/ を指定して Traditional WAS で動いているアプリケーションにアクセスしてみます。

    a. Traditional WAS はコンテナが起動してから数分で立ち上がります。

    b. Traditional WAS の管理コンソールには https://localhost:9043/ibm/console から下記のユーザー名とパスワードでログインできます。

        i. User: **wsadmin**

        ii. Password: **websphere**

11. Docker コンテナを起動して、VNC やリモートデスクトップで接続できることを確認したらワークショップの準備は完了です！

    下記の手順にしたがって、コンテナを終了します。まず“docker ps”コマンドで、コンテナ名を見つけ、”docker stop”コマンドでコンテナを指定して終了します。コンテナが終了しても”docker images” コマンドで、ダウンロードしたワークショップのイメージが保存されていることが確認できます。ワークショップを始めるときは、このイメージを使って再びコンテナを起動します。

```
C:\>docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED
STATUS              PORTS
NAMES
```

```
928a1531600c        kgibm/fedorawasdebug   "/entrypoint.sh"    About a minute
ago   Up About a minute   0.0.0.0:22->22/tcp, 0.0.0.0:3389->3389/tcp,
0.0.0.0:5901-5902->5901-5902/tcp, 0.0.0.0:9043->9043/tcp, 0.0.0.0:9080-
9081->9080-9081/tcp, 0.0.0.0:9443-9444->9443-9444/tcp   keen_brattain
```

```
C:\>docker stop keen_brattain
```

```
keen_brattain
```

```
C:\>docker ps
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
```

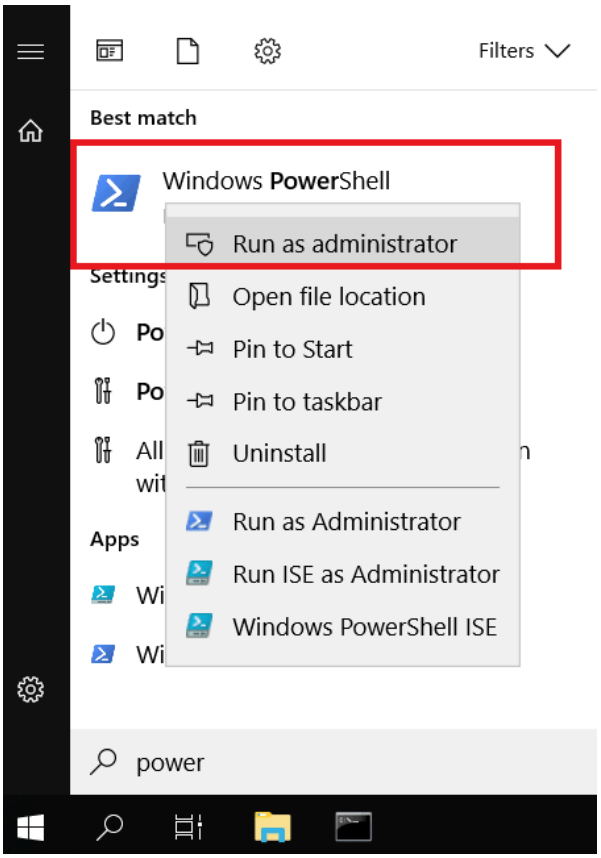
```
C:\>docker images
```

```
REPOSITORY           TAG       IMAGE ID        CREATED        SIZE
kgibm/fedorawasdebug latest    93ba47c73042    2 weeks ago    13.2GB
```

