



Nix at Chatroulette

How we use it and its role in the Scala ecosystem



> whoami

- Software Engineer at Chatroulette
- Functional Programmer (Haskell, Scala)
- Author of Practical FP in Scala
- Nix enthusiast



Nix at Chatroulette

How it was introduced it & how we currently use it



The need for Nix

Backend	Frontend	Infrastructure
<ul style="list-style-type: none">• Scala (JVM)• Docker	<ul style="list-style-type: none">• TypeScript (Node)	<ul style="list-style-type: none">• K8s• Istio• Terraform

- Linux
- MacOS



@volpegabriel87



@gvolpe



shell.nix

```
let
  common      = import ./common.nix;
  stdenv      = common.pkgs.stdenv;
  optionals   = stdenv.lib.optionals;

  commonPkgs  = with common.pkgs; [
    kubectl
    mill
    nodejs-14_x
  ];

  customPkgs  = [ common.jdk ];
  darwinPkgs  = [ common.pulsarctl-darwin ];
  linuxPkgs   = [ common.pulsarctl-linux ];

  packages = commonPkgs ++ customPkgs ++ optionals stdenv.isDarwin darwinPkgs
    ++ optionals stdenv.isLinux linuxPkgs;

in
  common.pkgs.mkShell {
    buildInputs = packages;
  }
```



@volpegabriel87



@gvolpe



Dev environment



```
> git clone git@gitlab.com:cr/app.git  
> cd app && nix-shell  
> ./run.sh
```



```
> git clone git@gitlab.com:cr/app.git  
> echo "use_nix" >> app/.envrc  
> cd app && direnv allow
```



@volpegabriel87



@gvolpe

Docker images

```
● ● ●  
  
{ imgName ? "mono-jre", extraPkgs ? [] }:  
  
let  
  common = import ../common.nix;  
  pkgs    = common.pkgs;  
  extra   = map (p: pkgs.${p}) extraPkgs;  
in  
  pkgs.dockerTools.buildLayeredImage {  
    name = imgName;  
    tag  = "latest";  
    created = "now";  
    contents = [ common.jre ] ++ extra;  
  }
```

```
● ● ●  
  
nix-build docker.nix -o result-mono  
nix-build docker.nix -o result-tess \  
  --arg extraPkgs '["tesseract"]' \  
  --argstr imgName mono-jre-tesseract
```

```
● ● ●  
  
FROM $imageName  
COPY $jarName /$jarName  
ENTRYPOINT ["java", "-jar", "/$jarName"]
```



Nix on Gitlab runners

```

.rollout: &rollout |
  nix-shell --run "kubectl rollout restart deployment prod-chatroulette-$SERVICE_NAME -n $PROD_NAMESPACE"
  nix-shell --run "kubectl rollout status -w deployment prod-chatroulette-$SERVICE_NAME -n $PROD_NAMESPACE"

.run-all: &run-all |
  nix-shell --run "cd web/uiiv && npm ci --ignore-optional && npm run serve &"
  SERVICE_PORT=$MATCHER_PORT nix-shell --run "java -jar scala/out/mono/matcher/assembly/dest/out.jar &"
  SERVICE_PORT=$STICKY_PORT nix-shell --run "java -jar scala/out/mono/sticky/assembly/dest/out.jar &"
  nix-shell --run "cd web/uiiv && npm install && npm run wait"

.publish-pulsar-function: &publish-pulsar-function |
  nix-shell --run "mc config host add minio $ARTIFACT_STORE $MINIO_ACCESS_KEY $MINIO_SECRET_KEY"
  nix-shell --run "mc mb --ignore-existing minio/public && mc policy set public minio/public"
  nix-shell --run "mc cp functions.jar minio/public/deployment/functions/analytics/latest/out.jar"
```



@volpegabriel87



@gvolpe




Nix at Chatroulette

How we use Nix in our OSS projects



Nix & OSS at CR

<https://github.com/cr-org>



Chatroulette

We connect people.

