

# Amphitheatre for Solana

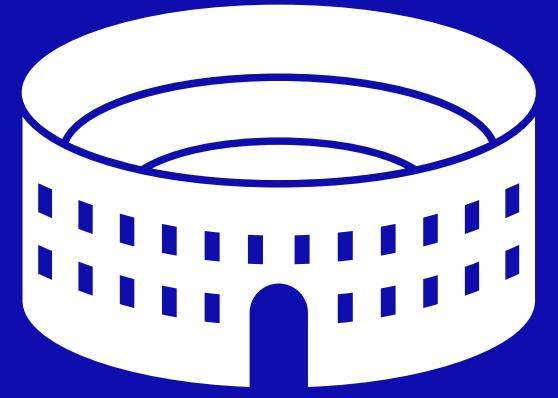
For Solana, Amphitheatre offers robust support with smart contracts in various programming languages (Native, Anchor, Seahorse, C and C++, Solang), along with the option for a local testnet.

---

**SOLANA HYPERDRIVE GLOBAL ONLINE HACKATHON**

**Eguo Wang, The Amphitheatre Author**

**October 13, 2023**

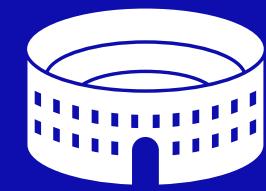


# Amphitheatre

Instantly spin up fresh, automated dev environments in the cloud. It provides on-demand, pre-configured tools, libraries, and dependencies, ensuring you can start coding immediately.

<https://amphitheatre.app>

- ✨ Zero local setup required.
- 🚀 Unlimited isolation environments.
- 📦 Supports apps with multiple components.
- ⚡ Instant code updates, no builds.
- 📸 Collaborate quickly, shared dev environment.
- ☁️ On-demand Amphitheatre Cloud, ready-to-use.
- 🧩 Automate end-to-end software integration.



# Amphitheatre

Instantly spin up fresh, automated dev environments in the cloud and start developing within seconds.

## Bring Your Code



Supports many of the many popular languages, frameworks, SDKs and runtimes.

amp init

## Streamline Software Integration



Works with developer tools & any dev resource, Adds internal & external dependencies.

