

Provision-as-a-Service: Automating data center operations with Airflow at Cloudflare

Jet Mariscal
SRE, Edge @ Cloudflare



CLOUDFLARE'S MISSION:

Help build a better Internet

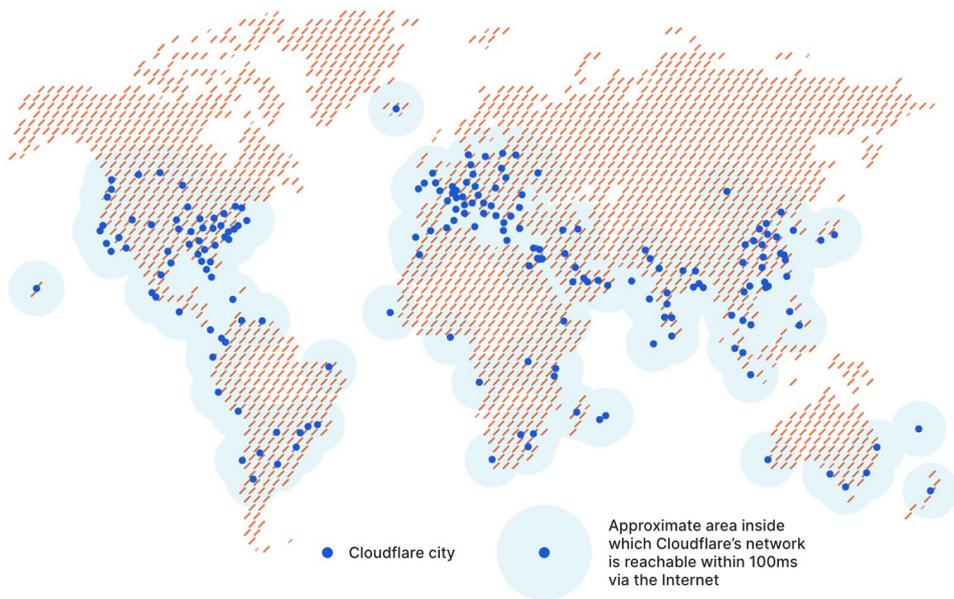
Cloudflare is an intelligent, integrated global cloud network that delivers security, performance, and reliability for all your Internet infrastructure, people and connected devices.

CLOUDFLARE'S MISSION:

Help build a better Internet

Cloudflare protects and accelerates any Internet application online without adding hardware, installing software, or changing a line of code.

