

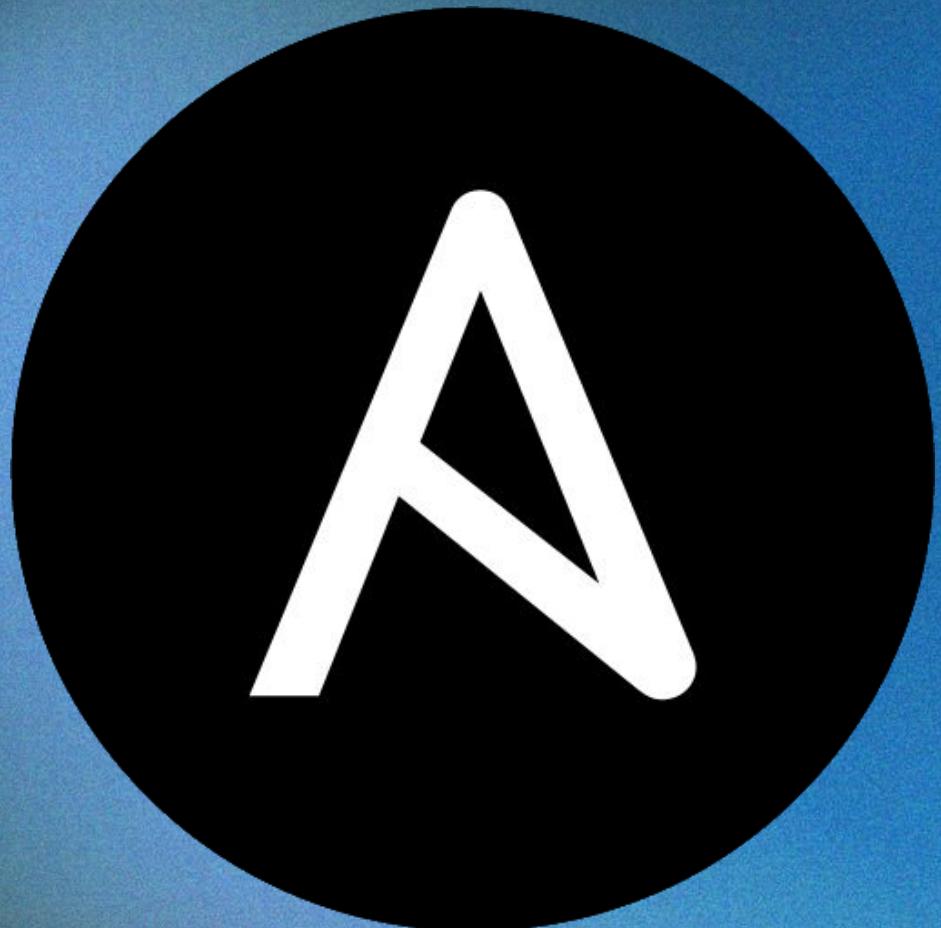
Grupo 10

Mizrahi, Thomas

Chayer, Iván

Flechard, Tristan

Di Toro, Camila



Ansible

Contenidos

01.

Ansible

02.

Objetivo

03.

Implementación

04.

Demo

01.

Conceptos de Ansible

¿Qué es Ansible?

- Ansible es una herramienta de automatización de código abierto.
- Facilita la gestión de configuraciones.
- Permite ejecutar tareas de forma simple.



¿Cómo funciona?

- Todo esto se realiza mediante conexiones SSH desde un *control node* hacia los *managed nodes*.

