



# NetBeez Tutorial

Release 10.1  
May 2022

## Tables of contents

[About the Tutorial](#)

[Version Changes](#)

[Overview](#)

[Dashboard](#)

[Server \(BeezKeeper\)](#)

[Agents](#)

[Control channel](#)

[Agent Categories](#)

[Hardware agents](#)

[Software agents](#)

[Remote Worker Agents](#)

[Targets](#)

[Scheduled tests](#)

[Server deployment](#)

[On-premise or cloud](#)

[Firewall rules](#)

[Server](#)

[Agents](#)

[Dashboard activation](#)

[Network monitoring](#)

[Targets and real-time testing](#)

[Creating a target](#)

[Path Analysis](#)

[Monitoring a web application](#)

[Monitoring a DNS service](#)

[Monitoring a TCP-based application](#)

[WAN performance monitoring](#)

[Gateway Testing](#)

[Tests Over VPN](#)

[Monitoring network performance with scheduled tests](#)

[Iperf](#)

[Network speed](#)

[VoIP](#)

[QoS](#)

## [WiFi Monitoring](#)

[WiFi monitoring on network agents](#)

[WiFi Networks](#)

[WiFi metrics](#)

[WiFi connection timing](#)

[SSID hopping](#)

[WiFi monitoring on Remote Worker Agents](#)

[Wired tests on WiFi agents](#)

[Packet Capture](#)

## [Anomaly detection](#)

[Alert profiles](#)

[Types of alert profiles](#)

[Percentile-based mean](#)

[Device alerts](#)

[Incidents](#)

[Notifications](#)

[Data retention](#)

[Users](#)

[User authentication](#)

[User Profile](#)

## [Reports and API](#)

[Legacy Reports](#)

[Reports Beta](#)

[Email reports](#)

[API](#)

[Public dashboard](#)

## [Troubleshooting with NetBeez](#)

[Using the Buzz Tab](#)

[Interactive console](#)

[Ad-hoc tests](#)

## [NetBeez Configuration Checklist](#)

## [Resources](#)



## About the Tutorial

Welcome to the NetBeez Tutorial! This guide was written with our users in mind and consolidates many hours spent with our customers deploying and configuring NetBeez. We hope that this tutorial will be a valuable instrument to help tune and optimize your installation. If you feel that some concepts are not well explained, require more information, or are missing, please let us know. Send your feedback and comments to [info@netbeez.net](mailto:info@netbeez.net).

## Version Changes

### [ v1.0 ]

First draft of this tutorial, based on NetBeez release 1.3.

### [ v1.1 ]

Update with new features included in the NetBeez release 1.4 and 1.5, including SSID and WiFi incidents.

### [ v4.1 ]

Update with new features included in the NetBeez release 4.1. Changed the tutorial version number to reflect the latest release.

### [ v5.0 ]

Update with new features included in the NetBeez release 5.0. Changed the tutorial version number to reflect the latest release.

### [ v6.0 ]

Update with new features included in the NetBeez release 6.0 such as Remote Worker agents. Changed the tutorial version number to reflect the latest release.

### [ v7.0 ]

Update with new features included in the NetBeez release 7.0 such as Remote Worker WiFi Metrics and the Jitter/MOS addition under Ping tests. Changed the tutorial version number to reflect the latest release.

### [ v8.0 ]

Update with new features included in the NetBeez release 8.0 such as Path Analysis, Packet Capture and Wired Tests on Wireless Sensors. Changed the tutorial version number to reflect the latest release.

[ v9.0 ]

Update with new features included in the NetBeez release 9.0 such as Endpoint Performance Metrics (CPU/RAM/HDD), ISP Tagging, and Single Sign-On with Azure AD. Changed the tutorial version number to reflect the latest release.

[ v10.1 ]

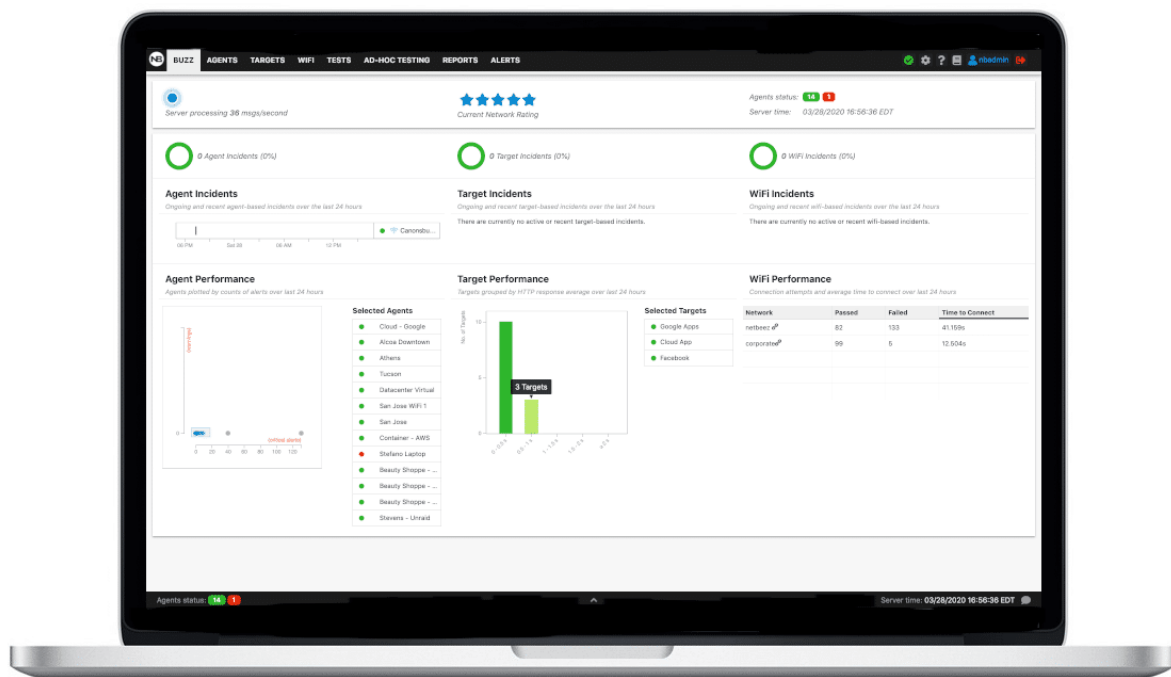
Update with new features included in the NetBeez release 10.0 and 10.1 such as Gateway Testing, Tests Over VPN, and Enhanced Visualization for Scheduled Tests. Changed the tutorial version number to reflect the latest release.

## Overview

NetBeez is a distributed network monitoring platform. The monitoring is performed from multiple points via hardware, software, and cloud agents, called 'Beez'. The agents are managed by a server, called the 'Beezkeeper', that is also located on-premise or in the cloud.

## Dashboard

The dashboard is the graphical user interface that is necessary to manage the agents, create monitoring tests, troubleshoot network or application problems, and receive alerts and reports. The Dashboard runs on a dedicated server, called BeezKeeper.



## Server (BeezKeeper)

The NetBeez server is the main component of the solution. The BeezKeeper is responsible for managing the agents, storing the data received by the agents, generating alerts, incidents, and notifications. The BeezKeeper can be hosted on premises as a virtual appliance or in the cloud, managed by either NetBeez (AWS) or by the customer itself (AWS, Azure, etc.). The amount of CPU, RAM, and disk space required to run the server varies depending on the number of agents managed and the number of tests running. Please consult the online documentation for more details for [server requirements](https://netbeez.net/server-requirements).



## Agents

The agents (also called Beez) serve as the monitoring endpoints that run tests on the network and against the applications or targets. We have three types of agents: wired, wireless, and virtual/software. You can review the installation instructions on [this documentation page](#).

### NETWORK MONITORING AGENTS

Technology Partners:

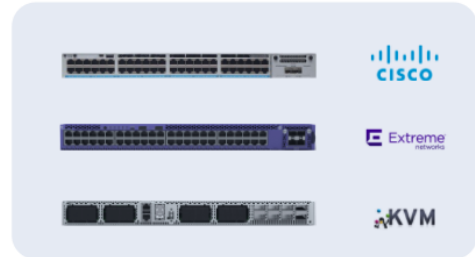


#### ON PREMISES



Wired Sensor    Wi-Fi Sensor    Virtual Machine

#### INTEGRATED

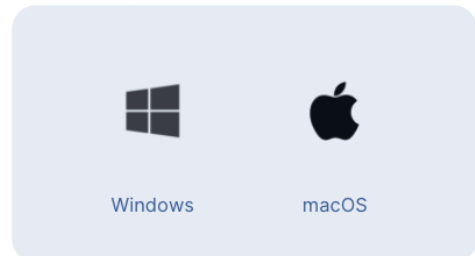


#### CLOUD



Linux    Container    Instance

#### ENDPOINTS



Windows    macOS

## Control channel

The agents are centrally managed through the dashboard to which they send real-time results. This is established via a TCP connection from the agents to the dashboard. By default, the control channel is an encrypted SSL socket to TCP port 20018 on the server.

The connection is initiated from the agents, simplifying the deployment of agents behind a NAT or a firewall. For this reason, each agent's interface can display up to three addresses:

- **IPv4 address:** This is the address associated with the Ethernet or WiFi interface.
- **IPv4 external:** This is the public IPv4 address, available if the agent is behind NAT.
- **IPv6 address:** This is the IPv6 address associated with the Ethernet interface.

### Control Interface (eth0)

```

IPV4 ADDRESS: 10.0.0.73
IPV4 EXTERNAL: 73.174.81.192
IPV6 ADDRESS: 2601:541:4303:6860::3a52/128
MAC ADDRESS: b8:27:eb:48:1c:7b
TX RATE: 20.12 Kbps
RX RATE: 6.38 Kbps
SPEED/DUPLEX: 100-full
  
```

## Agent Categories

NetBeez Agents are divided into two categories, Network Agents and Remote Worker Agents.

- Network Agents: Hardware sensors (GigE, Wi-Fi), virtual machines, Docker containers, cloud instances (AWS), and Linux packages for monitoring on-prem, cloud, and remote environments.
- Remote Worker Agents: Native applications for Windows, macOS, and IGEL OS systems installed on end-user desktops and laptops for monitoring work-from-home or remote users.

### Network Agents - Physical appliances

Hardware agents are convenient for plug-and-play deployments without software installation. Agents come pre-configured by NetBeez before being shipped to the user. All that is required is an Ethernet switch port.

Agents are powered either by PoE (Power over Ethernet) or by an external PSU (Power Supply Unit). A hardware agent can monitor via the Ethernet or the WiFi interface, based on the model selected. There are two models of hardware agents:

#### **Wired (GigE) - 10/100/1000Mbps**

The wired agent simulates an Ethernet client and can run throughput tests up to 1 Gbps. This agent cannot do WiFi testing.

#### **Wireless (WiFi) - 802.11ac**

The wireless agent simulates a WiFi client with an 802.11ac interface and can run throughput tests up to 150 Mbps. The control channel is established via an Ethernet connection. If available, otherwise via the WiFi interface.



A NetBeez WiFi sensor.

### Network Agents - Software-based

Software agents can be deployed as a virtual appliance, a Docker container, a Linux package for Debian and Ubuntu systems, or a cloud image for AWS (for Azure and other cloud providers, we recommend using the Linux package).

#### **Virtual agent**

A virtual agent is deployed at data centers or on any equipment that supports virtualization. NetBeez supports VMware, Microsoft HyperV, KVM, and vBox hypervisors. A virtual agent at the datacenter is used, for example, to measure connectivity and throughput from remote WAN locations. Virtual agents are also deployed at remote sites, hosted on networking hardware for example.



### **Linux agent**

The Linux agent is installed via the apt-get command on distributions such as Debian, Ubuntu, and Raspbian. The Linux agent is convenient to enable NetBeez network monitoring on an existing Linux host, such as a server, workstation, or single-board computers such as Raspberry Pi and Odroid.



### **Docker Agent**

The Docker agent is available via the [NetBeez Docker hub page](#). A NetBeez Docker agent is installed on a server, on an end-user desktop, or networking hardware (e.g. Cisco Catalyst Series switches with AppHosting). When deployed on a Mac OS or Windows 10 Professional or Enterprise system, it monitors the network performance of remote workers.



## Remote Worker Agents

Remote Worker Agents are designed to support employees who are working from home. These agents support:

- Real-time tests: ping, TCP ping, DNS, HTTP/S, Traceroute, Path Analysis.
- Scheduled tests: VoIP, Iperf, and Internet speed.
- Up to 20 real-time tests and 3 scheduled tests.
- Wi-Fi metrics for WLAN monitoring.

The remote worker agents can be installed on Windows, Mac, and IGEL systems.

### Windows Agent

The Windows Remote Worker agent is an executable program that runs on Windows Desktops and Laptops. Currently, both Windows 7 and 10 versions are supported. You can download the latest version of the executable (in MSI format) [here](#).



### Mac OS Agent



### IGEL Agent

IGEL OS is a Linux-based operating system for x86 machines built with industry-standard components, regardless of manufacturer, platform-independent. IGEL OS is installed on a variety of PCs, laptops, tablets, thin clients, and most every other x86-64 device.



