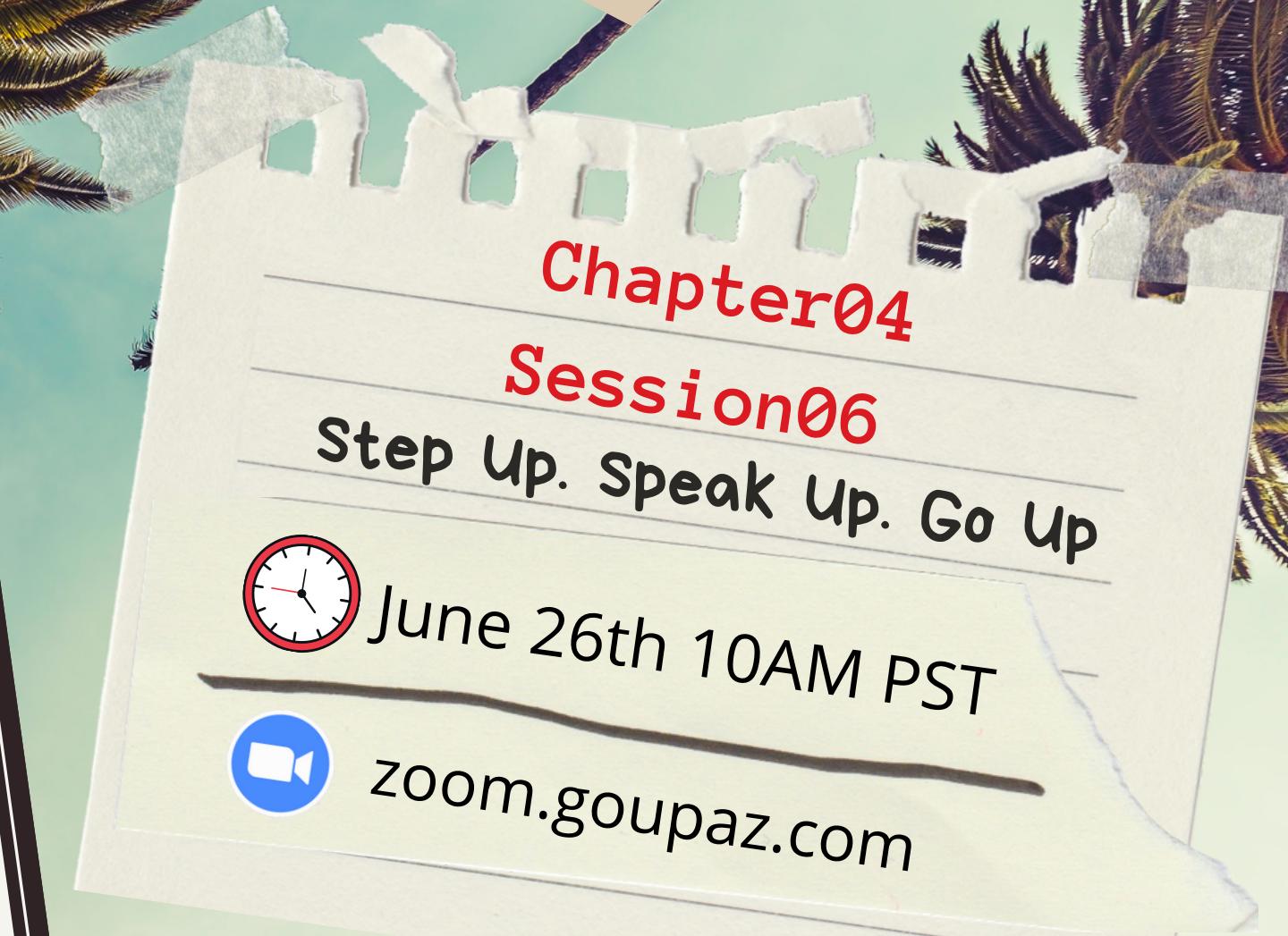
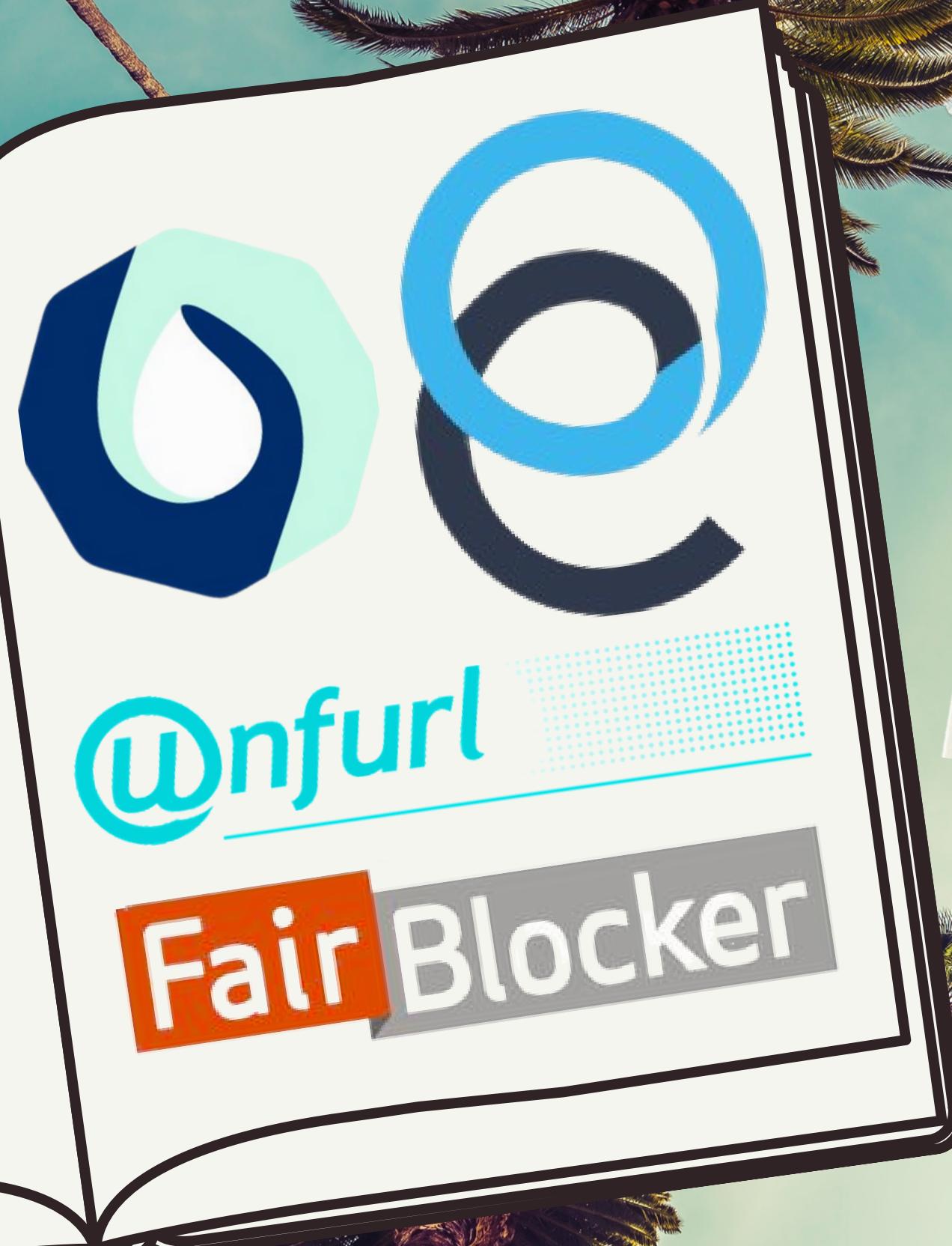


Building a Free and Open Cloud One Deployment at a time with Unfurl



Adam Souzis

Founder @ OneCommons
Co-Founder & Cto @ Octavia
Wellness
CTO @ Fairblocker
Co-Founder @ Graphite

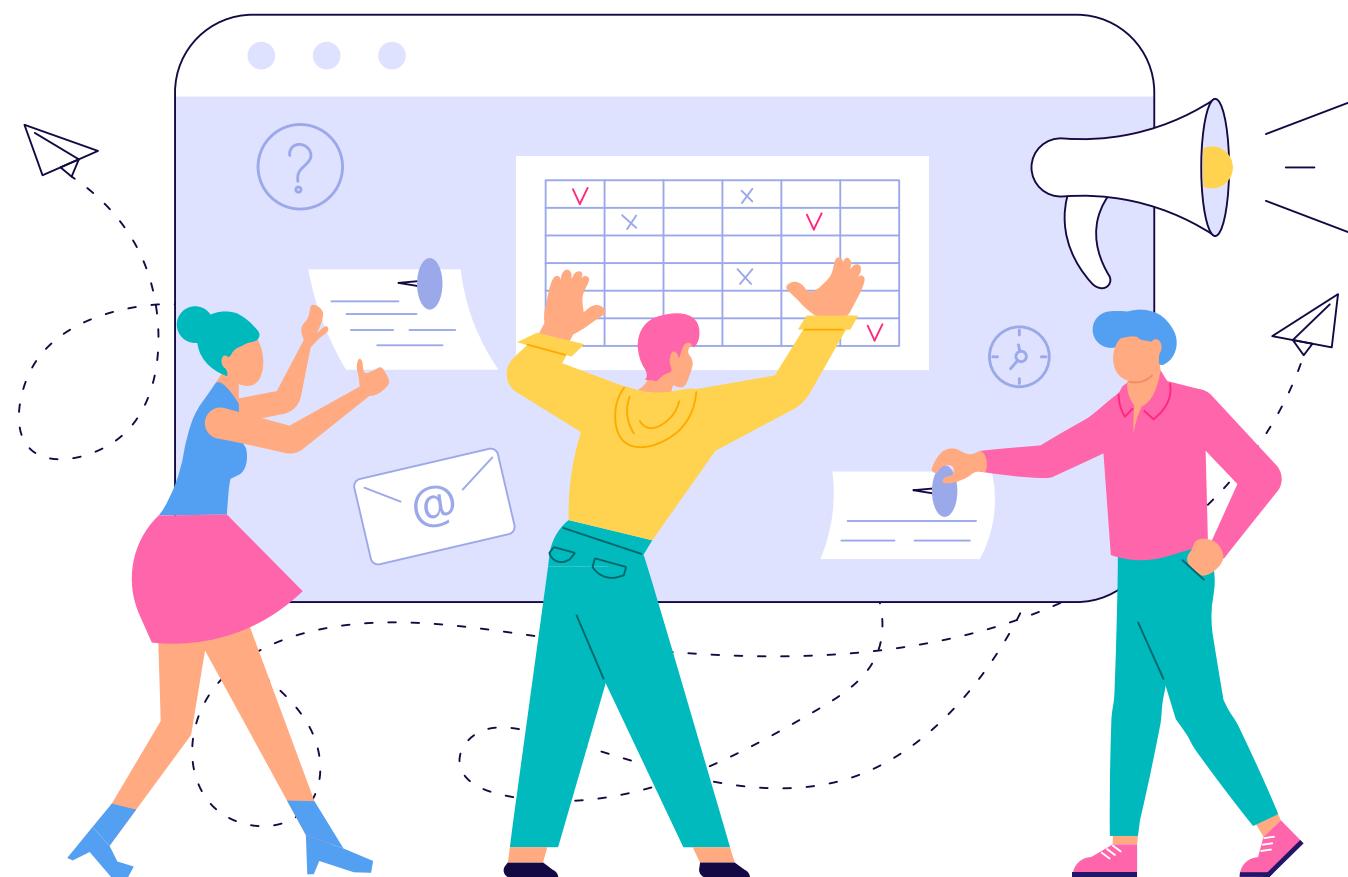


Hosted by



Metabob

Agenda



- **10:00-10:10 | Kickoff**
 - Introduction
 - Chit Chat
- **10:10-11:10 | Presentation**
 - Speaker
- **11:10-11:15 | Break**
 - Pre Q&A
- **11:15-11:30 | Q&A**
 - Discussion
- **11:30-11:35 | Wrap Up**
 - Contact Speaker

