TALLER DE LATENCIA I:

RIPE Atlas: ¿Dónde está la nube?

RIPE Atlas: Can we find where the Cloud is?

Authors: Antonio Bazco-Nogueras, Rita Ingabire, Vincenzo Mancuso
IMDEA Networks Institute, Madrid, Spain

Updated on: November 16, 2023

INSTRUCTOR GUIDE

This document serves as guide for the instructor of the course. We recommend the instructor to carefully read this document before coaching the session with students.

If you have any question or doubt, please contact with the authors by email:

antonio.bazco@imdea.org – rita.ingabire@imdea.org – vincenzo.mancuso@imdea.org

This guide has been created by the Opportunistic Architectures Lab of IMDEA Networks Institute (Madrid), in the frame of the TelecoRenta Proyect from the UNICO 5G Program.

Welcome to the instructor's guide for the measurements lab RIPE Atlas: Can we find where the Cloud is?. This document is similar to the corresponding students' guide, with the addition of comments and suggestions for the instructor.

All the information can be found at the GitHub repository. This document is named guide for the instructor PDF file. You can also obtain the student guide and other material.

Through this document,[1] the students will learn how we can measure the latency across the whole globe. For that, we explain how to run and retrieve measurements with the tool RIPE Atlas, which is a platform from RIPE that allows you to define measurements from any of the thousands of probes available in RIPE Atlas towards any destination (url or IP address).

This document is intended for people that may want to know:

- How does it work the global internet network?
- What is the path that my information follows until it comes back?
- Does it matter where the servers are located in the cloud?
- Can we know in advance the performance of the network?

This guide is structured in the following sections:

1) Software Requirements.
2) What RIPE Atlas is and how it works.
3) How you can run your own measurements.
4) How you can retrieve your measurement data.

[1] Disclaimer: Some of the content of this document has been retrieved from the web and documentation of RIPE Atlas, https://atlas.ripe.net/. The content is here summarized to facilitate the management of the whole process.

I Requirements and material

The first step is to ensure that everyone has access to all the required materials. The main software that you need is Python. We recommend to install the packages in this order: first, the base packages that will allow you to run the software on your computer; second, the Python package manager that enables you to install different python libraries for processing your data; finally, download the experiment files from the public repository. The full list of materials and links is presented next:

Base software:

1) PYTHON 3: The first requirement.
2) PIP: Python Package Manager.
3) JUPYTER LAB & NOTEBOOK: Notebook Interface for Python code.

Python packages:

1) RIPE Atlas Cousteau: A Python wrapper around the RIPE ATLAS API.

Course scripts:

1) Jupyter Notebook I: Code to create a measurement.
2) Jupyter Notebook II: Code to retrieve a measurement.

Finally, you can also be interested on the site of the main tool that we consider for measuring Internet latency: https://atlas.ripe.net. You can find also find all the code and supplementary materials in the following GitHub repository: Ripe-ATLAS guide.

II What is the RIPE Atlas platform and how does it work?

Let us start by introducing the platform that we will be using to run live experiments and gather data measurements.[2]

RIPE (Réseaux IP Européens): Regional Internet Registry for Europe, the Middle East and parts of Central Asia. As such, they allocate and register blocks of Internet number resources to Internet service providers (ISPs) and other organizations.

RIPE NCC (RIPE Network Coordination Center): Not-for-profit organization that works to support the RIPE (Réseaux IP Européens) community and the wider Internet community.

RIPE Atlas: The active Internet measurement network from the RIPE NCC. RIPE Atlas is the RIPE NCC's main Internet data collection system. It is a global network of devices, called probes and anchors, actively measuring Internet connectivity. Anyone can access the data via Internet traffic maps, streaming data visualisations, and an application programming interface (API). Users can perform customized measurements to gain valuable data about their own networks.

Probes form the backbone of the RIPE Atlas infrastructure. There are thousands of active probes in the RIPE Atlas network, and it is continually growing. Volunteers all over the world host these small hardware devices or tiny software packages that actively measure Internet connectivity through ping, traceroute, DNS, SSL/TLS, NTP and HTTP measurements. The data are collected and aggregated by RIPE NCC, which makes the data publicly available. Network operators, engineers, researchers and even home users have used these data for a wide range of purposes, from investigating network outages to DNS anycasting to testing IPv6 connectivity.