Cloudflare's network operates at massive scale



~25M

Internet properties

200+

Cities and 100+ countries

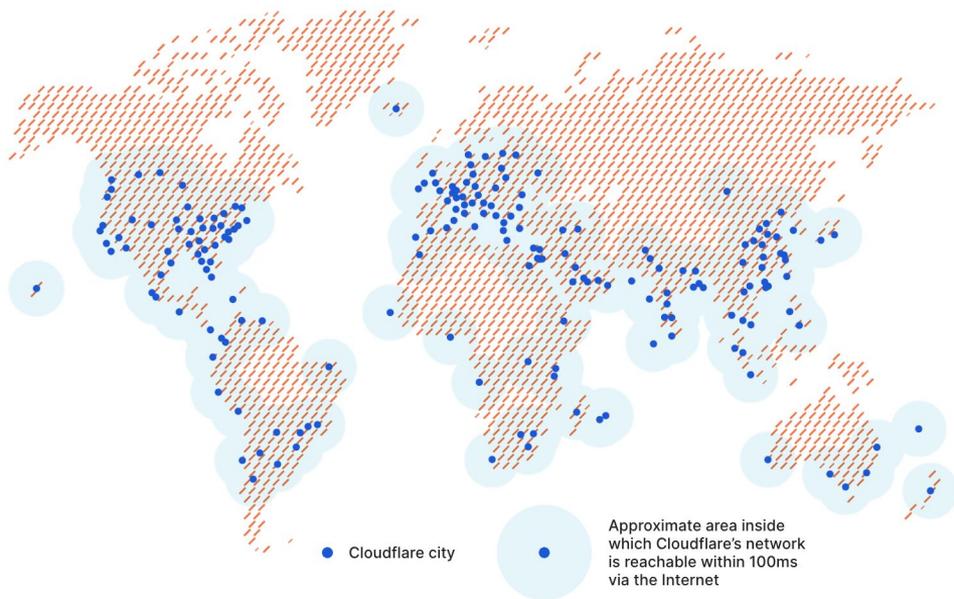
99%

Of the Internet-connected population in the developed world is located within 100 milliseconds of our network

17%

Of the Fortune 1000 are paying Cloudflare customers

Cloudflare's network operates at massive scale



25M

HTTP requests per second served on average, 30M+ at peak

9.4M

DNS queries per second, about 811 billion queries per day, and 24 trillion queries a month

70B

Cyber threats blocked each day in Q1'21

www.cloudflare.com

blog.cloudflare.com

cloudflare.tv

What is Provisioning?



Provisioning: Expansions and Decommissions



Expansion is the process of adding new machines to expand the capacity of a data center.



Decommission is the process of permanently removing machines for retirement in a data center.

Provisioning is complex



Connecting new Cloudflare servers to our network used to be so complex, in large part because of the amount of manual effort required and careful coordination between Data Center and Infrastructure Operations, Network Operations, and SREs.

Manual Provisioning: a process that can only scale so far

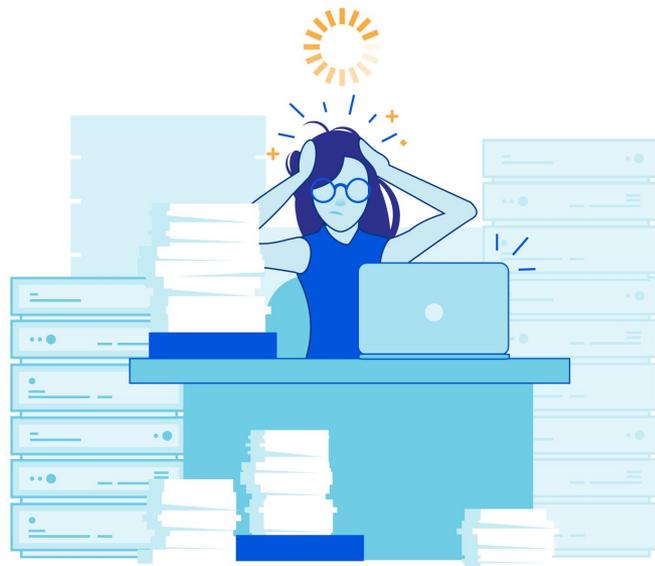


Engineers used to carefully follow steps from an extremely detailed standard operating procedure (SOP) document, often copying command-line snippets and pasting it into terminal windows.

Manual Provisioning: slow, error-prone, and very inefficient

- logging in to remote hosts via SSH
- lots of copy/pasting commands to run
- launching web browsers to view Grafana and other internal dashboards

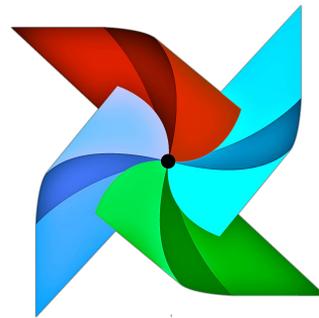
Manual Provisioning: tedious, time-consuming, and does not scale



Simultaneous expansions and/or decommissions became very challenging.

