Ansible

Velkommen!

- Ukentlige programmeringskvelder
- MAPS og Svett bytter på å arrangere
- Alt vi synes er interessant
- Har du et morsomt prosjekt? Vis oss!

Min Ansible-bakgrunn

- Har jobbet med Ansible i 8. måneder
- Ikke ekspert, men ikke lenger nybegynner

Oppgaven min

- Finne en robust, repeterbar måte å rulle ut kode til serverne våre (*application deployment)*

I dag

- Vi bruker Ansible til å rulle ut kode (*application deployment)*

- Vi bruker Ansible til å konfigurere og vedlikeholde serverne våre (*configuration management)*

- Vi bruker Ansible, i samspill med Vagrant, til å standardisere utviklermiljøene våre

Application Deployment

- Hvordan rulle ut ny funksjonalitet på en sikker, rask og repeterbar måte?

- Sende ulike ting til ulike servere (*Staging* versus *Production)*

- Verktøy: Fabric, Capistrano

Configuration Management

- Nøkkelbegrep: *integritet*
- Er programvaren og støttebibliotekene riktig installert?
- Er alle konfigurasjoner slik de skal være? Nginx, brannmurer, brukeraksess+++
- Verktøy: Puppet, Chef

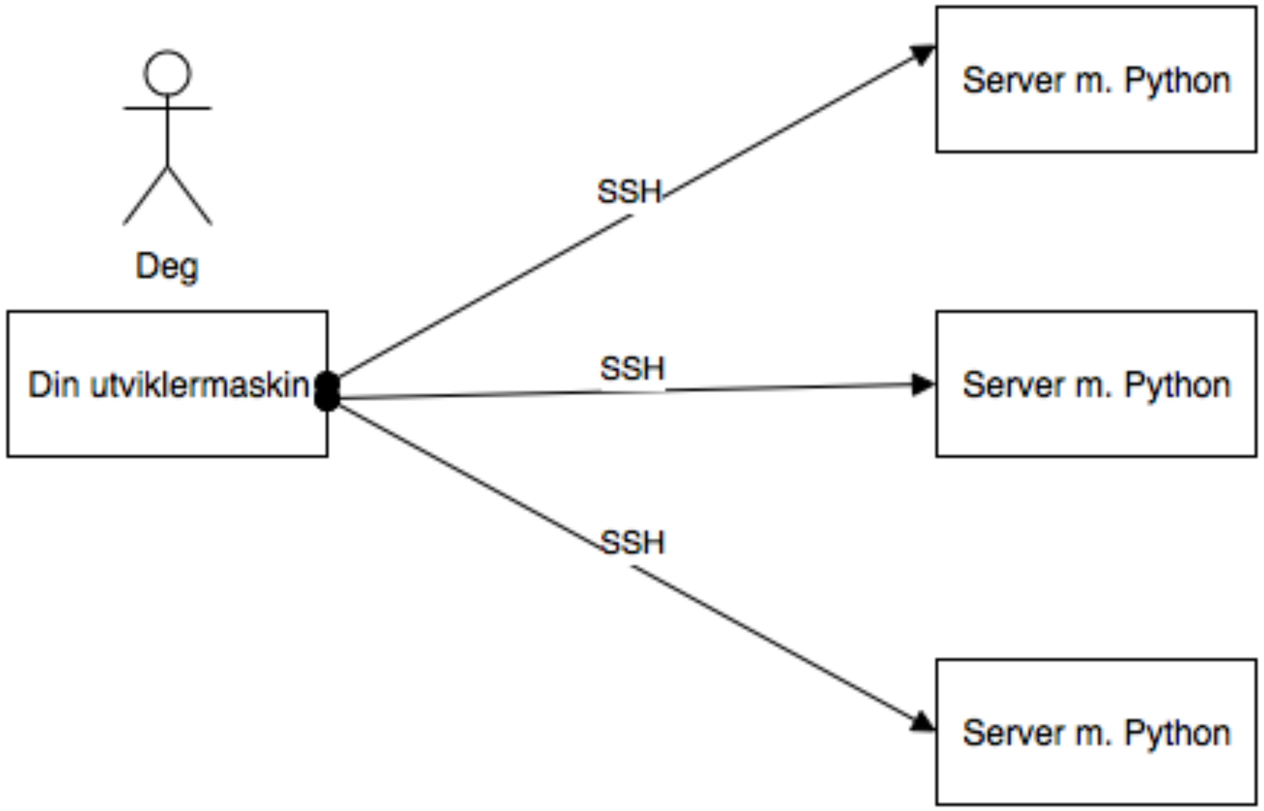
“Your servers should be cattle, not pets”