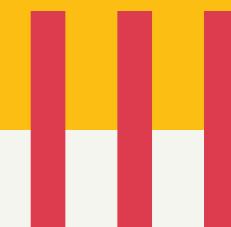
Hosted by

Axel L

Growth Marketing



Passionate about sports and technologies enhancing employee productivity



Metabob

It's the fast, easy, and visual way of debugging code.

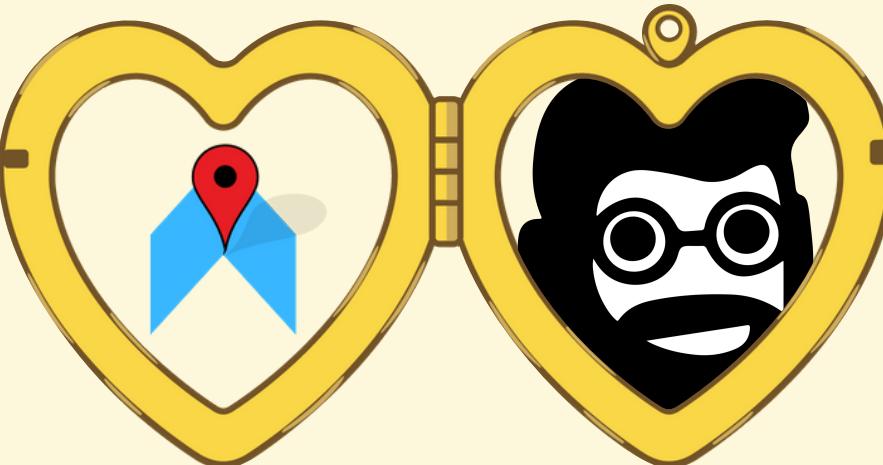
<https://metabob.com>

GOUP

Community driven Open source accelerator!
<https://goupaz.com>

Chapter04

#contributor



GOUP

Community driven Open source accelerator!
<https://goupaz.com>



YOU

GOUP 02hero/Chapter04
Contributor
About YOU

Step Up. Speak Up. Go Up.

Metabob

It's the fast, easy, and visual way of
debugging code.

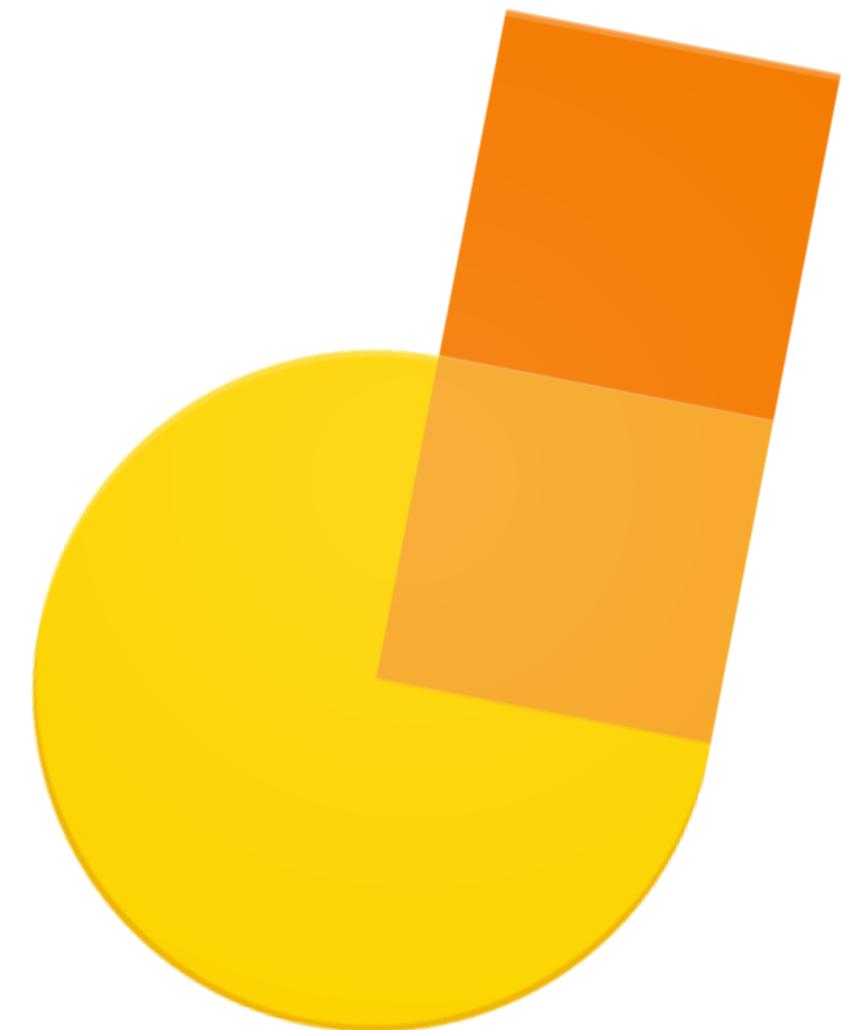
<https://metabob.com>

Are you ready?

Let's Begin!

CHITCHAT

JAMBOARD



What do you wish to learn from this webinar?

I wish to learn how Adam validates his business ideas

I wanna learn from Adam as much as possible today!

Sako - Competition landscape? Vs. with public cloud providers

Akhil Here - I've No Idea About Deployment. I Would Love To Know About It.

Different user growth projects

UX design & Accounting & Agile & Contribution enhancement

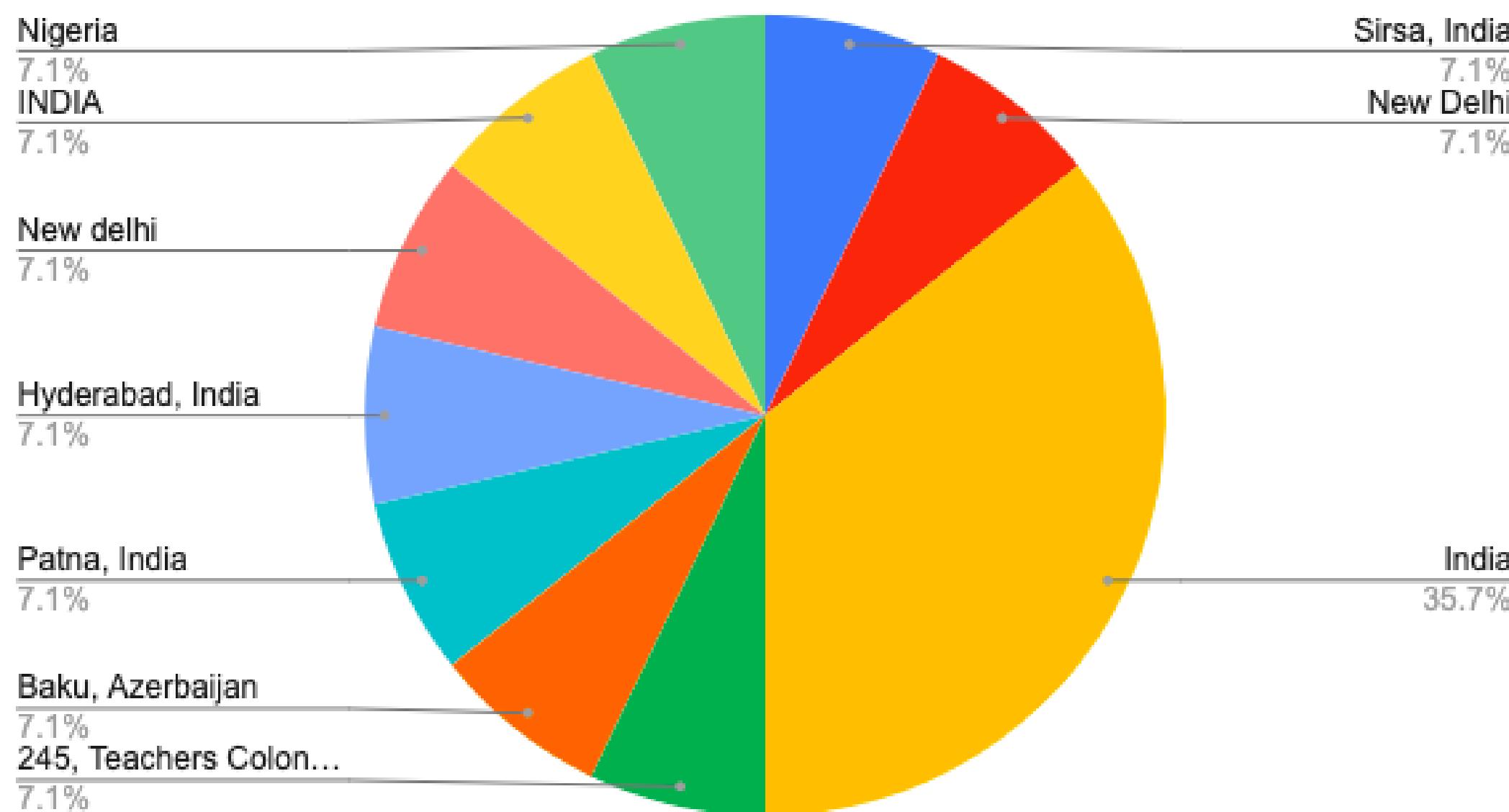
Sak0 - I was working on Service Mesh talk for MLH

I am Chandan. Want to learn about all stuffs from & in the topic itself "Building a free and open cloud one deployment at a time with Unfurl"

I am Biranchi and want to know what actually a deployment is ?

