

# **Red Hat Ansible Automation Platform on IBM Power Systems**

Hands-On Lab

**Christopher Bienko**

Worldwide Technology Sales  
IBM IT Infrastructure

---

**cdbienko@us.ibm.com**



# Lab Objectives

This hands-on lab will familiarize you with **Red Hat Ansible** on IBM AIX / PowerVM by putting you in the shoes of an administrator tasked with modernizing an IBM AIX environment using Ansible automation.

Note that Red Hat Ansible can also be configured on **IBM i** as well. More information is available on [\*\*Seismic\*\*](#).

Throughout the course of this lab, you will:

- Provision IBM AIX from the IBM Systems Cloud for Enablement and Co-Creation (CECC)
- Deploy and configure Red Hat Ansible clients within your PowerVM / AIX environment
- Execute a variety of Ansible Modules for running OS commands, test scripts, and manage users

You will need an estimated **60 to 120 minutes** to complete this lab.

Provision an IBM AIX environment from the [\*\*IBM Systems Cloud for Enablement and Co-Creation\*\*](#) (CECC).

**Reserve only for the time you need.** This is a finite resource and limiting your reservation time ensures more IBMers can leverage the resources for their client engagements. Thank you!

# Certification



**IMPORTANT:** For **IBMer**s and **Business Partners** to receive credit for completing this lab, you must pass a **certification quiz** after completing the hands-on portion of the course. The quiz consists of 5 multiple choice questions, with four possible responses (and only one correct answer) for each question.

The quiz questions will ask you about on-screen text or descriptions that will come up as you work through the lab guide. **You will be prompted in the lab guide instructions to RECORD this information** (or screenshot – your preference), so that you can easily breeze through the certification quiz questions at the end.

```
cecuser@p663-pvm1:/home/cecuser $ sudo vi /opt/freeware/etc/ansible.cfg
cecuser@p663-pvm1:/home/cecuser $ sudo mkdir $HOME/ansible-scripts
cecuser@p663-pvm1:/home/cecuser $ sudo vi /opt/freeware/etc/ansible/hosts
cecuser@p663-pvm1:/home/cecuser $ ansible sixclients -m script -a 'testscript.sh awesome'
[WARNING]: Platform aix on host 129.48.126.224 is using the discovered Python
interpreter at /usr/bin/python, but future installations of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices_interpreter_discovery.html for more information.
129.48.126.224 | SUCCESS =>
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform aix on host 129.48.126.225 is using the discovered Python
interpreter at /usr/bin/python, but future installations of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices_interpreter_discovery.html for more information.
129.48.126.225 | SUCCESS =>
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

SAMPLE  
OUTPUT

```
cecuser@p663-pvm1:/home/cecuser $ sudo vi testscript.sh
cecuser@p663-pvm1:/home/cecuser $ sudo chmod +x testscript.sh
cecuser@p663-pvm1:/home/cecuser $ ansible sixclients -m script -a 'testscript.sh awesome'
129.48.126.226 | CHANGED => {
  "changed": true,
  "rc": 0,
  "stdout": "shared connection to 129.48.126.226 closed.\n",
  "stderr": "shared connection to 129.48.126.226 closed.\n"
}, {
  "changed": false,
  "rc": 0,
  "stdout": "Executing my script with parameter awesome\nFri Sep 11 13:34:32 EDT 2020\nok: [129.48.126.226]\n[WARNING]: Platform aix on host 129.48.126.226 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices_interpreter_discovery.html for more information.
129.48.126.227 | CHANGED => {
  "changed": true,
  "rc": 0,
  "stdout": "shared connection to 129.48.126.227 closed.\n",
  "stderr": "shared connection to 129.48.126.227 closed.\n"
}, {
  "changed": false,
  "rc": 0,
  "stdout": "Executing my script with parameter awesome\nFri Sep 11 13:34:33 EDT 2020\nok: [129.48.126.227]\n[WARNING]: Platform aix on host 129.48.126.227 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices_interpreter_discovery.html for more information.
129.48.126.226 | ok=2 changed=1 unreachable=0 failed=0 skipped=0
129.48.126.227 | ok=2 changed=1 unreachable=0 failed=0 skipped=0
cecuser@p663-pvm1:/home/cecuser $
```

SAMPLE  
OUTPUT