The pitch

- Ansible er *configuration management* og *application deployment* i ett og samme verktøy
- Ble startet i 2012
- Utviklet av Michael Dehaan og andre i Red Hat-miljøet
- Skrevet i Python
- Tilgjengelig på alle *NIX (noe Windows-støtte også)
- Navnet “Ansible” kommer fra Science Fiction

Tre søyler

- Python
- SSH
- UNIX



Deg
Din utviklermaskin
SSH
SSH
SSH
Server m. Python
Server m. Python
Server m. Python

Hello World

Nøkkelkomponenter

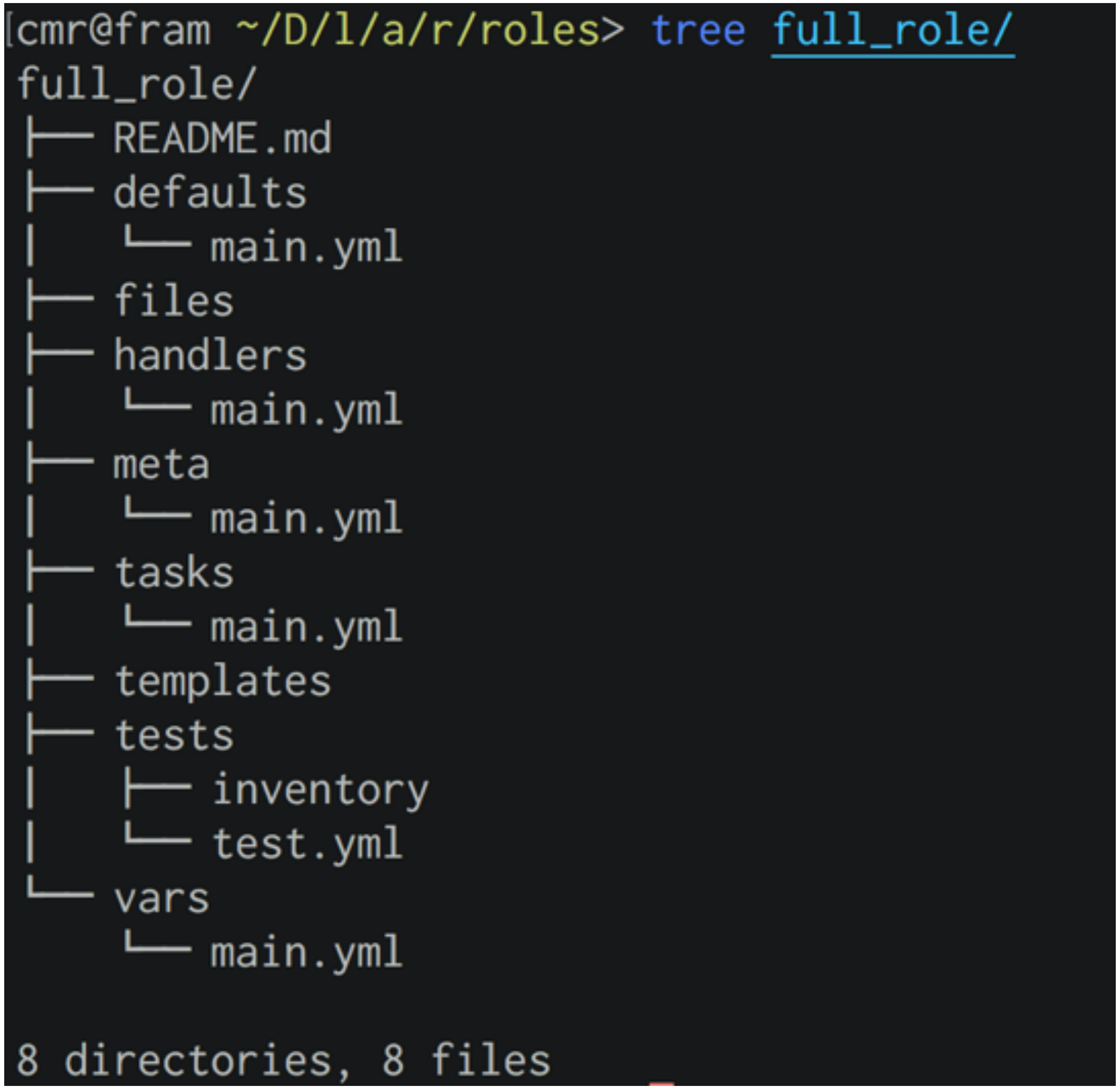
- Inventory
- ansible.cfg
- playbooks
- roller