Audience

Count of Location



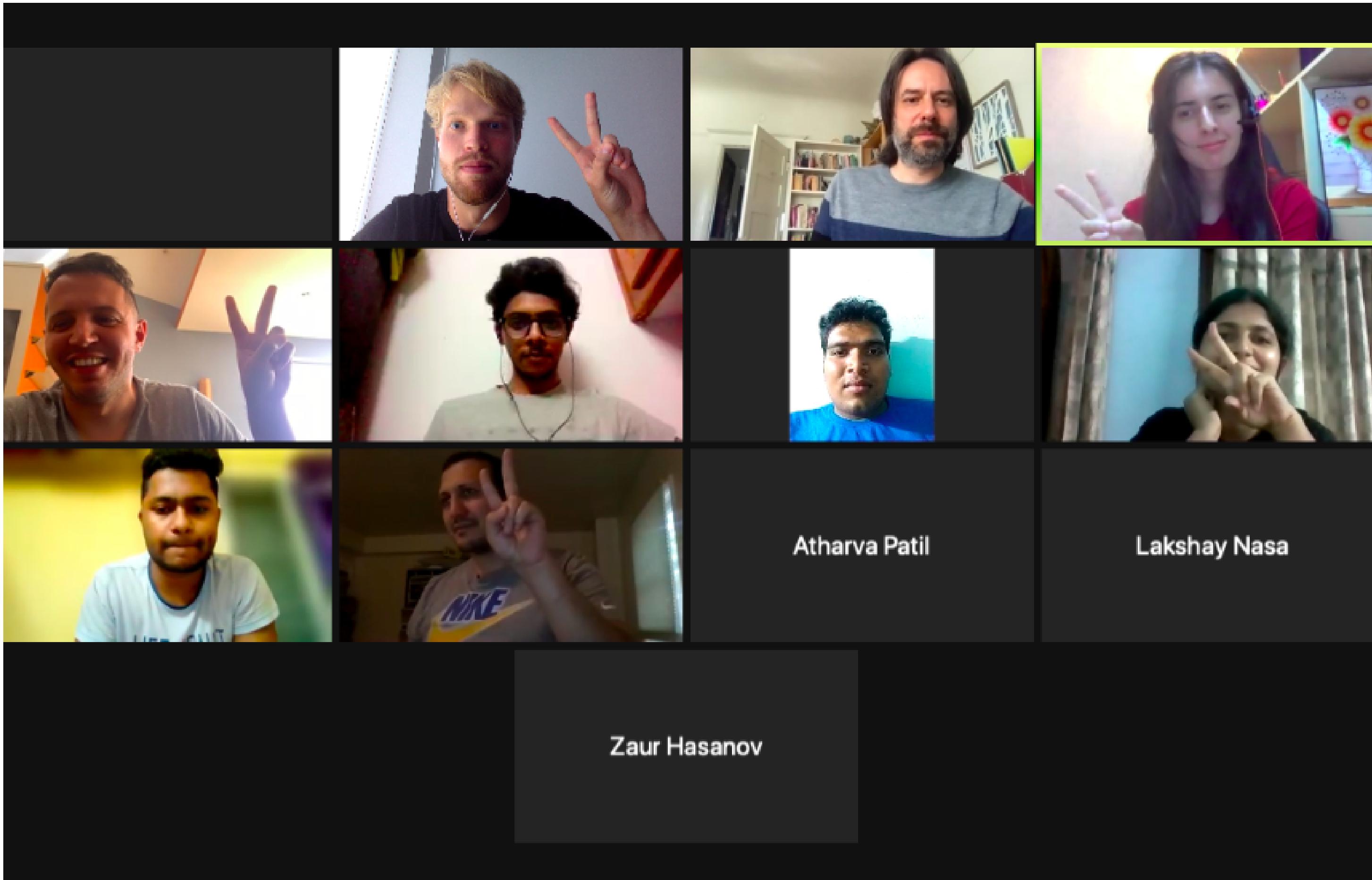
Code of Conduct

- 1 Learn, benefit, contribute
- 2 No marketing, selling, competing
- 3 Equality despite roles & bg



Photo Shoot Time

Please turn on your camera :D

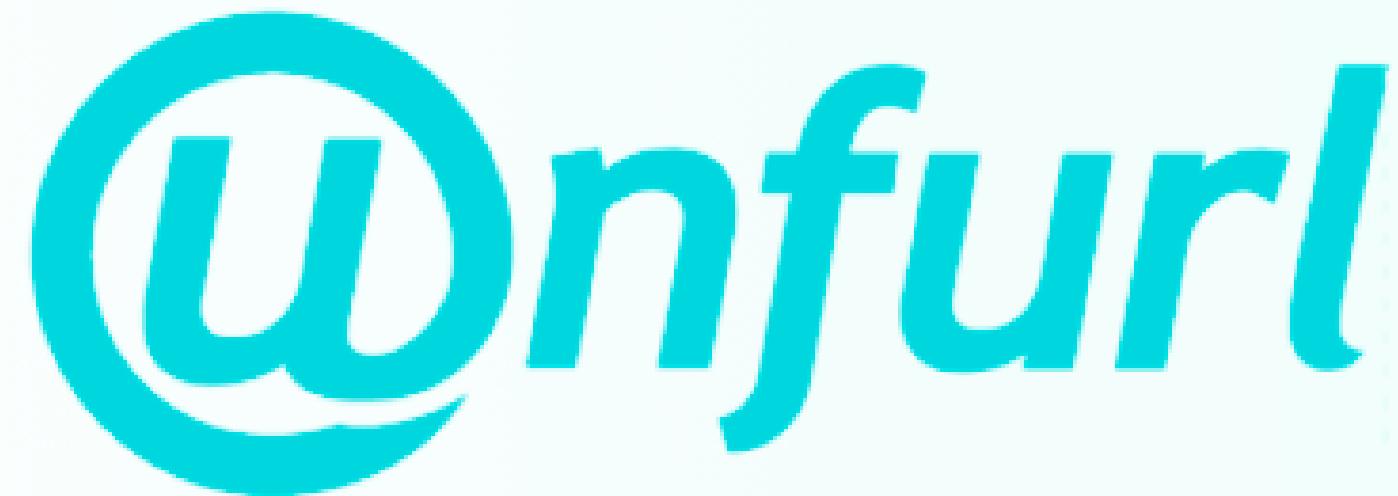


Atharva Patil

Lakshay Nasa

Zaur Hasanov

Presentation



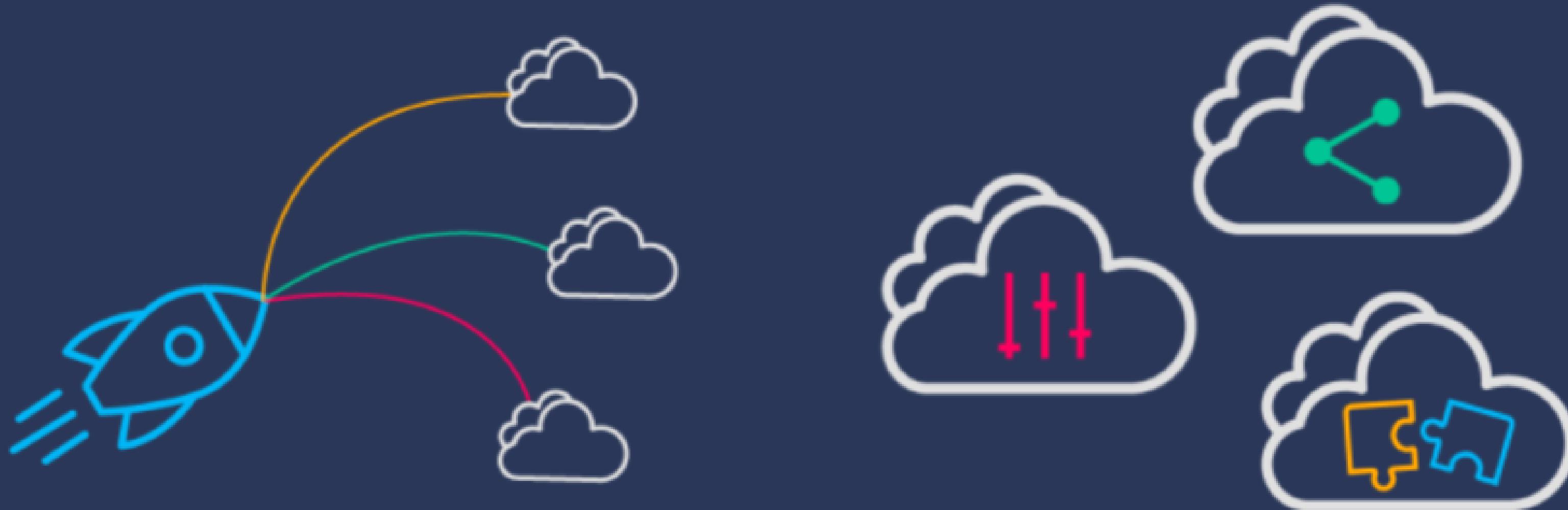
The logo for Unfurl consists of a teal-colored "u" character inside a circle, followed by the word "nfurl" in a lowercase, sans-serif font. A horizontal teal line extends from under the "u" across the width of the word "nfurl".

Who We Are

OneCommons is an
early-stage, mission-driven
startup dedicated to building
a **free and open cloud**.



MOTIVATION: A FREE AND OPEN CLOUD



Bring Open Source to the Cloud so that we can
collectively own and control the services our
lives depend on.

