

10th of March - Thursday:

P81 Sometimes it's alright to be lazy (Tools like Vagrant & Packer.io)

5:00 PM – 5:45 PM: Auditorium 1

P5 Oracle 12c New Features for Java-developers

6:00 PM – 6:45 PM: Congress 3

11th of March - Friday:

P49 10 mistakes your (Java) developers will do - if you don't assist 'em!

5:00 PM – 5:45 PM: Parliament 1+2

“Sometimes
It’s alright
To be ...

LAZY



EVRY

The best system administrators and developers I know
are downright ...



LAZY

Managing Personal Test Databases with Tools like ...

Oracle VirtualBox & Vagrant

By Lasse Jenssen

~~... & Packer.io~~
... & FlywayDB

“Sometimes it’s alright to be LAZY”

EVRY

Managing Personal Test Databases with Tools like ...

Oracle VirtualBox & Vagrant

By Lasse Jensen

**... & SALT
... & FlywayDB**

“Sometimes it’s alright to be LAZY”

EVRY

Safe Harbor Statement

As most of the OUGN16 attendees the author of the following presentation, was out enjoying a beer or two last night. Every single word and print will truly be presented with the best intentions for the audience. Regardless of this, there is no guarantee that there won't be any errors or mistakes in the following content.

Agenda:

- Introduction to Vagrant (& Virtualbox)
- How to get going?
- My small Oracle Vagrant Framework
- How to do ...
 - testing with snapshots & rollback
 - clean up your schema
 - create new databases
 - drop old databases
- Example: SALT to create Vagrant Box
- Example: Build schema w/flywayDB

Use-Cases:

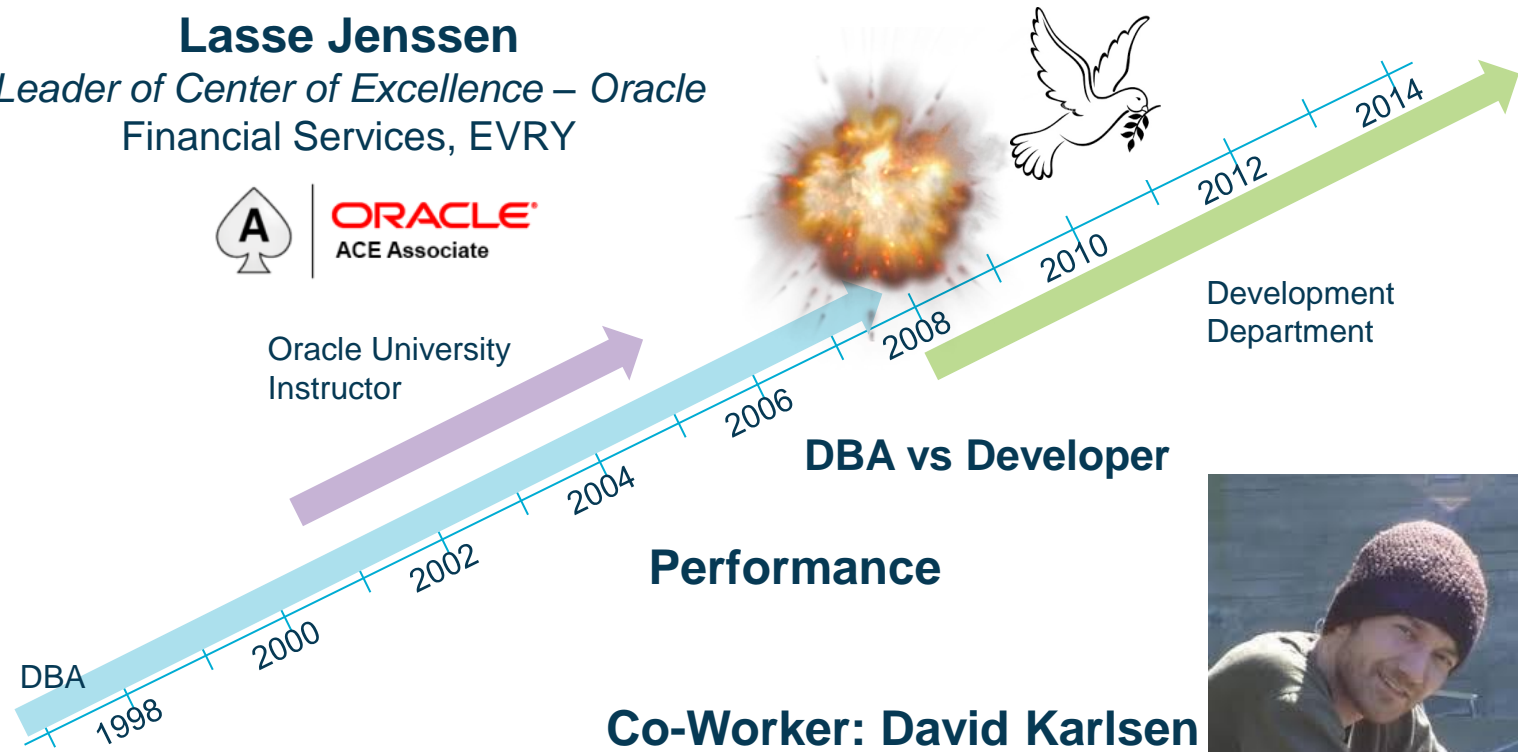
- Administering your **personal** test databases
- Providing an production like platform for you **developer** (on developers local laptop)
- Provide a testing platform to be integrated with **Jenkins** for automated testing
- Easy way to deploy test/development databases to your VMware environment (No provider for Oracle VM yet)

<https://github.com/lasjen/OracleVagrantFramework>

Lasse Jenssen
*Leader of Center of Excellence – Oracle
Financial Services, EVRY*



Oracle University
Instructor



Oracle End-to-end Metrics

DBA vs Developer

Performance

Co-Worker: David Karlsen
Leader of Center of Excellence – Java





Oracle VirtualBox

a hypervisor for x86 computers

freely available

Open Source Software under the terms of the
GNU General Public License (GPL) version 2

runs on Windows, Linux, Macintosh, and Solaris hosts

a large number of guest operating systems

including but not limited to

Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8),
DOS/Windows 3.x, Linux (2.4, 2.6 and 3.x), Solaris and OpenSolaris, OS/2, and OpenBSD.