📍 Zug, Switzerland 🔗 <https://about.chatroulette.com> @ChatrouletteAG ✉ info@chatroulette.com

📁 **Repositories** 56

📦 Packages

👤 People 10

👥 Teams 8

📁 Projects

Pinned repositories

📁 **neutron**

🔥 Purely functional Apache Pulsar client for Scala built on top of Fs2

🔴 Scala ☆ 36 🍴 4

📁 **supernova**

🔥 Apache Pulsar client for Haskell

🔴 Haskell ☆ 18 🍴 1

📁 **pulsar-demo**

🔥 Integration of supernova (Haskell client) and neutron (Scala client)

🔴 Scala ☆ 1



A first shell.nix

```
let
  nixpkgs = fetchTarball {
    name    = "NixOS-unstable-13-05-2020";
    url     = "https://github.com/NixOS/nixpkgs-channels/archive/6bcb1dec8ea.tar.gz";
    sha256  = "04x750byjr397d3mfwkl09b2cz7z71fcykhvn8ypxrck8w7kdi1h";
  };

  pkgs = import nixpkgs {};
in
  pkgs.mkShell {
    buildInputs = [
      pkgs.jdk11
      pkgs.sbt
    ];
  }
```



@volpegabriel87



@gvolpe



A first shell.nix

Running sbt



```
> sbt
[info] welcome to sbt 1.3.13 (Oracle Corporation Java 1.8.0_265)
[info] loading settings for project neutron-build-build-build from metals.sbt ...
[info] loading project definition from /home/gvolpe/workspace/cr/neutron/project/project/project
[info] loading settings for project neutron-build-build from metals.sbt ...
[info] loading project definition from /home/gvolpe/workspace/cr/neutron/project/project
[success] Generated .bloop/neutron-build-build.json
[success] Total time: 0 s, completed Sep 26, 2020 10:12:47 AM
```



@volpegabriel87



@gvolpe



pkgs.nix

```
{ java }:  
  
let  
  nixpkgs = fetchTarball {  
    name    = "nixos-unstable-2020-09-25";  
    url     = "https://github.com/NixOS/nixpkgs-channels/archive/72b9660dc18b.tar.gz";  
    sha256  = "1cqgpw263bz261bgz34j6hiawi4hi6smwp6981yz375fx0g6kmss";  
  };  
  
  config = {  
    packageOverrides = p: {  
      sbt = p.sbt.override {  
        jre = p.${java};  
      };  
    };  
  };  
  
  pkgs = import nixpkgs { inherit config; };  
in  
  pkgs
```



@volpegabriel87



@gvolpe

shell.nix

```
{ java ? "jdk11" }:  
  
let  
  pkgs = import nix/pkgs.nix { inherit java; };  
in  
  pkgs.mkShell {  
    buildInputs = [  
      pkgs.gnupg  
      pkgs.${java}  
      pkgs.sbt  
    ];  
  }
```



A better shell.nix

Running sbt



```
[info] welcome to sbt 1.3.13 (N/A Java 11.0.8)
[info] loading settings for project neutron-build-build-build from metals.sbt ...
[info] loading project definition from /home/gvolpe/workspace/cr/neutron/project/project/project
[info] loading settings for project neutron-build-build from metals.sbt ...
[info] loading project definition from /home/gvolpe/workspace/cr/neutron/project/project
[success] Generated .bloop/neutron-build-build.json
[success] Total time: 1 s, completed Oct 16, 2020, 11:22:20 AM
```



@volpegabriel87



@gvolpe



Nix on GH Actions

```
jobs:
  build:
    name: Build
    runs-on: ubuntu-18.04
    strategy:
      matrix:
        java:
          - jdk8
          - jdk11
          - jdk14
    steps:
      - name: "Install Cachix *"
        uses: cachix/cachix-action@v6
        with:
          name: neutron
          signingKey: "${{ secrets.CACHIX_SIGNING_KEY }}"

      - name: "Run with ${ matrix.java } 🚀"
        run: nix-shell --argstr java "${{ matrix.java }}" --run "sbt ++test" nix/ci.nix
```



@volpegabriel87



@gvolpe



Nix on GH Actions

Scala
on: pull_request

Build (jdk8)
succeeded 16 minutes ago in 3m 4s

Search logs

✓ Build (jdk8)	> ✓ Set up job	4s
✓ Build (jdk11)	> ✓ Run actions/checkout@v2.3.2	1s
✓ Build (jdk14)	> ✓ Starting up Pulsar 🐳	44s
	> ✓ Cache for sbt & coursier 🗄️	7s
	> ✓ Install Nix ❄️	3s
	> ✓ Install Cachix ❄️	7s
	> ✓ Run with jdk8 🚀	1m 54s
	> ✓ Shutting down Pulsar 🐳	3s
	> ✓ Post Install Cachix ❄️	1s
	> ✓ Post Run actions/checkout@v2.3.2	0s
	> ✓ Complete job	0s