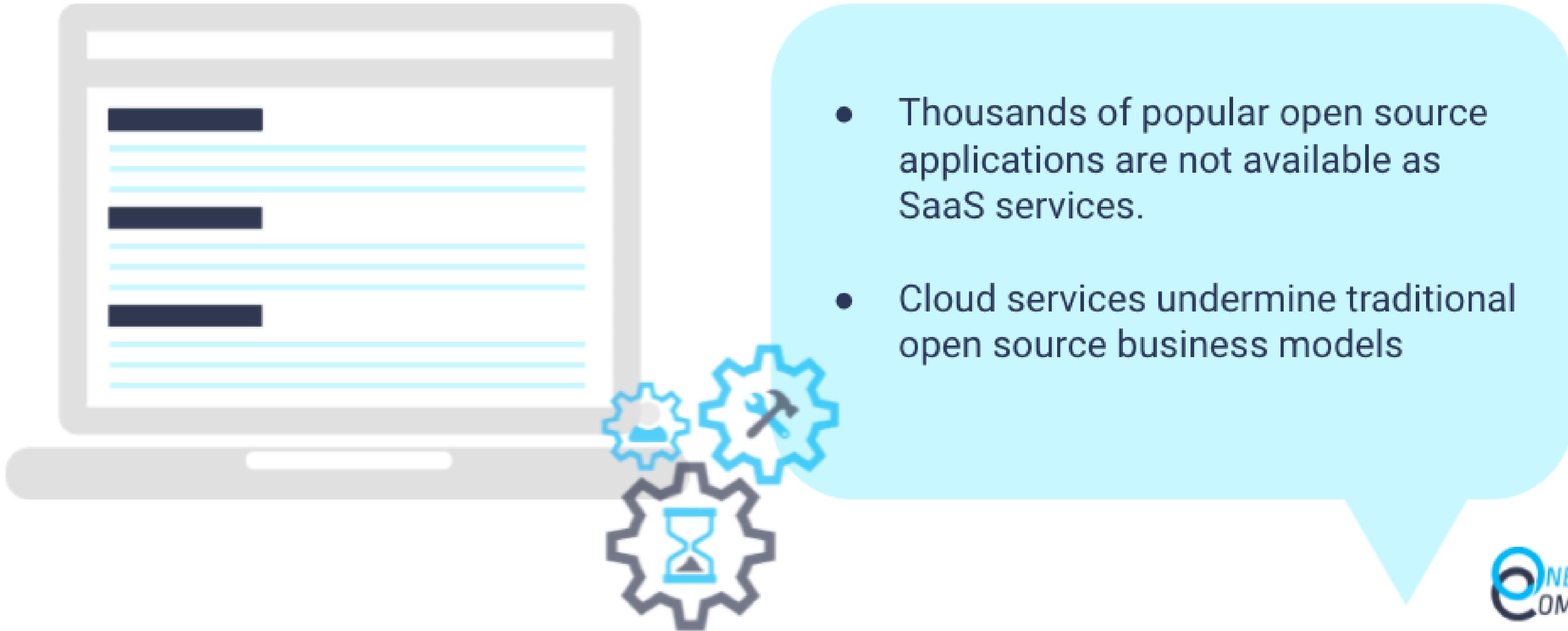
Everything is Moving to the Cloud...

- No one wants to run software, let alone servers
- Easy to get started
- Usage-based pricing, minimal upfront costs



Will Open Source Be Left Behind?

MOST POPULAR DEPLOYMENT TECHNIQUE: A LIST OF MANUAL STEPS



- Thousands of popular open source applications are not available as SaaS services.
- Cloud services undermine traditional open source business models

Developer Quandary

Develop w/Open Source

OR

Use Proprietary Services

PROS:

- ✓ Cost (long-term)
- ✓ Value-add
- ✓ Ownership

CONS:

- ✗ Hard
- ✗ Cost (short-term)



PROS:

- ✓ Easy
- ✓ Cost (short-term)
- ✓ Integration

CONS:

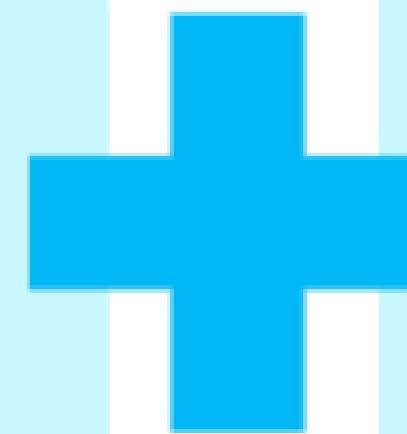
- ✗ Lock-in
- ✗ Cost (long-term)
- ✗ One-size-fits-all

Introducing: Open Cloud Services

open

adjective: COMPUTING

anyone is free to access, use, modify, and share — subject, at most, to measures that preserve **privacy** and **openness**.



cloud•service

noun: COMPUTING

an online service delivered on demand over the Internet without the user needing to use their own hardware.

Get the ease and simplicity of the cloud without giving up the freedom and control of open source.

HOW TO BUILD A FREE AND OPEN CLOUD



Step 1

1

Make it incredibly easy
to deploy and run open
source applications.



Step 2



2

Allow a small percentage of hosting costs to fund open source.

Step 3

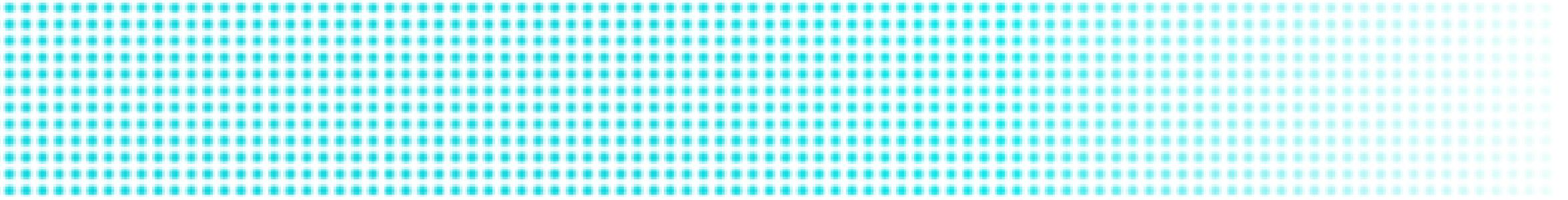
3

Sharable, forkable
open cloud services
+ Funding model
= New ecosystem.



Cloud Provider Independence





A TOOL FOR DEPLOYING AND
MANAGING ONLINE RESOURCES

How do you deploy?

- 1 Manually: Click around your cloud providers web console
- 2 Use a PaaS like Heroku or OpenShift
- 3 Ad-hoc / home-grown scripts and processes
- 4 DevOps tooling like Ansible or Terraform
- 5 Dunno / Someone else's problem

DevOps today

DevOps tools are designed for DevOps Engineers

No single tool provides a complete solution, coding is needed to integrate.

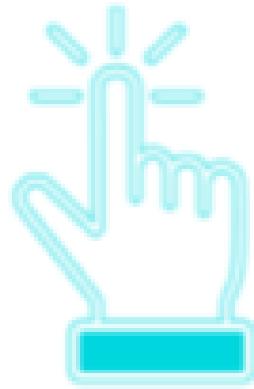
	Helm	Terraform/Pulumi	Ansible	Chef/Puppet /Salt	CI/CD/Gitops	Docker
Packaging	Kubernetes only	No	No	No	No	Minimal
Provisioning	No	Coding	Lo-code	No	No	No
Day two Operations	No	No	Lo-code	Coding	No	No
Change Management	Kubernetes only	Enterprise-only	No	No	Yes	No



THE BASICS:

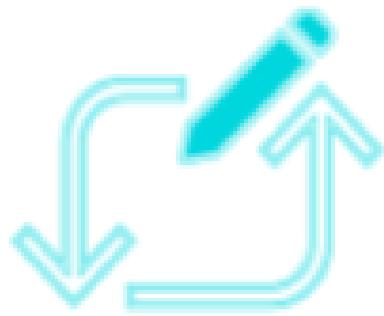
- Self-contained command-line tool that runs locally: **no server or agent software involved**
 - ✓ OPEN SOURCE
- All state is saved in user-editable **YAML** configuration files: **no database**
 - ✓ COMMUNITY-DRIVEN
- Manages **git** repositories for configuration and artifacts
 - ✓ DECENTRALIZED (VIA GIT)
 - ✓ STANDARDS- BASED (TOSCA)

Why Use Unfurl?



SIMPLICITY:

Hide complexity, no need
to be a DevOps guru



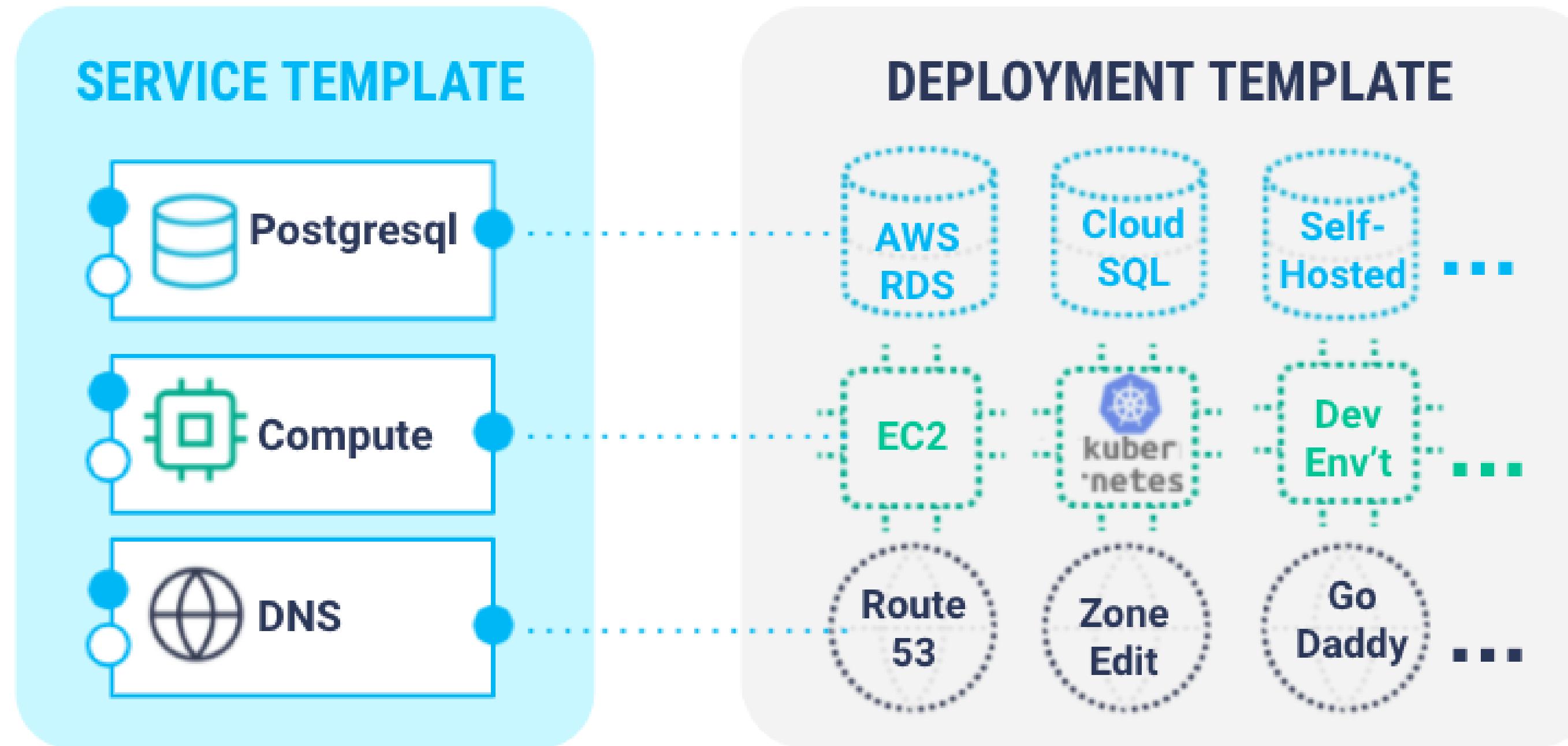
FLEXIBILITY:

Easily change infra as your
needs grow



NO LOCK-IN:

Open source
All data stored in Git
Cloud Independence



DEVELOPERS combine & connect
cloud-independent building blocks.

