



GELİŞTİRME ORTAMI OLARAK  
VAGRANT KULLANMAK

# EMİR KARŞIYAKALI

**LEAD DEVELOPER AT UBIT**

Sr. Goygoyer at **GoyGoy-TR**

**emirkarsiyakali@gmail.com**

github.com/**emir**

@**EmirKarsiyakali**

tr.linkedin.com/in/**emirkarsiyakali**

**emir.im**

# Nedir?

- Geliştirme, Yayın, Test ortamlarınızı yönetmeniz için geliştirilen bir araç.
- Mitchell Hasimoto tarafından geliştiriliyor. @**mitcellh**
- Sanal sunucularınızı yönetmeniz için biçilmiş kaftan.
- Ruby ile geliştiriliyor.
- VirtualBox, VMware gibi uygulamalar ile kullanılıyor.

Tařınabilir, yeniden kullanılabilir ve  
kolay yönetilebilir geliştirme  
ortamlarına sahip olmanıza yarayan  
araç.

# Niçin?

- Hızlı
- Kolayca sunucunuzun kopyasını oluşturabiliyorsunuz.
- Kolayca geliştirme ortamınızdan, sunucu ortamına çıkabiliyorsunuz.
- Yaygın, Yetişkin, Stabil

# İhtiyaç?

- Ekibinize yeni katılan birinin projeye adaptasyon sürecini hızlandırır.
- Başak burcu olanlar?
- Mac, Windows altında rahatlıkla çalışabilirsiniz.

# Kurulum

- [vagrantup.com](https://vagrantup.com) adresinden indiriliyor.
- Provider için VirtualBox [virtualbox.org](https://www.virtualbox.org) adresinden ücretsiz olarak indirilebilir.

# Sözlük

- Box
- Vagrantfile
- Vagrant CLI



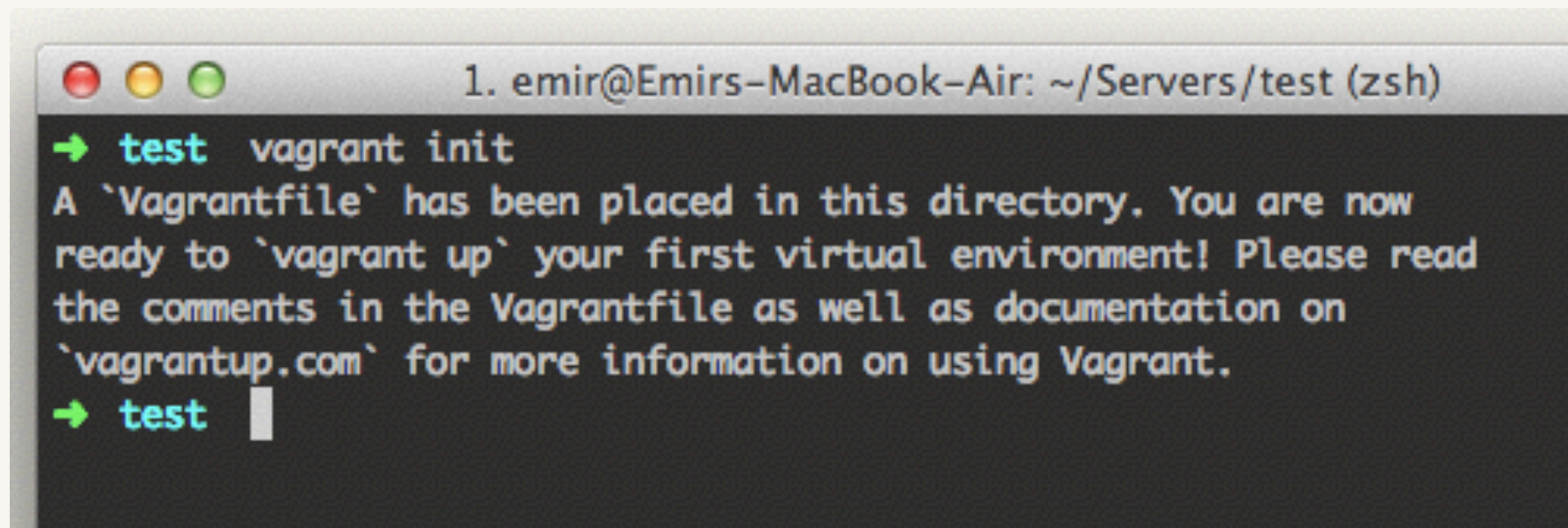
# Box

- [vagrantbox.es](http://vagrantbox.es)
- Debian, Ubuntu, OpenSuse, OpenBSD, CentOS...
- Provider



# Vagrantfile

- Vagrant için yönergeleri yazabileceğiniz dosyadır.