amp build

amp deploy

## Deploy & Run



Kubernetes Cluster with Containers

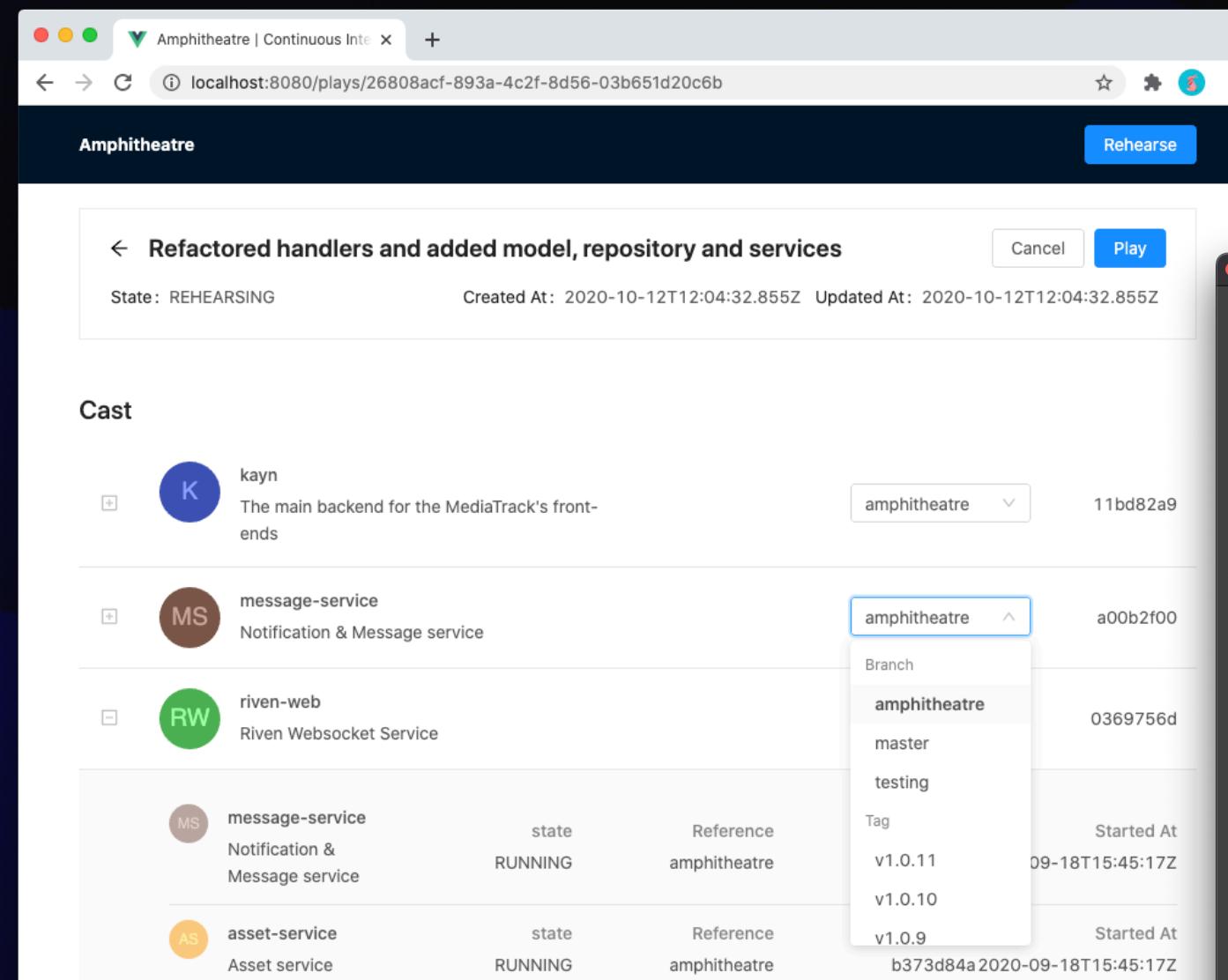
Amphitheatre Cloud

amp cloud deploy

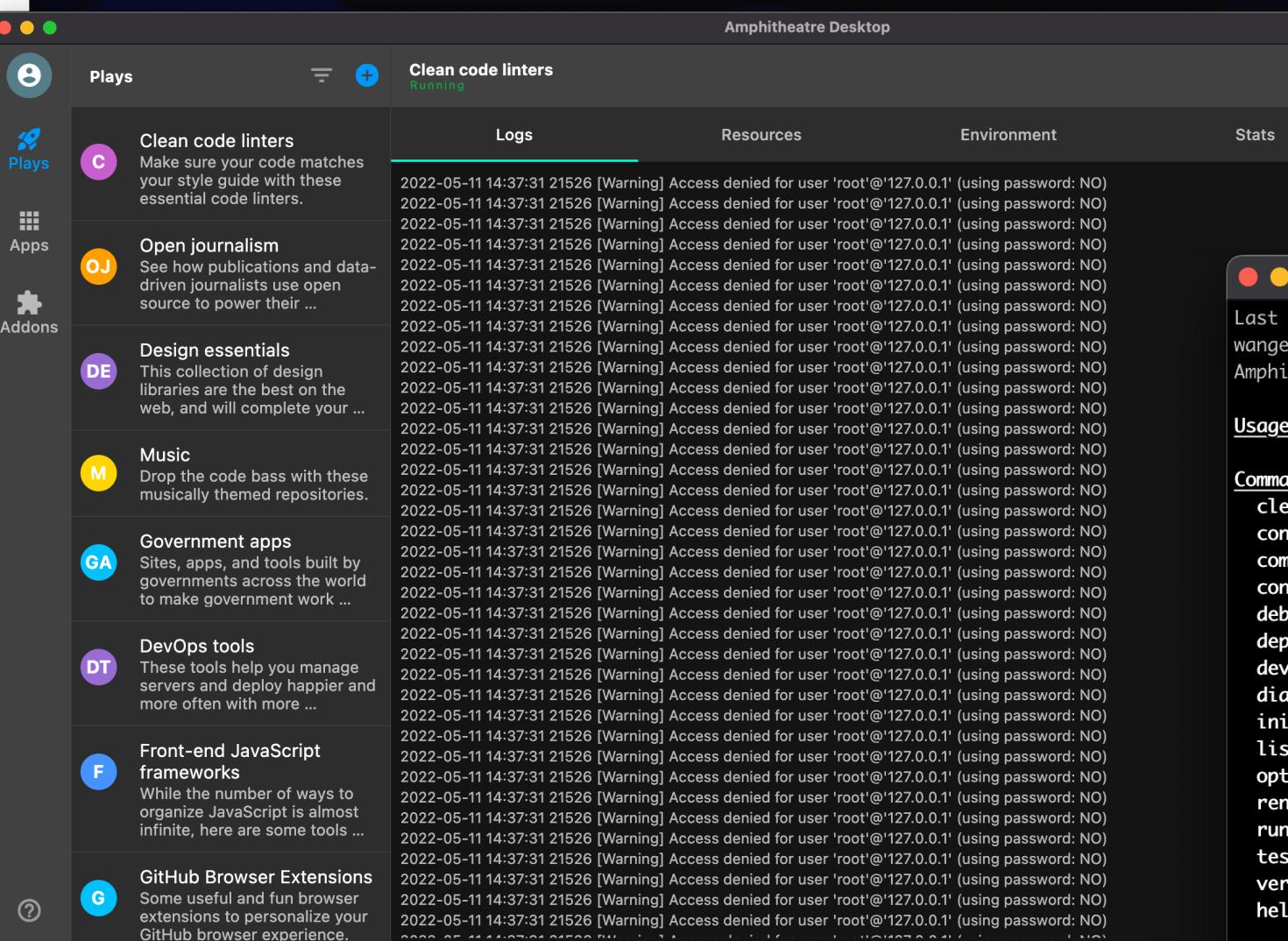
<http://localhost:3000>

# Screenshots

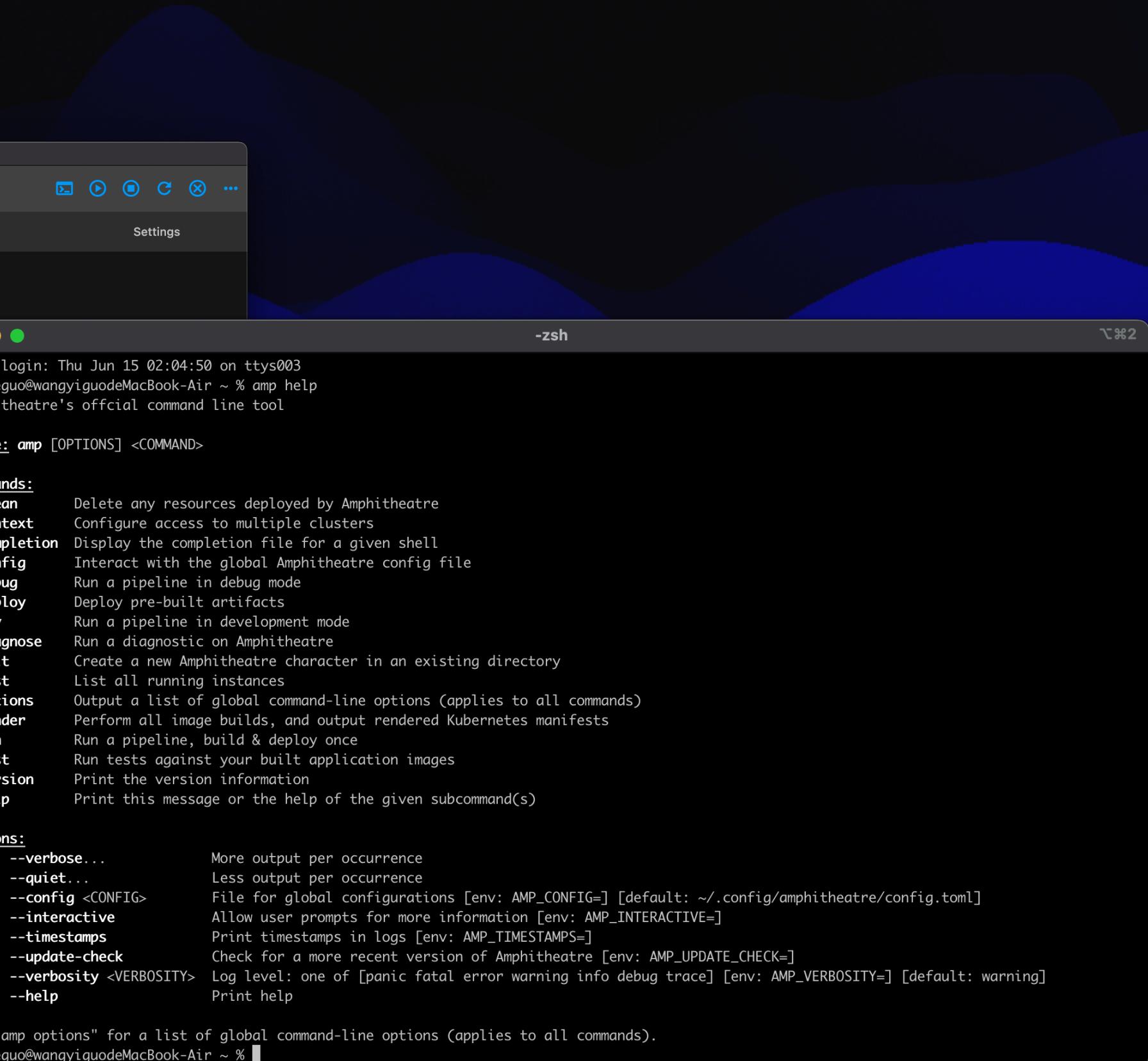
Compatible with various operating systems such as Windows, macOS, and Linux.



Web



Desktop



CLI

# Run a simple application (CLI)

## 1. It's a Rust application that loop print the 'Hello world'

```
fn main() {  
    loop {  
        println!("Hello, world!");  
    }  
}
```

## 2. Initialize the Character

```
$ amp init  
  
Scanning source code  
Detected Rust app  
Using the following build configuration  
