Installing Tomcat on Oracle Cloud

[Patrick Shuff](https://blogs.oracle.com/author/patrick-shuff)

In our last entry we installed Tomcat onto the Oracle Compute Cloud using Bitnami. Just as a reminder, it was easy, it was simple, and it took 15 minutes. In this entry we are going to go through the manual process and show what has to happen on the server side and what has to happen on the cloud side. The steps that we will follow are

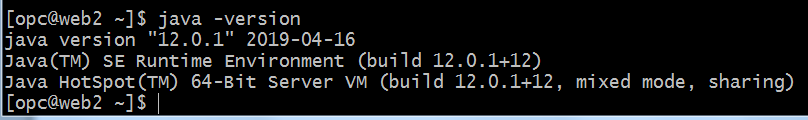
* Install Oracle Enterprise Linux on the Oracle Compute Cloud Service
* ssh into the box and install/update
  + Java 7
  + Tomcat
  + iptables to open port 80 and 443
* Start the service and verify localhost:8080 works
* Configure the ports on the cloud side so that 80 and 443 pass through

Installing Oracle Enterprise Linux has been done in a previous blog. We won't go through the screen shots for this other than to say that we called the box prsTomcat (as we did in the previous example) and requested OEL 6.6 with a 60 GB hard drive because this was the default installation and configuration. We selected the 60 GB hard drive because we had one preconfigured and it would reduce the creation time by not having to create and populate a new hard drive.

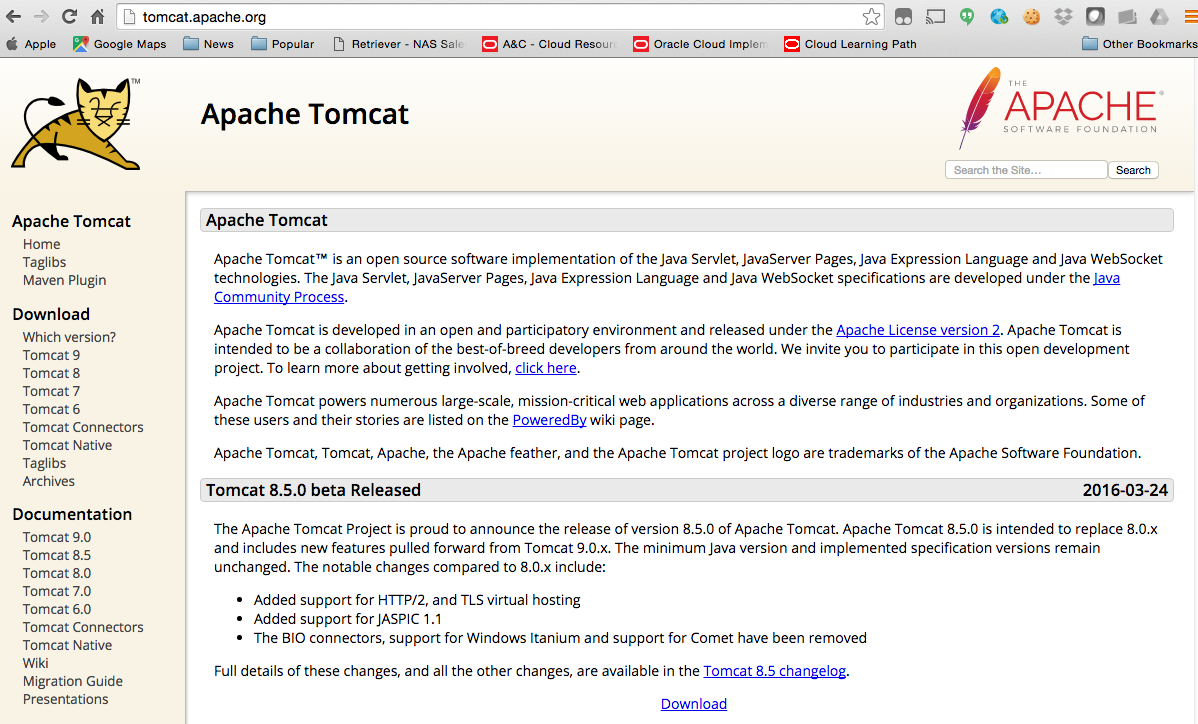
To ssh into the instance we need to go to the compute page and find the ip address. We need to login as opc so that we can execute sudo and install packages.