## Targets

A target is a web application or TCP-based service that is monitored from one or more agents running **real-time tests** such as ping, DNS, HTTP, traceroute and path analysis. A target is defined by one or more resources, which are defined by IP address, Fully Qualified Domain Name (FQDN), or URL. Each resource has its own assigned tests and alert profiles. We'll talk more about targets in the [Monitoring](#) section of this tutorial.

Google Apps

Tests: 55 55 0 0 0 0  
Agents: 5 5 0

### Incidents

Last 24 hours shown

There are currently no active or recent incidents.

### Target Resources

drive.google.com

PING

DNS

HTTP

TRCRT

mail.google.com

HTTP

PING

DNS

TRCRT

maps.google.com

PING

DNS

HTTP

## Scheduled tests

NetBeez supports three categories of scheduled tests: lperf, speed test, and VoIP. Different from a real-time test, which is defined by a testing interval, a scheduled test runs less frequently according to a schedule defined by the user.

**Schedule**

Every

At  minute(s)

**Human Readable**  
Every hour.

We'll talk more about scheduled tests in the [Monitoring](#) section of this tutorial.

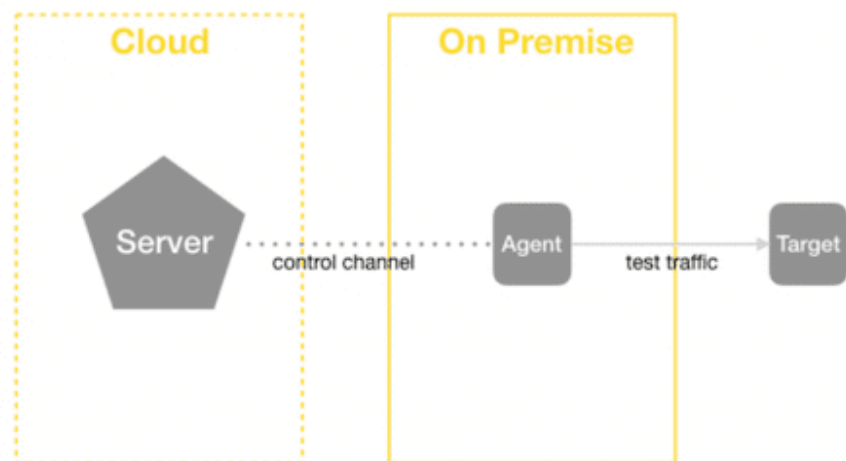
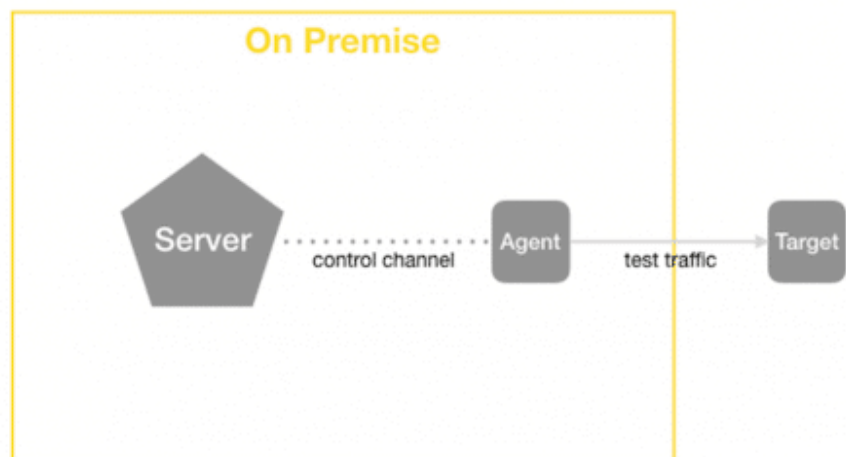
## Server deployment

### On-premise or cloud

The user has the option to run the BeezKeeper server on-premise or in the cloud.

An on-premise server requires a virtualization environment. The BeezKeeper is delivered as a pre-configured appliance that includes the operating system and the NetBeez software. The IP settings are provided by the customer, who's required to install a DNS entry associated with the BeezKeeper's FQDN.

Cloud installations are performed on the customer's cloud account. NetBeez will share privately the server's AMI (in the case of AWS). Installations on other public cloud providers are run on an Ubuntu server.



## Firewall rules

Firewall rules are needed to permit the agents to connect to the server and the server to download software updates from the public NetBeez repository. The BeezKeeper also supports software updates via an HTTP proxy if the network firewalls deny outbound access.

### Server

The BeezKeeper requires the following firewall rules to access NetBeez software repository:

- Outbound access to TCP port 80 to any host
- Outbound access to TCP port 443 to any host

These rules can't be limited to a specific host because the IP address of the NetBeez software repository often changes. If the above requirements are too relaxed, consider using an HTTP proxy server. To enable this option, please contact support.

## Agents

The agents require connectivity to the BeezKeeper to establish the control channel and download the latest software update when available. Below are two firewall rules that must be applied to the network in the presence of firewalls:

- Outbound access to TCP port 443 to the BeezKeeper's IP address for the software update
- Outbound access to TCP port 20018 to the BeezKeeper's IP address for the control channel

Please remember that Docker and Linux agents are updated using the regular update process for Linux packages and Docker containers.

## Dashboard activation

Once the BeezKeeper and the Beez have been deployed, the license activated with the [first-run wizard](#), it's time to point your browser to the NetBeez dashboard and review the configuration settings to make sure the system meets your requirements. Let the buzz begin!



## Network monitoring

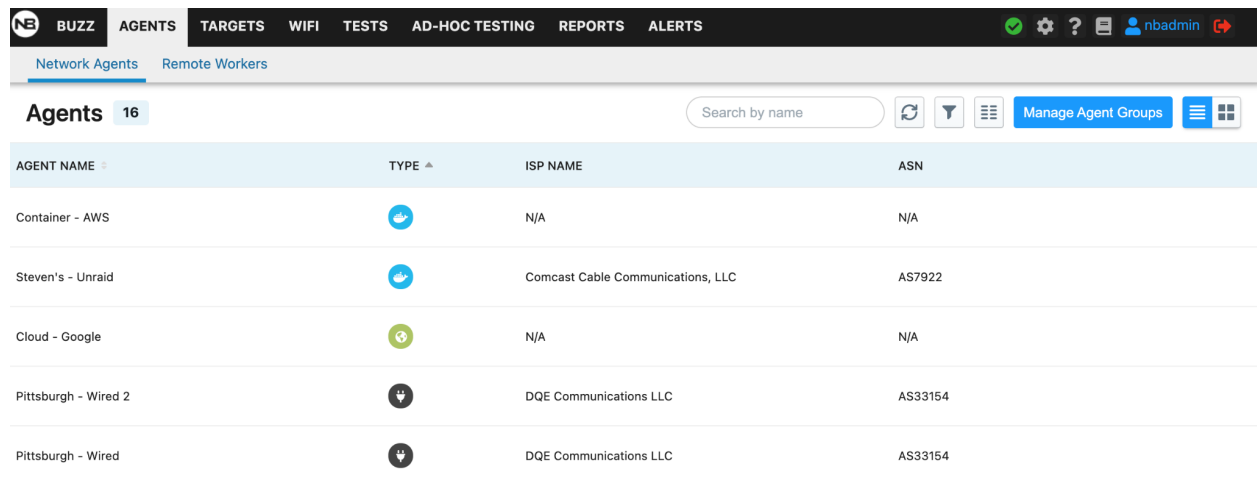
In this section, we'll set up NetBeez to monitor in real-time some applications and run periodic network performance tests.

### ISP Tagging

When an agent connects to the NetBeez dashboard it reports its external IP address. With that information, the NetBeez dashboard then applies the following two tags to an agent:

- ISP name, which is the name of the internet service provider that the remote user connects to.
- Autonomous System Number (ASN), which is a specific network number that belongs to that specific ISP.

With ISP tagging NetBeez users can filter agents based on these two tags, and reduce troubleshooting time to identify ISP issues affecting a set of remote users.

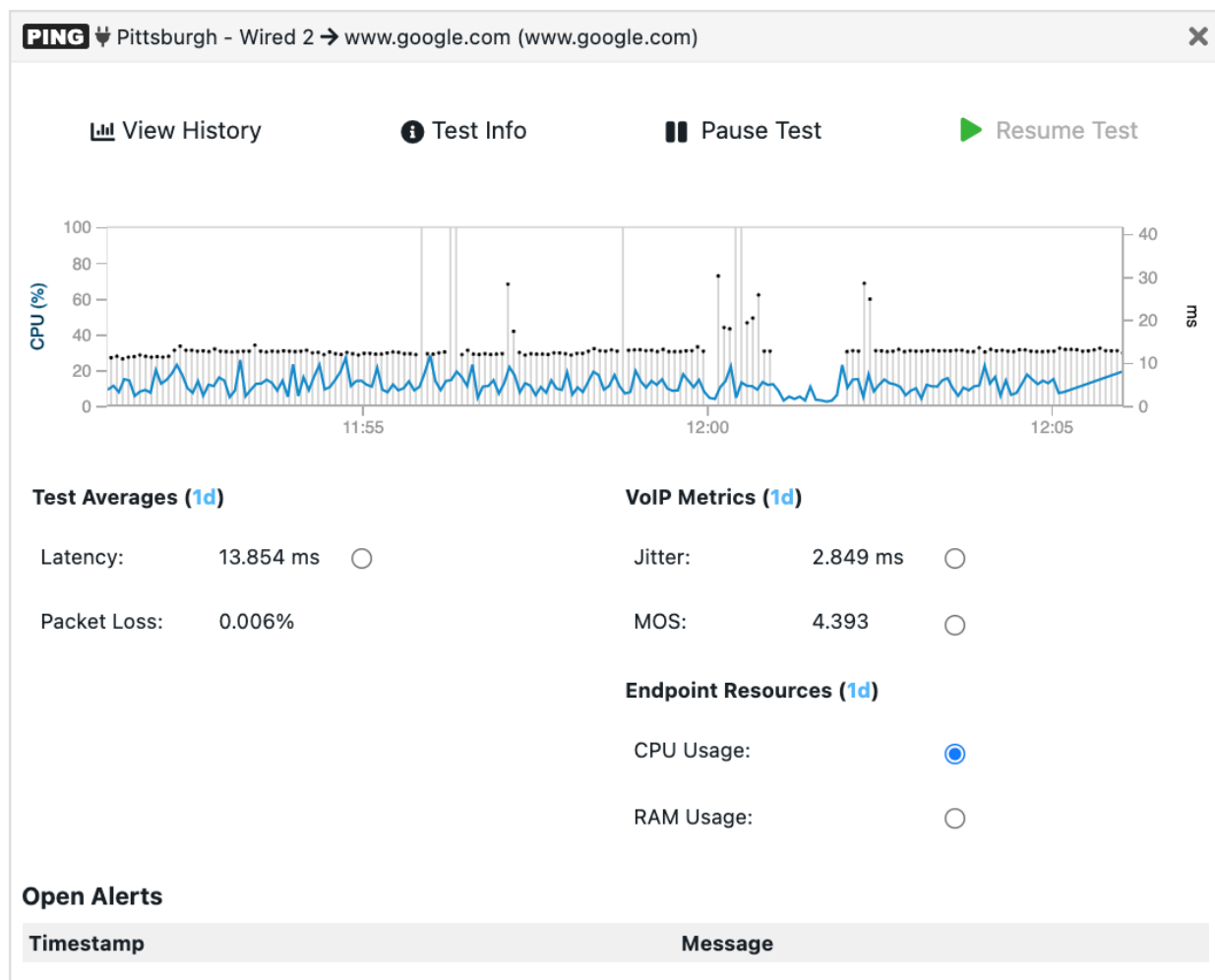


AGENT NAME	TYPE	ISP NAME	ASN
Container - AWS		N/A	N/A
Steven's - Unraid		Comcast Cable Communications, LLC	AS7922
Cloud - Google		N/A	N/A
Pittsburgh - Wired 2		DQE Communications LLC	AS33154
Pittsburgh - Wired		DQE Communications LLC	AS33154

Read more about ISP Tagging in our [documentation](#).

### Endpoint Performance Metrics

Device performance metrics are available on remote endpoints and network agents. On the agent details, you are able to view CPU, RAM, and HDD space. CPU and RAM data can be viewed on Ping, DNS, and HTTP tests' real-time and historical graphs. With this information it will be easier to troubleshoot remote end-user experience issues caused by endpoint performance degradation.



Read more about endpoint metrics in our [documentation](#).

## Targets and real-time testing

To monitor an application you need to create a target using its FQDN, IP, or web address and select the tests that will be included. The tests are selected based on the type of application, or service, that is monitored. The minimum test interval can be set to one second to reduce the time to detect problems as well as to have enough granular data to understand the behavior of the network and applications (except for Path Analysis where the minimum interval is 60 seconds).

Currently, NetBeez supports five types of real-time tests: Ping, DNS, HTTP, traceroute, and path analysis.