```
cecuser@p663-pvm1:/home/cecuser/ansible-scripts $ sudo vim copy_testfile.playbook.yaml
cecuser@p663-pvm1:/home/cecuser/ansible-scripts $ ansible-playbook copy_testfile.playbook.yaml
PLAY [sixclients] *****
TASK [gather_facts] *****
[WARNING]: Platform aix on host 129.48.126.226 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices_interpreter_discovery.html for more information.
ok: [129.48.126.226]
[WARNING]: Platform aix on host 129.48.126.227 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices_interpreter_discovery.html for more information.
ok: [129.48.126.227]
TASK [copy] *****
changed: [129.48.126.226]
changed: [129.48.126.227]
PLAY RECAP *****
129.48.126.226 : ok=2 changed=1 unreachable=0 failed=0 skipped=0
129.48.126.227 : ok=2 changed=1 unreachable=0 failed=0 skipped=0
cecuser@p663-pvm1:/home/cecuser/ansible-scripts $
```

SAMPLE  
OUTPUT

# Prerequisites

Please ensure you've prepared and reviewed the following before embarking on this lab:

- Request access to the [IBM Systems Cloud for Enablement and Co-Creation](#) (CECC)  
**Note:** You must agree to the terms and conditions of the environment to use these resources.
- Terminal, PuTTY, or an equivalent SSH console (to remotely connect to the AIX cluster).
- VPN or access to the IBM intranet.
- A web browser.

# Getting Started Reserving a Cluster

From a web browser, log in to the CECC at:

<https://www.ibm.com/it-infrastructure/services/cecc-portal/web/Catalog>

If this is your first visit to the CECC, you will first need to agree to the terms and conditions before accessing the environment.

- 1: From the Catalog tab, search for **IBM AIX**.
- 2: Click the **blue arrow** under the **POWER8 LPAR** tab (do not choose POWER9) to navigate to the configuration page.

The screenshot shows the CECC Portal interface. At the top, there is a navigation bar with links: CECC Portal, Catalog, Checkout, Projects, My Profile, and Support. Below the navigation bar, the title "CECC Portal" is displayed, followed by the subtitle "IBM Solutions Cloud for Enablement and Co-Creation". On the left side, there is a sidebar with a "Filter by" section containing a search bar with the text "IBM AIX" and a "Solution" section with a list of checkboxes for various IBM products. On the right side, there is a main content area titled "IaaS" with a list of items. The first item is "IBM AIX" with a thumbnail icon, followed by "IBM POWER8 LPAR" with a blue arrow icon. To the right of these items, there is another "IBM AIX" entry with a thumbnail icon and "IBM POWER9 LPAR" with a blue arrow icon. A yellow circle with the number "1" is placed over the search bar, and a yellow circle with the number "2" is placed over the blue arrow icon next to "IBM POWER8 LPAR".

# Getting Started Reserving a Cluster

The Checkout window will ask you to select the Image, Flavor, and Number of Instances for your cluster reservation.

3: From **Image**, select [IBM AIX 7.2 TL4](#).

4: From **Size**, select [IBM POWER8 LPAR](#) with the option for [2 vCPU + 4GB RAM + 50GB Disk](#).

5: From **Quantity**, select [3](#).

When satisfied, click **Add To Cart**.

IBM Solutions Cloud for  
Enablement and Co-Creation

## System details

### IBM POWER8 LPAR

This offering will provide you with IaaS access to IBM AIX POWER8 PowerVM LPARs including AIX 7.1 and adaptable to meet changing business demands. AIX on Power Systems delivers these capabilities and data requires. IBM PowerVM delivers industrial strength virtualization for AIX and Linux environments on through workload consolidation and capacity planning capabilities.

### IBM POWER8 LPAR

- IBM AIX 7.2 TL5
- IBM AIX 7.2 TL4
- IBM AIX 7.1 TL5

### Size

4

- IBM POWER8 LPAR: 2 vCPU, 4GB RAM, 50GB Disk
- IBM POWER8 LPAR: 4 vCPU, 8GB RAM, 50GB Disk
- IBM POWER8 LPAR: 8 vCPU, 16GB RAM, 50GB Disk
- IBM POWER8 LPAR: 16 vCPU, 32GB RAM, 50GB Disk

Quantity

3 ▾

5

[Cancel](#)

[Add to cart](#) 

# Getting Started Reserving a Cluster

Next you will configure the AIX cluster size and reservation period.

**6a:** Click the checkbox for your AIX image request.

**6b: Project Name** is set to your own preference.

**6c: Used for** is set to [Education / Training](#).



**Note:** Reserve only as much time as you need to complete this lab. You will need an estimate 60 to 90 minutes from start to finish. However, you can safely reserve several days of time if you feel you might benefit from it.

**6d:** When satisfied, click **Reserve Project**.

Your Cart

1 item selected

<input checked="" type="checkbox"/> Quantity	Image	Size
<input checked="" type="checkbox"/> 3	IBM AIX 7.2 TL4	IBM POWER8 LPAR: 2 vCPU, 4GB RAM, 50GB D

Project name **6a**: Bienko Ansible Cloud Experts 3c

Opportunity number (optional)

Number of VPNs: 0

Used for:  Education / Training

Availability check window:  30 Days  60 Days

Current date/time : Friday, June 18th 2021, 14:45:09

Start date: 06/18/2021  Start time: 15:14 **6b**

End date: 06/25/2021  End time: 15:14

**6c**

**6d**

[Continue shopping](#) [Reserve project](#)

# Getting Started Reserving a Cluster

After configuring the environment (the other variables not mentioned in the previous steps can be left as their default variables), you will be prompted with the following.

Click **OK** to dismiss.

Pay special attention to the **Date** and **Time** your IBM AIX environment is scheduled for. The CECC Dashboard may be brought online before your AIX cluster is ready. If you log into the CECC website too early, you'll only find incomplete URLs and credentials as the cluster is still provisioning.

Wait until an email arrives confirming that your environment is ready (which will be sent shortly after your scheduled time slot) before trying to connect to the IBM AIX cluster.

## Support

lement and Co-Creation (CECC) is an IBM Systems offering and is NOT affiliated with nor represents IBM Cloud offerings in any way. Please review our statement in the User Guide [here](#).

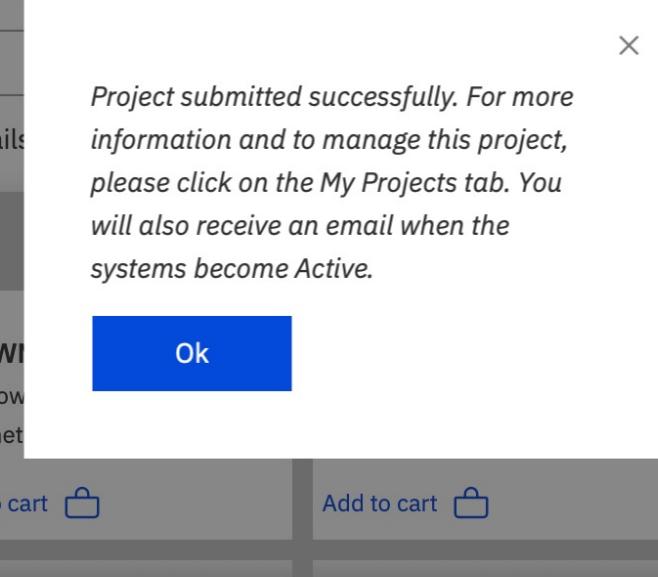
s must have their own VPN id. Refer to wiki or CECC [User Guide](#).

4.3.

ort. [Play video](#)

1 Systems Lab Services.

ONS within the same PROJECT will place the reserved systems in the same network. Each system would be entitled.



[Show More](#)

# Getting Started Accessing CECC Hardware

You can check on the status of your cluster reservation using the **My Projects** tab on [CECC](#).

If your cluster reservation looks like the following (with Project Credentials, Private Key, and URL fully populated) and you have received an email confirming the cluster is ready, then you are good to go!

**7:** Click the [key](#) icon next to **Project Credentials** and store this information for later.

**8:** View and save the **Private Key** for later.

**9:** Click the **Project Kit URL** to launch the CECC Dashboard, which contains information on your IBM AIX cluster.

The screenshot shows the 'My Projects' tab of the CECC interface. The page title is 'IBM Systems Cloud for Enablement and Co-Creation'. The main content area displays a table of cluster reservations. The first row, highlighted with yellow circles numbered 7, 8, and 9, represents the user's reservation. The columns in the table are: Project ID, Name, Description, Status, Start, End, Project Credentials, Private key, and Action. The reservation details are: Project ID 208170, Name Bienko OpenShift Cloud Experts, Description For use with Level 3 skills certification., Status Active, Start 2 September 2020 11:01 AM, End 4 September 2020 11:01 AM, Project Credentials cecuser (with a key icon), Private key (with a key icon), and Action (with a dropdown menu). Below the table, there are links for Contact IBM, Privacy, Terms of use, and Accessibility. At the bottom right, there is a language selector set to 'United States - English'.

Project ID	Name	Description	Status	Start	End	Project Credentials	Private key	Action
208170	Bienko OpenShift Cloud Experts	For use with Level 3 skills certification.	Active	2 September 2020 11:01 AM	4 September 2020 11:01 AM	cecuser	View	Link

# Getting Started Accessing CECC Hardware

Clicking the Project Kit URL will open a new page for the **CECC Dashboard**, shown here. This page contains all the details you'll need to know in order to connect with the PowerVM / AIX cluster.

**10:** Under the **Environment Summary** header, take note of the 3 PowerVM deployments. Record the [Hostnames](#) for all three instances.

**11:** Just to the right are the **IP Addresses**. Record all three of these ([129.40.126.225](#) , [...226](#) , [...227](#)) as they will be required later.

**12:** Under the **Account Credentials** header, record the [Username](#) and [Password](#) as shown. These will be needed to remote SSH into your AIX cluster.

The screenshot shows the CECC Dashboard with several sections:

- Environment Summary:** A table listing 3 PowerVM Guest environments. The columns are Type, Hostname, Alias, and IP Address. The rows show:
  - Type: PowerVM Guest, Hostname: p663-pvm1.p663.cecc.ihost.com, Alias: 129.40.126.225
  - Type: PowerVM Guest, Hostname: p663-pvm2.p663.cecc.ihost.com, Alias: 129.40.126.226
  - Type: PowerVM Guest, Hostname: p663-pvm3.p663.cecc.ihost.com, Alias: 129.40.126.227
- ACCOUNT CREDENTIALS & STORAGE:** A section with two subsections:
  - Root Account:** Notes that direct root or primary administrative logins over SSH are disabled.
  - Project Account:** Notes that every project has a generic project user account created locally on each provisioned environment. It also notes that an SSH key pair is automatically generated for the account.
- Account Credentials:** A table with two rows:
  - Username: cecuser
  - Password: JhnuIcfc\_b2t27dBelow the table is the RSA PRIVATE KEY content.

Callouts numbered 10, 11, and 12 point to specific items in the dashboard:

- Callout 10 points to the "Environment Summary" table.
- Callout 11 points to the "IP Address" column in the "Environment Summary" table.
- Callout 12 points to the "Account Credentials" table.

Type	Hostname	Alias	IP Address
PowerVM Guest	p663-pvm1.p663.cecc.ihost.com	129.40.126.225	10
PowerVM Guest	p663-pvm2.p663.cecc.ihost.com	129.40.126.226	11
PowerVM Guest	p663-pvm3.p663.cecc.ihost.com	129.40.126.227	

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAxwmYI8zJUvpjXikrjd80qK0oemKBse+1VhtC+vQn65RV1
Zn7imF83eiaKXQjFnbaekJxxJo9d1UBm7o9jO2PWBSmti8FJxY30EvqNW0W
PM4QVdb+8o7kkcLicMeXPhtKVdb9DYrvyHksud6efArzHu2zqZj5cmTdKB80v
e24yez4fSLugePMdMLL4CzBzshw5WsATjhuk3OF0jVhY2YgEo+CufxhqCZqEk
Ye6qD1DLm90dSu9IJe0uo5dA60y0XPNGeBImPFAjk0kKidTGEuKwHU4QoQqH
4HS/2E39961CpYfBmGLEyGtnp5Zkmsdd9rdDQIDAQABoIBAQCQboj1aaSqP
YPfHEpcrlcPg03Kz8Wf0DQJU4znYvKJxbAoBu4+o7Gdwj2kWS9hDMxc2mlV6j
ba2coX0Rs8LJ1Qz8asxIII+1KjNwe+tMp75E7W9zKvmEgGai013IYz2AbwOxN
AjgwJ+7pTf+JmdSEnA8kweo5tbG6e2sIxjC2ercjnfeZ+MDfM9muvYuMuKYZ
ZhiBu0mEEZc5ehMfr1IN1amcmD9xV5D1AD5ytVh3dpsIUWStvU0xYj3Rblcd
UP4TNJbaFyLvC7BbDZejLrxq4WuEW2Qiciw4ER5yRmlXwA6jyp4aKLqfvMv8s
oKO79keBaOGBA0MdOTikI6YNX0ZPDEUTQpi7JntGWNCK8yyUXaUd0WFvW4w5
ZT/GLDat2vGAouY0cgQP5suVV9T0dcnBurgyuIIhia1VXnsPJA+6DE6MILTDNA
```