@volpegabriel87



@gvolpe



sbt.nix

```
{ java ? "jdk11" }:  
  
let  
  pkgs = import ./pkgs.nix { inherit java; };  
in  
  pkgs.sbt
```



@volpegabriel87



@gvolpe



Nix & Scala

The Nix role in the Scala ecosystem



Nix & Scala

Nix in the eyes of the Scala community

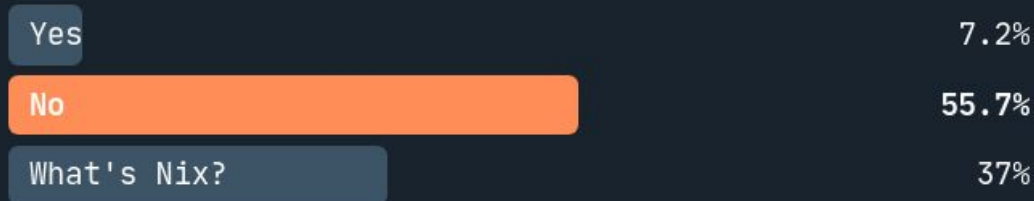


Gabriel Volpe @volpegabriel87 · Sep 26

Question for the #Scala community!

Do you use #Nix in Scala projects?

If you use it for more than creating dev environments with Nix Shell, I'd like to hear more!



332 votes · Final results



@volpegabriel87



@gvolpe



Nix & Scala

Managing binary dependencies (jars)

- <https://github.com/charleso/sbt2nix>
 - Last activity: Jan 2017
- <https://gitlab.com/teozkr/Sbtix>
 - Last activity: Mar 2020

```
{ pkgs ? import <nixpkgs> {} }:  
let  
  sbtixDir = fetchFromGitHub {  
    owner = "teozkr";  
    repo = "Sbtix";  
    rev = "<<current git rev>>"; # Replace as needed  
    sha256 = "<<<corresponding sha256 hash>>>"; # Replace as needed  
  };  
  sbtix = pkgs.callPackage "${sbtixDir}/plugin/nix-exprs/sbtix.nix" {};  
in  
  sbtix.buildSbtProgram {  
    name = "sbtix-example";  
    src = ./.;  
    repo = [ (import ./manual-repo.nix)  
             (import ./repo.nix)  
             (import ./project/repo.nix)  
          ];  
  }
```





Nix & Scala

<https://github.com/zaninime/sbt-derivation>

```
{ jdk ? "jdk11" }:  
  
let  
  pinned = import nix/pinned.nix;  
  config = import nix/config.nix { inherit jdk; };  
  sbtix = import pinned.sbt-derivation;  
  pkgs = import pinned.nixpkgs {  
    inherit config;  
    overlays = [ sbtix ];  
  };  
in  
pkgs.sbt.mkDerivation {  
  pname = "sbt-nixified";  
  version = "1.0.0";  
  
  depsSha256 = "02xxc6fy73v1m2awmavca7lg9r06fhjyg3q2q08c9r6nmy1s4b23";  
  
  src = ./.;  
  
  buildPhase = ''  
    sbt "sbt-nix-derivation/assembly"  
  '';  
  
  installPhase = ''  
    cp modules/nixified/target/scala-*/*-assembly-*.jar $out  
  '';  
}
```



@volpegabriel87



@gvolpe



Nix & Scala

<https://github.com/gvolpe/sbt-nix.g8>



```
nix-shell -p sbt --run "sbt new gvolpe/sbt-nix.g8"
```

DEMO TIME!

sbt-nix.g8

Get started with Nix and see how you can benefit from it in your Scala or cross-team projects.

- Motivation
- What is Nix?
- Install Nix
- Use cases
 - Reproducible development shell
 - Reproducible CI builds
 - Reproducible (and smaller) Docker images
 - Default Docker image using sbt-native-packager
 - Custom Nix Docker image using sbt-native-packager
 - Custom Nix Docker image using sbt-assembly
 - Docker images
- Pinning Nixpkgs
 - Using sbt with a different JDK
- Caching sbt derivations
- Managing dependencies (jars) with Nix
 - sbt-derivation
- Get started with sbt-nix.g8



@volpegabriel87



@gvolpe



THANKS!



@volpegabriel87



@gvolpe