USERS deploy implementations that
match their infrastructure and configuration.

What is TOSCA?



TOPOLOGY AND ORCHESTRATION SPECIFICATION FOR CLOUD APPLICATIONS

A Standard
developed by OASIS

- Used manage applications in cloud and telecom network management
- **YAML vocabulary** that describes:
 - **The architecture of a cloud application or service**
 - The infrastructure the application requires in order to operate
 - The operations for deploying and managing the application

KEY CONCEPTS:

- **Nodes**
- **Operations**
- **Relationships**
- **Requirements**
- **Capabilities**
- **Configurations**
- **Policies**

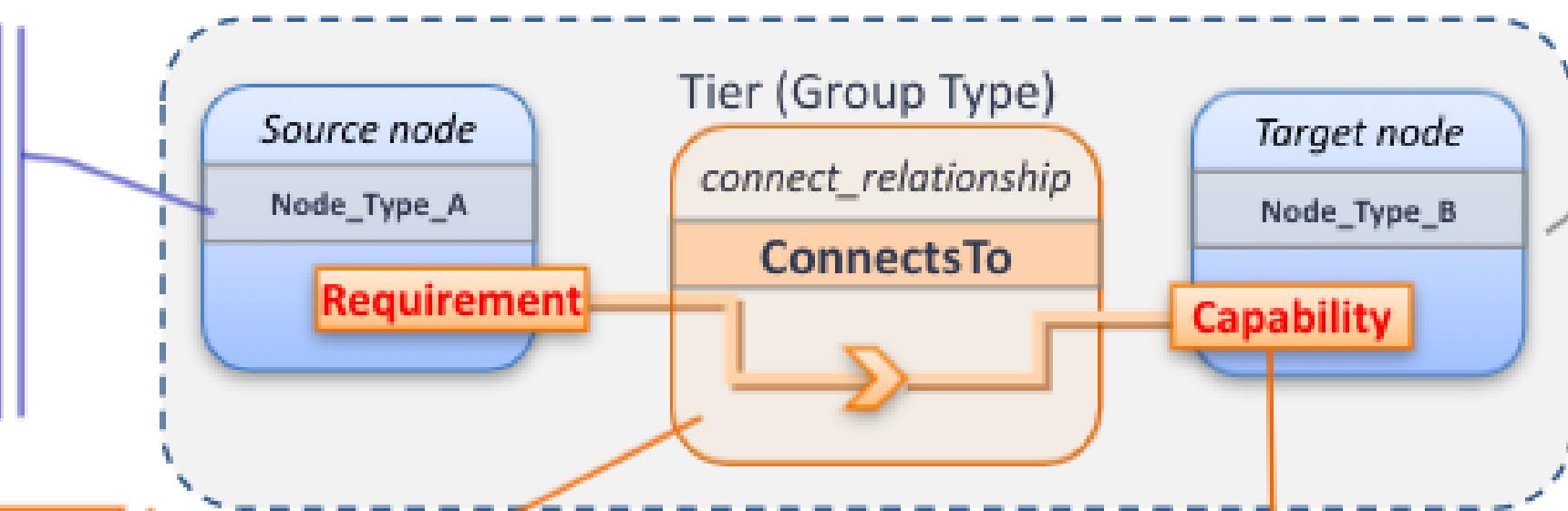
Topology – Define Topology with Nodes and Relationships

TOSCA DESCRIBES THE TOPOLOGY OF THE DEPLOYMENT OF CLOUD APPLICATIONS AND SERVICES

Node templates to describe components in the topology structure

Relationship templates to describe connections, dependencies, deployment ordering

Nodes - are the resources or components that will be materialized or consumed in the deployment topology



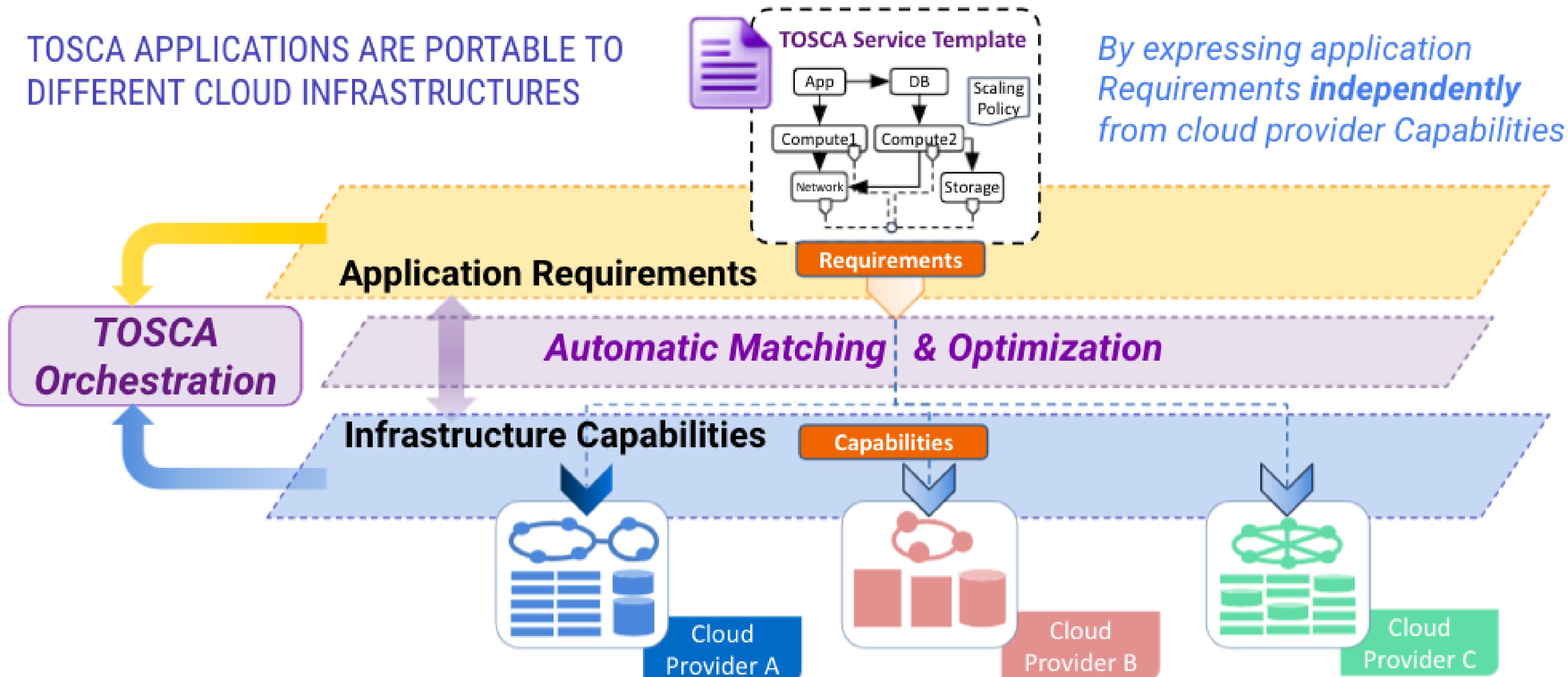
Groups
Create Logical, Management or Policy groups (1 or more nodes)

Relationships
express the dependencies between the nodes (not the traffic flow)

Requirement - Capability
Relationships can be customized to match specific source requirements to target capabilities

Portability – TOSCA Orchestrators find “Best Match” during deployment

TOSCA APPLICATIONS ARE PORTABLE TO DIFFERENT CLOUD INFRASTRUCTURES



Configurators and Installers

CONFIGURATORS are plugins that can be used as TOSCA operation implementations and do the real deployment work.

CURRENTLY SUPPORTED:

- Ansible
- Terraform
- Shell



Terraform



kubernetes

INSTALLERS are TOSCA type definitions with operations that know how to deploy and discover their native resources.



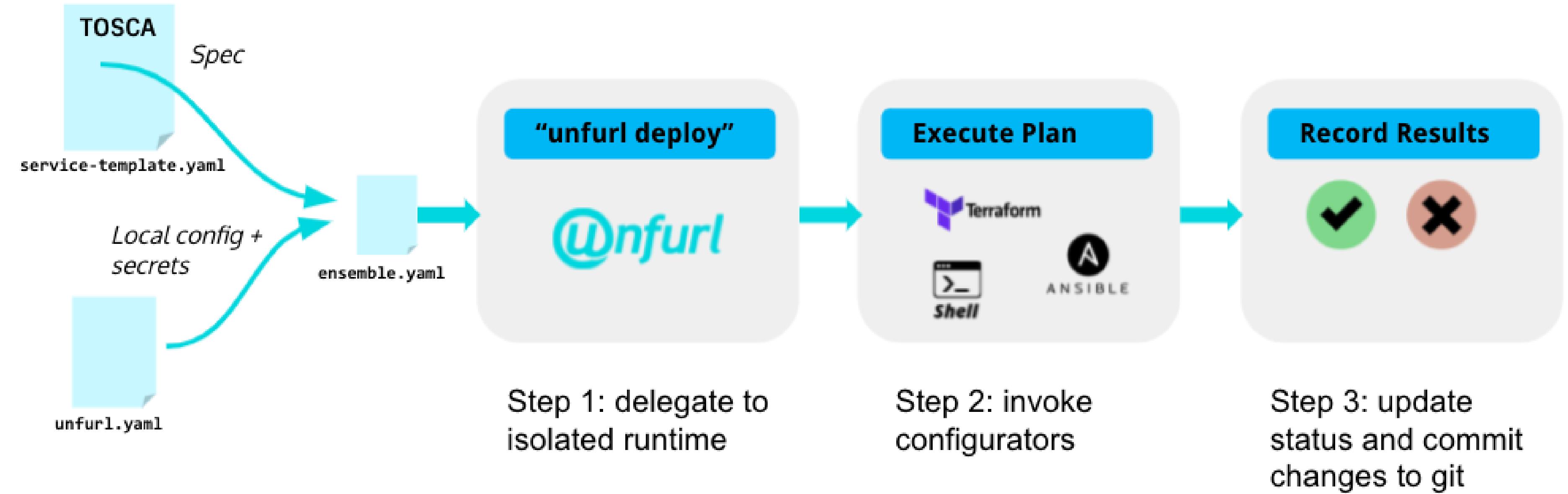
ANSIBLE



CURRENTLY SUPPORTED:

- Kubernetes:
 - Resources
 - Helm Charts
- Docker Engine
- Supervisor Process Manager

Unfurl Processing Model



Anatomy of ensemble.yaml

CONTEXT

Context: Describes the environment that the Unfurl deployment process runs in, e.g.