A screenshot of a terminal window on a Mac. The title bar shows the window name as '1. emir@Emirs-MacBook-Air: ~/Servers/test (zsh)'. The terminal content shows a green prompt character followed by the command 'test vagrant init'. The output is a multi-line message: 'A `Vagrantfile` has been placed in this directory. You are now ready to `vagrant up` your first virtual environment! Please read the comments in the Vagrantfile as well as documentation on `vagrantup.com` for more information on using Vagrant.' Below the output, the prompt 'test' is shown again with a cursor.

```
1. emir@Emirs-MacBook-Air: ~/Servers/test (zsh)
→ test vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
→ test
```

## Vagrantfile

```
1  # -*- mode: ruby -*-
2  # vi: set ft=ruby :
3
4  # Vagrantfile API/syntax version. Don't touch unless you know what you're doing!
5  VAGRANTFILE_API_VERSION = "2"
6
7  Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
8    # All Vagrant configuration is done here. The most common configuration
9    # options are documented and commented below. For a complete reference,
10    # please see the online documentation at vagrantup.com.
11
12    # Every Vagrant virtual environment requires a box to build off of.
13    config.vm.box = "base"
14
15    # The url from where the 'config.vm.box' box will be fetched if it
16    # doesn't already exist on the user's system.
17    config.vm.box_url = "http://domain.com/path/to/above.box"
18
19    # Create a forwarded port mapping which allows access to a specific port
20    # within the machine from a port on the host machine. In the example below,
21    # accessing "localhost:8080" will access port 80 on the guest machine.
22    config.vm.network :forwarded_port, guest: 80, host: 8080
23
24    # Create a private network, which allows host-only access to the machine
25    # using a specific IP.
26    config.vm.network :private_network, ip: "192.168.33.10"
27
28    # Create a public network, which generally matched to bridged network.
29    # Bridged networks make the machine appear as another physical device on
30    # your network.
31    # config.vm.network :public_network
32
33    # If true, then any SSH connections made will enable agent forwarding.
34    # Default value: false
35    # config.ssh.forward_agent = true
36
37    # Share an additional folder to the guest VM. The first argument is
38    # the path on the host to the actual folder. The second argument is
39    # the path on the guest to mount the folder. And the optional third
40    # argument is a set of non-required options.
41    config.vm.synced_folder "../data", "/vagrant_data"
42
43    # Provider-specific configuration so you can fine-tune various
44    # backing providers for Vagrant. These expose provider-specific options.
```

# Vagrant CLI

`vagrant up`: VM'i ayağa kaldırır.

`vagrant ssh`: VM'e ssh ile bağlanmanıza yarıyor.

`vagrant reload`: VM'inizi yeniden başlatır. (Vagrantfile ayarları değişirse yenisini okuyor.)