Control node

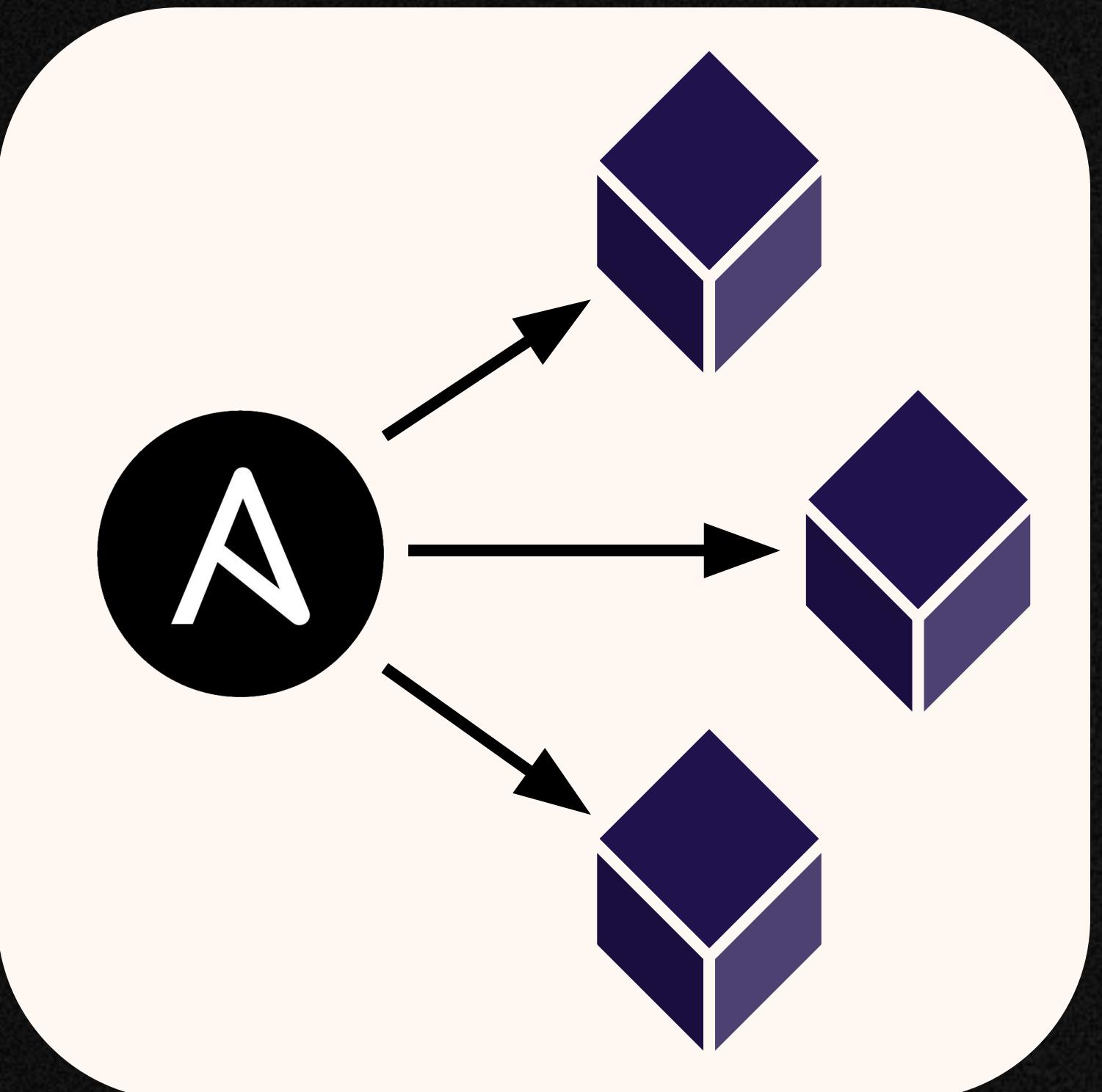
Máquina desde la que se corre Ansible

Target nodes

Son las máquinas que se van a coordinar con Ansible

Inventory

Reside en el *control node*, lista los *target nodes* y los agrupa en categorías



Playbooks



Playbook

Archivo YAML que lista un conjunto de *plays* para ser aplicados a un host

Play

La unidad de ejecución básica de Ansible; son un conjunto de tasks a ejecutar

Task

Una tarea individual a ejecutar, basada en un módulo

Ejemplo

```
- name: Ping all webservers
hosts: webservers
tasks:
  - name: Ping the servers
    ansible.builtin.ping:

  - name: Say hello
    ansible.builtin.debug:
      msg: Hello world
```

Características



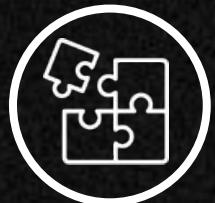
Arquitectura sin agentes

Los control nodes no precisan instalar nada en los managed nodes



Idempotencia

Las acciones realizadas por Ansible son predecibles y fácilmente repetibles



Simple y flexible

Se configura con YAML y usa credenciales del SO existentes



Escalable

Se pueden agregar más hosts fácilmente y soporta varios SOs

o2.