Roller

Roller

- Den viktigste abstraksjonsmekanismen i Ansible
- Lar deg lage gjenbrukbare moduler
- Samler oppgaver, avhengigheter, default-verdier o.l. i én ting.

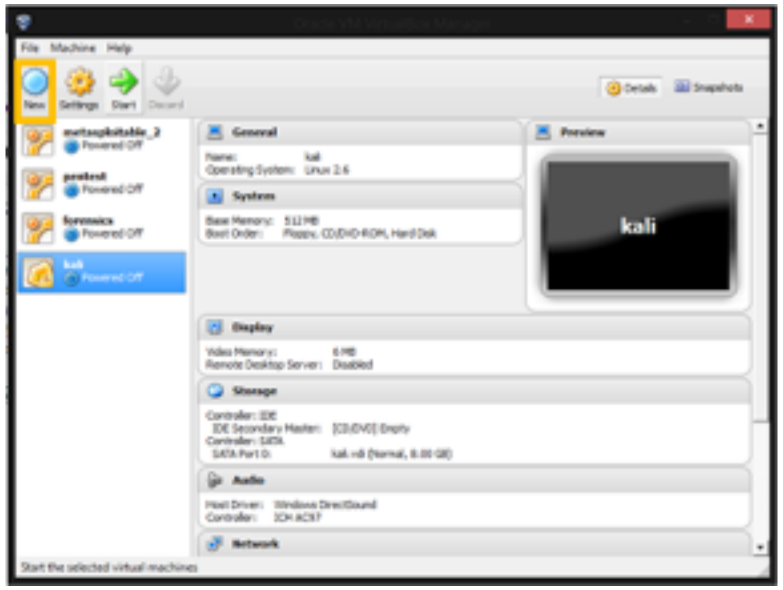
Mappestrukturen til en rolle



```
[cmr@fram ~/D/l/a/r/roles> tree full_role/
full_role/
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 8 files
```

- Det hadde vært fint å teste lokalt! Kanskje jeg skal prøve meg med en virtuell maskin…



