EdgeNet, a Production Internet-Scale Container-Based Distributed System Testbed

Berat Can Şenel, Maxime Mouchet, Justin Cappos, Timur Friedman, Olivier Fourmaux, Rick Mcgeer

▶ **To cite this version:**

HAL Id: hal-03762467

https://hal.science/hal-03762467

Submitted on 27 Aug 2022

Demo: EdgeNet, a Production Internet-Scale Container-Based Distributed System Testbed

Berat Can Şenel
Sorbonne Université, France
berat.senel@lip6.fr

Maxime Mouchet
Sorbonne Université, France
maxime.mouchet@lip6.fr

Justin Cappos
New York University, United States
jcappos@nyu.edu

Timur Friedman
Sorbonne Université, France
timur.friedman@sorbonne-universite.fr

Olivier Fourmaux
Sorbonne Université, France
olivier.fourmaux@sorbonne-universite.fr

Rick McGeer
engageLively, United States
rick.mcgeer@engagelively.com

Abstract—The EdgeNet software is free, open-source, liberally licensed code that extends the Kubernetes container orchestration system to the edge cloud. We use this code to run the EdgeNet testbed, an internet-scale edge cloud for distributed systems researchers. This demonstration showcases three features of EdgeNet: its multitenancy model, its multi-provider aspect, and its geographically-based selective deployment capability. *Multitenancy* allows multiple teams to use the platform concurrently; being *multi-provider*, independent contributors can make nodes available to the platform; and *selective deployment* facilitates location-based placement of software. Under our guidance, demo participants invoke the Kubernetes command-line interface to use the testbed. In so doing, they get experience with the testbed, which they can continue to use afterwards. They also gain insight into how the demonstrated features are useful for edge cloud container deployment in general. Participants who volunteer to help EdgeNet nodes receive Odroid devices to host in their homes or workplaces.

Index Terms—Distributed testbed, containers, multitenancy, Kubernetes, edge computing, virtual private networks.

I. Introduction

Distributed system researchers have long used edge cloud testbeds such as PlanetLab [1], PlanetLab Europe [2], Geni [3], G-Lab [4], V-Node [5], and Savi [6] to launch experiments at internet scale, with agents experiencing real-world propagation latencies, and congestion-related delays and losses as they communicate over the public network from vantage points located across the globe. Despite their notable successes, these testbeds suffered from serious software and hardware sustainability issues that made them hard to maintain and upgrade. The problem that we have sought to overcome is how to construct a testbed that will both be easier for the research community to use, and less costly for us, its administrators, to maintain.

The solution that we demonstrate is the EdgeNet testbed,[1] which has been successfully supporting third party experiments since entering production three years ago [7]. The key to both EdgeNet's ease of use and sustainability is that it has been built as a small set of extensions[2] to Kubernetes.[3] Since Kubernetes is the de facto industry standard for container orchestration, any user who is familiar with building Docker images and deploying containers with the kubectl[4] command-line tool already knows most of what they need in order to deploy their experiments on EdgeNet. And since Kubernetes is liberally-licensed, free, and open-source software, we, the EdgeNet administrators, minimize our dependency on bespoke code. Furthermore, a virtual machine can serve as a Kubernetes node, so EdgeNet does not rely upon dedicated hardware, as many previous testbeds have done. By simply running a quick script, anyone can contribute their Linux machine or a virtual machine (VM), even one that is located behind a firewall, to the EdgeNet cluster.

Our demonstration pairs practical hands-on experience of EdgeNet workflows with discussion of the innovative features that enable three of these workflows.

1) **Creating an Account / Multitenancy** Each participant obtains an EdgeNet account and is able to share their isolated environments with their colleagues (Sec. VI-A). In so doing, they learn about the novel multitenancy framework that we have conceived for Kubernetes (Sec. III).