    Builder: paketobuildpacks/builder:base  
    Buildpacks: docker.io/paketocommunity/rust  
Wrote config file .amp.toml  
Your Character is ready. run with `amp run`  
...
```

```
name = "amp-example-rust"  
version = "0.0.1"  
authors = ["Eguo Wang <wangeguo@gmail.com>"]  
edition = "v1"  
description = "A simple Rust example app"  
readme = "README.md"  
homepage = "https://github.com/amphitheatre-app/amp-example-rust"  
repository = "https://github.com/amphitheatre-app/amp-example-rust"  
license = "Apache-2.0"  
license-file = "LICENSE"  
keywords = ["example", "rust", "getting-started"]  
categories = ["example"]
```

Inside .amp.toml

## 3. 运行

```
amp run
```

# Run a multiple components application (Web)

1

Apply

title	status	Time
Fixed model's bug and added unit tests	REHEARSING	2020-10-12T12:02:46.407Z
MySQL configurations and added unit tests	REHEARSING	2020-10-12T12:03:06.294Z
Refactored handlers and added model, repository and services	REHEARSING	2020-10-12T12:04:32.855Z
The URL of the Leading Player	REHEARSING	2020-10-12T12:04:17.463Z
Updated README.md and added unit tests	REHEARSING	2020-10-12T12:03:26.634Z
Exit while env is empty	REHEARSING	2020-10-12T12:01:52.449Z
Renamed role with character	REHEARSING	2020-10-12T12:03:55.129Z
Updated README.md endpoint and token	REHEARSING	2020-10-12T12:01:36.113Z
Refactored play k8s resources deploy handler	REHEARSING	2020-10-12T12:01:14.776Z