- Tools and repository locations
- Credentials

SPEC

Spec: TOSCA service template and explicitly declared instances

STATUS

Status: Deployed instances and their current status

LOCK

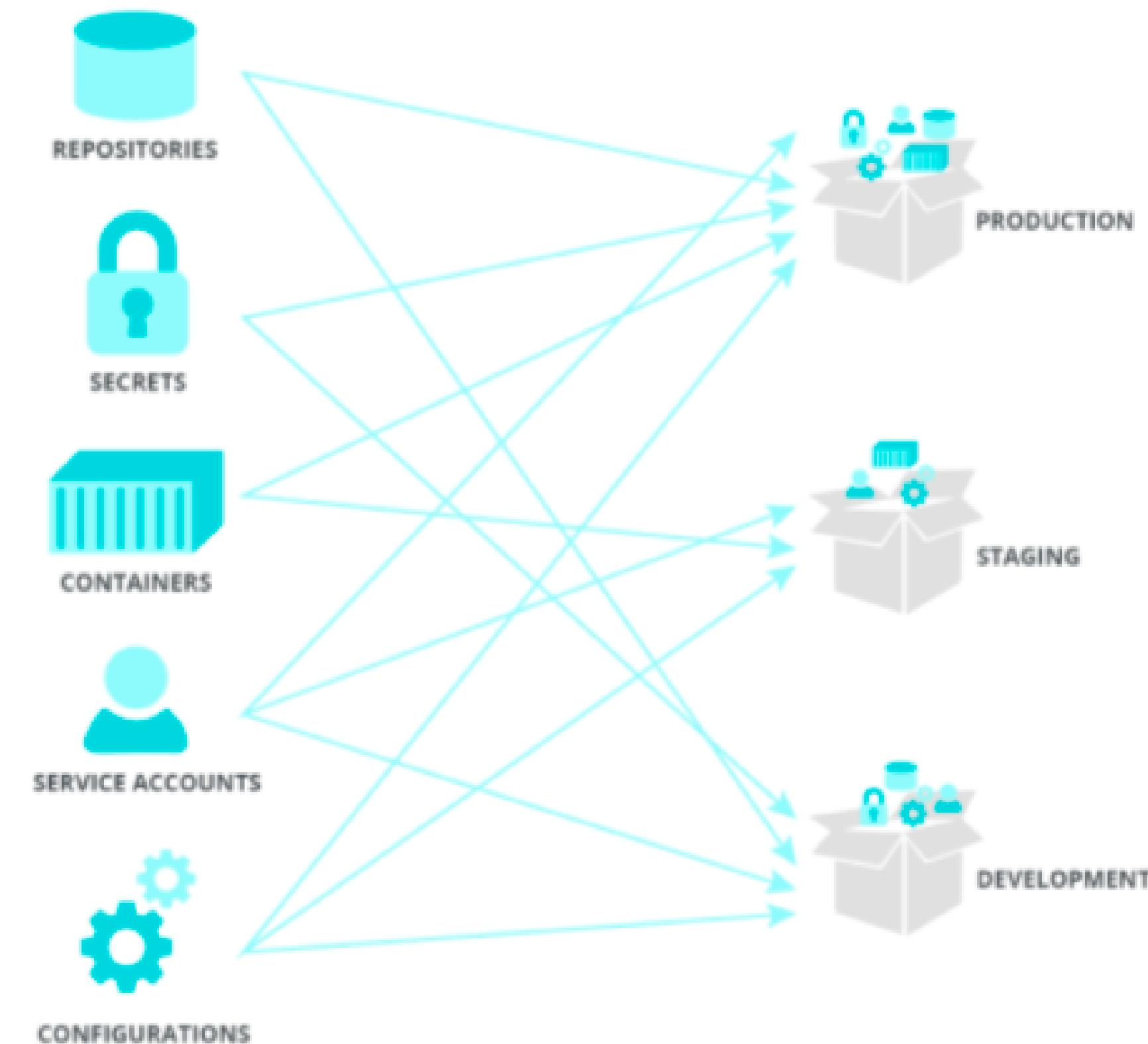
Lock: Record precise digests and versions of the artifacts and repositories used during deployment.

Context : Organize Your Deployment Environments

CONTEXT

```
context:  
  locals:  
    schema:  
      prop2:  
        type: number
```

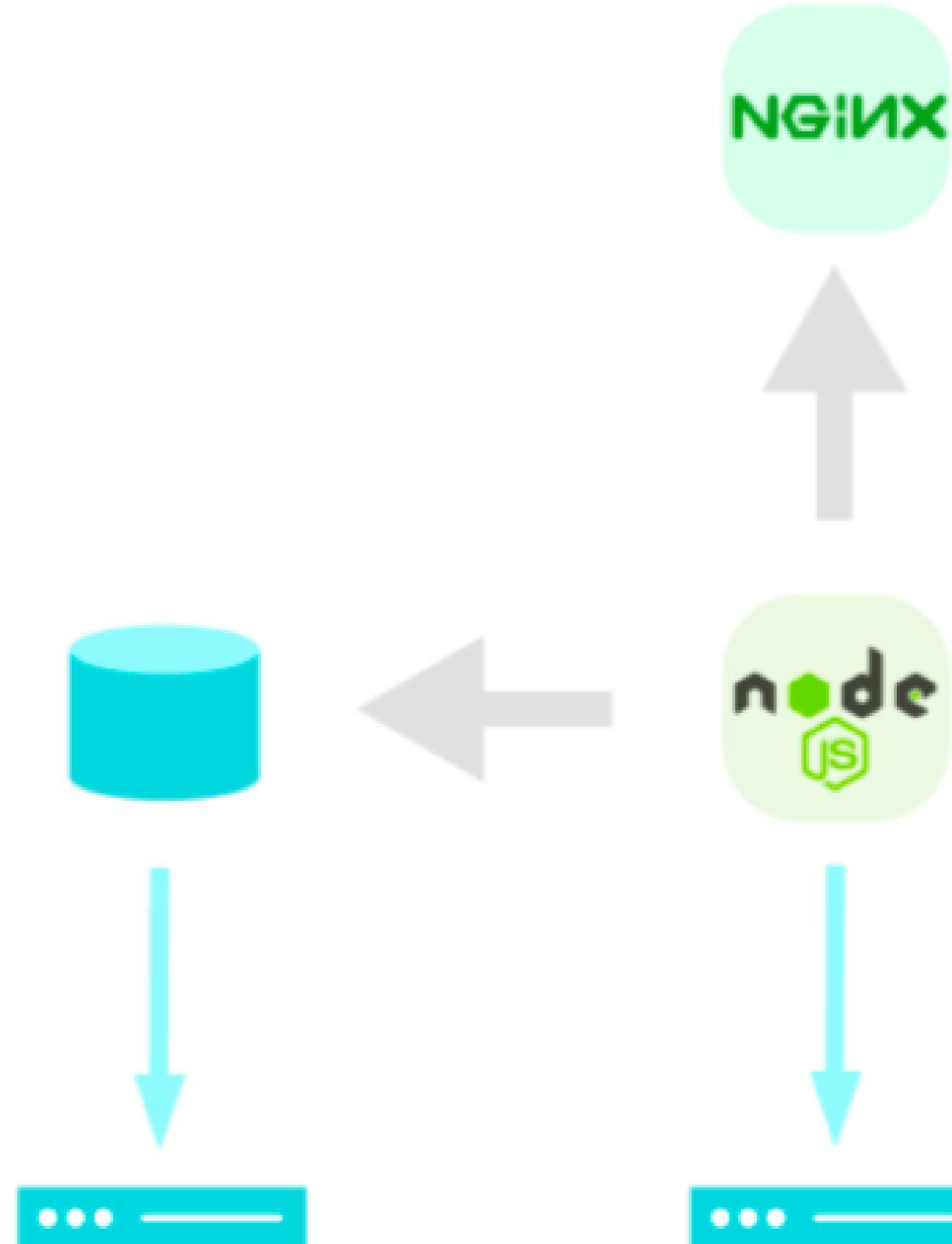
Unfurl allows you to collect your configuration information and organize them into contexts.



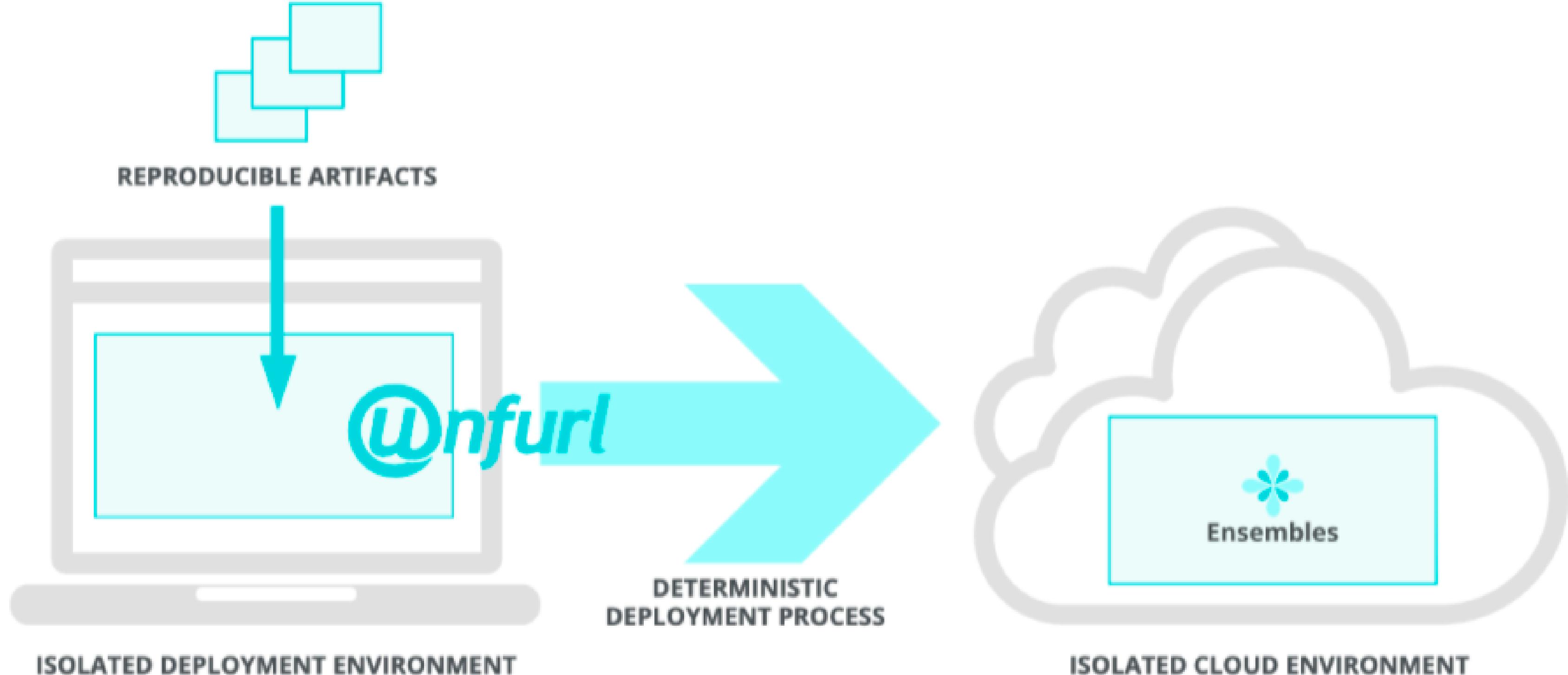
Spec : Model Your Cloud Infrastructure

SPEC

```
spec:  
  service_template:  
  topology_template:  
    inputs:  
      rootdomain:  
        type: string  
        default: unfurl.run
```