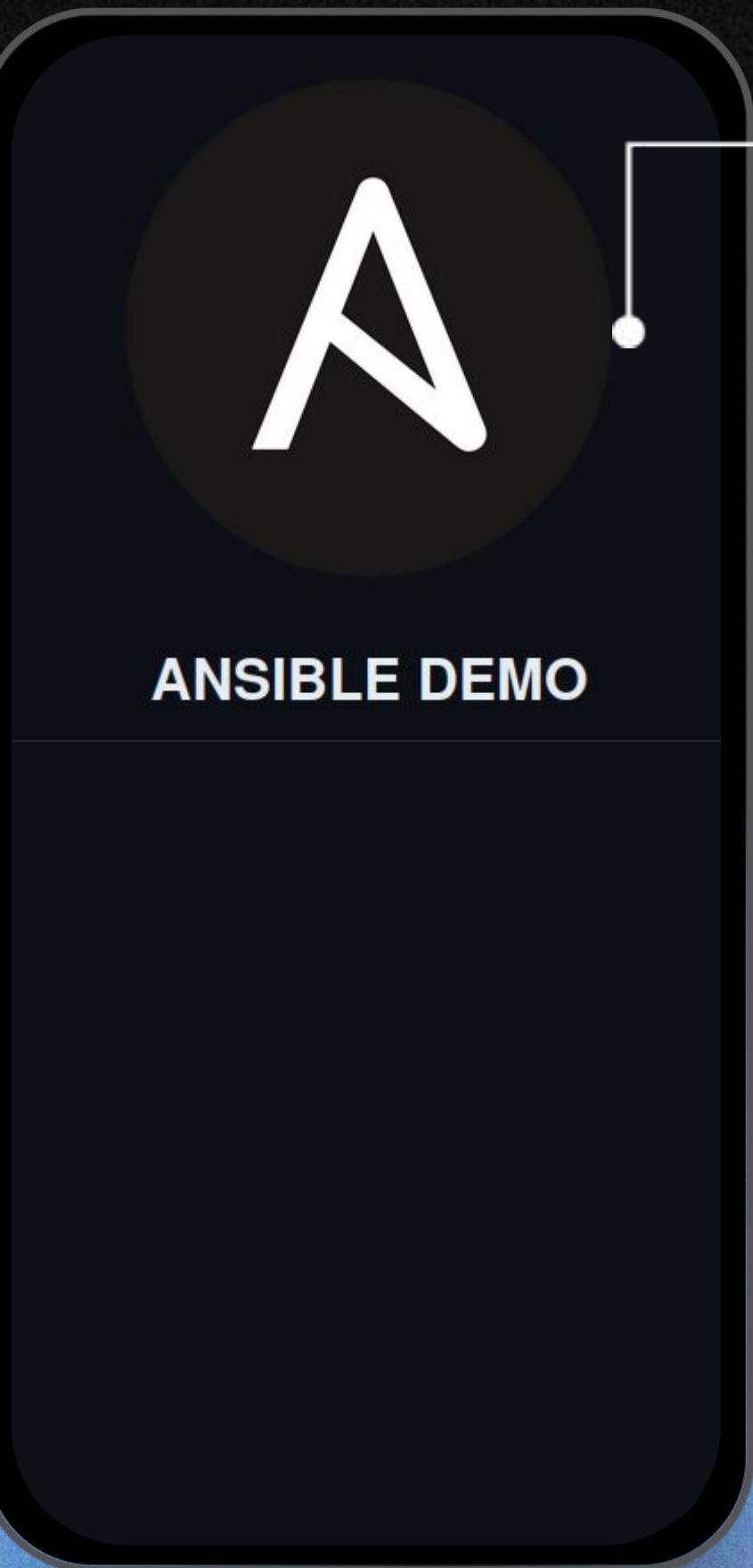
Objetivo



Ansible-Demo

Para ilustrar las posibilidades de Ansible, desarrollamos un servidor HTTP simple que almacena pares clave-valor en una base de datos.

La topología incluye un balanceador de carga para dos servidores web, que interactúan con dos bases de datos.



Tecnologías



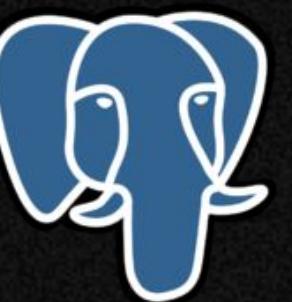
Rust

Lenguaje seguro y de alto rendimiento.



Nginx

Servidor web y proxy reverso de alto rendimiento con baja utilización de recursos



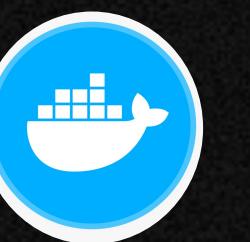
PostgreSQL

Base de datos relacional avanzada y de código abierto, conocida por su fiabilidad, robustez



