

Jenkins

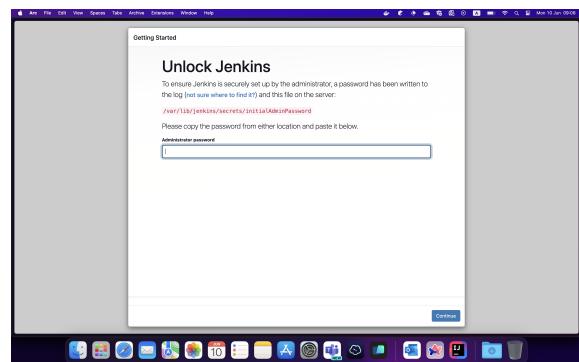
Installation

- Terraform install script
 - Vm setup is the same as Terraform
 - install Jenkins script

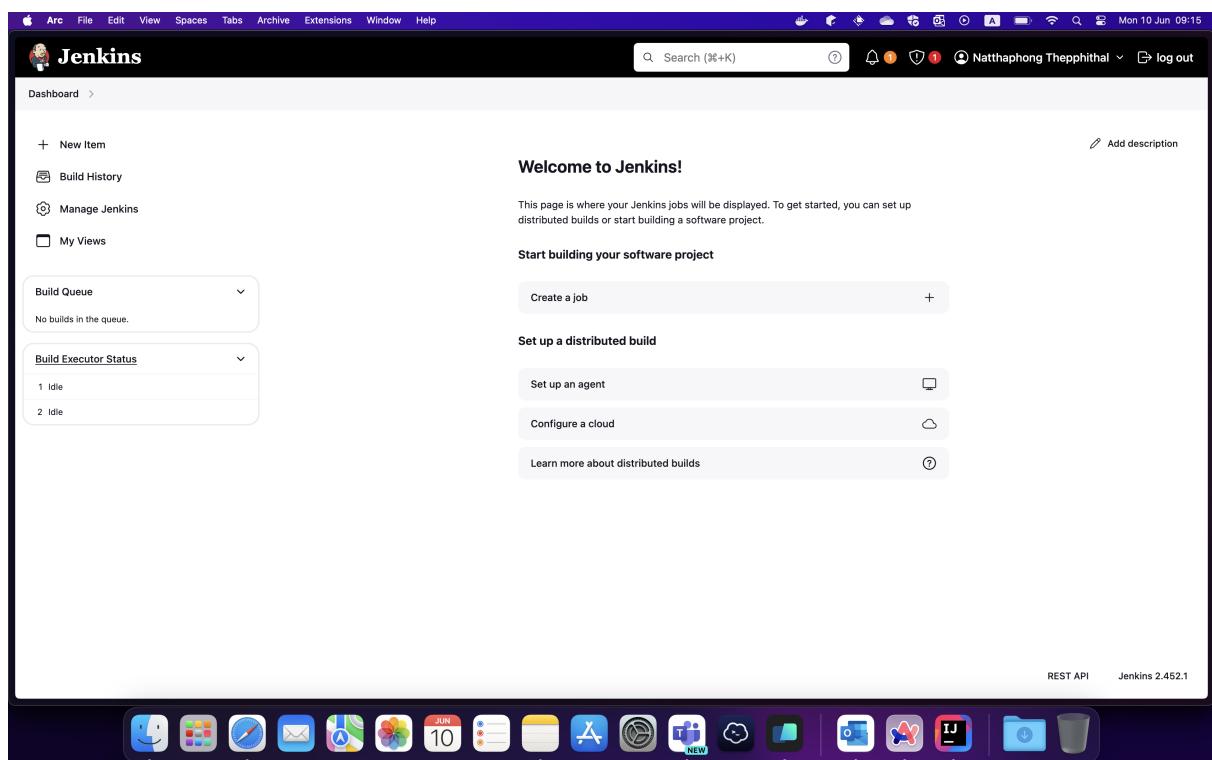
```
resource "null_resource" "install_jenkins" {  
    triggers = {  
        vm_id = azurerm_linux_virtual_machine.vm.id  
    }  
  
    connection {  
        type      = "ssh"  
        host      = azurerm_linux_virtual_machine.vm.p  
        user      = azurerm_linux_virtual_machine.vm.a  
        private_key = file("~/ssh/id_rsa")  
    }  
  
    provisioner "remote-exec" {  
        inline = [  
            "sudo apt-get update -y",  
            "sudo apt-get install -y openjdk-11-jdk",  
            "sudo wget -O /usr/share/keyrings/jenkins-keyr  
            https://pkg.jenkins.io/debian-stable/jenkins.i  
            "echo \"deb [signed-by=/usr/share/keyrings/jen  
            https://pkg.jenkins.io/debian-stable binary/\\"  
            sudo tee /etc/apt/sources.list.d/jenkins.list :  
            "sudo apt-get update -y",  
            "sudo apt-get install -y jenkins",  
            "sudo systemctl start jenkins",  
            "sudo systemctl enable jenkins"  
        ]  
    }  
}
```