How does RIPE Atlas work?

In brief, RIPE NCC users can request and install (upon acceptance) one of the active RIPE probes in their network. That action allows them to earn credits, which can later be spent to run their own customized experiments. These experiments can use any of the thousands of public probes available through the RIPE Atlas platform.

[2] Data analytic methods shown later in the lab can also run on top of data gathered through different platforms. Indeed, we also provide json files ready to be used with the data analysis tools presented in the guide—see Section V.

RIPE Atlas is a collaborative tool: You must install a probe to earn credits to run your own customized measurements; that is, contributing to the network. Yet, all measurements are public and available even if you do not own any probe.

III How can you run your own measurements?

[Comment for instructor:] Note that this section is mostly descriptive for the student. In this section, we explain how to create a new RIPE Atlas account, how to install a probe (which is necessary to obtain credits that allow you to run experiments), and how to create those experiments/measurements.

However, the approval by RIPE NCC of the probe is not immediate, and it takes at least one day to get accepted and activated. This means that doing the actions here explained during the course would not allow students to actually run experiments. We propose to the instructor two options:

- The instructor can follow these steps, create an account and install a probe, with some time in advance before the course. In this way, the students can use such account and the credits earned in the elapsed time to create new experiments in the same session.
- The instructor splits the course in two days. In the first one, the students arrive until the probe installation and request. In the second, once RIPE NCC has activated the probe, they can continue. However this option has some problems: 1) the course length for each of the days may be too short and 2) RIPE NCC may block or reject some probes if all come from the same network.

In any case, the best approach is to proceed with the first case. Then, during the course, the instructor can follow the steps until the probe submission as if it was the first time doing it. Then, the instructor can mention to the students that "now you need to wait until RIPE NCC answers. We can continue with an account and probe that I already activated some days ago."

We are going to follow the steps that are required to define your own measurements. We can split the process in three different parts. Please, follow your instructor's guidance to follow this section, as it can not be completely performed during the duration of the lesson.

A Becoming a member of the RIPE Atlas community

The first step before getting access to the RIPE Atlas platform is to create a RIPE NCC Access account. This is a simple step that can be done by accessing the following link https://access.ripe.net/registration. You just need your name, e-mail and password.

B Installing a RIPE Atlas probe

Initially, RIPE Atlas probes were hardware probes: small hardware devices that actively measure the Internet. Yet, hardware probes are currently (November 16, 2023) out of stock due to a shortage of semi-conductors, and RIPE Atlas provides an equivalent software probe as that can be easily installed in any device. We thus only explain the installation of the software model.

While not a replacement for their hardware counterparts, software RIPE Atlas probes are software packages that work just like regular probes. As such, hosts will install the probes on their own bits of infrastructure — e.g. virtual machines, home routers, servers, and so on. Various platforms and operating systems have different levels of support and ease of use. You can find which systems (hardware or OS) are supported, and how, below.

Any future host of a RIPE Atlas probe is expected to understand what it means to operate a probe for the RIPE Atlas network; you can read more on the RIPE Atlas about page. Details of the conditions and responsibilities are available in the RIPE Atlas Terms and Conditions.

Installation Options:

The RIPE Atlas probe code can be found on GitHub. RIPE NCC has developed the software for Centos 7 and 8. However, the community has released implementations for other platforms and operating systems, each of which comes with its own level of support and ease of use. Current options for installation include: CentOS 7 and 8 (binary RPM package and source installation), Debian (9 and 10) and Raspbian (source installation), Turris Routers (official software package from Turris), and Docker (preliminary source installation).

Full details on getting your software probe up and running for all the approaches listed above are available on RIPE Atlas website. In the following, we focus on the installation on Docker, because it is the simplest and most flexible of the implementations.

