RDPCodec User Guide

Table of Contents

Introduction

RDP Protocol Tool Set is a set of tools developed for RDP protocol testing. It contains RDP Codec Tool and Image Comparison tool.

This guide will help you to learn how to deploy the Codec web service on Windows and an Ubuntu machine.

Requirements

RDPCodec support with multiple operating systems, you can deploy it on operating systems which supports .NET 5.0.

Requirement	Description
Operation System	The operating system that can install .NET 5.0
Memory	2GB RAM
Disk space	10 GB of hard-driver free space
Internet access	Require internet access to install required package

Setup steps

1. Prepare an Ubuntu 18.04 or windows 10 VM and can access internet.

For windows:

- Windows 7 SP1 is supported with Extended Security Updates installed.
- Windows 10 1607 is the minimum version for support. See Out of support OS versions below for Windows 10 releases that are no longer supported.

Support Ubuntu version 20.10, 20.04, 18.04, 16.04

2. Install .NET 5.0 SDK and runtime.

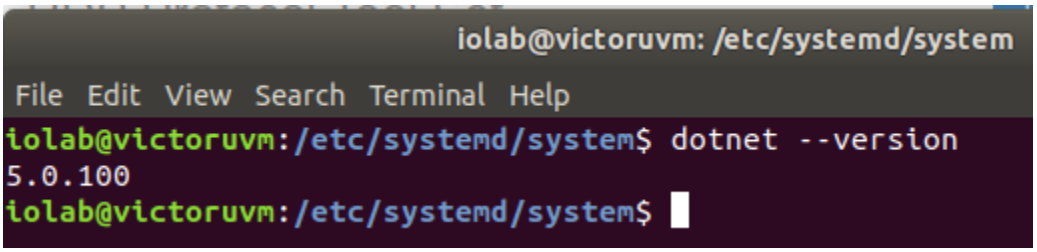
2.1 Install .NET 5.0 on Linux.

Follow this document: https://docs.microsoft.com/zh-cn/dotnet/core/install/linux-ubuntu

- Install steps:

```
Step1: Add Microsoft source package to repo:
wget https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb

Step2: Install SDK
sudo apt-get update
sudo apt-get install -y apt-transport-https
sudo apt-get update
sudo apt-get install -y dotnet-sdk-5.0

Install Runtime if need:
sudo apt-get update
sudo apt-get install -y aspnetcore-runtime-5.0
```

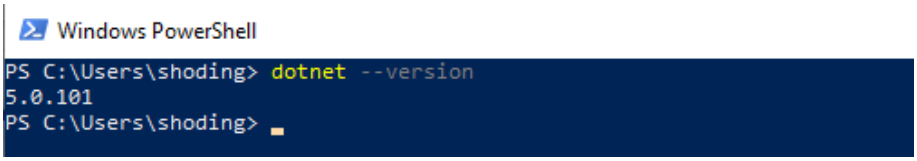
- Check .NET Core version:



2.2 Installation .NET 5.0 on Windows

Follow this document: https://docs.microsoft.com/zh-cn/dotnet/core/install/windows

- Download .NET 5.0 SDK and install the msi step by step.
- Check .NET Core version:



3. Clone and Build RDPCodec project.

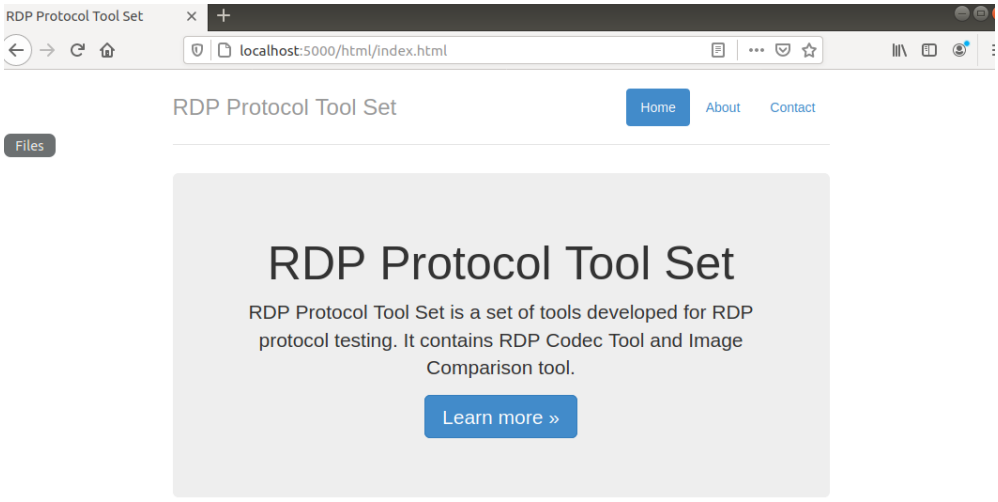
- Clone RDPCodec code from our github portal
- Locate to folder "WindowsProtocolTestSuites\TestSuites\RDP\Tools\RDPToolSet"
- Build and publish with below commands:

```
dotnet restore
dotnet build --configuration Release
dotnet publish -c Release
```

