



**Отладка и валидация сложных
инфраструктурных решений с
помощью Vagrant**



Что умеет Vagrant, как инструмент?

- Запускать виртуальные машины (VM) на базе разных систем виртуализации?
- Настраивать сетевую инфраструктуру виртуального окружения?
- Выполнять настройку запущенных VM?
- ...
- Да, но... из текстового файла-описателя.



Для каких задач предназначен этот инструмент?

- Разработка и отладка
 - Web-решения
 - Клиент-серверные решения
- Воспроизведение сборочных окружений
- Развёртывание в облачных решений
 - Развёртывание решений в частных и публичных IaaS-облаках



Подготовка: Vagrantfile

```
[sin@tor basealt-workstation]$
```

```
$ grep -v -e "^\s\+#" -e "^\#" -e "^\$" Vagrantfile
```

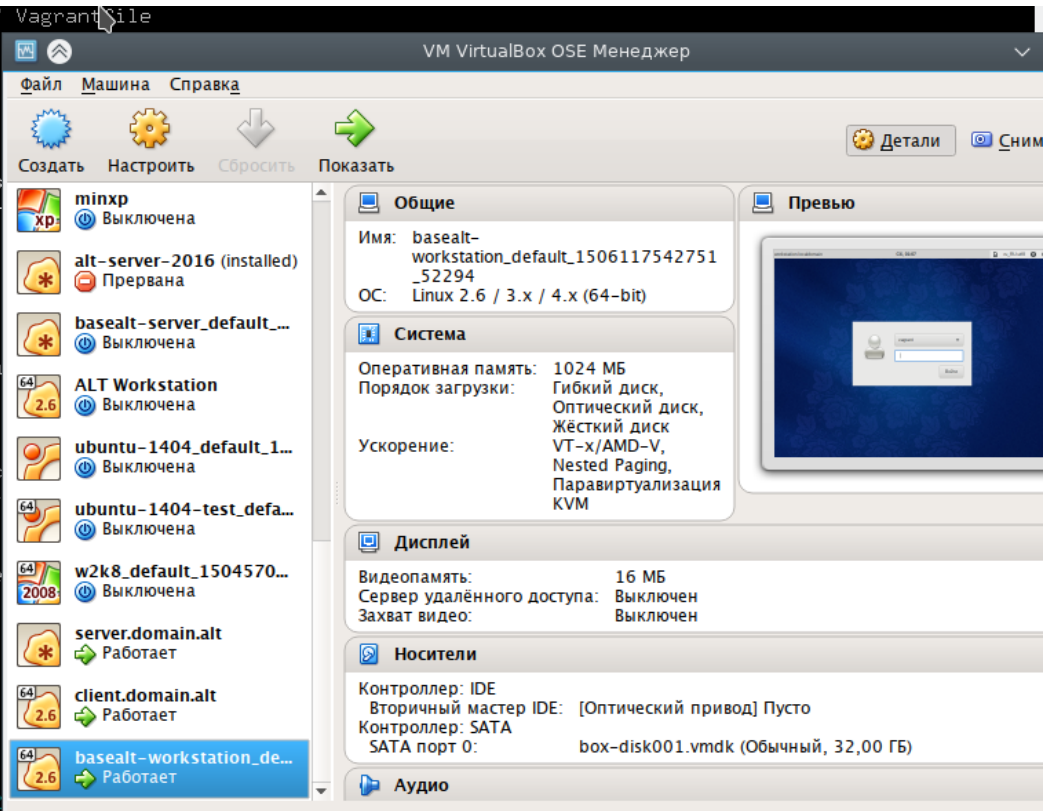
```
Vagrant.configure("2") do |config|  
  config.vm.box = "mastersin/basealt-p8-workstation"  
  config.vm.network "public_network", bridge: "eth0"  
  config.vm.hostname = "wsy.darkmastersin.net"
```

```
End
```




