Adding VMs to PATH

A NixOS VM built with `nixos-rebuild build-vm` is just another derivation. It can be even installed in a NixOS system's PATH by adapting the following snippet:

```
{ config, pkgs, lib, ... }:
let
  /* nixosVM is syntactic sugar for calling this. There is pkgs.nixos in the nixpkgs,
  but I don't use it here, because we need another kind of derivation. */
  nixosVM = configuration: (import <nixpkgs/nixos> { inherit configuration; }).vm;
  # This is a convenience for having the machine configurations.
  vms = {
    # You can write it inline...
    yuki = { config, pkgs, lib, ... }: {
      services.httpd.enable = true;
      services.httpd.adminAddr = "webmaster@example.com";
    };
    # Or import from another file.
    sakura = import VMs/sakura.nix;
  };
in {
  environment.systemPackages = [
    # Then just add it like this!
    (nixosVM vms.yuki)
  ];
}
```

To launch a VM automatically, a systemd service can be employed.

```
systemd.services.yuki-vm = {
  after = [ "network.target" ];
  path = [ (nixosVM vms.yuki) ];
  script = "run-*-vm";
  environment = {
    QEMU_NET_OPTS = "hostfwd=tcp::8022-:22";
    NIX_DISK_IMAGE = "/var/lib/vms/yuki.qcow2";
    QEMU_OPTS = "-m 1024M";
  };
  scriptArgs = "";
};
```

There is area for improvement, such as wrapping it in a module generating these on the fly.

The following Nix function dynamically generates a systemd service for a NixOS VM.

```
{ configuration,
  after ? [ "network.target" ],
  memory ? "384M",
  network_options ? "",
  args ? "",
  diskimagefolder ? "/var/lib/nixos-vms",
  restartIfChanged ? true }:
let
  nixos-system = configuration: (import <nixpkgs/nixos> { inherit configuration; });
  hostname = (nixos-system configuration).config.networking.hostName;
  nixos-vm = (nixos-system configuration).vm;
in {
  systemd.services.${hostname} = {
    inherit after restartIfChanged;
    environment = {
      QEMU_NET_OPTS = network_options;
      QEMU_ARGS = "${args} -m ${memory}";
      NIX_DISK_IMAGE = "${diskimagefolder}/${hostname}.qcow2";
    };
    script = "run-${hostname}-vm";
  };
}
```

This can be used like this:

```
{ config, pkgs, lib, ... }:
{
  imports = [
    (./nixos-vm-service.nix { configuration = (import vms/yuki.nix); memory = "1024M"; })
  ];
}
```

This will make it launch a virtual machine on boot.

Category (/wiki/Special:Categories): Virtualization (/wiki/Category:Virtualization)