2) **Launching an Experiment / Selective Deployment** Each participant uses kubectl to launch a pre-prepared experiment on nodes around the world (Sec. VI-B). This also serves to demonstrate the novel *selective deployment* extension that we have developed for Kubernetes, which allows the choice of nodes based on geographic criteria (Sec. IV).

3) **Contributing a Node / Multi-Provider** Each participant who is willing to spin up a virtual machine on a server

---

[1]The EdgeNet testbed https://edge-net.org
[2]The EdgeNet software https://github.com/EdgeNet-project
[3]Kubernetes https://kubernetes.io/
[4]kubectl https://kubernetes.io/docs/reference/kubectl/

that they own, so as to contribute it as an EdgeNet node, runs a simple script that causes the VM to join the EdgeNet cluster within a matter of minutes (Sec. VI-C). Participants who volunteer to host a physical device at their home or workplace are handed an ODROID mini-computer to serve as an EdgeNet node. In this way, participants also gain insight into EdgeNet's multi-provider model, and the means by which we are able to install nodes behind NATs (Sec. V).

The demonstration's **contributions** to the community are:

- Participants who conduct not-for-profit research receive accounts that allow them to continue using the testbed after the demo session.
- Those who volunteer to do so are able to support the research that is conducted on EdgeNet by hosting testbed nodes at either their home or their workplace.
- Participants are introduced to the liberally-licensed, free, open-source EdgeNet software, which they can use to deploy their own edge clouds.
- Those who are not yet familiar with Kubernetes are introduced to the basic notions of how to use this industry-standard container orchestration system.
- People who are interested in container orchestration for the edge cloud learn about our selective deployment, multitenancy, and multi-provider extensions to Kubernetes.

Sec. II describes the current status of EdgeNet. Sec. III provides an overview of how our multitenancy model works. Sec. IV describes the software entities that allow deployments to be made based on node locations. Sec. V discusses the node contribution procedure. The steps of the demonstration are described in Sec. VI.

II. STATUS

We operate two testbeds using the EdgeNet software: a primary testbed for researchers, called, simply, EdgeNet; and a testbed called EdgeNet@home that runs a set of experiments that have been curated for the more security-sensitive environment of nodes that are hosted in people's homes.

The primary testbed has been running for over three years, with over 60 nodes distributed worldwide: 4 in Europe, 1 in Brasil, 1 in Japan, 1 in Turkey, 1 in Canada and the others in the United States, as seen in Fig. 1. There are three control plane nodes and there is a DNS load balancer to ensure high availability. This cluster mainly consists of nodes hosted at universities and, in the United States, on the GENI infrastructure. To date, over 40 tenants have been registered with the primary testbed, each tenant consisting of an individual researcher or team of researchers. This cluster consists in around 212 vCPUs, with 249 GiB of memory; with each node typically offering either 2 or 4 vCPUs, approximately 2 or 4 GiB of memory, and 16 GiB of storage.

The EdgeNet@home testbed has just been inaugurated, and is relatively small. We deliver ODROIDs[5] to those who would like to support research on EdgeNet@home by hosting this
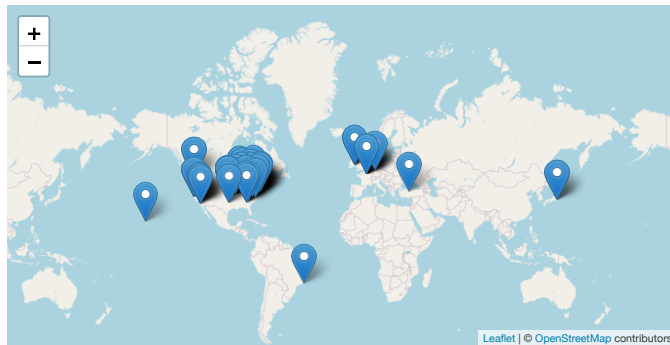

Fig. 1. Node distribution of the primary cluster.

mini-computer in their home network. The ODROIDs that we have been distributing each possess 6 ARM cores (4x Cortex-A73 + 2x Cortex-A53), 2 GB of memory, and 64 GB of eMMC storage. We also welcome VMs deployed on PCs as EdgeNet@home nodes. This cluster is only open to vetted projects and researchers, contrary to the primary cluster. The reason is that an individual is more vulnerable than a university in case of any suspicious activity on their network.

