**Sanjay Chadha HW**

## Day 1 Activity File: ELK Installation

---

### Instructions

#### 1. Creating a New VM

Make sure that you are logged into your personal Azure account, where your cloud security unit VMs are located.

- Create a new Ubuntu VM in your virtual network with the following configurations:

- \*\*RAM\*\*: 4 GB+

- \*\*IP Address\*\*: The VM must have a public IP address.

- \*\*Networking\*\*: The VM must be in the same VNet and network security group (NSG) as the machines created in the cloud security week.

- \*\*Access\*\*: The VM must use the same SSH keys as the Ansible container and your existing jump box. Also, ensure that the NSG only allows access to this new VM from your jump box.

Private IP 10.0.0.4

Jumpbox and also mapped in NSG as source for virtualnetworkSSH



Home public IP shown as .0.0 to mask the full IP, in NSG on Azure full IP address included.



- After creating the new VM in Azure, verify that it works as expected by connecting via SSH from your jump box VM.

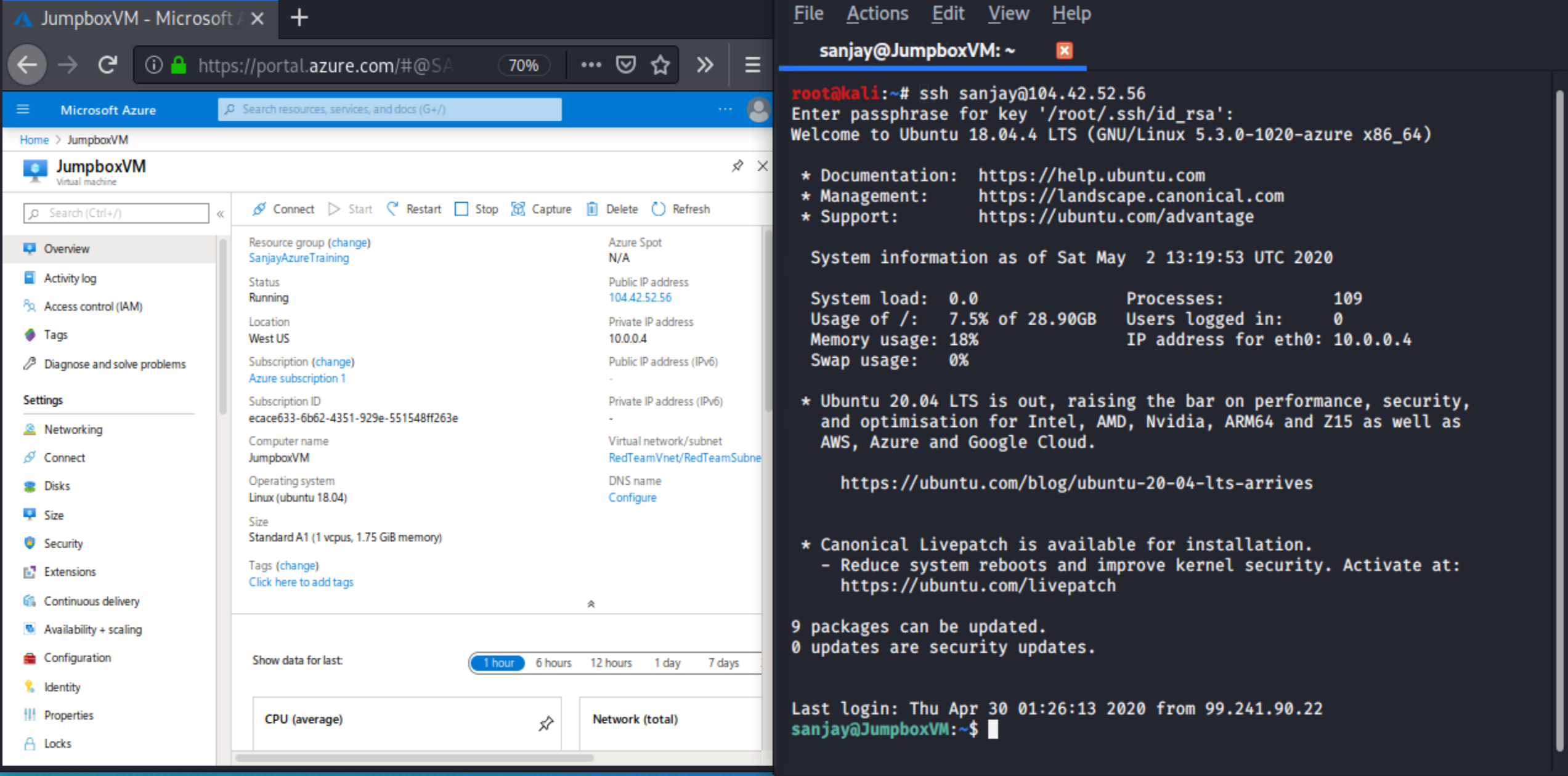
- Open a terminal on your computer and SSH into the jump box.

- From the jump box, SSH into the new VM you just created.

If the connection succeeds, you are ready to move on to the next step. If not, verify that you used the correct SSH key. If the problem persists, ask a partner or staff member for help.

Connection successful

Connect with public IP – secure connection established.