# Getting Started Accessing CECC Hardware

Open a **Terminal** window (for Mac) or a **PuTTY / SSH** client (for Windows) in your “Jump Box” (VM) environment.

Execute the following instruction to remote SSH into your PowerVM / AIX cluster. Substitute the placeholder **IP Address** with the **FIRST** address that you recorded in **Step 11**:

```
ssh cecuser@129.40.126.225
```

You will then be prompted for the password that you recorded in **Step 12**:

```
JhnuIcfc_b2t27d
```

Once successfully connected and authenticated, the console should look as shown here on the right.

IBM's internal systems must only be used for conducting IBM's business or for purposes authorized by IBM management. Use is subject to audit at any time by IBM management.

Unauthorized access will be investigated and penalties will be pursued in conformance with applicable laws and regulations. If you are not an authorized user disconnect now.

```
[cecuser@129.40.126.225's password:
```

```
-----  
Welcome to the IBM Systems Cloud for Enablement and Co-Creation  
-----
```

IBM's internal systems must only be used for conducting IBM's business or for purposes authorized by IBM management. Use is subject to audit at any time by IBM management.

For additional information about IaaS, PaaS, STaaS and other offerings see the IBM Systems Cloud for Enablement and Co-Creation Wiki found at <http://cecc-wiki.cecc.ihost.com>

You can find the AIX installation filesets under /cecc/repos/aix72 and an evaluation copy of the xlC compiler under /cecc/repos/lpp

For cecuser some support scripts for common tasks can be found in the support-scripts subdirectory, including one to install xlC. If you are not an IBMer some of the compilers and tools may be preinstalled.

The AIX Linux Toolbox yum repository is configured on this system, so most open source software packages for AIX can be installed with yum. For example to install gcc run: yum install gcc

```
IaaS AIX 7.2 TL 4 SP 2
```

```
PowerVM Guest, POWER8, 2 vCPU, 4356MHz, SMT 4, 4096MB Memory
```

```
cecuser@p663-pvm1:/home/cecuser $ █
```

# Getting Started Accessing CECC Hardware

Congratulations – the AIX / PowerVM cluster is now online and ready for instructions from your SSH console. Red Hat Ansible is not yet deployed inside the environment. To do so, use the accompanying lab guide and follow along to complete this course.

Use the accompanying lab guide to complete this course, and remember to **record when prompted** (with screenshots or in a text log) if asked to do so. You'll need these logs to pass the certification test.



**IMPORTANT:** **Record** the text displayed in the new tab window when you click ...

Refer to the beginning of this document for instructions on how to receive credit for completing this lab. Good luck!

# Section 1

## Installing Ansible

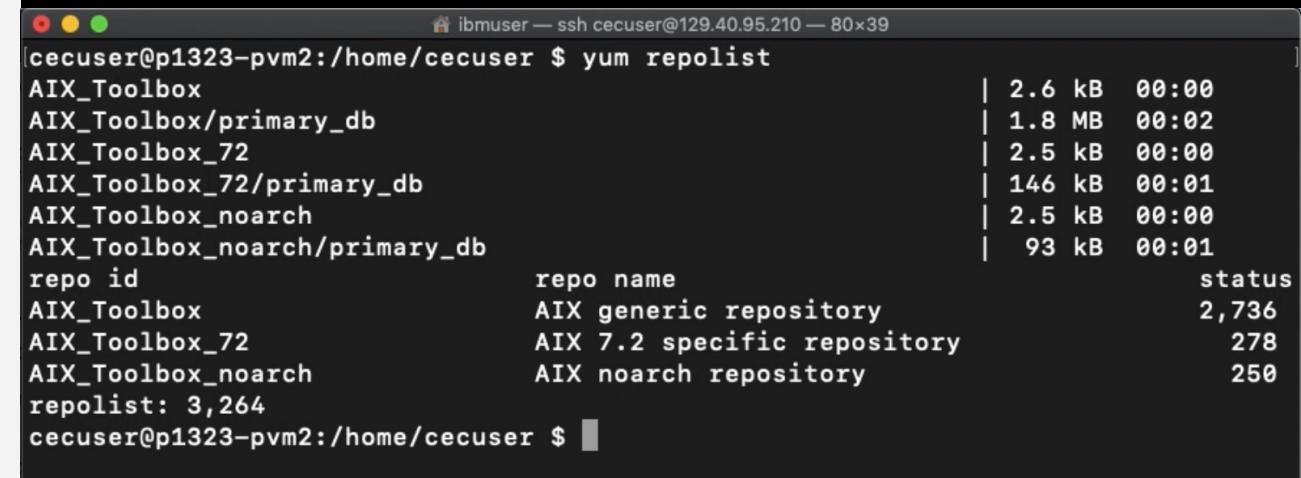
Our first order of business will be to install Ansible on the AIX cluster provisioned via the CECC.

**1:** With the **Terminal** or **Putty** console that you used to remotely connect to the cluster via SSH, enter the following instruction:

```
yum repolist
```

This will grab the necessary repositories for the cluster that Ansible will require for setup.

**Note** that for the remainder of this lab guide, it will be implicit that all code instructions should be entered into the console window (unless otherwise instructed).