[Details](#) [Snapshots](#)

64 WinCorp8
8:1 Running

64 oracle
Powered Off

64 OEL65-ORA11G-P...
Aborted

64 OEL65-ORA11G-P...
Aborted

64 OEL66-DOCKER-D...
Powered Off

64 OEL65-ORA11G-P...
Aborted

64 OEL65-ORA12C-P...
Powered Off

General

Name: WinCorp8
Operating System: Windows 8.1 (64-bit)

System

Base Memory: 6068 MB
Processors: 4
Boot Order: Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging

Display

Video Memory: 128 MB
Acceleration: 2D Video
Remote Desktop Server: Disabled
Video Capture: Disabled

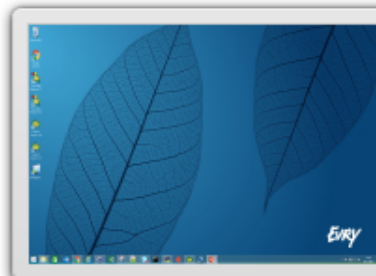
Storage

Controller: IDE
IDE Secondary Master: [CD/DVD] Empty
Controller: SATA
SATA Port 0: WinCorp8-disk2.vmdk (Normal, 100,00 GB)

Audio

Host Driver: PulseAudio

Preview





Vagrant



It's magic!

... running on Windows, Linux & Macintosh hosts

... working on the shoulder of giants

(like VirtualBox, VMware, Amazon WS, Google CE or any other provider)



Discover Vagrant Boxes

This page lets you discover and use Vagrant Boxes created by the community. You can search by operating system, architecture or provider.

Provider filter

Sort by



[kungfuice/ubuntu64-java](#) Ubuntu 14.04.2 LTS, Oracle Java 7, MySQL

801 downloads | 1.0 | last release 8 months ago



[abogatyrev/centos7-nginx-mysql-wildfly8.box](#)
CentOS 7 box with ansible, nginx, mysql, Oracle jdk8-8u65x64 and Wildfly8.2.1

162 downloads | 1.0.2 | last release 18 days ago

EVERY

```
$ mkdir ubuntu64_java
$ cd ubuntu64_java/
$ vagrant init kungfuice/ubuntu64-java
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! ...
```

```
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'kungfuice/ubuntu64-java'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'kungfuice/ubuntu64-java' is up to date...
==> default: Setting the name of the VM:
ubuntu64_java_default_1448663249906_18214
==> default: Fixed port collision for 22 => 2222. Now on port 2200.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 => 2200 (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes..
```

...

EVERY

```
$ vagrant ssh
```

```
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)
```

```
System information as of Fri Nov 27 17:29:54 EST 2015
```

```
System load: 0.47          Processes:          106  
Usage of /: 5.6% of 35.07GB Users logged in: 0  
Memory usage: 3%          IP address for eth0: 10.0.2.15  
Swap usage: 0%
```

```
Last login: Tue Mar 3 17:29:37 2015
```

```
vagrant@vagrant:~$ hostname
```

```
vagrant
```

```
vagrant@vagrant:~$ exit
```

```
logout
```

```
Connection to 127.0.0.1 closed.
```

```
$ hostname
```

```
ws-coe-ek2046
```

```
$ ssh vagrant@localhost -p2220
```

```
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)
```

```
...
```

The logo for 'EVERY' is written in a stylized, handwritten font. The letters are dark blue or black, with a slightly irregular, sketchy appearance. The 'E' and 'V' are particularly prominent, with the 'V' having a sharp, pointed bottom. The overall style is casual and modern.

What have I done to get this far?

Configured BIOS for Virtualization

Installed Oracle VirtualBox

Installed Vagrant

PhoenixBIOS Setup Utility

Main Advanced Security Power Boot Exit

System Time:	[08]:03:32	Item Specific Help <Tab>, <Shift-Tab>, or <Enter> selects field.
System Date:	[05/05/2008]	
Legacy Diskette A:	[1.44/1.25 MB 3 1/2"]	
Legacy Diskette B:	[Disabled]	
▶ Primary Master	[UMAware U]	
▶ Primary Slave	[None]	
▶ Secondary Master	[UMAware U]	
▶ Secondary Slave	[None]	
▶ Keyboard Features		
System Memory:	640 KB	
Extended Memory:	261120 KB	
Boot-time Diagnostic Screen:	[Enabled]	

F1 Help F1 Select Item -/+ Change
Esc Exit ++ Select Menu Enter Select

Virtualization

Intel (R) Virtualization Technology [Disabled]

Intel (R) VT-d Feature [Disabled]

Disabled
Enabled



Welcome Lasse

Account Sign Out Help Country ▾ Communities ▾ I am a... ▾ I want to... ▾

[Products](#) [Solutions](#) [Downloads](#) [Store](#) [Support](#) [Training](#) [Partners](#) [About](#) [OTN](#)

Oracle Technology Network > Server and Storage Systems > Oracle VM VirtualBox > Downloads

- Engineered Systems
- Ksplice
- Oracle Linux
- Oracle Optimized Solutions
- Oracle VM
- Oracle VM VirtualBox
- SAN Storage
- Secure Global Desktop
- Server Management Tools
- Solaris 10
- Solaris 11
- Solaris Cluster
- Solaris Studio
- SPARC Servers
- StorageTek Tape Storage

[Overview](#) [Downloads](#) [Documentation](#) [Community](#) [Support](#) [Learn More](#)

Oracle VM VirtualBox

The latest release is version 5.0.6.