III. MULTITENANCY

A tenant is an isolated environment for a group of users. Kubernetes has a single-tenant architecture that is not suitable for simultaneous use of a cluster by multiple independent groups of users, and so there is work in the Kubernetes community on enabling multitenancy.[6] In this context, we have developed a multitenancy model for EdgeNet, with the aim of opening the testbed to as many researchers as possible while ensuring fair resource sharing and allowing each user to flexibly configure their own isolated environment.

EdgeNet's multitenancy framework has features that tackle common challenges. First, a tenant may be used by an individual, a research team, a lab, an institution like a research center, or a university. When a tenant has a large number of users, it becomes necessary to differentiate among the permissions that each user has, with some being overall administrators, others having delegated intermediate responsibilities, and others being simple users. Our framework builds on Kubernetes' role-based access control feature[7] to achieve this.

Second, when there are many users, the need arises for different teams to have isolated workspaces in which to work. Kubernetes natively provides the namespace feature,[8] a logical entity that enables resource isolation. However, vanilla Kubernetes, due to the single-tenant architecture, lacks a feature that automatically isolates tenants from each other via namespaces. For EdgeNet, we use Kubernetes namespaces to enable subsidiary namespaces.

---

[5]Hardkernel, manufacturer of ODROID https://www.hardkernel.com/

[6]See the Kubernetes Multi-Tenancy WG https://github.com/kubernetes-sigs/multi-tenancy/

[7]Kubernetes RBAC documentation https://kubernetes.io/docs/reference/access-authn-authz/rbac/

[8]Kubernetes namespaces documentation https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

Third, a multitenancy framework must ensure fair resource sharing among tenants. At the same time, the testbed administrators should be able to straightforwardly adjust the resource quota assigned to a tenant. Thus, if an experiment requires more resources than usual, the administrators can increase the tenant resource quota for a given time. EdgeNet's tenant resource quota feature that applies a resource quota per tenant addresses these needs.

Fourth, and finally, there is a concern for the reproducibility of experimental results. To address this issue, EdgeNet brings the *slice* concept into Kubernetes, where a tenant can obtain isolated access to a node's resources. Once a slice is created, the containers of other tenants are removed from the reserved nodes. An experiment that is run and then rerun in a slice will face comparable on-node conditions, as it will be unaffected by resource usage by other users of that node.

IV. SELECTIVE DEPLOYMENT

EdgeNet's selective deployment feature allows users to choose particular locations around the world at which their code will be run. Two software entities enable this. First is the *node labeler*, which uses Maxmind's GeoIP2 Web Services database[9] to determine the location of a node via its IP address. Then it attaches geolocation-related labels to that node.

In addition to city, country, continent, and latitude-longitude coordinates, the labels also specify an IP address's internet service provider, autonomous system organization, and autonomous system number (ASN). The node labeler entity, for example, puts the following labels on a node located in France:

- *edge-net.io/isp=Renater*
- *edge-net.io/as=Renater*
- *edge-net.io/asn=1307*
- edge-net.io/city=Arcueil
- edge-net.io/state-iso=IDF
- edge-net.io/country-iso=FR
- edge-net.io/continent=Europe
- edge-net.io/lat=n48.799400
- edge-net.io/lon=e2.336700

The selective deployment feature uses these labels to deploy containers to desired locations, as shown in Fig. 2. In keeping with the Kubernetes approach of having the user specify a desired state while not requiring the user to enter into the details of how the system achieves that state, if there are several nodes that are available at a particular location, EdgeNet will select one, and if that node goes down, EdgeNet will redeploy to another, without the user needing to intervene.