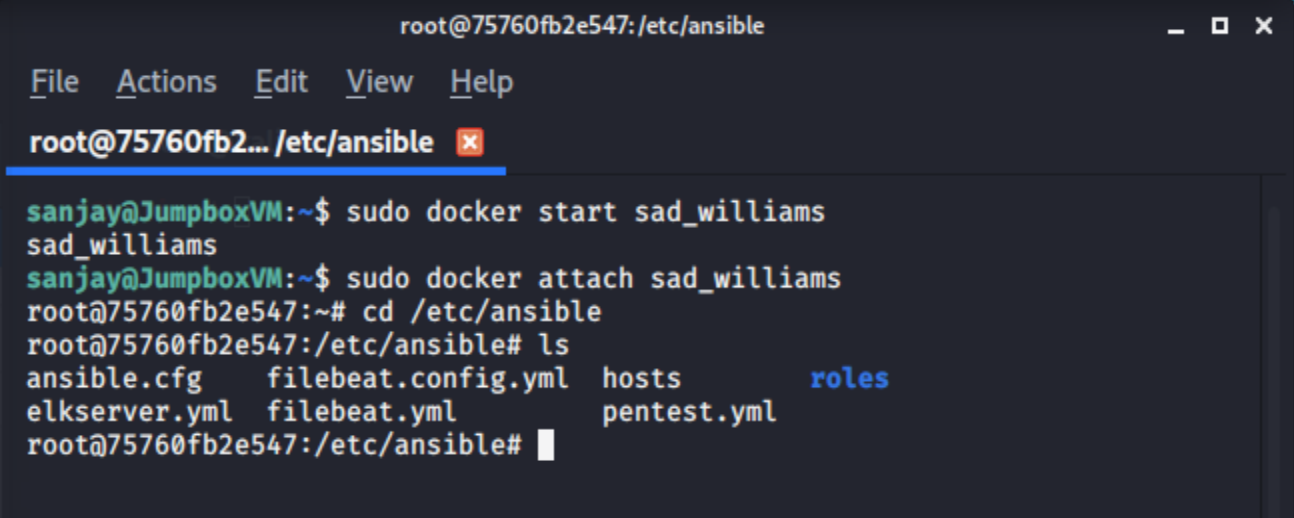
#### 2. Downloading and Configuring the Container

In this step, you will use Ansible to configure the newly created VM. You will:

- Start your Ansible container and log in.

- Add the new VM to Ansible's `hosts` file.

Hosts file identified



- Create a playbook that installs Docker and configures the container.

- Run the playbook to launch the container.

You just created a new VM, which you'll use to run your ELK stack. In order to use Ansible to configure this machine, you must add it to the list of machines Ansible can discover and connect to.

- This list is Ansible's \*\*inventory\*\*, and is stored in the `hosts` text file:

```

# /etc/ansible/hosts

[webservers]

192.168.0.1

192.168.0.2

[othermachines]

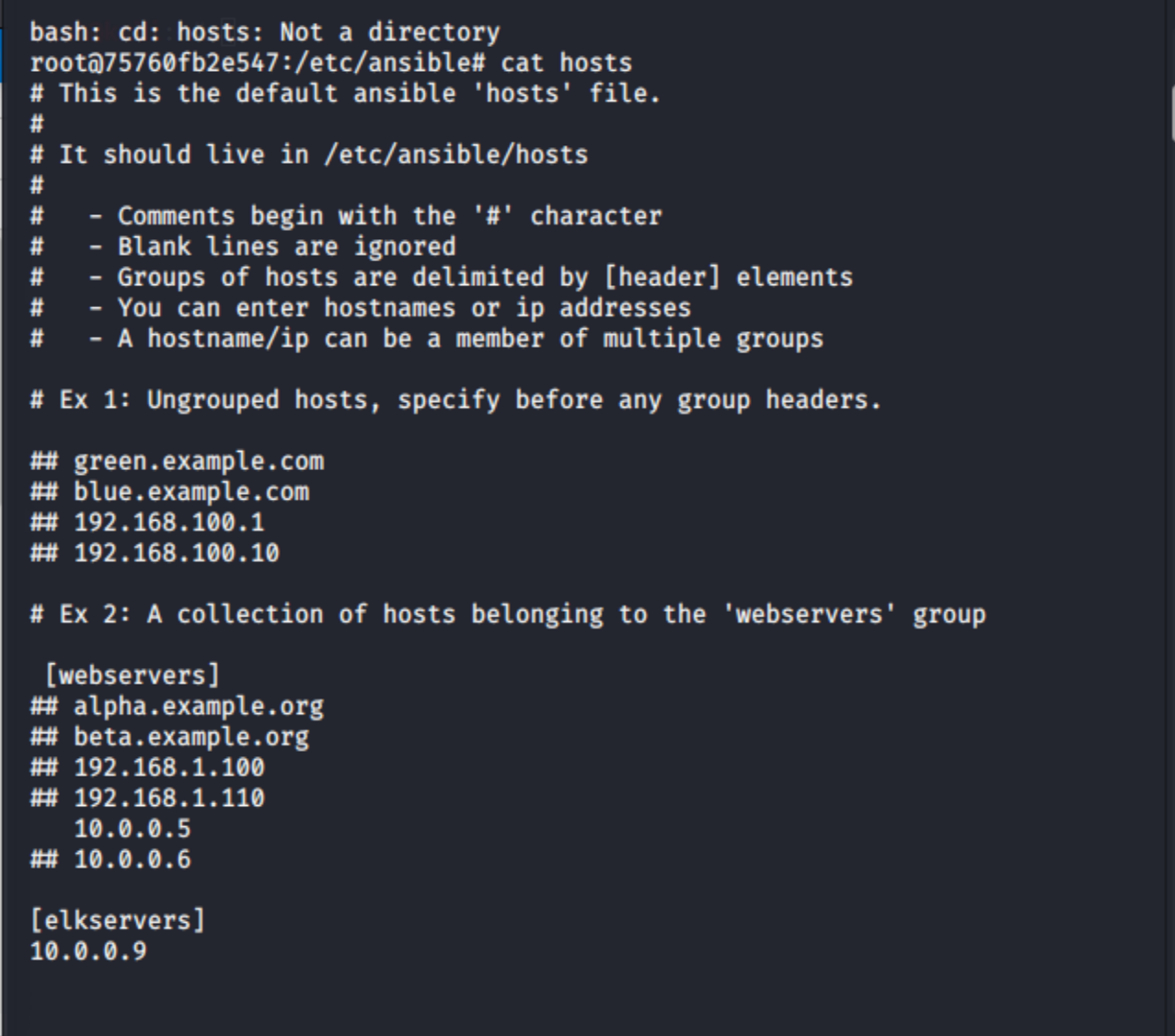
192.168.0.3

192.168.0.4