On this page you can download:

- [Oracle VM VirtualBox](#)
- [Oracle VM VirtualBox Extension Pack](#)
- [Oracle VM VirtualBox Source Code](#)
- [Oracle VM VirtualBox Pre-built Appliances](#)

ORACLE OPEN WORLD
Junho 23-25, 2015
São Paulo, Brasil
#ooowbr
[Inscreva-se](#)





DOWNLOAD

Latest

Old Versions

DOWNLOAD VAGRANT

Below are all available downloads for the latest version of Vagrant (1.7.4). Please download the proper package for your operating system and architecture. You can find SHA256 checksums for packages [here](#), and you can find the version changelog [here](#).

 **MAC OS X**
[Universal \(32 and 64-bit\)](#)

 **WINDOWS**
[Universal \(32 and 64-bit\)](#)

 **LINUX (DEB)**
[32-bit](#) | [64-bit](#)

 **LINUX (RPM)**
[32-bit](#) | [64-bit](#)



But we need to build our own boxes



package VM into



The screenshot shows the Oracle VM VirtualBox Manager interface. On the left, a list of virtual machines is visible, including 'WinCorp8 Running', 'oracle Powered Off', and several 'OEL65-ORA11G-PW...' and 'OEL65-ORA12C-PW...' machines. The main window displays the 'Create Virtual Machine' dialog box. The 'Name and operating system' section is active, showing a 'Name' field with 'OEL67-BASE', a 'Type' dropdown set to 'Linux', and a 'Version' dropdown set to 'Oracle (64-bit)'. The 'General' tab is selected, showing details for 'WinCorp8' (Operating System: Windows 8.1 (64-bit)).

```
> Vagrant package -base OEL67-BASE --output oel67-base.box
```

oel67-base.box



oel67-ora11g.box

oel67-ora12c.box

oel67-ora12c2.box

Hold on!
That is a lot of ...
WORK.

What happened to LAZY???

EVRY



Welcome Lasse

Account Sign Out Help Country ▾ Communities ▾ I am a... ▾ I want to... ▾

Products Solutions Downloads Store Support Training Partners About OTN

Oracle Technology Network > Server and Storage Systems > Oracle VM VirtualBox > Downloads

- Engineered Systems
- Ksplice
- Oracle Linux
- Oracle Optimized Solutions
- Oracle VM
- Oracle VM VirtualBox
- SAN Storage
- Secure Global Desktop
- Server Management Tools
- Solaris 10
- Solaris 11
- Solaris Cluster
- Solaris Studio
- SPARC Servers
- StorageTek Tape Storage

Overview **Downloads** Documentation Community Support Learn More

Oracle VM VirtualBox

The latest release is version 5.0.6.

On this page you can download:

- [Oracle VM VirtualBox](#)
- [Oracle VM VirtualBox Extension Pack](#)
- [Oracle VM VirtualBox Source Code](#)
- [Oracle VM VirtualBox Pre-built Appliances](#)

ORACLE OPEN WORLD
Junho 23-25, 2015
São Paulo, Brasil
#ooowbr
[Inscreva-se](#) >
ORACLE



	<ul style="list-style-type: none"> ▪ Oracle Enterprise Pack for Eclipse 	
Database App Development VM	<ul style="list-style-type: none"> ▪ Oracle Linux 7 ▪ Oracle Database 12c Release 1 Enterprise Edition (12.1.0.2 with In-Memory Option) ▪ Oracle XML DB ▪ Oracle SQL Developer ▪ Oracle SQL Developer Data Modeler ▪ Oracle Application Express ▪ Hands-On-Labs (accessed via the Toolbar Menu in Firefox) 	Downloads and Instructions
Oracle VM 3.2.4 Manager & Server VMs	<ul style="list-style-type: none"> ▪ Oracle Linux 5 update 9 with the Unbreakable Enterprise Kernel (2.6.39) ▪ Oracle VM Manager 3.2.4 	Download and Instructions

I got a **box**.
with Oracle
What now?

EVERY

```
$ vagrant package -base OEL67-ORA12 --output /tmp/oel67-ora12.box
```

```
$ vagrant box list
```

```
OEL65-ORA11 (virtualbox, 0)  
kungfuice/ubuntu64-java (virtualbox, 1.0)
```

```
$ vagrant box add OEL67-ORA12C /tmp/oel67-ora12c.box
```

```
$ vagrant box list
```

```
OEL65-ORA11 (virtualbox, 0)  
OEL67-ORA12 (virtualbox, 0)  
kungfuice/ubuntu64-java (virtualbox, 1.0)
```

```
-- Note! Remove old boxes not in use
```

```
$ vagrant box remove kungfuice/ubuntu64-java
```

the
Vagrantfile

My personal “Oracle Vagrant Framework”

```
$ cd 31_orcl/  
$ tree
```

```
.  
├── files  
│   ├── DBdumps  
│   ├── import_data.sh  
│   ├── provision.sh  
│   ├── setup  
│   │   └── common  
│   │       ├── create_directories_OS.sh  
│   │       ├── create_dev_users.sql  
│   │       ├── drop_dev_users.sql  
│   │       ├── dev_users_env.sql  
│   │       ├── partition_enable.sql  
│   │       ├── profiles.sql  
│   │       └── setup_db.sql  
│   └── work  
└── Vagrantfile
```

```
5 directories, 11 files  
$
```

This structure is a
“definition” of my
“potential”
database machine

The **Vagrantfile**
being the starting point

```
$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :
```

It's just a name reference

```
Vagrant.configure("2") do |config|
  # If several PWH virtual machines, separate by different APPENIX
  $V_HOSTNAME_APPENIX="31"
  $V_BOX_LINK="oel67_ora12.box"
  $V_ORA_VERSION="12"
  $V_OS_VERSION="67"
  $V_DB_NAME="ORCL"
  $V_USER_NAME="LJ"
```