Installing software probe with Docker:

There exist three different implementations for Docker containers, which can be accessed here. We consider in particular one that can be found on this GitHub repository. The installation is simple. First, we need to install Docker if it is not installed yet. Docker Engine is available on a variety of Linux distros, macOS, and Windows 10+. If Docker is already installed, you just need to use Docker run to start the container as:

Installation with Docker

```
        docker run --detach --restart=always \
          --log-driver json-file --log-opt max-size=10m \
          --cpus=1 --memory=64m --memory-reservation=64m \
          --cap-drop=ALL --cap-add=CHOWN --cap-add=SETUID --cap-add=SETGID --cap-add=DAC_OVERRIDE --cap-add=NET_RAW \
          -v /var/atlas-probe/etc:/var/atlas-probe/etc \
          -v /var/atlas-probe/status:/var/atlas-probe/status \
          -e RXTXRPT=yes \
          --name ripe-atlas --hostname ”$(hostname --fqdn)” \
          jamesits/ripe-atlas:latest
```

You could also use Docker Compose.

Next, you can fetch the generated public key from:

cat /var/atlas-probe/etc/probe_key.pub

and register the probe with this public key on RIPE Atlas website (you need a RIPE account to access the site as described in Section A). The registration will be manually processed and they will confirm your registration. Afterward, the new probe will be available in your account. Then, you will earn daily credits while your probe is online and available. Those credits allow you to run your own customized experiments, as detailed next. If you wish to learn more about RIPE Atlas you can also check [1].

C Running a measurement with RIPE Atlas

Next, we consider now that we have an account already created with a probe running since several days or weeks, so we have earned enough credits to perform some experiments. This account may be provided by your instructor, or you could proceed and create new measurements in the incoming days.

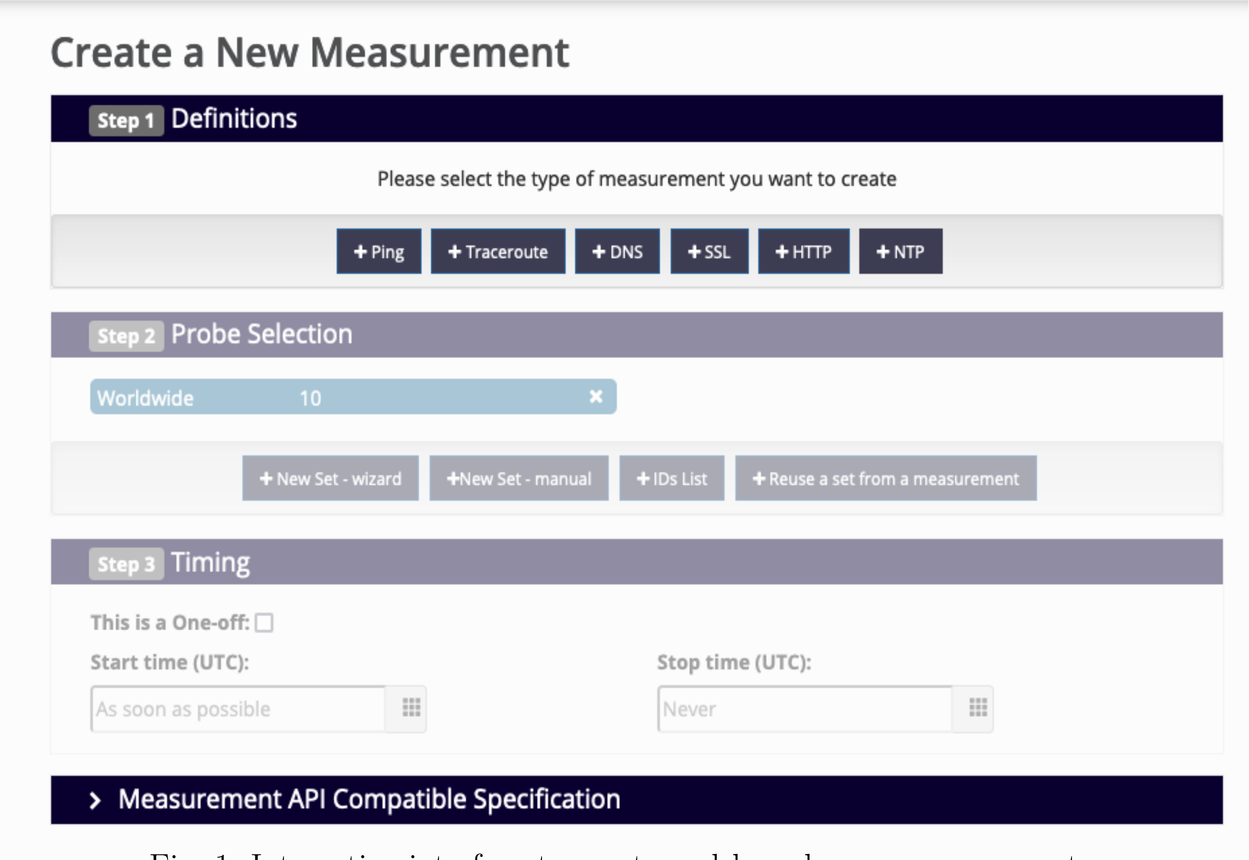
We present the straightforward process to start a measurement from any of the thousands available probes in RIPE Atlas. There are two main methods. The first one (simpler) consists on making use of the web interface that RIPE Atlas provides. It is a user-friendly interface requiring few steps. The second one is more advanced and can be skipped if you are not familiar with Python and APIs.