```

Added as list of machines

10.0.0.5



- `[webservers]` and `[othermachines]` are \*\*groups\*\*. When you run playbooks with Ansible, you specify which group to run them on. This allows you to run certain playbooks on some machines, but not on others.

\*\*Note:\*\* You do not need to include a group titled `[othermachines]` in your `hosts` file. This is included above as an example.

You should already be in your Ansible VM. Add an `[elkservers]` group to your Ansible VM's `hosts` file by following the steps below on the command line:

- Edit the inventory file `nano /etc/ansible/hosts`.

- Add a group called `[elkservers]` and specify the IP address of the VM you just created in Azure.

\*\*Note:\*\* If you get stuck, consult the starter `hosts` file:

- [Starter Host File](Activities/Stu\_Day\_One/Unsolved/Resources/hosts.yml)

##### Specifying Targets in Playbooks

Once you've created the `[elkservers]` group, you'll create a playbook to configure it.

- Today, you'll write a play to configure the ELK server. This play should only run on the VM in the `[elkservers]` group. Recall that you can use the `hosts` option in Ansible to specify which machines to run a set of tasks against.

sudo nano /etc/sysctl.co

```yaml

- hosts: elkservers

- become: True

- tasks:

- name: Install Packages

# Etc...

```

- Here, the `hosts` option specifies that these `tasks` should only be run on the machines in the `elkservers` group.

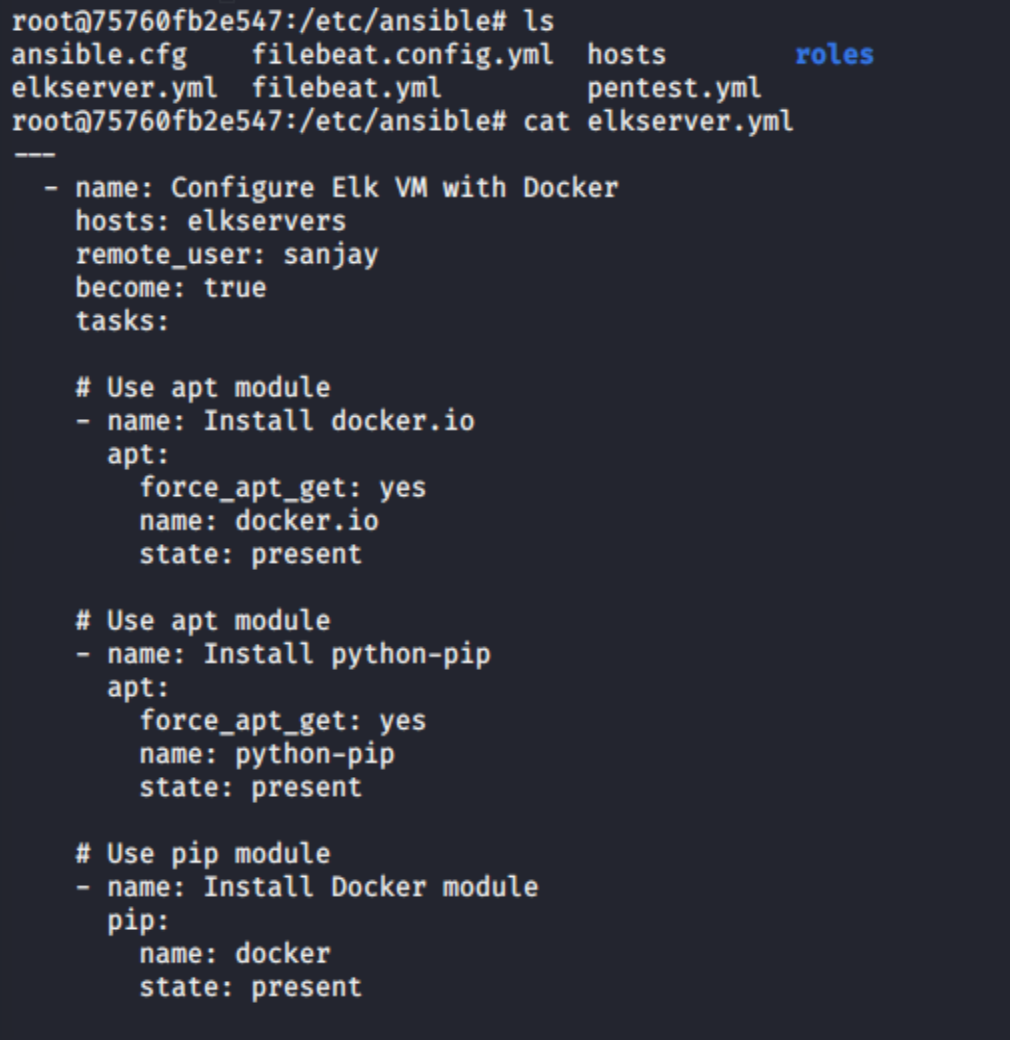
To create this playbook, continue using your terminal to complete the following steps:

- Verify you are in your Ansible container.

- Create a new playbook: `touch /etc/ansible/install-elk.yml`.

- Ensure that the header of the playbook looks like the YAML snippet above. Specifically, it must specify `elkservers` as the target hosts.

Done



Write `tasks` that do the following:

- Runs the following command through the shell: `sysctl -w vm.max\_map\_count=262144`

- This command configures the target VM (the machine being configured) to use more memory. This improves performance of the ELK container running inside.

- Installs the following packages:

- `docker.io`: The Docker engine, used for running containers.

- `python-pip`: Package used to install Python software.

- `docker`: Python client for Docker. Required by ELK.

- Downloads the Docker container called `sebp/elk`.

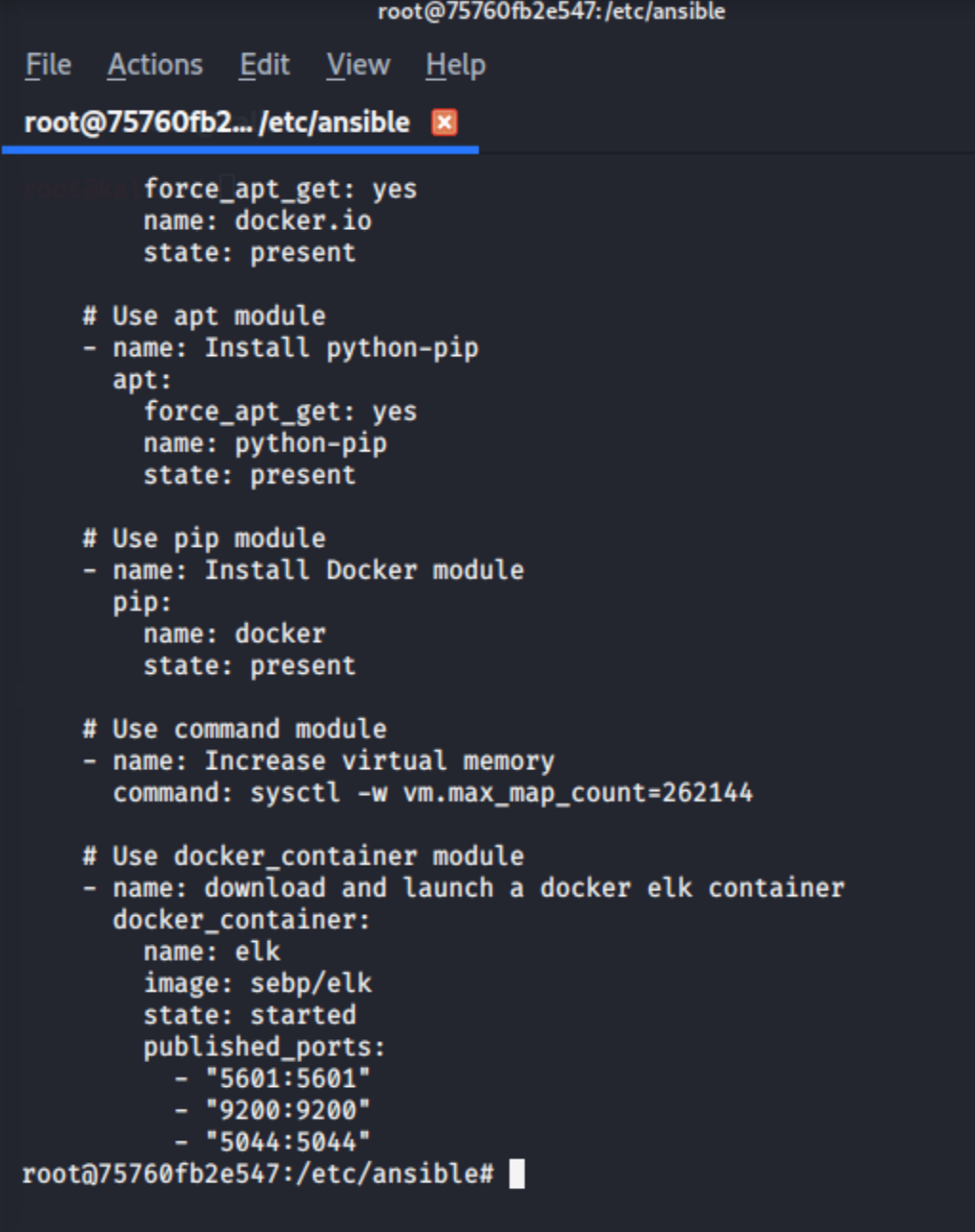
- Configures the container to start with the following port mappings:

- `5601:5601`

- `9200:9200`

- `5044:5044`

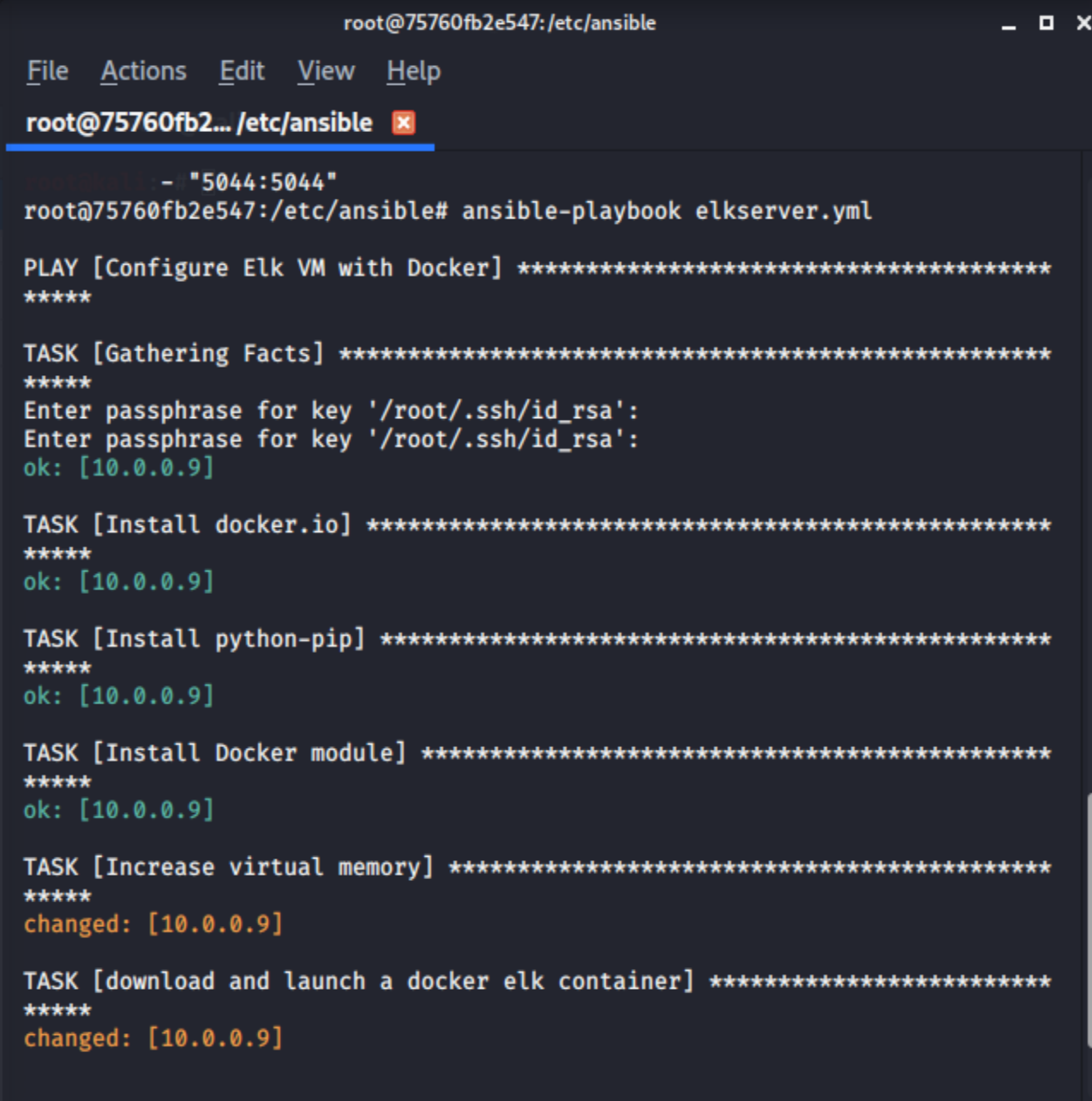
Done



\*\*Hint:\*\* Use the Ansible module `docker-container` along with published port mappings. [More info at nsible.com](https://docs.ansible.com/ansible/latest/modules/docker\_container\_module.html#examples).

- Starts the container.

Done



Refer to the documentation on Ansible's [docker\_container](https://docs.ansible.com/ansible/latest/modules/docker\_container\_module.html) and [docker\_image](https://docs.ansible.com/ansible/latest/modules/docker\_image\_module.html) modules for guidance.