V. MULTI-PROVIDER

As for earlier distributed testbeds for experimentation on the internet, nodes are contributed by researchers from their laboratories across the world. What makes EdgeNet different is that they can contribute VMs rather than physical machines. This means that EdgeNet can leverage existing hardware,
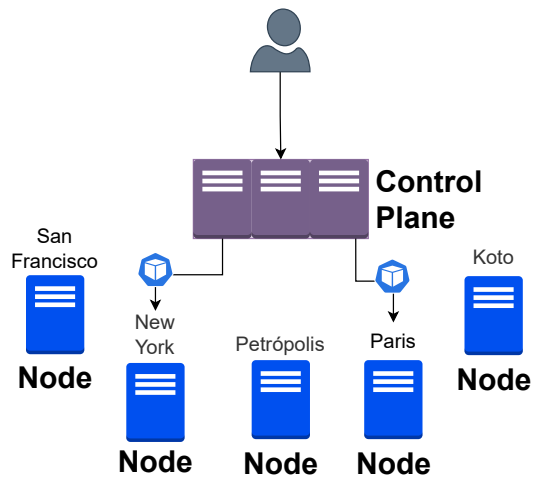

Fig. 2. Selective deployment, in which a user deploys containers to nodes in Paris and New York, and not to other locations.

lowering the administrative and logistic costs of entry into the system. The multi-provider model is illustrated in Fig. 3.

The procedure by which someone contributes a node to EdgeNet is extremely simple and takes less than ten minutes to complete. The provider runs a simple shell script that installs Ansible,[10] fetches the playbook, and orchestrates the operation. This playbook installs Kubernetes, the container runtime, and the EdgeNet agent. Having completed this step, the agent initializes in the node on which the shell script is run, and it configures the virtual private network settings for the node and the cluster. In a short time, that node will be added to the cluster. The ODROIDs that we provide use this procedure as well.

VI. DEMONSTRATION

There are three steps in the demonstration:

A. Creating an Account

A tenant may be registered on EdgeNet in one of two ways, as the participant prefers: either via the Kubernetes kubectl command-line tool or via our web-based graphical user interface.[11] In registering a new tenant, the participant obtains an EdgeNet account with administrative control over that tenant's resources. They are welcome to keep this account following the demo, and to sign up their colleagues to the same tenant so that they can collectively continue to use the EdgeNet testbed. Use of EdgeNet is free of charge to bona fide researchers at not-for-profit institutions.

B. Launching an Experiment

Following user registration, participants deploy containers to the testbed.[12] We lead them through the use of EdgeNet's

---

[9]MaxMind GeoIP2 Web Services database https://www.maxmind.com/en/geoip2-precision-services

[10]Ansible https://www.ansible.com/

[11]EdgeNet web console https://console.edge-net.org/

[12]For the detailed steps of running an experiment on EdgeNet, see our documentation: https://www.edge-net.org/pages/running-experiments.html
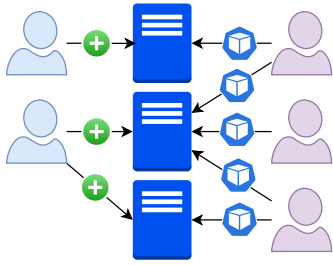
Fig. 3. Multiple providers (left, in blue) add their nodes to the EdgeNet cluster. Users (right, in purple) representing multiple tenants deploy their container-ized experiments on the contributed nodes. A provider can contribute several nodes. Each node can host the experiments of several tenants simultaneously.

selective deployment feature, using commands and configuration files that Kubernetes users are already familiar with. As for other sorts of deployments, users configure a YAML file[13] that is provided to kubectl as a command-line argument. The principal difference with vanilla Kubernetes is that users may specify locations around the world at which they would like their containers to be deployed.

This exercise gives users hands-on experience deploying containers to a real-world edge cloud. It also exposes them to the way in which EdgeNet's multitenancy model uses Kubernetes namespaces, as a user's containers are restricted to their tenant namespace. Interested participants can go further to explore EdgeNet's *subsidiary namespace* and *slice* features, which are part of an overall approach to multitenancy that, we contend, makes EdgeNet particularly suitable for edge cloud deployments at scale.