Provision-as-a-Service: Automation with Apache Airflow

- Totally eliminated the need of using SSH
- Guaranteed consistency, compared to any manual actions
- Democratized the provisioning process
- Faster and safer expansions and decommissions
- Eliminated toil



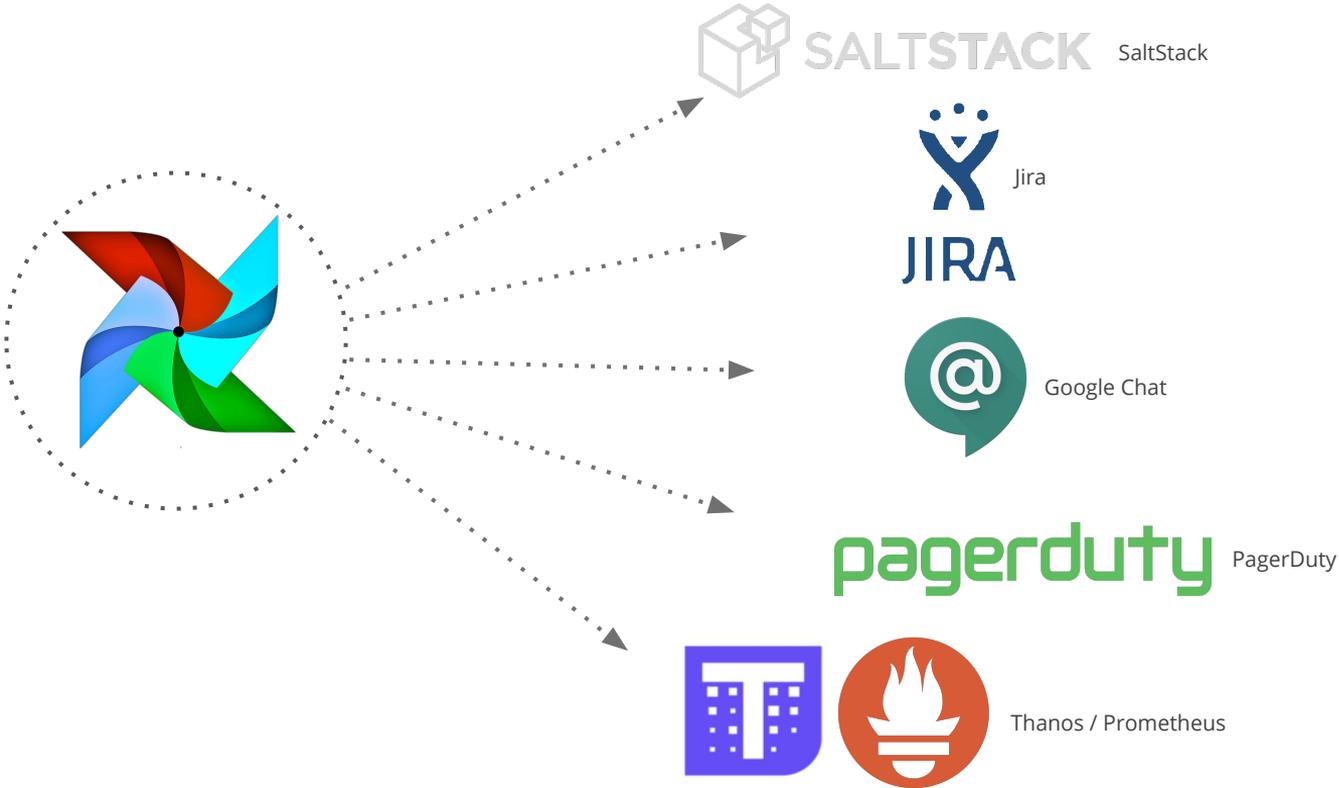
Cut by 90% the amount of time our team spent on
mundane operational tasks.