[Comment for instructor:] Depending on the level of the students, you can include the second method or just skip it.

Using the web Interface:

1) Log into the website: Go to the ripe atlas website and log in. Use the credentials created in Section A to be able to create measurements from the web interface.

2) Create the measurement: Go to the “create measurement” tab. There, you will find a form that allows you to easily create new measurements. This form is shown in Fig. 1. Note that in this form you can set the type of measurement you want to run (Ping, Traceroute, DNS, etc.), the probes from which you want to send the messages, the destination address (url or IP), and many technical parameters that are specific of the selected type of measurement. You can also set the periodicity and duration of the experiment.



Create a New Measurement

Step 1 Definitions

Please select the type of measurement you want to create

Step 2 Probe Selection

Worldwide 10 ×

Step 3 Timing

This is a One-off: □

Start time (UTC): As soon as possible

Stop time (UTC): Never

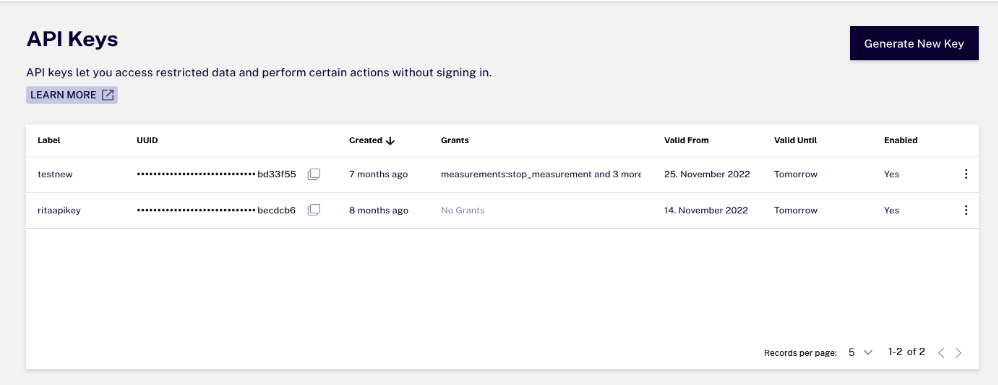
› Measurement API Compatible Specification

Fig. 1: Interactive interface to create and launch new measurements.

Using the Python API:

The second method utilizes the Python API provided by RIPE Atlas, which allows you to integrate the measurements creation in the same projects where you create your data analytics. If you prefer this option, which is much more flexible and can be automatized, you can create a script that takes the measurements for you. Ripe atlas provides "Cousteau": a Python wrapper library for RIPE Atlas API. An API key is often necessary to successfully implement requests.

1) Create the API key: Navigate to the API keys creation page on the RIPE Atlas page and set up the API key you will use for your calls. Make sure you give the right permissions to your API key to both create and retrieve your measurements.