The screenshot shows a terminal window with the title bar "ibmuser — ssh cecuser@129.40.95.210 — 80x39". The command "yum repolist" is run, listing several repositories. The output is as follows:

repo id	repo name	status
AIX_Toolbox	AIX generic repository	2,736
AIX_Toolbox_72	AIX 7.2 specific repository	278
AIX_Toolbox_noarch	AIX noarch repository	250
repolist: 3,264		

# Section 1

## Installing Ansible

**2:** Next, we will need to install Ansible. One of the remarkable features of Ansible's automation is how elegant and simple it makes even complex tasks like installing software across a cluster.

Execute the following instruction to install Ansible on the AIX cluster:

```
sudo yum install ansible
```

Enter **Y** when prompted.

**3:** Verify the integrity of your Ansible installation:

```
ansible --version
```

```
Downloading Packages:
(1/8): ansible-2.9.7-1.aix6.1.noarch.rpm | 20 MB 00:04
(2/8): python3-3.7.6-1.aix6.1.ppc.rpm | 38 MB 00:07
(3/8): python3-jinja2-2.10.3-1.aix6.1.noarch.rpm | 299 kB 00:00
(4/8): python3-jmespath-0.9.4-1.aix6.1.noarch.rpm | 45 kB 00:00
(5/8): python3-markupsafe-1.1.1-1.aix6.1.ppc.rpm | 43 kB 00:00
(6/8): python3-pyyaml-5.3.1-1.aix6.1.ppc.rpm | 264 kB 00:00
(7/8): python3-six-1.13.0-1.aix6.1.noarch.rpm | 502 kB 00:00
(8/8): sshpass-1.06-2.aix6.1.ppc.rpm | 34 kB 00:00
-----
Total                                         4.6 MB/s | 60 MB 00:13

Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : python3-3.7.6-1.ppc
  Installing : python3-jmespath-0.9.4-1.noarch
  Installing : python3-markupsafe-1.1.1-1.ppc
  Installing : python3-jinja2-2.10.3-1.noarch
  Installing : python3-six-1.13.0-1.noarch
  Installing : python3-pyyaml-5.3.1-1.ppc
  Installing : sshpass-1.06-2.ppc
  Installing : ansible-2.9.7-1.noarch
  1/8
  2/8
  3/8
  4/8
  5/8
  6/8
  7/8
  8/8

Installed:
  ansible.noarch 0:2.9.7-1

Dependency Installed:
  python3.ppc 0:3.7.6-1
  python3-jmespath.noarch 0:0.9.4-1
  python3-pyyaml.ppc 0:5.3.1-1
  sshpass.ppc 0:1.06-2
  python3-jinja2.noarch 0:2.10.3-1
  python3-markupsafe.ppc 0:1.1.1-1
  python3-six.noarch 0:1.13.0-1

Complete!
cecuser@p648-pvm2:/home/cecuser $
```

# Section 1

## Installing Ansible

Next, we need to edit the

`/OPT/FREWARE/ETC/ANSIBLE/ANSIBLE.CFG` file in  
order to disable host key checking and to use  
Python as the default interpreter. Disabling the  
host key checks will save you having to re-enter  
your password throughout the lab.

**4:** Open the config file with the following:

```
sudo vi /opt/freeware/etc/ansible/ansible.cfg
```

If you are not familiar with **vi**, there is a cheat sheet provided in your lab documentation.  
Alternatively, you can use your own editor of preference (although the instructions for how to save and exit your work may differ).

# Section 1

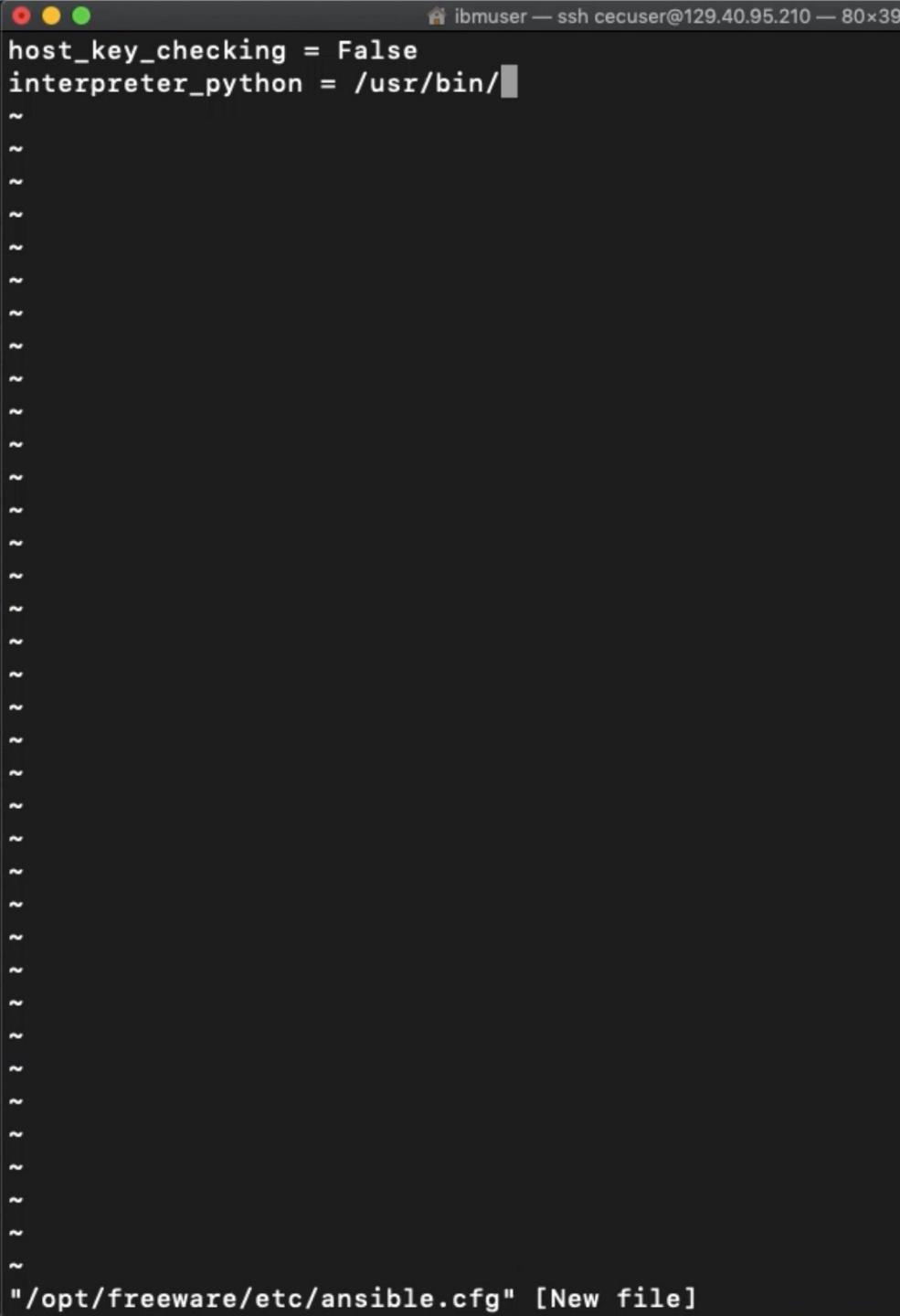
## Installing Ansible

5: Add the following lines after [\[Defaults\]](#):

```
host_key_checking = False
```

```
interpreter_python = /usr/bin/python
```

When satisfied, save and exit the file.



