



ICE Rallypoints

Product guide for prerelease

Copyright © 2024, Instant Connect Software, LLC. All rights reserved.
Document version 1841, produced on Friday, September 06, 2024.

main 90adc8bf40040649230176bbdd465f6261a2d8e0

ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. STA GROUP DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL INSTANT CONNECT LLC OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF STA GROUP OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Trademarks mentioned in this document are the properties of their respective owners.

Contents

1 Document History	5
2 Introduction	5
3 Prerequisites	5
3.1 Operating System	5
3.2 IP Ports	6
3.2.1 TCP	6
3.2.2 UDP	6
4 Installation	7
4.0.1 Prepackaged	7
4.0.2 Manual Installation	7
5 Operation	8

List of Tables

2 Systemd operations 8

1 Document History

Publication Date	Product Release	Notes
May 29, 2024	3.5.1	No updates.
April 15, 2024	3.5.0	No updates.
September 20, 2023	3.4.0	No updates.
July 24, 2023	3.3.0	New release.
December 1, 2022	3.2.0	New release.
September 26, 2022	3.1.2	No updates.
April 18, 2022	3.1.1	Added CentOS 7.
March 15, 2022	3.1.0	Document created.

2 Introduction

Instant Connect uses multicast capabilities inherent in your network to provide a transport mechanism between Engines (and therefore the users of those Engines.). However, what if your network doesn't support multicast or, often the case, you need to communicate over someone else's network - such as the Internet?

That's where ICE Rallypoints come in.

ICE's Rallypoints are small, super-fast, packet routers designed to securely forward packets between ICE clients that are unable to communicate with each other over multicast. So, in our case of where ICE users need to speak with each other over something like the Internet, Rallypoints provide the means to do so.

3 Prerequisites

3.1 Operating System

Rallypoints run on Linux, MacOSX, and Microsoft Windows at this time - with Linux being the preferred platform. Rallypoint Server / VM needs to meet the following minimum specifications.

- 4 CPU Cores

- 8GB RAM
- 64GB Disk Space

Operating Systems

- Ubuntu Linux 18.04 LTS/20.04 LTS
- RHEL 7
- CentOS 7

3.2 IP Ports

3.2.1 TCP

The default inbound TCP port is 7443 and uses TLS v1.3. Make sure this port is open for inbound connections from ICE clients and other Rallypoints through your firewalls and other network infrastructure. (We mention that this is TLS so that if your infrastructure environment conducts deep packet inspection, TLS-passthrough, or other such operation for purposes of DoS attack detection; you can configure it accordingly.)

Also, for environments where load balancers or other network infrastructure systems check on the availability or health of the process by opening TCP connections, the Rallypoint may be configured to listen for inbound connections on that port. If you are operating in such an environment, make sure that the port you configure for this purpose is opened.

3.2.2 UDP

While a Rallypoint fundamentally serves to route packets between entities using TCP, it also supports UDP over multicast. This functionality is provided with the intent of Rallypoints forwarding traffic from unicast TCP to multicast UDP. This capability can be used to create a multicast backbone link between Rallypoints and/or route traffic from non-Rallypoint entities (including multicasting ICE clients) operating on multicast to ICE-based entities using unicast. If you are going to be forwarding multicast traffic over unicast (and vice versa), make sure your Rallypoint machine has its firewall setup for multicast RX and TX and that the necessary UDP ports are opened for inbound and outbound traffic.

4 Installation

4.0.1 Prepackaged

A Rallypoint is most easily installed by using the package manager for your operating system using the appropriate installation package provided by Rally Tactical Systems. These packages will install the necessary binaries, factory default certificates, and a baseline configuration. They will also setup the Rallypoint to operate as a daemon (background service) that starts at operating system boot time. This is done using *systemd* on Linux platforms and *launchd* on OSX.

For Red Hat and CentOS distributions:

```
sudo yum install <rallypoint_package_file>.rpm
```

For Debian-based distributions:

```
sudo apt install <rallypoint_package_file>.deb
```

NOTE: In the above examples we're telling `yum` or `apt` to run the installation from a file. So, to ensure that these tools will try to use the file and not a named package from a repository, you need to tell them you're referring to a *file*. Do this by changing to the directory where the file is located and then preceding the file name with `./`. For example:

```
sudo yum install ./<rallypoint_package_file>.rpm
```

For OSX, open the .dmg file and double-click the install link icon.

4.0.2 Manual Installation

If you need to conduct a more sophisticated installation procedure, need to run the Rallypoint process manually (not as a background daemon for example), or generally just have more complex needs for your Rallypoint setup, you will need to install the relevant items manually. This is a pretty straightforward process so it shouldn't be too difficult.

Let's get going by assuming we're not yet configuring (or perhaps never configuring) to run as a background service.

- Place the `rallypointd` executable anywhere you'd like. This can be in a custom directory or in a standard executable location such as `/usr/sbin`. As long as the code can be executed from that location, you're good to go.
- Place the security-related certificate and key files in a location where the Rallypoint can read them. These include the file containing the Rallypoint's certificate, and the file containing that

certificate's private key. (Be sure that this location is strictly only accessible to the Rallypoint and any other authorized applications and/or users.) You will also need to place CA certificates used to verify client and peer Rallypoint certificates in a location accessible to the Rallypoint.

- Finally, place your configuration file in a location where the Rallypoint can read it. By default, the Rallypoint looks for `/etc/rallypointd/rallypointd_conf.json` for its configuration.

Note: If you do place your configuration file in a different location or give it a different name, then you'll need to tell the Rallypoint to use that file. Do this with the `-cfg` command-line parameter. For example:

```
rallypointd -cfg:my_custom_configuration.json
```

4.0.2.1 Manual Daemon Configuration Now that you've got everything installed manually, you may still want to setup the Rallypoint to run as a daemon on startup and avail yourself of the services offered by the operating system. In a Linux environment, this is easily done by setting up the required configuration for *systemd* or the more old-style *init.rc* method. Refer to your operating system instructions on how to do this.

Setting up the service for *systemd*-like operation on Mac OSX systems is a little more tricky. Your best bet is to refer to Apple's documentation at:

<https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html>

5 Operation

Once the code is installed and configured (more on that below), your Rallypoint should just start up and begin accepting connections from clients and/or other Rallypoint peers. If the code is running as a daemon under *systemd*, you can use the standard *systemd*-related methods of interacting with your daemon. Such as:

Table 2: Systemd operations

Operation	Command Line
Starting the service	<code>sudo systemctl start rallypointd</code>
Stopping the service	<code>sudo systemctl stop rallypointd</code>

Operation	Command Line
Restarting the service	<code>sudo systemctl restart rallypointd</code>
Query service status	<code>sudo systemctl status rallypointd</code>
Watch the log	<code>sudo journalctl -f -u rallypointd</code>
Check Rallypoint Version	<code>apt show rallypointd</code>
Remove deb package	<code>sudo apt remove rallypointd</code>

If you are running `rallypointd` from the command line, you will see the log output in the terminal window. To stop the process, simply press `Ctrl-C` or use the `kill` command to stop the process.