Setup and install NixOS and Dotnet

Table of Contents

[Install and setup NixOS 2](#_Toc2879983)

[Install .Net Core SDK 15](#_Toc2879984)

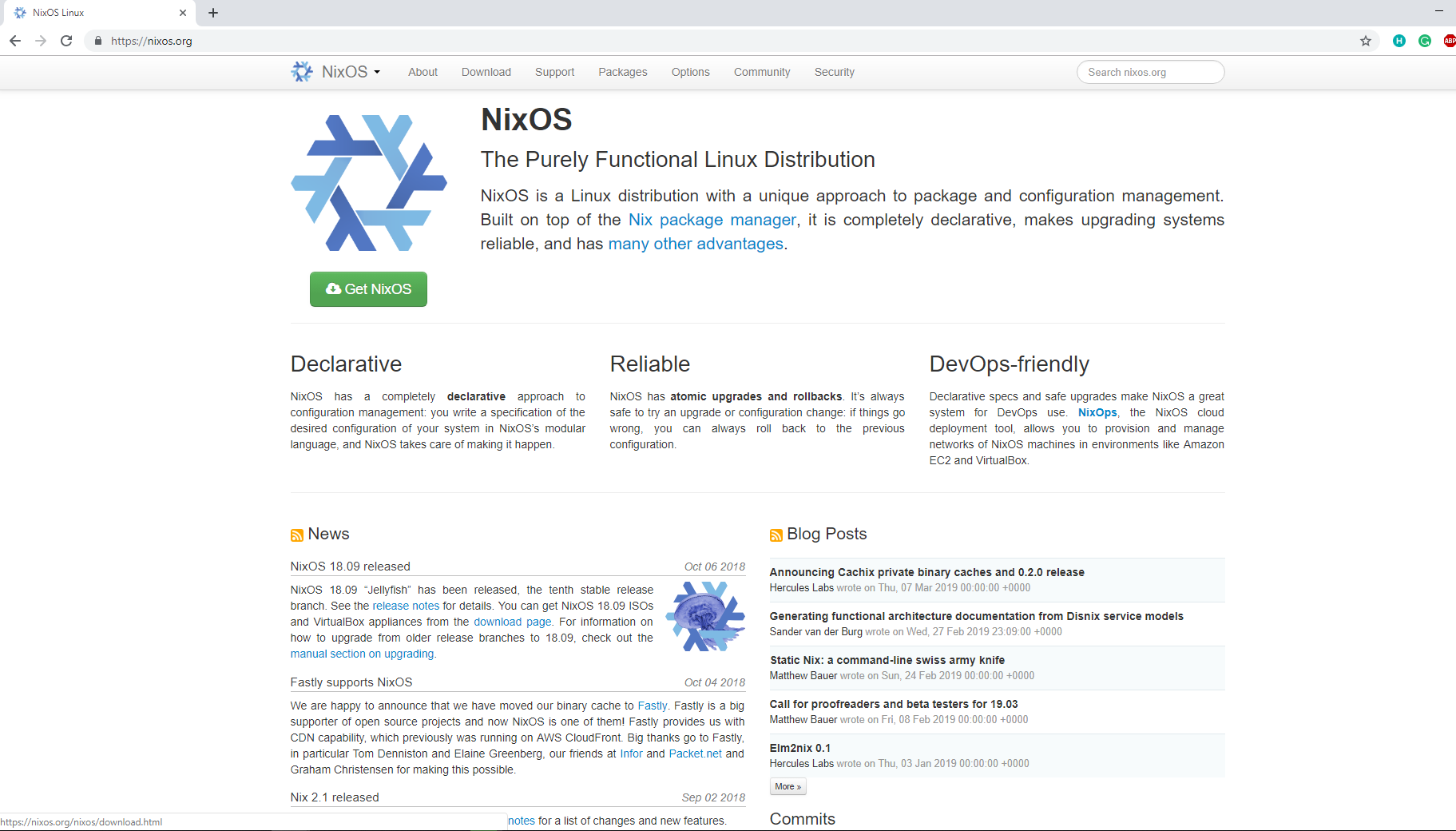
[Creating a sample Dotnet Core App 16](#_Toc2879985)

[Appendix 20](#_Toc2879986)

[Configuration file 20](#_Toc2879987)

# Install and setup NixOS

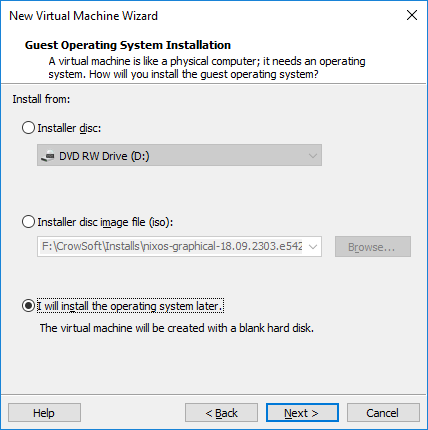
Go to <https://nixos.org> and download NixOS (Choose this version: [Graphical live CD, 64-bit Intel/AMD](https://d3g5gsiof5omrk.cloudfront.net/nixos/18.09/nixos-18.09.2318.4c089d328fe/nixos-graphical-18.09.2318.4c089d328fe-x86_64-linux.iso))

