

PRODUCTION



Operating Distributed Systems in Production

Juju Does Everything.

Here Is What That Is Like.

By Madhur Jain

Can you spot Juju in this image?

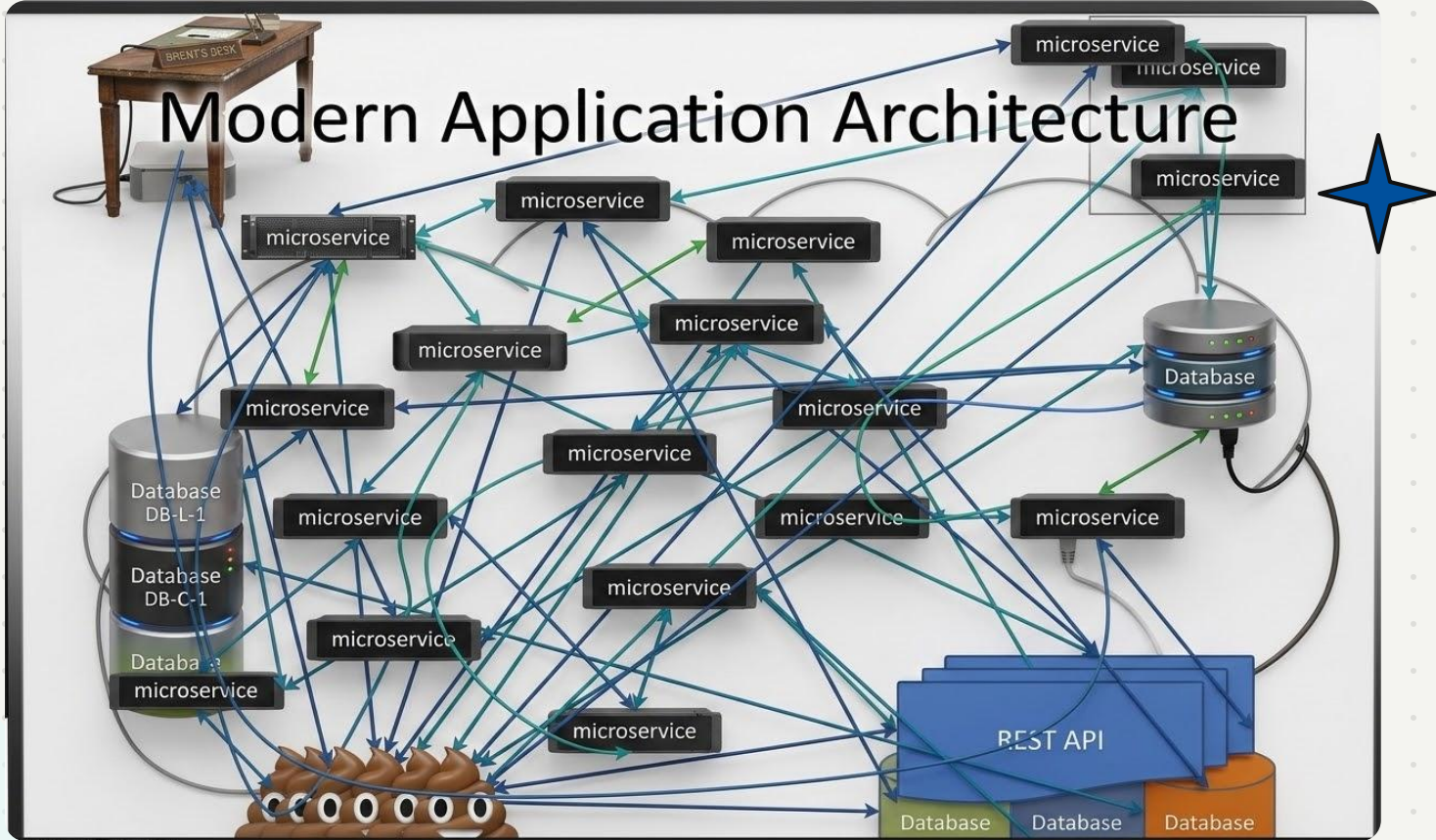




What is Juju?

Juju is an open source orchestration engine for software operators that enables the **deployment, integration, and lifecycle** management of applications in the cloud using special software operators called **'charms'**.

Why do we even need a Maestro?



Why do we even need a Maestro?

DAY 0: PLAN

Architecture Mapping:

Define service boundaries and communication flows.

Capacity Planning:

Estimate CPU, RAM, storage, and network bandwidth.

Tech Stack Selection:

Choose databases, runtimes, and orchestration tooling.

DAY 1: DEPLOY

Infra Provisioning:

Allocate VPCs, subnets, and Kubernetes clusters.

Software Bootstrapping:

Fetch container images and install dependencies.

Initial Configuration:

Set env variables, inject secrets, and initialize DBs.