Replacing manual steps with an API-call equivalent

1. Login to a remote system.
2. Copy and paste the command in the terminal.
3. Replace the router placeholder in the command snippet with the actual value.
4. Execute the command.

```
enable_anycast = builder.wrap_class(AsyncSaltAPIOperator)(
    task_id='enable_anycast',
    target='{{ params.netops }}',
    function='cmd.run',
    fun_kwargs={'cmd': 'salt {{ get_router(params.colo_name) }} '
                  'anycast.enable --out=json --out-indent=-1'},
    salt_conn_id='salt_api',
    trigger_rule='one_success')
```

Custom Operators to integrate with other systems



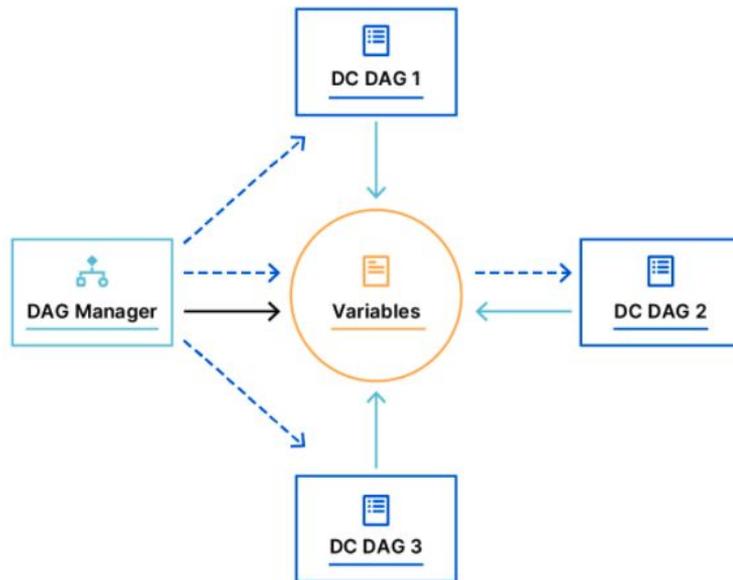
Adapting tasks for preconditions and human intervention

Using sensors to set dependencies between tasks or even DAGs, so that one does not run until the dependency has been met.

```
verify_node_dns = builder.wrap_class(DNSSensor)(
    task_id='verify_node_dns',
    zone=domain,
    nodes_from='{{ to_json(run_ctx.globals.import_nodes_via_mpl) }}',
    timeout=60 * 30,
    poke_interval=60 * 10,
    mode='reschedule')
```

```
verify_jira_input = builder.wrap_class(InputSensor)(
    task_id='verify_jira_input',
    var_key='jira',
    prompt='Please provide the Change Request ticket.',
    notify=True,
    require_human=True)
```

Accepting inputs and responding to human interventions



Custom forms for accepting user inputs

Exclude Nodes

Data Center

Please choose a data center

Please select a data center.

Jira Ticket *

Jira Ticket

Please provide the DCA ticket id for this operation.

Nodes *

Nodes

Override with this space-separated list of nodes for this operation.