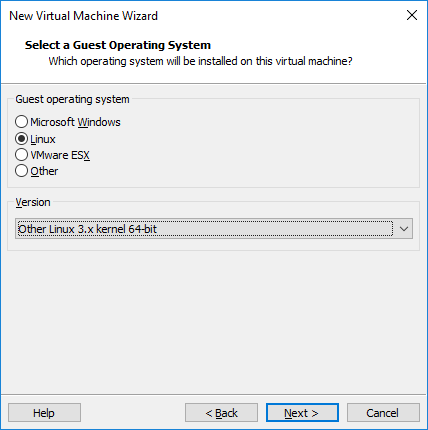


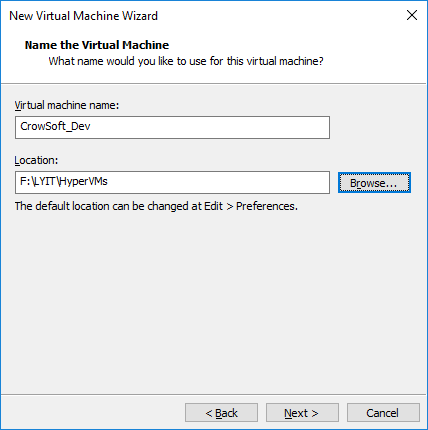
This is the NixOS ISO you need to download.