API Keys

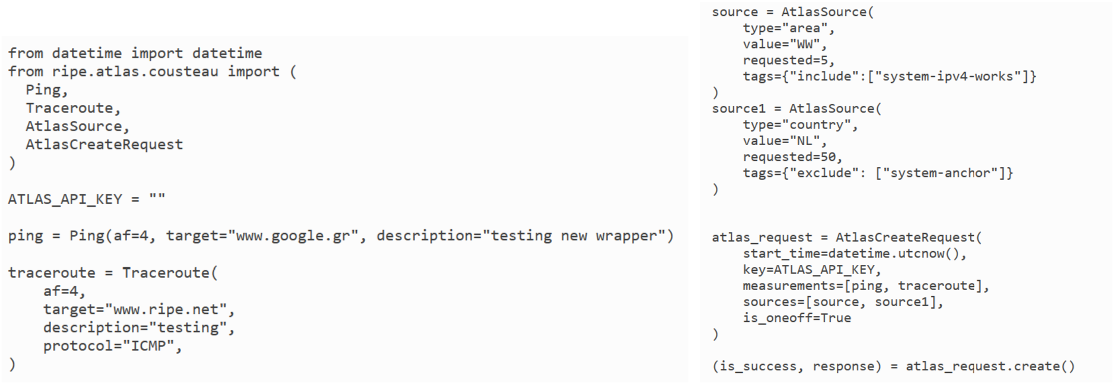
API keys let you access restricted data and perform certain actions without signing in.

Label	UUID	Created ↓	Grants	Valid From	Valid Until	Enabled
testnew	••••••••••••••••••••••••••••bd33f55	7 months ago	measurements:stop_measurement and 3 more	25. November 2022	Tomorrow	Yes
ritaapikey	••••••••••••••••••••••••••••becdcb6	8 months ago	No Grants	14. November 2022	Tomorrow	Yes

Records per page: 5 1-2 of 2

Fig. 2: Interface to create, edit, and remove API Keys

2) Making the measurement: The Cousteau documentation provides some boilerplate text to provide a starting point. We provide a Jupyter Notebook with a template to create your own measurements. The only requisite is to have an API key and define your parameters.
[Comment for instructor:] Here, make them go to this Jupyter notebook and create the measurement. They will need the API key.



```
from datetime import datetime
from ripe.atlas.cousteau import (
  Ping,
  Traceroute,
  AtlasSource,
  AtlasCreateRequest
)

ATLAS_API_KEY = ""

ping = Ping(af=4, target="www.google.gr", description="testing new wrapper")

traceroute = Traceroute(
    af=4,
    target="www.ripe.net",
    description="testing",
    protocol="ICMP",
)
```

```
source = AtlasSource(
    type="area",
    value="WW",
    requested=5,
    tags={"include":["system-ipv4-works"]}
)
source1 = AtlasSource(
    type="country",
    value="NL",
    requested=50,
    tags={"exclude": ["system-anchor"]}
)


atlas_request = AtlasCreateRequest(
    start_time=datetime.utcnow(),
    key=ATLAS_API_KEY,
    measurements=[ping, traceroute],
    sources=[source, source1],
    is_oneoff=True
)

(is_success, response) = atlas_request.create()
```

Fig. 3: Cousteau boilerplate text for creating a measurement from two probes to www.ripe.net

IV Retrieving data from RIPE Atlas and the course materials

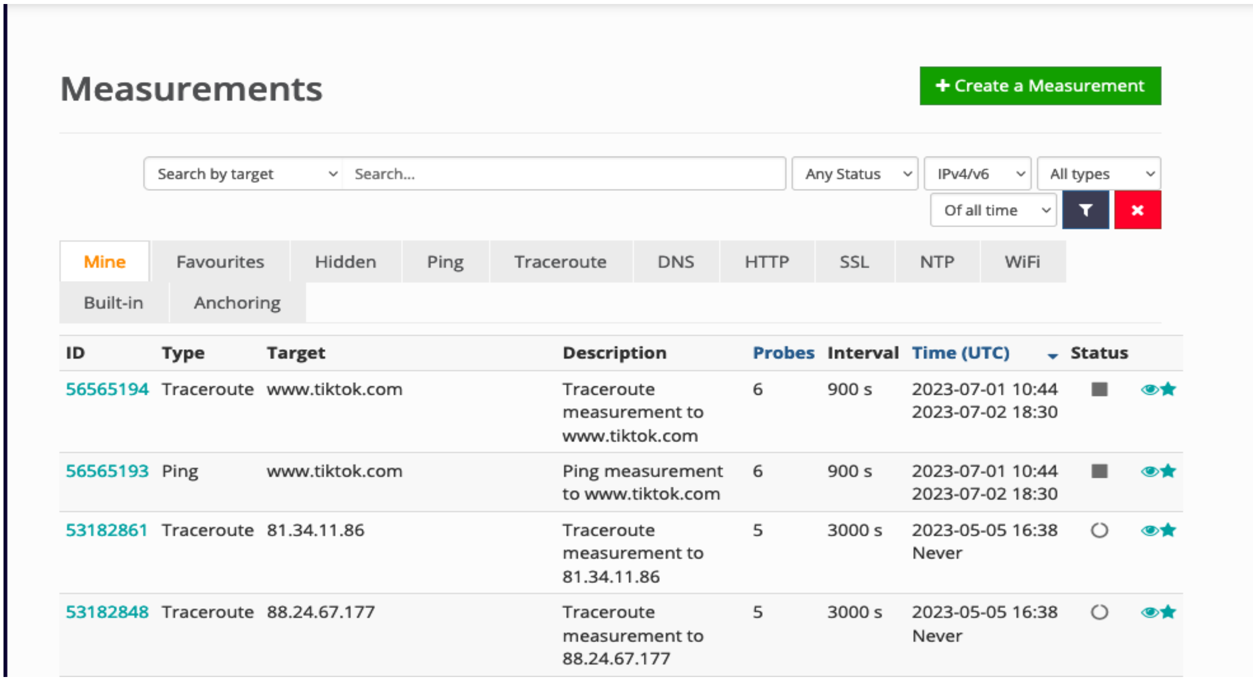
We now know how to create measurements. But, how can we get the results?