Redis

Base de datos key-value en memoria rápida y escalable



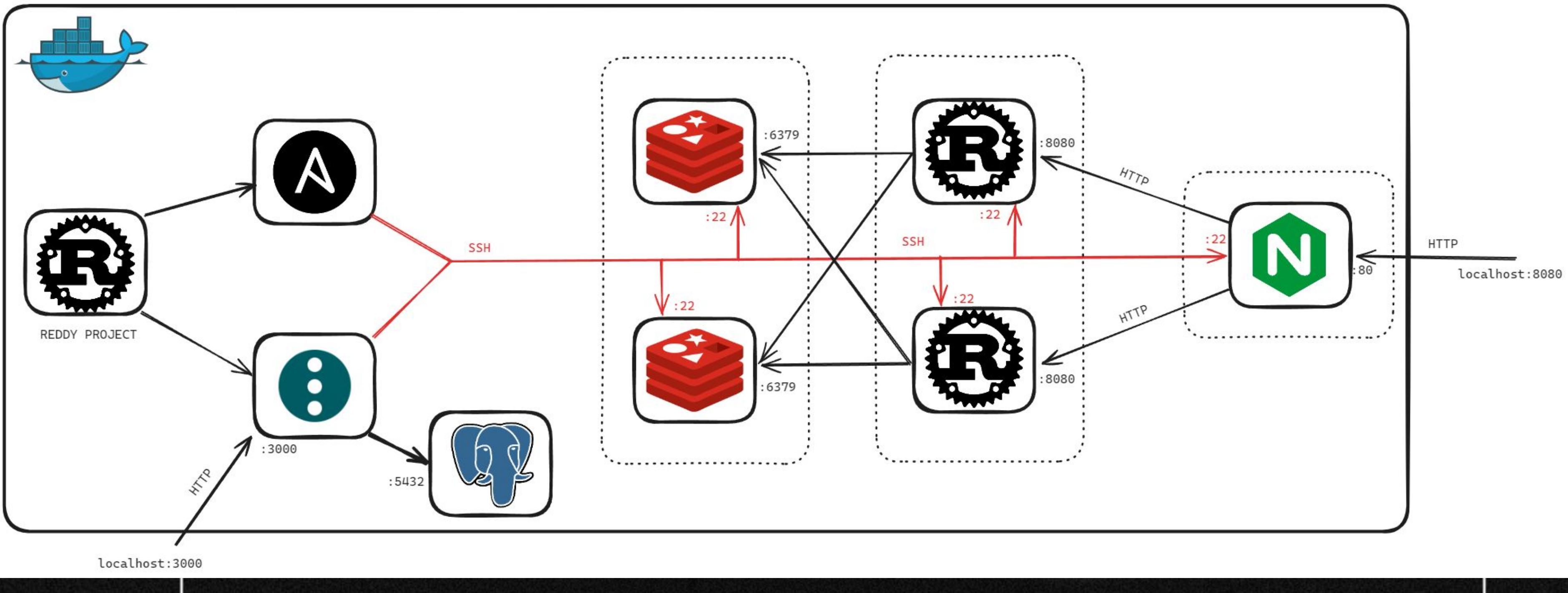
Docker

Plataforma de contenedores rápida, escalable, portable y eficiente

o2.

Implementación

Topologia



Reddy



Reddy es un pequeño wrapper de HTTP a Redis, escrito en Rust haciendo uso de Tokio y Axum.

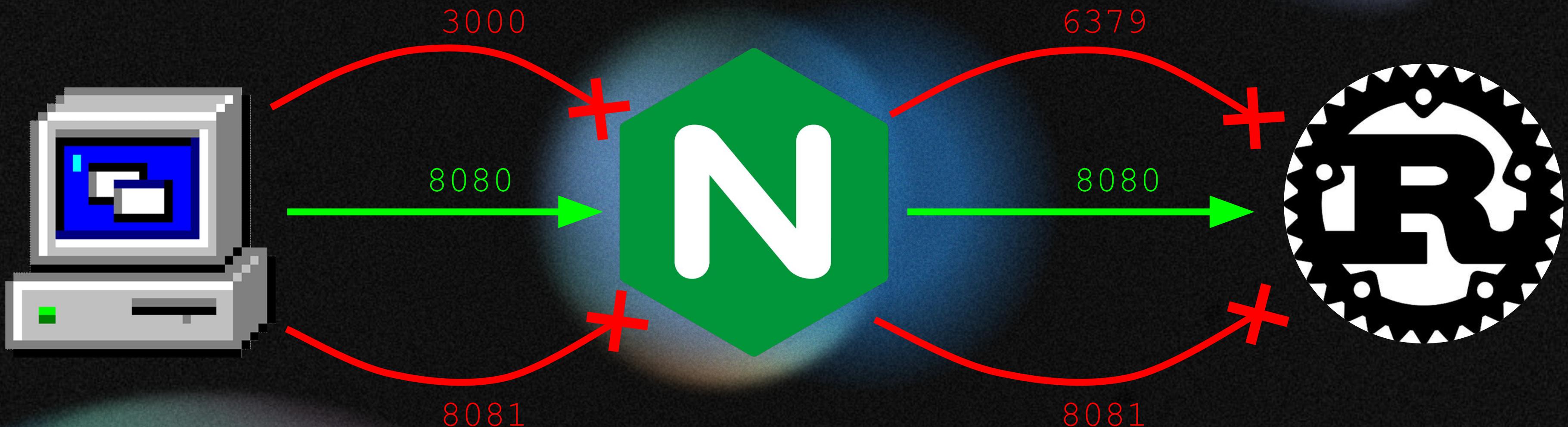
Su trabajo es traducir pedidos **HTTP GET** a pedidos **Redis GET** y traducir **HTTP POST** a **Redis SET**.

Reddy también se encarga de aplicar sharding, accediendo a una instancia de Redis distinta en base a la clave.

```
126
127     async fn set_key(
128         Path(key: String): Path<String>,
129         State(state: Arc<SharedStateInner>): State<SharedState>,
130         bytes: Bytes
131     ) -> Response<String> {
132         let redis_instance_index: usize = get_key_instance_index(&key, redis_instances);
133         let client: &Client = &state.clients[redis_instance_index];
134         let response: Builder = Response::builder() Builder
135             .header(key: "X-Reddy-Instance-Name", value: state.instance_name.clone()
136
137         let response_result: Result<Response<String>, ...> = match client.get_multi()
138             Err(err: RedisError) => response Builder
139                 .status(StatusCode::INTERNAL_SERVER_ERROR) Builder
140                 .body(format!("Could not connect to Redis: {err}")),
141             Ok(mut conn: MultiplexedConnection) => {
142                 let response: Builder = response Builder
143                     .header(key: "X-Redis-Instance-Index", value: redis_instance_index);
144
145                 match conn.set(key, value: bytes.deref()).await {
146                     Ok(s: String) => response.status(StatusCode::OK).body(s),
147                     Err(err: RedisError) if err.kind() == redis::ErrorKind::TypeMismatch =>
148                         response Builder
149                             .status(StatusCode::NOT_FOUND) Builder
150                             .body(String::new()),
151                     Err(err: RedisError) => response Builder
152                         .status(StatusCode::INTERNAL_SERVER_ERROR) Builder
153                         .body(format!("Error during request to Redis: {err}"))
154                 }
155             }
156         }
157     }
```



Reglas de UFW





Jinja2 Templates

Ansible permite utilizar templates Jinja2 para crear archivos parametrizados y evitar tener valores hardcodeados.

