



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

DevOps (CS457)

ASSIGNMENT 3 – Task 3

Add users to EC2 instances with SSH Access – Ansible

Submitted to:

Dr. Uma S

Submitted by:

Team 1

Sumith Sai Budde (18BCS101)

Syed Sufyan Ahmed (18BCS103)

Shaik Fharook (18BCS091)

Parvati Jayakumar (18BEC036)

P Chethan Krishna (18BEC040)

G Rithika (18BCS031)

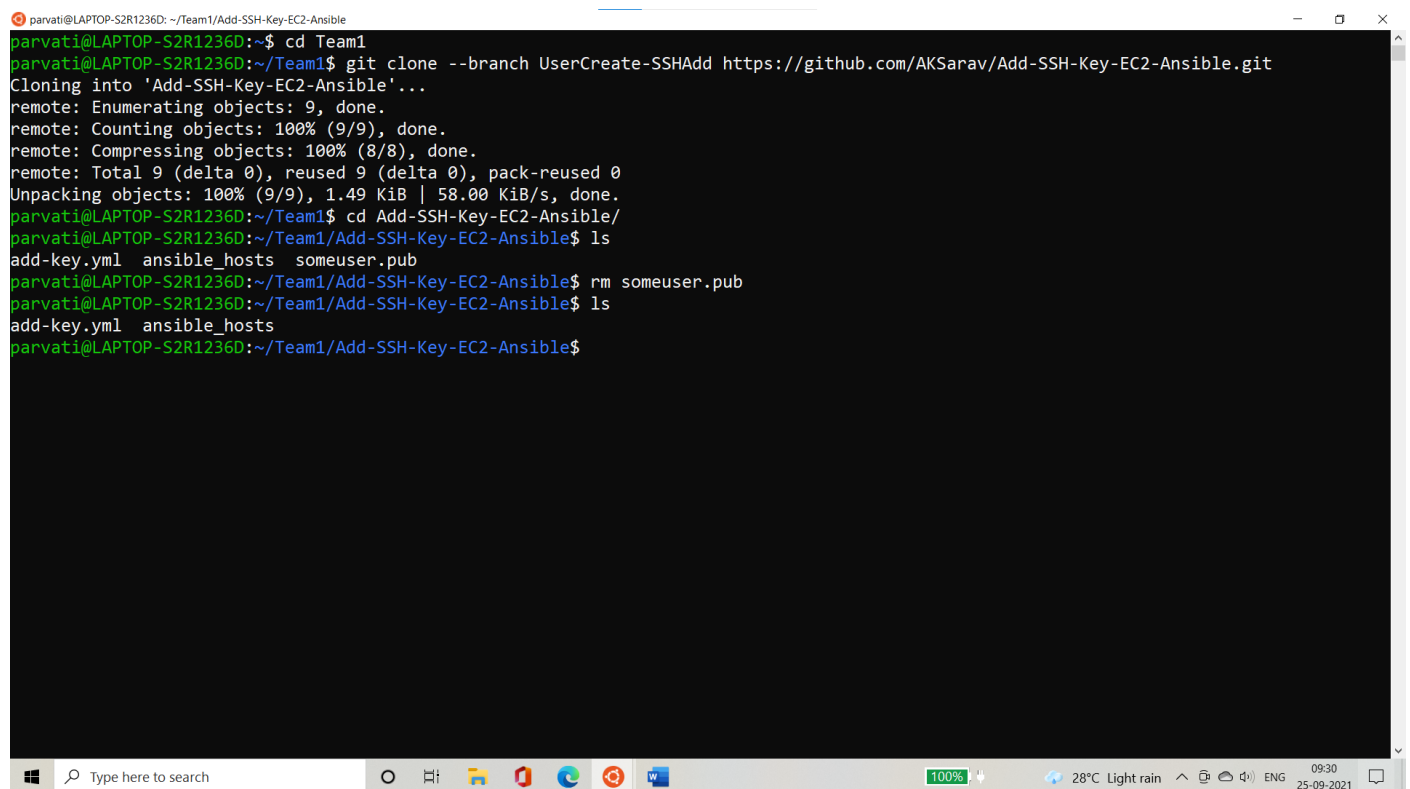
Pokala Dattatreya (18BCS067)

