

**WebSphere Application Server Troubleshooting and Performance Lab on Docker - Lab Preparation**

**Authors**

* Kevin Grigorenko ([kevin.grigorenko@us.ibm.com](mailto:kevin.grigorenko@us.ibm.com))

**Contents**

[1 Lab Preparation 2](#_Toc8801262)

[2 Appendix 6](#_Toc8801263)

[2.1 Windows Remote Desktop Client 6](#_Toc8801264)

# Lab Preparation

1. Install Docker:
   1. Windows ("Requires Microsoft Windows 10 Professional or Enterprise 64-bit.")
      * Download: <https://hub.docker.com/editions/community/docker-ce-desktop-windows>
      * For details, see <https://docs.docker.com/docker-for-windows/install/>
2. Mac ("Requires Apple Mac OS Sierra 10.12 or above")
   * + Download: <https://hub.docker.com/editions/community/docker-ce-desktop-mac>
     + For details, see <https://docs.docker.com/docker-for-mac/install/>
3. For a Linux host, simply install and start Docker (sudo systemctl start docker):
   * + For an example, see <https://docs.docker.com/install/linux/docker-ce/fedora/>
4. Ensure that Docker is started. For example, start Docker Desktop and ensure it is running:  
     
   macOS:  
   

Windows:  
  

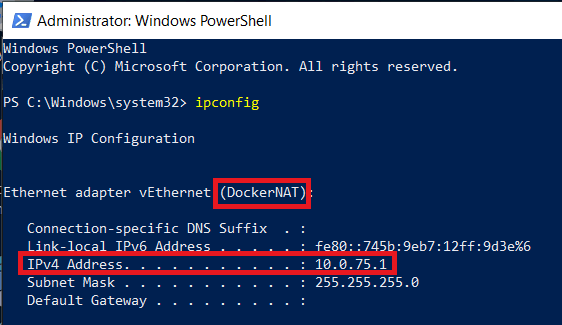

1. Ensure that Docker receives sufficient resources, particularly memory:
   1. Click the Docker Desktop icon and select “Preferences…” (on macOS) or “Settings” (on Windows)
   2. Select the Advanced tab.
   3. Increase Memory, ideally to at least 8GB.
   4. Click Apply  
        
      macOS:  
        
        
      Windows:  
        
      
2. Open a terminal or command prompt:  
     
   macOS:  
     
     
   Windows:  
   
3. Download the images:

docker pull kgibm/fedorawasdebug

* 1. Note that these images are about 20GB. If you plan to run this in a classroom setting, consider performing all the steps up to and including this item before arriving at the classroom.