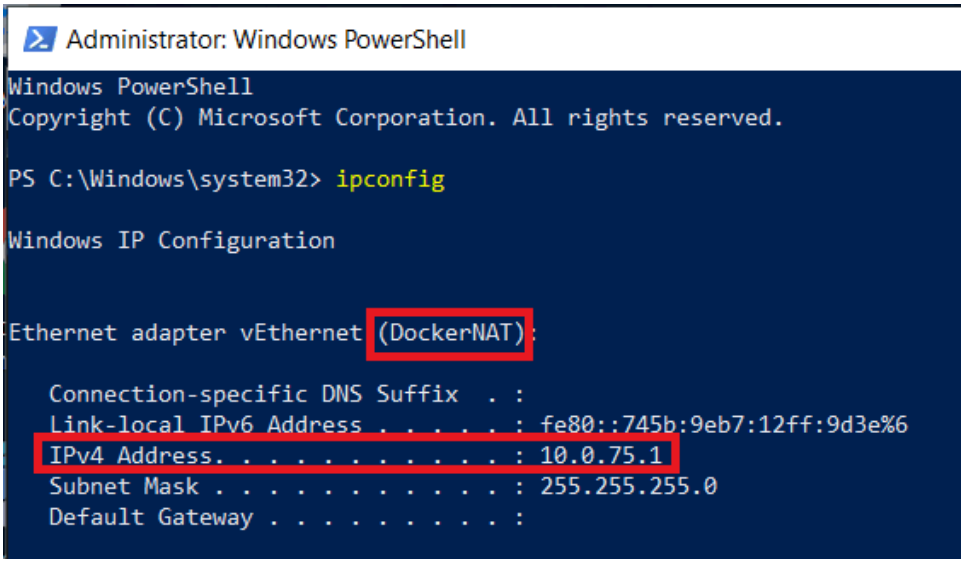
2 補足

2.1 Windows のリモートデスクトップから Docker コンテナにアクセスする

1. Windows の PowerShell を管理者権限でオープンします。



2. ipconfig コマンドを実行し DockerNAT adapter の IP アドレスを見つけます。



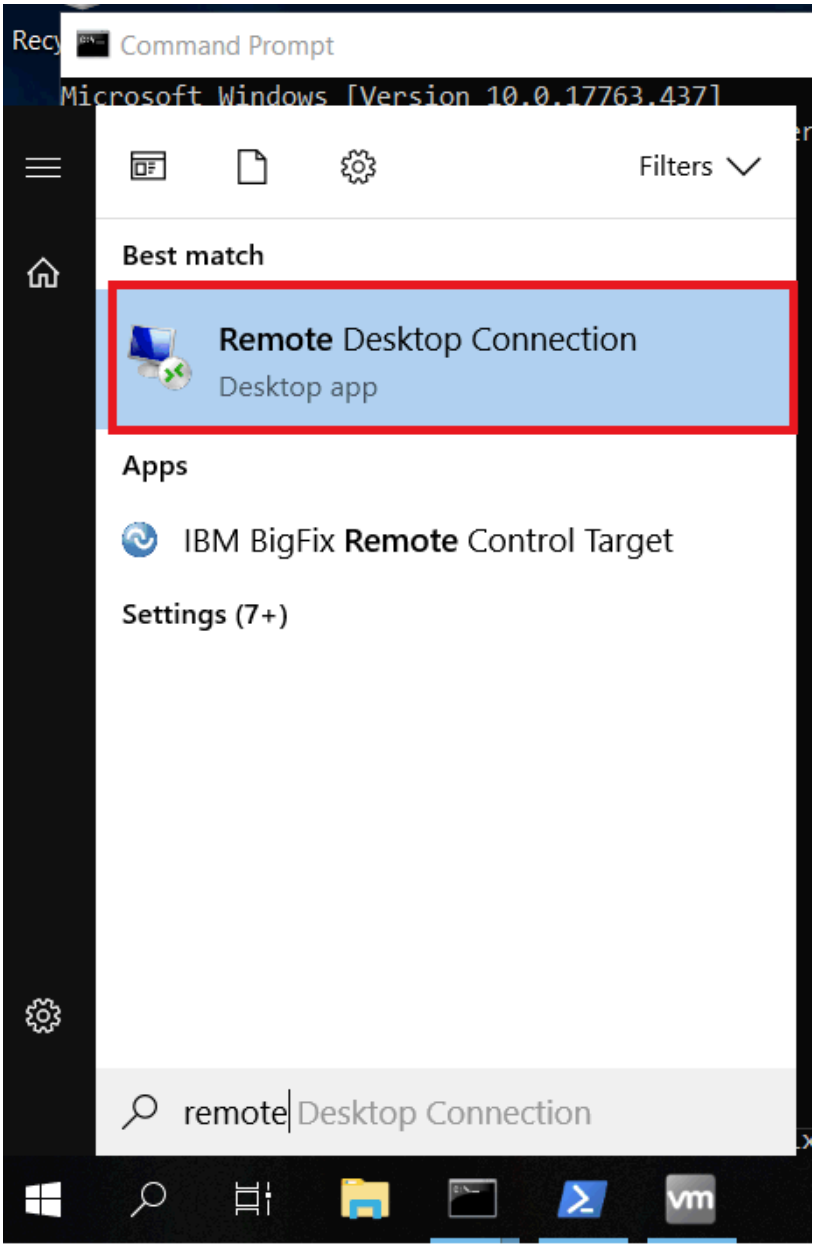
3. 下記のコマンドを PowerShell から実行し Docker のポートが Firewall でも見えるよ

