**UNIT 13 ELK STACK DEPLOYMENT – LINUX COMMANDS TO CREATE THE DEPLOYMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **12.1** |  |  | **INTRO TO CLOUD COMPUTING** |  |
| 12.1 | 3 | LINUX | Generate an SSH public/private key pair using terminal on the local host machine.  Set the SSH password for the JumpBox and Web VMs in Azure using the SSH public key. | myterminal:~$ ssh-keygen  myterminal:~$ cat .ssh/id\_rsa.pub  myterminal:~$ cp .ssh/id\_rsa.pub |
| **12.2** |  |  | **CLOUD SYSTEMS MANAGEMENT** |  |
| 12.2 | 2 | LINUX | Connect from local host machine to the JumpBox VM using SSH on port 22.  Once connected to the JumpBox VM, check sudo permissions. | myterminal:~$ ssh azadmin@52.187.237.72  **azadmin@JumpBox2**:**~**$ sudo -l |
| 12.2 | 3 | LINUX | Install Docker onto the Jumpbox VM. | **azadmin@JumpBox2**:**~**$ sudo apt update  **azadmin@JumpBox2**:**~**$ sudo apt install docker.io |
| 12.2 | 3 | LINUX | Once Docker is installed, pull the cyberxsecurity/ansible container onto the Jumpbox VM. | **azadmin@JumpBox2**:**~**$ sudo docker pull cyberxsecurity/ansible. |
| 12.2 | 3 | LINUX | Launch the Ansible container in a bash shell and connect to it. | **azadmin@JumpBox2**:**~**$ docker run -ti cyberxsecurity/ansible:latest bash |
| 12.2 | 3 | LINUX | One it has been successfully launched, exit the container. | root@79af822c5787:~# exit |
| 12.2 | 3 | AZURE | Create a new Network Security Group Rule for the RedTeam which allows the JumpBox full access to the Vnet |  |
| 12.2 | 4 | LINUX | Find the previously installed cyberxsecurity/ansible container and connect with it.  *Note – the image for the cyberxsecurity/ansible container is* ***cool\_saha*** | **azadmin@JumpBox2**:**~**$ sudo docker container list -a  **azadmin@JumpBox2**:**~**$ docker run -it cyberxsecurity/ansible /bin/bash |
| 12.2 | 4 | LINUX | Generate a new SSH public/private key pair from inside the Ansible container and reset the VM passwords with the new public key. | root@79af822c5787:~# ssh-keygen  root@79af822c5787:~# cat .ssh/id\_rsa.pub  root@79af822c5787:~# cp .ssh/id\_rsa.pub |
| 12.2 | 4 | LINUX | Test connection from the Ansible container to the Web-VMs using ping.  Access the Web-VMs from the Ansible container using SSH. | **Web-1:**  root@79af822c5787:~# ping 10.1.0.5  root@79af822c5787:~# ssh azadmin@10.1.0.5  **Web-2:**  root@79af822c5787:~# ping 10.1.0.6  root@79af822c5787:~# ssh azadmin@10.1.0.6 |
| 12.2 | 4 | LINUX | Locate the Ansible **hosts** file | root@79af822c5787:~# ls /etc/ansible/  **…hosts…** |
| 12.2 | 4 | LINUX | Update the Ansible **hosts** file to include IPs for the Web-VMs.  Note – the python line needs to be included with each IP: ansible\_python\_interpreter=/usr/bin/python3 | root@79af822c5787:~# nano /etc/ansible/hosts  Uncomment the [webservers] header line  Add the Web-VM IPs:  10.1.0.5 ansible\_python\_interpreter=/usr/bin/python3  10.1.0.6 ansible\_python\_interpreter=/usr/bin/python3  Save changes and exit the nano file:  ^C > Y > enter |
| 12.2 | 4 | LINUX | Locate the Ansible **config** file | root@79af822c5787:~# ls /etc/ansible/  **…ansible.config…** |
| 12.2 | 4 | LINUX | Update the remote\_user in the Ansible **config** file to include azadmin, the admin username for the JumpBox and Web VMs. | root@79af822c5787:~# nano /etc/ansible/ansible.cfg  Uncomment the remote\_user line and replace root with azadmin:  remote\_user = azadmin  Save changes and exit the nano file:  ^C > Y > enter |
| 12.2 | 4 | LINUX | Check updates to the **hosts** and **config** files by testing connections to the VMs from the Ansible container. | root@79af822c5787:~# ansible all -m ping |
| **12.3** |  |  | **LOAD BALANCING & REDUNDANCY** |  |
| 12.3 | 1 | LINUX | Connect to the Jump Box VM using terminal on the host machine . | myterminal:~$ ssh azadmin@52.187.237.72 |
| 12.3 | 1 | LINUX | Once in the JumpBox VM, in the previously installed Ansible container find the image. Start it and then connect with it. | **azadmin@JumpBox2**:**~**$ docker container list -a  **azadmin@JumpBox2**:**~**$ docker start cool\_saha  **azadmin@JumpBox2**:**~**$ docker attach cool\_saha |