*Let's me set different
IP addressees and port numbers*

```
# -----
# WARNING! DO NOT EDIT BELOW THIS LINE!
# -----
$V_PRIVATE_IP="10.0.2.1"+$V_HOSTNAME_APPENIX
$V_SSH_PORT="22"+$V_HOSTNAME_APPENIX
$V_TNS_PORT="15"+$V_HOSTNAME_APPENIX
$V_BOX_NAME="OEL"+$V_OS_VERSION+"-ORA"+$V_ORA_VERSION
$V_BOX_URL="file:/data/vagrant_boxes/"
$V_HOST_NAME=$V_BOX_NAME+"-"+$V_DB_NAME+"-"+$V_HOSTNAME_APPENIX
```

EVRY

```
config.vm.box=$V_BOX_NAME           #The name of the registered box
config.vm.box_url=$V_BOX_URL        #If not in local repo
config.vm.hostname=$V_HOST_NAME     #The name shown in Virtualbox (unique)
```

```
if Vagrant.has_plugin?("vagrant-proxyconf")
  config.proxy.http      = "http://proxy.edb.com:8080/"
  config.proxy.https     = "http://proxy.edb.com:8080/"
  config.proxy.no_proxy = "localhost,127.0.0.1,.evry.com,.edb.com"
end
```

```
if Vagrant.has_plugin?("vagrant-vbplugin")
  config.vbguest.auto_update = true           -- VboxGuestAddition
end
```

```
config.vm.synced_folder "./", "/vagrant", id: "vagrant-root",
  owner: "vagrant",
  group: "dba",
  mount_options: ["dmode=775,fmode=664"]
```

```
config.vm.synced_folder "/data/ora_sql", "/ora_sql"
  owner: "oracle",
  group: "dba",
  mount_options: ["dmode=755,fmode=664"]
```

```
# Forward Oracle SQLNET and SSH port
config.vm.network :forwarded_port, guest: 1521, host: $V_TNS_PORT
config.vm.network :forwarded_port, id: 'ssh', guest: 22, host: $V_SSH_PORT

# Define private network interface to connect between VMs
#config.vm.network "private_network", ip: $V_PRIVATE_IP
```

```
config.vm.provider :virtualbox do |vb| # Provider specific config
  #vb.gui = true
  vb.cpus = 2
  vb.name = config.vm.hostname
  # Use VBoxManage to customize the VM
  vb.customize ["modifyvm", :id,
    "--memory", "2048",
    # Enable DNS behind NAT
    "--natdnshostresolver1", "on"]
end
```

```
# Provision part, this set up the vagrant box with a oracle test user
config.vm.provision "shell", path: "files/provision.sh", args: $V_ORA_VERSION
end
```

EVERY

READY to
GO!

EVERY


```
$ vagrant plugin install vagrant-proxyconf
```

```
Installing the 'vagrant-proxyconf' plugin. This can take a few minutes...
```

```
Installed the plugin 'vagrant-proxyconf (1.5.2)'
```

```
$ vagrant plugin install vagrant-vbguest
```

```
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
```

```
Installed the plugin 'vagrant-vbguest (0.11.0)'
```

```
$ cd 31_orcl
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'OEL67-ORA12'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: OEL67-ORA12-ORCL-31
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 1521 => 1531 (adapter 1)
    default: 22 => 2231 (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2231
    default: SSH username: vagrant
    default: SSH auth method: password
    default: Warning: Connection timeout. Retrying...
    default: Warning: Connection timeout. Retrying...
    default: Warning: Connection timeout. Retrying...
```

```
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 new SSH key...
==> default: Machine booted and ready!
==> default: Configuring proxy environment variables...
==> default: Configuring proxy for Yum...
GuestAdditions 5.0.10 running --- OK.
==> default: Checking for guest additions in VM...
==> default: Mounting shared folders...
default: /vagrant => /data/vagrant_test/31_orcl
default: /ora_sql => /data/ora_sql
==> default: Running provisioner: shell...
default: Running: /tmp/vagrant-shell120151205-16447-1mzckys.sh
```