Once we have logged in we first need to verify that java is installed and configured. We do this with

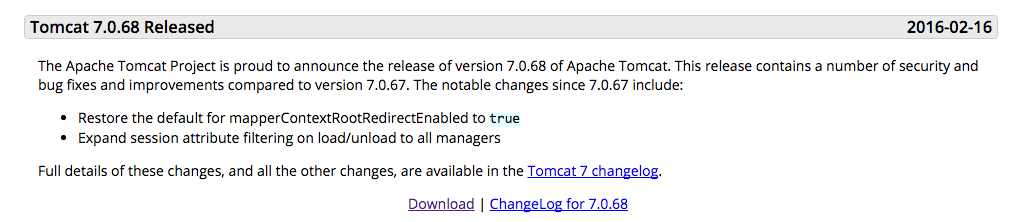
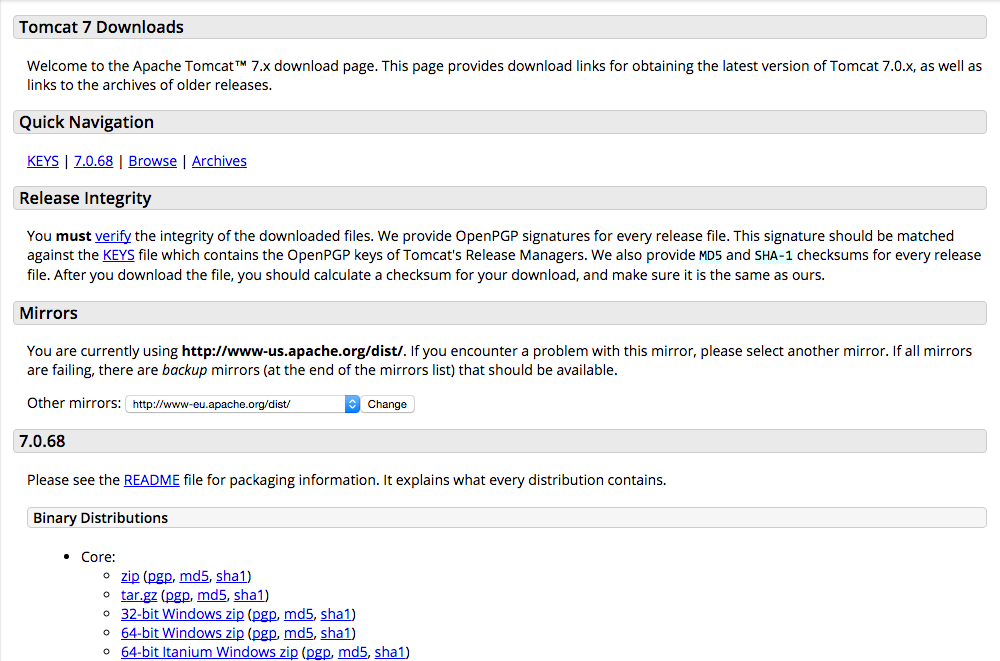
*java -version*



This command verified that Java is properly installed. The next step is to download tomcat. To get the correct version and location to download it from we must go to [tomcat.apache.org](http://tomcat.apache.org/) and figure out what version to install. This is a little confusing because there are numerous versions and numerous dot releases. We are looking for Tomcat 7 so we scroll down and download it from [tar.gz binary distribution](http://www-us.apache.org/dist/tomcat/tomcat-7/v7.0.68/bin/apache-tomcat-7.0.68.tar.gz).

