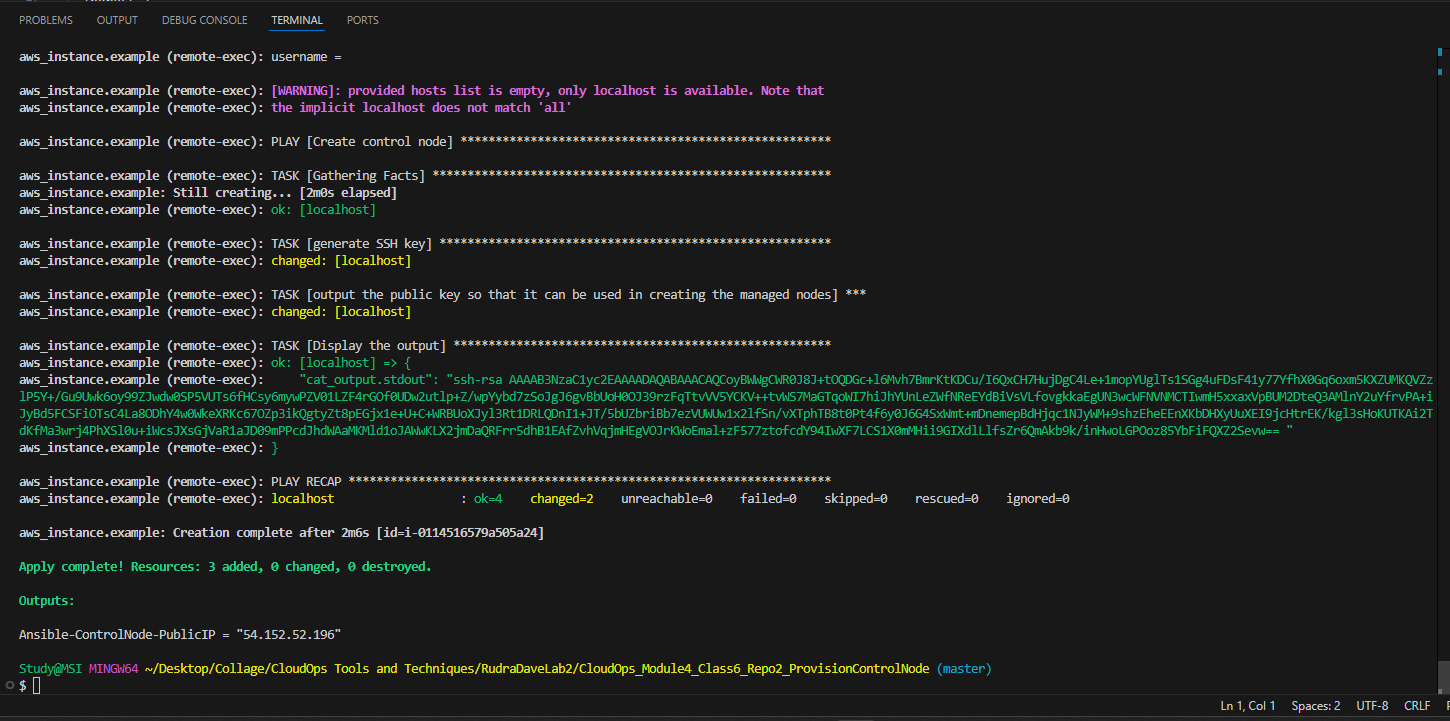
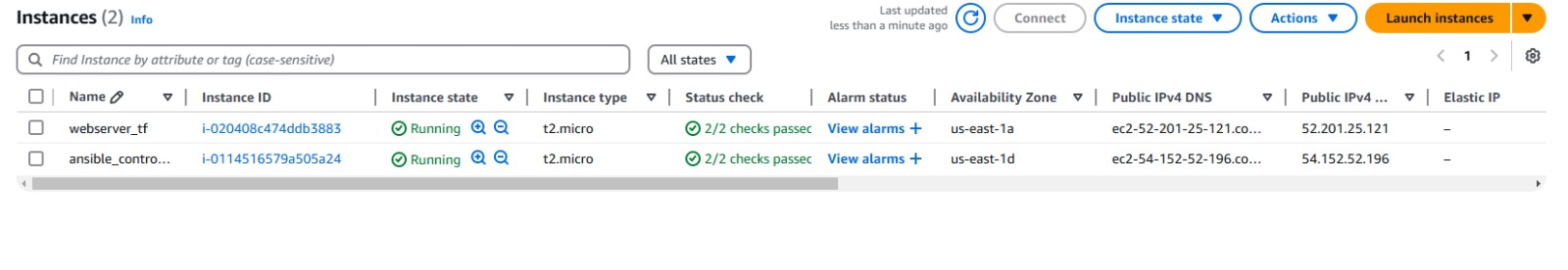
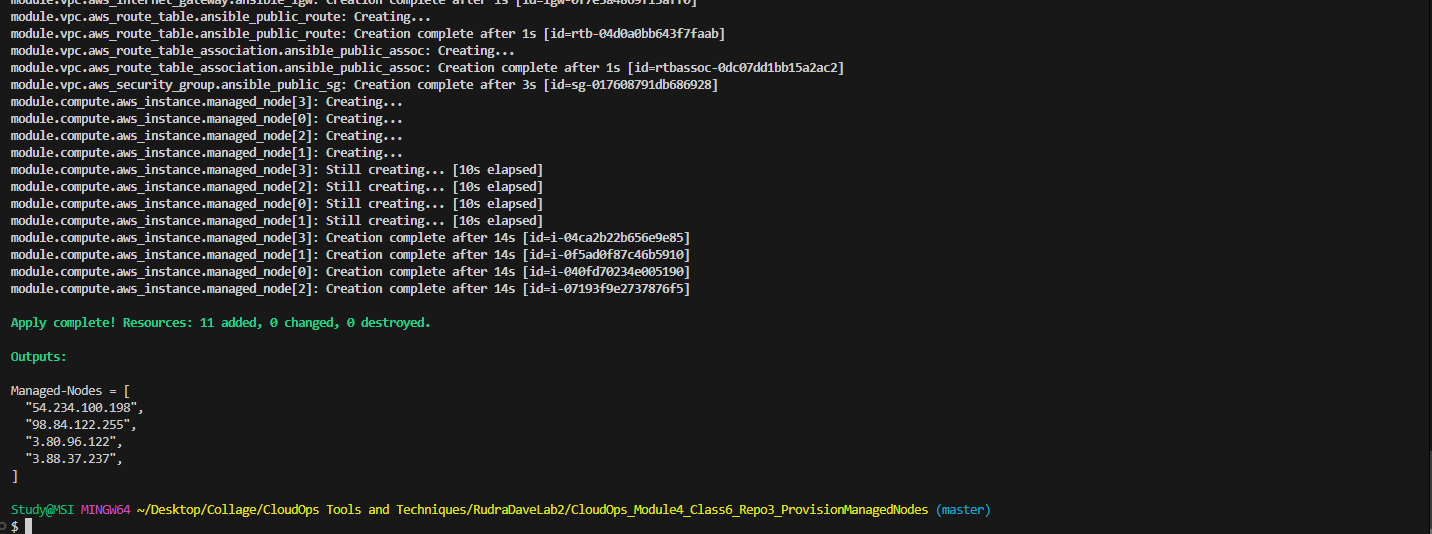
1. Using the Managing Multiple Servers using Ansible section from class 6 as your guide, you will create and manage infrastructure using Terraform and Ansible.
2. The following repos are used in this lab. Fork these into your own GitHub account
   * [Repo: ProvisionControlNode](https://github.com/YorkU-SCS-Cloud-Ops/CloudOps_Module4_Class6_Repo2_ProvisionControlNode)
   * [Repo: ProvisionManagedNodes](https://github.com/YorkU-SCS-Cloud-Ops/CloudOps_Module4_Class6_Repo3_ProvisionManagedNodes)
   * [Repo: ControlNodeFiles](https://github.com/YorkU-SCS-Cloud-Ops/CloudOps_Module4_Class6_Repo4_ControlNodeFiles)