This is the **provision.sh** script

the provision.sh

```
# Define APP_USER from in-parameter  
V_ORA_VER=${1}
```

```
cd /vagrant/files/setup/common/  
echo "Creating OS directories..."  
sh create_directories_OS.sh
```

```
sudo mkdir -p /u01/app/oracle/oradata/orcl  
sudo chown -R oracle:oinstall /u01
```

```
#sleep 30  
echo "Creating common objects and users in DB..."  
sudo -u oracle -i <<EOF  
cd /vagrant/files/setup/common  
sqlplus /nolog @setup_db.sql $V_ORA_VER  
EOF
```

```
connect / as sysdba  
@@drop_dev_users.sql &1  
@@create_dev_users.sql &1  
exit
```

```
echo "Importing data"  
if [ -f /vagrant/files/DBdumps/dpump_data.dmp.gz ]; then  
    sudo -u oracle -i sh /vagrant/files/data_import.sh  
fi
```

```
echo "Provisioning finished! "
```

```
==> default: Running provisioner: shell...
      default: Running: /tmp/vagrant-shell120151205-16447-1mzckys.sh
==> default: Creating OS directories...
==> default: Creating common objects and users in DB...
==> default: SQL*Plus: Release 12.1.0.2.0 Prod... on Sat Dec 5 21:28:15 2015
==> default: Copyright (c) 1982, 2014, Oracle. All rights reserved.
==> default: Connected.
==> default: * ----- *
==> default: * Script: drop_dev_users.sql is running (please wait ...)
==> default: * ----- *
==> default: *** ENVIRONMENT SETTINGS: ***
==> default: Data Owner           : LJDATA
==> default: Application User      : LJ
==> default: Support User          : LJSUPP
==> default: Oracle Version        : 12
==> default: Grant Package         : USER_GRANT
==> default: Read-Only Role        : LJ_RO
==> default: Read-Write Role       : LJ_RW
==> default: Atomikos Enabled      : TRUE
==> default: Tablespace Directory  : /u01/app/oracle/oradata/orcl
==> default: Pluggable Name        : ORCL
```

```
==> default: ***          Dropping USERS (waiting)          ***
==> default: User LJDATA does not exist.
==> default: User LJSUPP does not exist.
==> default: User LJ does not exist.
==> default: ***          Dropping ROLES                      ***
==> default: Role LJ_RO does not exist.
==> default: Role LJ_RW does not exist.
==> default: ***          Dropping Tablespaces (waiting)     ***
==> default: Tablespace LJ_DATA does not exist.
==> default: Tablespace LJ_IDX does not exist.
==> default: Tablespace LJ_LOB does not exist.
==> default: ***          Dropping Triggers                  ***
==> default: Triggers dropped with users.
```

```
==> default: * ----- *
==> default: * Script: create_dev_users.sql is running (please wait ...)
==> default: * ----- *
==> default: Successfully set container(PDB) to ORCL
==> default: *** ENVIRONMENT SETTINGS: ***
==> default: Data Owner          : LJDATA
==> default: Application User       : LJ
==> default: Support User         : LJSUPP
...
==> default: ***          Creating TABLESPACES ... (waiting)          ***
==> default: Creating tablespace LJ_DATA succeeded.
==> default: Creating tablespace LJ_IDX succeeded.
==> default: Creating tablespace LJ_LOB succeeded.
==> default: ***          Creating ROLES and give GRANTS          ***
==> default: Creating role LJ_RO succeeded.
==> default: Creating role LJ_RW succeeded.
==> default: Granting SELECT on sys.dba_pending_transactions to LJ_RW succ ..
==> default: Granting SELECT on sys.pending_trans$ to LJ_RW succeeded
==> default: Granting SELECT on sys.dba_2pc_pending to LJ_RW succeeded
==> default: Granting EXECUTE on sys.dbms_xa to LJ_RW succeeded
```

```
==> default: ***          Creating users ... (waiting)          ***
==> default: User LJ created successfully.
==> default: User LJ granted CREATE SESSION successfully
==> default: User LJDATA created successfully.
==> default: User LJDATA granted CREATE SESSION successfully
==> default: User LJDATA granted owner rights successfully.
==> default: User LJSUPP created successfully.
==> default: User LJSUPP granted CREATE SESSION successfully
==> default: User LJ granted LJ_RW successfully.
==> default: User LJSUPP granted LJ_RO successfully.
==> default: ***          Creating triggers ...          ***
==> default: Trigger TRG_LJ created successfully.
==> default: Trigger TRG_LJSUPP created successfully.
==> default: ***          Creating GRANT package ...          ***
==> default: Package USER_GRANT created successfully.
==> default: Package body USER_GRANT created successfully.
==> default: Synonym for USER_GRANT created successfully.
==> default: Disconnected from Oracle 12c EE Release 12.1.0.2.0 - 64bit ...
==> default: With the Partitioning, OLAP, Adv. Analytics and RAT options
==> default: Importing data
==> default: Provision finished!
```



```
$ vagrant ssh
```

```
Last login: Sat Dec 5 22:01:24 2015 from 10.0.2.2
```

```
-----  
Oracle Enterprise Linux 6.7
```

```
built 2015-11-23  
-----
```

```
[vagrant@OEL67-ORA12-ORCL-32 ~]$ sudo su - oracle
```

```
[oracle@OEL67-ORA12-ORCL-32 ~]$ sqlplus ljdata/lj@orcl
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Sat Dec 5 22:30:43 2015
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
```

```
...
```

```
SQL> create table test (id number, name varchar2(10));
```

```
Table created.
```

```
SQL> insert into test values (1,'LASSE');
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

The logo for 'EVERY' is written in a stylized, blue, handwritten-style font. The letters are bold and slightly irregular, with the 'Y' having a long, sweeping tail that curves upwards.

New / Select Database Connection

Connection Name	Connection Details
orcl31_ljdata	ljdata@//localhost...

Connection Name: orcl31_ljdata
Username: ljdata
Password: ..

Save Password Connection Color

Oracle

Connection Type: Basic Role: default

Hostname: localhost
Port: 1531

SID
 Service name: orcl

OS Authentication Kerberos Authentication

Status : Success



Connections



Connections

- orcl31_ljdata
 - Tables (Filtered)
 - TEST
 - Views
 - Editing Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Queues
 - Queues Tables
 - Triggers
 - Crossedition Triggers

Reports

- All Reports
 - Data Dictionary Reports
 - Data Modeler Reports
 - OLAP Reports
 - TimesTen Reports
 - User Defined Reports

Start Page | orcl31_ljdata | TEST

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID	NUMBER	Yes	(null)	1 (null)	
2	NAME	VARCHAR2(10 BYTE)	Yes	(null)	2 (null)	

**If I want to
Test ...**

```
$ vagrant plugin install sahara
Installing the 'sahara' plugin. This can take a few minutes...
Installed the plugin 'sahara (0.0.17) '!
$ vagrant sandbox on
[default] Starting sandbox mode...
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
$ vagrant ssh
[vagrant@OEL67-ORA12C-ORCL-31 ~]$ sudo su - oracle
[oracle@OEL67-ORA12C-ORCL-31 ~]$ sqlplus lldata/lj@orcl
SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 6 09:16:04 2015
...
SQL> delete from test;
1 row deleted.
SQL> commit;
Commit complete.