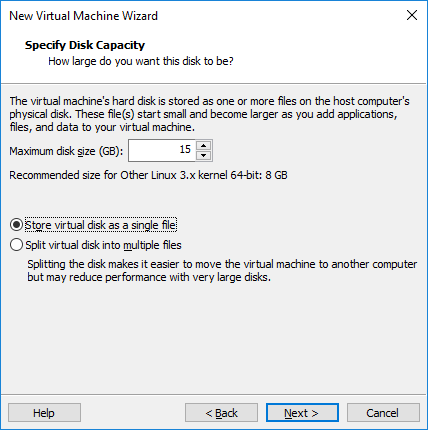


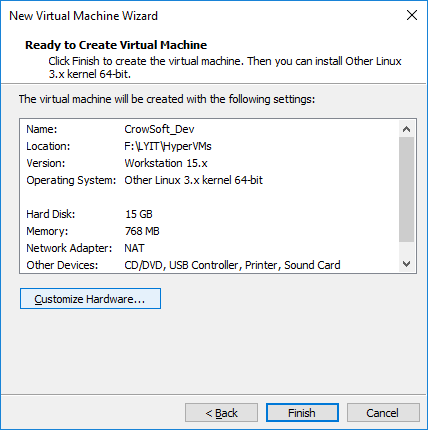


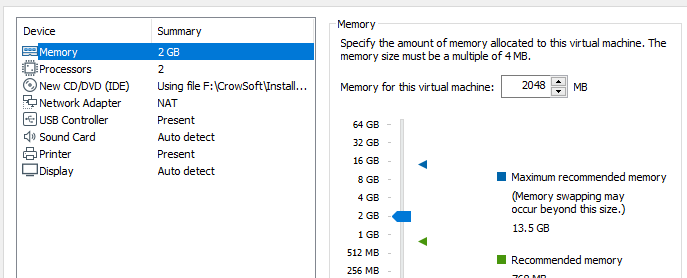


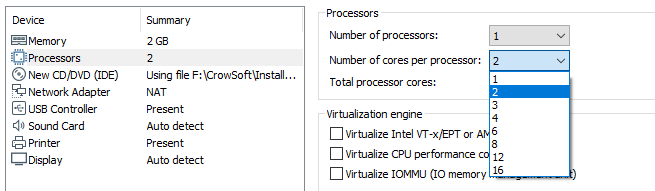


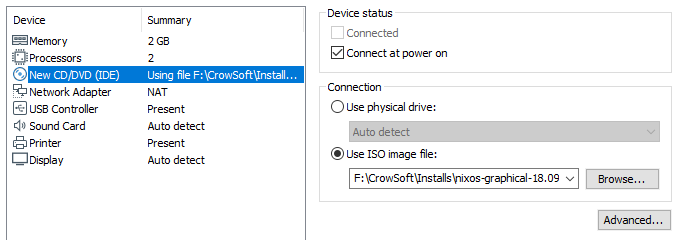








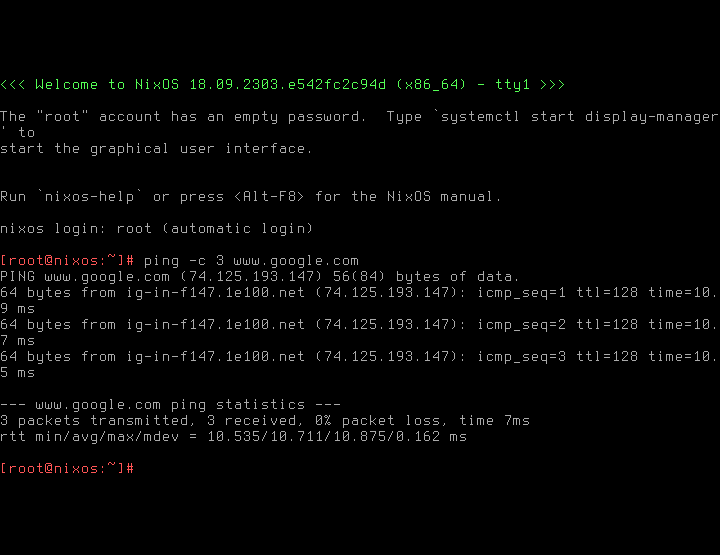




Then Click Finish

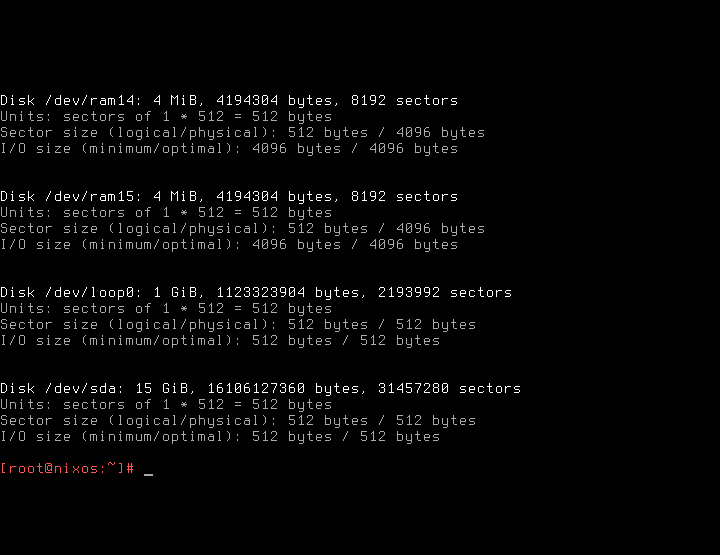
Power on the VM you just created. The installation will run automatically.

Use the following line to test internet connection: ping –c 3 www.google.com



You need to create your partitions: using : cfdisk

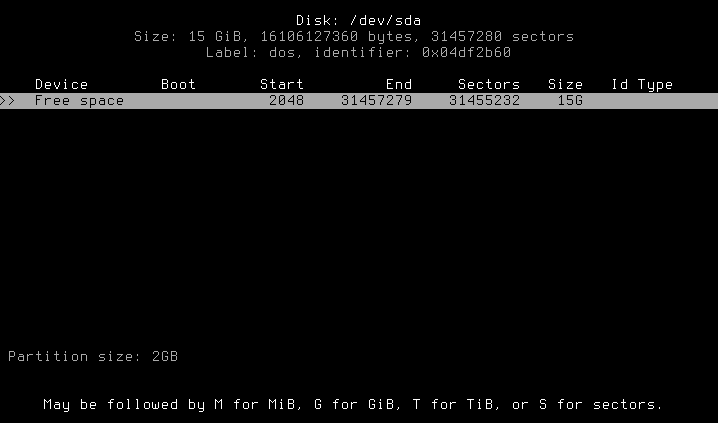
Use *fdisk -l* to check disk sectors

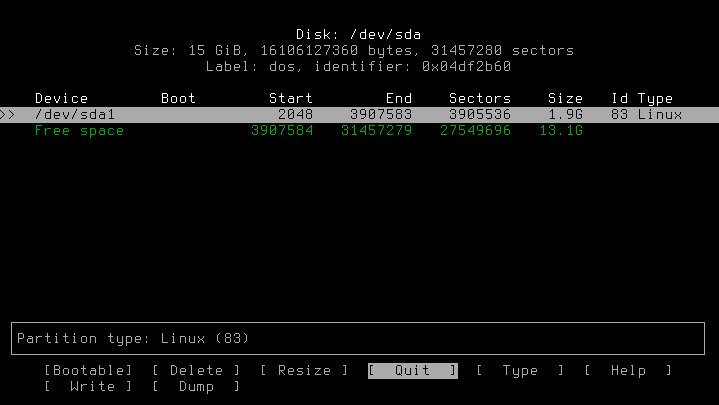


Choose *dos* and press enter

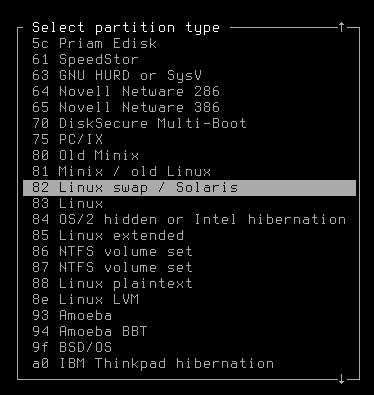


Select new, and change partition size to 2GB, and then choose Primary

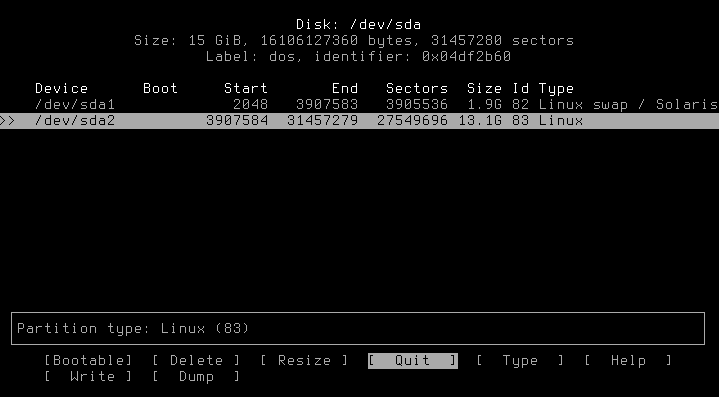




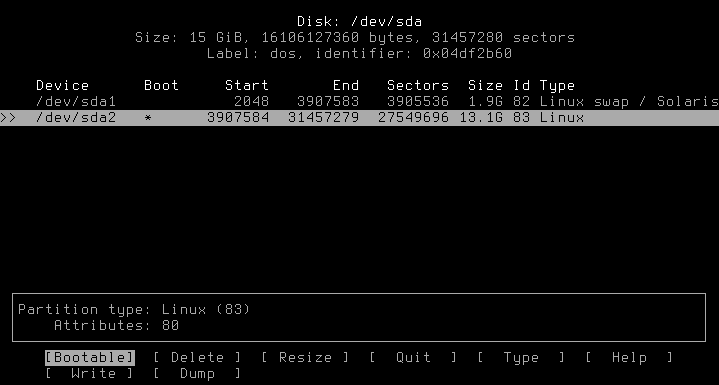
Then choose Type, and select Linux swap / Solaris