| 12.3 | 1 | YAML FILE | Create an Ansible playbook named pentest.yml to install Docker and configure the Web-VMs with the DVWA web app.   * Use apt module to install docker.io and python3- * Update the cache * Use the Ansible pip module to install docker * Install the cyberxsecurity/dvwa container. Use port 80 on the container to port 80 on the host. * Set the restart policy so that the container always restarts with the VM. * Use the systemd module to restart the docker service when the machine reboots.   *NB. To check syntax of YAML files, use YAMLlint: www.yamllint.com* | root@79af822c5787:~# nano /etc/ansible/pentest.yml  INSERT LINK TO ANSIBLE PLAYBOOK **pentest.yml** – 12.3 ACTIVITY 1 |
| 12.3 | 1 | LINUX | Run the Ansible pentest.yml playbook. | root@79af822c5787:~# ansible-playbook /etc/ansible/pentest.yml |
| 12.3 | 1 | LINUX | Test that DVWA is running on the new VMs.  Use SSH to connect with each of the Web VMs from the Ansible container.  Run curl localhost/setup.php to test the connection to the DVWA container is working. | root@79af822c5787:~# ssh [azadmin@10.1.0.5](mailto:azadmin@10.1.0.5)  Then run:  azadmin@Web-1:~$ curl localhost/setup.php  To yield the following HTML result :  Graphical user interface, text, application  Description automatically generated  Repeat for Web-VM (Web-2): 10.1.0.6 |
| 12.3 | 4 | AZURE | Set up an additional Web-VM (Web-3) in Azure. |  |
| 12.3 | 4 | LINUX | In order to complete setup, connect to the JumpBox from terminal on the host machine and then start the existing Ansible container to access the public SSH key. | myterminal:~$ ssh azadmin@52.187.237.72  **azadmin@JumpBox2**:**~**$ docker start cool\_saha  **azadmin@JumpBox2**:**~**$ docker attach cool\_saha  root@79af822c5787:~# cat .ssh/id\_rsa.pub  root@79af822c5787:~# cp .ssh/id\_rsa.pub |
| 12.3 | 4 | LINUX | Once the new Web-VM with internal IP 10.1.0.7 is set up in Azure, test the connection using SSH.  Once the connection is established, exit the Web-VM. | root@79af822c5787:~# ssh azadmin@10.1.0.7  azadmin@Web-3:~$ exit |
| 12.3 | 4 | LINUX | Update the Ansible **hosts** file to include the IP for the new Web-VM.  This needs to include the python line: ansible\_python\_interpreter=/usr/bin/python3 | root@79af822c5787:~# nano /etc/ansible/hosts  Add the Web-VM IP underneath IPs for the existing Web-VMs:  10.1.0.7 ansible\_python\_interpreter=/usr/bin/python3  Save changes and exit the nano file:  ^C > Y > enter |
| 12.3 | 4 | LINUX | Check update to the Ansible **hosts** file using ping. | root@79af822c5787:~# ansible all -m ping |
| 12.3 | 4 | LINUX | Run the Ansible playbook named pentest.yml to install Docker and configure the new Web-VM with the DVWA web app. | root@79af822c5787:~# ansible-playbook /etc/ansible/pentest.yml |
| 12.3 | 4 | LINUX | Test that DVWA is running on the new VM.  Use SSH to connect to the new Web VM from the Ansible container, then run the curl command to test the connection to the DVWA container.  Exit the Web-3 VM. | root@79af822c5787:~# ssh azadin@10.1.0.7  Then run:  azadmin@Web-3:~$ curl localhost/setup.php  To yield the following HTML result :  Graphical user interface, text, application  Description automatically generated |
| **12.4** |  |  | **TESTING REDUNDANT SYSTEMS** |  |
| 12.4 | 2 | LINUX | Gather the hostname of each of the DVWA containers running on the Web-VMs.  **Results:**  Web-1: 4874702d5ba7  Web-2: 0fb1ec00aad  Web-3: e6oa4oc8f9b7 | For each Web-VM:  azadmin@Web-1:~$ sudo docker container list-a  azadmin@Web-1:~$ sudo docker start <container>  azadmin@Web-1:~$ sudo docker attach <container>  root@4874702d5ba7:~$ hostname |
| **13.1** |  |  | **ELK INSTALLATION** |  |
| 13.1 | 1 | AZURE | Set up a new ELK-STACK VM in Azure in the existing Resource Group using a new region and separate Vnet. |  |
| 13.1 | 1 | LINUX | In order to complete setup, connect to the JumpBox from terminal on the host machine and then start the existing Ansible container to access the public SSH key. | myterminal:~$ ssh azadmin@52.187.237.72  **azadmin@JumpBox2**:**~**$ docker start cool\_saha  **azadmin@JumpBox2**:**~**$ docker attach cool\_saha  root@79af822c5787:~# cat .ssh/id\_rsa.pub  root@79af822c5787:~# cp .ssh/id\_rsa.pub |