Next, we provide two different manners of obtaining the data you need to run your experiments. First, we show how to retrieve data from RIPE Atlas, both from the website (simple method) and the Python API (more advanced). Second, we explain how you can retrieve the data that will be used in the subsequent analysis.

[Comment for instructor:] Again, depending on the level of the students, you decide whether to skip the API method.

A From the website

You can access the public measurements that have been made by all the collaborators of RIPE atlas without being registered. You can also follow on-going measurements. They are shown in this site. If you are registered, you will find a tab “Mine” with the measurements you created.

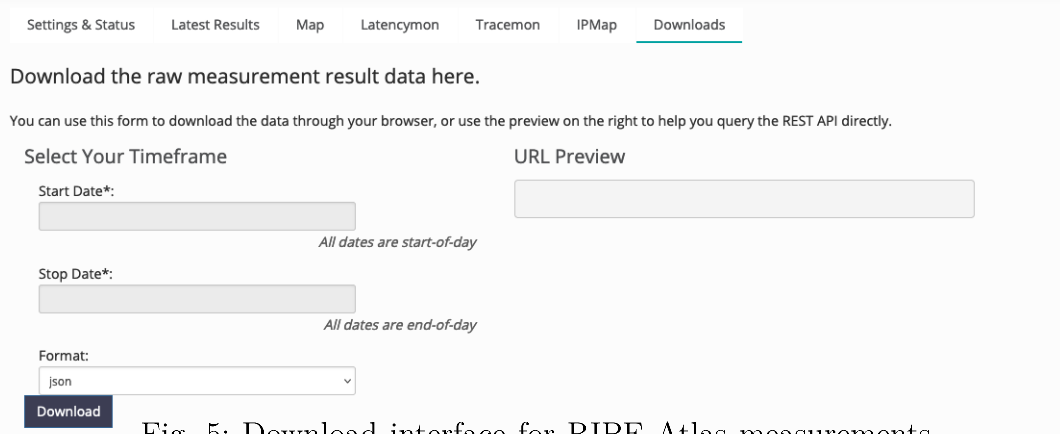


Measurements

ID	Type	Target	Description	Probes	Interval	Time (UTC)	Status
56565194	Traceroute	www.tiktok.com	Traceroute measurement to www.tiktok.com	6	900 s	2023-07-01 10:44 2023-07-02 18:30	■
56565193	Ping	www.tiktok.com	Ping measurement to www.tiktok.com	6	900 s	2023-07-01 10:44 2023-07-02 18:30	■
53182861	Traceroute	81.34.11.86	Traceroute measurement to 81.34.11.86	5	3000 s	2023-05-05 16:38 Never	○
53182848	Traceroute	88.24.67.177	Traceroute measurement to 88.24.67.177	5	3000 s	2023-05-05 16:38 Never	○

Fig. 4: Interface to browse through all the measurements

The data retrieval is very simple: Click on the measurement corresponding with the experiment you wish to download. This will lead you to an interface as the one in Fig. 5. You can specify the data range you are interested in. Click on the download tab and proceed to obtain the measurement data file.