A screenshot of a terminal window titled "ibmuser — ssh cecuser@129.40.95.210 — 80x39". The window shows a configuration file being edited. The top part of the file contains the following lines:

```
host_key_checking = False
interpreter_python = /usr/bin/
```

The rest of the file consists of many blank lines, indicated by the character "~". At the bottom of the terminal window, the path "/opt/freeware/etc/ansible.cfg" is shown in brackets, indicating it is a new file.

```
"/opt/freeware/etc/ansible.cfg" [New file]
```

# Section 2

## Configure Ansible Clients

**6:** Create the [/ANSIBLE-SCRIPTS/](#) directory:

```
sudo mkdir $HOME/ansible-scripts
```

In the Ansible Hosts file, add the IP addresses in the stanzas for AIX LPARs at the top of the file. We will use **[aixservers]** and **[aixclients]** as the group names for organizing clients and servers.

You need to put the IP address of each of your servers in this file. Ansible supports hostname too, but this will not work here due to lack of a DNS-updated date on our test AIX servers.

# Section 2

## Configure Ansible Clients

7: Use the vi editor to modify the following:

```
sudo vi /opt/freeware/etc/ansible/hosts
```

At the **top** of the file, add the following. Modify as appropriate to match the host addresses you recorded earlier — **do not** include the italicized comments with round ellipses.

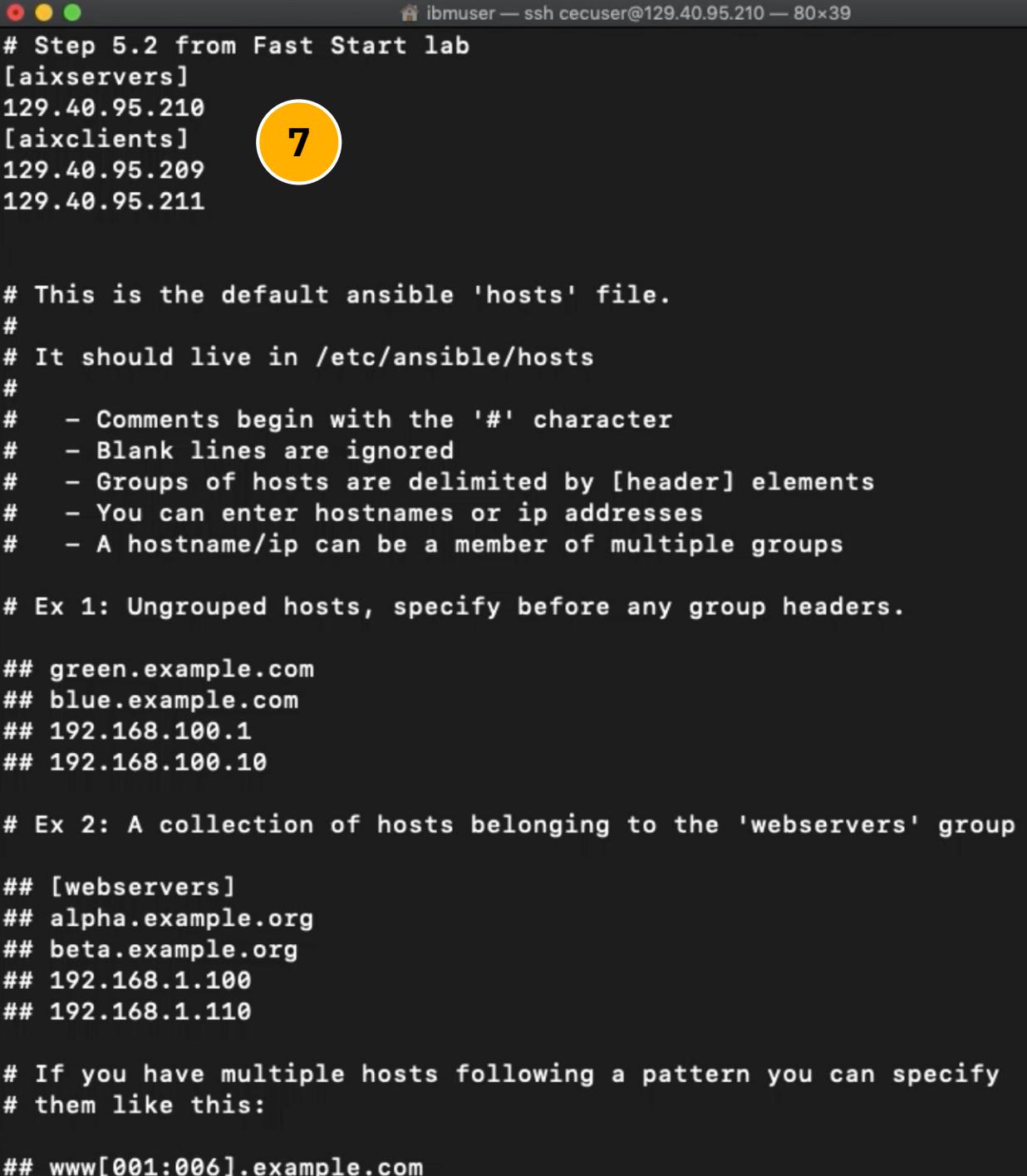
```
[aixservers]
```

```
129.40.yyy.xxx  (Your PowerVM1 IP)
```

```
[aixclients]
```

```
129.40.yyy.xxx  (Your PowerVM2 IP)
```

```
129.40.yyy.xxx  (Your PowerVM3 IP)
```



```
# Step 5.2 from Fast Start lab
[aixservers]
129.40.95.210
[aixclients]
129.40.95.209
129.40.95.211

# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com
```

## Section 2

# Configure Ansible Clients

A comment on generating SSH keys for [CECUSER](#) (and Ansible clusters in general).

Ansible is an “agentless” tool and it relies on SSH to connect to its targets. SSH is an open source secure connection used worldwide.

**Ansible supports both userID/password or password-free SSH connections.**

The environment you will be using to conduct this lab (provided by the CECC) has taken care of SSH key generation ahead of time. No additional work is needed on your part in creating these keys.

# Section 2

## Configure Ansible Clients

**8:** Let's verify the cluster's connectivity to the Ansible client endpoints we just defined.

```
ansible aixclients -m ping
```

This will ping the clients defined in the `/HOSTS/` directory earlier.

You should receive a response from all your **aixclients**, indicating that you can run commands against this AIX LPAR without needing to provide a password each time.



**IMPORTANT: Record** the response displayed in the console returned from the ping request submitted to the aixclients.

```
ibmuser -- ssh cecuser@129.40.95.210 -- 80x39
cecuser@p1323-pvm2:/home/cecuser $ ansible aixclients -m ping
[WARNING]: Platform aix on host 129.40.95.209 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
129.40.95.209 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
```

8

# Section 3

## Running Ansible Modules

9: We need to run the **setup** module to view the characteristics for each target ([aixclient](#)):

```
ansible aixclients -m setup
```

This module provides very extensive information in JSON format about each of your AIX clients. This information is essentially an Inventory of all your server's hardware and software.