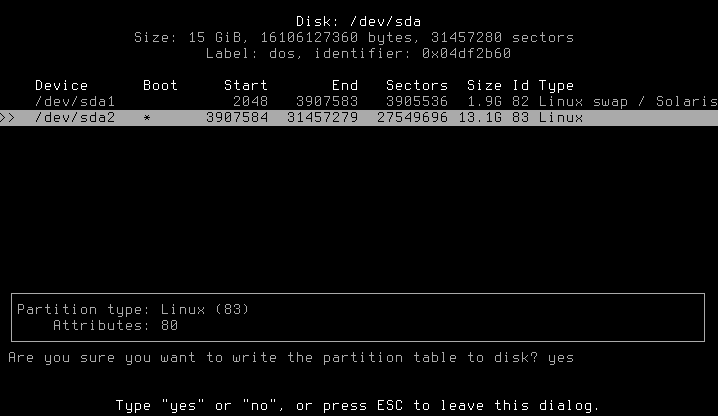
Select Free Space, then Enter for the remainder size available, then choose *Primary* and enter



Then choose *bootable* and press enter



Then choose *write*, and type *yes* to write the partition table, and press enter, and then *Quit.*

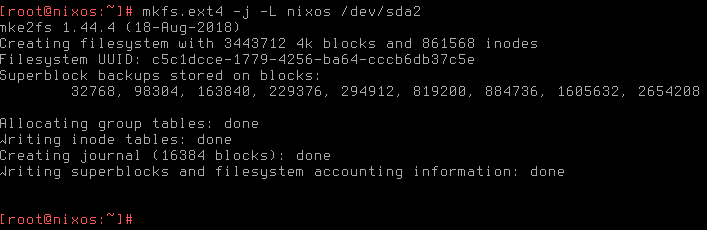


Then type: *mkswap /dev/sda1*

Then type: *swapon /dev/sda1*



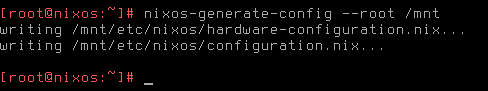
The type the following line: *mkfs.ext4 –j –L nixos /dev/sda2*



Enter the following line.



Enter the following line to generate a config file.



To edit the config file, type the following line:



Configuration File changes:

* There are a few settings in the configuration.nix file that needs to be changed.
* Look at the [appendix](#_Configuration_file) on the doc for the example. I uncommented a few lines and its highlighted in the appendix.

