

# Oracle VM VirtualBox

## Updates and Vision

Simon Coter

Director of Product Management, Oracle VM & VirtualBox

Klaus Espenlaub

Director of Software Development, VirtualBox

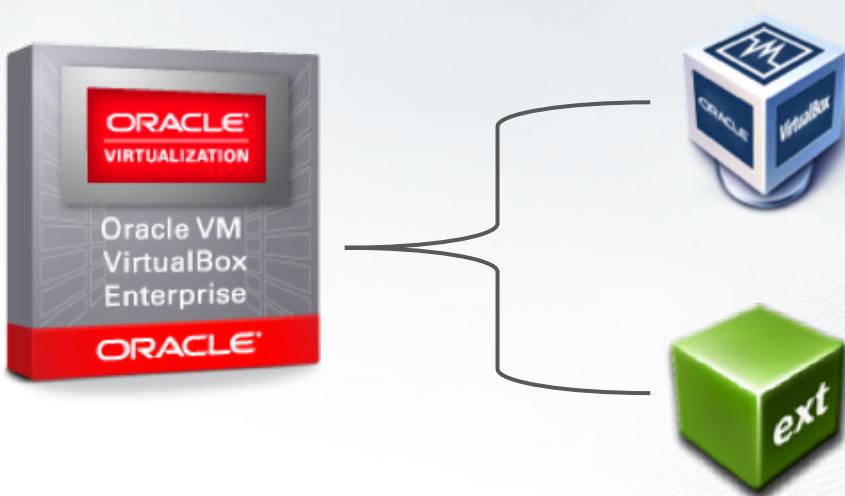
February 13<sup>th</sup>, 2018

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Oracle VM VirtualBox Components

## Enterprise Features part of the Extension Pack



### Base package

- Includes all open-source components, licensed under the [GPL v2](#)
- Available for all common x86 platforms (Windows, Linux, Solaris, OS X)
- Features: Base Hypervisor, GUI

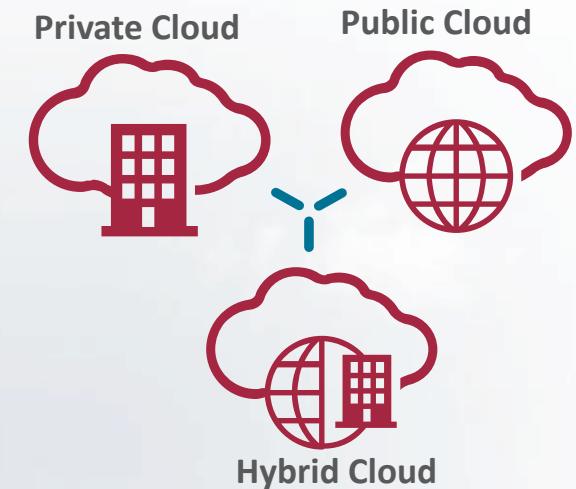
### Extension Pack

- Binaries released under the VirtualBox [Personal Use and Evaluation License \(PUEL\)](#)
- License Text/FAQ manages the Cloud features
- Same component for all platforms
- Existing Features: Encryption, USB2/3 devices, Remote Management, PXE-Boot, Webcam Passthrough, NVMe emulation, Remote Management (RDP)

# Oracle VM VirtualBox

Free and open source, Cross-Platform Virtualization Software

- Available for Windows, Mac OS X, Linux and Oracle Solaris
- Supports a wide range of guest platforms
- Easy-to-use graphical user interface
- Powerful, scriptable command line interface
- Import and export virtual machines using OVF/OVA standards
- Shared folders between guest and host
- Create a multi-platform test and development environment
- Extend the lifetime and usefulness of existing computers



Download  
& Test

Dev &  
Deploy



# Oracle VM VirtualBox 5.2

## New Features & Important things to know



VirtualBox  
**5.2**

- Export to “Oracle Public Cloud” format (Classic)
- Includes very latest OS releases (Win-2016)
- Automatic guest OS installation for Virtual Machines
- New Virtual Machine & Global Tools
- New GUI Icons and refreshed interface
- Audio:
  - Ability to enable or disable audio input/output on-the-fly
  - Device enumeration support for audio backends
  - Support for host device callbacks (optional) and further