| 13.1 | 2 | LINUX | Update the Ansible **hosts** file to include the new ELK-VM.  Create a separate group heading, [elk].  Add the IP for the new ELK-VM: 10.0.0.4.  Include the python line: ansible\_python\_interpreter=/usr/bin/python3 | root@79af822c5787:~# nano /etc/ansible/hosts  Add the ELK-VM IP underneath a new ELK group heading:  [elk]  10.0.0.4 ansible\_python\_interpreter=/usr/bin/python3  Save changes and exit the nano file:  ^C > Y > enter |
| 13.1 | 3 | YAML | Create an Ansible playbook in YAML to configure the new ELK-VM server.   * This playbook needs to specify the applicable group (ie. elk. * In order to run the ELK container virtual memory needs to be increased. * Install docker.io and python3-pip and docker. * After Docker is installed, download and run the sebp/elk:761 container. * The container should bee started with the following ports:   5601:5601  9200:9200  5044:5044  se port 80 on the container to port 80 on the host.   * Use the systemd module to restart the docker service when the machine reboots.   *NB. To check syntax of YAML files, use YAMLlint: www.yamllint.com* | root@79af822c5787:~# nano /etc/ansible/install-elk.yml  INSERT LINK TO ANSIBLE PLAYBOOK **install-elk.yml** – 13.1 ACTIVITY 3 |
| 13.1 | 4 | LINUX | Run the Ansible install-elk.yml playbook. | root@79af822c5787:~# ansible-playbook /etc/ansible/install-elk.yml |
| 13.1 | 5 | LINUX | After the playbook has run, SSH to the ELK-VM and double check that the elk-docker container is running.  Take a screenshot of the result. | root@79af822c5787:~# ssh azadmin@10.0.0.4  Then run:  sudo docker ps  Take a screenshot of the result.  **INSERT LINK** |
| 13.1 | 6 | AZURE | Create a new incoming rule for the new Network Security Group which allows TCP traffic over port 5601 from the local host address. |  |
| 13.1 | 7 | KIBANA HOME PAGE | Test the setup is working correctly by navigating to the Kibana home page using the ELK-VM public IP. | http://40.87.108.196:5601/app/kibana#/home |
| **13.2** |  |  | **FILEBEAT INSTALLATION** |  |
| 13.2 | 1 | LINUX | Navigate into the ELK-VM and start the docker container to check that the ELK server container is up and running, then exit. | myterminal:~$ ssh azadmin@52.187.237.72  **azadmin@JumpBox2**:**~**$ docker start cool\_saha  **azadmin@JumpBox2**:**~**$ docker attach cool\_saha  root@79af822c5787:~# ssh [azadmin@10.0.0.4](mailto:azadmin@10.0.0.4)  azadmin@ELK-VM:~$ docker container list -a  azadmin@ELK-VM:~$ exit |
| 13.2 | 2 | LINUX | Create a Filebeat configuration file:  - Navigate into the Jump Box  - Open the Ansible container  - Copy the filebeat-config.yml configuration template using curl into the etc/ansible/ folder | **azadmin@JumpBox2**:**~**$ docker start cool\_saha  **azadmin@JumpBox2**:**~**$ docker attach cool\_saha  root@79af822c5787:~# curl https://gist.githubusercontent.com/slape/5cc350109583af6cbe577bbcc0710c93/raw/eca603b72586fbe148c11f9c87bf96a63cb25760/Filebeat >> /etc/ansible/filebeat-config.yml |
| 13.2 | 3 | LINUX | Open the filebeat-config.yml in nano and edit it as follows:  - Update line 1106 and replace the IP with the private IP of the ELK machine  - Update line 1806 and replace the IP with the private IP of the ELK machine  - Save the update configuration file by making a copy to the /etc/ansible/files/ folder | root@79af822c5787:~# nano /etc/ansible/filebeat-config.yml  #1106  output.elasticsearch:  hosts: ["10.1.0.4:9200"]  username: "elastic"  password: "changeme"  #1186  setup.kibana:  host: "10.1.0.4:5601"  root@79af822c5787:~# cp /etc/ansible/filebeat-config.yml /etc/ansible/files/filebeat-config.yml. |
| 13.2 | 3 | LINUX | Create a Filebeat installation playbook:  Download the .deb file from [artifacts.elastic.co](https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.0-amd64.deb).and then install it using the dpkg command. | root@79af822c5787:~# dpkg -i filebeat-7.4.0-amd64.deb |
| 13.2 | 3 | LINUX | Update the filebeat-playbook.yml and locate it in the etc/ansible/roles/ folder | INSERT LINK TO filebeat-playbook.yml |
| 13.2 | 3 | LINUX | Run the playbook  To check if successfully installed, return to the ELK Stack homepage and scroll to Step5: Module to ‘Check Data’. It should be receiving logs. | root@79af822c5787:~# ansible-playbook filebeat-playbook.yml |
|  |  |  |  |  |