うに Rule を作ります。

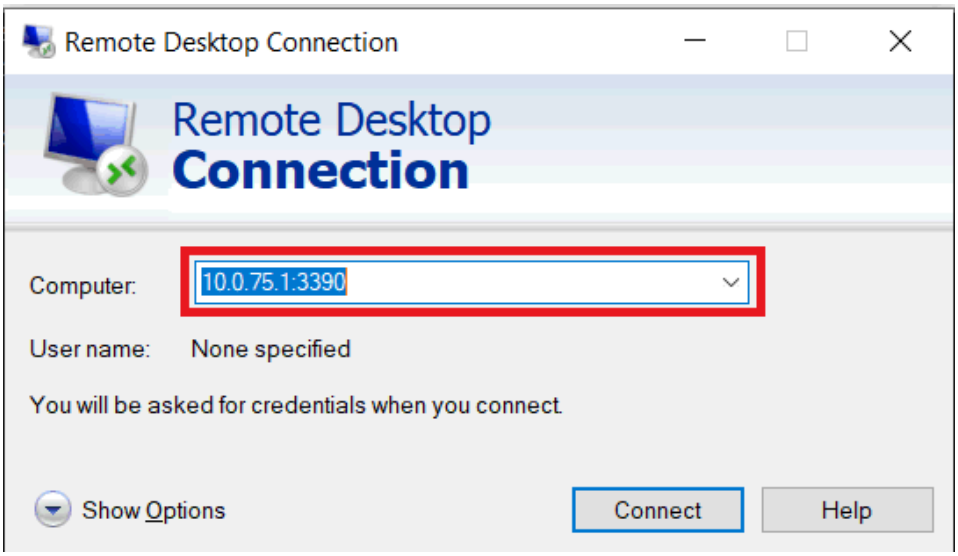
```
New-NetFirewallRule -Name "myRDP" -DisplayName "Remote Desktop Protocol" -Protocol
TCP -LocalPort @(3389) -Action Allow

New-NetFirewallRule -Name "myContainerRDP" -DisplayName "RDP Port for connecting to
Container" -Protocol TCP -LocalPort @(3390) -Action Allow
```