The information provided in the output can all be referenced inside an **Ansible Playbook** to customize your tasks. This is a more advanced feature of Ansible which we will not cover here.

```
cecuser@p648-pvm2:/home/ansible-scripts $ ansible aixclients -m setup
[WARNING]: Platform aix on host 129.40.125.241 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
129.40.125.241 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "129.40.125.241"
        ],
        "ansible_all_ipv6_addresses": [
            "::1%1"
        ],
        "ansible_apparmor": {
            "status": "disabled"
        },
        "ansible_architecture": "chrp",
        "ansible_system": "AIX",
        "ansible_user_dir": "/home/cecuser",
        "ansible_user_gecos": "",
        "ansible_user_gid": 204,
        "ansible_user_id": "cecuser",
        "ansible_user_shell": "/usr/bin/ksh",
        "ansible_user_uid": 206,
        "ansible_userspace_bits": "32",
        "ansible_vgs": {
            "rootvg": [
                {
                    "free_pps": "460",
                    "pp_size": "64 megabyte(s)",
                    "pv_name": "hdisk0",
                    "pv_state": "active",
                    "total_pps": "799"
                }
            ]
        },
        "ansible_virtualization_role": "",
        "ansible_virtualization_type": "",
        "discovered_interpreter_python": "/usr/bin/python",
        "gather_subset": [
            "all"
        ],
        "module_setup": true
    },
    "changed": false
}
cecuser@p648-pvm2:/home/ansible-scripts $
```

# Section 3

## Running Ansible Modules

**10:** Identify the hostnames for each of the **aixclients** group members using the command module — which is used for executing OS level commands on Ansible client endpoints.

```
ansible aixclients -m command -a 'hostname'
```

The hostname will be the first value returned in the string.



**IMPORTANT: Record** how many clients respond to hostname command issued to aixclients.

Next, you'll be exposed to how Ansible allows customers to reuse their existing scripts (written in nearly any scripting language) and integrate those with Ansible's automation engine. We will do so via the **SCRIPT** module.

```
cecuser@p648-pvm2:/home/ansible-scripts $ s -m command -a 'hostname'      <
[WARNING]: Platform aix on host 129.40.125.243 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
129.40.125.243 | CHANGED | rc=0 >>
p648-pvm3
[WARNING]: Platform aix on host 129.40.125.241 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
129.40.125.241 | CHANGED | rc=0 >>
p648-pvm1
cecuser@p648-pvm2:/home/ansible-scripts $
```

# Section 3

## Running Ansible Modules

### 11: Create a local script called **testscript.sh**

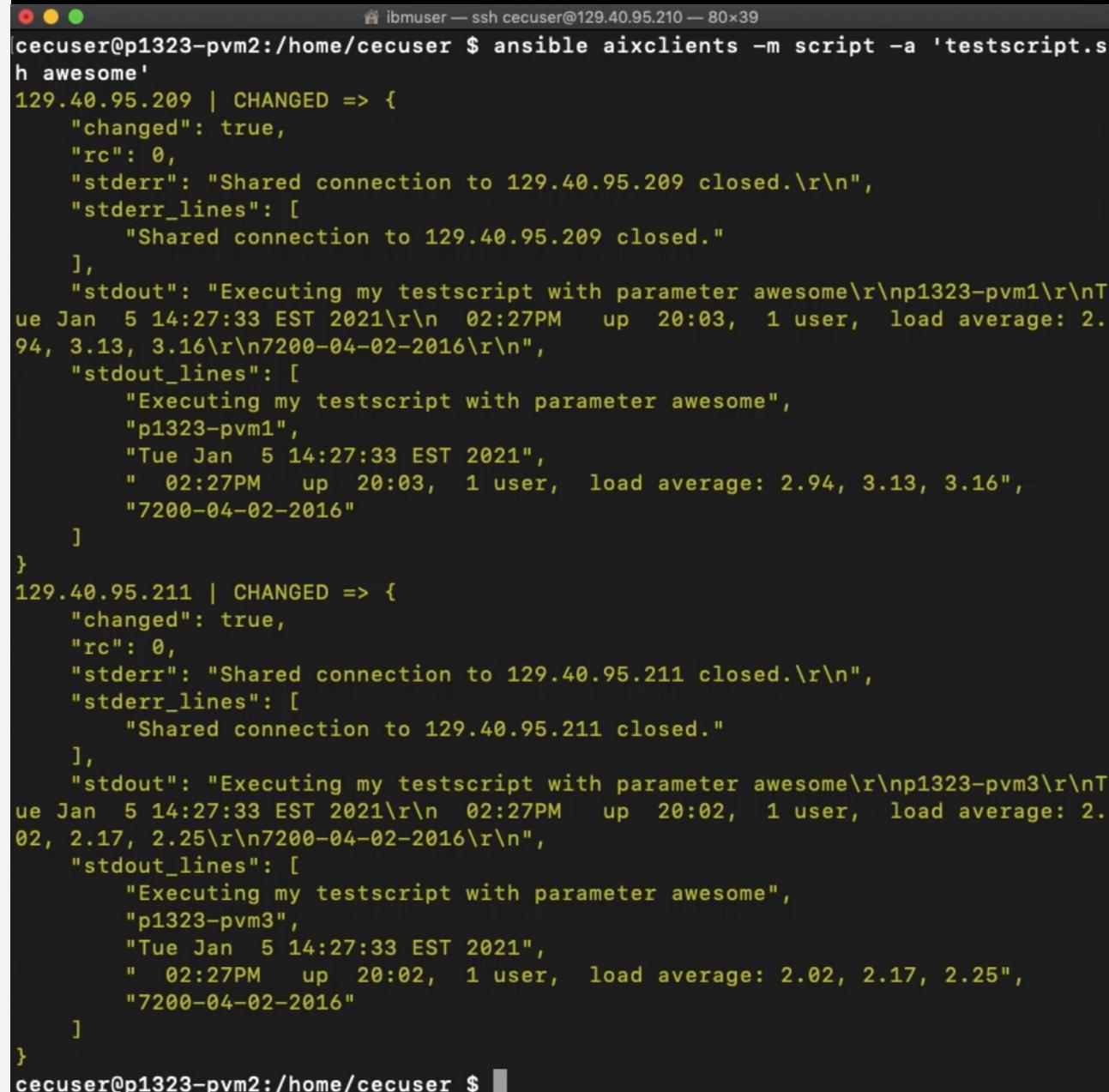
```
echo "Executing my testscript with parameter $1"  
hostname  
date  
uptime  
oslevel -s  
exit
```

### 12: Make **testscript.sh** executable

```
sudo chmod +x testscript.sh
```

### 13: Finally, execute the script across all clients

```
ansible aixclients -m script -a 'testscript.sh awesome'
```



The screenshot shows a terminal window titled 'ibmuser — ssh cecuser@129.40.95.210 — 80x39'. The command run is 'ansible aixclients -m script -a 'testscript.sh awesome''. The output displays results for three hosts:

- Host 1 (129.40.95.209):
  - Changed: true
  - RC: 0
  - Stderr: "Shared connection to 129.40.95.209 closed.\r\n"
  - Stderr\_lines: ["Shared connection to 129.40.95.209 closed."]
  - Stdout: "Executing my testscript with parameter awesome\r\nnp1323-pvm1\r\nTue Jan 5 14:27:33 EST 2021\r\n 02:27PM up 20:03, 1 user, load average: 2.94, 3.13, 3.16\r\n7200-04-02-2016\r\n"
  - Stdout\_lines: ["Executing my testscript with parameter awesome", "p1323-pvm1", "Tue Jan 5 14:27:33 EST 2021", " 02:27PM up 20:03, 1 user, load average: 2.94, 3.13, 3.16", "7200-04-02-2016"]
- Host 2 (129.40.95.211):
  - Changed: true
  - RC: 0
  - Stderr: "Shared connection to 129.40.95.211 closed.\r\n"
  - Stderr\_lines: ["Shared connection to 129.40.95.211 closed."]
  - Stdout: "Executing my testscript with parameter awesome\r\nnp1323-pvm3\r\nTue Jan 5 14:27:33 EST 2021\r\n 02:27PM up 20:02, 1 user, load average: 2.02, 2.17, 2.25\r\n7200-04-02-2016\r\n"
  - Stdout\_lines: ["Executing my testscript with parameter awesome", "p1323-pvm3", "Tue Jan 5 14:27:33 EST 2021", " 02:27PM up 20:02, 1 user, load average: 2.02, 2.17, 2.25", "7200-04-02-2016"]
- Host 3 (129.40.95.210):
  - Changed: true
  - RC: 0
  - Stderr: "Shared connection to 129.40.95.210 closed.\r\n"
  - Stderr\_lines: ["Shared connection to 129.40.95.210 closed."]
  - Stdout: "Executing my testscript with parameter awesome\r\nnp1323-pvm2\r\nTue Jan 5 14:27:33 EST 2021\r\n 02:27PM up 20:03, 1 user, load average: 2.94, 3.13, 3.16\r\n7200-04-02-2016\r\n"
  - Stdout\_lines: ["Executing my testscript with parameter awesome", "p1323-pvm2", "Tue Jan 5 14:27:33 EST 2021", " 02:27PM up 20:03, 1 user, load average: 2.94, 3.13, 3.16", "7200-04-02-2016"]