# Oracle VM VirtualBox 5.2

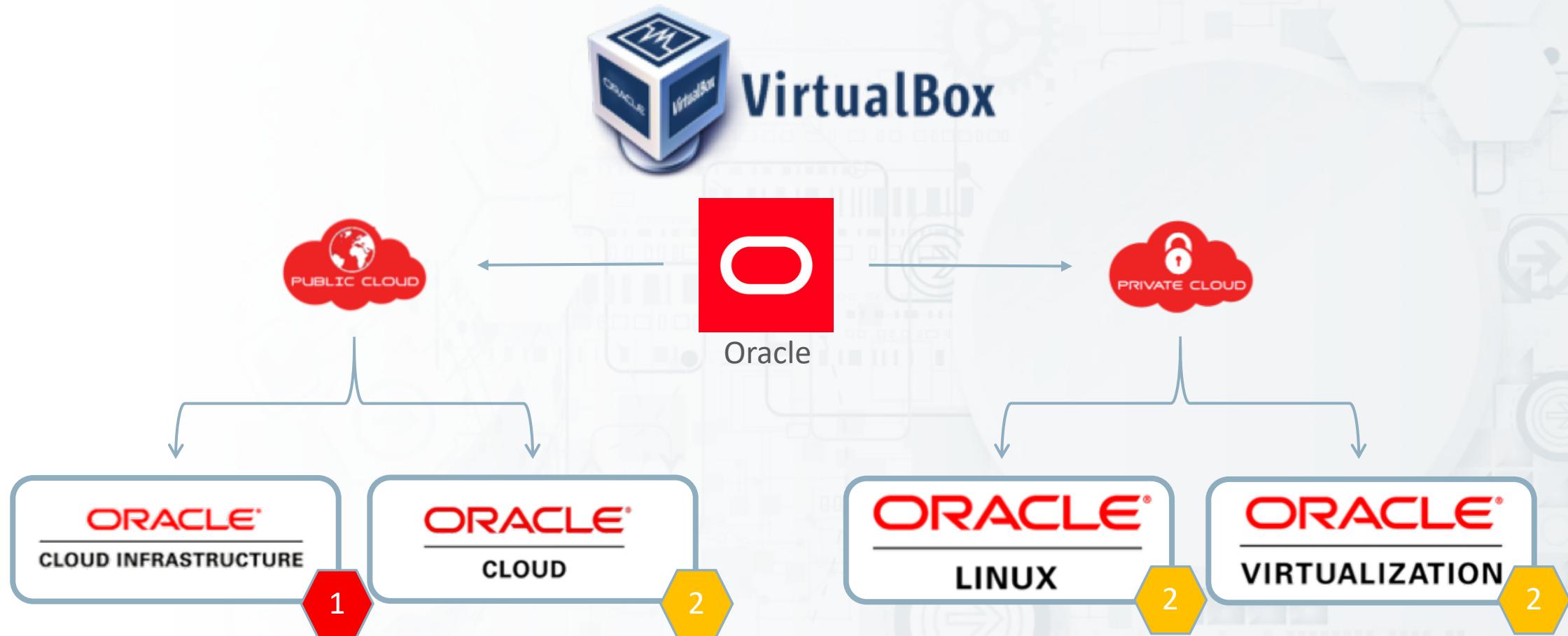
## New Features & Important things to know