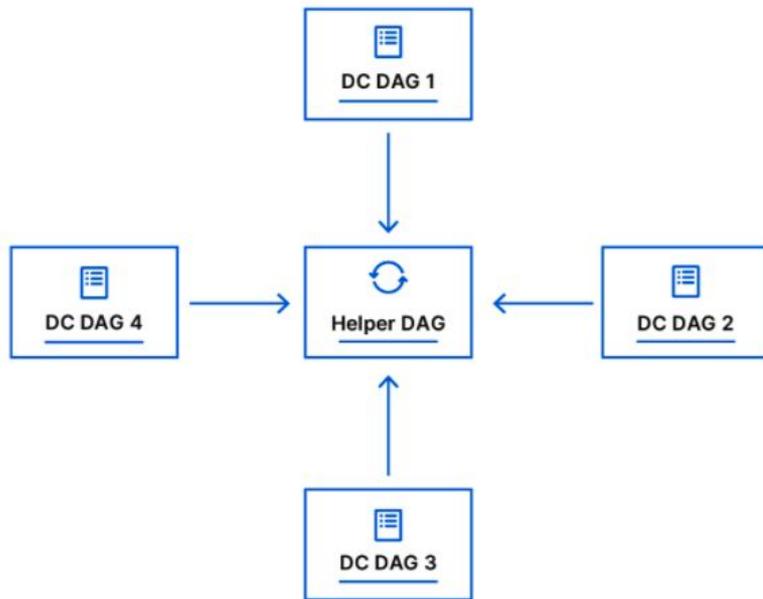
Operation

expansion

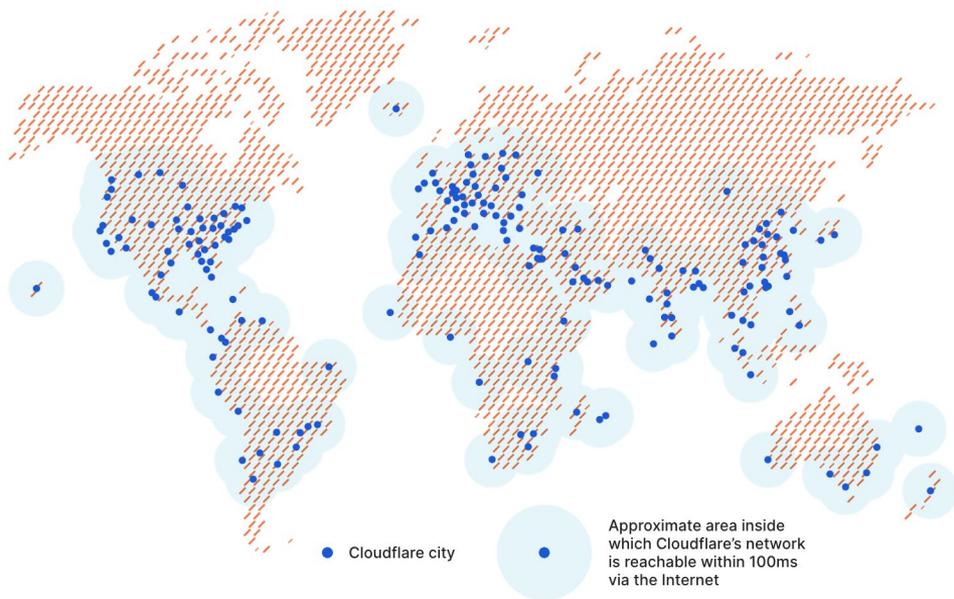
Save



Solving complex workflows with Branching and Multi-DAGs



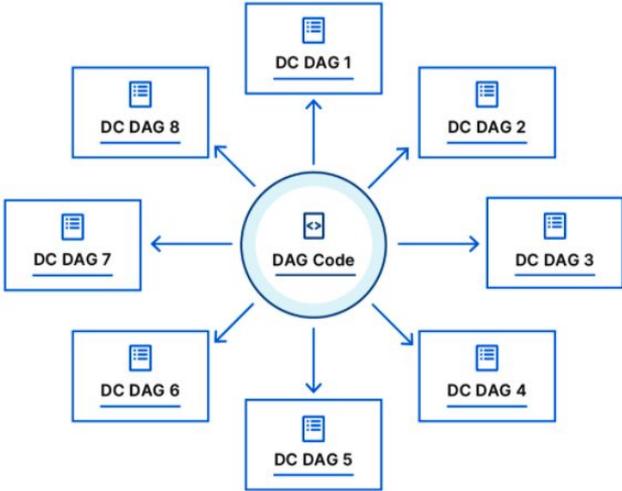
Creating DAGs that scale and executing tasks at scale



