Využití Nix/NixOps pro průběžnou integraci a nasazení software při vývoji

Tomáš Vlk

vedoucí práce: RNDr. Marek Rychlý, Ph.D.

vlktomas/nix-examples

Nix – správce balíčků

```
{ stdenv, fetchurl, someDependency }:

stdenv.mkDerivation rec {
  pname = "example";
  version = "1.0";

  src = fetchurl {
    url =
  "https://example.org/${pname}-${version}";
    sha256 = "0ssi1wpafc...7c9lng89nd";
  };


  buildInputs = [ someDependency ];


  buildPhase = ''
    gcc example.c -o example
  '';
}
```

NixOS – linuxová distribuce

```
{ config, pkgs, ... }: {

  imports = [
    ./hello.nix
  ];


  fileSystems."/mnt" = {
    fsType = "ext4";
    device = "/dev/sda1";
  };


  services.openssh.enable = true;


  environment.systemPackages = with pkgs; [
    wget vim
  ];
}
```

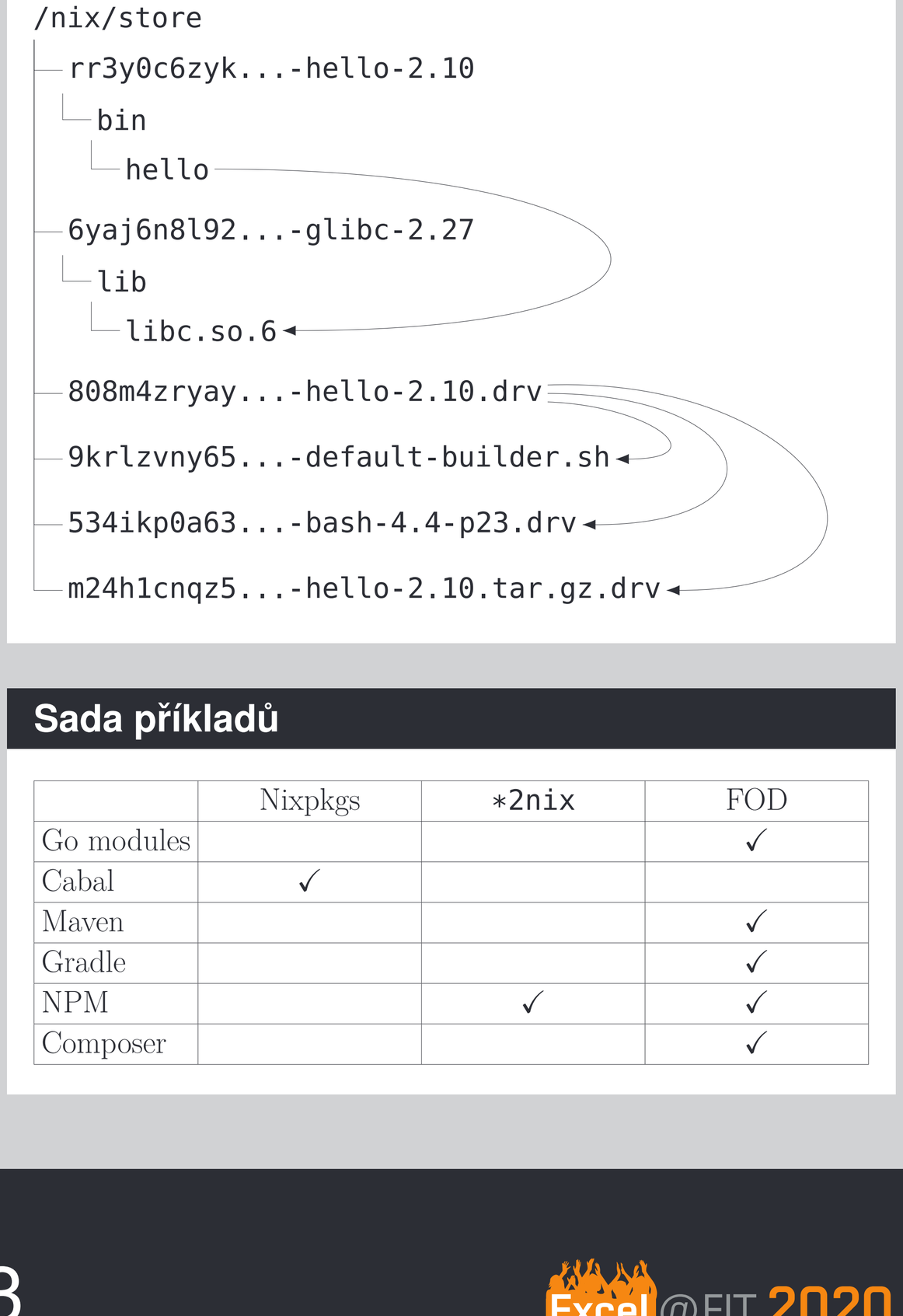
NixOps – IaC

```
{
  webserver = {
    deployment.targetEnv = "virtualbox";
    services.httpd.enable = true;
    services.httpd.virtualHosts = {
      "example.org" = {
        documentRoot = "/data";
    };
    fileSystems."/data" = {
      fsType = "nfs4";
      device = "fileserver:/";
    };
  };

  fileserver = {
    deployment.targetEnv = "virtualbox";
    services.nfs.server.enable = true;
    services.nfs.server.exports = "...";
  };
}
```

CI/CD pomocí Nix

```
{ pkgs ? import <nixpkgs> {} }:
with pkgs; rec {
  build = import ./default.nix { inherit pkgs };

  buildRaspberryPi = import ./default.nix {
    pkgs = pkgsCross.raspberryPi;
  };

  tests = pkgs.runCommand  "tests" { buildInputs = [ build ]; } ''
      echo "Hello world" > expected
      hello > given
      diff expected given > $out/result
    '';

  debPackage = releaseTools.debBuild {
    diskImage = vmTools.diskImageFuns.debian8x86_64 {};
    src       = build.src;
    name      = "${build.pname}-${build.version}-deb";
  };

  dockerImage = dockerTools.buildImage {
    name     = "hello";
    tag      = "latest";
    contents = [ build ];
    config   = { Cmd = [ "/bin/hello" ]; };
  };

  # toto může být vykonáno paralelně
  release = [ debPackage dockerImage ];

  # toto jedině sekvenčně
  pipeline = mkPipeline [ build tests release ];

  # vytvoření závislostí mezi fázemi
  mkPipeline = phases: (foldl mkDependency null phases);
  mkDependency = prev: next: next.overrideAttrs (
    oldAttrs: { prev = prev; }
  );
}
```

Závislosti v Nix store



Sada příkladů

	Nixpkgs	*2nix	FOD
Go modules			✓
Cabal	✓		
Maven			✓
Gradle			✓
NPM		✓	✓
Composer			✓