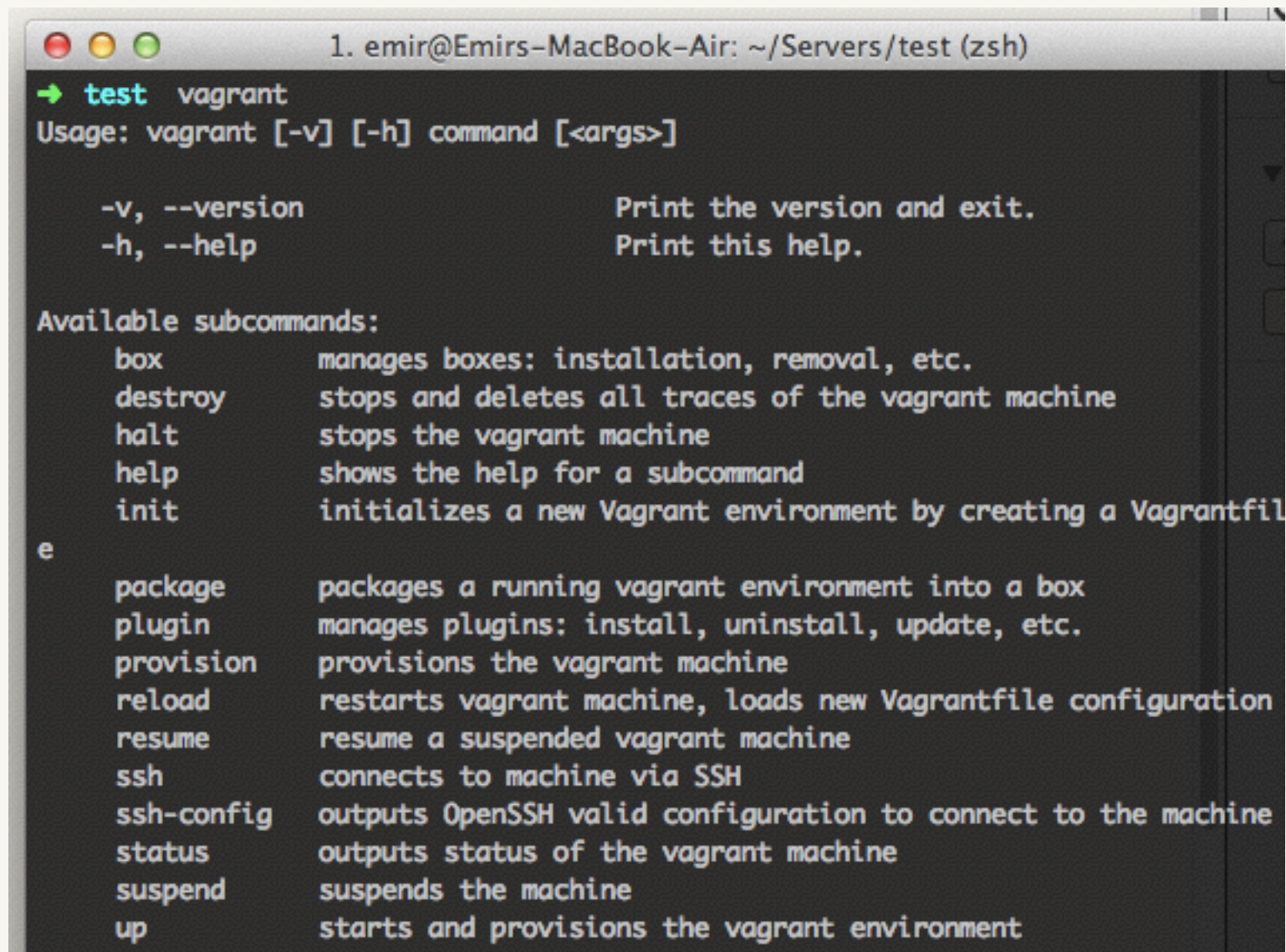
`vagrant status`: VM'inizin durumunu sorgulayabilirsiniz.

`vagrant destroy`: VM'inizi silebilirsiniz.

`vagrant halt`: VM'inizin kapatabilirsiniz.