General

Name: kali
Operating System: Linux 2.6

System

Base Memory: 512 MB
Boot Order: Floppy, CD/DVD-ROM, Hard Disk

Preview

Display

Video Memory: 6 MB
Remote Desktop Server: Disabled

Storage

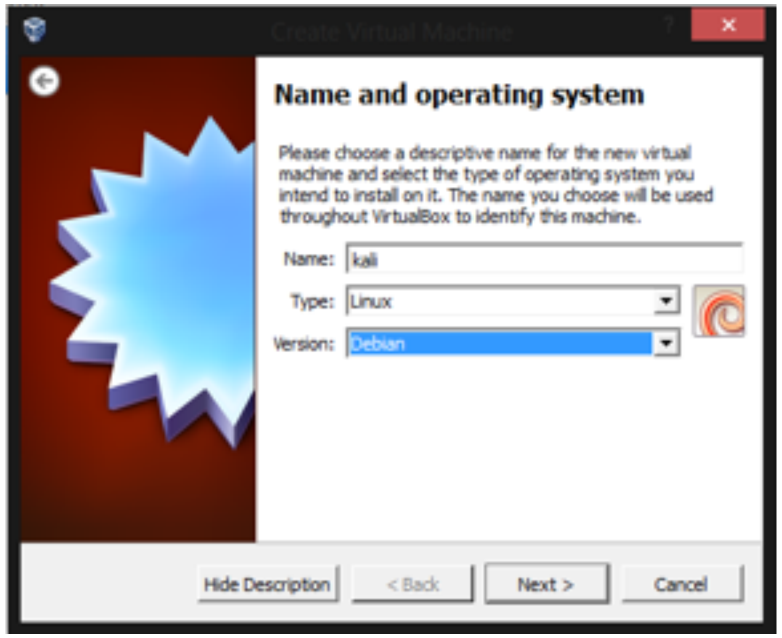
Controller: IDE
IDE Secondary Master: [CD/DVD] Empty
Controller: SATA
SATA Port 0: kali.vdi (Normal, 8.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

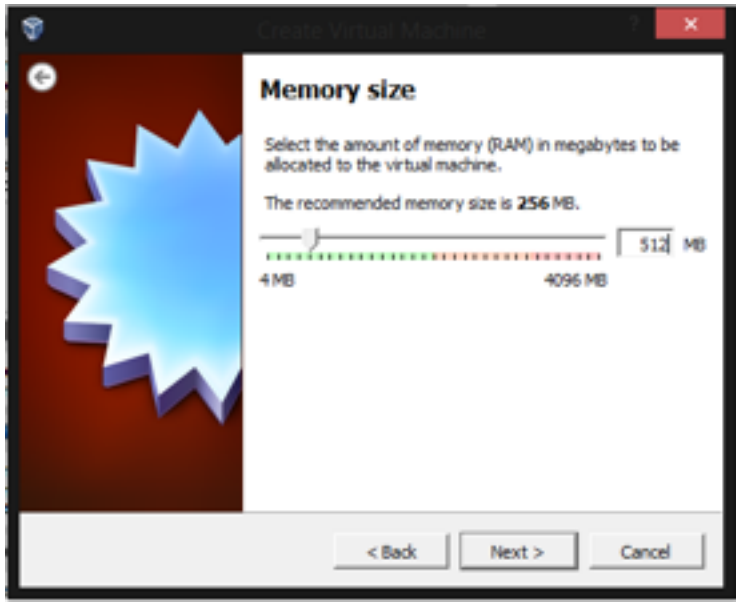
Start the selected virtual machines



Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: kali
Type: Linux
Version: Debian