# Section 3

## Running Ansible Modules

**14:** Let's manipulate our **aixclients** group further but running the **FETCH** module, which will copy client files from their host endpoints into the Ansible controller.

This technique can be immensely valuable for operations teams. For example, we could use the following instructions to copy log files:

```
cd $HOME
```

```
sudo mkdir /clientlogs
```

```
sudo chmod o+w /clientlogs
```

```
ansible aixclients -m fetch -a  
'src=/var/log/cloud-init.log dest=/clientlogs'
```

```
ls -l /clientlogs/*/*log/*
```

```
cecuser@p648-pvm2:/home/ansible-scripts $ sudo mkdir /clientlogs  
cecuser@p648-pvm2:/home/ansible-scripts $ sudo chmod o+w /clientlogs  
cecuser@p648-pvm2:/home/ansible-scripts $ init.log dest=/clientlogs'  
129.40.125.243 | CHANGED => {  
    "changed": true,  
    "checksum": "da39a3ee5e6b4b0d3255bfef95601890af80709",  
    "dest": "/clientlogs/129.40.125.243/var/log/cloud-init.log",  
    "md5sum": "d41d8cd98f00b204e9800998ecf8427e",  
    "remote_checksum": "da39a3ee5e6b4b0d3255bfef95601890af80709",  
    "remote_md5sum": null  
}  
129.40.125.241 | CHANGED => {  
    "changed": true,  
    "checksum": "da39a3ee5e6b4b0d3255bfef95601890af80709",  
    "dest": "/clientlogs/129.40.125.241/var/log/cloud-init.log",  
    "md5sum": "d41d8cd98f00b204e9800998ecf8427e",  
    "remote_checksum": "da39a3ee5e6b4b0d3255bfef95601890af80709",  
    "remote_md5sum": null  
}  
cecuser@p648-pvm2:/home/ansible-scripts $ ls -l /clientlogs/*/*log/*  
-rw-r--r-- 1 cecuser cecuser 0 Aug 21 18:16 /clientlogs/129.40.1  
25.241/var/log/cloud-init.log  
-rw-r--r-- 1 cecuser cecuser 0 Aug 21 18:16 /clientlogs/129.40.1  
25.243/var/log/cloud-init.log
```

# Section 3

## Running Ansible Modules

**15:** We can go further. Let's suppose the administrator of this environment wanted to change access permissions on files stored on **aixclients** group members.

First, identify the current level of permissions:

```
ansible aixclients -m command -a 'ls -l  
/var/log/cloud-init.log'
```

Then, add write permissions for that group:

```
ansible aixclients -m file -become -a  
'path=/var/log/cloud-init.log mode=g+w'
```

Finally, verify those permissions were changed:

```
ansible aixclients -m command -a 'ls -l  
/var/log/cloud-init.log'
```

```
cecuser@p648-pvm2:/home/ansible-scripts $ g/cloud-init.log mode=g+w' <  
[WARNING]: Platform aix on host 129.40.125.241 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.241 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "gid": 0,  
    "group": "system",  
    "mode": "0664",  
    "owner": "root",  
    "path": "/var/log/cloud-init.log",  
    "size": 0,  
    "state": "file",  
    "uid": 0  
}  
[WARNING]: Platform aix on host 129.40.125.243 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.243 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "gid": 0,  
    "group": "system",  
    "mode": "0664",  
    "owner": "root",  
    "path": "/var/log/cloud-init.log",  
    "size": 0,  
    "state": "file",  
    "uid": 0  
}  
cecuser@p648-pvm2:/home/ansible-scripts $ □
```

# Section 3

## Running Ansible Modules

**16:** The number of groups (and members to those groups) can easily be expanded with Ansible. Use the following commands to create a new group and user on **aixclients**:

```
ansible aixclients -m group -become -a  
'name=team1 state=present'
```

```
ansible aixclients -m user -become -a  
'name=user1 group=team1'
```

Then list the attributes of **user1**:

```
ansible aixclients -m command -a 'lsuser user1'
```

```
cecuser@p648-pvm2:/home/ansible-scripts $ -a 'name=user1 group=team1' <  
[WARNING]: Platform aix on host 129.40.125.243 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.243 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "comment": "",  
    "create_home": true,  
    "group": 207,  
    "home": "/home/user1",  
    "name": "user1",  
    "shell": "/usr/bin/ksh",  
    "state": "present",  
    "system": false,  
    "uid": 207  
}  
[WARNING]: Platform aix on host 129.40.125.241 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.241 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "comment": "",  
    "create_home": true,  
    "group": 207,  
    "home": "/home/user1",  
    "name": "user1",  
    "shell": "/usr/bin/ksh",  
    "state": "present",  
    "system": false,  
    "uid": 207  
}  
cecuser@p648-pvm2:/home/ansible-scripts $ □
```

# Section 3

## Running Ansible Modules

**17:** Next, clean up (delete) this new group and user with the following command:

```
ansible aixclients -m user -become -a  
'name=user1 state=absent remove=yes'
```

```
ansible aixclients -m group -become -a  
'name=team1 state=absent'
```

Verify that the group and user no longer exist:

```
ansible aixclients -m command -become -a 'lsuser  
user1'
```



**IMPORTANT:** Record the FAILED, User ... does not exist response from aixclients, which indicates the group was deleted successfully.

```
cecuser@p648-pvm2:/home/ansible-scripts $ -m user -become -a 'name=user1 s*'  
[WARNING]: Platform aix on host 129.40.125.241 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
n  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.241 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "force": false,  
    "name": "user1",  
    "remove": true,  
    "state": "absent"  
}  
[WARNING]: Platform aix on host 129.40.125.243 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
n  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.243 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "force": false,  
    "name": "user1",  
    "remove": true,  
    "state": "absent"  
}  
cecuser@p648-pvm2:/home/ansible-scripts $  
cecuser@p648-pvm2:/home/ansible-scripts $ ome -a 'lsuser user1'  
[WARNING]: Platform aix on host 129.40.125.241 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
n  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.241 | FAILED | rc=2 >>  
3004-687 User "user1" does not exist.non-zero return code  
[WARNING]: Platform aix on host 129.40.125.243 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
n  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.243 | FAILED | rc=2 >>  
3004-687 User "user1" does not exist.non-zero return code  
cecuser@p648-pvm2:/home/ansible-scripts $
```

# Section 4

## Ansible Playbooks

In this section, we will create a few Ansible **Playbooks** to show how they can be used to run Modules and perform orchestration.