- **Ping** - Ping can be run as an ICMP echo request or as a TCP socket connection. The ICMP command supports extra parameters such as: Maximum Transmission Unit (MTU), Don't Fragment (DF) bit, DSCP value. The TCP supports various connection parameters like SYN, ACK, FIN, PUSH, ... Jitter and Mean Opinions Score (MOS) can be enabled on ping tests to help monitor online voice and video conferencing performance. Ping tests should have their interval be equal to 5 seconds or less.
- **DNS** - A DNS test executes a lookup for a given FQDN. This test is implemented with the dig command. The DNS supports the DNS server as an optional parameter. If left empty, each agent will use the assigned DNS server.
- **HTTP** - The HTTP test consists of a GET request, implemented with the command curl. The command supports both HTTP and HTTPS pages, custom URLs, requests via proxy servers (authenticated and non), basic and NTLM authentication for pages.
- **Traceroute** - Traceroute tests support the ICMP, TCP, and UDP protocols. This command supports extra parameters: Destination port number, max hops, timeouts per hop, and DSCP value.
- **Path Analysis** - Path Analysis augments the existing traceroute capability for more accurate, real-time visibility on equal-cost multi-path network topologies. It is advised to add a ping and/or HTTP test for better analysis.

The below table reports the default timing intervals associated with tests included in a target. Test intervals can be adjusted anytime by the user.

Test	Runs every (default)
Ping	5 seconds
DNS	30 seconds
HTTP	60 seconds
Traceroute	120 seconds
Path Analysis	300 seconds

In the rest of this section, we'll review some target templates that can be used to monitor specific applications.

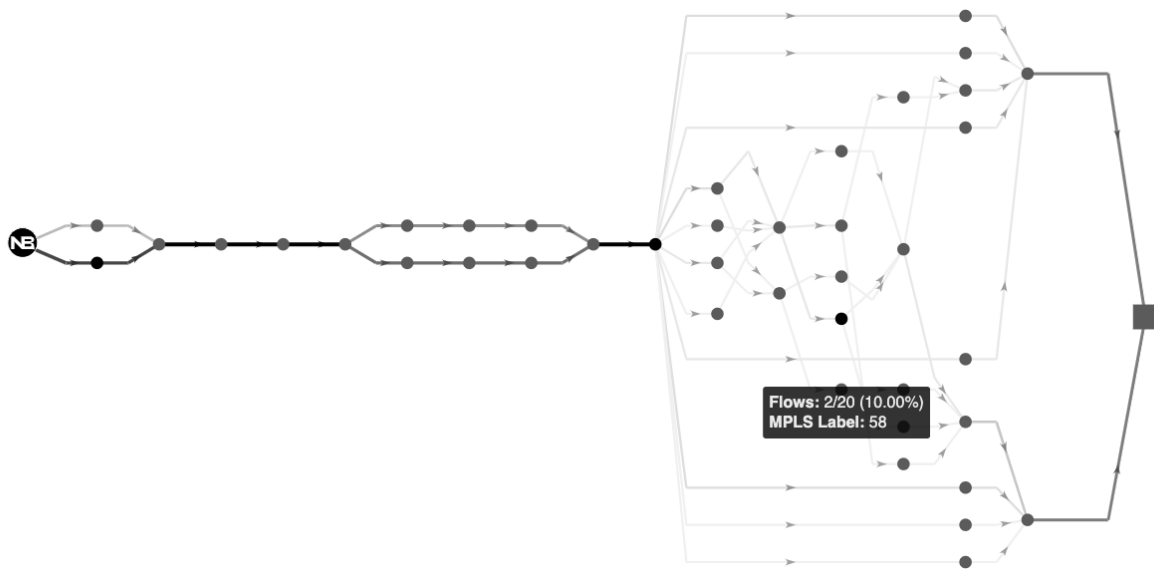
### Creating a target

In the following sections, we'll report some strategies that can be used to monitor web applications and network services in real-time. Before creating a target, it's a good practice to verify on the ad-hoc tab that the tests are correctly configured. Common configuration mistakes include:

- The monitored service doesn't allow ping tests; if that's the case, verify if it's reachable via a TCP-based ping.
- The web URL specified in an HTTP test has a redirection or doesn't allow access to authenticated users (consider configuring HTTP basic or NTLM authentication).
- The FQDN or DNS name is incorrect.

## Path Analysis

The Path Analysis feature augments the existing traceroute capability for more accurate, real-time visibility on equal-cost multi-path network topologies, such as the Internet.



For each hop, path analysis provides extensive information that is valuable for network analysis and troubleshooting, such as:

- IP address and reverse DNS (if available)
- Round-trip time (RTT) real-time and historical
- Color coding of the RTT value (orange if above 100 ms. and red if above 150 ms.)
- Autonomous System Number (ASN), and AS Name
- Geo-IP location with coordinates

The path analysis plot also enables the user to highlight hops based on RTT, IP, DNS, and ASN, making it easier to troubleshoot performance issues with specific nodes, as well as aggregating nodes based on ASN and DNS domains to better understand the high-level topology your users traverse to reach their destination.

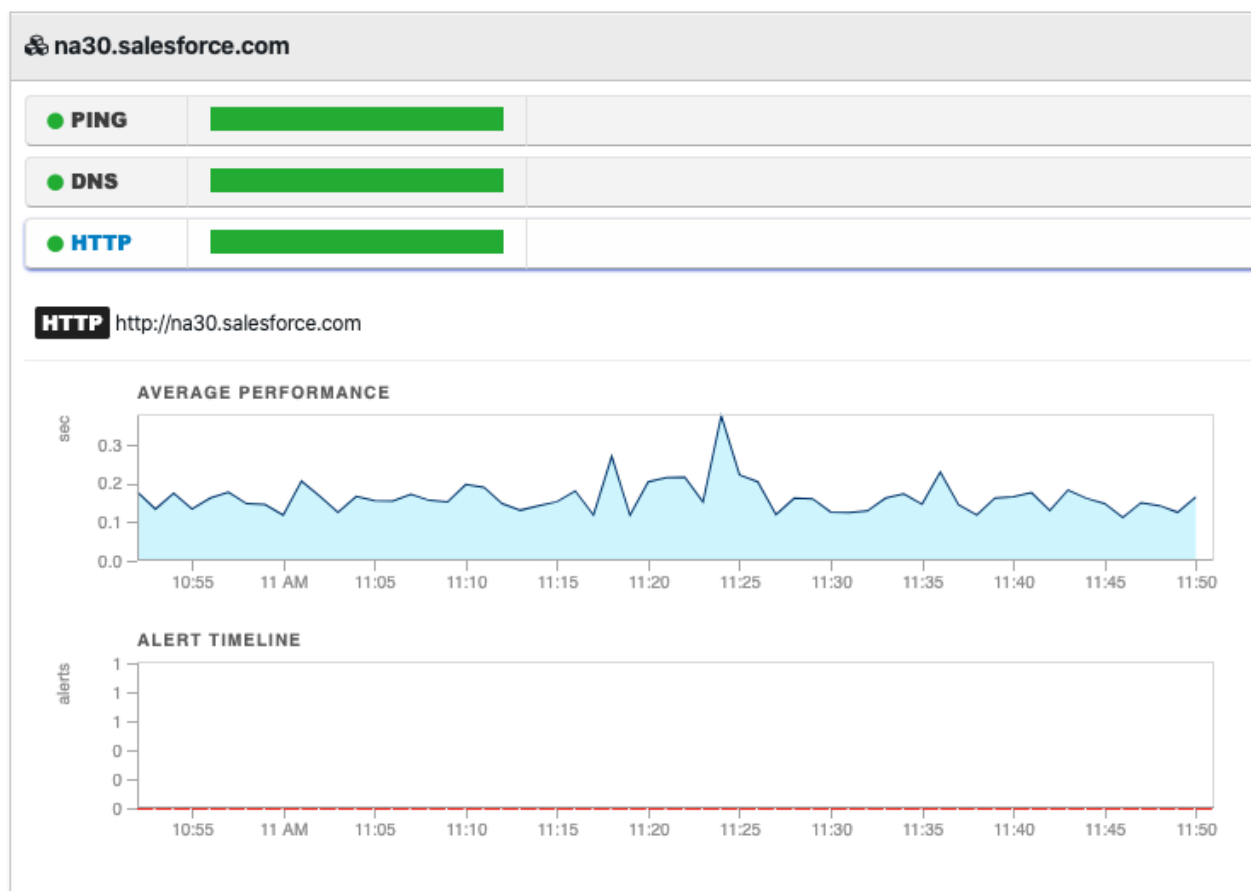
Visit our [documentation](#) for more information on how to set up Path Analysis.

## Monitoring a web application

To monitor a web service include the following tests:

- Ping - Reports the round-trip time and packet loss to the server where the application is hosted; if ICMP is not allowed, the user can configure a TCP-based ping test to the remote host's TCP port 80 or 443.
- DNS - Verifies that the web service's FQDN is working and the end-user clients can resolve the FQDN associated with the application.
- HTTP - Performs an HTTP(S) GET to the URL provided in the address, verifying that the service is available to the users.

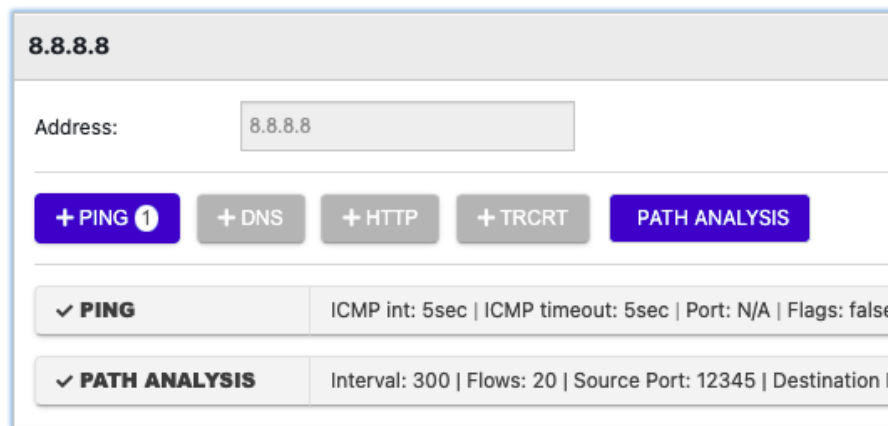
By comparing the status and performance of these three tests from multiple agents, NetBeez can determine whether a problem is related to the network, the webserver, or the DNS. In the [anomaly detection](#) section of this manual, we'll cover in detail alerts, incidents, and notifications.



## Monitoring a DNS service

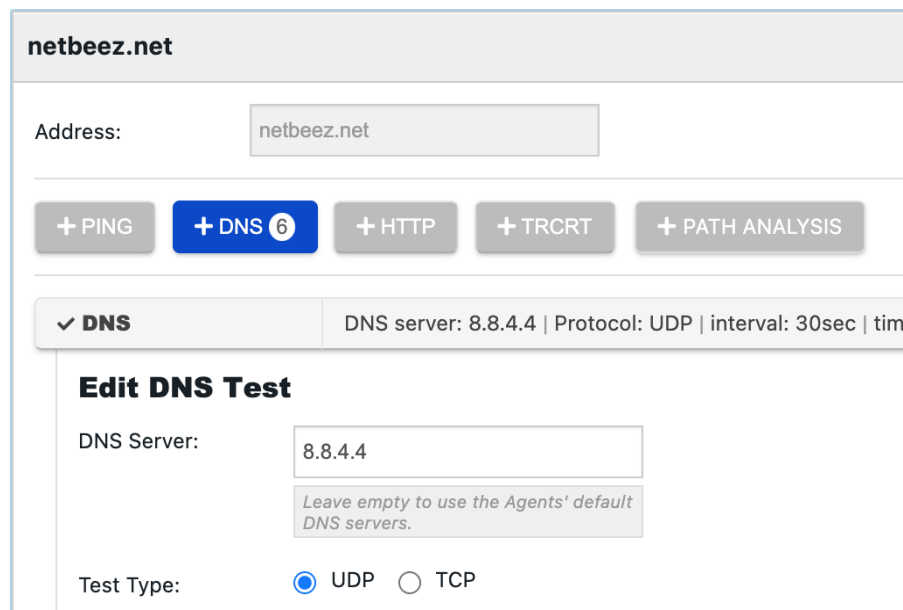
When monitoring a DNS server we want to make sure that those DNS servers are reachable by the end-users and that the DNS service on the servers is correctly working (responding to DNS requests). For this reason, we'll create the following resources:

- One resource for each DNS server to be monitored; set the DNS server's IP address as destination, and include ping and traceroute or path analysis tests.



The screenshot shows the NetBeez configuration interface for a resource named "8.8.8.8". The "Address" field is set to "8.8.8.8". Below the address, there are buttons for "+ PING 1", "+ DNS", "+ HTTP", "+ TRCRT", and "PATH ANALYSIS". The "PATH ANALYSIS" button is highlighted. Below these buttons, there are two expandable sections: "✓ PING" and "✓ PATH ANALYSIS". The "PING" section shows settings: "ICMP int: 5sec | ICMP timeout: 5sec | Port: N/A | Flags: false". The "PATH ANALYSIS" section shows settings: "Interval: 300 | Flows: 20 | Source Port: 12345 | Destination: 8.8.8.8".