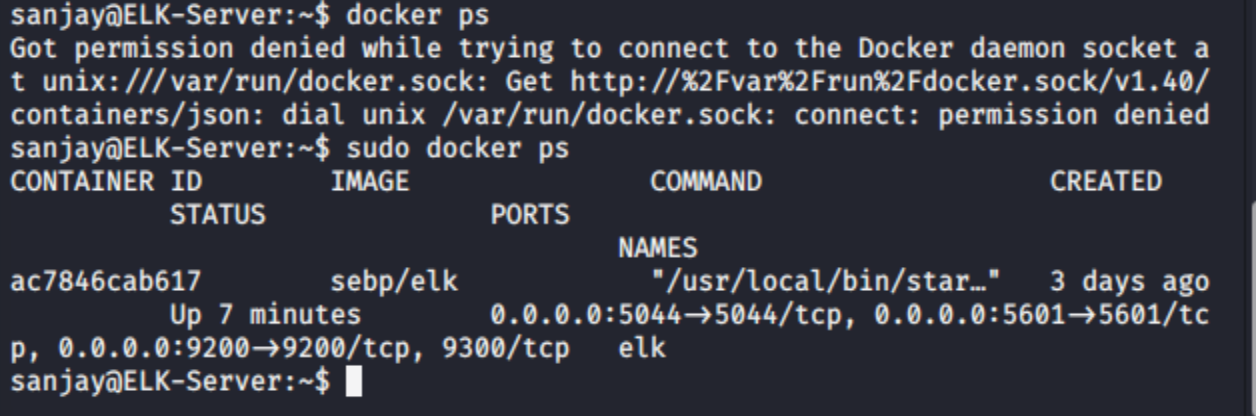
#### 3. Launching and Exposing the Container

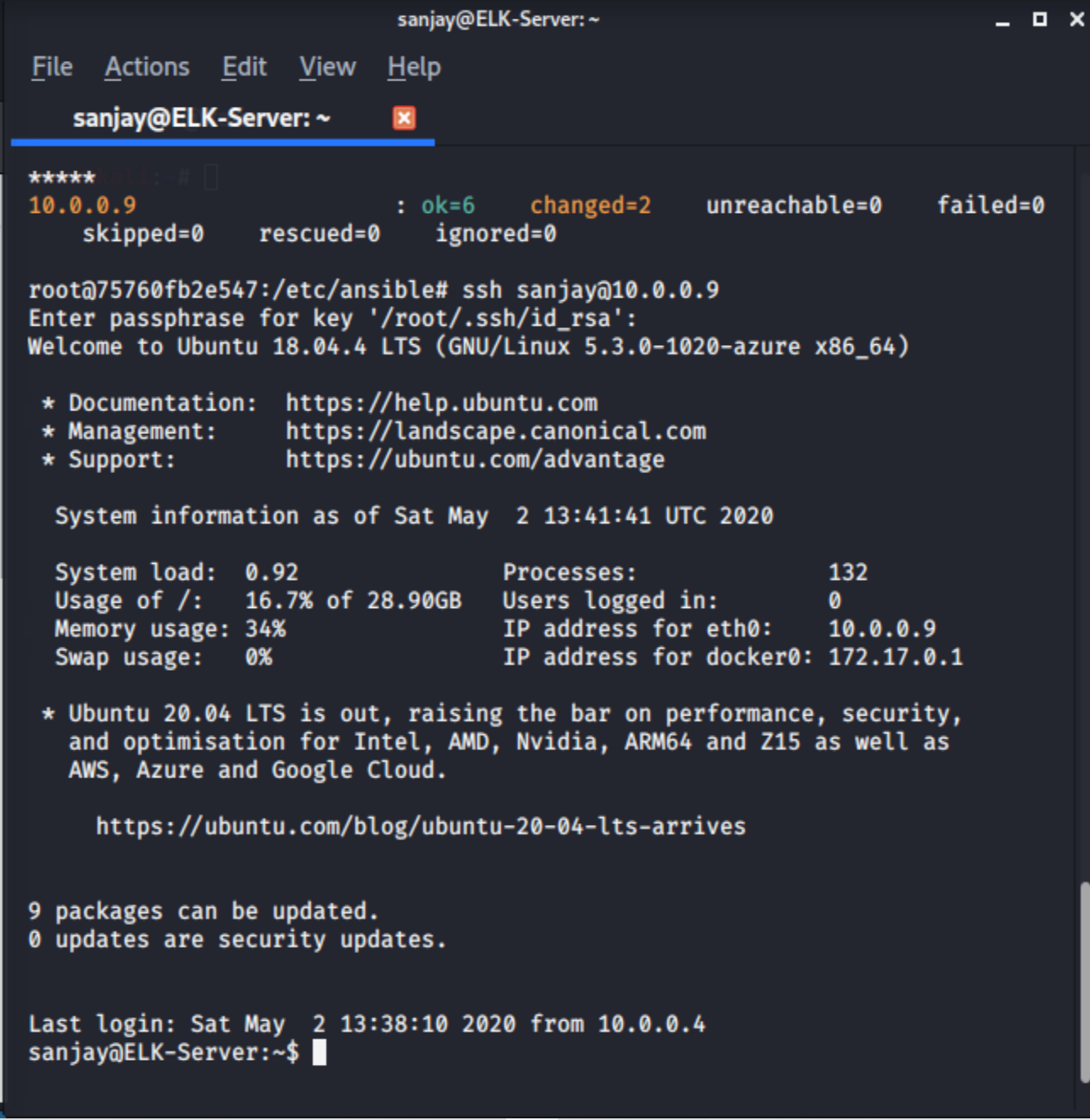
Check your playbook for typos and other errors, then run it.

After the playbook completes, you should still be in the Ansible container. From there, use the command line to SSH into the ELK server and ensure that the `sebp/elk` container is running by running: `docker ps`.