An Ansible Playbook is way to organize a collection of tasks into one file. These are written in **YAML** (Yet Another Markup Language). If you'd like to dig further into Playbooks, we highly recommend the resources listed in the Appendix.



**Note:** Ansible Playbooks are very particular about indentation. Remember to use **TABS** to create indentation at the beginning of new lines, as shown in the examples.

# Section 4

## Ansible Playbooks

**18:** Our first step will be to create and execute a simple Playbook to use the [COPY](#) module for replicating a file from the Ansible controller to all connected clients (**aixclients**). Set up a directory on the controller node for the Playbook:

```
echo 'Hello World' | cat > /tmp/testfile
```

```
cd $HOME/ansible-scripts
```

Create a file called [copy\\_testfile.playbook.yml](#) in the directory. We will define the contents in the next step:

```
sudo vi copy_testfile.playbook.yml
```

# Section 4

## Ansible Playbooks

**19:** Define the Playbook as shown:

```
- hosts: aixclients
  tasks:
    - copy:
        src: /tmp/testfile
        dest: /tmp/client_testfile
```

Then run the Playbook with the following:

```
ansible-playbook copy_testfile.playbook.yml
```



**IMPORTANT:** Record the names of all jobs, such as PLAY [aixclients], in the sequence they are performed, that are executed by the Playbook.

```
- hosts: aixclients
  tasks:
    - copy:
        src: /tmp/testfile
        dest: /tmp/client_testfile
~
~
~

cecuser@p1323-pvm2:/home/cecuser/ansible-scripts $ ansible-playbook copy_testfile.playbook.yml

PLAY [aixclients] ****
TASK [Gathering Facts] ****
[WARNING]: Platform aix on host 129.40.95.209 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [129.40.95.209]
[WARNING]: Platform aix on host 129.40.95.211 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [129.40.95.211]

TASK [copy] ****
changed: [129.40.95.209]
changed: [129.40.95.211]
```

# Section 4

## Ansible Playbooks

**20:** Verify that the file was copied into the clients:

```
ansible aixclients -m command -a 'cat  
/tmp/client_testfile'
```



**IMPORTANT:** Record the response from the two client hosts that results from these instructions.

**21:** Our final step will be to craft a Playbook that uses the `STAT`, `DEBUG`, and `FILE` Ansible modules. These will display messages on the existence of the testfile, delete the testfile, and verify that the testfile no longer exists, respectively.

Create a file named **delete\_testfile.playbook.yml**

```
sudo vi delete_testfile.playbook.yml
```

```
cecuser@p648-pvm2:/home/ansible-scripts $ cat /tmp/client_testfile' <  
[WARNING]: Platform aix on host 129.40.125.241 is using the discovered Python  
interpreter at /usr/bin/python, but future installation of another Python  
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen  
ce_appendices/interpreter_discovery.html for more information.  
129.40.125.241 | CHANGED | rc=0 >>
```

# Section 4

# Ansible Playbooks

## 22: Define **delete\_testfile.playbook.yml**:

```
---
```

```
- hosts: aixclients
  tasks:
    - name: does the testfile exist
      stat:
        path: /tmp/client_testfile
      register: sym

    - debug:
        msg: "Our /tmp/client_testfile does NOT exist"
      when: sym.stat.exists == False

    - debug:
        msg: "Our /tmp/client_testfile does exist"
      when: sym.stat.exists == True

    - name: delete the testfile
      file:
        path: /tmp/client_testfile
        state: absent

    - name: does the testfile exist
      stat:
        path: /tmp/client_testfile
      register: sym

    - debug:
        msg: "Our /tmp/client_testfile does NOT exist"
      when: sym.stat.exists == False

    - debug:
        msg: "Our /tmp/client_testfile does exist"
      when: sym.stat.exists == True
```

```
  - hosts: aixclients
    tasks:
      - name: does the testfile exist
        stat:
          path: /tmp/client_testfile
        register: sym

      - debug:
          msg: "Our /tmp/client_testfile does NOT exist"
        when: sym.stat.exists == False

      - debug:
          msg: "Our /tmp/client_testfile DOES exist"
        when: sym.stat.exists == True

      - name: delete the testfile
        file:
          path: /tmp/client_testfile
          state: absent

      - name: does the testfile exist
        file:
          stat:
            path: /tmp/client_testfile
          register: sym

      - debug:
          msg: "Our /tmp/client_testfile does NOT exist"
        when: sym.stat.exists == False

      - debug:
          msg: "Our /tmp/client_testfile DOES exist"
        when: sym.stat.exists == True
```

~  
~  
~

"delete\_testfile.playbook.yml" 35 lines, 725 characters

# Section 4

# Ansible Playbooks

23: Execute the Playbook:

```
ansible-playbook delete_testfile.playbook.yml
```

**Congratulations** – you've taken your first steps towards implementing fully-automated systems using Red Hat Ansible.

If you have additional questions or opportunities you'd like to discuss with the author, please feel free to reach out.

```
TASK [debug] ****
** skipping: [129.40.125.241]
skipping: [129.40.125.243]

TASK [debug] ****
** ok: [129.40.125.241] => {
    "msg": "Our /tmp/client_testfile does exist"
}
ok: [129.40.125.243] => {
    "msg": "Our /tmp/client_testfile does exist"
}

TASK [delete the testfile] ****
** changed: [129.40.125.243]
changed: [129.40.125.241]

TASK [does the testfile exist] ****
** ok: [129.40.125.243]
ok: [129.40.125.241]

TASK [debug] ****
** ok: [129.40.125.241] => {
    "msg": "Our /tmp/client_testfile does NOT exist"
}
ok: [129.40.125.243] => {
    "msg": "Our /tmp/client_testfile does NOT exist"
}

TASK [debug] ****
** skipping: [129.40.125.241]
skipping: [129.40.125.243]

PLAY RECAP ****
** 129.40.125.241 : ok=6    changed=1    unreachable=0    failed=0
      skipped=2    rescued=0     ignored=0
129.40.125.243 : ok=6    changed=1    unreachable=0    failed=0
      skipped=2    rescued=0     ignored=0

cecuser@p648-pvm2:/home/ansible-scripts $ 
```

# Appendix

## Call to Action & Resources

Visit [\*\*Modernize Infrastructure & Apps\*\*](#) and [\*\*Private Cloud Solutions\*\*](#) sales kits on Seismic for additional resources!

Learn about IBM's [\*\*Hybrid Multicloud Showcase\*\*](#) demos via the cloud brochure.

Additional [\*\*IBM Garage for Systems\*\*](#) and [\*\*Lab Services\*\*](#) resources are available.



### CALL TO ACTION

Apply the lessons learned here with clients and customers. Don't simply have the conversation — make it a demonstration.

**Showcase this demo with 5 clients in 1Q21**



# Appendix Closing Out

Additional resources on Ansible and Playbooks:

<http://docs.ansible.com>

<https://github.com/aixoss/ansible-playbooks>

<https://github.com/kairoaraujo/ansible-aix-support>

<https://www.djouxtech.net/posts/ansible-on-aix>

[Managing AIX Updates Best Practices](#)

[Fix Level Recommendation Tool Vulnerability Checker](#)

[https://github.com/pmarkham/writing-ansible-modules-in-bash/blob/master/ansible\\_bash\\_modules.md](https://github.com/pmarkham/writing-ansible-modules-in-bash/blob/master/ansible_bash_modules.md)



# Legal Notices

Copyright © 2021 by International Business Machines Corporation. All rights reserved.

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml).

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER OR IMPLIED. IBM LY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504- 785  
U.S.A.