- One resource (or more if needed) to verify that the DNS resolution process is working; set as destination one FQDN, then create a DNS test for each one of the DNS servers to be monitored. In the DNS test settings, specific the IP address of the DNS server



The screenshot shows the NetBeez configuration interface for a resource named "netbeez.net". The "Address" field is set to "netbeez.net". Below the address, there are buttons for "+ PING", "+ DNS 6", "+ HTTP", "+ TRCRT", and "+ PATH ANALYSIS". The "+ DNS 6" button is highlighted. Below these buttons, there is an expandable section "✓ DNS" showing settings: "DNS server: 8.8.4.4 | Protocol: UDP | interval: 30sec | tim". Below the "DNS" section, there is a section titled "Edit DNS Test". In this section, the "DNS Server" field is set to "8.8.4.4" with a note: "Leave empty to use the Agents' default DNS servers." The "Test Type" is set to "UDP" (selected) and "TCP" (unselected).

## Monitoring a TCP-based application

Other applications can be monitored using a TCP-based ping test, which verifies that a specific TCP/IP port is open and how long it takes to establish a connection. This test, complemented with ICMP-based ping provides good data to verify the status and health of a generic TCP-based application.

**171.16.233.2**

Address:

+ PING 1

+ DNS

+ HTTP

+ TRCRT 1

+ PATH ANALYSIS

✓ PING

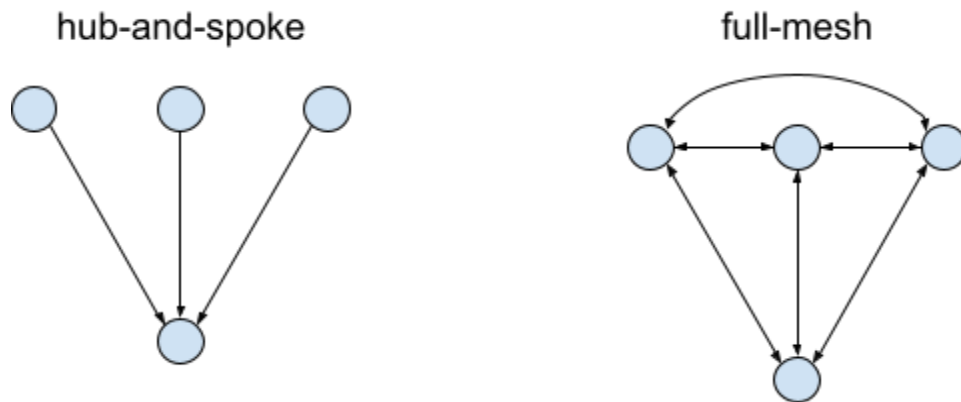
TCP int: 60sec | TCP timeout: 5sec | Port: 3306 | Flags: -

✓ TRCRT

interval: 120sec | TR type: TCP | port: 443 | hop timeout:

## WAN performance monitoring

To monitor the performance of a Wide Area Network, users can set up a hub-and-spoke or a full-mesh target. A hub-and-spoke target is used to verify connectivity and performance from remote sites to one or more WAN routers (aggregation points). A full-mesh target is used to verify connectivity and performance from each remote site to another remote site. The former is easier to set up and requires fewer tests than the latter. The latter is more complete because it takes into account the source and the destination. Users should consider the pros and cons of each solution before picking one.



To set up a hub-and-spoke target, create one resource for each WAN router, using the WAN router's loopback IP as a destination, and select ping and traceroute tests. As an alternative to the router's loopback, a dedicated hardware or software sensor connected to a WAN router can be used as a destination. Lastly, include the agents that are located at the remote WAN locations as monitoring endpoints.

In a full-mesh target, create one resource for each agent deployed at a WAN site, include ping and traceroute tests, and select all WAN agents as monitoring endpoints. Please keep in mind that in a full-mesh target with  $N$  locations,  $N * (N - 1)$  ping and traceroute tests will be created. In the case of a large  $N$ , users may want to reduce the test interval to reduce the test traffic on the network.

## Gateway Testing

Gateway testing gives the ability to define a generic '\_gateway\_' target that the agent translates locally to its default gateway. The remote worker agents test the wired gateway if it is connected via Ethernet, otherwise, they test the wireless gateway if it is connected via WiFi. The network agents can test both wired and wireless gateways simultaneously depending on how the target is configured.

Only ping tests can be added to targets using gateway testing; DNS, HTTP, Traceroute, and Path Analysis are disabled.

\_gateway\_

PING 1

DNS

HTTP

TRCRT

PATH ANALYSIS

⚙️

🗑️

Address:

\_gateway\_

Label:

Enter a human-readable label (optio

+ PING 1

+ DNS

+ HTTP

+ TRCRT

+ PATH ANALYSIS

✓ PING

ICMP int: 5sec | ICMP timeout: 5sec | Port: N/A | Flags: false | MTU: 54 | DF: false | TOS: none |

⚙️

🗑️



You can review setting up gateway testing on [this documentation page](#).

## Tests Over VPN

Targets can be configured to run tests only when the VPN interface on assigned agents is connected. This helps eliminate false positive alerts in NetBeez when performing tests that only exist while connected to a VPN.

Agents that do not have a VPN interface and are assigned to a target running tests on VPN, no tests will be run until the endpoint detects a VPN interface. Currently, only remote worker agents support testing over VPN.

Tests on the vpn interface				
DESTINATION	PING	DNS	HTTP	TRA...
NetBeez VPN (172.29.0.1(172...	?			?

You can review setting up tests over VPN on [this documentation page](#).

## Monitoring network performance with scheduled tests

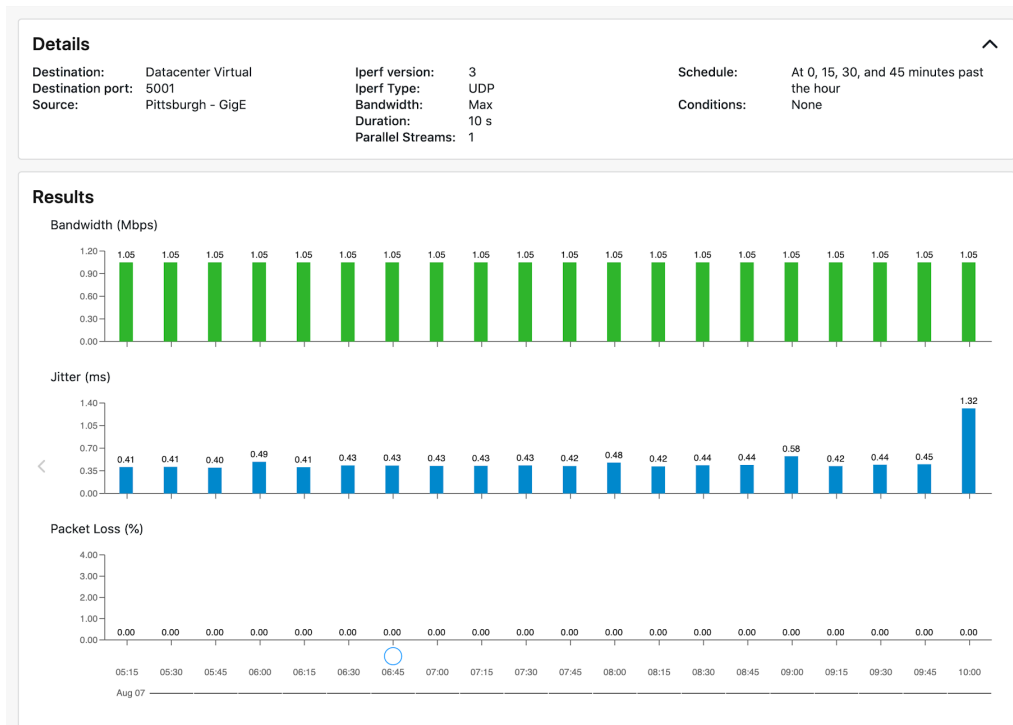
NetBeez supports three types of scheduled tests: Iperf, network speed, and VoIP. The goal of scheduled tests is to build a baseline for network performance on download, upload, and call quality. The schedule is user-defined and periodic: it can be hourly, daily, weekly, or custom, running on specific hours throughout the day or certain days in a week or month.

### Iperf

NetBeez has implemented an automated way to run Iperf throughput tests without needing an operator to do so. For those unfamiliar with iperf<sup>1</sup>, it's an open-source utility that executes TCP, UDP, and multicast throughput tests between one or more end-points. An Iperf test returns the one-way throughput between the client(s) and the server. In the case of UDP tests, iperf also reports packet loss and jitter. Iperf can be also configured to run multiple parallel tests, stress testing the network.

---

<sup>1</sup> The project's homepage is available at <https://iperf.fr/>.

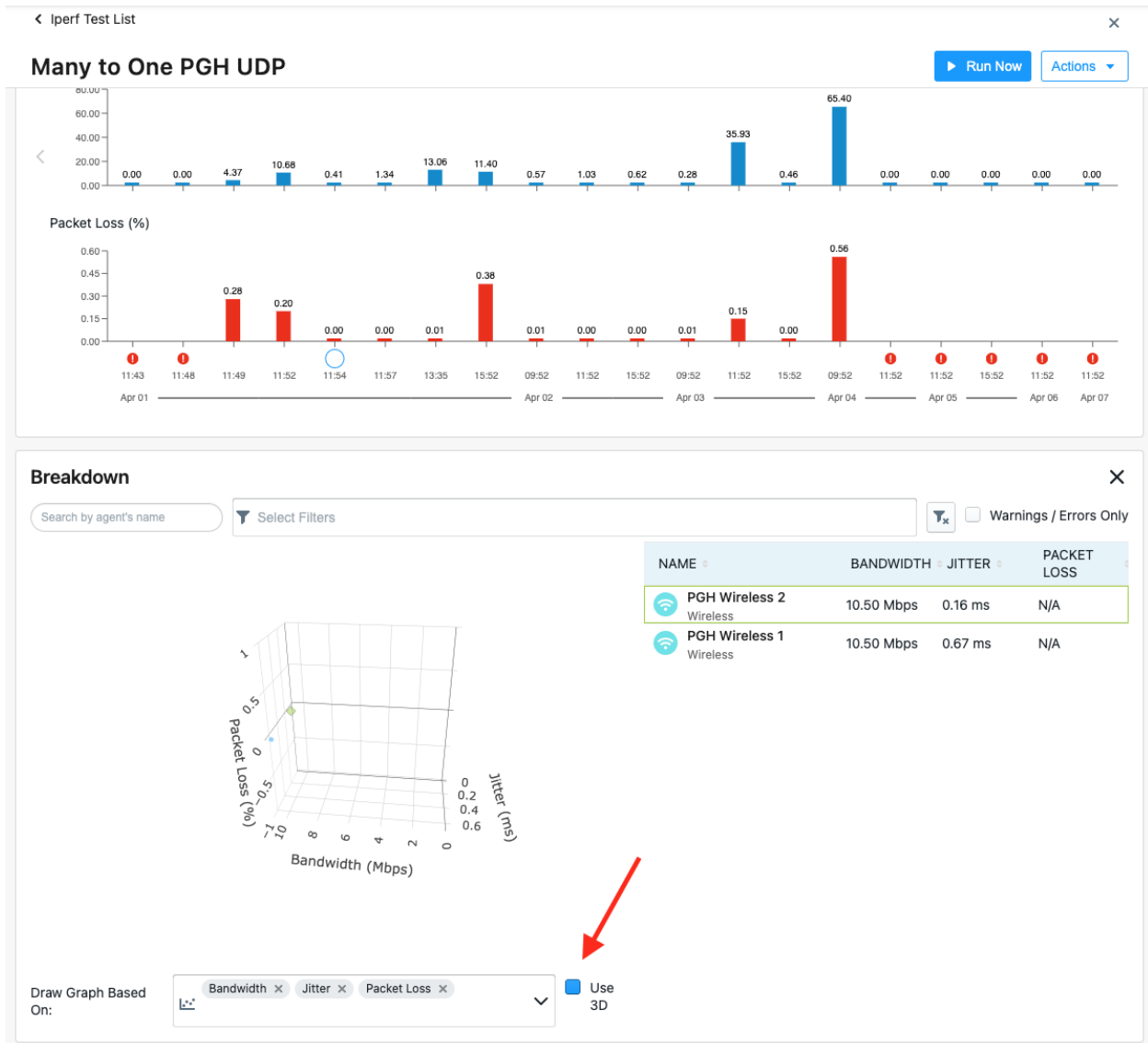


NetBeez supports two patterns: from one agent to another one, or from one or more agents to an Iperf server, which is not managed by the NetBeez dashboard itself. This case is useful to stress test a hub-and-spoke topology.

Another important option is the reverse flag. With this option, the throughput test can originate from the server to the client. This configuration is particularly useful when the client is behind a NAT device and there are no other ways to verify throughput.

### Enhanced Scheduled Test Result Visualization For Iperf Tests

An enhanced user interface has been added to multiple agents to server Iperf test results. Agents can be filtered by name, agent groups, agent type, ISP, ASN, bandwidth, jitter (UDP Only), and packet loss (UDP Only). Clicking on an agent in the table will display the agent's historical data in the next section.



In the section 'Draw Graph Based On': if bandwidth, jitter, and packet loss are selected, there will be a checkbox with the option to transform the graph into 3d. Viewing 3d Iperf test results is only available for UDP Iperf tests.

You can review setting Iperf tests on [this documentation page](#).

## Network speed

Network speed tests are useful to measure and build a baseline of download and upload speed. This data can be used to enforce Service Level Agreements (SLAs) with Internet Service Providers (ISPs) or troubleshoot performance issues at remote sites.