Rama Dundi Saketh (18BCS076)

STEP 1: Clone the GitHub repository and modify the files

Step 1.1: Clone the repository using the command:

```
git clone --branch UserCreate-SSHAdd https://github.com/AKSarav/Add-SSH-Key-EC2-Ansible.git
```



```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~$ cd Team1
parvati@LAPTOP-S2R1236D:~/Team1$ git clone --branch UserCreate-SSHAdd https://github.com/AKSarav/Add-SSH-Key-EC2-Ansible.git
Cloning into 'Add-SSH-Key-EC2-Ansible'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 1.49 KiB | 58.00 KiB/s, done.
parvati@LAPTOP-S2R1236D:~/Team1$ cd Add-SSH-Key-EC2-Ansible/
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ls
add-key.yml  ansible_hosts  someuser.pub
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ rm someuser.pub
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ls
add-key.yml  ansible_hosts
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$
```

The screenshot shows a terminal window with the following content: The user is in the directory ~/Team1/Add-SSH-Key-EC2-Ansible. They run 'cd Team1' and then 'git clone --branch UserCreate-SSHAdd https://github.com/AKSarav/Add-SSH-Key-EC2-Ansible.git'. The output shows the cloning process, including enumerating, counting, and compressing objects. After cloning, they run 'cd Add-SSH-Key-EC2-Ansible/' and 'ls', which shows the files 'add-key.yml', 'ansible_hosts', and 'someuser.pub'. Then they run 'rm someuser.pub' and 'ls' again, showing 'add-key.yml' and 'ansible_hosts'.

STEP 2: Set up the remote instance

Step 2.1: Launch a Red Hat Enterprise Linux served (Example: as server named 'Task_5').

Make sure that port -22 is open.

The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The main content area displays the 'Instances (1/4)' page. A table lists the instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP...
RHEL7	i-09c3d9444fe0d3d91	Stopped	t2.micro	-	No alarms	us-east-2a	-
Task_3	i-02f06b648650cf0ac	Stopped	t2.micro	-	No alarms	us-east-2b	-
ubuntu	i-0b7559358b73995c1	Stopped	t2.micro	-	No alarms	us-east-2b	-
Task_5	i-090afc835c065b78a	Running	t2.micro	Initializing	No alarms	us-east-2c	ec2-18-191-2...

