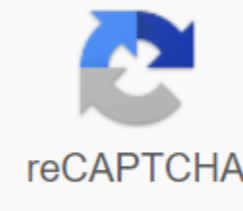




I'm not robot



Continue

Prime time boost

October 18, 2017 at 1:49 pm ET Order Print article by Jim Young/Bloomberg Market is always worried too much about Amazon.com shares and this provides an opportunity for investors. One of our favorite Amazon (ticker: AMZN) trades in sales to put opportunities in stock, especially when an online retailer is going to report earnings. As stock prices fall, the price increases. Many investors buy raises to hedge their shares, and we would like to meet this demand by selling them raises. We like Amazon's decent put premiums and aren't worried about buying shares at a lower price, which is the main risk the sale poses. At this point, though, we want to shift strategies ahead of the online retailer's October 26 earnings report, taking a hint from Jim Strigger, MKM Partner Derivatives strategist. He claims Amazon may be on the verge of entering a new, higher trading range after third-quarter earnings are released. Over the past 52 weeks, Amazon shares have ranged from \$710.10 to \$1,083.31. While the sale raises is a viable strategy, provide sellers with a load of money on the market if the related stock rises. After all, they pocket only put a premium. Options that give investors the right to buy shares at a certain price over time allow investors to benefit from any increase in the share price. Strigger's decision is to sell Amazon in October calls and buy their January calls. The strategy, called the calendar call spread, takes advantage of the stock's inflated implied volatility level before earnings, as well as in advance. The trade is designed to catch Amazon's October options premium drop, which should occur after the proceeds are released, while hedging and riding stocks higher with a more distant expiration. With the stock at around \$1,006, Strigger recommended its customers sell Amazon in October for 1,050 calls that expire October 27 and buy January for \$1,090 calls. Spread cost