NetBeez agents can run three different implementations of network speed tests:

- **Speedtest** - NetBeez is using speedtest-cli, the open-source implementation of the Ookla speed test service. The GitHub page of the speedtest-cli project is available at <https://github.com/sivel/speedtest-cli>. This test measures the download, upload speed, and latency to an Internet Ookla speed test server. The user can select one specific speed test server from a public list<sup>2</sup> or to let the algorithm pick the server with the lowest latency.
- **NDT** - NDT is an *open-source* speed test service promoted by the [M-Lab consortium](#). Like the Ookla speed test service, NDT reports download, upload speed, and latency. NDT is also available via the command-line interface and reports additional information on the link between the NDT client, the NetBeez agent, and the NDT server, such as:
  - slowest link speed traversed by the packets,
  - whether there are network firewalls,
  - whether the client or the server is behind NAT.
 If you want to learn more about this command, read the [blog post on NDT](#) that we wrote.
- **Fast.com** - This is a download speed only test to the Netflix CDN network to verify the performance of Netflix streaming. This feature is particularly useful to Internet Service Providers interested in measuring the home subscriber end-user experience.

<sup>2</sup> A list of available ookla servers is available here <https://www.speedtest.net/speedtest-servers.php>.

## Enhanced Scheduled Test Result Visualization For Network Speed Tests

Network Speed test results optimizes the user interface when viewing scheduled test results for hundreds or thousands of agents making it easy to filter and search for specific agent results. All network speed test results utilize this enhanced feature.



## VoIP

VoIP tests are used to assess the quality of the network to deliver real-time, voice calls. VoIP calls require a low latency network, with zero or minimal packet loss and jitter (variation of latency).

The test simulates a VoIP call between two agents and returns the Mean Opinion Score ([MOS](#)), which is a key performance indicator of call quality. The test also reports the jitter, packet loss, and latency values, which are factors that affect the call quality and, consequently the MOS calculation. If you want to learn more about this, read our [blog post on MOS](#).



When creating a new VoIP test, the user can pick the codec that is used to run the test. Please refer to the table below for a list of codecs and their characteristics. All the VoIP tests generate a UDP stream between the two agents selected that conform to the codec's specifications. A common characteristic of these different codecs is that they're delivered as UDP packets marked with the EF (DSCP 46) IP Type of Service.

Codec Name	Payload Size	Voice Speech	Pkts per Sec.	Bit Rate
G.711	160 Bytes	20 ms	50	64 Kbps
G.729	20 Bytes	20 ms	50	8 Kbps
G.723.1-63	24 Bytes	30 ms	33.3	21.9 Kbps
G.723.1-53	20 Bytes	30 ms	33.3	20.8 Kbps
G.726	80 Bytes	20 ms	50	55.2 Kbps
G.728	60 Bytes	30 ms	33.3	31.5 Kbps
G.722	160 Bytes	20 ms	50	38.4 Kbps

## QoS

NetBeez has included DSCP marking of IP packets in the following tests: ping, traceroute, lperf, and VoIP (pre-set). DSCP marking can be used to test differences in test performance based on the QoS policy applied to network devices.

## WiFi Monitoring

NetBeez is a valuable tool to collect network performance data needed to troubleshoot WiFi networks. Both NetBeez network and remote worker agents support WiFi metrics. The following table summarizes the features that each agent type supports.

	Network Agent	Remote Worker Agent
<b>Wi-Fi configuration</b>	WiFi configuration (SSID profile) is done on the dashboard, then assigned to the agent	WiFi configuration is done on the laptop/desktop where the remote worker agent is running
<b>Wi-Fi Metrics</b>	Available	Available
<b>SSID Scanning</b>	Available	Available
<b>Wi-Fi Hopping</b>	Available	Not available
<b>Wi-Fi Connection Timing</b>	Available	Not available

### WiFi Monitoring on network agents

NetBeez WiFi network agents are small hardware units equipped with an external 802.11ac dual-mode card. These units have two interfaces: the Ethernet one, which is only used to establish the control channel with the server, and the WiFi one, which is used to run tests for monitoring. A WiFi agent is not capable of running any monitoring on the Ethernet interface. Once the agent is connected to the SSID network via the WiFi interface, the Ethernet plug can be removed, if desired<sup>3</sup>.

---

<sup>3</sup> Please remember that, a WiFi agent without an Ethernet connection may disconnect from the dashboard, if the WiFi network is unstable or completely off. It's highly recommended to, where possible, always keep the Ethernet connection active.





Front



Back

## WiFi networks

WiFi network agents must be configured with the SSID to be monitored. For this reason, one or more WiFi networks can be configured on the dashboard and then assigned to the agents. A WiFi network is defined by the following information:

- **Network name** - This optional field is displayed on the dashboard;
- **SSID** - This required field is the name of the SSID to needs to be monitored;
- **Description** - This is a description of the WiFi network that will be displayed on the dashboard;
- **Reconnection interval** - This optional field defines how often the agents will reconnect to the SSID, testing and timing the connection process;
- **Network verification test** - This optional test will be run right after an agent is connected to an SSID;
- **Band** - The profile can be set to a specific band (optional);
- **Security type** - WiFi agents support the following security settings:
  - Open,
  - WEP,
  - WPA pre-shared key,
  - WPA with EAP methodologies (also called Radius or 802.1x).

Configure

Select Agents

STEP 1

STEP 2

Security Type

Open

WEP64

WEP128

WEP256

WPA/WPA2-PSK

✓ WPA/WPA2-EAP

EAP Method:

EAP-TLS

Identity:

CA Certificate:

Browse

Client Certificate:

Browse

Private Key:

Browse

Private Key Password:

If you wish to learn more about this configuration, please check out [this documentation page](#).

## WiFi metrics

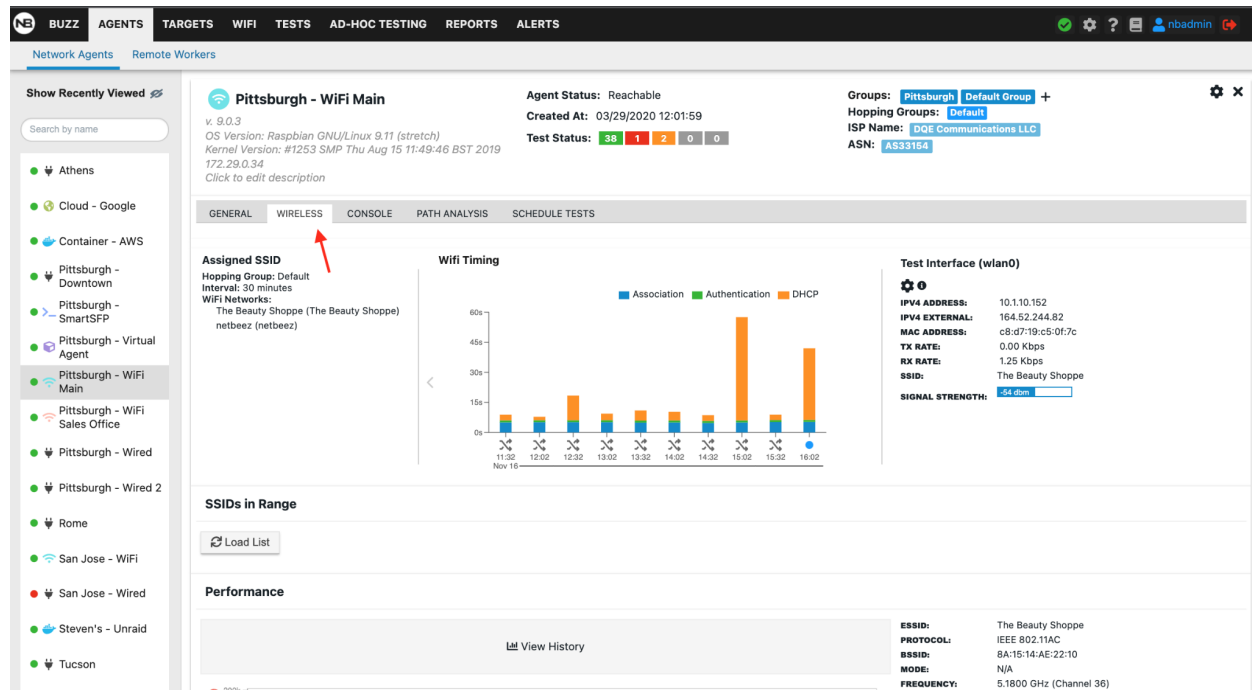
A WiFi network agent reports the following WiFi metrics about the monitored network:

Metric to be plotted	Data Type
TX/RX data on wlan interface	Bps (number)
SSID	string up to 32 chars
BSSID	HEX string similar to MAC address (FF:FF:FF:FF:FF:FF)
Signal Strength (RSSI)	dBm
Link Quality	%
Bitrate	Mbps
Channel	Numerical value based on band <ul style="list-style-type: none"> <li>- 5GHz - between 7 and 196</li> <li>- 2.4 GHz - between 1 and 14</li> </ul>

Channel Frequency

GHz ([https://en.wikipedia.org/wiki/List\\_of\\_WLAN\\_channels](https://en.wikipedia.org/wiki/List_of_WLAN_channels))

The WiFi metrics are available under an agent's details view, in a separate tab labeled "WIRELESS". Please refer to the following screenshot for more information.



## SSID scanning

A WiFi network agent can run scans of locally available wireless networks. Users can initiate a WiFi SSID scan within the wireless tab available. SSID scans are useful to find how many external SSID are using the same channels and perhaps causing more contention on certain frequencies.

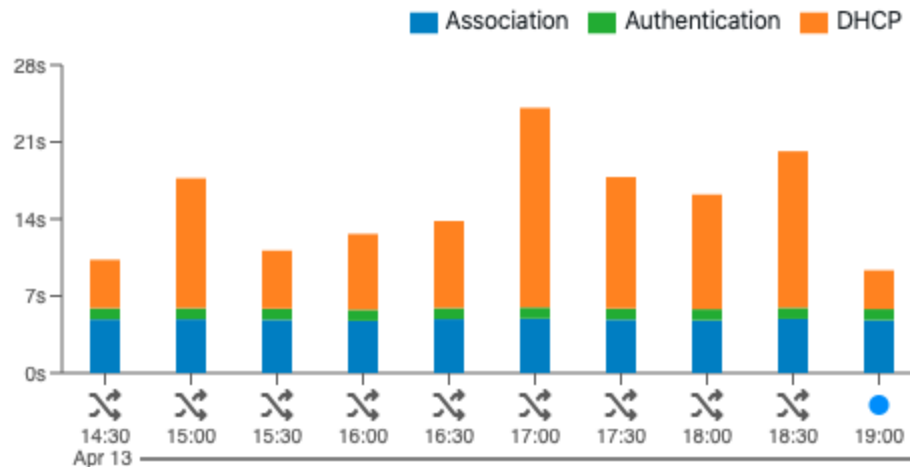
An SSID scan reports the following information:

- SSID
- BSSID
- Signal strength
- Frequency
- Channel

## WiFi connection timing

If a WiFi network is configured with a reconnection interval, the WiFi agents assigned to it will periodically test and time the WiFi connection process. The WiFi connection process is divided into three phases:

1. Association with a BSSID (access point) in proximity
2. Authentication with the selected SSID
3. DHCP to obtain an IP address

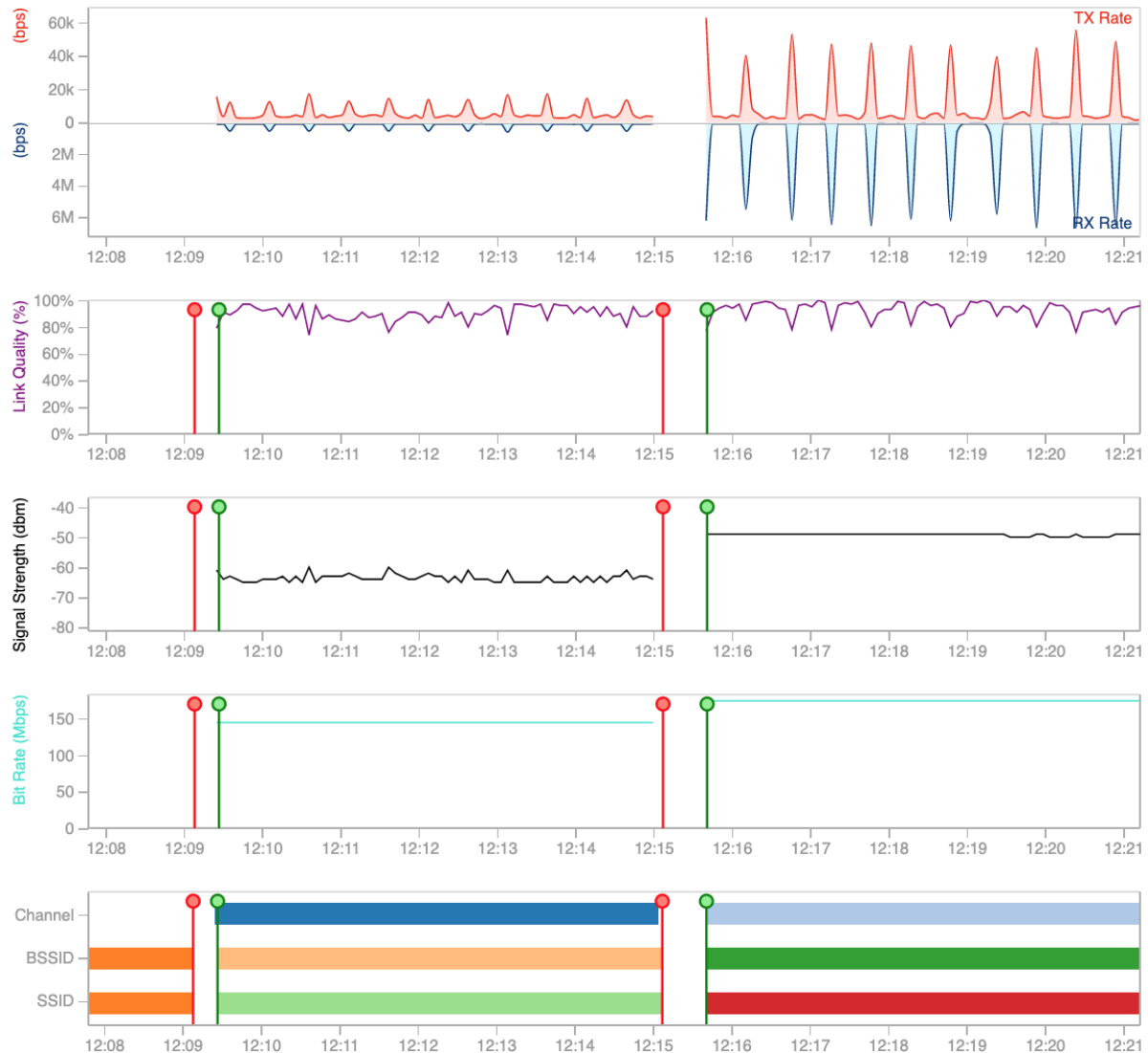


This feature is used to verify that a WiFi network can authenticate clients, and times the time that it takes for WiFi clients to connect and receive an IP address by the DHCP server.