4. Copy binary files to your host machine and check if RDPCodec can work.

Execute dotnet run to check your app can work correctly in Ubuntu system.

```
dotnet RDPToolSet.Web.dll
```

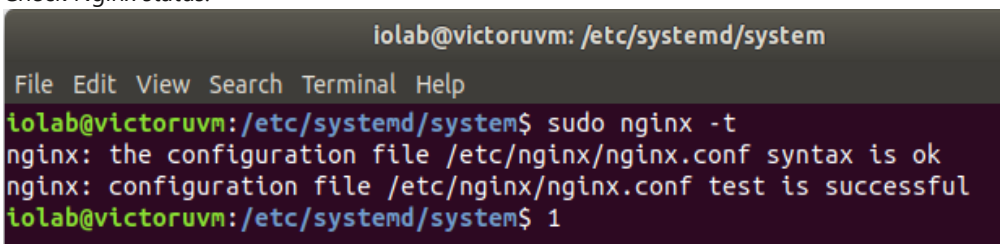


5. Install and configure web proxy tool.

5.1 Install Nginx on Linux platform:

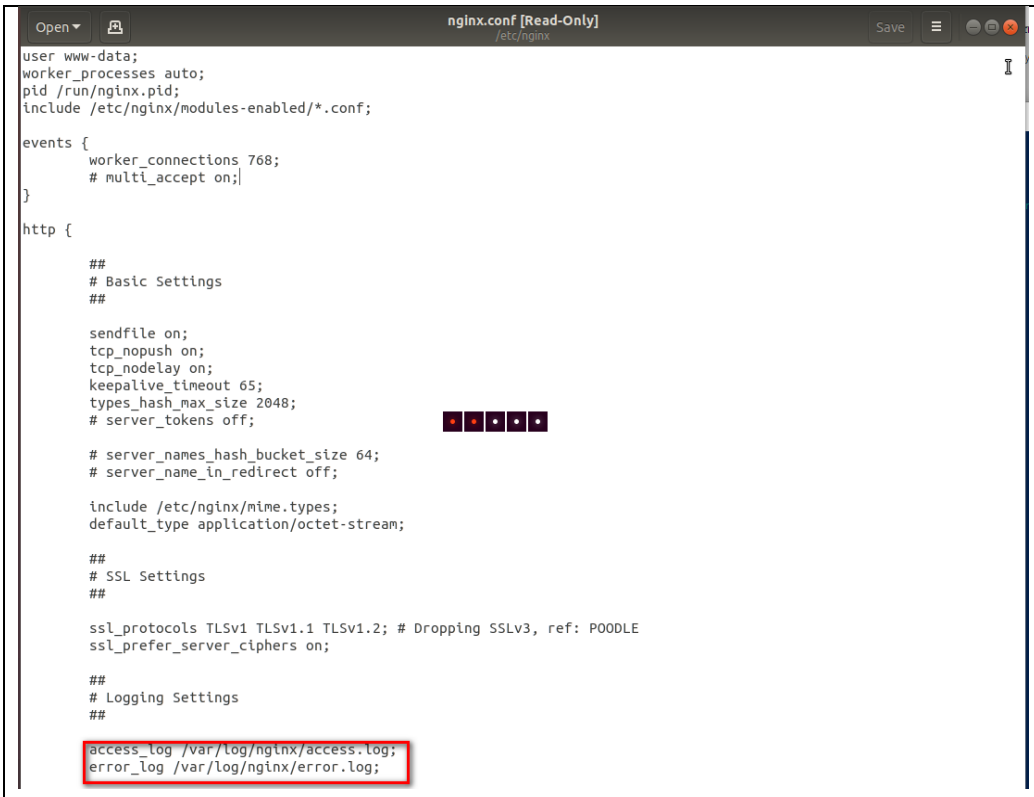
- Install Nginx:
  sudo apt-get install nginx

- Check Nginx status:



- Configure Nginx:

edit nginx.conf sudo vim /etc/nginx/nginx.conf



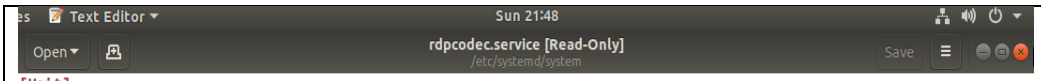
sudo vim /etc/sites-available/default
edit default as below:

```
server {
    listen 80;
    location / {
        proxy_pass http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection keep-alive;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Save above changes and execute below command:
sudo nginx -s reload

- Create the service file.
  The server is setup to forward requests made to http://<serveraddress>:80 on to the ASP.NET Core app running on Kestrel at http://127.0.0.1:5000. However, Nginx isn't set up to manage the Kestrel process. systemd can be used to create a service file to start and monitor the underlying web app. systemd is an init system that provides many powerful features for starting, stopping, and managing processes.

```
sudo vim /etc/systemd/system/rdpcodec.service
```



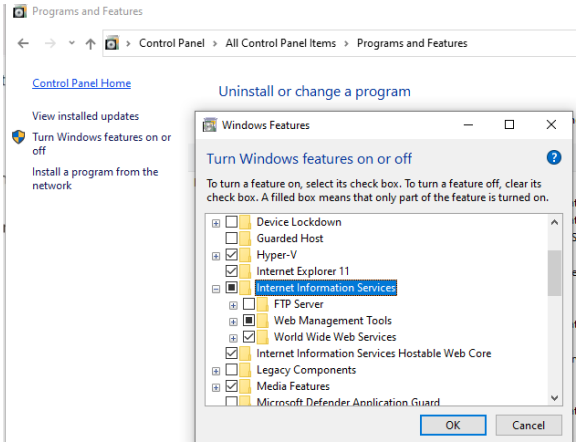

```
[Unit]
Description=RDP Codec on linux

[Service]
WorkingDirectory=/home/iolab/www
ExecStart=/usr/bin/dotnet /home/iolab/www/RDPToolSet.Web.dll
Restart=always
# Restart service after 10 seconds if the dotnet service crashes
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=rdpcodec
User=root
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false

[Install]
WantedBy=multi-user.target
```

Save the service file and enable the service.
`sudo systemctl enable rdpcodec.service`

Check the output and verify service is running, any error you can check the log file which configured in /etc/nginx/nginx.conf file.

5.2 Install and config IIS on Windows platform:

Open Programs on windows and then click “Turn Windows features on or off”

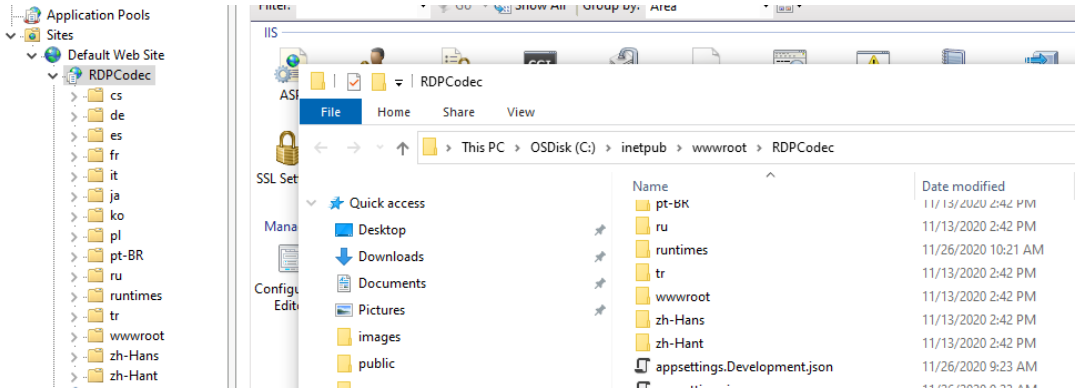


Create the IIS site: on the IIS server, create a folder to contain the app’s published folders and files. In IIS Manager open the server’s node in the **Connections** panel, Right-click the Sites folder and select **Add Website** from the contextual menu. Then provider a Site name and set the physical path to the app’s deployment folder that you created.

Publish and deploy the app:

- In a command shell, publish the app in Release configuration with the dotnet publish command.
  `dotnet publish --configuration Release`

- Copy the contents of the bin/Release/{TARGET FRAMEWORK}/publish folder to the IIS site folder on the server, which is the site's physical path in IIS Manager.



Check and verify RDPCodec web site.

Open your internet browser and launch RDPCodec site and check if it works.