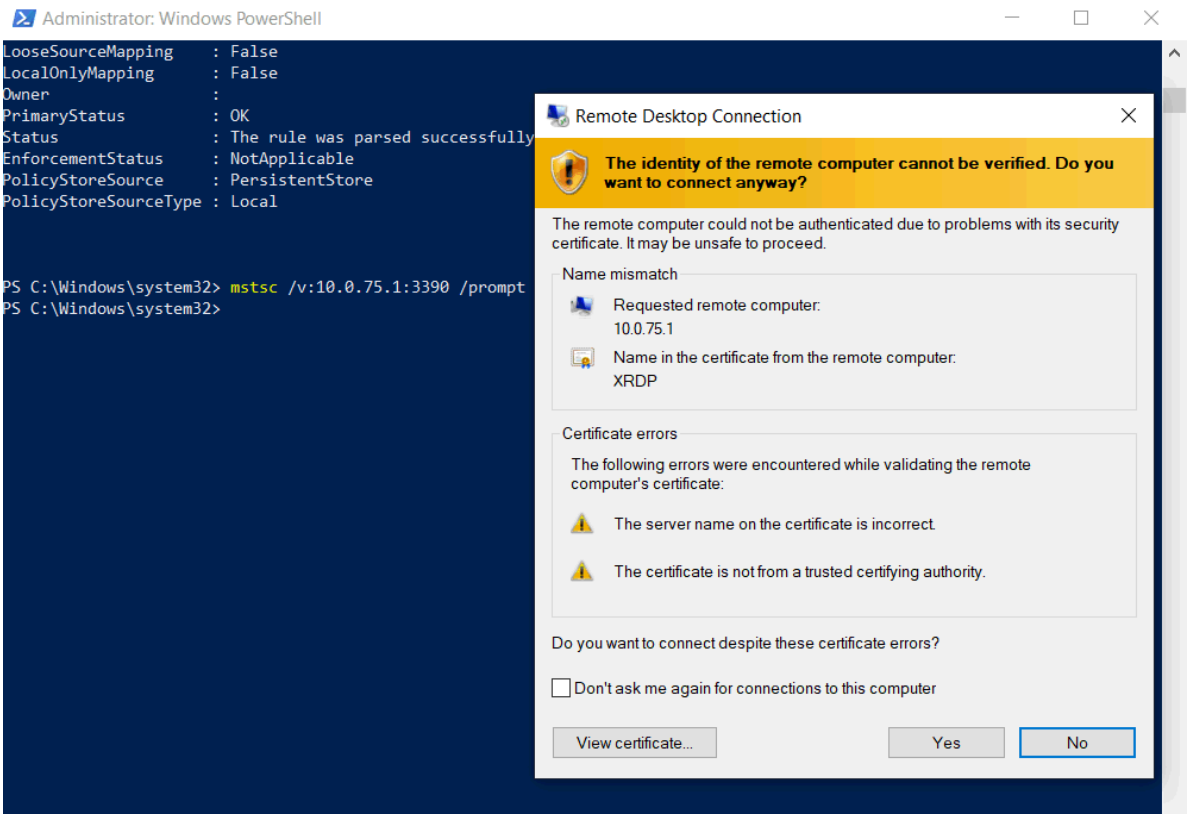
4. リモートデスクトップを起動します。



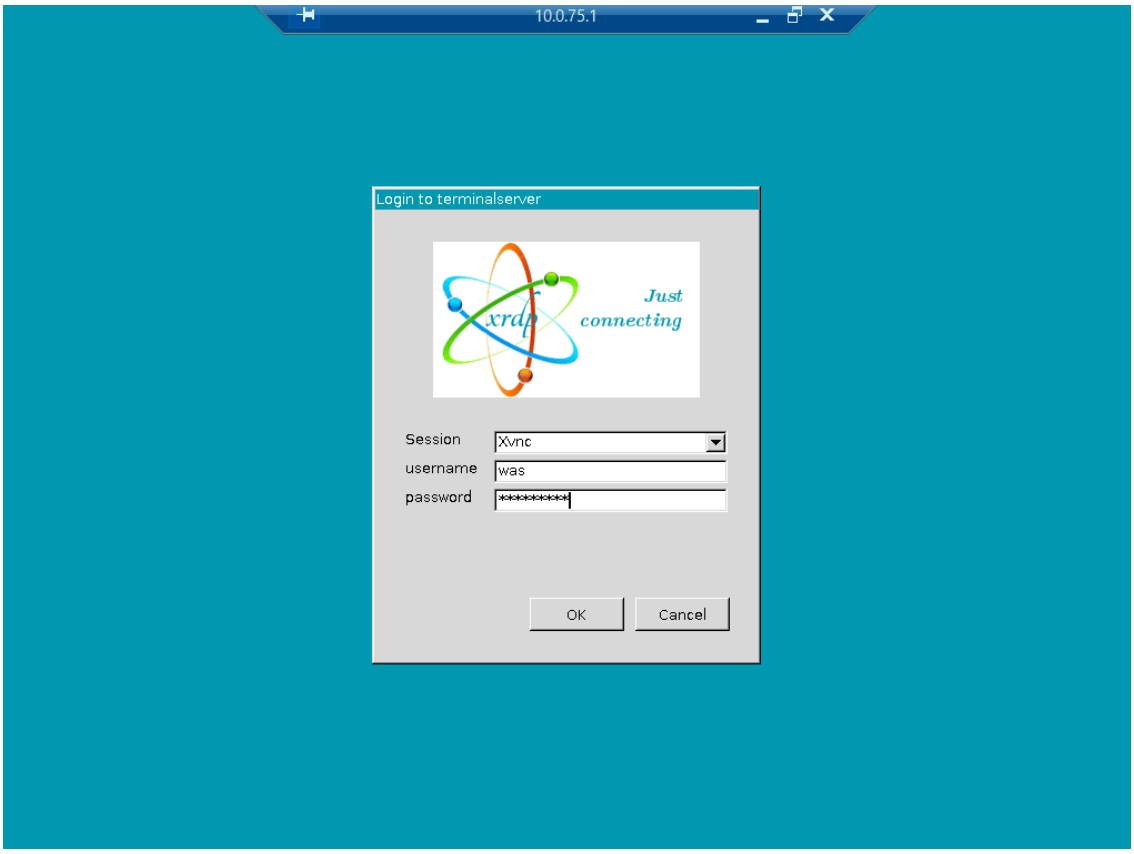
5. 先ほど項番2.で見つけた DockerNAT IP address (例: 10.0.75.1) にポート番号:3390 を指定して "Connect"を押します。