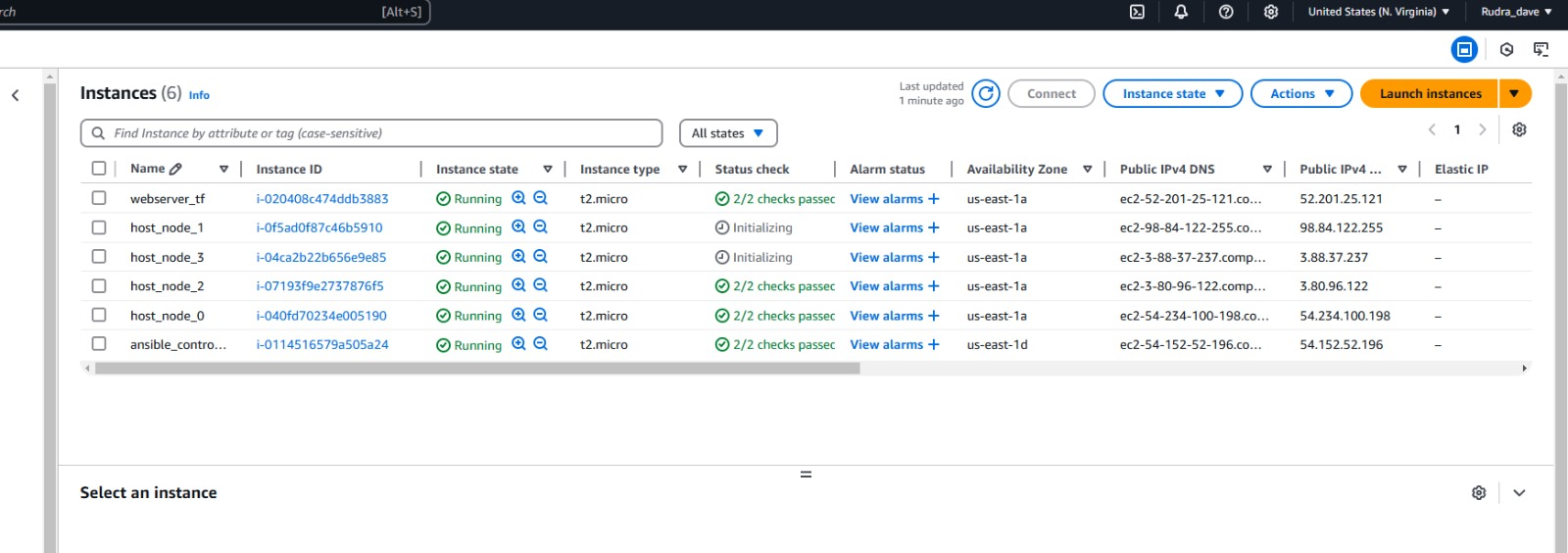


Follow the step-by-step instructions in the slides for Class 6 to provision the Control Node

Managed nodes [including IP addresses for four web servers]



* AWS EC2 list showing all created servers, including public IPs



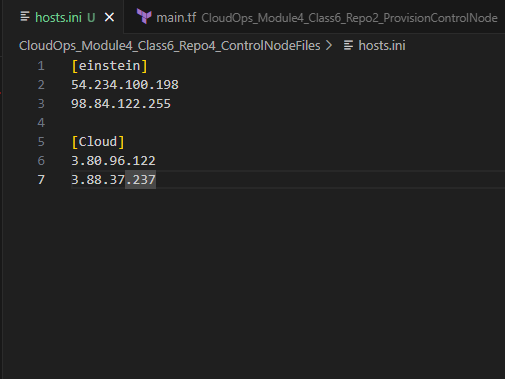
Copying the SSH key from the Control Node output, use it to create an id\_rsa.pub key in the ProvisionManagedNodes git repo. Modify your code to create 4 web servers. Create a hosts.ini file that has two groups, first two webservers as the [einstein] group, second batch of webservers as the [Cloud] group

 Rename install\_httpd.yaml as install\_Einstein.yaml, updating so that hosts is einstein. Execute this using the ansible-playbook command and screenshot the output. Capture screenshots of both web servers, as per slide 32 of Class 6 deck.

6. Create an install\_Cloud.yaml to follow the same pattern, but deploying the York U Cloud computing page, the link for which is <https://github.com/wessamabdelwahab/CSCC1030>.

7. Execute install\_Cloud.yaml using the ansible-playbook command and screenshot the output. Capture screenshots of both web servers, as per slide 32 of Class 6 deck.

Hosts.ini



Output from running ansible-playbook commands for both playbooks.