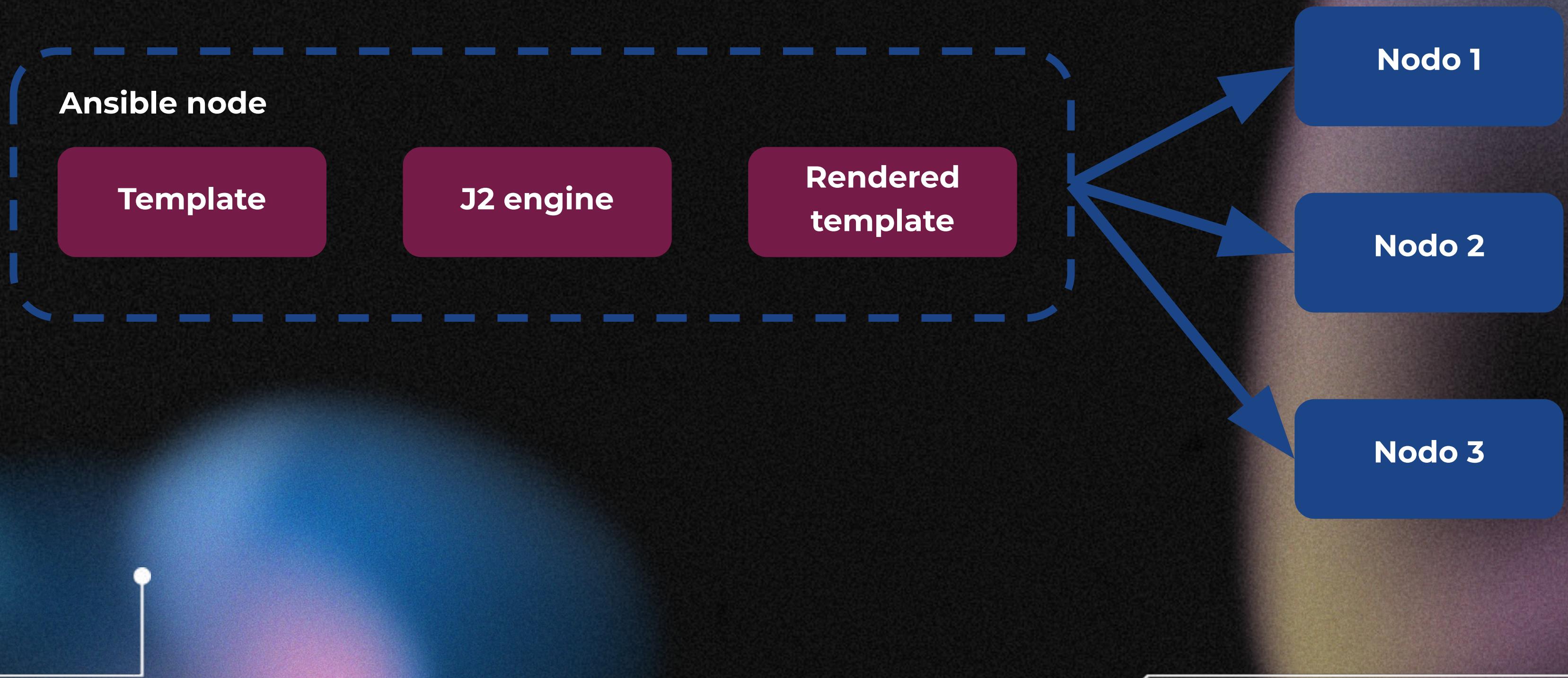
En este ejemplo usamos un template Jinja2 para crear dinámicamente un bloque de servidor de Nginx que es utilizado por el load balancer.

Se pueden usar variables injectadas al invocar el template o variables globales, como los nombres de hosts.

```
nginx_load_balancer.conf.j2 M X
templates > nginx_load_balancer.conf.j2
1 upstream app_servers {
2     {% for server in groups['webserver'] %}
3         server {{ hostvars[server]['ansible_host'] }}:8080 weight=1;
4     {% endfor %}
5 }
6
7 server {
8     listen 80;
9
10    location / {
11        proxy_pass http://app_servers;
12        proxy_set_header Host $host;
13        proxy_set_header X-Real-IP $remote_addr;
14        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
15        add_header X-Backend-Server $upstream_addr;
16    }
17
18 }
```