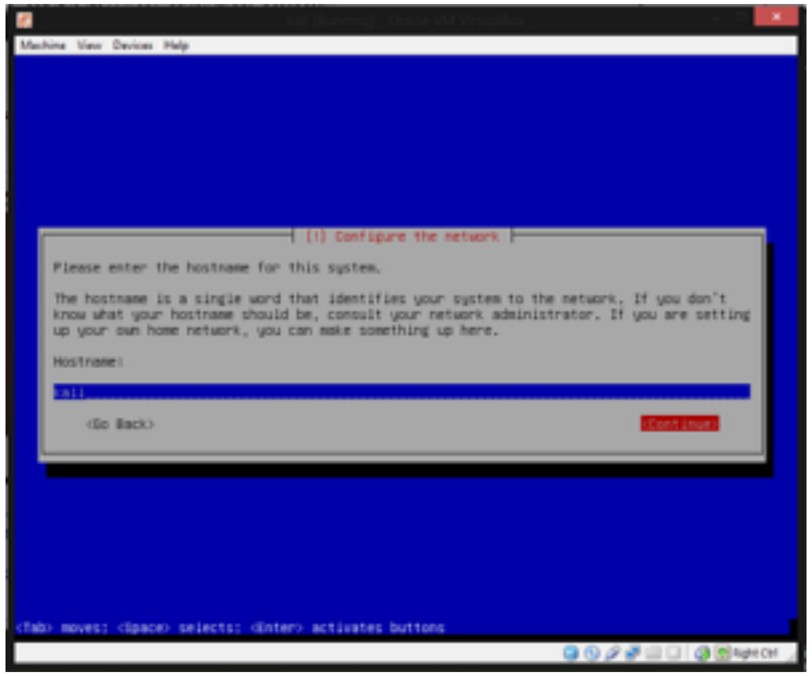


Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **256** MB.

512 MB



```
[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't
know what your hostname should be, consult your network administrator. If you are setting
up your own home network, you can make something up here.

Hostname:

kali

    <Go Back>                                                              <Continue>

<Tab> moves; <Space> selects; <Enter> activates buttons
```

AIN'T NOBODY GOT
TIME FOR THAT

Vagrant to the rescue!

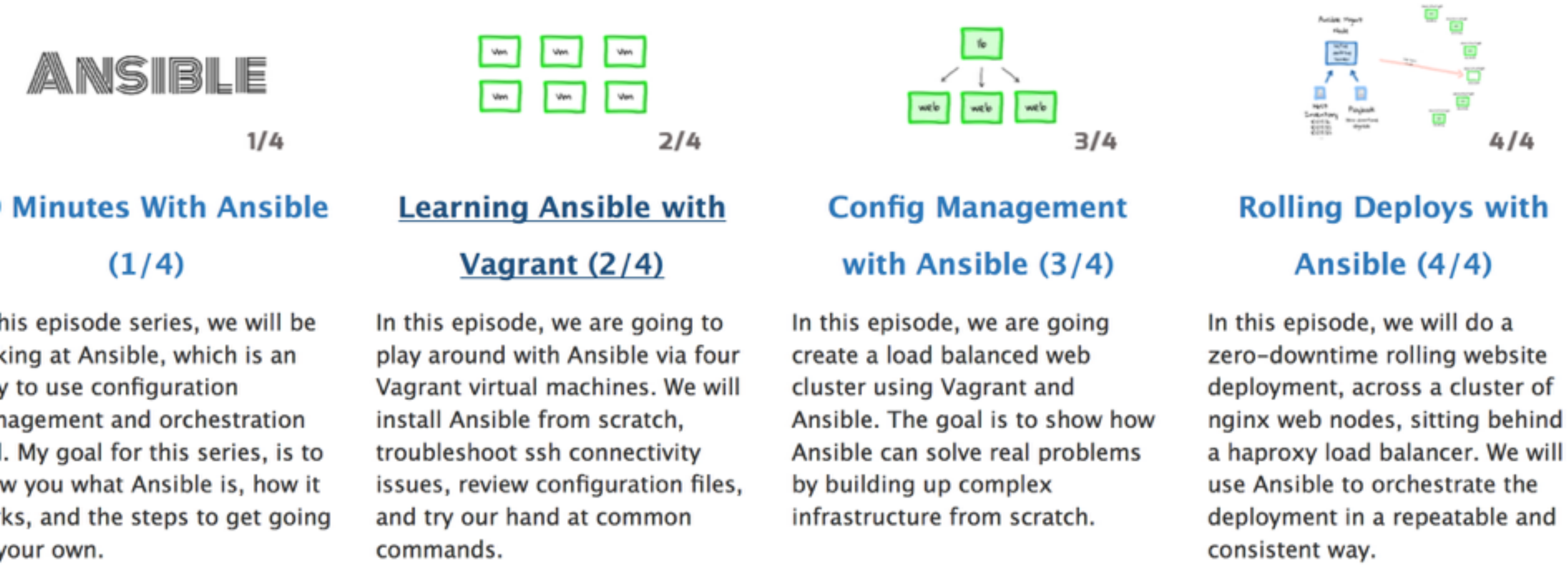
- Gjør det kjempelett å bruke Virtuelle maskiner!
- Gir deg en meny av ferdiglagede, minimale virtuelle maskiner
- Fikser ssh, delte mapper, private nettverk og mye mer.
- www.vagrantup.com
- https://atlas.hashicorp.com/boxes/search

