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
Install Docker onto the Jumpbox VM.	azadmin@JumpBox2:~$ sudo apt update azadmin@JumpBox2:~$ sudo apt install docker.io
Once Docker is installed, pull the cyberxsecurity/ansible container onto the Jumpbox VM.	azadmin@JumpBox2:~$ sudo docker pull cyberxsecurity/ansible.
Launch the Ansible container in a bash shell and connect to it.	azadmin@JumpBox2:~$ docker run -ti cyberxsecurity/ansible:latest bash
One it has been successfully launched, exit the container.	root@79af822c5787:~# exit
Create a new Network Security Group Rule for the RedTeam which allows the JumpBox full access to the Vnet	
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
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
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
Locate the Ansible hosts file	root@79af822c5787:~# ls /etc/ansible/ …hosts…
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
Locate the Ansible config file	root@79af822c5787:~# ls /etc/ansible/ …ansible.config…
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

Check updates to the hosts and config files by testing connections to the VMs from the Ansible container.	root@79af822c5787:~# ansible all -m ping
Create an Ansible playbook 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/config-WebVMs.yml
Run the Ansible pentest.yml playbook.	root@79af822c5787:~# ansible-playbook /etc/ansible/pentest.yml
Set up a new ELK-STACK VM in Azure in the existing Resource Group using a new region and separate Vnet.	
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
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
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.	root@79af822c5787:~# nano /etc/ansible/install-elk.yml Playbook: install-elk.yml

NB. To check syntax of YAML files, use YAMLlint: www.yamllint.com	
Run the Ansible install-elk.yml playbook.	root@79af822c5787:~# ansible-playbook /etc/ansible/install-elk.yml
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.
Create a new incoming rule for the new Network Security Group which allows TCP traffic over port 5601 from the local host address.	
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
Navigate back into the ELK-VM and start the docker container to check that the ELK server container is up and running, then exit.	root@79af822c5787:~# ssh azadmin@10.0.0.4 azadmin@ELK-VM:~$ docker container list -a azadmin@ELK-VM:~$ exit
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
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.
Create a Filebeat installation playbook: Download the .deb file from artifacts.elastic.co.and then install it using the dpkg command.	root@79af822c5787:~# dpkg -i filebeat-7.4.0-amd64.deb
Update the filebeat-playbook.yml and locate it in the etc/ansible/roles/ folder	Playbook: filebeat-playbook.yml
Run the playbook	root@79af822c5787:~# ansible-playbook filebeat-playbook.yml
To check if successfully installed, return to the Kibana homepage and scroll to Step5: Module to 'Check Data'. It should be receiving logs.	