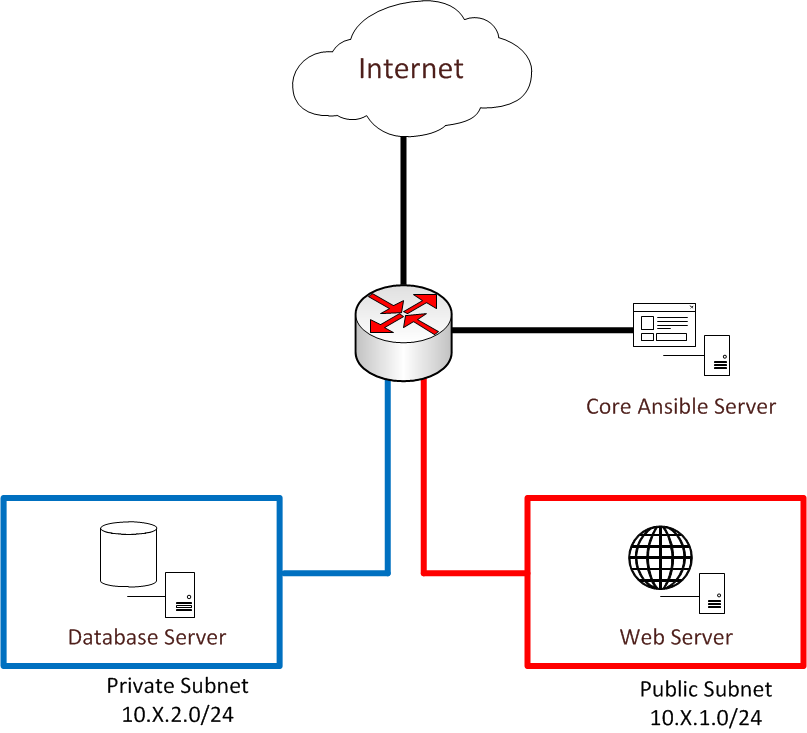
# NextHop Tech Talk – Ansible

## Topology



The topology for this demo is shown above. You will receive the specific details about your individual topology via email. These details will include:

* Private IP addresses of the web and database servers
* Public IP address of the web server
* A username and password for logging into the “Core Ansible Server”

We will log into the Core Ansible Server with the provided credentials. From this server, we will be able to execute Ansible playbooks against our hosts. The topology is designed with two subnets: private and public. The webserver in the public subnet has an associated public IP address. The database server in the private subnet is only accessible via the Core Ansible Server and the Web Server. Internet connectivity is provided via a NAT.

## Demo Scenario

The goal of this demo scenario is to provision a WordPress site via Ansible playbooks with little administrator interaction. Ansible will be responsible for installing the web server software (Apache) and downloading WordPress. The database will be installed and configured using Ansible modules that work with MySQL.

# Instructions

Note: commands are *italicized* in the instructions below.

## Prepare servers to be managed by Ansible

There are 3 pieces of configuration that should be completed on the servers that we want to manage with Ansible:

* Add an “ansible” user
* Allow the ansible user to perform passwordless sudo
* Copy SSH keys from the Ansible control server to the servers that we want to manage

The first two steps have already been completed. However, we still need to create an SSH key pair and copy the public key to the webserver and database server.

1. Log into the Core Ansible Server using your emailed username (i.e. student1) and password.
2. Create an SSH key pair
   1. *ssh-keygen*
3. Copy the SSH public key to the webserver and database server. Enter your emailed password when prompted. Be sure to copy the key to both the webserver and database server.
   1. *ssh-copy-id ansible@<Private IP of server>*
4. Confirm that you can log into both the web server and database server with your key.
   1. *ssh ansible@<Private IP of server>*

## Download and configure the Ansible scripts

To save time, we have a pre-written Ansible script that can be used to deploy the environment. Perform the following on the Core Ansible server:

1. Download the pre-written Ansible script
   1. *wget https://www.acritelli.com/presentations/nexthopAnsible2017/main.yml*
2. Download the pre-written inventory file
   1. *wget https://www.acritelli.com/presentations/nexthopAnsible2017/hosts*
3. Modify the inventory file with the appropriate private IPs for each group (webservers and dbservers).
4. Review the main.yml file to understand the different tasks and modules being used
5. Edit the main.yml file:
   1. Change the “webserver” variable under the dbservers play so that the IP matches your private IP address
   2. Change the “db\_user” and “db\_password” variable to your desired values.
6. Run the ansible script
   1. *ansible-playbook –i hosts main.yml*

## Verify and deploy WordPress

1. Open a web browser and navigate to: <webserver public IP>/wordpress
2. Complete the WordPress installation steps as prompted.
   1. For the database, specify the following:
      1. Database name – leave as default (wordpress)
      2. Username: the username that you specified in the Ansible playbook
      3. Password: the password that you specified in the Ansible playbook
      4. Database host: the private IP address of your database server
      5. Table prefix – leave as default (wp\_)
   2. Customize the website (site title, username, password, etc.) in any way that you wish.
3. You should now have a fully functional WordPress site! Navigating to <webserver public IP>/wordpress should now show your website!

**Important:** You shouldn’t use this set of playbooks in production. They’re only basic and do not implement any security best practices. For example: the root password of the MySQL user is left unchanged.