Запуск виртуальной машины

```
[sin@tor basealt-workstation]$ grep -v -e "^\s\+#" -e "^#" -e "^\$" Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "mastersin/basealt-p8-workstation"
  config.vm.network "public_network", bridge: "eth0"
  config.vm.hostname = "wsy.darkmastersin.net"
end
[sin@tor basealt-workstation]$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'mastersin/basealt-p8-workstation' is
==> default: Fixed port collision for 22 => 2222. Now on port 2201
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration..
default: Adapter 1: nat
default: Adapter 2: bridged
==> default: Forwarding ports...
default: 22 (guest) => 2201 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minu
default: SSH address: 127.0.0.1:2201
default: SSH username: vagrant
default: SSH auth method: private key
default:
default: Vagrant insecure key detected. Vagrant will automatic
default: this with a newly generated keypair for better securi
default:
default: Inserting generated public key within guest...
default: Removing insecure key from the guest if it's present.
default: Key inserted! Disconnecting and reconnecting using ne
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Setting hostname...
==> default: Configuring and enabling network interfaces...
default: SSH address: 127.0.0.1:2201
default: SSH username: vagrant
default: SSH auth method: private key
==> default: Mounting shared folders...
default: /vagrant => /home/sin/vagrant/basealt-workstation
[sin@tor basealt-workstation]$
tor[0]: 4 0
```



