Runbook

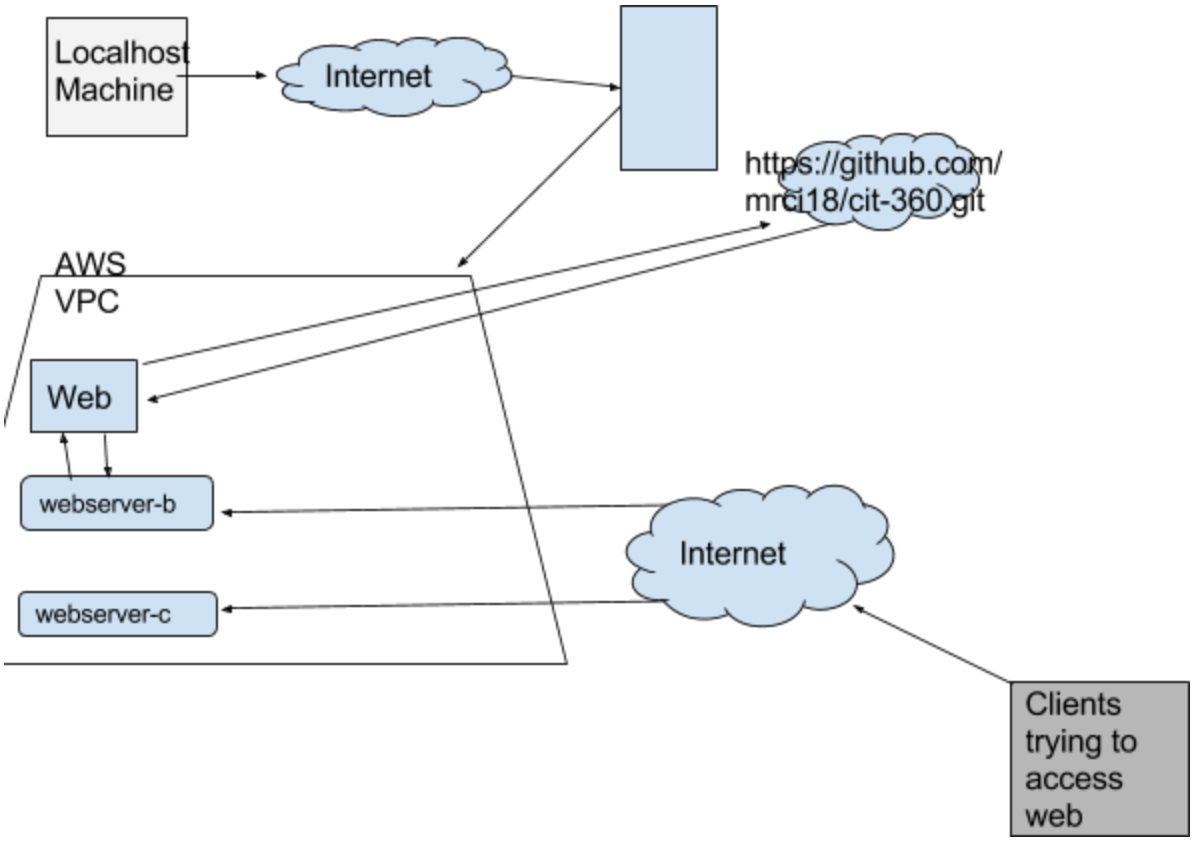
Short Description

The purpose of this runbook is to describe how to launch an aws vpc built with 3 instances to run a DB and Web server.

Required Software

1. A Linux environment computer
    a. Virtualbox
2. Git
3. Ansible
4. AWS

Architecture Diagram



Deployment

Describe how to deploy the software. Note any required dependencies for the deploy (for example, the tool used for deployment). Describe the steps, including commands, to run in the proper order and the expected output of the commands.