- vagrant up
- vagrant ssh
- vagrant provision
- vagrant destroy

Vagrant + Ansible

- Standard provisioner
- ansible_local
- Privat nettverk

Ting å sjekke ut



Minutes With Ansible (1/4)

his episode series, we will be king at Ansible, which is an y to use configuration nagement and orchestration l. My goal for this series, is to w you what Ansible is, how it ks, and the steps to get going your own.

Learning Ansible with Vagrant (2/4)

In this episode, we are going to play around with Ansible via four Vagrant virtual machines. We will install Ansible from scratch, troubleshoot ssh connectivity issues, review configuration files, and try our hand at common commands.

Config Management with Ansible (3/4)

In this episode, we are going create a load balanced web cluster using Vagrant and Ansible. The goal is to show how Ansible can solve real problems by building up complex infrastructure from scratch.

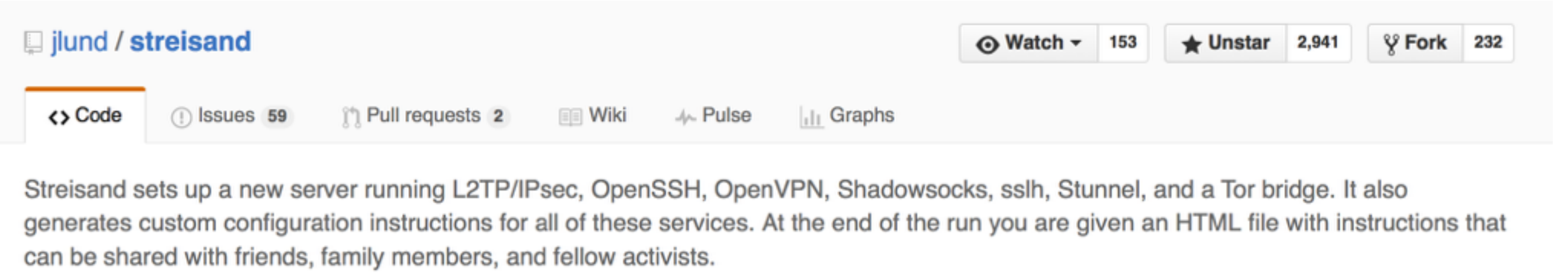
Rolling Deploys with Ansible (4/4)

In this episode, we will do a zero-downtime rolling website deployment, across a cluster of nginx web nodes, sitting behind a haproxy load balancer. We will use Ansible to orchestrate the deployment in a repeatable and consistent way.