## SSID hopping

SSID hopping enables a single NetBeez WiFi monitoring sensor to test up to four SSIDs by regularly connecting to each one of them. This feature enables enterprises to use one single sensor at each location to monitor all of their WiFi networks, without having to deploy one sensor for each SSID.

SSID hopping works by having WiFi sensors connect to the first SSID for a certain amount of time, called the "Hopping Interval", then disconnect and connect to the next SSID for the same amount of time, and so on. At the end of the cycle, the WiFi sensors repeat the loop. At each hop, the WiFi sensors will monitor the targets they've been assigned to and will run any scheduled tests configured for.



The WiFi connection timing feature in this case is provided by the hopping mechanism itself. At each hop, the WiFi sensors test how long it takes to connect to the next SSID. If a WiFi sensor can't authenticate, or get a DHCP address, within the "WiFi disconnection threshold", it will trigger an alert. By default, the WiFi disconnection threshold is set to 60 seconds. This value can be updated in the [Anomaly detection](#) (under the "Device alerts") section of the NetBeez dashboard.

An agent that is hopping through two or more SSIDs will display as many copies of the same test as many SSID it's monitoring. This is done to identify performance or connectivity issues on a per-SSID basis. For example, in the screenshot below the SSID "lab-test" is reporting performance issues on ping tests that the SSID "lab-test-guest" is not.

lab-test (WPA/WPA2-PSK)				
DESTINATION	PING	DNS	HTTP	TRA...
End-User Experience (Google ...	Orange	Green	Green	Grey
End-User Experience (YouTub...	Orange	Green	Green	Grey

lab-test-guest (WPA/WPA2-PSK)				
DESTINATION	PING	DNS	HTTP	TRA...
End-User Experience (Google ...	Green	Green	Green	Grey
End-User Experience (YouTub...	Green	Green	Green	Grey

## Band hopping

To enable band hopping, create two separate WiFi networks with the same SSID. In the first WiFi network configuration, force the agents to connect to the 2.4 GHz. In the second one, force the agents to connect to the 5.0GHz band. Lastly, add the two SSID into a hopping group.

### Add New WiFi Network

Configure

Select Agents

STEP 1

STEP 2

#### Configure WiFi Network

Network Name: Corporate 2.4 GHz  
SSID: corporate  
Description:  
☐ Periodically reconnect  
☒ Verify Network Connectivity  
Agents will verify network connectivity after a successful association to an SSID.  
Test type: HTTP  
Address: URL

Security Type: Open  
Band:

✓ Auto

Force 2.4 GHz

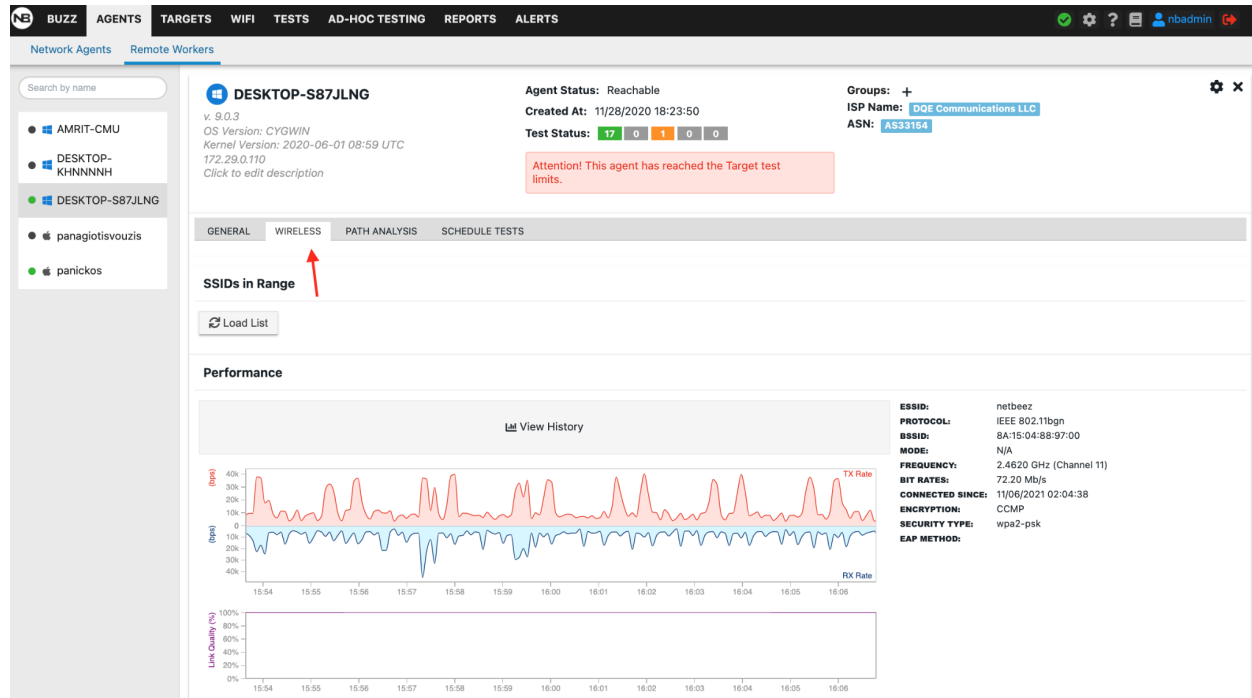
Force 5.0 GHz

Cancel

Save and Continue

## WiFi monitoring on remote worker agents

NetBeez remote worker agents for Windows and Mac support the reporting of WiFi metrics in real-time to the dashboard. Similar to network agents, remote workers' WiFi metrics are displayed in a separate tab within the details view.



Please keep in mind that the Windows and macOS operating systems don't report the same data. The table below lists all the metrics available based on the operating system where the remote worker agent is running on.

Metric to be plotted	Data Type	Windows	macOS
TX/RX	Bps (number)	yes	yes
SSID	string up to 32 chars	yes	yes
BSSID	HEX string similar to MAC address (FF:FF:FF:FF:FF:FF)	yes	yes
Signal Strength (RSSI)	dBm	yes	yes
Noise	dBm	no	yes
Link Quality	%	yes	no
Bitrate	Mbps	yes	yes
Channel	Numerical value based on band - 5GHz - between 7 and 196 - 2.4 GHz - between 1 and 14	yes	yes
Channel	GHz ( <a href="https://en.wikipedia.org/wiki/List_of_WLAN_channels">https://en.wikipedia.org/wiki/List_of_WLAN_channels</a> )	yes	yes

Frequency			
<a href="#">MCS</a>	Value between 0 and 9	no	yes


In both cases, the Wi-Fi metrics collected by the NetBeez agent are sent in real-time to the server. The data is then displayed on the user dashboard under an agent's details view. The data is also stored in the database for historical review and is available based on the data retention period defined by the user.

NetBeez remote worker agents also report connection and disconnection events to the dashboard. This information is valuable in many cases as it tells the network support team when the user is connected or disconnected to the wireless network.

Via the NetBeez dashboard the user can also perform an SSID scan to verify if any wireless networks in proximity are using the same channel and causing performance issues to the remote user. Here's an example of an SSID scan.


## Wired Tests on WiFi Agents

WiFi sensors have the ability to run real-time tests on the ethernet interface. Wired tests on WiFi are enabled on a per-target basis. If enabled on a target with WiFi sensors assigned to it, those sensors will run that target's tests on the WiFi as well as the wired interface simultaneously. This feature is only available on the Network WiFi agents.


**Tests on the wired interface**

—

DESTINATION	PING	DNS	HTTP	TRA...
Google (www.google.com:44...	<div></div>	<div></div>	<div></div>	<div></div>


**netbeez (netbeez)**

—

DESTINATION	PING	DNS	HTTP	TRA...
Google (www.google.com (w...	<div></div>	<div></div>	<div></div>	<div></div>

## Packet Capture

Packet capture is available for WiFi sensors in the ad-hoc tab of the NetBeez dashboard. During this operation, the NetBeez sensor pauses real-time and scheduled tests for the duration of the packet capture process. At the end of the packet capture, you can download the captured frames in a pcap file for further analysis (using tools like Wireshark), while the sensor resumes its regular network monitoring operations.



The screenshot displays the NetBeez AD-HOC TESTING interface. The top navigation bar includes tabs for BUZZ, AGENTS, TARGETS, WIFI, TESTS, AD-HOC TESTING (active), REPORTS, and ALERTS. The user is logged in as 'admin'.

**Run Ad-Hoc Tests**

Buttons: Ping, DNS, HTTP, Traceroute, Iperf, Network Speed, VoIP, Packet Capture.

**Run Ad-Hoc Packet Capture**

Source: San Jose

Test duration (s): 20

Radio Band: ☒ 2.4 GHz ☐ 5 GHz

Channel Width: ☒ 20 MHz ☐ 40 MHz

Channel Selection: 3 Selected

Channel hopping interval (ms): 200

Optional cli parameters (tcpdump): [Instructions and examples here.](#)

Buttons: Test running, Clear, Run

**PACKET CAPTURE: San Jose | TIME: 5/26/2021 12:00:26 AM**

Duration: 20s | Channels: 1,6,11 | Channel width: 20 MHz | Channel Hopping Interval: 200ms

Test running

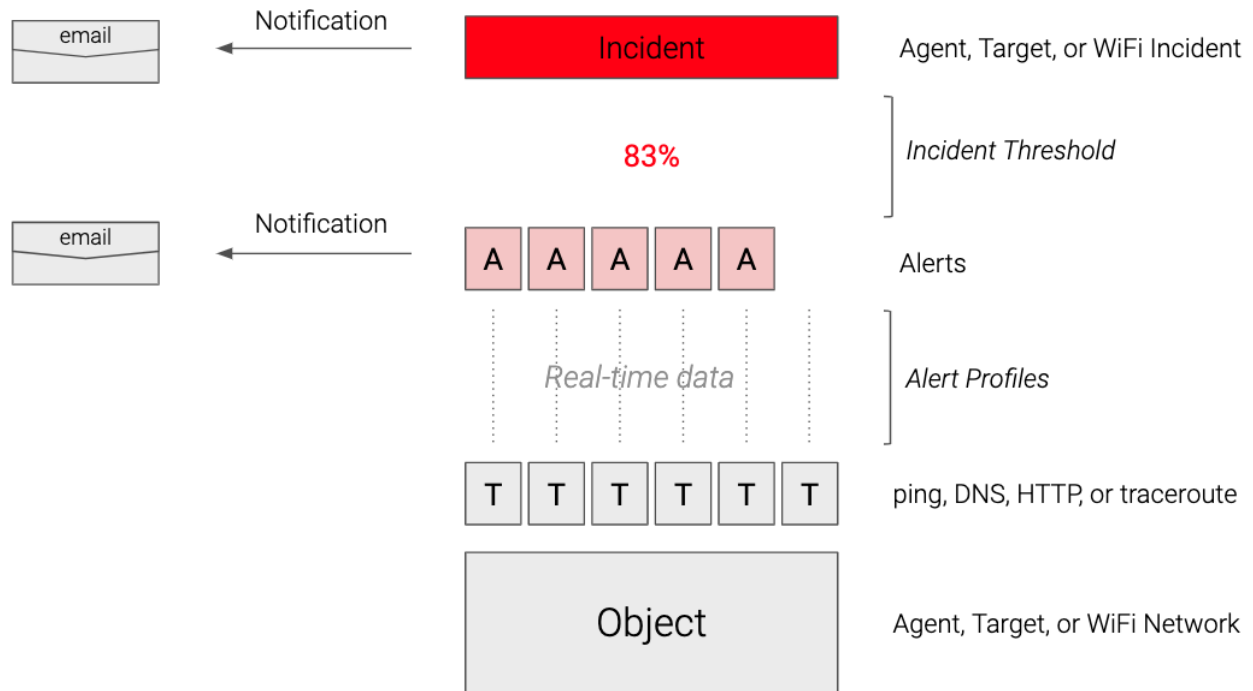
INFO: captured file size: 0.061440 MB (base64: 0.081920 MB (81920 bytes))  
 INFO: launched tcpdump command: 'timeout -s 9 -k 5 40 tcpdump -i wlan0 -W 1 -G 20 -l -w - | base64 -w 8192 > /var/log/netbeez/captu  
 INFO: Capturing in channel 1 (2412 MHz), width: 20 MHz, center1: 2412 MHz  
 INFO: Capturing in channel 6 (2437 MHz), width: 20 MHz, center1: 2437 MHz  
 INFO: Capturing in channel 11 (2462 MHz), width: 20 MHz, center1: 2462 MHz

## Anomaly detection

Anomaly detection is one of the primary functionalities of NetBeez. Fast and proactive detection is possible thanks to the real-time testing performed by the agents in conjunction with the alert detection process running on the BeezKeeper server. NetBeez users should familiarize themselves with the key concepts that are covered in this section.