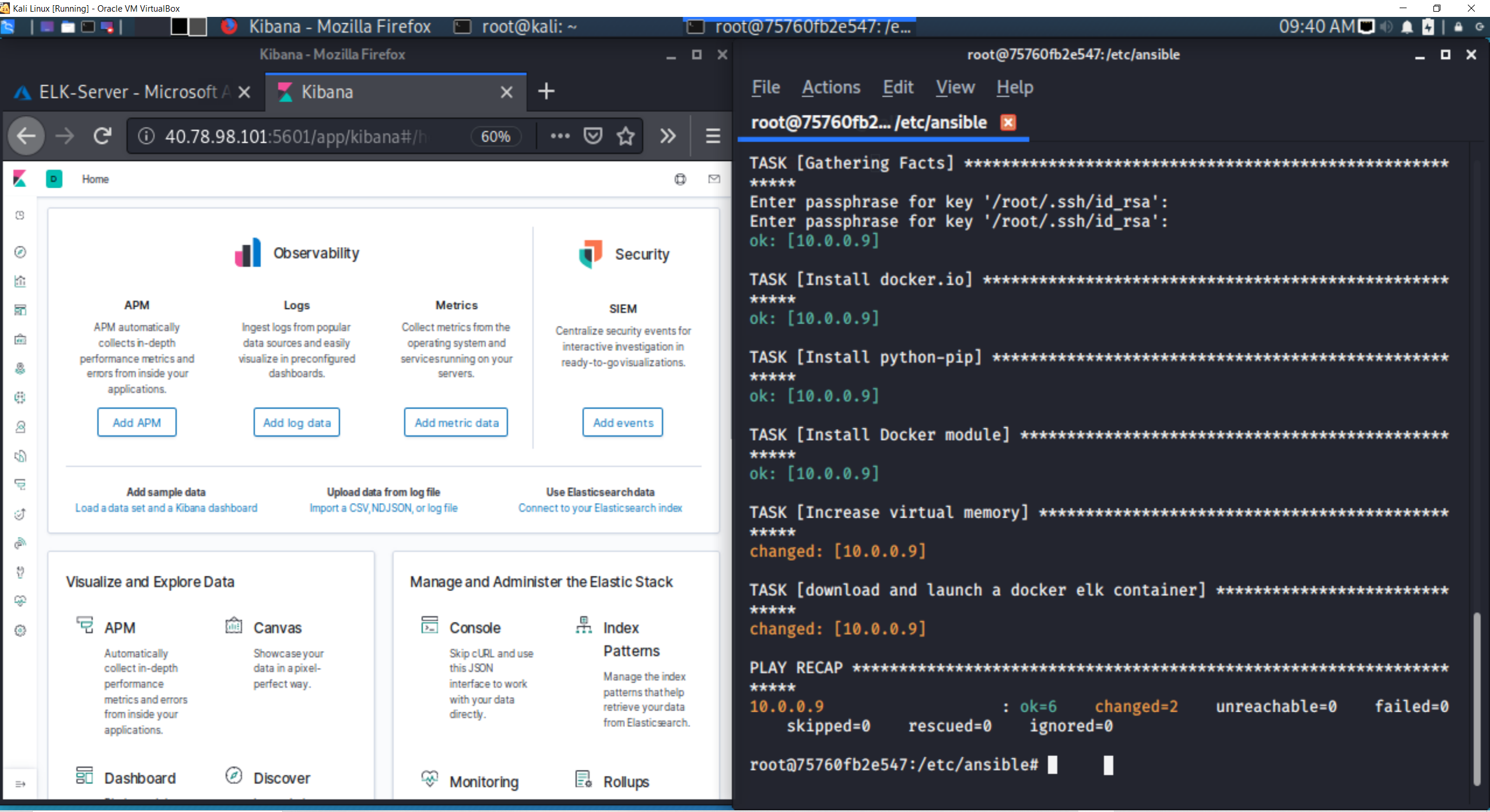
- You should see a single row whose second column is `sebp/elk`. Take a screenshot before proceeding to the next step.

Evidence of sebp/elk included





\*\*Note\*\*: If you're stuck, refer to the starter `elk-playbook.yaml` file provided: [Starter `elk-playbook` File](Activities/Stu\_Day\_One/Unsolved/Resources/install-elk.yml).



##### Troubleshooting

If your container fails to start, use the following links to troubleshoot it.

- The most common reason for the container failing to launch is a limit on map counts. Follow [these instructions from Elastic.co](https://www.elastic.co/guide/en/elasticsearch/reference/5.0/vm-max-map-count.html#vm-max-map-count) to verify you meet the requirement.

- Review the [official ELK stack documentation](https://elk-docker.readthedocs.io/#prerequisites).

- If the above steps don't help, you can try manually launching the container with this command:

- `sudo docker run -p 5601:5601 -p 9200:9200 -p 5044:5044 -it --name elk sebp/elk`