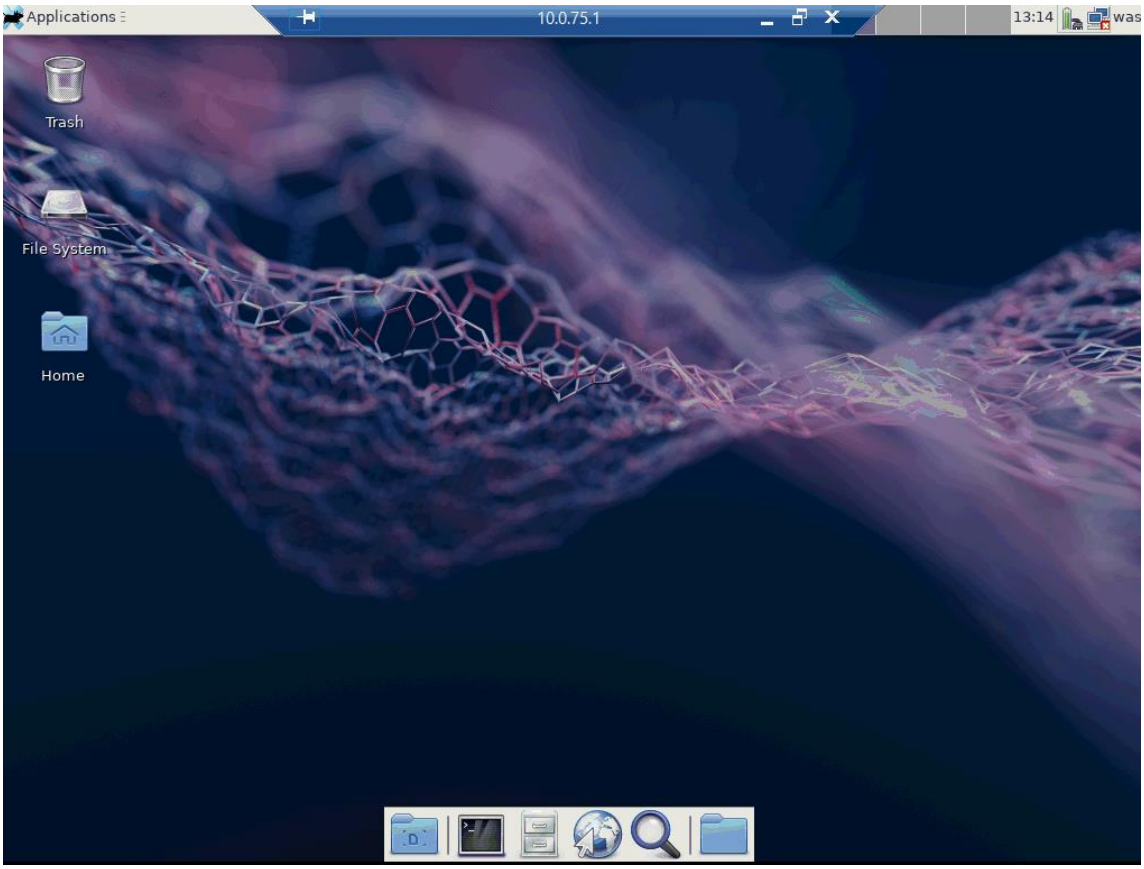
6. Certificate の Warning が出ますが、ワークショップ用のコンテナ接続で、同じ端末内なので "Yes" を押して接続します。



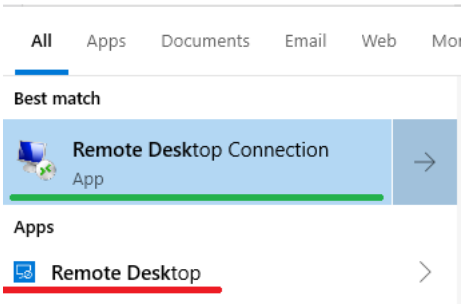
7. ユーザー名に **was** を、パスワードに **websphere** を入力します。



8. コンテナにリモートデスクトップ接続しました。



9. Windows10では、複数のリモートデスクトップのメニューが出ることがあります。動作するものを使ってください。[1]



10. このワークショップでは、Docker コンテナへのリモートデスクトップ接続を可能にするために、一般的な RDP ポート(3389)をコンテナのポート(3390)にマップしています。[2]

[1] https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/clients/remote-desktop-app-compare
[2] https://social.msdn.microsoft.com/Forums/en-US/872129e4-07a5-48c3-86f7-996854e7a920/how-to-connect-via-rdp-to-container?forum=windowscontainers