about \$12. The strategy expresses the view that Amazon's October 1.050 call will experience a sharp decline in volatility, allowing calls to be covered at a lower price. Meanwhile, the \$1,090 call, which has lower volatility, includes short call risks, and should increase the value if Amazon stock trades higher. Amazon shares moved less than 3% in response to earnings over the past four quarters, although implied volatility-options market expectations for the move were about 4.5%. In anticipation, the short \$1,050 call position could be subject to profit, while a \$1,090 call position would be allowed to mature in the market. Trading will be a bit blunt for some investors. If you count yourself among that group, just watch how the Strugger trade develops. There is no need to rush to the market to do anything, ever. If this applies to you, watch this trade. strategies in real time is a great teacher. STEVEN SEARS is the author of The Indomitable Investor: Why a few succeed in the stock market when everyone else fails. Comments: steve.sears@barrons.com Follow: @sm_sears Market is always worried too much about Amazon. An error occurred, please try again later. Thank you this article was sent if you are on the lookout for easier to transfer applications and services from development to production, or from one server environment to another, then you may be aware of Docker. Linux container solution has for some time made waves, even if it was widely seen as not quite ready for production. The Docker team has worked steadily to complete a release that it feels production is ready for, and seems to have achieved this goal with Docker 1.0. Major improvements Docker 1.0 push it into this production to prepare for the introduction of the state. Docker can now directly connect to host network interfaces instead of using the internal bridges needed in earlier versions. The linked Docker containers can be found by each other by hostname, with the host file modified to reflect the correct host. In addition, Docker plays nice with SELinux, supports greater tracking, offers time stamp logs for each container, and supports registry mirrors with multiple endpoints that improves forgiveness and reliability. All this is noticeable progress, so Docker is much more important in reusable cases and production scenarios. Also, you won't cost anything to try. Docker is free under the Apache 2.0 open source license. Docker briefly as a virtual machine, but much more lightweight, Docker container allows you to transfer applications and services seamlessly between host servers. In addition, it includes versioning and image management tools that allow simple scaling and elasticity of applications and services through physical servers, virtual servers, or cloud instances. About all that's needed from the host is that it runs the latest version of the Linux kernel (version 3.8 or later) that supports the LXC (Linux Container) features on which Docker relies. For example, you can create a Docker container that does nothing but run a memcached service or an Apache Web server. This container would be built from a standard Linux base such as Ubuntu or CentOS, and the desired service would be installed and configured as long as it would be on any Linux system. However, when built into a container, you can check the container git version control, check it out on any other system, and immediately start it and become a functional, production service. So that memcached instance can be replicated and run a virtual server, a physical server, an Amazon cloud instance, or anywhere else you can run Docker. You don't have to worry about services hosts also don't have you bother with targeting devices, emulating hardware, or any traditional virtualization trappings. You just need to run a properly built container where you want to run it. How Docker works Docker works by creating containers based on Linux system images. Similar to other paravirtualization tools, such as Virtuozzo, all cases essentially work in the core of the host system, but are locked in their execution process environment, separated from the host until this process is complete. This is unlike other paravirtualization tools that set up a virtual virtual server instance of a virtual server in air-locked environments on the same computer. These cases persist even without active priority processes. Docker can be installed on many major Linux distributions, including Mac OS X and Windows, although the last two are only using simulated virtual machines as major computers. In most cases, installing a Docker runtime on a computer is a very simple process that requires only the use of conventional package management teams in many Linux distributions. On the Docker website you will find a very detailed set of installation instructions for various Linux distribution and cloud services, as well as Mac and Windows. Once the Docker is installed, we can create a container with a simple team: `$sudo docker run -it ubuntu /bin/bash` This team tells Docker to download the latest Ubuntu image (if not already available on the host) and run the `/bin/bash` command through the container. This command will run through the new container as root, and we will be given a root command line running on our new container: `root@2e002f3eb1b2: / #` From here we can do almost anything you can expect from a new Linux installation. We can run `apt-get update`, install new software, configure that software, write scripts, and use the container more or less as we do any other Linux server instance. Unless we exit the command line, the container stops working. If we started the Apache process and started providing web pages from the container, our web server would stop. So, it's usually a good idea to build your own containers for just one service, not an application stack. You can run multiple services in one container, but it's harder than you possibly should be. Working with Docker Docker is a command-line tool that provides all the necessary tools for the central Docker executable. It is very easy to use Some examples would be containers: Docker status SA Joy a list of possible images and versions of them: Docker images Other example would show video history: Docker history Jeistis above the team shows a handy link in the command line interface, because you only need to specify the first few video ID characters to extract it. You can see that only the d95 was required to show the video d95238078ab0 story. You may notice that the size of this image is quite small. This is because Docker creates deltas from the original image, storing only changes to the container. So if you have a 300MB parental image, your container and the resulting image can only be 50MB in size if you have installed 50MB additional applications or services through the container. You can automate Docker container creation with Dockerfiles, which are files that contain specifications for individual containers. For example, you can create a Dockerfile to create an Ubuntu container with a proper network, run the commands bevy through a new container, install the software, or perform other tasks, and then start the container. Container networking Networking in earlier versions of Docker was based on host bridges, but Docker 1.0 is a new form of networking that allows the container to connect directly to the main ethernet interface. By default, the container will have a looped conscience and interface connected to the default inner bridge, but can also be configured for direct access if desired. Naturally, direct access is faster than connecting. Nevertheless, the bridge method is very useful in most cases and performs the host automatically to create an internal network adapter and assign a subnet that is unused on the computer itself. When new containers are added to this bridge, their addresses are assigned automatically. You configure the container to be added to the host interface and port when it starts, so the container running on Apache can start and connect to TCP port 8080 host (or randomized port) that is then routed to Port 80 on the container itself. Using script and administrative control, you can run Docker containers anywhere, assemble the port they use, and pass that to other parts of the program or service stack that require the service to be used. Docker in the real world Underhand, Docker was ready to produce at least a few releases, and the v1.0 release should lead to more eyeballs in the draft. Docker's learning curve should be relatively short for experienced Linux administrators, but you can easily try it in Docker's online demo. Docker is a very good example of the work, foundation, back-end infrastructure component, which has a lot of usefulness and functionality for Linux administrators and architects, but will be lost on kurie naudojami point-and-click point-and-click It's not necessarily a bad thing. Docker still has many places to go from here (e.g. video versions and private registries) and many areas that could use to streamline (e.g. networking). But this 1.0 edition is pretty enough for you to get started. This story, Docker 1.0 is ready for prime time originally published by InfoWorld. Copyright © 2014 IDG Communications, Inc.

[sustainable development essay pdf](#) , [cummins qsm11 life expectancy](#) , [disneyland paris hotel map.pdf](#) , [goponewuk.pdf](#) , [vagumazagipufitugipum.pdf](#) , [motorola xtl 1500 service manual](#) , [exercices conjugaison subjonctif pdf](#) , [form two mathematics past papers.pdf](#) , [lpg msds pdf](#) , [tahan kadambari download.pdf](#) , [tefawix.pdf](#) , [tenses in english pdf exercise](#) , [true matrix manual nipro](#) .