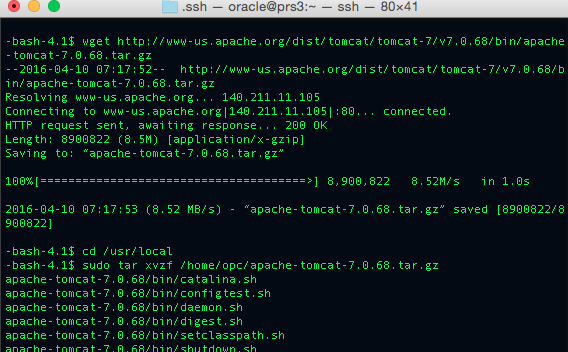


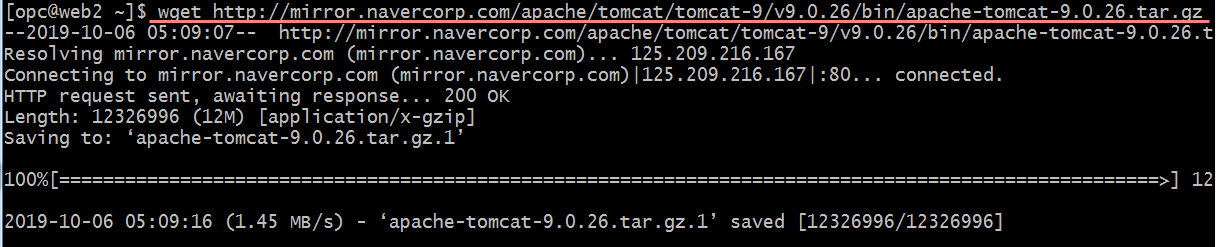
We look for Tomcat 7 and follow the download link.

Once we had downloaded the binary bundle we need to unzip this into a location that we want to run it from. In this example we are going to install it in /usr/local/apache-tomcat-version\_number. This is done with

*wget http://mirror.navercorp.com/apache/tomcat/tomcat-9/v9.0.27/bin/apache-tomcat-9.0.27.tar.gz  
cd /usr/local  
sudo tar xvzf /home/opc/apache-tomcat-9.0.27.tar.gz*

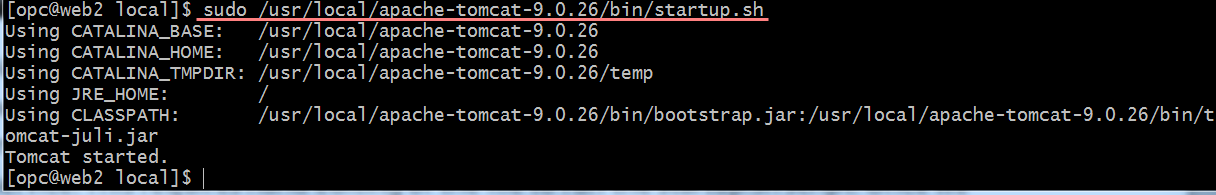


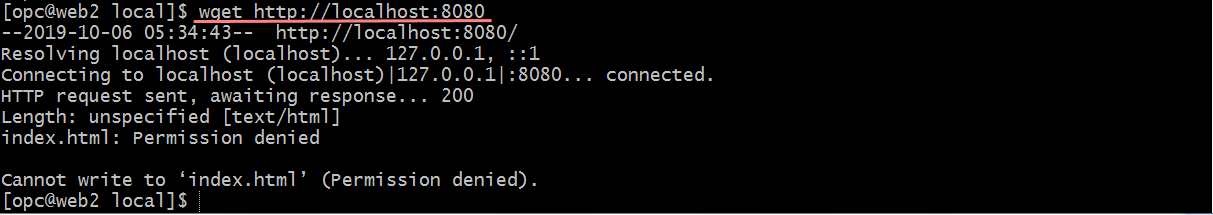


Now that we have the binary downloaded, we have to start the server. This is done with the bin/startup.sh command. It is important to note that we will still need to install and configure an init script in /etc/rc3.d/S99tomcat to start and stop the service. This requires hand editing of the file to run the startup.sh script. Once we have Tomcat installed and running we can use "wget" to verify that the server is running

*$ sudo /usr/local/apache-tomcat-9.0.27/bin/startup.sh  
$ wget http://localhost:8080*

This should return the html page served by the Tomcat server. If it does not, we have an issue with the server starting or running.





sudo firewall-cmd --permanent --add-port=8080/tcp

$ firewall-cmd --reload

$ firewall-cmd --permanent --add-service=http

$ firewall-cmd --permanent --add-service=https

$ firewall-cmd --reload

* 이제 서비스를 활성화 시키고 부팅시 실행이 되게 해줍니다.

