**Overview**

[Ansible](https://www.ansible.com/) is an open-source tool that automates cloud provisioning, configuration management, and application deployments. Using Ansible you can provision virtual machines, containers, network, and complete cloud infrastructures. In addition, Ansible allows you to automate the deployment and configuration of resources in your environment.

Ansible includes a suite of [Ansible modules](https://docs.ansible.com/ansible/2.9/modules/modules_by_category.html) that can be executed directly on remote hosts or via playbooks. Users can also create their own modules. Modules can be used to control system resources - such as services, packages, or files - or execute system commands.

For interacting with Azure services, Ansible includes a suite of [Ansible cloud modules](https://docs.ansible.com/ansible/2.9/modules/list_of_cloud_modules.html#azure) that provides the tools to easily create and orchestrate your infrastructure on Azure.

***What’s covered in this lab***

In this lab, you will see

1. How Ansible can be used to implement Infrastructure as Code (IaC)
2. How to automate infrastructure deployments in the Cloud with Ansible and Azure pipelines.

The following image will walk you through all the steps explained in this lab

***Setting up the Environment***

***Task 1: Create an Azure service principal with Azure CLI***

Ansible includes a suite of modules for interacting with Azure Resource Manager, giving you the tools to easily create and orchestrate infrastructure on the Microsoft Azure Cloud. Using the Azure Resource Manager modules requires authenticating with the Azure API. In this lab, you will use Azure service principal for authentication.

1. Login to the [Azure portal](https://portal.azure.com/).
2. Click **Cloud Shell** and select **Bash**.

Graphical user interface, application, Word

Description automatically generated

1. Enter the following command to get Azure SubscriptionID and copy the same to notepad.

az account show

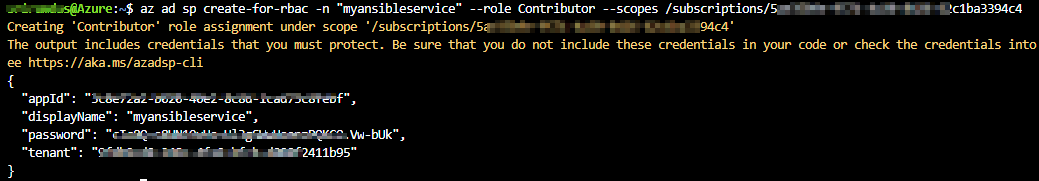
A screenshot of a computer

Description automatically generated with medium confidence

1. Enter the following command by replacing ServicePrincipalName with your desired value and Subscription ID from the previous step.

az ad sp create-for-rbac --name ServicePrincipalName --role Contributor --scopes /subscriptions/<subscriptionid>

It will give you a JSON output as shown in the image. Copy the output to notepad. This details required in your next tasks.



For more information about Azure service principal click [here](https://docs.microsoft.com/en-us/cli/azure/create-an-azure-service-principal-azure-cli?view=azure-cli-latest#create-the-service-principal).

***Task 2: Configure Ansible in a Linux machine***

To create and provision the resources in Azure with Ansible, we need to have a Linux VM with Ansible configured. In this exercise, you will deploy an Azure Linux VM and configure Ansible on the virtual machine

1. In the Azure Cloud shell enter below command to create Azure resource group

az group create --name AnsibleVM --location eastus

1. Create the Azure virtual machine for Ansible.

az vm create --resource-group AnsibleVM --name AnsibleVM --image OpenLogic:CentOS:7.7:latest --admin-username azureuser --admin-password <password>

Replace the <password> with your password.

1. Once the deployment is successful, navigate to the resource group and select the VM.

Graphical user interface, text, application, email

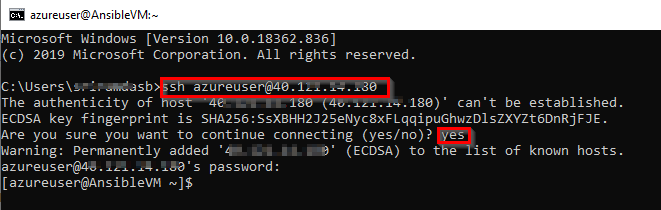
Description automatically generated

1. Select **Overview** and copy the **Public IP address**.

Graphical user interface, text, application, email

Description automatically generated

1. Open a Command prompt and enter the below command ssh azureuser@<PublicIP> to login to VM. It will prompt for confirmation to connect, type **Yes** and provide the Password you have given in step 1.



Note: Replace *azureuser* with your VM username in the above command.

1. Run the following commands to configure Ansible on Centos:
2. #!/bin/bash
3. # Update all packages that have available updates.
4. sudo yum update -y
5. # Install Python 3 and pip.
6. sudo yum install -y python3-pip
7. # Upgrade pip3.
8. sudo pip3 install --upgrade pip
9. # Install Ansible.
10. pip3 install "ansible==2.9.17"
11. # Install Ansible azure\_rm module for interacting with Azure.
12. pip3 install ansible[azure]
13. Now we must create a directory named **.azure** in the home directory and a credentials file under it. This local credentials file is to provide credentials to Ansible. Type the following commands to create them.
14. mkdir ~/.azure
15. Sudo vi ~/.azure/credentials
16. Insert the following lines into the **credentials** file. Replace the placeholders with the information from the service principal details you copied in the previuous task. Press **Ctrl+O** to save the file and **Ctrl+X** to exit from the text editor.

[default]

subscription\_id=<your-Azure-subscription\_id>

client\_id=<azure service-principal-appid>

secret=<azure service-principal-password>

tenant=<azure serviceprincipal-tenant>

1. Run sudo vi .bashrc and insert the following text into **.bashrc**. Press **Ctrl+O** to save the file and **Ctrl+X** to exit from the text editor.