There are three main components of the NetBeez anomaly detection system:

- **Alerts** - Alerts are triggered by real-time tests based on certain conditions, as defined in alert detectors; alert detectors are assigned to targets.
- **Incidents** - Incidents are triggered by an agent, target, or WiFi network when a certain percentage of tests trigger alerts (incident threshold); incidents are a good way to reduce the noise from multiple alerts related to the same resource.
- **Notifications** - Notifications are delivered via email, Slack, SNMP traps, and other methods when an alert is triggered and/or an incident is raised.



If the above chart is not clear, keep reading. In the next paragraphs, we'll review in detail each one of these components.

## Alert profiles

Alert profiles are assigned to targets to detect problems such as loss of connectivity or performance degradation to a remote service or application. Alert profiles are test-specific, that is, are related to a specific type of tests (ping, DNS, HTTP, traceroute, ...). NetBeez offers default alert profiles, which can be edited or deleted. The user can create new alert profiles, and attach them to new or existing targets. Any change to an alert profile will be immediately pushed to all the targets where that profile has been enabled.

## Types of alert profiles

There are five types of alert profiles:

1. **Up-Down** - An up-down alert is triggered by a real-time test when it fails for a given number of consecutive tries. By default, the up-down multiplier is set to five. This value can be adjusted by the user at any time.

Name:  Alert Type:  Default: ☒

### Triggering Conditions

Consecutive Tests Failed:

Test Type:

Up-down alerts are useful to detect loss of reachability to a remote host, network, or application.

2. **Performance Baseline** - A baseline alert is triggered by a real-time test when it detects a performance degradation issue. Performance degradation is detected by comparing the short-term moving average to its long-term, which is considered the performance baseline. In fact, for each test the server calculates the following moving averages:
  - a. Short-term: 1 minute, 15 minutes, 1 hour, 4 hours.
  - b. Long-term: 1 day, 1 week, 1 month.

If the short-term average of a test is a certain number of times higher than its long-term average, an alert is triggered.

Name:  Alert Type:  Default: ☐

### Triggering Conditions

Send alert if:

over a period of  is greater than  times the  average.

This type of alert profile is suited when a target is applied to many agents that have different performance results against the same application, due to their geographical location or other factors.

3. **Performance Watermark** - Watermark alerts are triggered when a real-time test doesn't meet specific performance requirements. These alerts are used to enforce service level agreements with network services and applications. Watermark alerts are configured by

comparing a short-term average against a user-defined threshold (eg. packet loss is higher than 5% or DNS resolution time is higher than 100 ms).

Name:  Alert Type:  Default: ☐

## Triggering Conditions

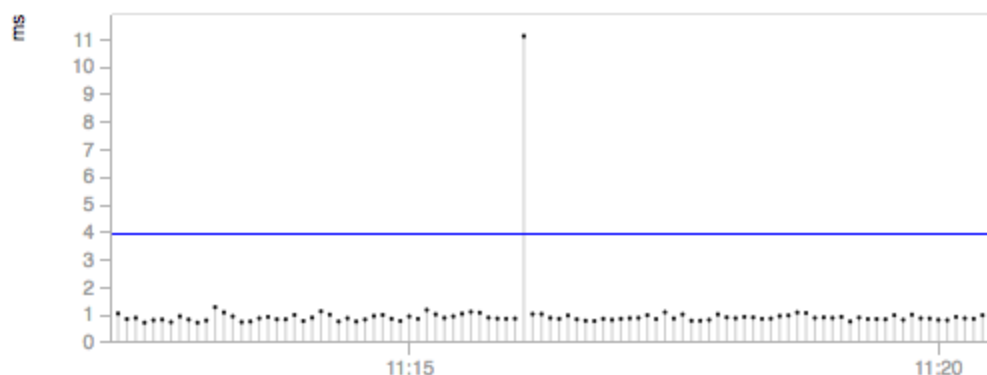
Send alert if:

over a period of  is greater than  s.

4. **Performance Baseline AND Watermark** - This alert profile merges both performance baseline and watermark rules. If both detection methods are satisfied, then an alert will be triggered.
5. **Down-Up** - Down-up alerts are triggered when a test succeeds. The mechanism is the reciprocal of the up-down alert. Down-up alerts are used to enforce security policies, such as verifying content filtering (e.g. users can't access certain websites) or firewall rules (e.g. an isolated network can't access the Internet).

### Percentile-based mean

When calculating an average for a given time period, all test results are accounted for. However, especially in stable time series, one or a few outliers could dramatically skew the mathematical average, like in the below example. When this happens, performance alerts may be triggered, causing false positives.



NetBeez supports the use of a percentile-based mean (95th percentile), which filters out outliers that fall two times outside the standard deviation interval (from the mathematical average). The benefit of this function is that should a single data point skew the mathematical average, the percentile-based mean won't be affected, thus reducing the number of false positives (alerts

noise). To enable this feature, the user can enable the use of a percentile-based mean in the [Anomaly Detection settings](#) section.

Enable calculation of percentile-based mean: ☒

Warning: Enabling percentile-based mean calculation will increase the database storage requirement by up to 50%.

If you want to learn more about alert profiles and percentile-based mean, please refer to the [online documentation page](#).

## Device alerts

Device alerts are triggered when an agent is either unreachable from the dashboard, or a WiFi agent isn't able to connect to a WiFi network for a given period of time.

**Agent Device Alerts**  
Set the threshold for when the system will generate an alert for device level disconnection events.

Agent unreachable threshold:	<input type="text" value="45"/> <small>In seconds</small>	Wifi disconnection threshold:	<input type="text" value="60"/> <small>In seconds</small>
------------------------------	--	-------------------------------	--

One important thing to remember is that, when a device is not connected to the dashboard anymore, all tests are placed in unknown status (marked with “?”), incidents are cleared, and so are alerts.

## Incidents

Incidents are periods of degraded or otherwise abnormal performance of an agent, target, or WiFi network. This functionality is designed to help users identify problems and performance variations with a network location (agent), service or application (target), and WiFi network. Another benefit of incidents is that they reduce the number of notifications that a user has to receive.

An incident is triggered when a certain percentage of tests within one agent, target, or WiFi network trigger an alert. Such thresholds are defined by the user in the NetBeez “Incidents Configuration” settings, under the “Anomaly Detection” section.

### Incidents Configuration

*Configure incidents to better detect patterns in performance and outage issues*

#### Agent Incidents

Set alert percentage thresholds for Agent incidents:

**Ping:**

Incident threshold: 90% of Ping tests with alert status

**DNS:**

Incident threshold: 90% of DNS tests with alert status

**HTTP:**

Incident threshold: 90% of HTTP tests with alert status

**Traceroute:**

Incident threshold: 90% of Traceroute tests with alert status

Incidents can be acknowledged, and users can post comments to include more information about the undergoing performance issue, or explain the reason why a specific incident was acknowledged or de-acknowledged.

If you want to read more about Incidents, please consult [this documentation page](#).

## Notifications

Even when not in front of the dashboard, users can receive notifications on alerts and incidents. NetBeez supports different delivery methods for notifying users about new alerts and incidents raised and closed, such as SNMP traps, Syslog messages, and emails. For each delivery method, the user can pick what to receive: only alerts, only incidents, or both of them. In general, it's good practice to enable notifications for agent incidents and alerts, target incidents, and wifi incidents.

### Email Notification Settings

Configure global notifications and default recipients for Agents, Targets, and WiFi Networks:

SMTP Notifications: ☒

#### Agents

Incidents: ☒

Device Alerts: ☒

Emails:

#### Targets

Incidents: ☒

Alerts: ☐

Emails:

#### WiFi Profiles

Incidents: ☒

Alerts: ☐

Emails:

NetBeez also provides integrations with many third-party tools, such as Splunk, PagerDuty, and Slack. You can learn more about these integrations on the [online documentation](#).

## Data retention

NetBeez supports user-defined data retention settings. The user can set for how long the central server should retain performance data collected by the agents. In practice, the data retention dictates how far back historical test data and reports data can go. Some of the variables to be set are:

- **Raw results** - This is the raw data from test results, where each data point is the result of a test. Raw test data is displayed in real-time and historical graphs.
- **1-min average** - This is the average of test results collected in one minute. The 1-min average is used to generate performance alerts and reports.
- **1-hour average** - This is the average of test results collected in one hour. The 1-hour average is used to generate performance alerts and reports.
- **24-hour average** - This is the average of test results collected in twenty-four hours. The 24-hour average is used to generate performance alerts and reports.

The resulting disk space required is dependent on the time period selected, the number of tests, and their interval. This configuration setting can be easily applied from the NetBeez Settings in a few clicks.

**Data Retention**  
 Configure how long each metric should be retained for.

Data Type	Retention Period	Space Utilized
Raw results	Past month	Results: 2.41 MB Traceroute Results: 64.00 KB
1-Min Average	Past 3 months	Nb Agent Statistics: 28.09 MB
1-Hour Average	Past 6 months	Nb Test Statistics: 59.62 MB
24-Hour Average	Past year	Result Values: 144.00 KB Scheduled Nb Test Results: 128.00 KB
Scheduled Test Results	Past 3 months	
Alerts	Past 6 months	112.00 KB
Incidents	Past 6 months	64.00 KB
Agent Logs	Past week	80.00 KB
WiFi Metrics	Past 3 months	2.78 MB
Endpoint Performance Metrics	Past 2 weeks	8.05 MB

Save Data Retention Settings

Please refer to the [online documentation page](#) to learn more about this.

## Users

The NetBeez dashboard is multi-user and multi-role. There are three access levels in NetBeez:

- **Administrators** - This user role can do anything: create and edit targets and scheduled tests, run ad-hoc tests, create reports, and configure any aspect of the solution, via the NetBeez Settings panel. Generally, this account profile is assigned to the application owners and senior network engineers of the organization



- **Read-Write** - This access role can create and edit targets and scheduled tests, run ad-hoc tests, and create reports. Generally, this account profile is assigned to intermediate network engineers and network managers.
- **Read-Only** - Accounts with read-only privileges will be able to review targets, scheduled tests, but not create and edit targets or scheduled tests. This profile should be assigned to support operators, and junior network engineers, and analysts.

## User authentication

There are two types of user authentication methods for users to log on to the NetBeez dashboard: local and enterprise authentication.

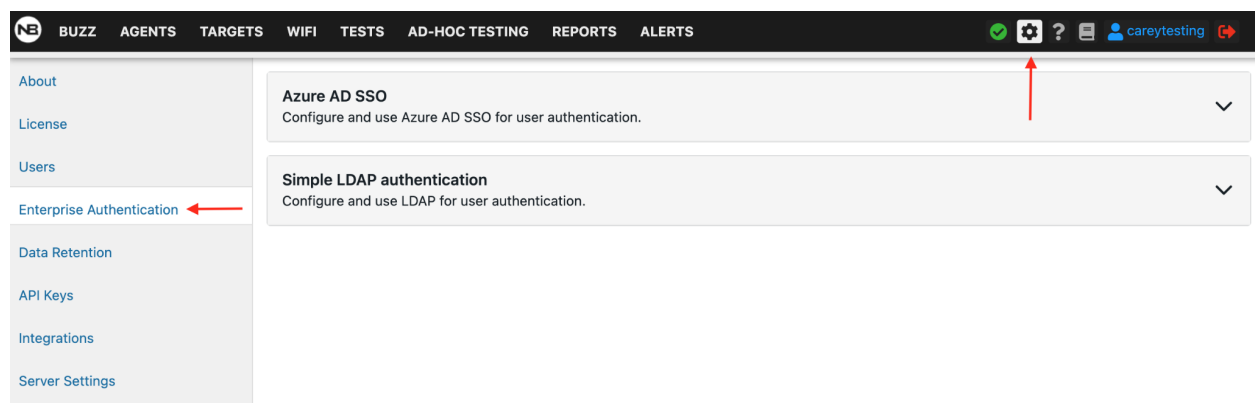
### Local Authentication

The default authentication method of dashboard users is local with complex passwords (8 characters, 1 upper case, 1 lower case, and 1 number).