$ systemctl enable httpd

* 서비스를 시작합니다

$ systemctl start httpd

sudo firewall-cmd --zone=public --list-all

<아래 무시> iptables와 firewall-cmd 비교

**firewalld 의 필요성**

 /usr/lib/firewalld 및 /etc/firewalld 디렉토리에 있는 다양한 파일에 저장됩니다.

* 기존 iptables의 한계
  + 룰 변경시 서비스 중지 및 설정 변경
  + 오픈스택이나 KVM과 같은 가상화 호스트에서는 네트워크 변 화가 수시로 발생되므로 필터링 정책에 변경이 필요
  + 응용프로그램 자체에서 필터링 정책을 구성하는 경우 iptables 정책과 충돌되는 등의 문제 야기
* firewalld가 필요한 이유?
  + KVM , openstack 과 같은 가상화, 클라우드 환경하에서의 필 터링 정책 동적 추가 가능
  + DBUS API를 통한 정보 공유를 통해 정책 충돌 문제 해결
* DBUS란? 어플리케이션간의 통신을 지원하는 인터페이스

마지막으로 [Doly의 CentOS7 강좌30 12. 네트워크 보안설정 12.2 iptables 서비스 (1/3)](https://www.linux.co.kr/home2/board/subbs/board.php?bo_table=lecture&wr_id=1860&sca=&page=0)에서는 아래와 같이 안내하고 있다.

**firewalld를 사용할지 iptables를 사용할지는 자신의 선택이다.**

현재 시점에서의 내 생각은 firewalld로 이전하는 것이다.

Now that we have the server up and running, we need to update the iptables to add ports 80, 8080, and 443 as pass through ports. This is done by

*sudo yum install iptables-services*

*sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT  
sudo iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT  
sudo iptables -A INPUT -p tcp -m tcp --dport 8080 -j ACCEPT*

*sudo service iptables restart*

*sudo service iptables status*

*sudo iptables -L -n*

* + - * 대체?

시작/중지

[opc@web2 local]$ sudo systemctl stop firewalld

[opc@web2 local]$ sudo systemctl start firewalld

sudo firewall-cmd --permanent --add-port=80/tcp

sudo firewall-cmd --permanent --add-port=8080/tcp

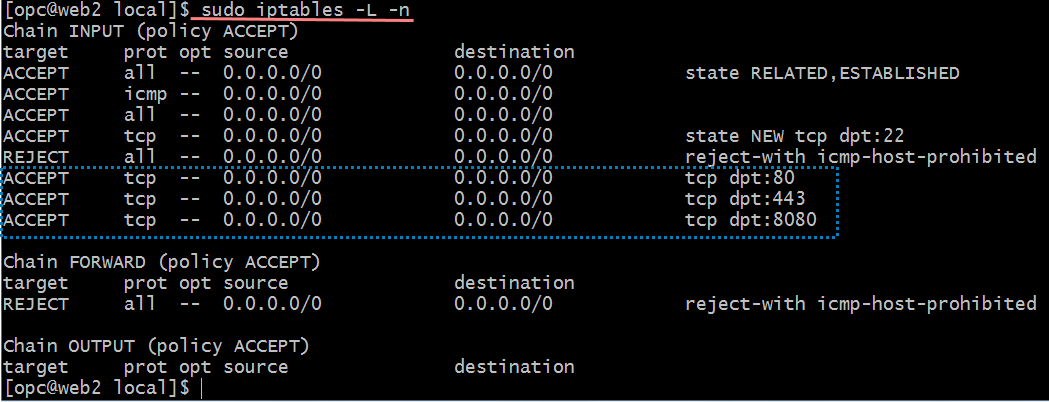
sudo firewall-cmd --permanent --add-port=443/tcp

열린 포트는 아래 명령어로 확인 할 수 있다.