SQL> exit
[oracle@OEL67-ORA12C-ORCL-31 ~]$ exit
[vagrant@OEL67-ORA12C-ORCL-31 ~]$ exit
Connection to 127.0.0.1 closed.
$
```

```
$ vagrant sandbox rollback
```

```
[default] Rolling back the virtual machine...
```

```
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

```
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

```
$ vagrant ssh
```

```
[vagrant@OEL67-ORA12C-ORCL-31 ~]$ sudo su - oracle
```

```
[oracle@OEL67-ORA12C-ORCL-31 ~]$ sqlplus lldata/lj@orcl
```

```
SQL> select * from test;
```

```
      ID NAME
```

```
-----
```

```
1 LASSE
```

```
SQL>
```

\$ vagrant sandbox commit

Save snapshot

[default] Committing the virtual machine...

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

\$ vagrant sandbox off

delete snapshots

[default] Stopping sandbox mode...

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

**If I want to
Empty DB ...**

\$ vagrant provision

```
==> default: Configuring proxy environment variables...
==> default: Configuring proxy for Yum...
==> default: Running provisioner: shell...
    default: Running: /tmp/vagrant-shell20151206-25974-1ju0emf.sh
==> default: Creating OS directories...
==> default: Creating common objects and users in DB...
==> default: SQL*Plus: Release 12.1.0.2.0 on Sat Dec 5 21:28:15 2015
==> default: Copyright (c) 1982, 2014, Oracle. All rights reserved.
==> default: Connected.
==> default: * ----- *
==> default: * Script: drop_dev_users.sql is running (please wait ...)
==> default: * ----- *
...
==> default: * ----- *
==> default: * Script: create_dev_users.sql is running (please wait ...)
==> default: * ----- *
...
==> default: Disconnected from Oracle DB 12c EE Release 12.1.0.2.0 - 64bit ..
==> default: With the Partitioning, OLAP, Advanced Analytics and RAT options
==> default: Importing data
==> default: Provision finished!
```

EVRY

**If I want a
New Server**

```
$ cp -R 31_orcl 32_orcl
$ cd 32_orcl
$ vi Vagrantfile
$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :
```

```
Vagrant.configure("2") do |config|
  # If several virtual machines, separate by different APPENDIX
  $V_HOSTNAME_APPENIX="32"
  $V_BOX_LINK="oel67_ora12c.box"
  $V_ORA_VERSION="12"
  $V_OS_VERSION="67"
  $V_DB_NAME="ORCL"
  $V_USER_NAME="LJ"
  ...

$ rm -rf .vagrant/
$ vagrant up
```

**If I want a
Clean up ...**

```
$ cd 32_orcl
$ vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Forcing shutdown of VM...
==> default: Destroying VM and associated drives...
==> default: Running cleanup tasks for 'shell' provisioner...
$ cd ..
$ rm -rf 32_orcl
```

or Jenkins
**If your developer wants to
run SQLs from his source (git) to
Build DATA schema**

[▶ GET STARTED](#)[DOCUMENTATION](#)[CONTRIBUTE](#)[BLOG](#)[SUPPORT](#)[Overview](#)[Why database migrations](#)[How Flyway works](#)**FIRST STEPS**[Command-line](#)[API](#)[Maven](#)[Gradle](#)[Ant](#)[SBT](#)**DOWNLOAD**[Download Flyway](#)

Download

These are the downloads for the latest version of Flyway:

Client	Downloads	Source
Command-line Tool	flyway-commandline-3.2.1-windows-x64.zip flyway-commandline-3.2.1-linux-x64.tar.gz flyway-commandline-3.2.1-macosx-x64.tar.gz flyway-commandline-3.2.1.zip (without JRE) flyway-commandline-3.2.1.tar.gz (without JRE)	flyway-commandline-3.2.1-sources.jar
API	flyway-core-3.2.1.jar	flyway-core-3.2.1-sources.jar
Maven Plugin	flyway-maven-plugin-3.2.1.jar	flyway-maven-plugin-3.2.1-sources.jar
Gradle Plugin	flyway-gradle-plugin-3.2.1.jar	flyway-gradle-plugin-3.2.1-sources.jar
Ant Tasks	flyway-ant-3.2.1.zip flyway-ant-3.2.1.tar.gz	flyway-ant-3.2.1-sources.jar
SBT Plugin	flyway-sbt-3.2.1.jar	flyway-sbt-3.2.1-sources.jar

#Configure FlywayDB:

```
$ vi /data/flyway-3.2.1/conf/flyway.conf
flyway.url=jdbc:oracle:thin:@//localhost:1521/orcl
flyway.user=ljdata
flyway.password=lj
flyway.locations=filesystem:/ora_sql/flyway/
```

#Add this to Vagrantfile

```
config.vm.synced_folder "/data/apps/flyway-3.2.1", "/flyway",
    owner: "vagrant", group: "dba", mount_options: ["dmode=775,fmode=775"]
```

#Copy Oracle driver to the drivers folder (standing inside my guest os)

```
$ cd /flyway/drivers
$ cp /u01/app/oracle/product/12.1.0/dbhome_1/jdbc/lib/ojdbc6.jar .
```


#Copy your scripts to your flyway directory

```
$ ls -l /data/ora_sql/flyway/
```

```
-rw-rw-r-- 1 ek2046 ek2046  77 des.    8 15:15 V1_1__Create_person_table.sql
-rw-rw-r-- 1 ek2046 ek2046 156 des.    8 15:15 V2__Add_people.sql
```

#Add to provision.sh