Jinja2 Templates

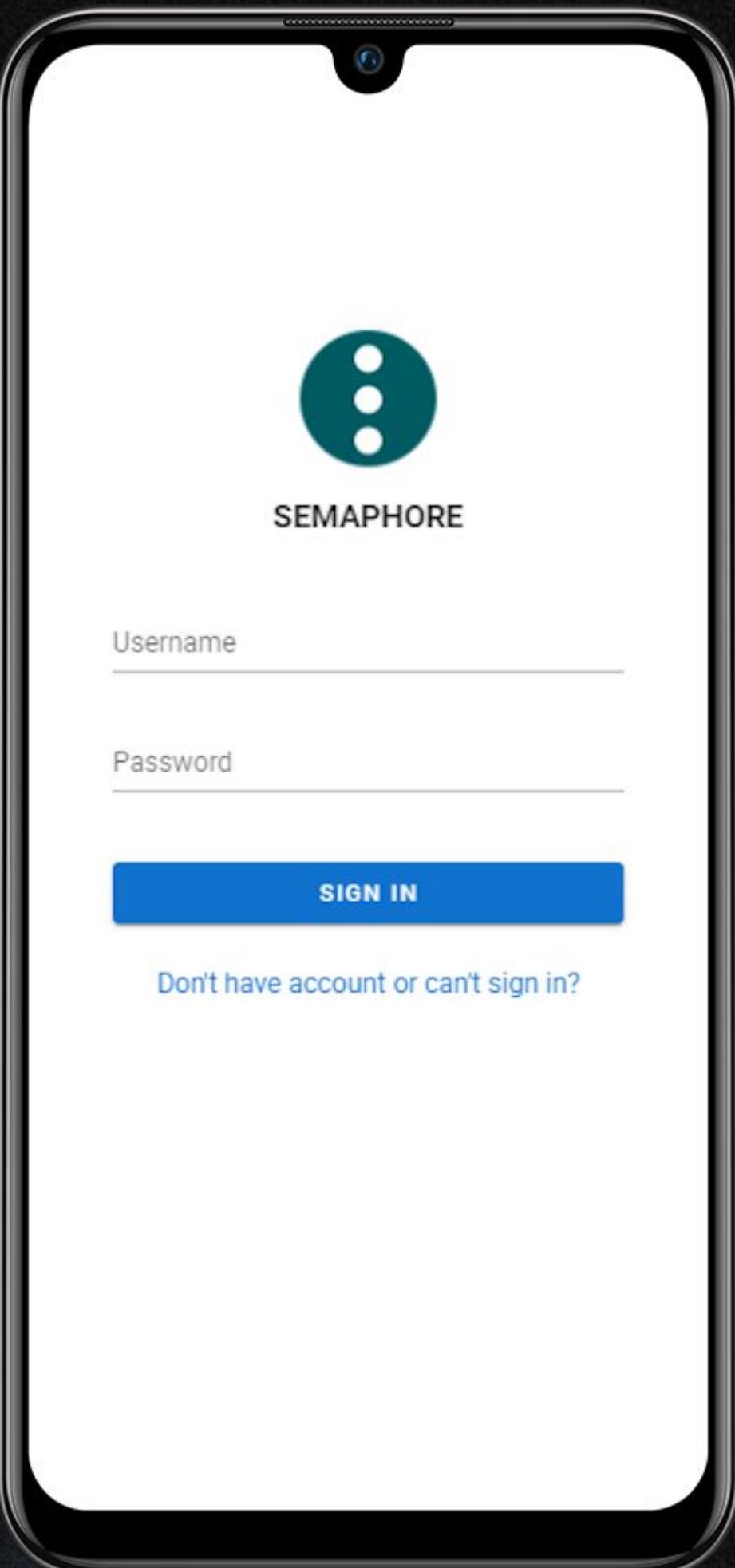


Semaphore

Interfaz gráfica web open source para ejecutar, programar y supervisar los playbooks.

- Manejo de claves SSH
- Control de variables de ambiente
- Pullea de un repositorio de GitHub para obtener información (inventory, playbooks, etc)