C. Contributing a Node

EdgeNet is a multi-provider testbed, allowing volunteers, whether they are testbed users or not, to contribute nodes. This step is for participants who would like to contribute one or more nodes, or who are considering doing so.[14]

If a participant is able to spin up a Linux VM on a server on which they have administrator rights, they can run a quick Ansible script that configures the VM and causes it to join the EdgeNet Kubernetes cluster. They can make this contribution on a trial basis, just for the demo, or they can maintain it for the long term.

In addition, participants are invited to host a physical device at their home or workplace. We provide ODROID mini-computers that participants can take back with them upon signing a simple acknowledgement of Sorbonne University's ownership of the device and affirming that they will host it for its intended purpose.

Since research experiments are deployed on EdgeNet nodes, those nodes' traffic can appear unusual, and so we mitigate the risk of alarms that such traffic might raise. We only open the testbed to bona fide researchers who are bound by an acceptable use policy regarding the sorts of traffic that they

can generate.[15] In addition, we maintain a trace of every packet that enters or leaves a node, and so we can identify the researchers who are responsible for any traffic that might raise questions. Our experience has been that these precautions are sufficient to allay any concerns that third parties have raised. As a further precaution, devices that are deployed in people's homes belong to a special cluster dedicated to that purpose, to which only a screened set of experiments can be deployed.

REFERENCES

[1] L. Peterson, A. Bavier, M. E. Fiuczynski, and S. Muir, "Experiences building PlanetLab," in *Proc. 7th Symposium on Operating Systems Design and Implementation (USENIX OSDI)*, 2006. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/1298455.1298489

[2] S. Fdida, T. Friedman, and T. Parmentelat, "OneLab: An open federated facility for experimentally driven future internet research," in *New Network Architectures: The Path to the Future Internet*, ser. Studies in Computational Intelligence, T. Tronco, Ed. Springer, 2010, vol. 297, pp. 141–152. [Online]. Available: https://doi.org/10.1007/978-3-642-13247-6

[3] R. McGeer, M. Berman, C. Elliott, and R. Ricci, Eds., *The GENI Book*. Springer, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-33769-2

[4] P. Mueller and S. Fischer, "Europe's mission in next-generation networking with special emphasis on the German-Lab project," in *The GENI Book*, R. McGeer, M. Berman, C. Elliott, and R. Ricci, Eds. Springer, 2016, ch. 21. [Online]. Available: https://doi.org/10.1007/978-3-319-33769-2_21

[5] A. Nakao and K. Yamada, "Research and development on network virtualization technologies in Japan: VNode and FLARE projects," in *The GENI Book*, R. McGeer, M. Berman, C. Elliott, and R. Ricci, Eds. Springer, 2016, ch. 23. [Online]. Available: https://doi.org/10.1007/978-3-319-33769-2_23

[6] A. Leon-Garcia and H. Bannazadeh, "SAVI testbed for applications on software-defined infrastructure," in *The GENI Book*, R. McGeer, M. Berman, C. Elliott, and R. Ricci, Eds. Springer, 2016, ch. 22. [Online]. Available: https://doi.org/10.1007/978-3-319-33769-2_22

[7] B. C. Şenel, M. Mouchet, J. Cappos, O. Fourmaux, T. Friedman, and R. McGeer, "EdgeNet: A multi-tenant and multi-provider edge cloud," in *Proc. 4th International Workshop on Edge Systems, Analytics and Networking (ACM EdgeSys)*, 2021. [Online]. Available: https://doi.org/10.1145/3434770.3459737

---

[13]YAML https://yaml.org/

[14]For the detailed steps of contributing a node to EdgeNet, see our documentation: https://www.edge-net.org/pages/node-contribution.html

[15]EdgeNet's AUP: https://www.edge-net.org/pages/usage-policy.html