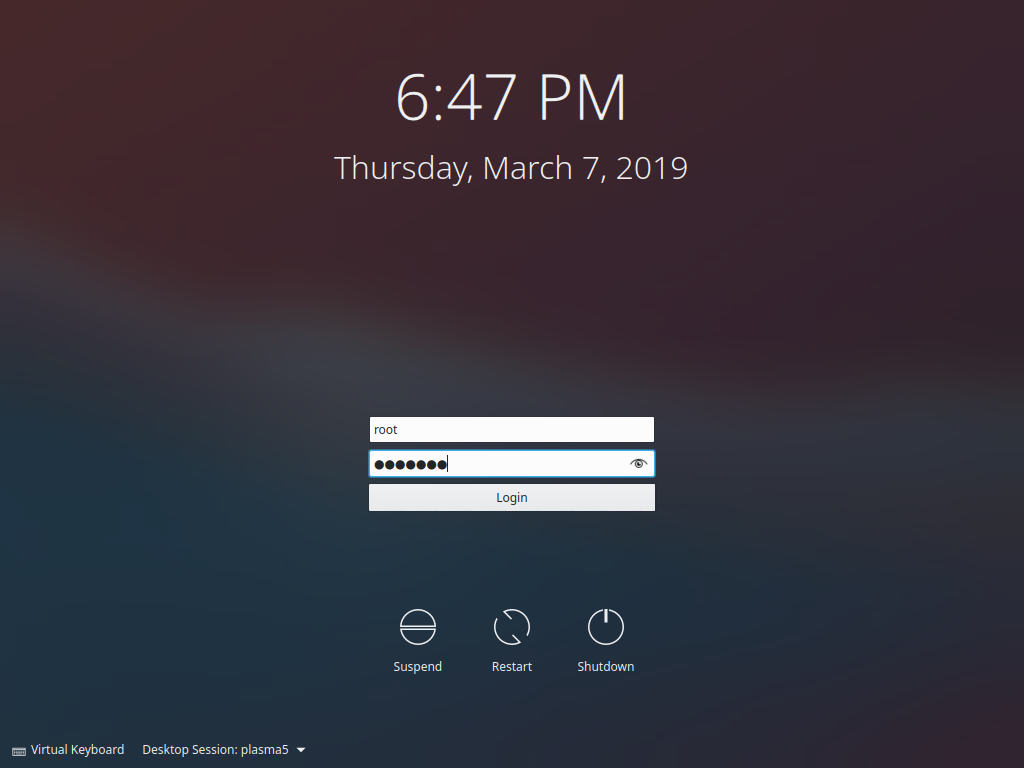
Then type nixos-install



Then enter the root password, and type *reboot* and enter



Enter with root user

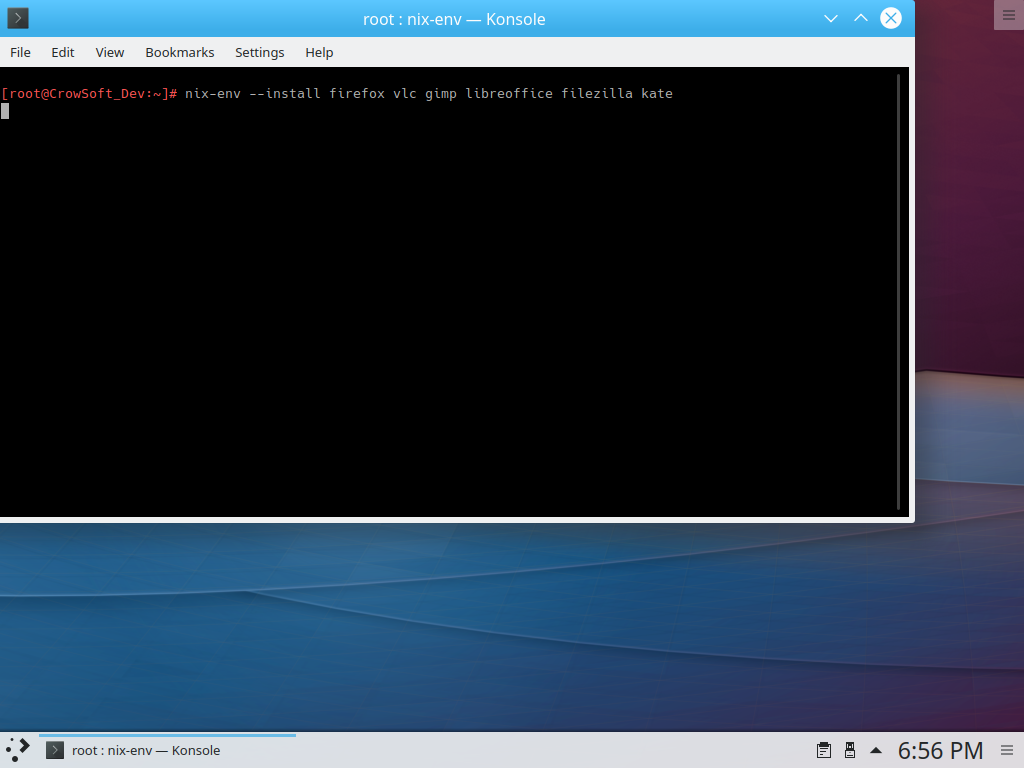


Add the following line to allow the VMWare guest to access your VM in etc/nixos/configuration.nix (use nano to edit).



The following command rebuilds your config: *nixos-rebuild switch*

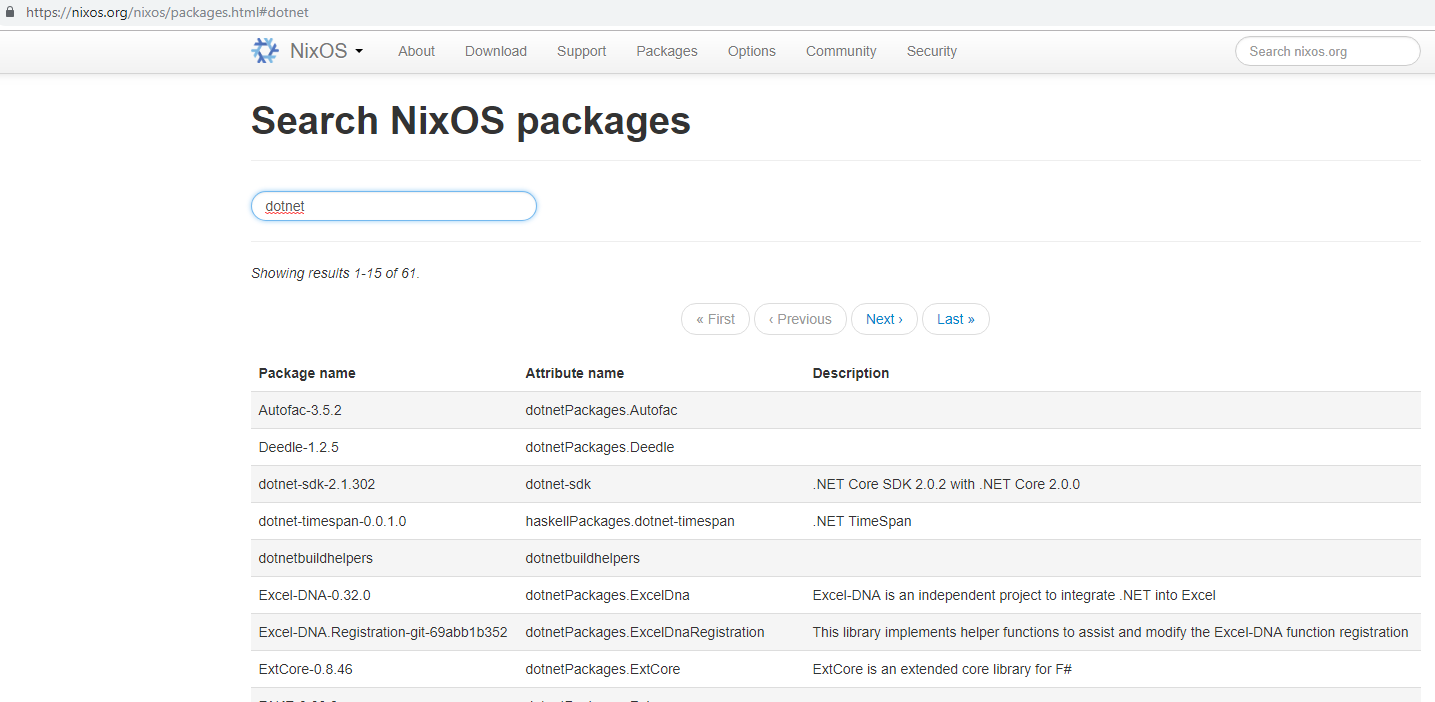
Install the following tools by using the following command as below.