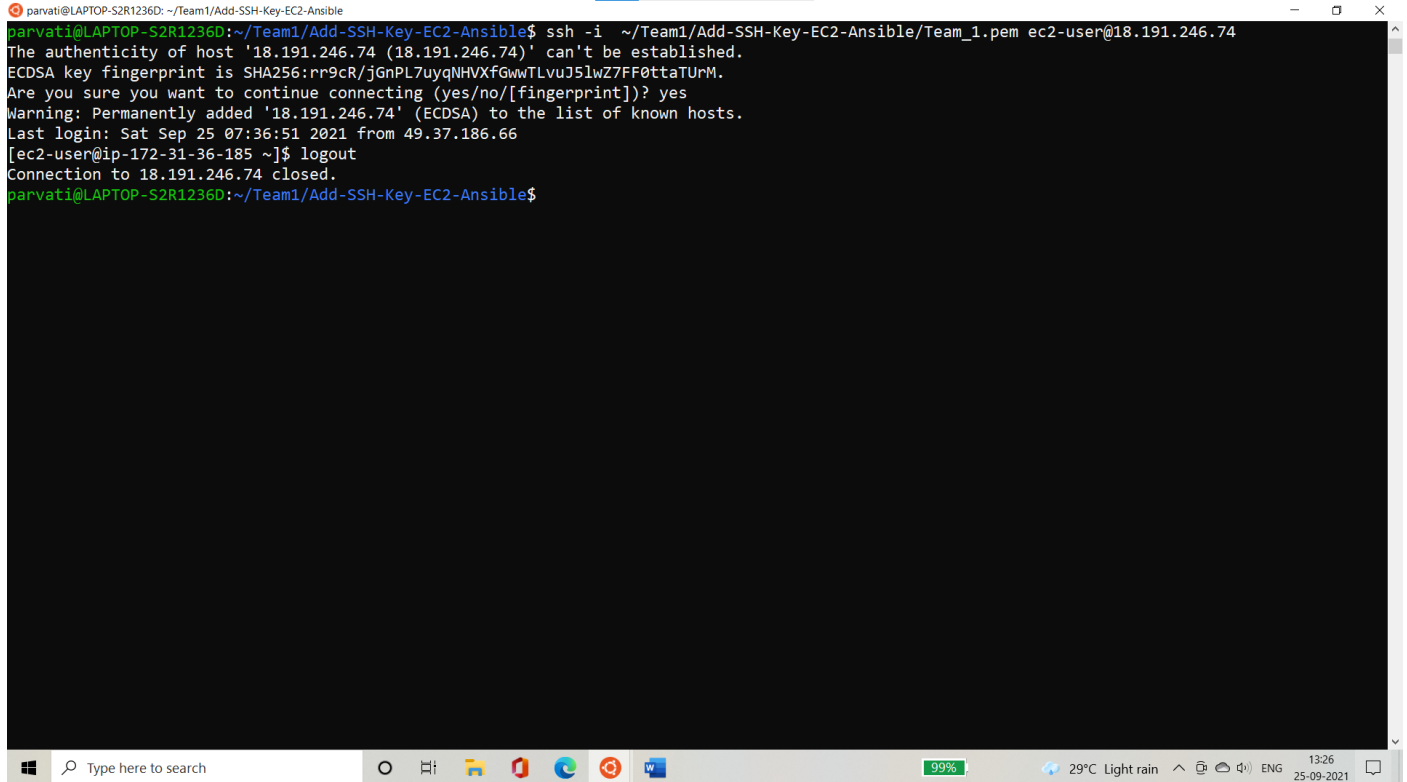
The instance 'Task_5' is selected, and the details panel at the bottom shows 'Instance: i-090afc835c065b78a (RHEL)'.

Step 2.2: Generate a SSH key.

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/parvati/.ssh/id_rsa):
Created directory '/home/parvati/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/parvati/.ssh/id_rsa
Your public key has been saved in /home/parvati/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qbX4Ijk0gAKdt/pNj9hUMNhgbsIj7+UOHFfnCvQUtU0 parvati@LAPTOP-S2R1236D
The key's randomart image is:
+---[RSA 3072]-----+
| . . . . E |
| . o = . . + |
| . + * o o o . |
| o + B * + |
| + + + S . |
| .. + = * o |
| o + . * o |
| o + O . + |
| . + O = . o |
+---[SHA256]-----+
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$
```

Step 2.3: Make sure that the remote server 'Task_5' can be accessed through our local machine.

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ssh -i ~/Team1/Add-SSH-Key-EC2-Ansible/Team_1.pem ec2-user@18.191.246.74
The authenticity of host '18.191.246.74 (18.191.246.74)' can't be established.
ECDSA key fingerprint is SHA256:rr9cR/jGnPL7uyqNHVXfGwwTLvuJ5lwZ7FF0ttaTurM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.191.246.74' (ECDSA) to the list of known hosts.
Last login: Sat Sep 25 07:36:51 2021 from 49.37.186.66
[ec2-user@ip-172-31-36-185 ~]$ logout
Connection to 18.191.246.74 closed.
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$
```



STEP 3: Modify the files required for executing the playbook

Step 3.1: Configure the files: ansible_hosts, add-key.yml.