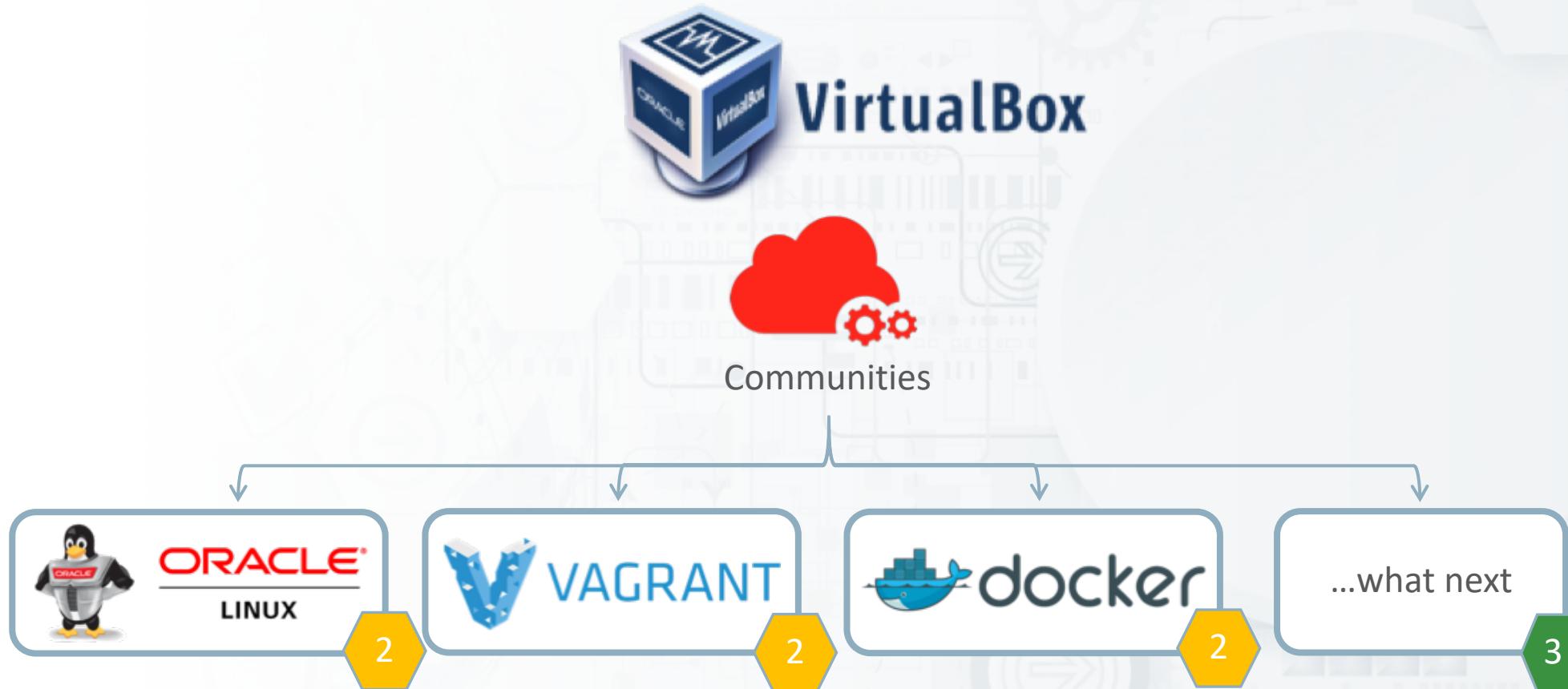
VirtualBox  
**5.2**

- Updated [VirtualBox Data-Sheet for 5.2](#)
- Updated [List of Supported Platforms](#)
- VirtualBox Guest-Additions RPMs for Oracle Linux
  - Very soon public available on OL Yum Repositories
- Further details on enhancement, fixes available on official [VirtualBox Changelog](#)
- Official Announcement on [Virtualization Blog](#)
- 5.2.6 is latest release available today
- 5.2.8 expected very soon