### Enterprise Authentication

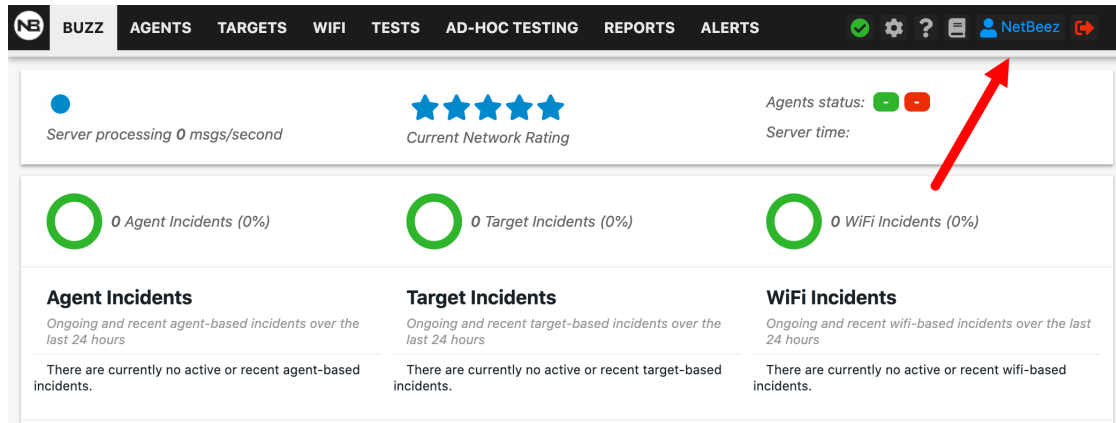
There are two enterprise authentication methods available: LDAP and Azure AD.

- **LDAP** - Lightweight Directory Access Protocol is an open and cross-platform protocol used for directory services authentication. LDAP provides the communication language that applications use to communicate with other directory services servers. Read more about setting up [LDAP Authentication](#) in our documentation.
- **Azure AD** - Azure Active Directory is Microsoft's cloud-based identity and access management service, which helps users sign in and access resources. Read more about setting up [Azure AD](#) in our documentation.



## User Profile

Dashboard users can manage their profile by clicking on the user icon located at the top right corner of the screen.



Within the user profile panel, a user can change:

- Profile Information - Update their first name, last name, email address, and username
- Password - Change their current password
- Security Question - Update their security question and answer

The screenshot shows the NetBeez user profile management page. At the top is a navigation bar with links: BUZZ, AGENTS, TARGETS, WIFI, TESTS, AD-HOC TESTING, REPORTS, and ALERTS. Below the navigation bar is a header section with a circular profile icon containing 'NN', the name 'Netbeez NetBeez', and the role 'ADMINISTRATOR'. The main content area contains three sections: 'Update Profile', 'Change Password', and 'Update Security Question'. Each section has a 'Save' button. The 'Update Profile' section has fields for First Name, Last Name, Username, and Email. The 'Change Password' section has fields for Current Password, New Password, and New Password Confirmation, with a list of password requirements. The 'Update Security Question' section has fields for Current Password, Security Question, and Answer.

NetBeez NetBeez  
ADMINISTRATOR

**Update Profile** [Save Changes](#)

First Name:  Username:

Last Name:  Email:

**Change Password** [Save Password](#)

Current Password:

New Password:

- Contains 1 uppercase letter
- Contains 1 lowercase letter
- Contains 1 number
- Contains at least 8 characters

New Password Confirmation:

**Update Security Question** [Save Changes](#)

Current Password:

Security Question:

Answer:

Agents status: 1 6 Server time: 07/17/2020 14:33:43 EDT

The user profile management page is not available if the dashboard authentication is handled by LDAP.

## Reports and API

NetBeez users can generate reports to review historical network and application performance data based on agents deployed, targets and scheduled tests configured. Reports can be generated on the dashboard or can be scheduled to be sent via email.

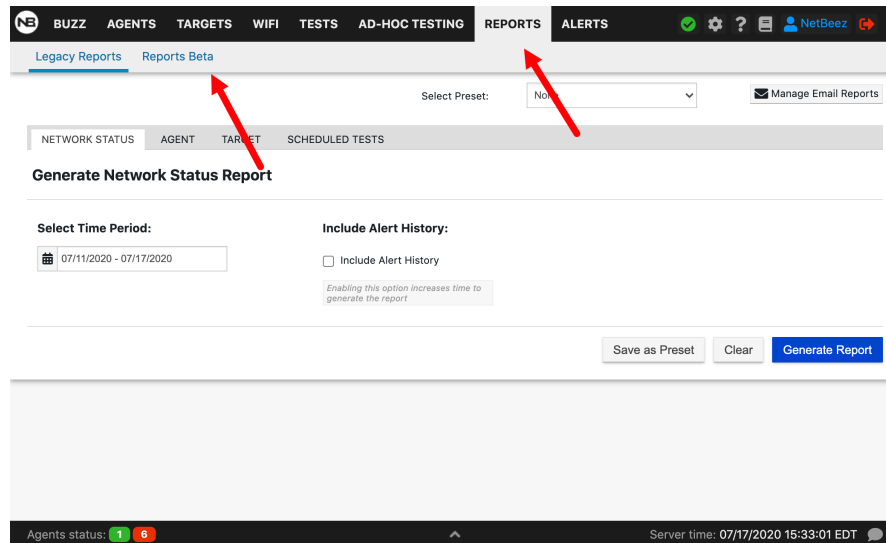
### Legacy Reports

In the Legacy Reports section under the Reports tab is located the legacy reporting feature. Starting from version 5.0, the NetBeez dashboard is offering an improved and more efficient reporting experience. Until the Reports Beta is completed, the original reporting feature can be utilized under the Legacy Reports tab. Four types of reports can be generated under Legacy Reports:

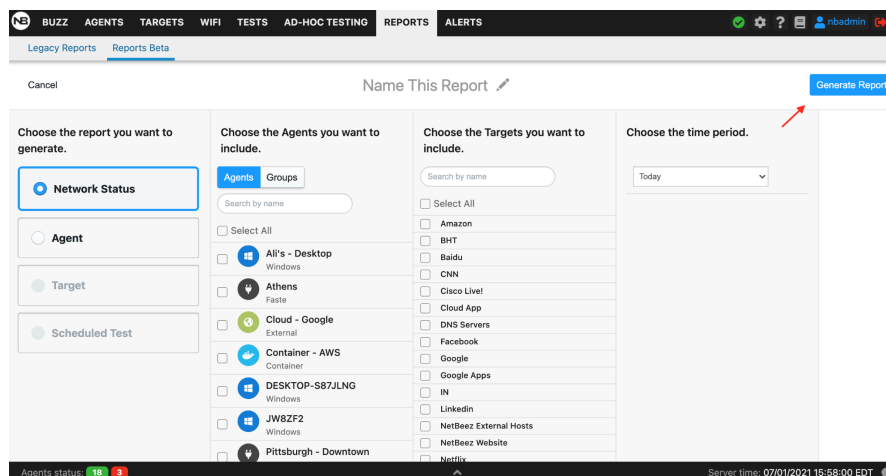
- **Network summary** - This report provides a high-level overview of agents and targets. For each agent, the user can review availability, download, and upload speed if that agent had a speed test configured, and the number of alerts. For each target, the user can review the average HTTP GET time and number of alerts.
- **Agents** - The user can select one or more agents to get a report on alerts occurrence per target monitored, test performance, and HTTP targets performance.
- **Targets** - The user can select one or more targets to get a report on alerts occurrence per agent included, test performance, and HTTP performance per agent.
- **Scheduled tests** - This report plots in a line graph format selected Iperf, speed test, and VoIP scheduled tests.

When generating a report on the dashboard, it's possible to extract such a report in PDF format or save that report in a preset, which is the definition of the report type along with the input parameters selected (e.g. report time period, elements to be included in the report, etc.). As we'll see in the next paragraph, presets must be created to schedule email reports.

### Reports Beta



The new reports experience is in beta so the functionality currently is limited. For this reason, the fully functional reports are available under "Legacy Reports" subtab. The redesign makes generating reports easier and more intuitive. As of version 8.0, network status reports and agent reports can be generated. Future releases will include targets and scheduled tests reports. Please be aware that in the current version there is no email functionality or PDF download option. The Legacy Reporting tab will remain for these needs until the new Reports Beta is completed.



## Email reports

NetBeez users can send PDF reports via email. The reports that are included in the email are defined via presets. Emails are sent to one or more email addresses based on a user-defined schedule. To learn more about email reports, please consult [this documentation page](#).

## API

NetBeez has a public API, which can be reviewed on the swagger page available with each instance at the URL `https://<DASHBOARD_FQDN>/swagger/index.html`. The current version of the API makes it possible for external applications to get the object's status, network, and application performance data collected by the agents.

## Public dashboard

The public dashboard is an open-source project developed in PHP and uses the NetBeez public API. The public dashboard can be installed on any web-server (libcurl required) to provide a service status dashboard based on the targets and agents configured. The benefits of a public dashboard are that the network engineers and managers can share the status of the network and applications monitoring with the end-users by NetBeez. Also, the public dashboard can be installed on an intranet or public website, without compromising the NetBeez server's security. If you want to learn more about this, check out the [public dashboard GitHub page](#).

## Troubleshooting with NetBeez

NetBeez can be helpful in troubleshooting network and application performance issues in large and complex Wide Area Networks (WAN). The real-time data reported by the agents is used to identify the scale (e.g. number of locations) and layer (e.g. network versus application) of performance issues.

### Using the Buzz Tab

The Buzz Tab was designed to provide the user with the most important information that NetBeez collects, such as:

- **Agent, target, and WiFi performance** - The user can review the agent and HTTP target performance distribution. In the agent performance chart, each agent is represented as a dot, and whose XY coordinates are determined by its number of performance alerts (y) and up-down alerts (x) triggered in the last 24 hours. *In the agent performance bar chart, each target is included in a bar, based on its 24-hour average HTTP response time.*
- **Open and recent incidents** - This section of the Buzz Tab reports all agents and target incidents logged over the past 24 hours. This is a good way to quickly find if there are network or application issues occurring. By clicking on a specific incident, the user can check out what tests and alerts caused the incident to be generated by the system.

### Interactive console

An interactive console is built into the NetBeez dashboard and enables users to type commands directly to the reachable NetBeez agents. The console can be used to review log files to troubleshoot agent-related issues, run command line commands that are not yet integrated with the dashboard. Some of the commands that are oftentimes used by network engineers on the interactive console are: nmap, tcpdump, and arp. The interactive console is only available to administrators and is accessible from the agent details view.

### Ad-hoc tests

Ad-hoc tests can be used to troubleshoot network and application problems on the spot, without having to create a target. The user selects the test type, the source agent, and destination IP, FQDN, or agent where applicable. *The command will run for the amount count indicated*, and then exit. Results are logged in real-time on the NetBeez dashboard.

- [Ping](#)
- [DNS](#)
- [HTTP](#)
- [Traceroute](#)
- [Iperf](#)

- [Network Speed](#)
- [VoIP](#)
- [Packet Capture](#)



## NetBeez Configuration Checklist

Here's a high-level list of items that are necessary to have a working configuration:

<b>Agents</b>	<ul style="list-style-type: none"><li>• Connect all hardware agents (FastE, GigE, WiFi) to a network switch via the Ethernet interface; make sure that they report to the dashboard; if any agents have problems reporting to the dashboard, here's the <a href="#">troubleshooting procedure</a>.</li><li>• Configure the WiFi agents with the appropriate SSID profile to connect to your wireless network(s).</li><li>• Rename the agents and create agent groups if needed.</li></ul>
<b>Targets</b>	<a href="#">Create targets</a> based on the prospect's monitoring goals; this is a very important part, so make sure to clearly define the applications and network services that they want to monitor.
<b>Scheduled Tests</b>	Setup scheduled tests for running throughput tests to other network locations (Iperf), or to the Internet (speed test); VoIP tests can be used to verify the performance and quality of VoIP calls.
<b>Users</b>	Invite your team to access the NetBeez dashboard: send them an invitation via email so they can create their own account; review the documentation page on <a href="#">user management</a> to learn more about the different roles and privileges available to users.
<b>Email Reports</b>	Setup daily, weekly, or monthly <a href="#">reports</a> to be sent via email as PDF attachments.
<b>Alerts and Incidents configuration</b>	Define what type of performance alerts to enable; review the <a href="#">alerts</a> documentation page to learn the difference between up-down, baseline, and watermark alerts.
<b>Notifications</b>	Receive notifications on alerts and incidents via <a href="#">SMTP</a> , <a href="#">SNMP</a> , or <a href="#">Syslog</a> ; in the alternative, review the list of available <a href="#">integrations</a> , such as Splunk, PagerDuty, and Slack.



## Resources

If you want to learn more about NetBeez, there are several online resources available:

- NetBeez online documentation: <https://netbeez.zendesk.com/hc/en-us>
- NetBeez blog: <https://netbeez.net/blog>
- NetBeez YouTube channel:  
<https://www.youtube.com/channel/UC89nekW3nqHyIKK6OaTH6uA>