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ vi ansible_hosts
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ cat ansible_hosts
[hosts_to_add_key]
18.191.246.74 ansible_user=ec2-user ansible_port=22

[hosts_to_add_key:vars]
ansible_ssh_common_args="-o StrictHostKeyChecking=no"
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ vi add-key.yml
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ cat add-key.yml
---
- name: "Playbook to Create User and Add Key to EC2 Instance"
  hosts: hosts_to_add_key
  become: true
  tasks:

  - name : "Create Groups"
    group:
      name: "{{item}}"
      state: "present"
    with_items:
      - TeamA
      - TeamB

  - name : "Create a user"
    user:
      name: "{{item.name}}"
      create_home: yes
      group: "{{item.group}}"
      state: present
      ssh_key_file: .ssh/id_rsa
      ssh_key_type: rsa
    with_items:
      - { name: 'user1', group: 'TeamA'}
      - { name: 'user2', group: 'TeamB'}

  - name: "Copy the authorized key file from"
```

```
      - TeamB

  - name : "Create a user"
    user:
      name: "{{item.name}}"
      create_home: yes
      group: "{{item.group}}"
      state: present
      ssh_key_file: .ssh/id_rsa
      ssh_key_type: rsa
    with_items:
      - { name: 'user1', group: 'TeamA'}
      - { name: 'user2', group: 'TeamB'}

  - name: "Copy the authorized key file from"
    authorized_key:
      user: "{{item.name}}"
      state: "{{item.userstate}}"
      key: "{{ lookup('file', '{{ item.key }}') }}"
    with_items:
      - { name: 'user1', key: 'user1.pub', userstate: 'present'}
      - { name: 'user2', key: 'user2.pub', userstate: 'present'}

parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ls
Team_1.pem add-key.yml ansible-hosts ansible_hosts user1 user1.pub user2 user2.pub
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$

  - name: "Copy the authorized key file from"
```

STEP 4: Execute the playbook

Step 4.1: Ping to check if the connection is successful

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ansible hosts_to_add_key -i ansible_hosts -m ping --key-file ~/Team1/Add-SSH-Key-EC2-Ansible/Team_1.pem --user ec2-user
18.191.246.74 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ echo "We were successfully able to ping the server."|cowsay

  __  __
 /  __  \
(  ____/
 \_____)
  ||----w |
  ||     ||

parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$
```

Step 4.2: Use the command ansible-playbook to execute add-key.yml

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ansible-playbook add-key.yml -i ansible_hosts --user parvati --key-file ~/Team1/Add-SSH-Key-EC2-Ansible/Team_1.pem -e "key=~/.ssh/id_rsa.pub"

< PLAY [Playbook to Create User and Add Key to EC2 Instance] >
-----
  __  __
 /  __  \
(  ____/
 \_____)
  ||----w |
  ||     ||

< TASK [Gathering Facts] >
-----
  __  __
 /  __  \
(  ____/
 \_____)
  ||----w |
  ||     ||

ok: [18.191.246.74]

< TASK [Create Groups] >
-----
  __  __
 /  __  \
(  ____/
 \_____)
  ||----w |
  ||     ||

ok: [18.191.246.74] => (item=TeamA)
ok: [18.191.246.74] => (item=TeamB)