DAY 2: OPERATE

Scaling & Healing:

Add/remove replicas and auto-replace failed nodes.

Upgrades & Patching:

Apply OS and application updates without downtime.

Backups & Security:

Automate snapshots and rotate secrets/certificates.

Kubernetes: Orchestration vs. Lifecycle

WHAT K8S IS



Container Management



Automated Orchestration

WHAT K8S IS NOT

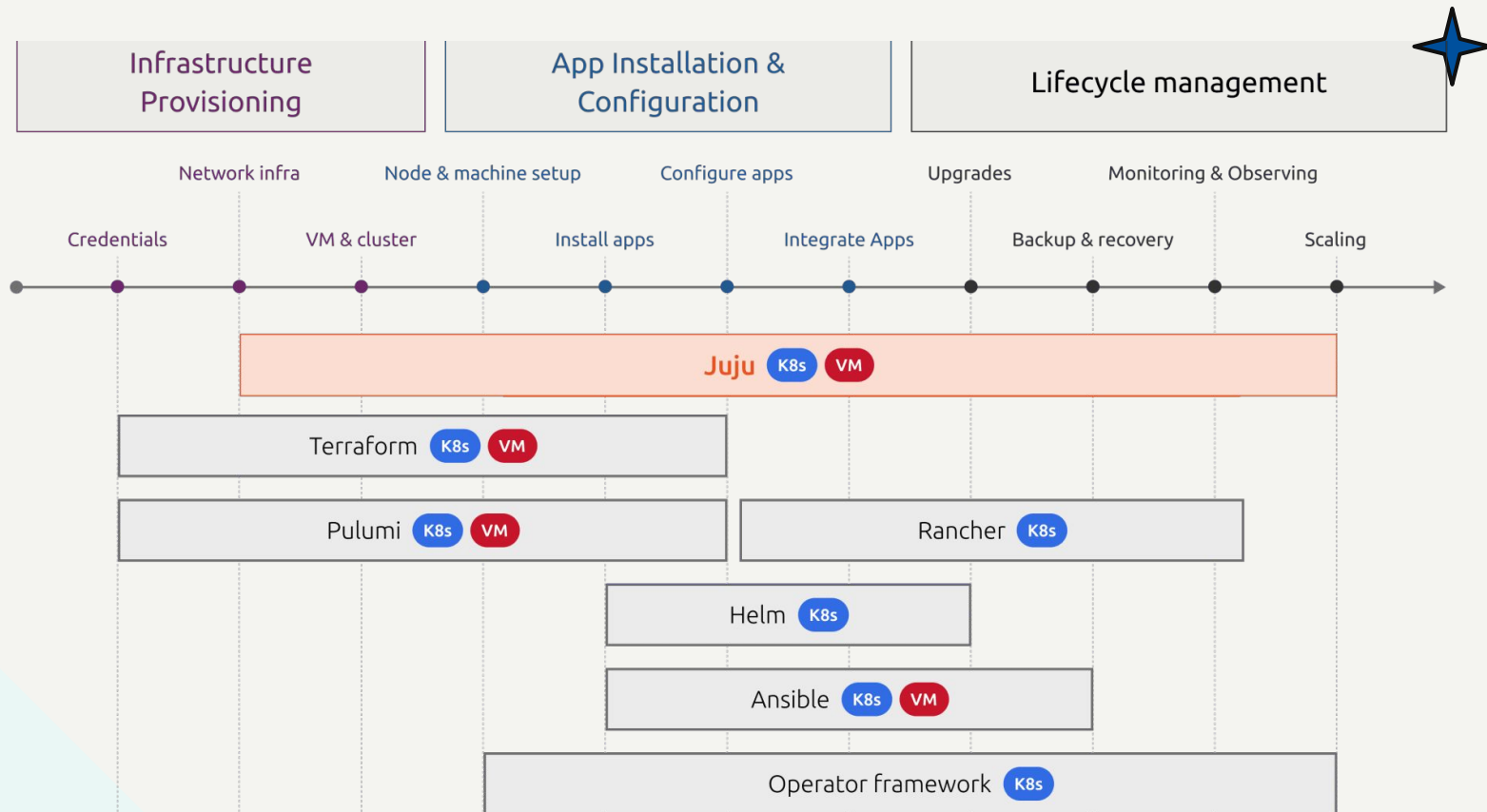


No Compute Setup



Manual Day-2 Operations

Juju: One tool to rule them All



App-Centric Modeling



Stop Managing Machines.

You declare the goal:

"I want a high-availability PostgreSQL database."

Why it shines:

It automatically figures out the underlying compute, subnets, and firewalls to make your topology a reality.

Charms: Encoding Day-2 Intelligence



Charms are literal application operators.