# Oracle VM VirtualBox Roadmap: Oracle Ecosystem



# Oracle VM VirtualBox Roadmap: Community Ecosystem



# Oracle VM VirtualBox: Priorities, Features and Releases

## Oracle: Public & Private Cloud

Ecosystem	Target	Priority	Release	Comments
	OCI emulation by VirtualBox	1	6.0	<ul style="list-style-type: none"><li>Network, Storage (NVMe, iSCSI), Automation (cloud-init); possible “Service Machine” OL7 based.</li></ul>
	VirtualBox as Dev Platform for OCI	1	6.0	<ul style="list-style-type: none"><li>Instances created on VirtualBox and deployed to BMCS</li><li>BYOI for both bare-metal and virtual instances</li></ul>
	Improved VM deployment	2	5.2.x/6.0	<ul style="list-style-type: none"><li>Virtual Machine auto-install (ongoing project for 5.2)</li><li>Adding value to both Cloud and on-premise solutions</li></ul>
	Nested Virtualization	2	5.2.x/6.0	<ul style="list-style-type: none"><li>VT-x/AMD-V passthrough on VirtualBox VMs</li></ul>
	Improved automation	2	6.0	<ul style="list-style-type: none"><li>Cloud-init integrated in VirtualBox (BMCS / OVM 4)</li><li>Possible integration with OVM 3</li></ul>

# Oracle VM VirtualBox: Priorities, Features and Releases

## DevOps: Linux, Vagrant and Docker

Ecosystem	Target	Priority	Release	Comments
 VAGRANT	Improve Vagrant Experience	 2	5.2.x/6.0	<ul style="list-style-type: none"><li>• Improve shared-folders performance</li><li>• Integration with Vagrant Box Repositories (TBD)</li><li>• Updated Vagrant Boxes for OL</li></ul>
 docker	VirtualBox as a Docker GUI	 2	6.0	<ul style="list-style-type: none"><li>• Manage local and remote Docker installations</li><li>• Integrated with Docker registry</li><li>• Integrated with Docker Container Registry (Oracle)</li></ul>

# Custom VirtualBox automation needed?

For easy cases: CLI

- Many operations can be automated using VBoxManage
  - Command synopsis is a mile long, but always worth a look
  - Widely known: VM creation, configuration changes, snapshot operations, disk image management, showing lists and detail information, virtual appliance export/import
  - Less known: control VM while it is running, e.g. causing a clean shutdown, triggering keypresses, reconfiguring networking including a built-in packet sniffer, attach/detach USB devices, record a video of a VM session and many more
  - Least known: copy files from/to VM, execute programs inside VM
- Downside: reasonable error handling is tricky in shell scripts

# Custom VirtualBox automation needed?

For complex cases: API

- API is available locally and through an optional web service (SOAP)
  - VirtualBox GUI and VBoxManage are regular API clients
  - Full flexibility for complex custom automation, including powerful error handling
  - Events available for various status changes
  - All long-running operations are asynchronous, with progress information, allowing far more parallelism than the GUI or VBoxManage
  - Easy to use from many programming languages: Python, Java, PHP and more
- Suitable for sophisticated VM management
- More knowledge and effort needed than for simple VBoxManage use

# VirtualBox API programming

## How it looks in Python

```
from vboxapi import VirtualBoxManager

mgr = VirtualBoxManager(None, None)

vbox = mgr.vbox

mach = vbox.findMachine("OL7.4")

session = mgr.getSessionObject()

progress = mach.launchVMProcess(session, "gui", "")

progress.waitForCompletion(-1)

session.unlockMachine()
```

# VirtualBox: unattended OS installation

## Automated Virtual Machine installation

- Create Virtual Machine by “VBoxManage”

```
[scoter@techserv: ~]# VM='Win10-autoinstall'
[scoter@techserv: ~]# VBoxManage createvm --name $VM --ostype "Windows10_64" --register
Virtual machine 'Win10-autoinstall' is created and registered.
UUID: e24362ce-d4e4-4f8e-bb54-f638b5dfcfcd
Settings file: '/VirtualBox/Win10-autoinstall/Win10-autoinstall.vbox'
[scoter@techserv: ~]#
```