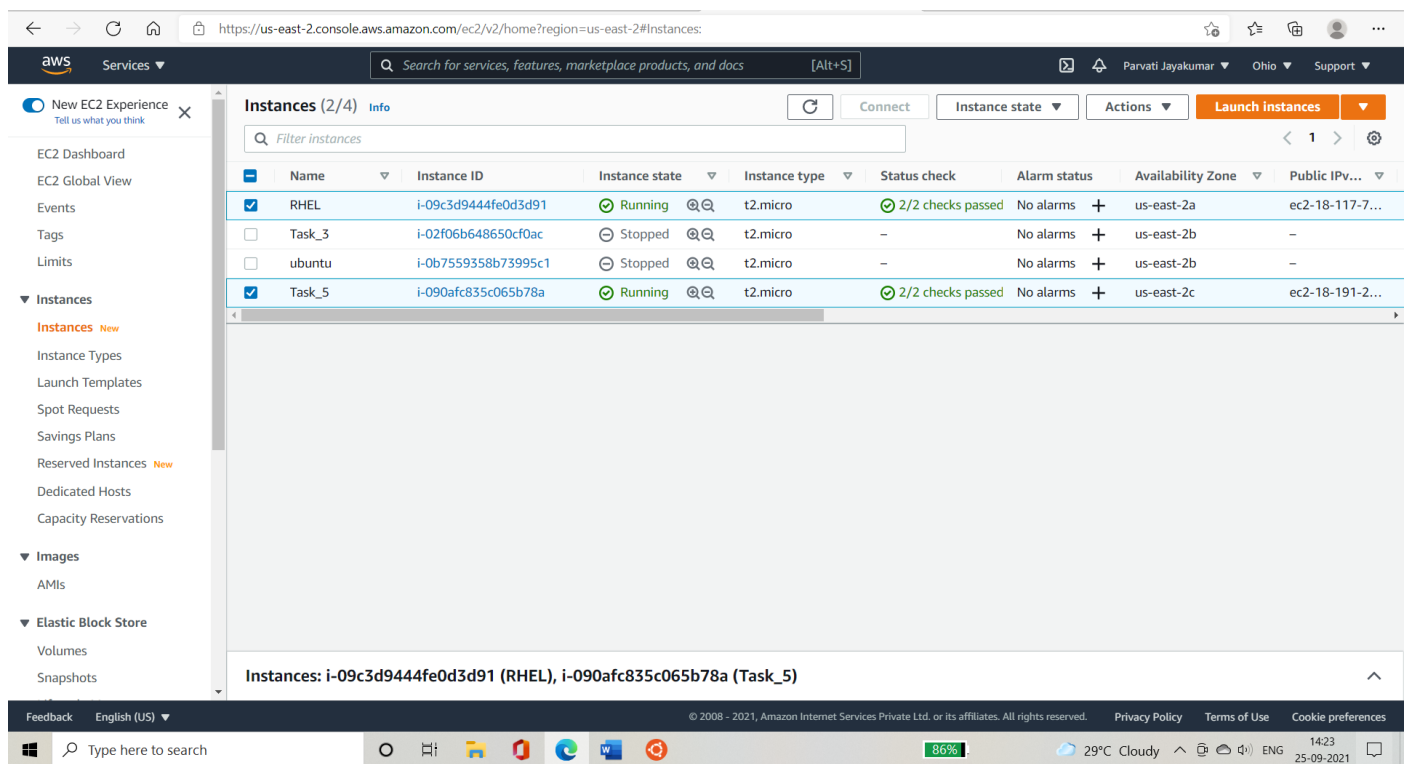
< TASK [Create a user] >
-----
  __  __
 /  __  \
(  ____/
 \_____)
  ||----w |
  ||     ||

ok: [18.191.246.74] => (item={'name': 'user1', 'group': 'TeamA'})
ok: [18.191.246.74] => (item={'name': 'user2', 'group': 'TeamB'})