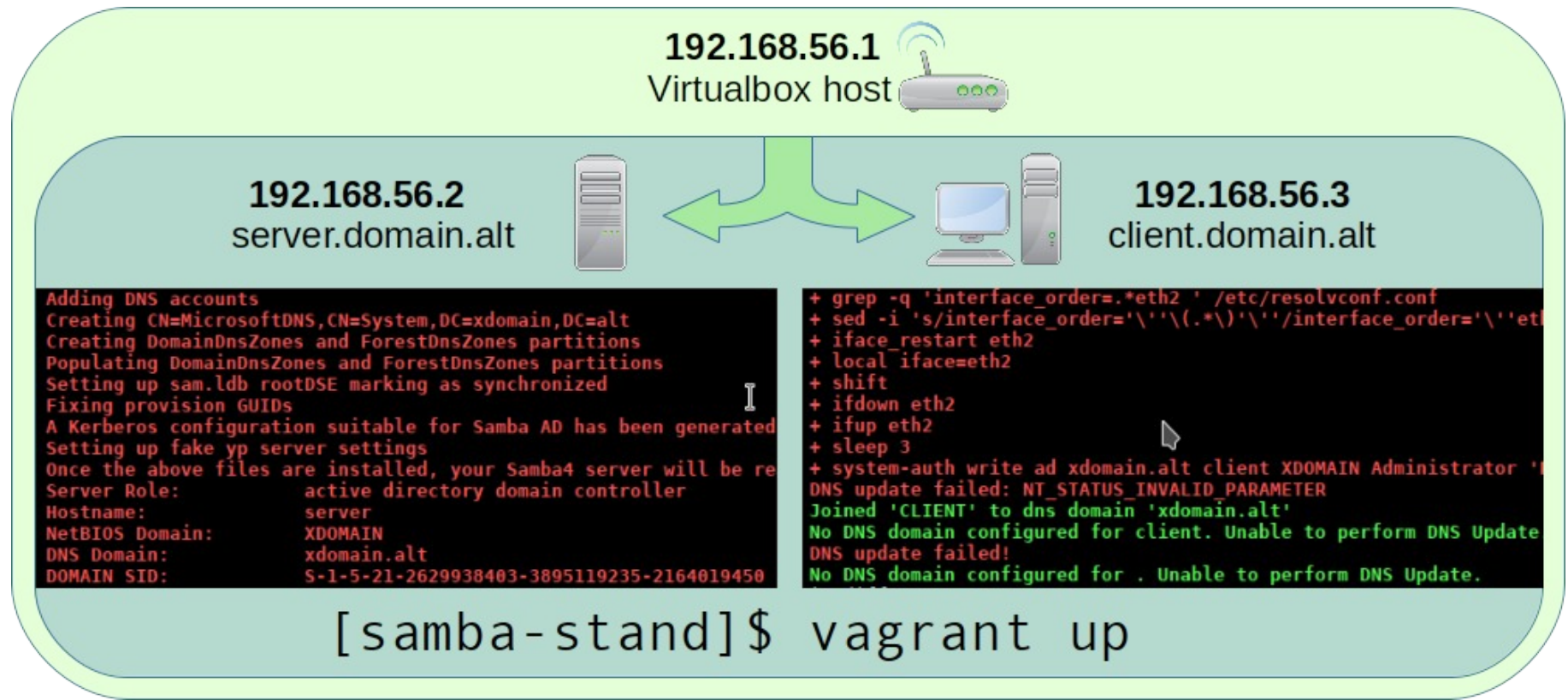
\$ vagrant up



- Отладка SambaDC
- Настройка сети в Vagrant
- Подготовка и настройка сети для AD-инфраструктуры на базе Samba
- Сетевые настройки на Samba-сервере и на Samba-клиенте
- Следующие шаги по отладке...



Отладка SambaDC





Настройка сети в Vagrant

```
if (machine[:ip] == "dhcp")
    node.vm.network "public_network", bridge: 'eth0'
else
    node.vm.network "public_network", ip: machine[:ip], netmask: 24,
    bridge: 'eth0'
end

if (machine[:ip_int] == "dhcp")
    node.vm.network "private_network", virtualbox__intnet: "intnet"
else
    node.vm.network "private_network", ip: machine[:ip_int], netmask:
    24, virtualbox__intnet: "intnet"
end
```




Сетевые настройки на Samba-сервере

- Статические IP, привязанные к DNS-домену
- localhost, как nameserver
- FSMO-роли и DNS



Настройка сервера

```
apt-get install samba-DC samba-DC-client bind-utils krb5-  
kinit ldb-tools
```

```
rm -f /etc/samba/smb.conf  
init_krb5_conf "$REALM"
```

```
samba-tool domain provision --realm="$REALM" --domain  
"$WORKGROUP" --adminpass="$PASSWORD" --dns-  
backend=SAMBA_INTERNAL --server-role=dc --use-rfc2307 --  
host-ip="$host_ip"
```

```
disable_dhcpd_resolvconf_hook  
set_etc_hosts "$host_ip" "$host_name"  
update_smb_conf  
set_nameserver 127.0.0.1 "$DOMAIN"  
service samba start
```



Сетевые настройки на Samba-клиенте

- Настройки DNS-серверов через DHCP
 - Несколько интерфейсов
 - Настройки openresolv
 - Настройки NetworkManager
 - Список поиска узлов «своего» доменов
- DHCP и обновление DNS-записей



Настройка клиента

```
apt-get install samba-client sssd-ad bind-utils krb5-kinit  
ldb-tools task-auth-ad-sssd
```

```
init_krb5_conf "$REALM"  
if [ -n "$host_nameserver" ]; then  
    disable_networkmanager_dns  
    disable_dhcpd_resolvconf_hook  
    set_nameserver "$host_nameserver" "$DOMAIN"  
fi
```

```
system-auth write ad $DOMAIN $HOST $WORKGROUP  
'Administrator' "$PASSWORD"
```