```
echo "Migrating schema if flyway script is available ..."
```

```
if [ "$(ls -A /ora_sql/flyway)" ]; then
```

```
    sudo -u oracle -i sh /flyway/flyway migrate -baselineOnMigrate=true
```

```
else
```

```
    echo "Flyway: No scripts available"
```

```
fi
```

Ready to GO

```
$ vagrant provision
```

\$ vagrant provision

```
...
==> default: Connected.
==> default: * ----- *
==> default: * Script: drop_dev_users.sql is running (please wait ...)
==> default: * ----- *
==> default: * ----- *
==> default: * Script: create_dev_users.sql is running (please wait ...)
==> default: * ----- *
==> default: Migrating schema if flyway script is available ...
==> default: Flyway 3.2.1 by Boxfuse
==> default: Database: jdbc:oracle:thin:@//localhost:1521/orcl (Oracle 12.1)
==> default: Validated 2 migrations (execution time 00:00.107s)
==> default: Creating Metadata table: "LJDATA"."schema_version"
==> default: Schema baselined with version: 1
==> default: Current version of schema "LJDATA": 1
==> default: Migrating schema "LJDATA" to version 1.1 - Create person table
==> default: Migrating schema "LJDATA" to version 2 - Add people
==> default: Successfully applied 2 migrations to schema "LJDATA" (execution
time 00:00.051s).
==> default: Provision finished!
```

Back to Creating Boxes



Provisioning

[Basic Usage](#)[File](#)[Shell](#)[Ansible](#)[CFEngine](#)[Chef Solo](#)[Chef Zero](#)[Chef Client](#)[Chef Apply](#)[Docker](#)[Puppet Apply](#)[Puppet Agent](#)[Salt](#)

If you've never used a configuration management system before, it is recommended you start with basic [shell scripts](#) for provisioning.

You can find the full list of built-in provisioners and usage of these provisioners in the navigational area to the left.

WHEN PROVISIONING HAPPENS

Provisioning happens at certain points during the lifetime of your Vagrant environment:

- On the first `vagrant up` that creates the environment, provisioning is run. If the environment was already created and the up is just resuming a machine or booting it up, they won't run unless the `--provision` flag is explicitly provided.
- When `vagrant provision` is used on a running environment.
- When `vagrant reload --provision` is called. The `--provision` flag must be present to force provisioning.

You can also bring up your environment and explicitly *not* run provisioners by specifying `no-provision`.

EVERY



Packer.io



... running on FreeBSD, OpenBSD, Windows, Linux & Macintosh hosts

... working on the shoulder of giants

(like Amazon EC2, Digital Ocean, Docker, Google Compute Engine, VMware, VirtualBox w/more)

Configure your buildfile, and build your box ...

```
> packer build -only=virtualbox-iso oraclelinux-6.6-x86_64.json
```

EVERY



SALTSTACK

Salt Open



SALTSTACK

a configuration management system,
capable of maintaining remote nodes in defined states

a distributed remote execution system used to
execute commands and query data on remote nodes

Master

Minion

freely available

Open Source Software under the terms of the
GNU General Public License (GPL) version 2

Tested and packaged to run on
CentOS, Debian, RHEL, Ubuntu, Windows

EVERY

```
$ tree
```

```
.
├── Makefile
├── minion
├── srv
│   ├── pillar
│   │   ├── oracle12c
│   │   │   └── init.sls
│   │   └── top.sls
│   └── salt
│       ├── oracle12c
│       │   ├── <some files>
│       │   ├── <some files>
│       │   └── <some files>
│       └── ...
│           ├── init.sls
│           ├── map.jinja
│           └── top.sls
├── uploadboxtoshare.sh
└── Vagrantfile
```

```
$ make export
```

```
clean:  vagrant destroy -f
        rm -f *box

setup:  clean
        set vagrant --version
        VBoxManage --version
        VBoxManage setproperty machinefolder
                $(VBOX_USER_HOME)/machinefolder
        VBoxManage setproperty hwvirtexclusive off
        vagrant plugin install vagrant-proxyconf
        vagrant plugin install vagrant-vbguest
        vagrant plugin install vagrant-reload

build:  setup
        vagrant up

export: build
        vagrant halt
        vagrant package --output $(EXPORT_BOX_FILE)
        vagrant destroy -f

upload: export
        ./uploadboxtoshare.sh $(EXPORT_BOX_FILE)

all:    upload
```

EVRY

```
$ tree
```

```
.
├── Makefile
├── minion
├── srv
│   ├── pillar
│   │   ├── oracle12c
│   │   │   └── init.sls
│   │   └── top.sls
│   └── salt
│       ├── oracle12c
│       │   ├── <some files>
│       │   ├── <some files>
│       │   └── <some files>
│       └── ...
│           ├── init.sls
│           ├── map.jinja
│           └── top.sls
├── uploadboxtoshare.sh
└── Vagrantfile
```

We use a pre built Oracle Linux Box:

```
vagrant.configure("2") do |config|
  config.vm.box = "boxcutter/ol67"
  config.vm.synced_folder "srv", "/srv/"
  config.vm.synced_folder ".", "/vagrant"
```

Then we install Salt inside our box with Vagrant:

```
config.vm.provision :salt do |salt|
  salt.verbose = true
end

#...issues/5973
config.vm.provision :shell, :inline => _
  "sudo cp /vagrant/minion /etc/salt/minion && _
  sudo service salt-minion restart && _
  salt-call state.highstate"

#then reload so we can...
config.vm.provision :reload
```



```
/etc/salt/minion
state_top: top.sls

file_client: local

file_roots:
  base:
    - /srv/salt

pillar_roots:
  base:
    - /srv/pillar
```

(default) starting point

Defines that we are using
Standalone minion

Define **path** for the “**base**” environment
to local files (not using master)

```
/srv/salt/top.sls
base:
  '*':
    - oracle12c
```

Apply SLS files from the directory root for the 'base' environment

For all minions with a minion_id that match '*'

Apply the state in file named '**oracle12c.sls**'

“Environments are directory hierarchies which contain a top files and a set of state files.

*Environments can be used in many ways, however there is no requirement that they be used at all. In fact, the most common way to deploy Salt is with a single environment, called **base**.”*

```
$ tree
```

```
.
├── Makefile
├── minion
├── srv
│   ├── pillar
│   │   ├── oracle12c
│   │   │   └── init.sls
│   │   └── top.sls
│   └── salt
│       ├── oracle12c
│       │   ├── <some files>
│       │   ├── <some files>
│       │   └── <some files>
│       └── ...
│           ├── init.sls
│           ├── map.jinja
│           └── top.sls
├── uploadboxtoshare.sh
└── Vagrantfile
```

Starting SALT provisioning:

```
salt-call state.highstate
```

The top.sls -> oracle12c

SALT find the default file: **init.sls**

```
$ cat srv/salt/oracle12c/init.sls
{% from "oracle12c/map.jinja" import extractdir, ora_app, ora_prod with
context %}
                                /tmp/install_files