1. Install Git and clone cit-360 repo (Work done one debian based environment)
    a. Go to your terminal and install git with the command
        i. sudo yum install git-all
    b. Clone the cit-360 repo by inputting the command
        i. Git clone https://github.com/mrci18/cit-360
2. Install terraform
    a. Select the correct version for your device on the website and download
        i. https://www.terraform.io/downloads.html
        ii. In this case it was Linux 64bit
        iii. Extract the file within the path /home/you-user-name/
    b. Make sure the terraform file you downloaded is on the correct path. Input command in terminal
        i. PATH=/usr/local/terraform/bin:/home/your-user-name/terraform:$PATH
3. Launch the terraform infrastructure
    a. Naviage to the cit-360 repo that you cloned
        i. Navigate to the terraform directory
        ii. Input the following command to build the aws infrastructure
            1. Terraform apply
4. Connect to EC2 instance web
    a. Log on to AWS account
    b. Navigate to running instances
    c. Select “web” and click Connect
    d. Use the example command to ssh into the instance
        i. ssh -i "cit360.pem" ec2-user@ec2-35-164-80-192.us-west-2.compute.amazonaws.com
        ii. Replace “cit360.pem” with the exact location of the permission key on your host computer
            1. Ex. /home/charles/Desktop/cit360.pem
            2. Type “yes” when prompted
5. Install Ansible on EC2 instance
    a. Now that you have ssh in the EC2, download ansible with the command
        i. Sudo pip install ansible
6. Install git and copy cit-360 repo on EC2 instance
    a. Follow step 1
7. Run ansible playbooks
    a. Navigate to cit-360/ansible/
    b. Once you are inside ansible run db.yml file with the following command

i. ansible-playbook -i hosts.ini db.yml

c. Once you are inside ansible run web.yml file with the following command
   i. ansible-playbook -i hosts.ini web.yml --ask-vault-pass
   ii. When prompted use password: “secret”

Issues

Title: Terraform download
Description: The terraform file that is extracted needs to be on the correct path.
Remediation Steps:

1. Extract the terraform file somewhere easy to navigate. The best way would be to follow the path that the terraform website uses to troubleshoot the problem
2. PATH=/usr/local/terraform/bin:/**home/your-user-name**/terraform:$PATH

Title: Create key pair
Description: If you have lost your key pair file or need to create one, log on to AWS and create a new one
Remediation Steps:

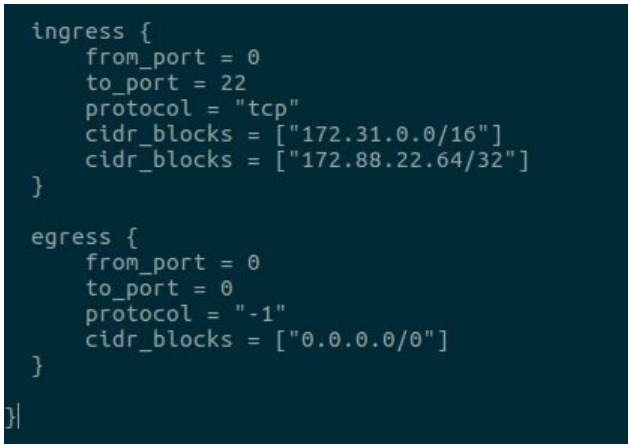
1. Log on to AWS
2. On the left hand side click on “Services”
3. Select EC2
   a. On the left hand side select “Key Pairs”
   b. Create Key Pair and name it
4. Save key pair file somewhere you won’t lose it

Title: Can’t connect to AWS Instance
Description: The current ip address of the aws security group named allow_to_22 must be updated on the infra.tf file
Remediation Steps:

1. Google the phrase
   a. What is my ip address
   b. Copy and paste the address to replace the current
      i. 172.88.22.64

Title: Keeps timing out when I try to ssh into instance
Description: Make sure that the security group that is associated with the instance allows access from your current public IP address to an instance on port 22. Or make sure to add an egress rule to your security group. “When creating a new Security Group inside a VPC, **Terraform will remove this default rule”**
Remediation Steps:

1. Use the following terraform code as a guide

2. 
```
ingress {
    from_port = 0
    to_port = 22
    protocol = "tcp"
    cidr_blocks = ["172.31.0.0/16"]
    cidr_blocks = ["172.88.22.64/32"]
}

egress {
    from_port = 0
    to_port = 0
    protocol = "-1"
    cidr_blocks = ["0.0.0.0/0"]
}

}
```

Title: Can’t install packages as root on AWS instance
Description: The reason is still unknown but for now download packages with sudo command instead of logging in as root
Remediation Steps:

1. Use command such as
    a. Sudo pip install ansible