< TASK [Copy the authorized key file from] >
-----
  __  __
 /  __  \
(  ____/
 \_____)
  ||----w |
  ||     ||
```


STEP 5: You can also try to add users at once by giving multiple hosts.

Step 5.1: Launch a new instance (Example: RHEL)

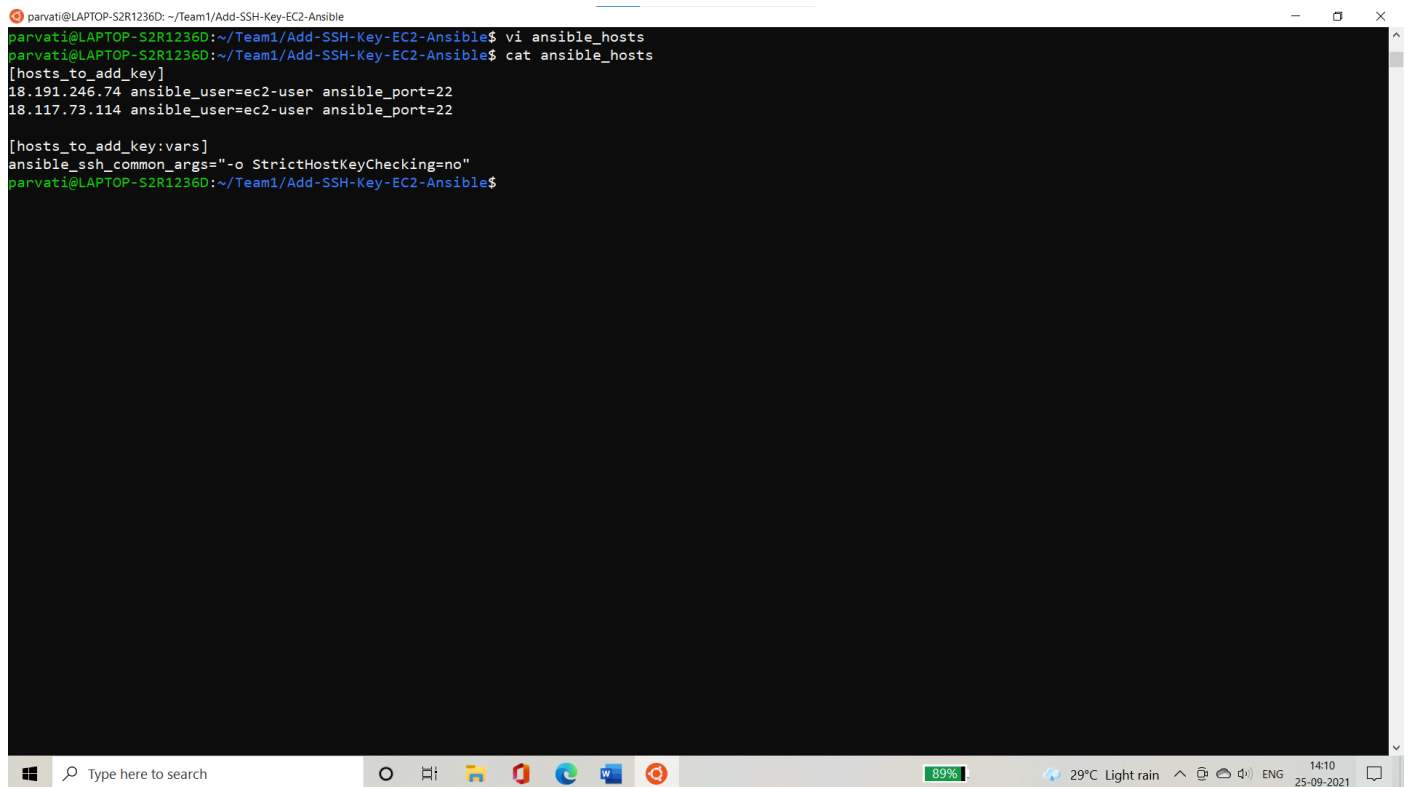


The screenshot shows the AWS Management Console for the 'us-east-2' region. The 'Instances' page is active, displaying a list of four EC2 instances. The instances are:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP...
RHEL	i-09c3d9444fe0d3d91	Running	t2.micro	2/2 checks passed	No alarms	us-east-2a	ec2-18-117-7...
Task_3	i-02f06b648650cf0ac	Stopped	t2.micro	-	No alarms	us-east-2b	-
ubuntu	i-0b7559358b73995c1	Stopped	t2.micro	-	No alarms	us-east-2b	-
Task_5	i-090afc835c065b78a	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c	ec2-18-191-2...

The left sidebar shows the navigation menu with 'Instances' selected. The top bar includes the AWS logo, search bar, and user profile. The bottom status bar shows the system clock and weather.

Step 5.2: Configure ansible_hosts file



The screenshot shows a terminal window with the following commands and output:

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ vi ansible_hosts
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ cat ansible_hosts
[hosts_to_add_key]
18.191.246.74 ansible_user=ec2-user ansible_port=22
18.117.73.114 ansible_user=ec2-user ansible_port=22

[hosts_to_add_key:vars]
ansible_ssh_common_args="-o StrictHostKeyChecking=no"
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$
```