https://sysadmincasts.com/

Ting å sjekke ut

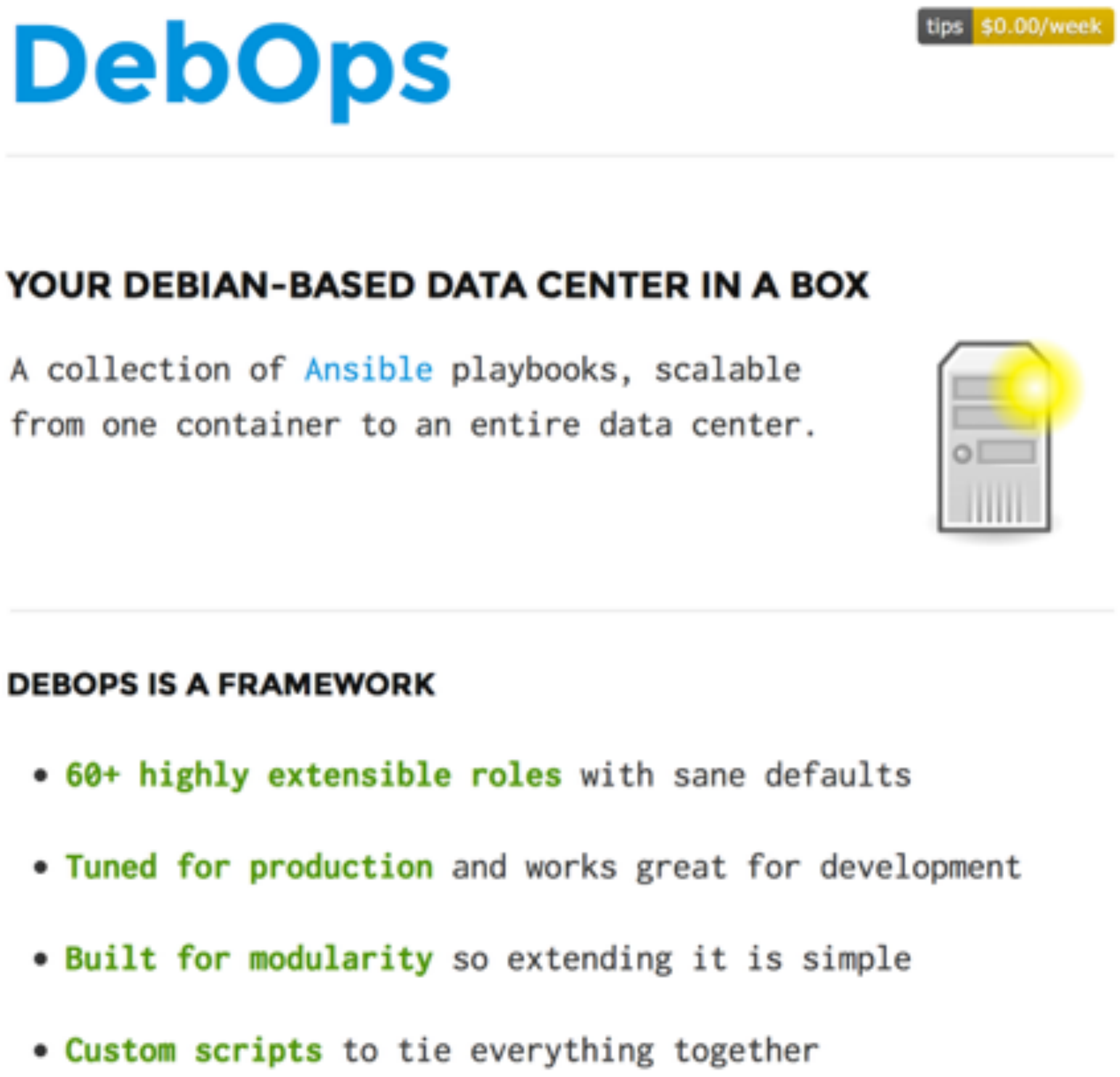
Ting å sjekke ut



Streisand sets up a new server running L2TP/IPsec, OpenSSH, OpenVPN, Shadowsocks, sslh, Stunnel, and a Tor bridge. It also generates custom configuration instructions for all of these services. At the end of the run you are given an HTML file with instructions that can be shared with friends, family members, and fellow activists.

https://github.com/jlund/streisand

Ting å sjekke ut



DebOps

YOUR DEBIAN-BASED DATA CENTER IN A BOX

A collection of Ansible playbooks, scalable from one container to an entire data center.

DEBOPS IS A FRAMEWORK

- **60+ highly extensible roles** with sane defaults
- **Tuned for production** and works great for development
- **Built for modularity** so extending it is simple
- **Custom scripts** to tie everything together

http://debops.org/

- docs.ansible.com
- vagrantup.com
- github.com/cmrosenberg/setup