Phase 1 - machines are powered on, boots our custom Linux kernel

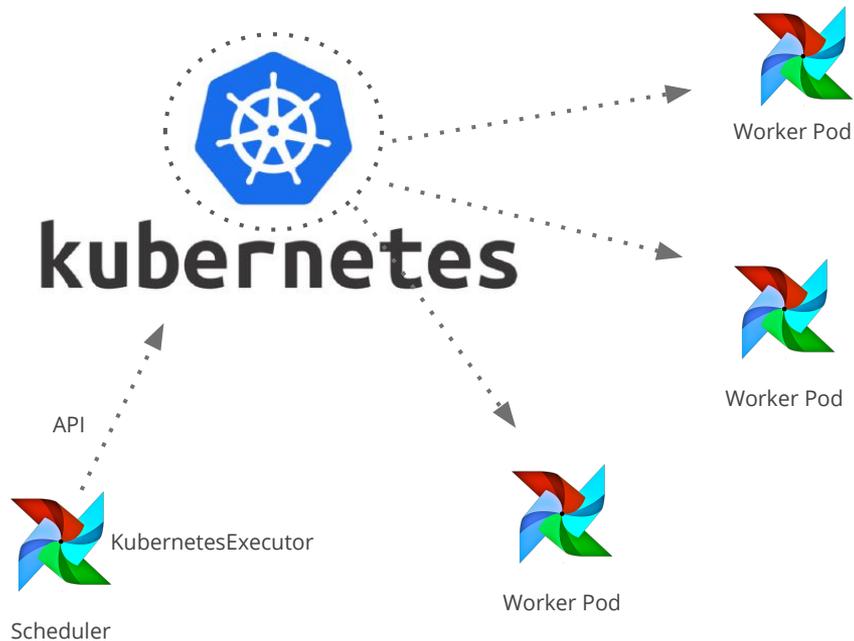
Phase 2 - newly provisioned machines are enabled to receive production traffic

Creating DAGs that scale



Generating DAGs for each new data center instantly, without writing a single line of code.

Executing DAGs at scale



KubernetesExecutor - creates a new worker pod for every task instance that needs to be executed

The worker pod gets killed on completion of the task.

Improving the Usability and User experience



DAGs

Security ▾

Browse ▾

Admin ▾

Docs ▾

Provisioning ▾

Hey there! PraaS Advisor does the heavy work for you, so you can sit back and relax!

Decom Node Generator

Expansion Phase 1

Override Nodes

Exclude Nodes

Expansion Advisor

IPs Availability Check

Primary DM Transition

INAT Operations

Improving the Usability and User experience



The screenshot displays the Apache Airflow web interface. At the top left, the Airflow logo is followed by navigation links: DAGs, Security, Browse, Admin, Docs, and Provisioning. On the top right, the current time is 07:04 UTC and there is a user profile icon labeled 'AA'. A yellow notification banner at the top contains the text 'DEMO01-001 - DEMO1 is ready for phase 2 expansion' with a close button 'x'. Below the notification is a grey bar with a dropdown arrow and the text 'Expansion Plan'. A blue button labeled 'Trigger DAG' is positioned below the grey bar.

Improving the Usability and User experience

DEMO01-001 - DEMO1 is ready for phase 2 expansion

Expansion Plan

Expansion - enable_metals_demo1

Batch 1 for 64 metals, 18%

Batch	Node Name	Node Status
1	demo	P

Improving the Usability and User experience



Please confirm the final list of nodes to decommission by selecting the nodes below.