Download the raw measurement result data here.

You can use this form to download the data through your browser, or use the preview on the right to help you query the REST API directly.

Select Your Timeframe

Start Date*:

All dates are start-of-day

Stop Date*:

All dates are end-of-day

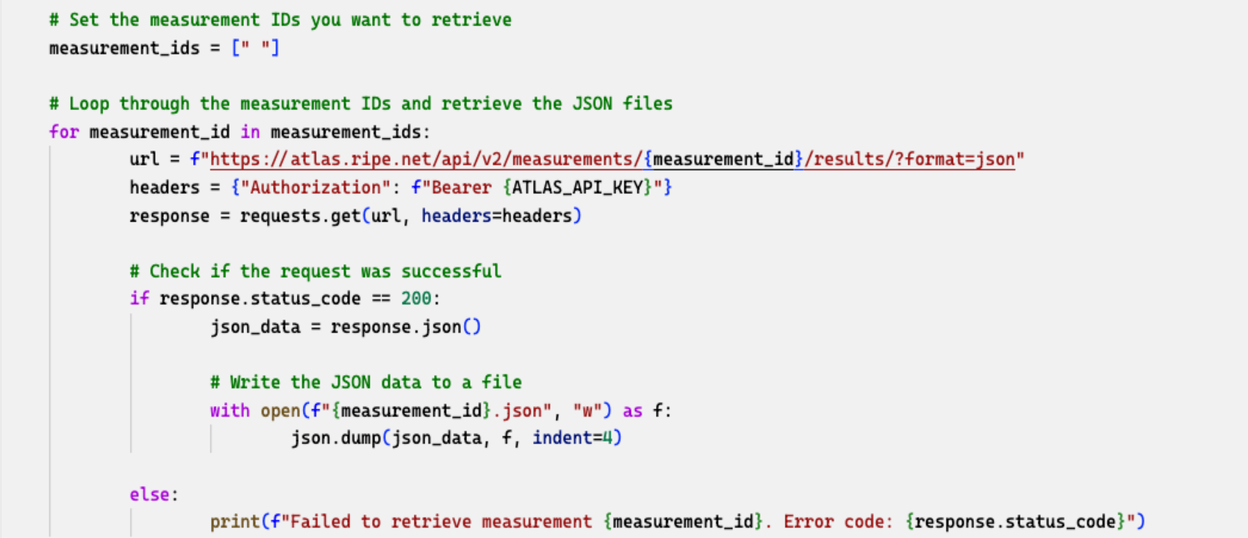
Format: json

URL Preview

Fig. 5: Download interface for RIPE Atlas measurements

B Using the API

In general, the API code for experiment retrieval looks like this. This is a snapshot but if you want to view the it clearly look at the latency1_whereIsTheCloud_B_getMeasurement.ipynb notebook on parsing measurements.



```
# Set the measurement IDs you want to retrieve
measurement_ids = [" "]

# Loop through the measurement IDs and retrieve the JSON files
for measurement_id in measurement_ids:
        url = f"https://atlas.ripe.net/api/v2/measurements/{measurement_id}/results/?format=json"
        headers = {"Authorization": f"Bearer {ATLAS_API_KEY}"}
        response = requests.get(url, headers=headers)

        # Check if the request was successful
        if response.status_code == 200:
                json_data = response.json()

                # Write the JSON data to a file
                with open(f"{measurement_id}.json", "w") as f:
                        json.dump(json_data, f, indent=4)

        else:
                print(f"Failed to retrieve measurement {measurement_id}. Error code: {response.status_code}")
```

Fig. 6: API call to retrieve measurements

V Closing Remarks & Extension

There are several experiments available in JSON format in the public GitHub repository.

If you are interested in understanding how to analyze the data you may have just created or retrieved, please check the other guide available in the repository. In that guide, you will play with the data to see how latency behaves in time and space, and you will use statistical and machine learning methods to understand the latency data.

References

[1] V. Manojlovic. Ripe atlas tutorial. [Online]. Available: https://www.sanog.org/resources/sanog27/SANOG27-Tutorial_RIPE_Atlas.pdf