

The Nix logo is a stylized, geometric representation of a snowflake or a complex crystalline structure. It is composed of several interlocking, angular shapes in shades of blue and grey, creating a symmetrical, six-pointed star-like pattern. The logo is positioned on the left side of the slide, partially overlapping the title text.

Nix: InstagramClone

Dsys 2023 Challenge

Content



Goal



Developer Environment



Production Environment



Lessons learned



Demo

Goal

"Instagram Clone"

Minimal Docker images

K3S with traefik Load Balancer

Rails replication

Instagram Clone

- Simple App to upload / view / delete images

[Upload a New Image](#)

```
outputs = { self, nixpkgs }: {  
  nixosConfigurations = {  
    dsys = nixpkgs.lib.nixosSystem {  
      system = "x86_64-linux";  
      modules = [  
        ./configuration.nix  
        ./k3s.nix  
        ./gitlab-runner.nix  
      ];  
    };  
  };  
};
```

test image

asdfasf

[Download](#)

Delete

Local Development: Rails

Nix
develop

- Provides Dev Environment

Nix
build .#
docker

- Build rails image
- Minimal Image
- build dependencies from nixpkgs

Local Development rubyEnv

- Flake builds gemset.nix
- Dependencies are locked
- Dependencies from nixpkgs

```
let
  rubyEnv = pkgs.bundlerEnv {
    name = "rails-env";
    ruby = pkgs.ruby_3_1;
    gemdir = ./.;
  };

in {
  image = pkgs.dockerTools.buildLayeredImage {
    name = "dsys/rails";
    tag = "latest";
    created = "now";
    contents = [ pkgs.ruby_3_1 ./. ];
    config = {
      Cmd = [ "${rubyEnv}/bin/bundle" "exec" "rails" "s" "-b" "0.0.0.0"];
    };
    maxLayers = 120;
  };
}
```

Local Development rubyEnv

- Local VM with K3S
- Build from Flake

```
base = { lib, modulesPath, ... }: {
  imports = [ "${modulesPath}/virtualisation/qemu-vm.nix" ];

  # https://github.com/utmapp/UTM/issues/2353
  networking.nameservers = lib.mkIf pkgs.stdenv.isDarwin [ "8.8.8"

  virtualisation = {
    graphics = false;
    host = { inherit pkgs; };
    memorySize = 4096;
    diskSize = 10240;
    sharedDirectories.deployment = {
      source = "$SRC";
      target = "/home/deployment";
    };
  };
};

machine = nixpkgs.lib.nixosSystem {
  system = builtins.replaceStrings [ "darwin" ] [ "linux" ] syste

  modules = [ base ./module.nix ./k3s/k3s.nix ];
};

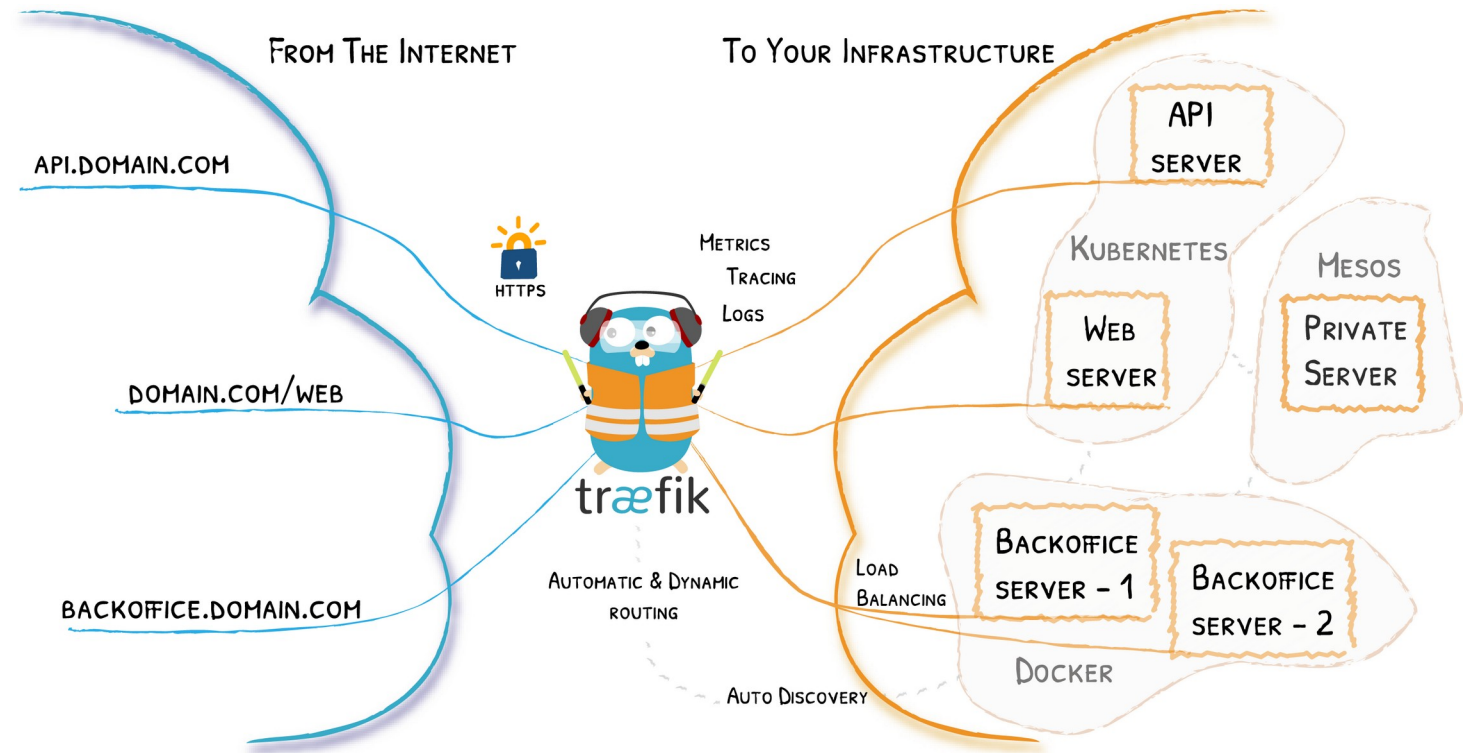
program = pkgs.writeShellScript "run-vm.sh" ''
  export NIX_DISK_IMAGE=$(mktemp -u -t nixos.qcow2)

  trap "rm -f $NIX_DISK_IMAGE" EXIT

  ${machine.config.system.build.vm}/bin/run-nixos-vm
'';
```

Local Deployment: traefik

- Deployed with yaml file



Production Environment

- AWS EC2
- NixOS
- K3S
- Gitlab-runner

```
outputs = { self, nixpkgs }: {  
  nixosConfigurations = {  
    dsys = nixpkgs.lib.nixosSystem {  
      system = "x86_64-linux";  
      modules = [  
        ./configuration.nix  
        ./k3s.nix  
        ./gitlab-runner.nix  
      ];  
    };  
  };  
};
```

Limitations



NO CENTRALIZED STORAGE



ONLY WORKS WITH
HTTP/1.1

Lessons learned



- Fabio:
 - Nix: good Tool bad documentation
 - Overkill?
- Timo:
 - Nix Kubernetes infrastructure is still underdeveloped
 - Traefik is very difficult to configure with Kubernetes/ TLS/ HTTP/3
 - Deployment and local development



Demo