PATH=$PATH:$HOME/.local/bin:$HOME/bin

1. Ansible is an agentless architecture based automation tool . Only it needs ssh authentication using Ansible Control Machine private/public key pair. Now let us create a pair of private and public keys. Run the following command to generate a private/public key pair for ssh and to install the public key in the local machine.

ssh-keygen -t rsa

chmod 755 ~/.ssh

touch ~/.ssh/authorized\_keys

chmod 644 ~/.ssh/authorized\_keys

ssh-copy-id azureuser@127.0.0.1

Note: Replace **azureuser** with your VM username in the above command.

Text

Description automatically generated

1. In the next task, you need SSH private key to created SSH endpoint in Azure DevOps service. Run the following command to get the private key. Copy the private key to notepad.

cat ~/.ssh/id\_rsa

Text

Description automatically generated

***Task 3: Create a SSH Service Connection in Azure DevOps***

To connect and run playbooks through Ansible VM in Azure pipelines, we need to have a connection between Azure DevOps and Ansible VM. This service connection provides authentication to Ansible.

1. Navigate to **Project Settings** –> **Service Connections**. Select **Create service connection**.

Graphical user interface, application, Word

Description automatically generated

1. In **New Service Connection** windows select **SSH** and click **Next**

Graphical user interface, text, application

Description automatically generated

1. In **New SSH service connection** window provide the required details and click **Save** to save the connection.

Graphical user interface, text, application

Description automatically generated

***Exercise 1: Examine the Ansible playbook (IaC) in your Source code***

In this lab, we will use **SmartHotel360-CouponManagement**, a sample Java application backed by a MySQL database. We will examine the Ansible playbook which helps you to provision the Azure resources required to deploy SmartHotel java application.

1. Navigate to your project. Select **Repos**.
2. Select the **webapp.yml** file under the **ansible-scripts** folder. Go through the code.

A picture containing graphical user interface

Description automatically generated

**webapp.yml** is an Ansible playbook file written in YAML format. [Ansible Playbooks](https://docs.ansible.com/ansible/latest/user_guide/playbooks.html) are Ansible’s configuration, deployment, and orchestration language. They can describe a policy you want your remote systems to enforce or a set of steps in a general IT process. These playbooks use YAML file format to define a model of a configuration or a process.

Ansible includes a suite of modules for interacting with Azure Resource Manager, giving you the tools to easily create and orchestrate infrastructure on the Microsoft Azure Cloud.

In this example, we want to deploy an Azure Resource group, App service plan, App service and MySQL database required to deploy the website. And we have added playbook file (Infrastructure as Code) to source control repository in your Azure DevOps project which can deploy the required Azure resources.

Update the playbook **webapp.yml** as the following.

Change the name of the App service from Linux to Windows as the following image Text

Description automatically generated

Also, change the EndIpAdress from 255.255.255.255 to 0.0.0.0 as the following image

Graphical user interface, application

Description automatically generated

If you would like to learn more about the Ansible playbooks for Azure click [here](https://docs.ansible.com/ansible/2.5/scenario_guides/guide_azure.html).

### *Exercise 2: Build your application using Azure CI Pipeline*

In this exercise, you will build your application and publish the required files to an artifact called drop.

1. Navigate to **Pipelines –> Pipelines**. Create **Ansible-CI**

Your build pipeline will look like as below. This CI pipeline has tasks to compile Java project. The Maven in the pipeline will restore dependencies, build, test and publish the build output into a war file (package) which can be deployed to a web application.

Graphical user interface, text, application, email

Description automatically generated

For more guidance on how to build Java projects with Azure Pipelines see [here](https://docs.microsoft.com/en-us/azure/devops/pipelines/languages/java?toc=%2Fazure%2Fdevops%2Fjava%2Ftoc.json&bc=%2Fazure%2Fdevops%2Fpipelines%2Fbreadcrumb%2Ftoc.json&view=vsts).

1. In addition to the application build, we need to publish Ansible scripts so that it will be available in CD pipeline. So, we configured Copy files task to copy Ansible playbook **.yml** and the java web package **.war** file to Artifacts directory.

Graphical user interface, application, email

Description automatically generated

1. Now click **Queue** to trigger the build. Once the build success, verify that the artifacts have **ansible\_scripts** folder and **ROOT.war** file in the drop.

Graphical user interface, text, application, email, website

Description automatically generated

### *Exercise 3: Deploy resources using Ansible in Azure CD Pipeline*

In this exercise, we will create azure resources using Ansible as part of our deployment (CD) pipeline and deploy the SmartHotel Coupon management application to the App service provisioned by Ansible.

Navigate to **Pipelines » Releases**. Create  **Ansible-CD**

1. Once you are done **Save** the changes and **Create a release**.

Graphical user interface, text, application

Description automatically generated

1. Once the release is success navigate to your Azure portal. Search for **smh360web** in App services. Select the app that created with **smh360-xxxx** and browse to view the application deployed.

App settings

-SPRING\_DATASOURCE\_USERNAME $(mysqlAdmin)@$(mysqlServerName) -SPRING\_DATASOURCE\_PASSWORD $(mysqlAdminSecret) -SPRING\_DATASOURCE\_URL

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface

Description automatically generated

1. You can **Login** to the site with the following credentials.

Username: me@smarthotel360.com

Password: 1234

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

### *Summary*

In this lab, you have learnt how to deploy Azure resources automatically with **Ansible** and deploy the application with **Azure Pipelines**.

For Ansible on Azure documentation and Quick starts click [here](https://docs.microsoft.com/en-us/azure/ansible/).