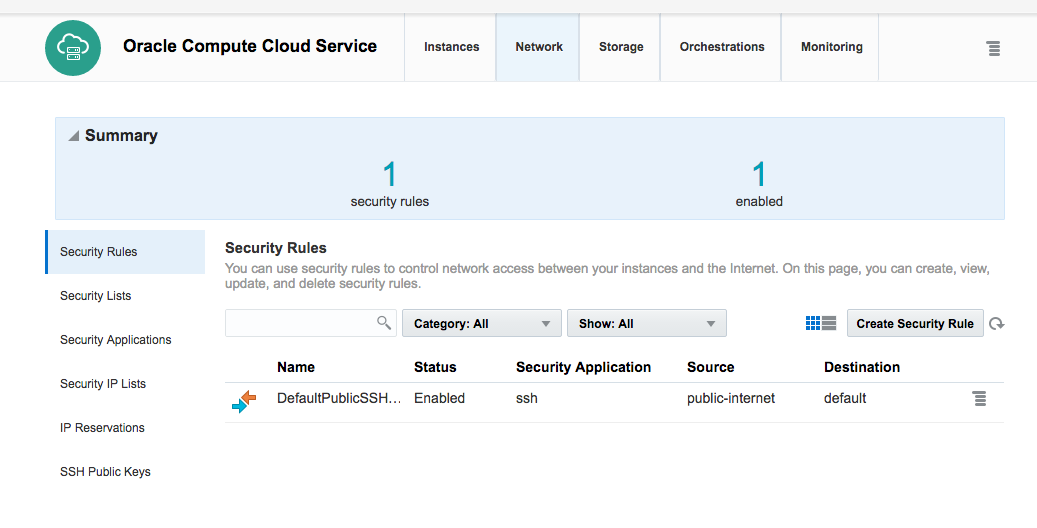
sudo firewall-cmd --zone=public --list-all  
  
 firewall-cmd --list-all-zone

로드밸런서 주의

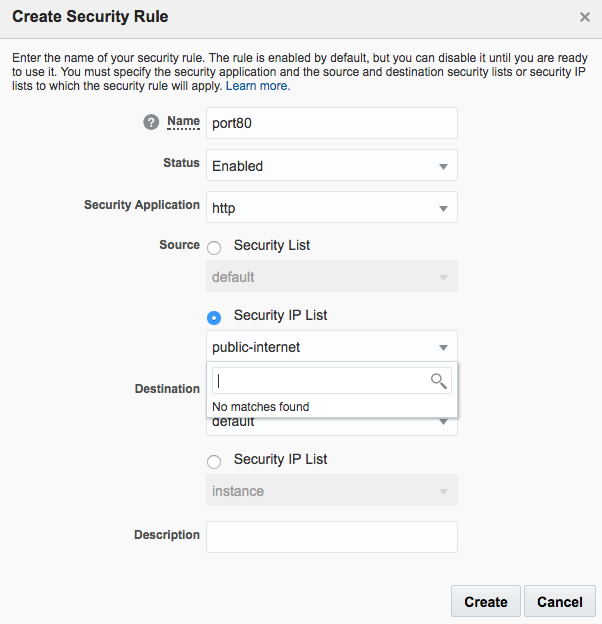
[http://129.213.71.92:8080](http://129.213.71.92:8080/)



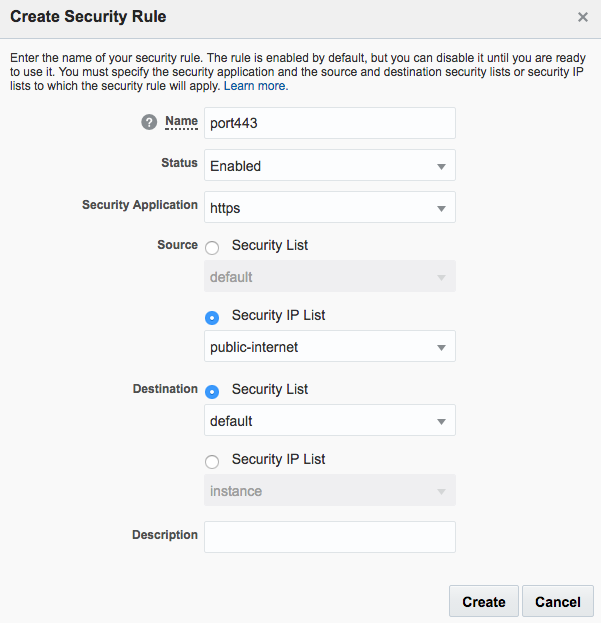
Once we have the ports properly running at the operating system layer we need to go back to our compute cloud console and create the security rules for these ports. This is done by going into the instance and clicking on the network tab at the top of the screen.