They literally have the entire logic to perform Day-0,1..N operations for application

Seamless Integration (The Relationship Graph)



One Command

"juju integrate"

Why it shines:

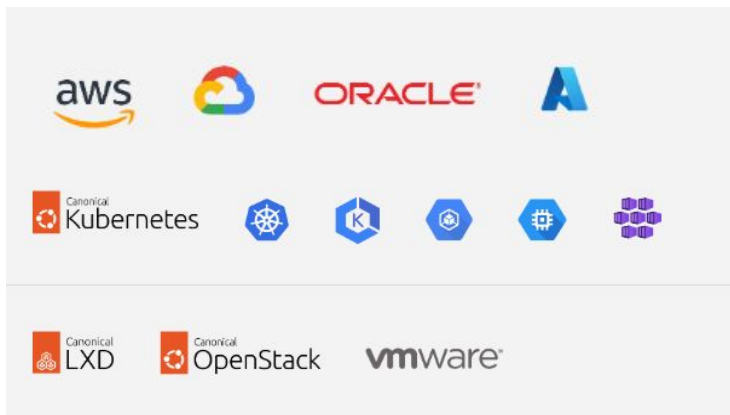
- No overhead to handle IP addresses, exchanging credentials, or updating config files.

Cross-Model and Multi-Cloud Operations



Single codebase across **clouds,**
VMs and Bare Metal !

And you can have
cross-**anything** connections!



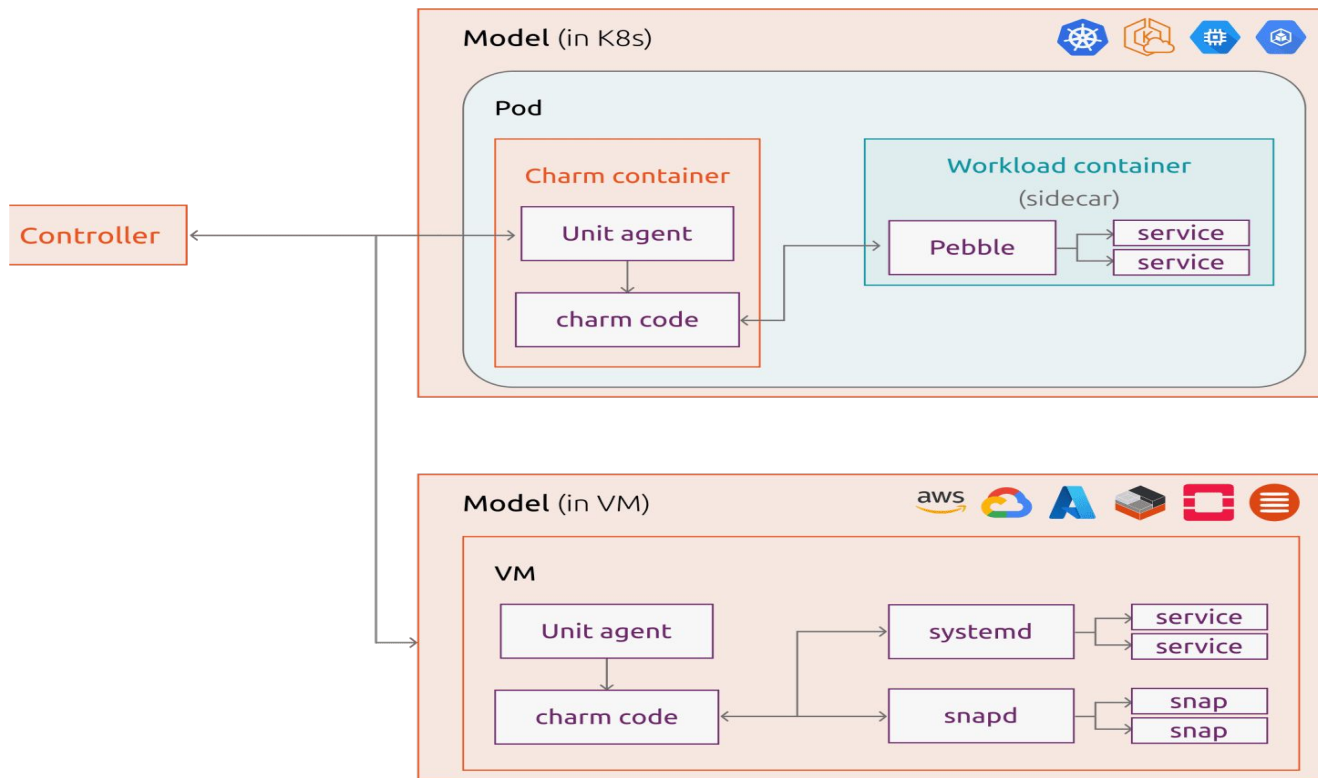
Universal Control



Using just single juju interface you can control everything across clouds and machines