#### 4. Identity and Access Management

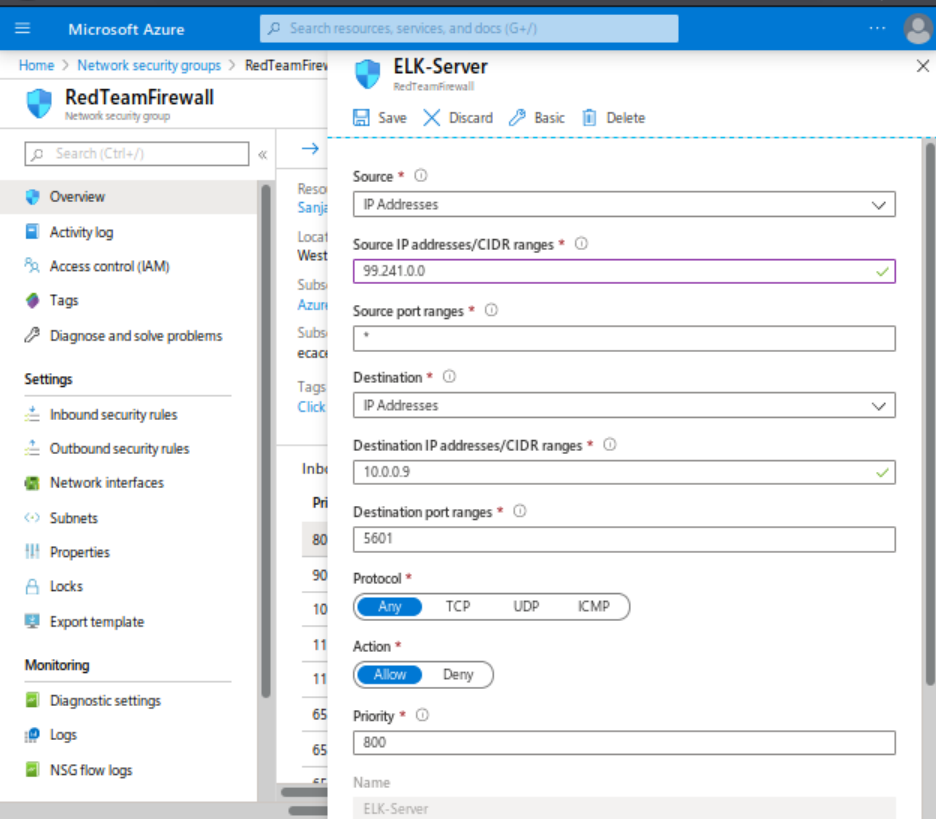
The final step is to restrict access to the ELK VM using Azure's network security groups (NSGs). You need to add your public IP address to a whitelist, just as you did when clearing access to your jump box.

- \*\*Note:\*\* Make sure you are on your Azure account for this step.

Recall that the ELK stack's web server runs on port `5601`. Open your virtual network's existing NSG and create an incoming rule for your security group that allows TCP traffic over port `5601` from your public IP address.

Incoming rule within security group for ELK-Server restricting to port 5601





- \*\*Note:\*\* If you finish this step in the classroom, you will need to repeat this step at home to connect to Kibana from your personal network.

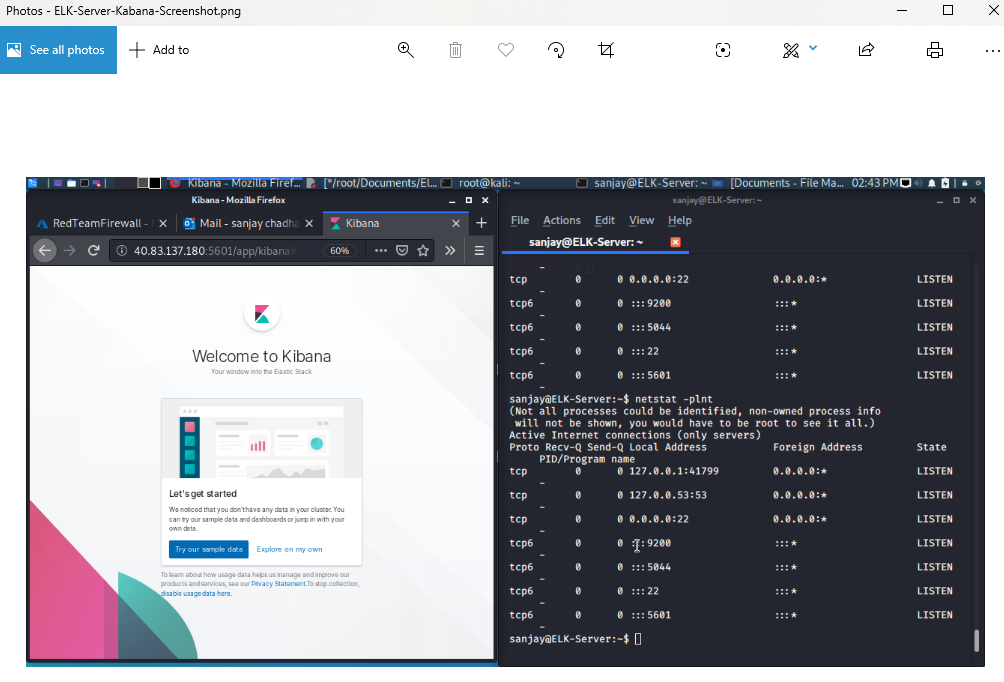
Verify that you can access your server by navigating to http://[your.VM.IP]:5601. Use the public IP address of your new VM.

- You should see this webpage:

![Kibana](Resources/Kibana\_Home.png)

If this is what you see, congratulations! Take a screenshot of your dashboard before proceeding.

Kibana Home page



---

### Day 1 Milestone

In today's class, you:

- Deployed a new VM on your virtual network.

- Created an Ansible play to install and configure an ELK instance.

- Restricted access to the new server.

Completing these steps required you to leverage your systems administration, virtualization, cloud, and automation skills. This is an impressive set of tools to have in your toolkit.

---

Â© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.