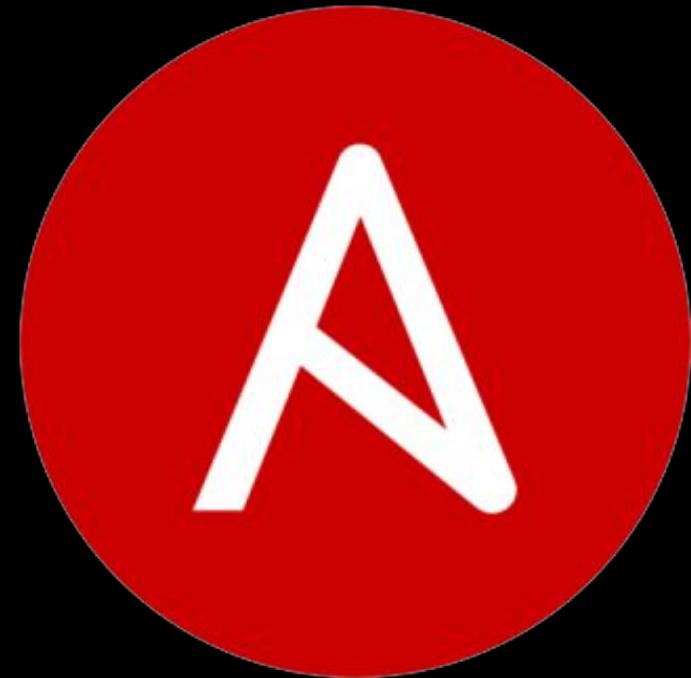
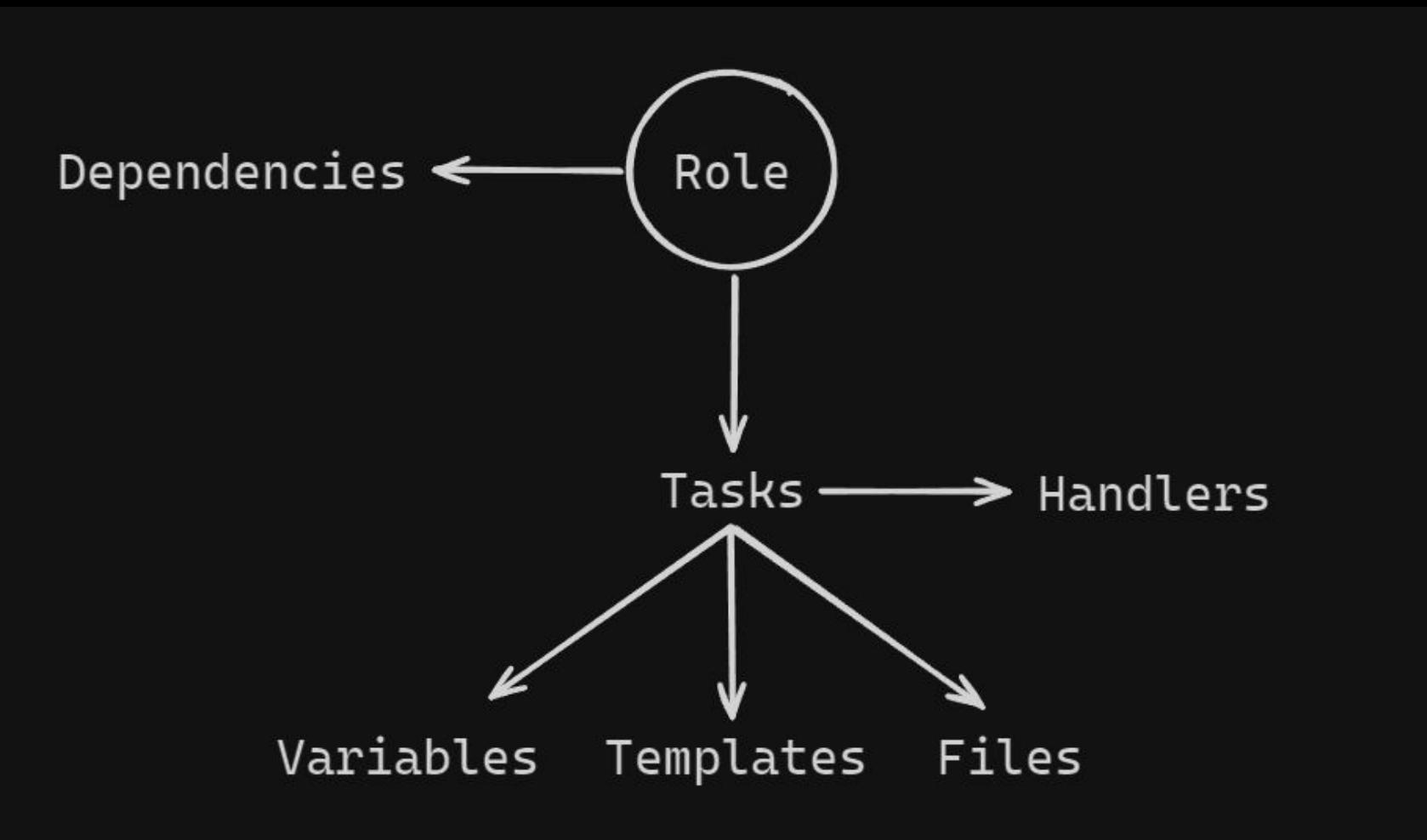
Permite interactuar con Ansible de forma amigable, sin tener que correr comandos en la terminal.