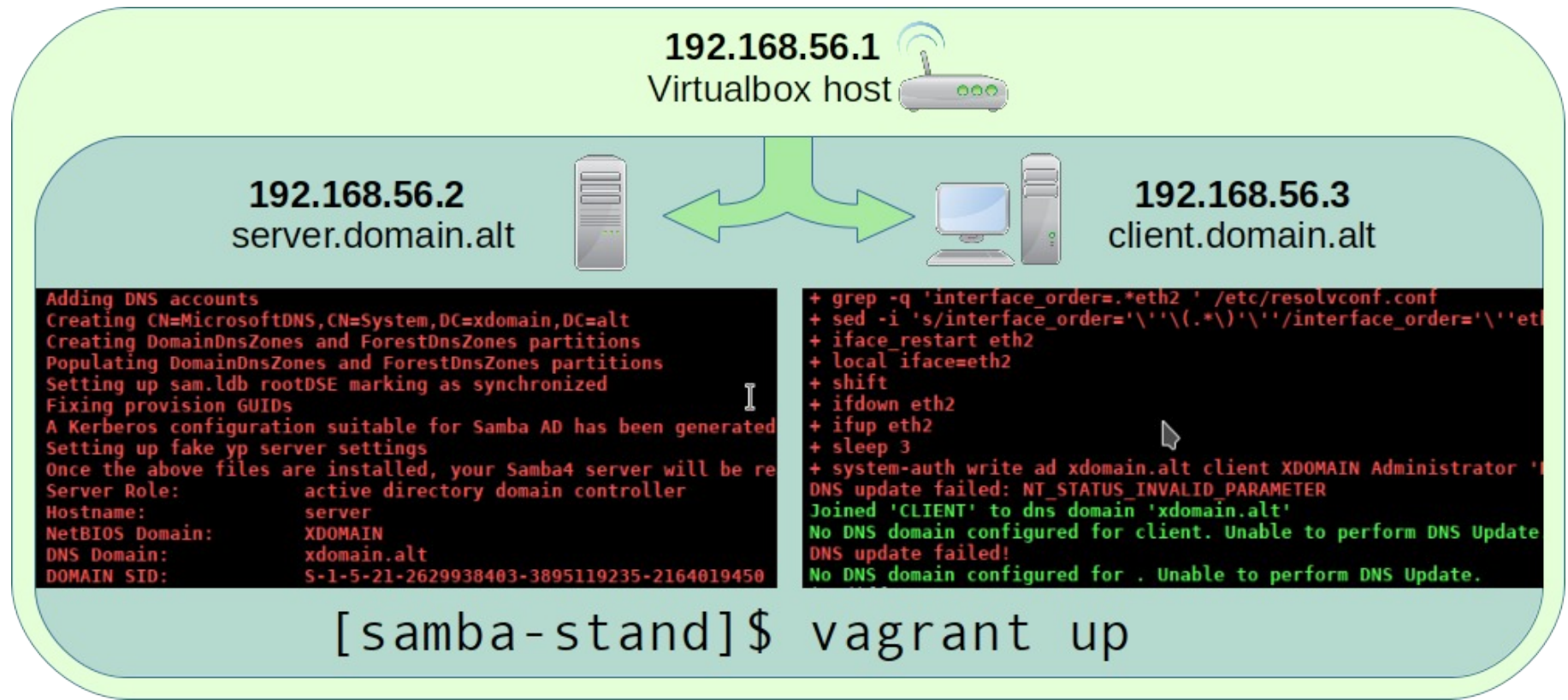



Настройки для AD-инфраструктуры

- DNS
- Kerberos
- NTP
- Samba ± Winbind
- SSSD = PAM + NSS



Отладка SambaDC





Следующие шаги по отладке...

- Репликация домена — включение контроллера в уже созданный ранее лес
- Создание узлов с выделенными службами:
 - Файловый сервер
 - Web-служба с поддержкой SSO
 - SSH + GSSAPI
- Подготовка и проверка клиентов



Заключение

Samba stand with Vagrant:

<http://github.com/mastersin/samba-stand>

Vagrant+ALT - <https://app.vagrantup.com/mastersin/>

- **mastersin/basealt-p8-server** — сервер на базе SysVinit.
- **mastersin/basealt-p8-server-systemd** — сервер на базе Systemd.
- **mastersin/basealt-p8-workstation** — клиент на базе Systemd и NetworkManager.