{% set ora_d = extractdir ~ '/ora' %}
                                /u01/app/oracle
                                /u01/app/oracle/product/12.1.0

unzip:
  pkg.installed

# Limiting Maximum Number of Processes Available for the Oracle User
/etc/security/limits.d/90-nproc.conf:
  file.managed:
    - source: salt://oracle12c/etc/security/limits.d/90-nproc.conf

/etc/fstab:
  file.managed:
    - source: salt://oracle12c/etc/fstab
```

```
wget -O /tmp/p17694377_121020_Linux-x86-64_1of8.zip  
http://fsfiles.evry.com/vagrant_boxes/sw/oracle_source/p17694377_121020_Linux  
-x86-64_1of8.zip:
```

cmd.run

Hack to work around issue with “archive.extracted”

```
wget -O /tmp/p17694377_121020_Linux-x86-64_2of8.zip  
http://fsfiles.evry.com/vagrant_boxes/sw/oracle_source/p17694377_121020_Linux  
-x86-64_2of8.zip:
```

cmd.run

```
mkdir -p {{ora_d}} && unzip /tmp/p17694377_121020_Linux-x86-64_1of8.zip -d  
{{ora_d}} && unzip /tmp/p17694377_121020_Linux-x86-64_2of8.zip -d {{ora_d}}  
&& rm /tmp/p17694377_121020_Linux-x86-64_1of8.zip  
/tmp/p17694377_121020_Linux-x86-64_2of8.zip:
```

cmd.run

```
#https://github.com/saltstack/salt/issues/23822:  
chmod -R a+x {{extractdir}}:
```

cmd.run

```
oracle-rdbms-server-12cR1-preinstall:  
  pkg.installed
```

```
{% for file in [ '12c_oracle_EE.rsp', '12c_cfgrsp.properties' ] %}  
{{extractdir}}/{{file}}:  
  file.managed:  
    - source: salt://oracle12c/{{file}}  
{% endfor %}
```

Python Code

```
{{ora_app}}:  
  file.directory:  
    - user: oracle  
    - group: oinstall  
    - mode: 775  
    - makedirs: True
```

```
#need to -ignorePrereq since numprocesses are not yet read by this shell:  
{{ora_d}}/database/runInstaller -silent -ignorePrereq -waitforcompletion -  
ignoreSysPrereqs -responseFile {{extractdir}}/12c_oracle_EE.rsp:  
  cmd.run:  
    - user: oracle
```

EVERY

```
/u01/app/oraInventory/orainstRoot.sh:  
  cmd.run
```

```
{{ora_prod}}/dbhome_1/root.sh:  
  cmd.run
```

```
{{ora_prod}}/dbhome_1/cfgtoollogs/configToolAllCommands  
RESPONSE_FILE={{extractdir}}/12c_cfgrsp.properties:  
  cmd.run:  
    - user: oracle
```

```
/tmp/12c_pdb_autostart.sh: # trigger  
  file.managed:  
    - source: salt://oracle12c/12c_pdb_autostart.sh  
    - mode: 0755  
    - user: oracle  
    - group: oinstall  
  cmd.run:  
    - user: oracle
```

```
/etc/init.d/dbora:
  file.managed:
    - source: salt://oracle12c/etc/init.d/dbora
    - user: root
    - group: root
    - mode: 755

chkconfig --add dbora:          # Run dbora when restarting
  cmd.run

/home/oracle/.profile:         # Set Oracle Environment
  file.managed:
    - source: salt://oracle12c/home/oracle/.profile
    - user: oracle
    - group: oinstall

/home/oracle/.bash_profile:
  file.managed:
    - source: salt://oracle12c/home/oracle/.profile
    - user: oracle
    - group: oinstall
```



```
/home/oracle/scripts:
  file.recurse:
    - source: salt://oracle12c/home/oracle/scripts
    - user: oracle
    - file_mode: 0755

#start database so we can run sqlplus on it afterwards:
/etc/init.d/dbora start:
  cmd.run

{% for script in [ 'noexpirepw', 'create_dev_users', 'alteruser' ] %}
/tmp/{{script}}.sql:
  file.managed:
    - source: salt://oracle12c/{{script}}.sql

{{ora_prod}}/dbhome_1/bin/sqlplus "/ as sysdba" @/tmp/{{script}}.sql >
/tmp/{{script}}.log:
  cmd.run:
    - user: oracle
{% endfor %}
```

```
/tmp/shrink_tablespaces_part1.sql:                # Resize TEMP and UNDO
  file.managed:
    - source: salt://oracle12c/shrink_tablespaces_part1.sql

/tmp/shrink_tablespaces_part2.sql:
  file.managed:
    - source: salt://oracle12c/shrink_tablespaces_part2.sql

# Network Config
{{ora_prod}}/dbhome_1/network/admin/listener.ora
  file.managed:
    - source:
salt://oracle12c/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listene
r.ora

{{ora_prod}}/dbhome_1/network/admin/tnsnames.ora
  file.managed:
    - source:
salt://oracle12c/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/tnsname
s.ora
```

```
{{ora_prod}}/dbhome_1/bin/sqlplus / as sysdba
@/tmp/shrink_tablespaces_part1.sql > /tmp/shrink_tablespaces.log:
  cmd.run:
    - user: oracle

restartDB_ts:
  cmd.run:
    - user: oracle
    - name: /home/oracle/scripts/shutdown.sh &&
/home/oracle/scripts/startup.sh

{{ora_prod}}/dbhome_1/bin/sqlplus / as sysdba
@/tmp/shrink_tablespaces_part2.sql >> /tmp/shrink_tablespaces.log:
  cmd.run:
    - user: oracle

#cleanup
{{extractdir}}:
  file.absent
```

Just `$ make export`
push the button

... and Vagrant & Salt makes the box for you!

Any

Questions *www.jcon.no/oracle*
#lasjen

EVERY

Thanks!

www.jcon.no/oracle

#lasjen

EVERY



EVERY

We bring information to life