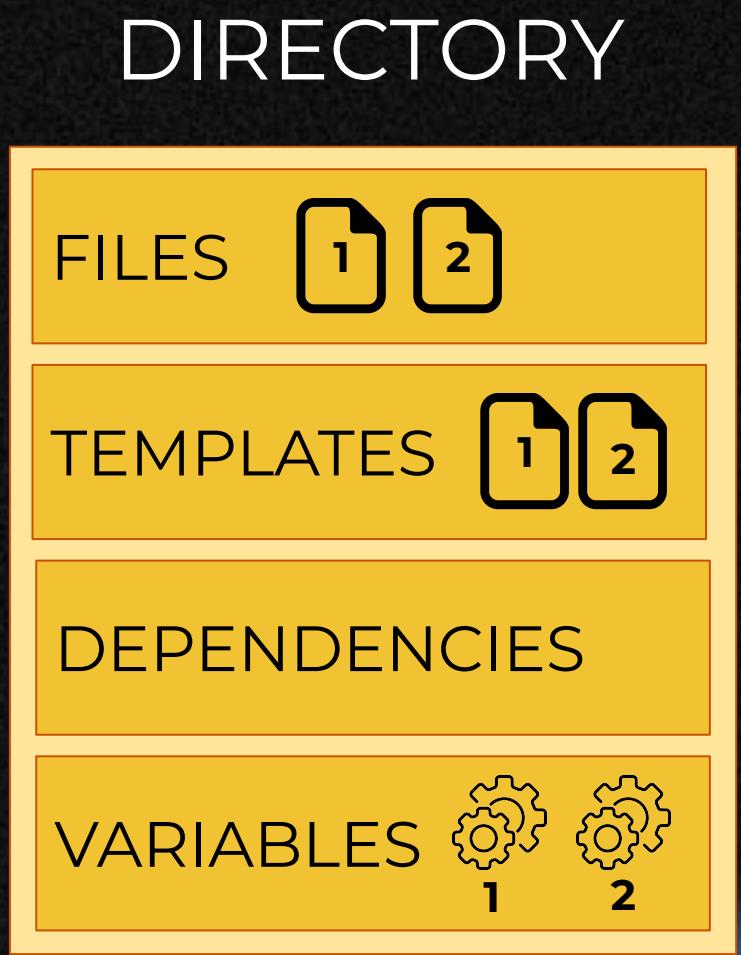
Ansible Roles

Un rol en Ansible es una manera de organizar y estructurar tasks, variables, archivos y templates en un formato modular y reutilizable.

- Facilita la reutilización de código
- Facilita la mantenibilidad de los playbooks



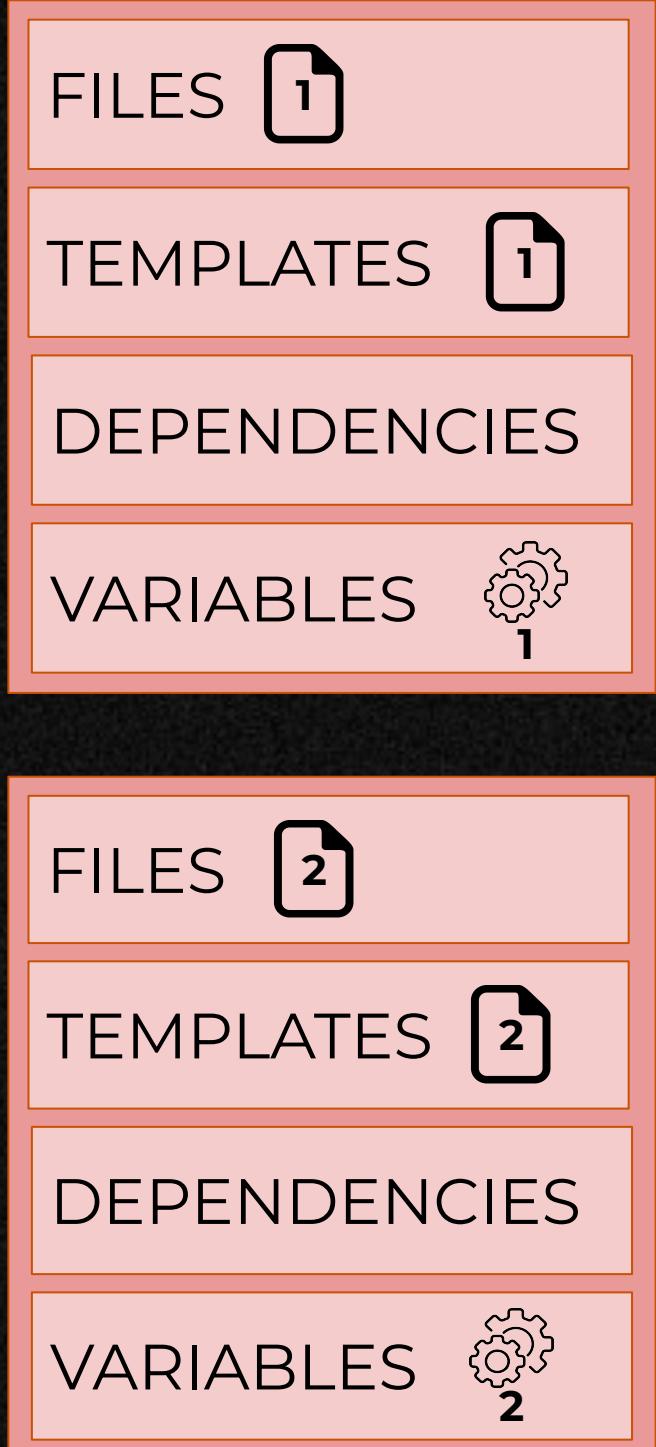
Ansible Roles



Role A

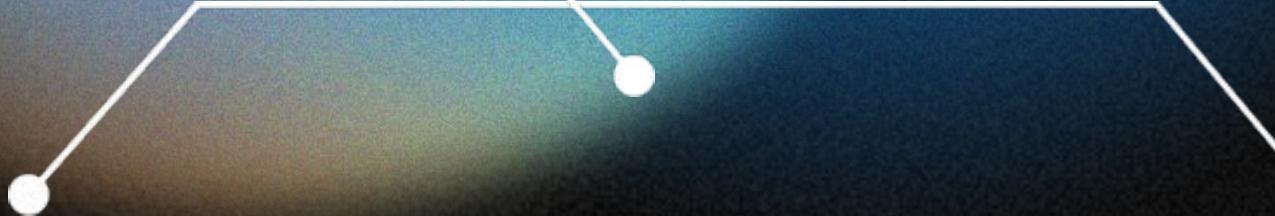


Role B



02.

Demo





Ansible

Gracias!

¿Preguntas?

CREDITS

Slides Carnival

**This presentation template is free for everyone to
use thanks to the following:**

SlidesCarnival for the presentation template

Pexels for the photos

HAPPY DESIGNING!