Running instances

3

Run

← Refactored handlers and added model, repository and services

State: REHEARSING      Created At: 2020-10-12T12:04:32.855Z      Updated At: 2020-10-12T12:04:32.855Z

### Cast

Component	Description	Branch	Tag
kayn	The main backend for the MediaTrack's front-ends	amphitheatre	11bd82a9
message-service	Notification & Message service	amphitheatre	a00b2f00
riven-web	Riven Websocket Service	amphitheatre	0369756d

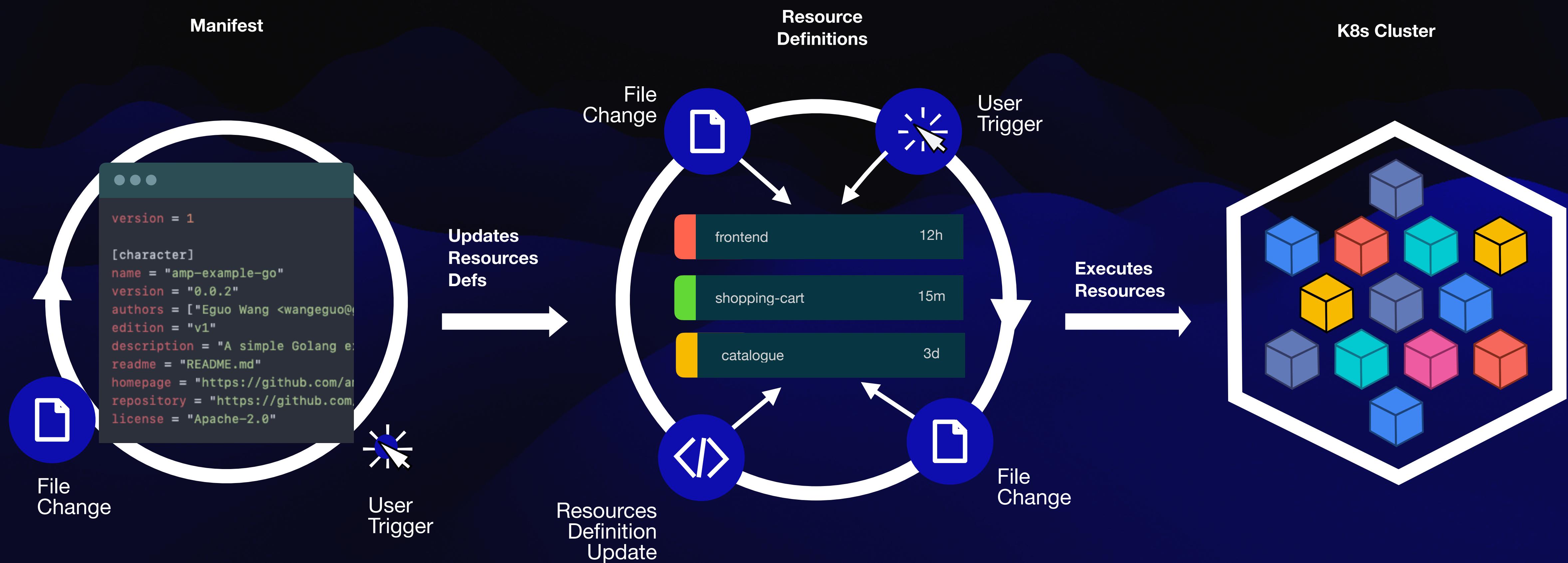
### dependencies

Component	Description	state	Reference
message-service	Notification & Message service	RUNNING	amphitheatre
asset-service	Asset service	RUNNING	amphitheatre

