



Fosdem Feb 6 2022





A TOOL FOR MANAGING YOUR DEVOPS INFRASTRUCTURE



What is **Unfurl**?

A command-line tool that:

1. Deploys applications using a high-level declarative vocabulary so you can describe your infrastructure independent of cloud provider details.

2. Manages your deployments (configuration, tools, history, state) in Git.





The Basics

THE BASICS:

- Self-contained command-line tool that runs locally: no server or agent software involved
- All state is saved in user-editable
 YAML configuration files: no database
- Manages **git** repositories for configuration and artifacts

- OPEN SOURCE
- COMMUNITY-DRIVEN
- DECENTRALIZED (VIA GIT)
- STANDARDS- BASED (TOSCA)





What it does

Unfurl tracks your:

- Configuration
- Secrets
- Code dependencies
- Software version
- Deployment history

... all in git!





Works with your tools

Includes out-of-the box support for

- Terraform
- Ansible
- Shell commands
- Helm



- OctoDNS
- Kubernetes
- Docker
- Supervisor



Motivation: Imagine if you could freely reuse, modify, and fork live services just like open source code.





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





Why Use Unfurl?

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

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





How it works

MMONS









Unfurl Processing Model







Anatomy of ensemble.yaml

CONTR	
SONTEXT	Environments: 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.



Unfurl Organize Your Deployment Environments



Unfurl allows you to collect your configuration information and organize them by environment.





Openational Spec : Model Your Cloud Infrastructure













Combine & connect cloud-independent building blocks.

Deploy implementations that match your infrastructure and configuration.





TASCA

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



Portability – TOSCA Orchestrators find "Best Match" during deployment





status and lock sections







Reproducible Ensembles



COMMONS



Vision: A Free and Open Cloud





THANK YOU!



adam@onecommons.org

www.unfurl.run github.com/onecommons/unfurl www.onecommons.org