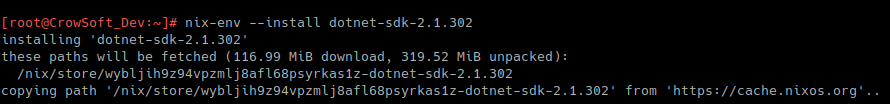
# Install .Net Core SDK

To add packages, you need the package name. Go to <https://nixos.org/nixos/packages.html>

Search for dotnet,

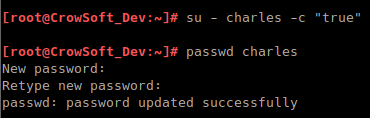


To install dotnet, type: *nix-env –install dotnet-sdk-2.1.302*



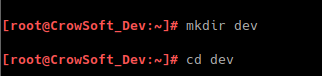
Type *reboot*

Add your user to sudoers and create a password (NB! Use your own username):

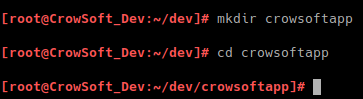


# Creating a sample Dotnet Core App

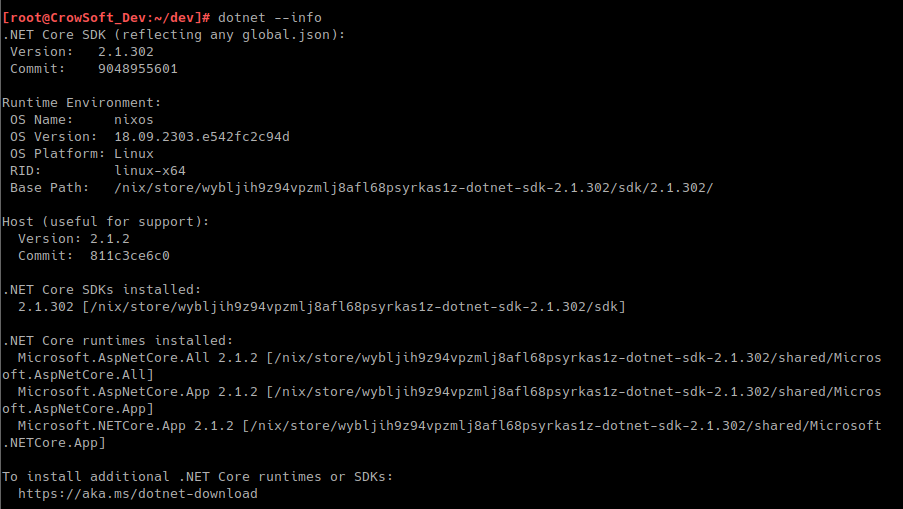
Create a new directory, called dev (dev environment).



Create a directory for your test app: called crowsoftapp

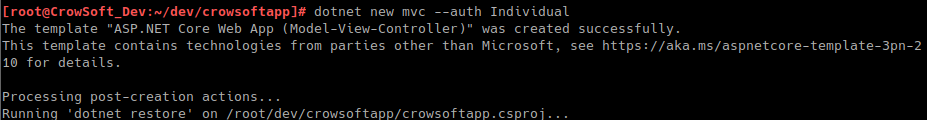


Test if dotnet is running: *dotnet --info*

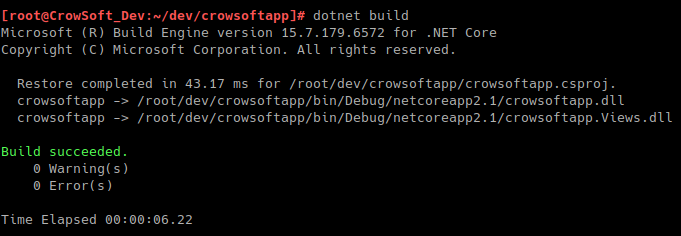


To create a sample web app, type the following line and press enter:

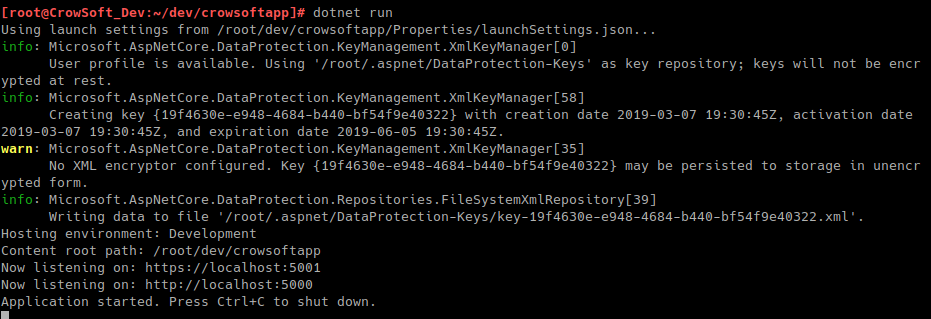
*dotnet new mvc --auth Individual*



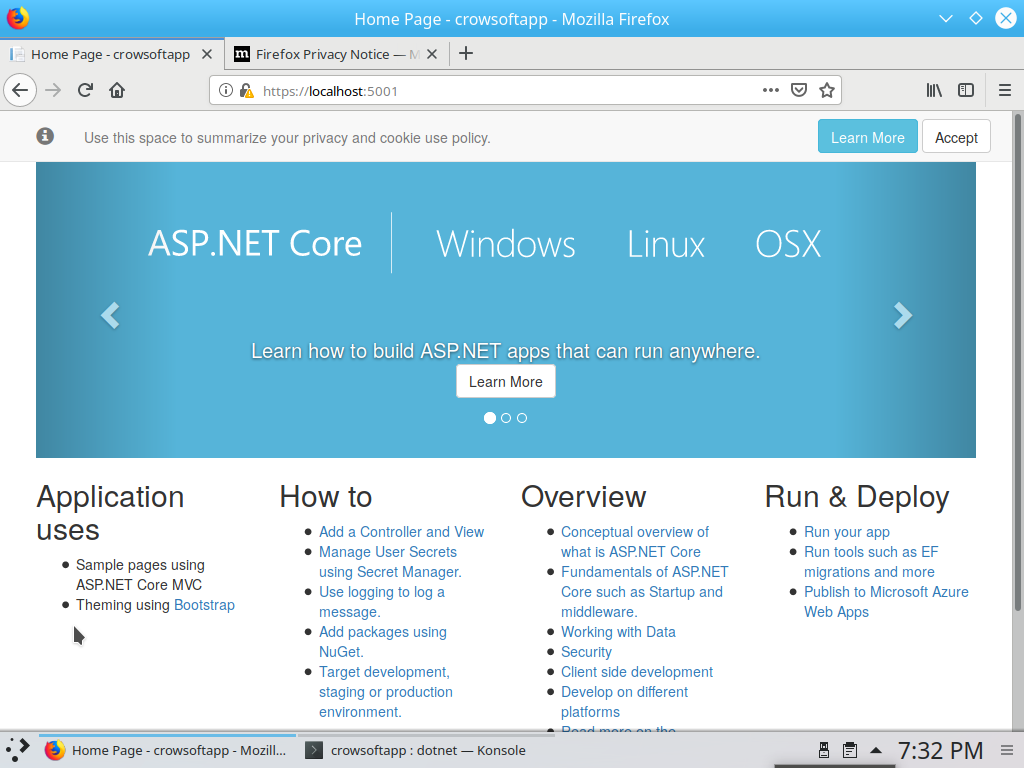
Then type and enter: *dotnet build*



To run the app, type: dotnet run

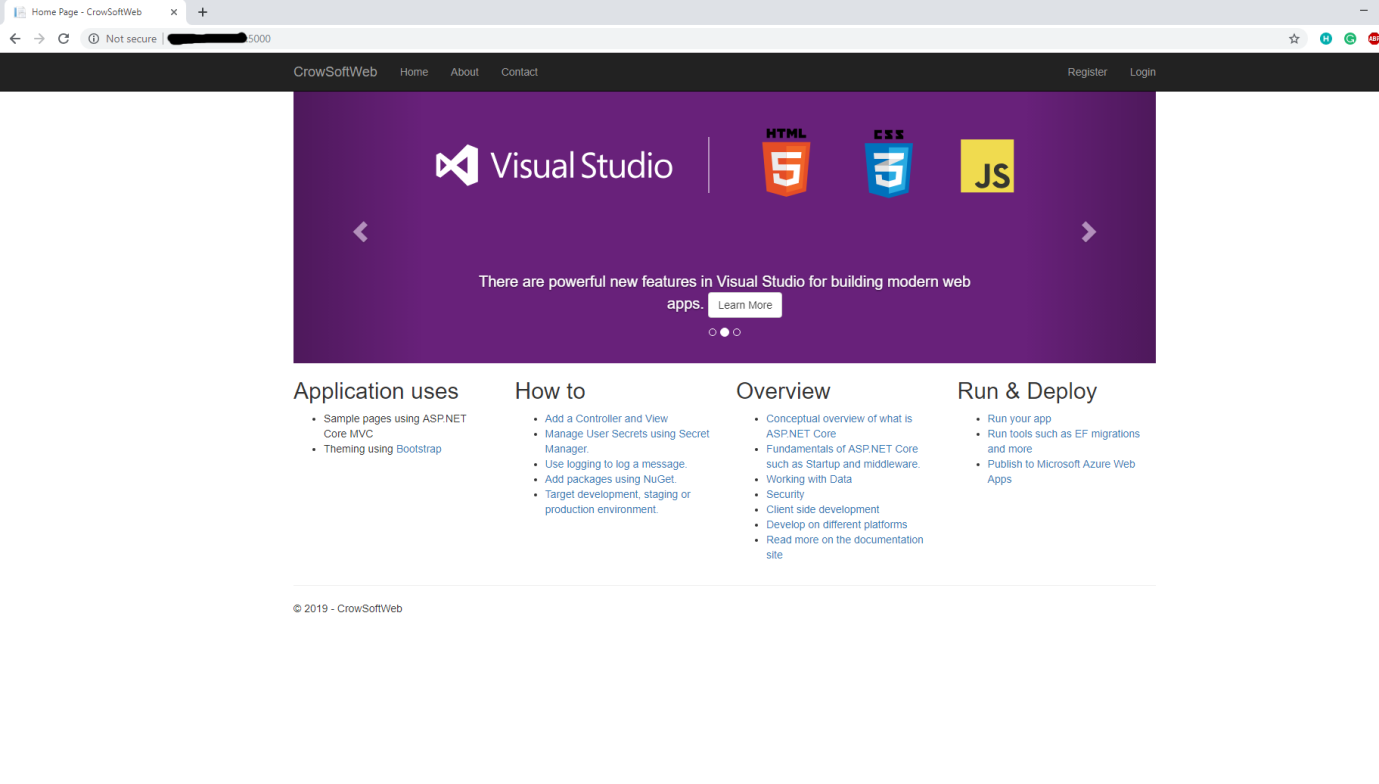


Leave the terminal running, and open firefox, the type in ***localhost:5000*** and press enter. You might need to add the exception due to no certificate running yet on https to open the new web app.



Running from your desktop (Windows 10 Pro), using Chrome, type in the NixOS VM’s IP address and port 5000.

Example: 0.0.0.0:5000



# Appendix

## Configuration file

# Edit this configuration file to define what should be installed on

# your system. Help is available in the configuration.nix(5) man page

# and in the NixOS manual (accessible by running ‘nixos-help’).

{ config, pkgs, ... }:

{

imports =

[ # Include the results of the hardware scan.

./hardware-configuration.nix

];

# Use the GRUB 2 boot loader.

boot.loader.grub.enable = true;

boot.loader.grub.version = 2;

# boot.loader.grub.efiSupport = true;

# boot.loader.grub.efiInstallAsRemovable = true;

# boot.loader.efi.efiSysMountPoint = "/boot/efi";

# Define on which hard drive you want to install Grub.

boot.loader.grub.device = "/dev/sda"; # or "nodev" for efi only

networking.hostName = "NixOS18\_Test"; # Define your hostname.

# networking.wireless.enable = true; # Enables wireless support via wpa\_supplicant.

# Configure network proxy if necessary

# networking.proxy.default = "http://user:password@proxy:port/";

networking.proxy.noProxy = "127.0.0.1,localhost,internal.domain";

# Select internationalisation properties.

# i18n = {

# consoleFont = "Lat2-Terminus16";

# consoleKeyMap = "us";

# defaultLocale = "en\_US.UTF-8";

# };

# Set your time zone.

# time.timeZone = "Europe/Amsterdam";

# List packages installed in system profile. To search, run:

# $ nix search wget

# environment.systemPackages = with pkgs; [

# wget vim

# ];

# Some programs need SUID wrappers, can be configured further or are

# started in user sessions.

# programs.mtr.enable = true;

# programs.gnupg.agent = { enable = true; enableSSHSupport = true; };

# List services that you want to enable:

# Enable the OpenSSH daemon.

services.openssh.enable = true;