# Vagrant CLI



```
1. emir@Emirs-MacBook-Air: ~/Servers/test (zsh)
→ test vagrant
Usage: vagrant [-v] [-h] command [<args>]

    -v, --version          Print the version and exit.
    -h, --help             Print this help.

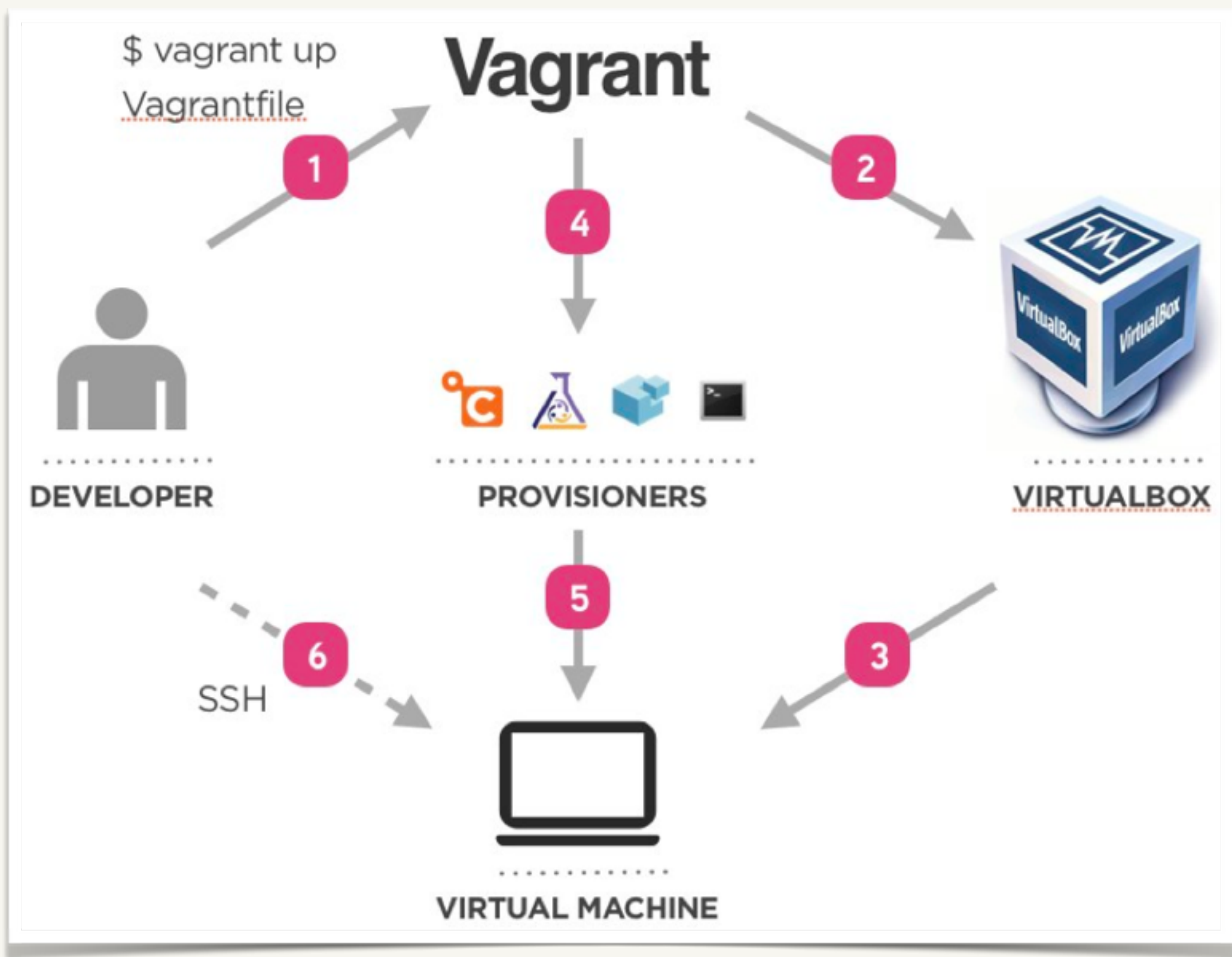
Available subcommands:
    box                    manages boxes: installation, removal, etc.
    destroy                stops and deletes all traces of the vagrant machine
    halt                   stops the vagrant machine
    help                   shows the help for a subcommand
    init                   initializes a new Vagrant environment by creating a Vagrantfile
    package                packages a running vagrant environment into a box
    plugin                 manages plugins: install, uninstall, update, etc.
    provision              provisions the vagrant machine
    reload                 restarts vagrant machine, loads new Vagrantfile configuration
    resume                 resume a suspended vagrant machine
    ssh                    connects to machine via SSH
    ssh-config              outputs OpenSSH valid configuration to connect to the machine
    status                 outputs status of the vagrant machine
    suspend                suspends the machine
    up                     starts and provisions the vagrant environment
```

# Providers

- VM's  
VirtualBox, VMware, Hyper-V, libvirt, KVM
- Containers:  
lxc, Docker
- Services:  
AWS, DigitalOcean, HP Cloud, Joyent, Rackspace,  
OpenStack, CloudStack, vSphere...

# Provisioning

- Paket kurulumları ve ayarlarının yapılmasını otomatize etmek için.
  - Shell Scripts
  - Puppet
  - Chef
  - Ansible
  - Salt





# Synced Folders

- /vagrant dizini ön tanımlı olarak geliyor.
- Vagrantfile'dan düzenlenebilir.

Mozilla

Disqus

O'REILLY

BBC

...

# Destek?

- GitHub mitchellh/vagrant
- IRC Freenode #vagrant
- HashiCorp  
[www.hashicorp.com](http://www.hashicorp.com)

# PuPHPet

- puphpet.com aracılığı ile Vagrantfile türettirebilir ve türettiğiniz bu Vagrantfile'in sunucu, veritabanı, dil ayarlarını PuPHPet arayüzü ile yapabilirsiniz.

**Sorularınız?**

**Teşekkürler!**