# Appendix

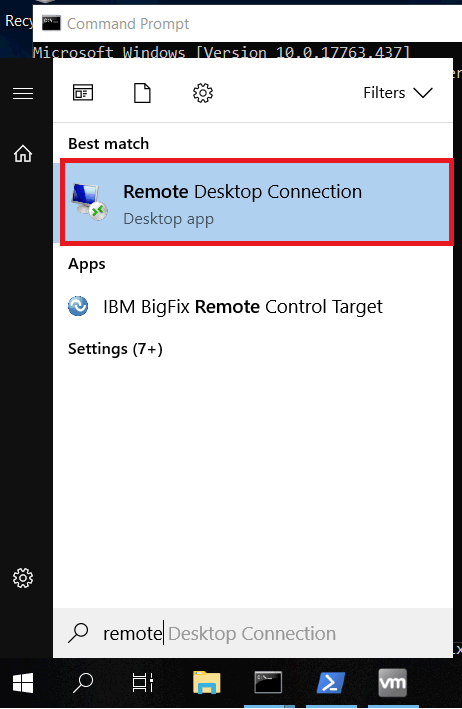
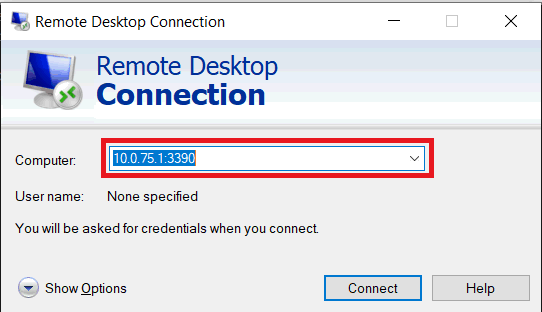
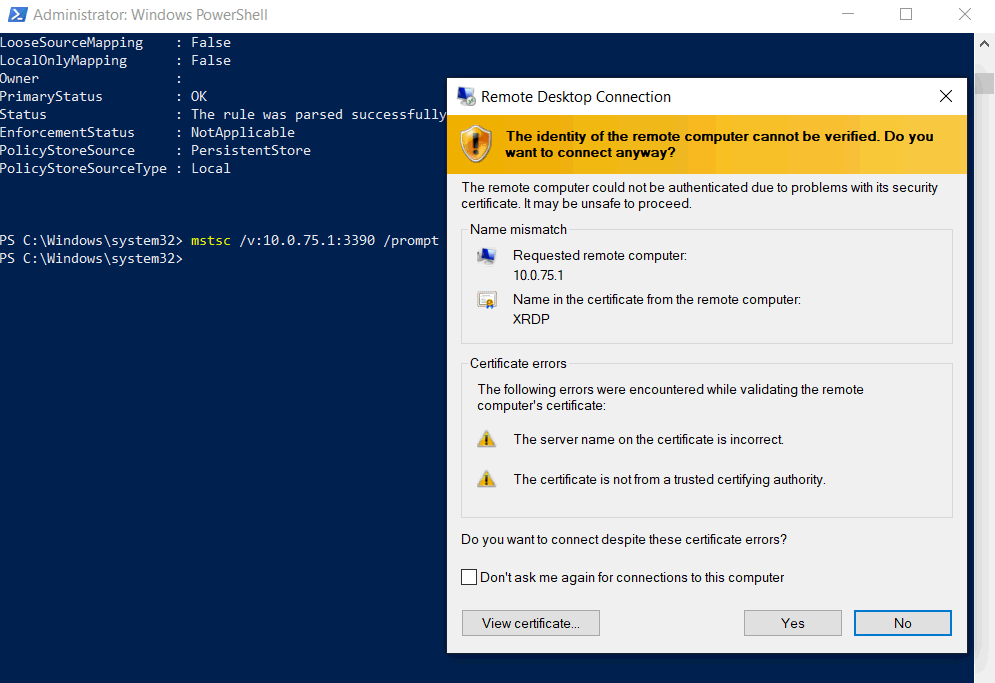
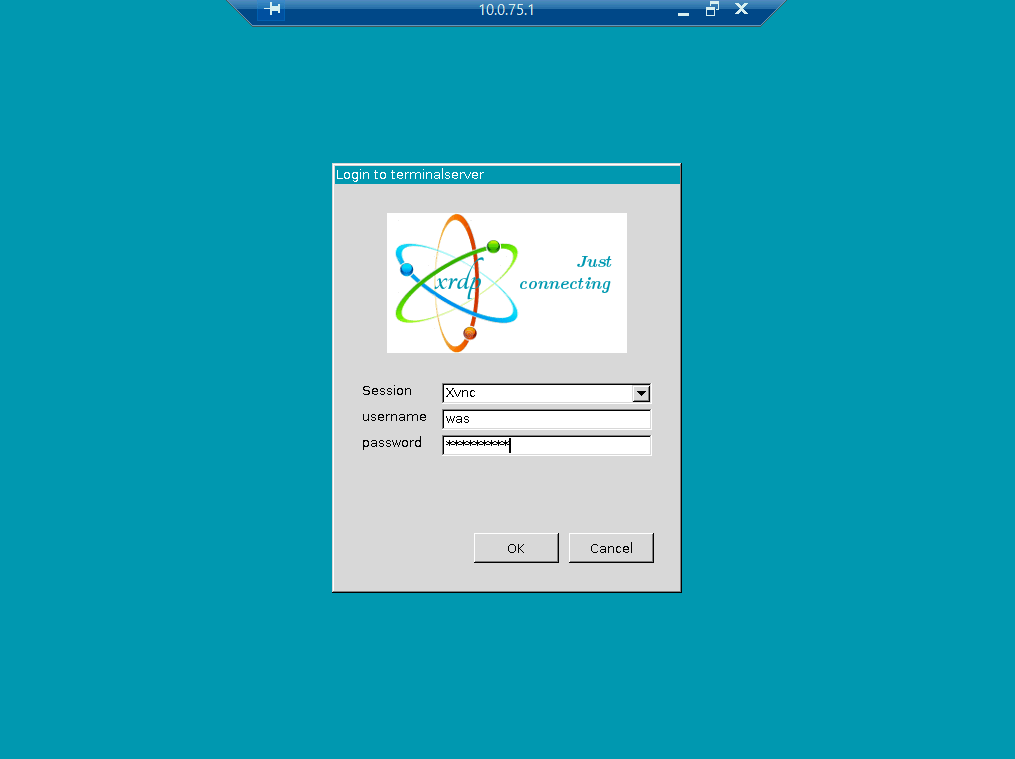
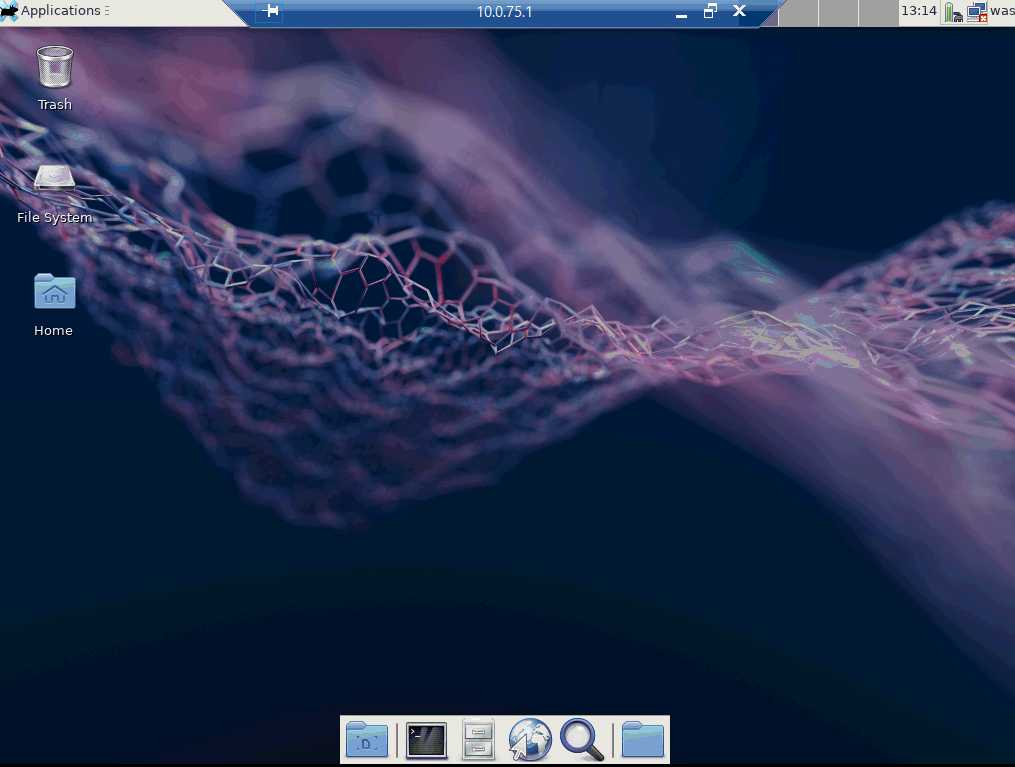
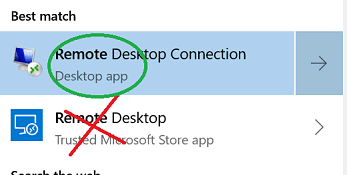
## Windows Remote Desktop Client

* 1. Open PowerShell as Administrator:  
       
     
  2. Run ipconfig and copy the IPv4 address of the DockerNAT adapter. For example:  
       
     
  3. Run the following command in PowerShell:

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

* 1. Run the following command in PowerShell:

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

* 1. Run Remote Desktop  
     
  2. Enter the DockerNAT IP address (for example, 10.0.75.1) followed by :3390 as "Computer" and click "Connect":  
       
     
  3. You'll see a certificate warning because of the name mismatch. Click "Yes" to connect:  
       
     
  4. Type username = **was** and password = **websphere**  
       
     
  5. You should now be remote desktop’ed into the container:  
       
     
  6. Note: In some cases, only the Remote Desktop Connection application worked, and not Remote Desktop:  
       
     
  7. Also note: for an unknown reason, the above instructions do not work on the classic RDP port of 3389.