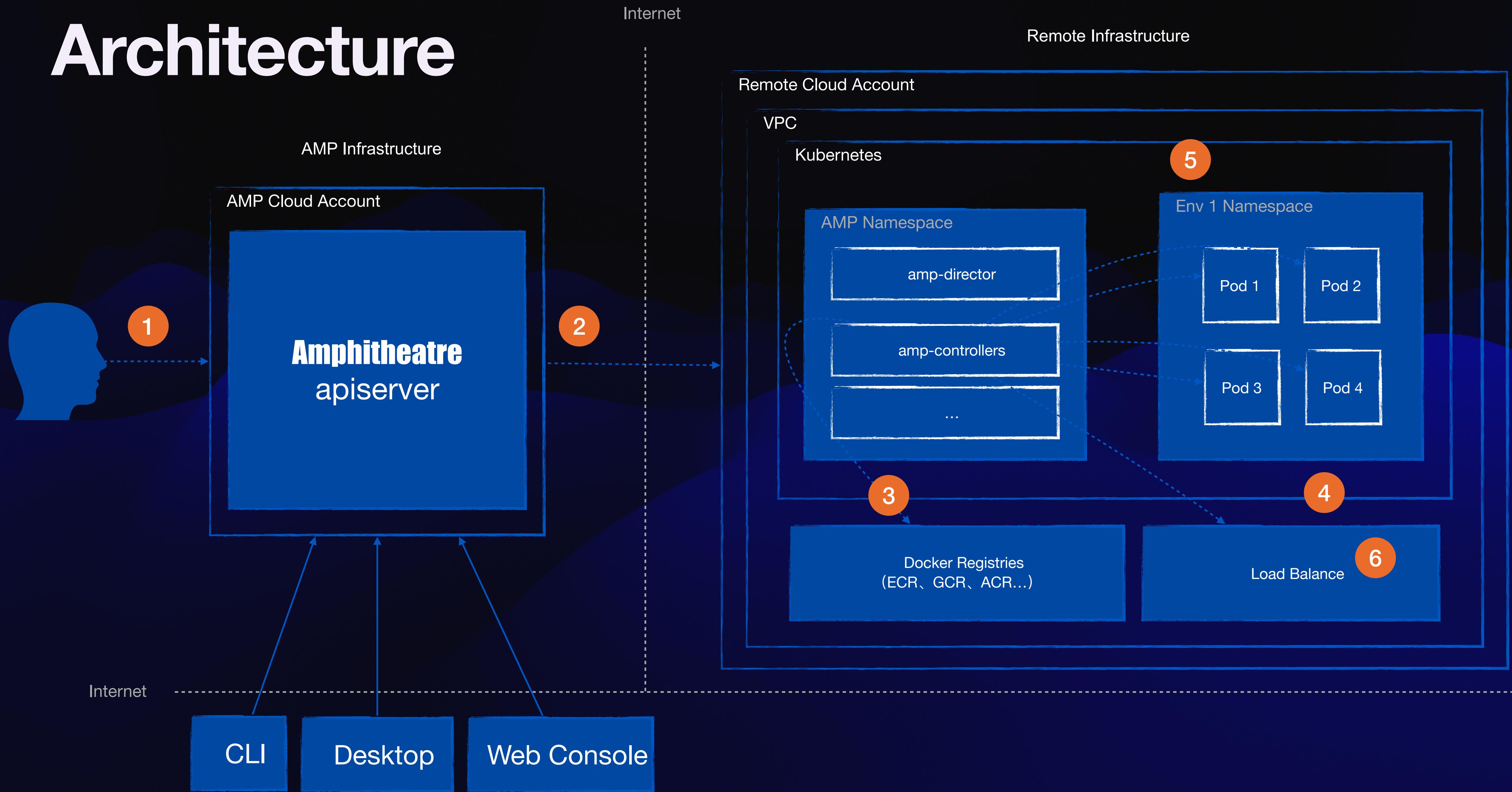
Select version

2

# How it works



# Architecture



# Special support for Solana



**Native**

Written using Solana's  
native Rust crates



**Anchor**

Written using the Rust  
framework Anchor



**Seahorse**

Written using the Python  
framework Seahorse



**C / C++**

Written using C and C++  
programming languages



**Solang**

Written using Solang  
programming languages

In addition, a testnet is automatically deployed locally (optional). For additional information, please visit the 'Amphitheatre for Solana' GitHub repository:  
<https://github.com/amphitheatre-app/solana-examples>

# Thanks