- Setup Virtual Storage for the Virtual Machine

```
[scoter@techserv: ~]# VBoxManage createhd --filename /VirtualBox/$VM/$VM.vdi --size 32768
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Medium created. UUID: d4537089-6db6-40dc-8adf-756c897c3e8e
[scoter@techserv: ~]# VBoxManage storagectl $VM --name "SATA Controller" --add sata --controller IntelAHCI
[scoter@techserv: ~]# VBoxManage storageattach $VM --storagectl "SATA Controller" --port 0 --device 0 --type hdd --medium /VirtualBox/$VM/$VM.vdi
[scoter@techserv: ~]# VBoxManage storagectl $VM --name "IDE Controller" --add ide
[scoter@techserv: ~]# VBoxManage storageattach $VM --storagectl "IDE Controller" --port 0 --device 0 --type dvddrive --medium /u01/Software/en_windows_10_pro_10240_x64_dvd.iso
[scoter@techserv: ~]#
```

- Misc Settings for the Virtual Machine (boot, memory, video, network)

```
[scoter@techserv: ~]# VBoxManage modifyvm $VM --ioapic on
[scoter@techserv: ~]# VBoxManage modifyvm $VM --boot1 dvd --boot2 disk --boot3 none --boot4 none
[scoter@techserv: ~]# VBoxManage modifyvm $VM --memory 8192 --vram 128
[scoter@techserv: ~]# VBoxManage modifyvm $VM --nic1 bridged --bridgeadapter1 e1000g0
[scoter@techserv: ~]#
```

# VirtualBox: unattended OS installation

## Automated Virtual Machine installation

- Unattended Installation Configuration

```
1. scoter@techserv: ~# VBoxManage unattended install $VM --iso=/u01/Software/en_windows_10_pro_10240_x64_dvd.iso --user=scoter --full-user-name "Mr. Coter" --password secret --key=NPPR9-FWDCX-D2C8J-H872K-2YT43 --install-additions --time-zone=UTC
VBoxManage: info: Preparing unattended installation of Windows10_64 in machine 'Win10-autoinstall' (e24362ce-d4e4-4f8e-bb54-f638b5dfcfcd).
VBoxManage: info: Using values:
    isoPath = /u01/Software/en_windows_10_pro_10240_x64_dvd.iso
        user = scoter
        password = secret
    fullUserName = Mr. Coter
        productKey = NPPR9-FWDCX-D2C8J-H872K-2YT43
    additionsIsoPath = /opt/VirtualBox/additions/VBoxGuestAdditions.iso
    installGuestAdditions = true
    validationKitIsoPath =
    installTestExecService = false
        locale = en_US
        country = CT
        timeZone = UTC
        proxy =
        hostname = Win10-autoinstall.myguest.virtualbox.org
    packageSelectionAdjustments =
        auxiliaryBasePath = /VirtualBox/Win10-autoinstall/Unattended-e24362ce-d4e4-4f8e-bb54-f638b5dfcfcd-
            imageIndex = 1
        scriptTemplatePath = /opt/VirtualBox/UnattendedTemplates/win_nt6_unattended.xml
    postInstallScriptTemplatePath = /opt/VirtualBox/UnattendedTemplates/win_postinstall.cmd
    postInstallCommand =
    extraInstallKernelParameters =
        language = en-US
        detectedOSTypeId = WindowsNT_64
        detectedOSVersion =
        detectedOSFlavor =
        detectedOSLanguages = en-US
        detectedSHints =
VBoxManage: info: VM 'Win10-autoinstall' (e24362ce-d4e4-4f8e-bb54-f638b5dfcfcd) is ready to be started (e.g. VBoxManage startvm).
[scoter@techserv: ~# ]
```

# VirtualBox: unattended OS installation

## Automated Virtual Machine installation

- Start Virtual Machine to get it completely installed

```
● ● ● 1. scoter@techserv.it.oracle.com: /opt/VirtualBox/UnattendedTemplates (ssh)
[scoter@techserv: ~]# VBoxManage startvm $VM --type headless
Waiting for VM "Win10-autoinstall" to power on...
VM "Win10-autoinstall" has been successfully started.
```

# Integrated Cloud Applications & Platform Services

**ORACLE®**