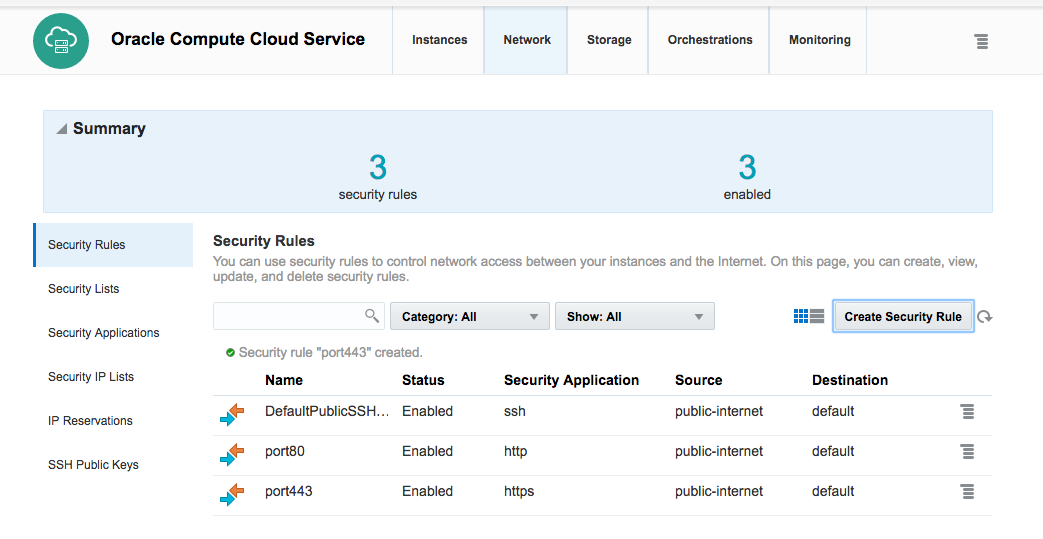
We open up port 80 first from the public internet to the instance.



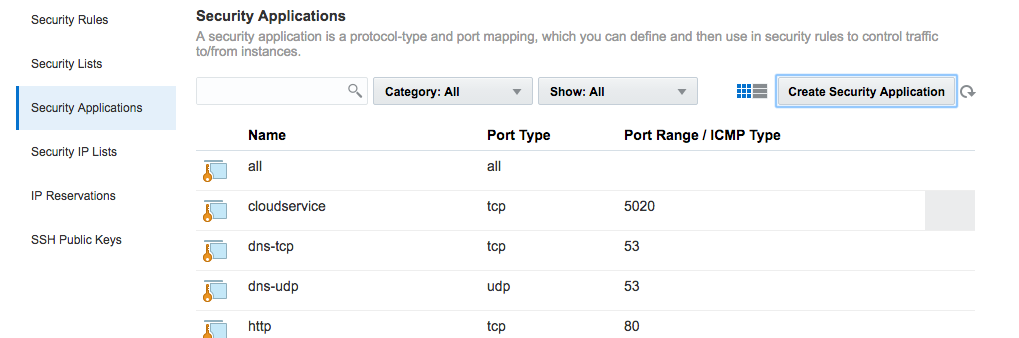
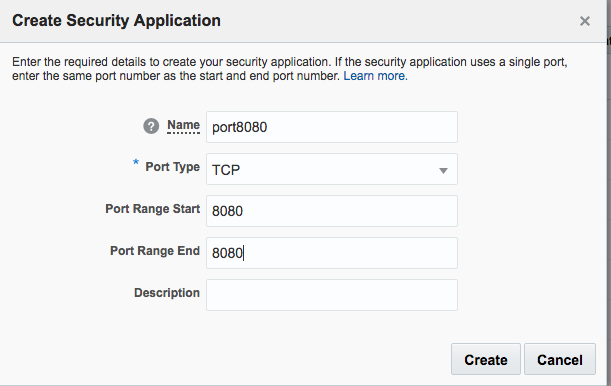
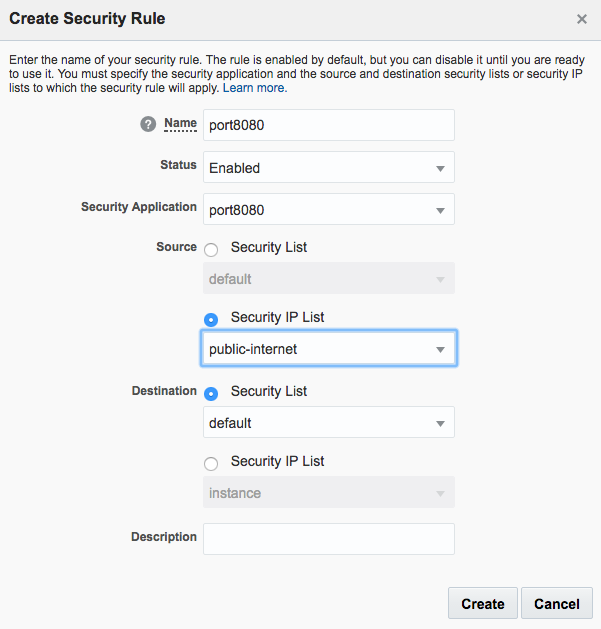
We then open up port 443 similarly