Decom Nodes Generator

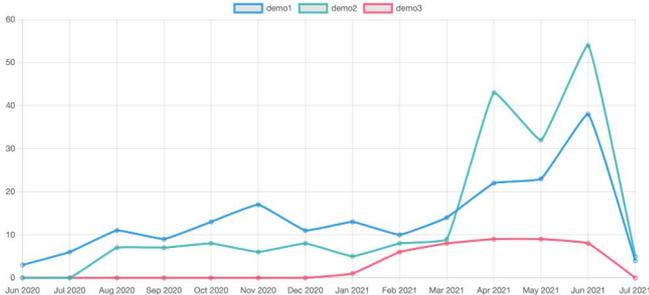
Total Candidate Nodes: 64

<input type="checkbox"/>	Node Name	Serial Number	Node Status	Rack	Hardware Generation
<input checked="" type="checkbox"/>	demom4	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo100	demo	V	demo (demo:06:61120:0501)	demo
<input checked="" type="checkbox"/>	demo101	demo	V	demo (demo:06:61120:0502)	demo
<input type="checkbox"/>	demo102	demo	P	demo (demo:06:61120:0501)	demo
<input checked="" type="checkbox"/>	demo38	demo	V	demo (demo:06:61120:0502)	demo
<input type="checkbox"/>	demo39	demo	R	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo40	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo41	demo	V	demo (demo:06:61120:0502)	demo
<input type="checkbox"/>	demo42	demo	R	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo43	demo	R	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo44	demo	R	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo45	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo46	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo47	demo	R	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo48	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo49	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo50	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo51	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo52	demo	R	demo (demo:06:61120:0502)	demo
<input type="checkbox"/>	demo53	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo54	demo	V	demo (demo:06:61120:0502)	demo
<input checked="" type="checkbox"/>	demo55	demo	V	demo (demo:06:61120:0502)	demo

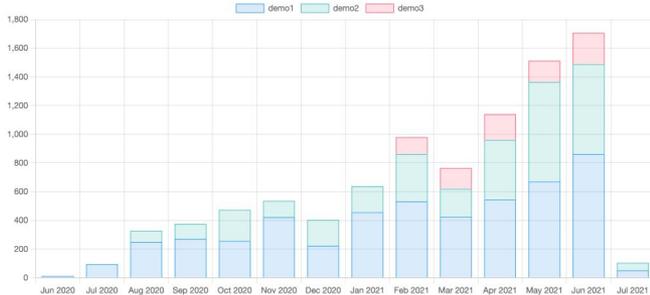
Custom dashboards for better insights



427 operations - Demo1 194, Demo2 192, Demo3 41



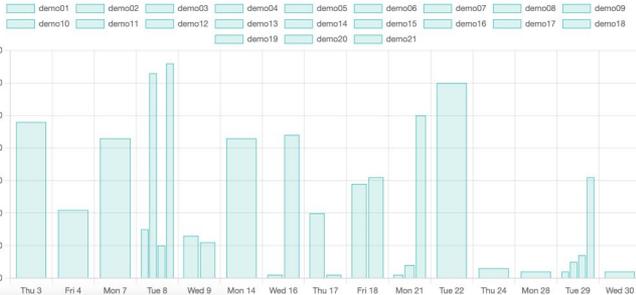
9,074 demo - Demo1 5,079, Demo2 3,186, Demo3 809



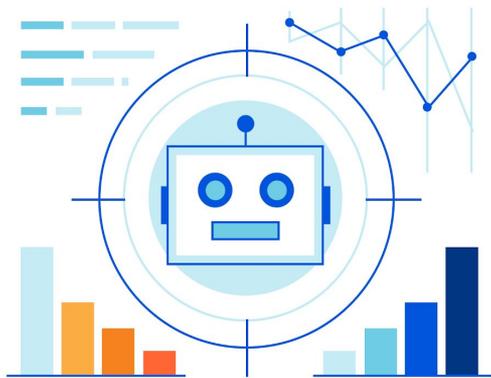
Demo1 with 218 demos - Jun 2021



Demo2 with 626 demos - Jun 2021



Ultimate Goal: Autonomous Provision-as-a-Service



For expansions, our ultimate goal is a fully autonomous system that monitors whether new servers have been racked in our edge data centers — and automatically triggers expansions — with no human intervention.

Thank You!

Like our network, **Cloudflare** continues to rapidly grow. If working at a rapidly expanding, globally diverse company interests you, we're **hiring** for scores of position in the **Infrastructure Engineering** team.