# Enable VMWare Tools

services.vmwareGuest.enable = true;

# Open ports in the firewall.

# networking.firewall.allowedTCPPorts = [ ... ];

# networking.firewall.allowedUDPPorts = [ ... ];

# Or disable the firewall altogether.

networking.firewall.enable = false;

networking.firewall.allowedTCPPorts = [ 80 443 5000];

# Enable CUPS to print documents.

# services.printing.enable = true;

# Enable sound.

# sound.enable = true;

# hardware.pulseaudio.enable = true;

# Enable the X11 windowing system.

services.xserver.enable = true;

services.xserver.layout = "us";

services.xserver.xkbOptions = "eurosign:e";

# Enable touchpad support.

# services.xserver.libinput.enable = true;

# Enable the KDE Desktop Environment.

services.xserver.displayManager.sddm.enable = true;

services.xserver.desktopManager.plasma5.enable = true;

# Define a user account. Don't forget to set a password with ‘passwd’.

# users.users.guest = {

# isNormalUser = true;

# uid = 1000;

# };

security.sudo.enable = true;

security.sudo.extraConfig = ''

%wheel ALL=(ALL:ALL) NOPASSWD: ${pkgs.systemd}/bin/poweroff

%wheel ALL=(ALL:ALL) NOPASSWD: ${pkgs.systemd}/bin/reboot

%wheel ALL=(ALL:ALL) NOPASSWD: ${pkgs.systemd}/bin/systemctl suspend

'';

# Define a user account. Don't forget to set a password with ‘passwd’.

users.extraUsers.myuser = {

isNormalUser = true;

extraGroups = [

"audio"

"networkmanager"

"systemd-journal"

"vboxusers"

"video"

"wheel"

];

};

users.users.charles = {

isNormalUser = true;

home = "/home/charles";

description = "charles Foobar";

extraGroups = [ "wheel" "networkmanager" ];

openssh.authorizedKeys.keys = [ "ssh-dss AAAAB3Nza... charles@foobar" ];

};

# This value determines the NixOS release with which your system is to be

# compatible, in order to avoid breaking some software such as database

# servers. You should change this only after NixOS release notes say you

# should.

system.stateVersion = "18.09"; # Did you read the comment?