The terminal window shows the user 'parvati' at the prompt. The commands executed are 'vi ansible_hosts' to edit the file and 'cat ansible_hosts' to display its contents. The output shows the configuration for two hosts in the 'hosts_to_add_key' group, with their IP addresses, usernames, and ports. A variable 'ansible_ssh_common_args' is also defined with the value '-o StrictHostKeyChecking=no'.

Step 5.3: Execute the YAML file

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ansible hosts_to_add_key -i ansible_hosts -m ping --key-file ~/Team1/Add-SSH-Key-EC2-Ansible/Team_1.pem --user ec2-user
18.191.246.74 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
18.117.73.114 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ansible-playbook add-key.yml -i ansible_hosts --user parvati --key-file ~/Team1/Add-SSH-Key-EC2-Ansible/Team_1.pem -e "key~/ssh/id_rsa.pub"

< PLAY [Playbook to Create User and Add Key to EC2 Instance] >

  \   ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||

< TASK [Gathering Facts] >

  \   ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||

ok: [18.117.73.114]
ok: [18.191.246.74]

< TASK [Create Groups] >

  \   ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||

changed: [18.117.73.114] => (item=TeamA)
ok: [18.191.246.74] => (item=TeamA)
changed: [18.117.73.114] => (item=TeamB)
ok: [18.191.246.74] => (item=TeamB)
```

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible

  \   ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||

changed: [18.117.73.114] => (item=TeamA)
ok: [18.191.246.74] => (item=TeamA)
changed: [18.117.73.114] => (item=TeamB)
ok: [18.191.246.74] => (item=TeamB)

< TASK [Create a user] >

  \   ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||

ok: [18.191.246.74] => (item={'name': 'user1', 'group': 'TeamA'})
changed: [18.117.73.114] => (item={'name': 'user1', 'group': 'TeamA'})
changed: [18.117.73.114] => (item={'name': 'user2', 'group': 'TeamB'})
ok: [18.191.246.74] => (item={'name': 'user2', 'group': 'TeamB'})

< TASK [Copy the authorized key file from] >

  \   ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||

ok: [18.191.246.74] => (item={'name': 'user1', 'key': 'user1.pub', 'userstate': 'present'})
changed: [18.117.73.114] => (item={'name': 'user1', 'key': 'user1.pub', 'userstate': 'present'})
ok: [18.191.246.74] => (item={'name': 'user2', 'key': 'user2.pub', 'userstate': 'present'})
changed: [18.117.73.114] => (item={'name': 'user2', 'key': 'user2.pub', 'userstate': 'present'})

< PLAY RECAP >

  \   ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||

18.117.73.114      : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
18.191.246.74     : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Step 5.4: Validate the results by checking users in both the instances

```
parvati@LAPTOP-S2R1236D: ~/Team1/Add-SSH-Key-EC2-Ansible
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ssh -p 22 -i ~/Team1/Add-SSH-Key-EC2-Ansible/Team_1.pem ec2-user@18.117.73.114 "id user1 && id user2"
uid=1003(user1) gid=1003(TeamA) groups=1003(TeamA)
uid=1004(user2) gid=1004(TeamB) groups=1004(TeamB)
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ^C
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$ ssh -p 22 -i ~/Team1/Add-SSH-Key-EC2-Ansible/Team_1.pem ec2-user@18.191.246.74 "id user1 && id user2"
uid=1001(user1) gid=1003(TeamA) groups=1003(TeamA)
uid=1002(user2) gid=1004(TeamB) groups=1004(TeamB)
parvati@LAPTOP-S2R1236D:~/Team1/Add-SSH-Key-EC2-Ansible$
```