Status and Lock: Reproducible Ensembles



Ensemble.yaml: status and lock sections

STATUS

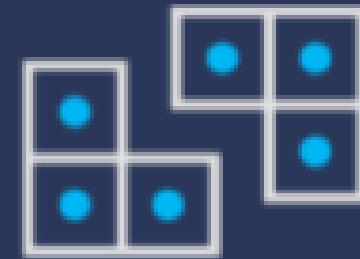
```
status:  
  inputs:  
    rootdomain: unfurl.run  
  outputs: {}  
  instances:  
    staging_site:  
      template: staging_site  
      readyState:  
        local: ok  
        state: created
```

LOCK

```
lock:  
  runtime:  
    unfurl:  
      version: 0.1.1.dev20  
      digest: 0.1.0-29-gf41588a-dirty  
    toolVersions:  
      terraform:  
        - 0.12.29  
  repositories:  
  - name: self  
    url: git-local://5fd0f9...d0f8e80429:self/.  
    revision: 6c1706140dce9338182ac46f37d04b5397ce0723-dirty
```

SaaS Platform Sneak Peak

STEP 1: PACKAGE



The screenshot shows the One Commons platform interface. On the left, a node template named "Compute" is displayed, featuring a cloud icon and a green letter "a". On the right, a modal window titled "Add Node Template" lists various node types with their descriptions and versions:

Type	Description	Version
Compute	Cloud-based compute resources	1.0.0-ALIEN20
Port	Cloud-based port resources	1.0.0-ALIEN20
Bash	clic pass script	1.0.1
Private	clic access	1.0.1
Public	clic access	1.0.1
HelloWorld	clic samples python	1.0.0
Nodejs	nodejs	1.0.1
SSHBastionHost	SSH Bastion Host	1.0.0

Developers package their app using simple,
cloud independent “lego bricks”

STEP 2: PUBLISH



vue.config.js	Enable tslint in graphql files	25 days ago
yarn.lock	Add Vue router to package.json	21 days ago

README.md

Notes

See Apollo boilerplate added with `vue add apollo ...` see <https://apollo.vuejs.org/>

Need to run `yarn run apollo:start` first to start the apollo graphql server (runs on port 4000).

see <https://gitlab-org.gitlab.io/gitlab-ui/> and <https://gitlab-org.gitlab.io/gitlab-svgs/>

How to Deploy

Deploy with OneCommons 164

Developers share our deployment buttons,
e.g. on their Github readme

STEP 3: DEPLOY



The screenshot shows the OneCommons platform's deployment interface. On the left, a sidebar for 'Demo Project' has 'Project overview' selected. The main area shows a project named 'GCP-test'. The 'Word Press' template is selected. The interface includes sections for 'Inputs' (with two input fields labeled 'Input'), 'Requirements' (listing 'Compute', 'Database - MySQL', 'DNS', and 'Mail Server' with descriptions and 'Connect' and 'Create' buttons), and deployment buttons at the bottom for 'Merge Request', 'Save template', and 'Deploy'.

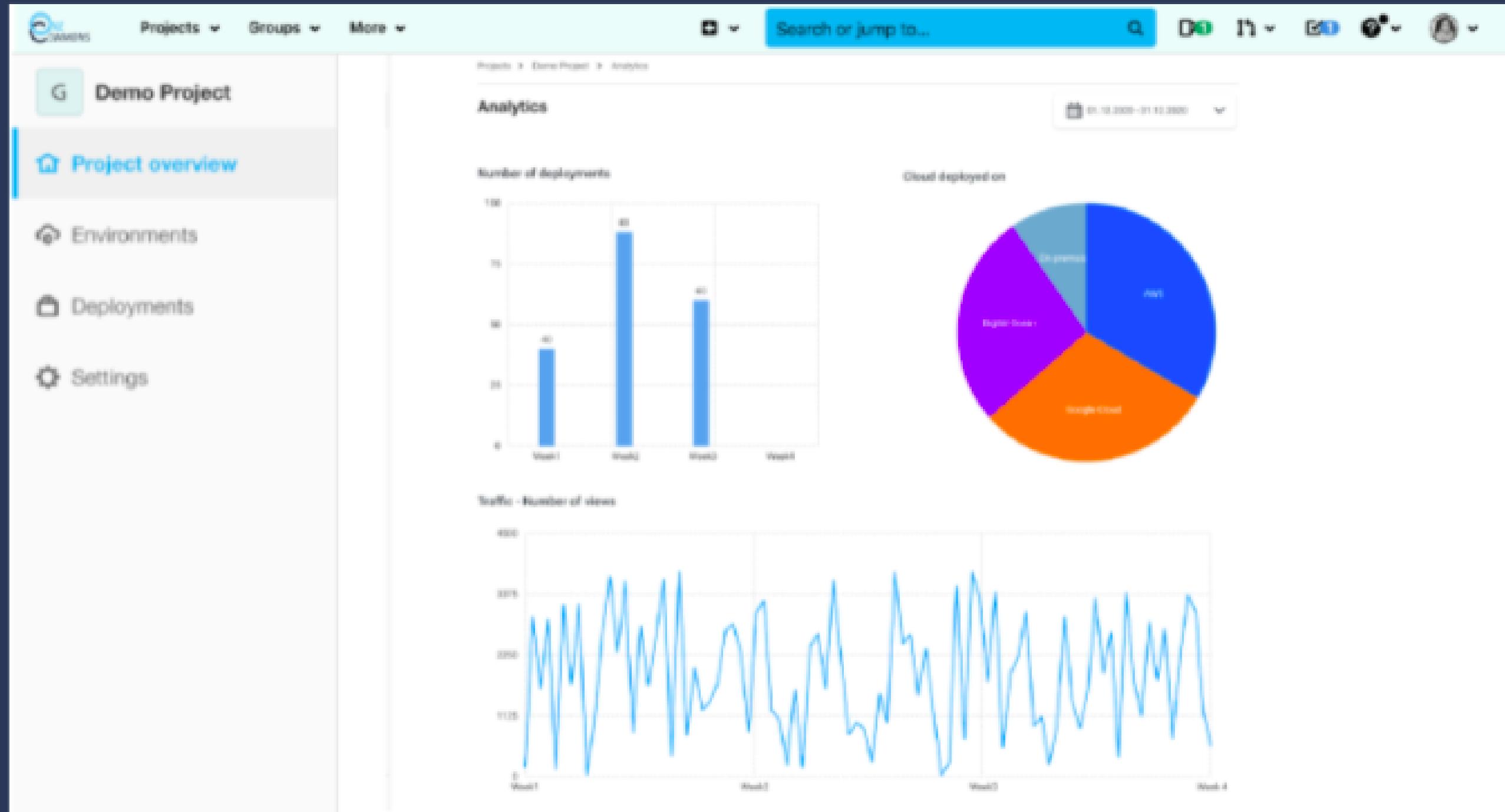
Their users click through to OneCommons and deploy to their own cloud accounts or machines.

Simple OAuth flow for easy deployment

The diagram illustrates a two-step OAuth process:

- Step 1: Google Sign-in**
A screenshot of a Google sign-in dialog titled "Sign in with Google". It shows a "Confirm your choices" section where the user "adam@onecommons.org" is granting permission to "onecommons.org" to "See, edit, configure, and delete your Google Cloud Platform data". A large blue arrow points from this step to the next.
- Step 2: Google Cloud Platform Project Configuration**
A screenshot of the "onecommons" Google Cloud Platform project settings. The "Environments" page is shown, with "Demo > Apostrophe Demo > Environments". It displays the selected "Google Cloud Platform project" as "public" and the "Zone" as "us-central1-a". A "Save" button is visible at the bottom.

STEP 4: REVEAL



We manage deployment and provide hard-to-obtain user analytics to the developer.

Cloud Funding: The Billion \$ Question

How much of your hosting bill would you be willing to contribute to fund the open source projects your application depends on?