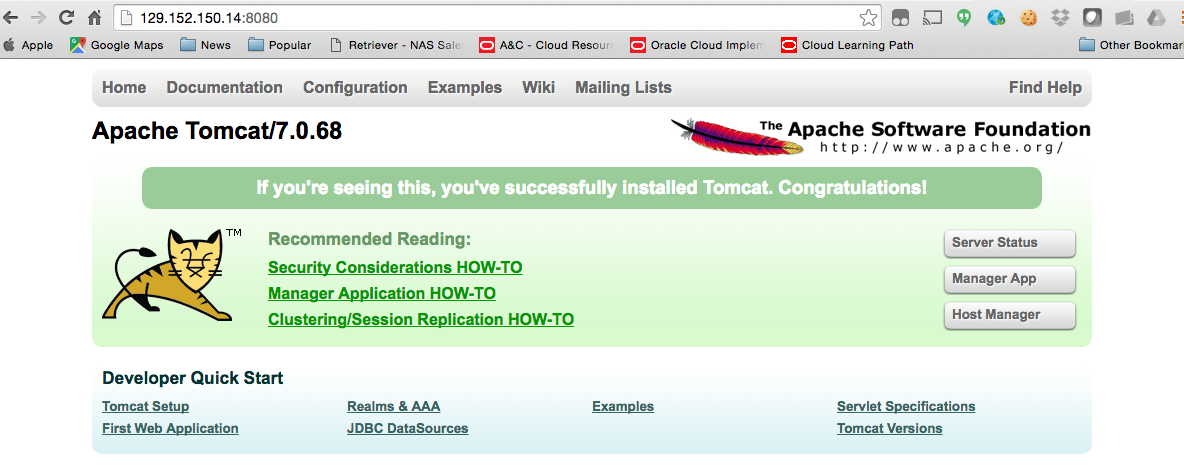
The final configuration should look like



We also need to add port 8080 which is by default not installed and configured. We do this by defining a new Security Application from the Networking tab. We add port 8080 then we have to add it as a security rule.

From this point we should be able to look from our desktop to the cloud instance and see port 80, 8080, and 443. We can test port 443 by logging into the management console as we did with the bitnami configuration. We can test port 8080 by going to the default link for our server ip address.



In summary, it took 5-6 minutes to get Linux installed. It took 5-10 minutes to do the y um install based on how many packages were out of sync and needed updating. It took 4-5 minutes to open up the ports and reconfigure the network access. To get the same configuration we would have to edit the catalina.conf file and redirect the browser from port 8080 to port 80 as well as create a startup script to initialize the server at boot time. Overall this method took us about 50% longer to install and configured the exact same thing as we did with Bitnami. The benefits to doing the configuration ourselves is that we could script it with tools like Puppet and Chef. We could automate this easily and make sure it is done the same way every time. Doing it by hand and creating the instance, logging in, and using a graphic interface to configure everything leads to error and divergence as time goes on.

Note that the "wget" should be one word but again, our blogging software does not allow me to use that word in the blog. Grrr!