

[Appliances and OVF: The packaged approach to deployment](#)

Appliances are a great way to approach system deployments for your VMware environment. We are seeing more and more of this method of delivery of application environments, and with the increased number of vendors jumping on board, this is a sign that something is being done right.



What is an Appliance?

The appliance in a VMware sense is a packaged server with a pre-installed application environment that is provided in OVF format which we can deploy directly into our vSphere environment without having to build the base machine and do all of the installation process ourselves.

Using appliances is much like the physical, black-box approach which can ease the pain of getting a new application environment up and running. Configuration is done minimally through the console, and a web application is almost always the way to complete additional customization and configuration.

Using the OVF Format

OVF, or Open Virtualization Format, is an industry standard for packaging defined by the DMTF (Distributed Management Task Force) to provide vendors and software providers with a supportable and standards based delivery format. Details on the OVF standard can be found here: <http://www.dmtf.org/standards/ovf>

A virtual appliance will come with a pre-configured virtual disk file and a configuration file to provide the vSphere with what is required to craft together the virtual machine including any additional hardware items such as network cards, drive controllers and anything that can be delivered as virtual hardware via the vSphere platform.

What wizardry is this?

The OVF deployment wizard in vSphere walks you through the configuration of specific virtual hardware including the selection of disk type, location, network type and location, IP configuration and also the VM location including resource pool, cluster and folder.

Very simply, you open the vSphere Client and under the File menu, select Deploy OVF Template. This will bring up the deployment wizard as you can see below.



The end result of the wizard is a fully functioning virtual guest. It doesn't get much simpler than that.

Why OVF?

A question that people may ask is “why use an OVF instead of just a VMDK and VMX file?”. The reason for the use of an OVF is to apply standardized deployment techniques to allow the deployer to make the choice about hardware specifics within their own environment.

Another clear reason is that the OVF standard is defined for open use across hypervisors. Yup, some people use other ones ☐

Using OVF makes the build and deployment of virtual appliances a snap, and for any vendor, enabling their customers is key when getting products to market.

What is the difference between OVF and a virtual appliance?

OVF is a virtual machine packaging method. The virtual appliance is the product itself. It is typically a blackbox type of build which would be not much different than a 1U server that you would have put into your physical network rack.

✘ Using VMware Studio (<http://www.vmware.com/products/studio/overview.html>) you can build your appliance using a supported operating system (Supported OS list [here](#)) and package your application build.

You can also deploy a vApp with an OVF so you are not limited to a single machine deployment. Using the first-boot configuration script you have plenty of options for post deployment configuration. There are also a lot of great resources for building your OVF so getting help is just a few clicks away.

Should OVF replace my VM Templates?

Unless you have a lot of pre-defined applications and packaged builds, you probably won't see the benefit of using OVF machines for your corporate environment. What you should concentrate on is the automation of your builds with other great tools like vCenter Orchestrator.

There may be some cases in your corporate environment where OVF is appropriate so perhaps you could give it a try and the worst case scenario is that you will become more familiar with build automation and deployment scripting, and nobody will lose in that scenario ☐

What about upgrades?

The upgrade process can be scary with any environment. Many virtual appliance vendors will provide a simple configuration backup process (export to file, FTP etc...) and you can simply deploy an upgrade in place. It is even possible to use Update Manager to upgrade your virtual appliances.

The long and the short of it is that virtual appliances are fast becoming a part of many production environments and it may be a great way for you to speed deployment and ease the installation process for products. Anything to reduce the workload is definitely a positive step in my opinion.



The VMware Appliance Marketplace

Now that you are ready to take a look, there is no better place to begin than the [VMware Virtual Appliance Marketplace](#). Through the marketplace site you can search for various application solutions from the VMware Certified vendors and from other supported partners who provide appliance based solutions.



So have a look around, and happy deploying!

[Deploying the VMware vCenter Virtual Appliance on OSX](#)



For those who want to leverage VMware vCenter for your lab, but you are running on Mac OSX with VMware Fusion, there are a couple of quick steps to get you working. The secondary title to this article could really be “How to deploy an OVF template on VMware Fusion”, but the specific goal I’m after is my vCenter appliance today.

I am assuming that you are already comfortable with VMware Fusion, and with administering your lab environment so this is really more of a quick walkthrough just to deploy the virtual appliance using command line tools.

The virtual appliance is available as an OVF file. I’ve used it for my production deployment, but for this case I want to use the same OVF package to deploy into my nifty little lab that I pack inside my Macbook. Here is the challenge: there is no native way to import or deploy the OVF template using VMware Fusion.

Luckily we have the OVFTool available which lets us use the same files we have by converting the OVF file into a VMX to be used by VMware Fusion. The OVFTool details and documentation are found here: <http://www.vmware.com/technical-resources/virtualization-topics/virtual-appliances/ovf>

First, you need to download the VMware vCenter virtual appliance files from the VMware download site. There are 3 files which will include the System disk, the Data disk and the OVF file which is what you would normally use to deploy into your ESX host using the Virtual Infrastructure Client.

The vCenter Server Appliance download can be found here: <https://my.vmware.com/group/vmware/details?downloadGroup=VC50&productId=229>

Secondly, you need to download and install the OVFTool package for OSX. The file for OSX is a tar.gz file which contains a PKG file. You will find that tar file

here: <http://communities.vmware.com/community/vmtn/server/vsphere/automationtools/ovf>

Once you unzip the file into a folder and run the PKG file to install the product it will place the

contents in the **/Applications/VMware OVF Tool/** folder.

Open a Terminal window. Change directory to the location of the OVF tool (note that it will show as **cd /Applications/VMware OVF Tool/** because of the spaces in the path).



Next you run the ovftool using the the command line of **ovftool source destination** which in this case I've stored the source file under my Downloads folder in a vCenter subfolder.



The tool will work its magic for a while and when it is done it will have crafted a ready to use VMX file and VMDK file set which you can now open with your VMware Fusion application. Because we have converted to a new VM file set, the files are duplicated so in this case you have to be sure you have space for another 5.5 GB or so. You will be notified of the successful completion:



Simply go to your VMware Fusion app now and on the File menu select Open



Point to your location that you have stored your new VMX and VMDK files. For my example I have simply left them in place. You will most likely want to move them to a folder with your other virtual machines.



Now we see the machine in our Library view and you may now start up the server. It is just that easy!