0%

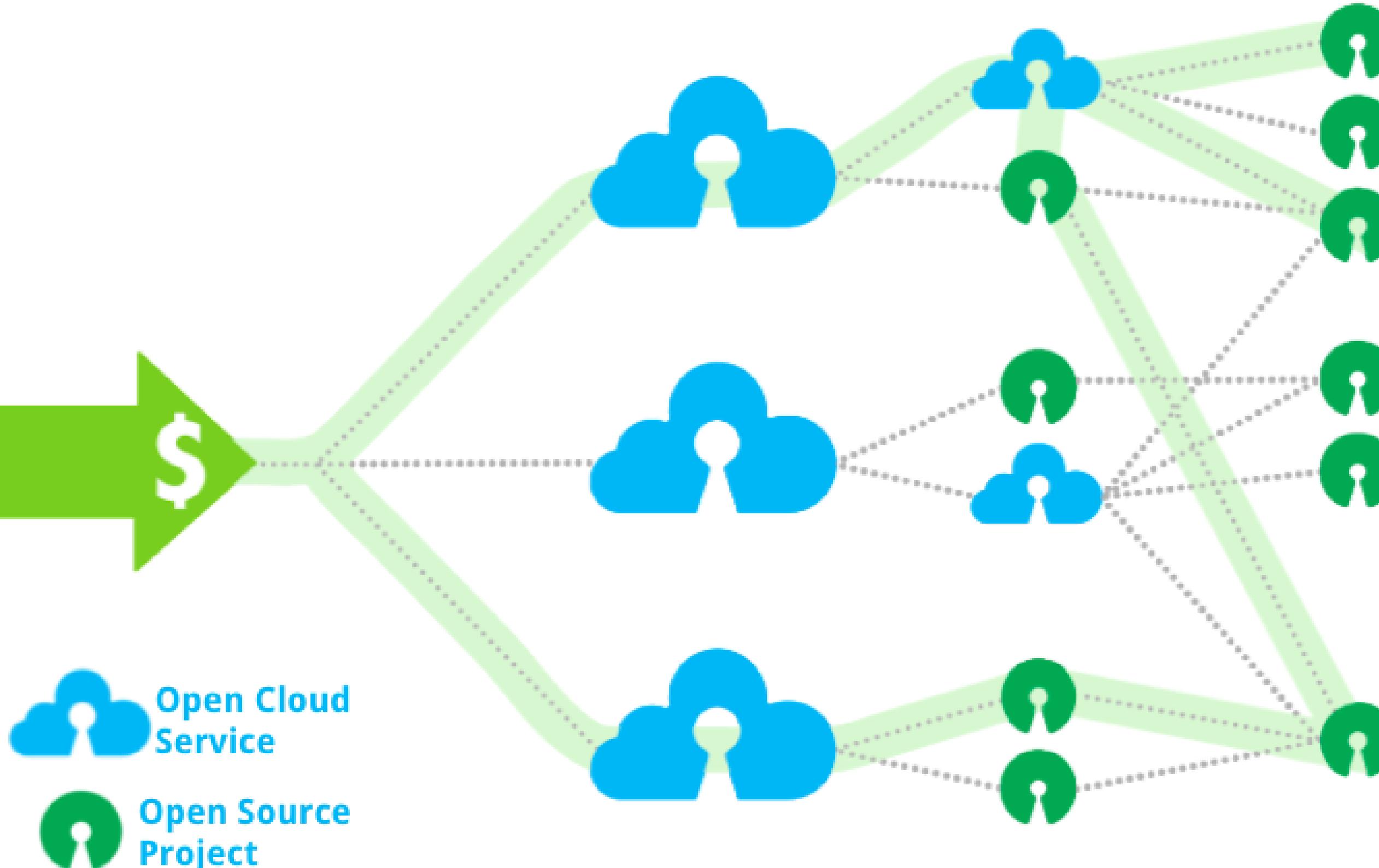
3%

5%

10%

> 10%

Cloud Funding



- + Managed Service Fees
- + Consulting & Outsourcing Opportunities

Cloud Funding



Developers register projects, ensembles and **set fundraising goals**.



Users deploy ensembles on their cloud infrastructure with a “gratuity” added.



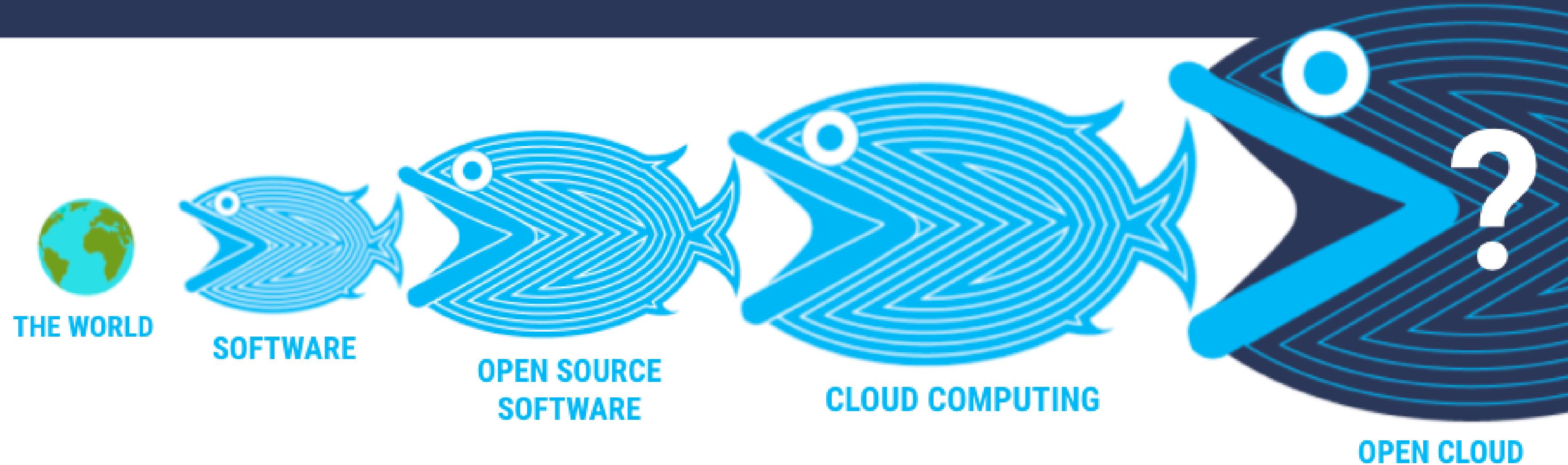
Successful projects **pledge to give 50% back** to one or more of the projects it depends on.

What Becomes Possible

Three Levels of Openness



What if?



Cloud Services

- Convenience
- Minimal upfront costs
- Usage-based pricing
- Data Network effects

Open Source

- Free and Open
- Transparency
- Community knowledge
- Code Network effects

THANK
YOU!



adam@onecommons.org

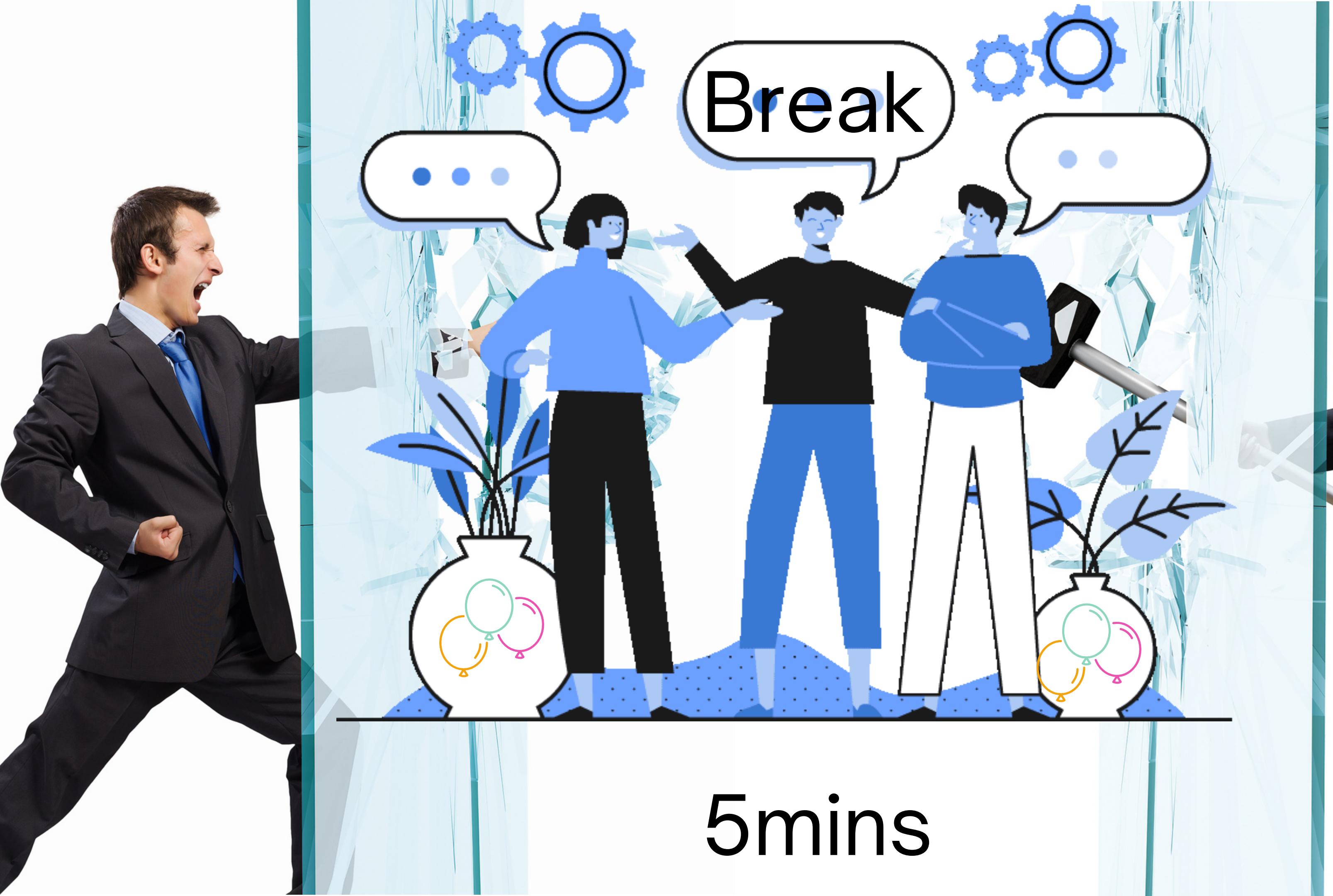
www.unfurl.run

github.com/onecommons/unfurl

www.onecommons.org

Q&A | Discussion

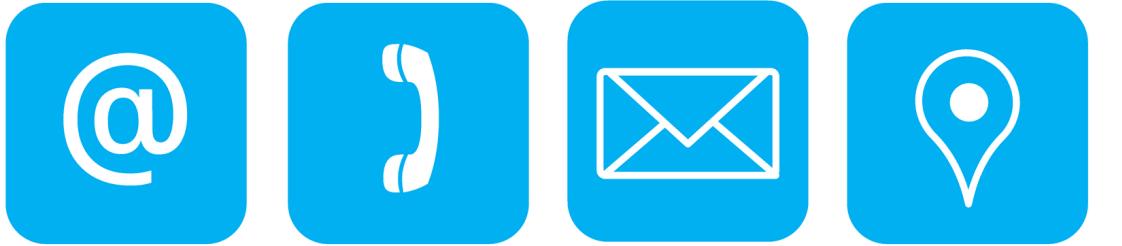




5mins

Wrap Up

HOWTO



Remember!

- Re-Search
- Be short and clear
- Re-mind
- Q&A over Slack

Linkedin: @adamsouzis

Email: adam@onecommons.org



Adam Souzis



Join Us!

HTTPS://GOUPAZ.COM

HTTPS://METABOB.COM

- 1** Community Managers
- 2** Tech Writers
- 3** In/Out Ambassadors
- 4** Marketing Creators & Editors
- 5** Course Creators
- 6** Project Creators

Thank You

Culture

#egoless #collaborative #competent #decentralized #scalable #fun

Open source

#creator #contributor

Diversity

#age #gender #location #economics #religion #politicalview

How can we do
better?