`"juju scale-application"`

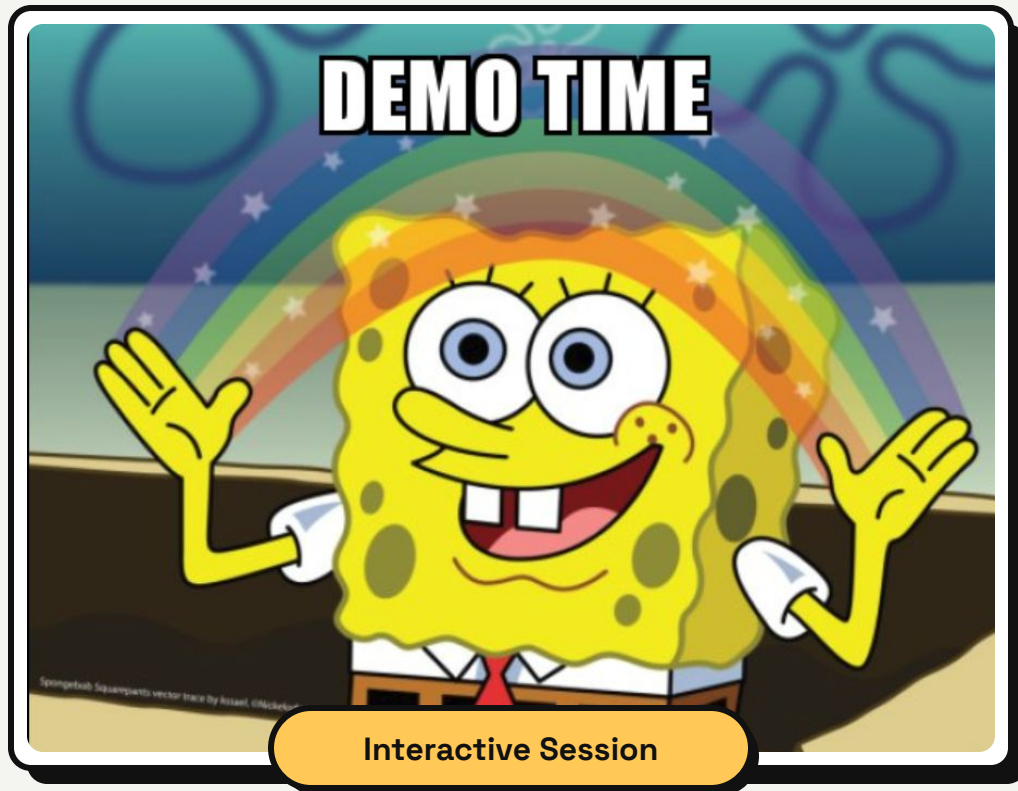
JUJU ARCHITECTURE



Kubernetes to Juju Terminology Map

How familiar cloud-native abstractions translate directly into Juju's operational model.

KUBERNETES CONCEPT		JUJU EQUIVALENT
Deployment	→	Application Defines the logical service and layout configuration.
Pod	→	Unit An individual active instance of an application.
Namespace	→	Model Houses application groupings and relations.
YAML templates	→	Charm Active Operational Logic Contains code to manage the app lifecycle (Day 0-2 ops), unlike static Helm YAML.
kubelet	→	Unit Agent Internal daemon coordinating local operations.
<i>None (No direct equivalent)</i>	→	Relation Juju-Only Differentiator Enables automatic cross-application integration.





THE JUJU DEBUGGING TOOLKIT

```
● ● ● juju-operator-diagnostics --interactive
```

1. Units stuck in maintenance

```
$ juju debug-log --include unit:app/0 --replay
```

2. Relation data is stale

```
$ juju run app/0 -- relation-get -r <rel-id> - app
```

3. Silently failing hooks

```
$ juju debug-hooks app/0
```

4. Detect configuration drift

```
$ juju export-bundle > current.yaml && diff bundle.yaml current.yaml
```

5. Pod fails to start (bypass Juju directly)

```
$ kubectl logs -n <model> <pod>
```


Where to look further?

Explore official documentation, community hubs, custom operators, and open-source codebases.

DOCUMENTATION

Juju CLI Docs

Comprehensive commands, configuration, and reference manuals.
canonical.com

CHARMHUB

Charm Store

Discover, deploy, and share production-ready charmed operators.
charmhub.io

DEVELOPMENT

Craft Charms

Build your own charms and orchestrate bespoke system logic.
charmcraft

OPEN SOURCE

GitHub Repo

Explore, contribute, and fork Juju's complete open-source code.
github.com/juju

Reach Me Out

Madhur Jain

Email: 21112002mj@gmail.com

Github: [immortalcodes](https://github.com/immortalcodes)

Linkedin: linkedin.com/in/madhur-jain-37a243218/



Any Questions?