enter ip-address:8080

- After that do following Jenkins instructions.

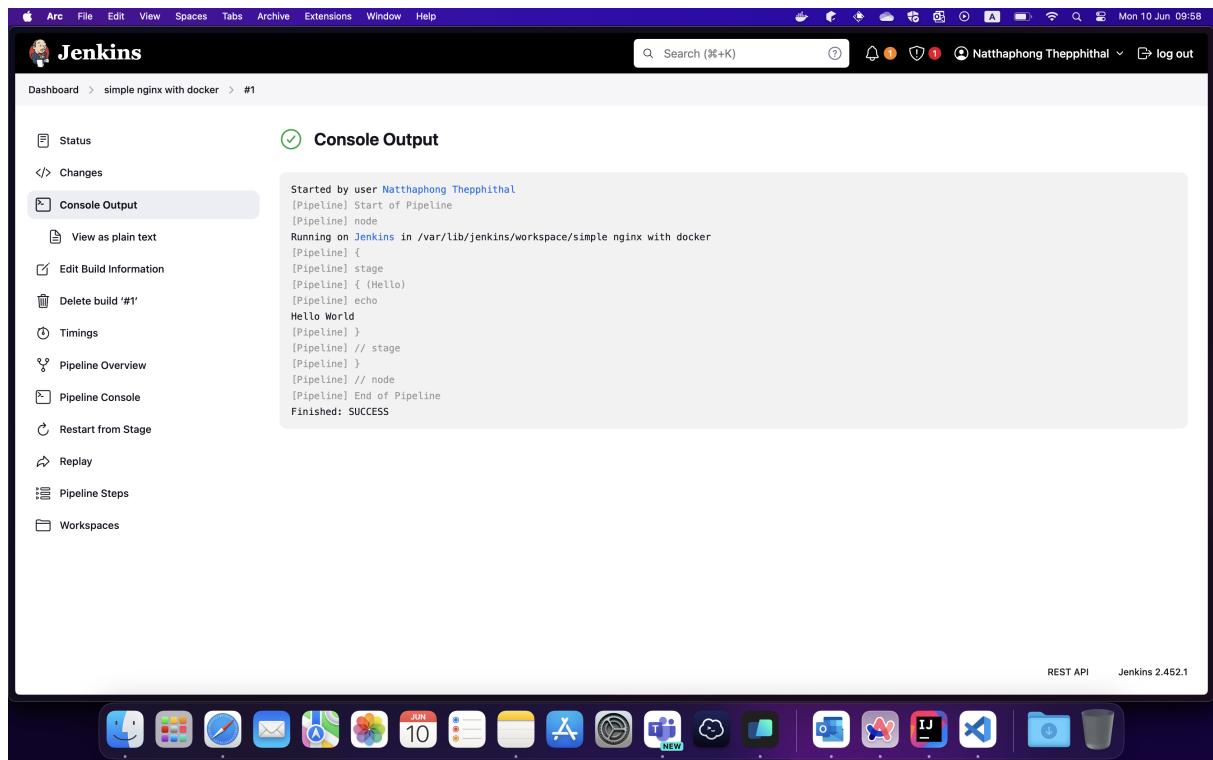


If done with initial setup will see :



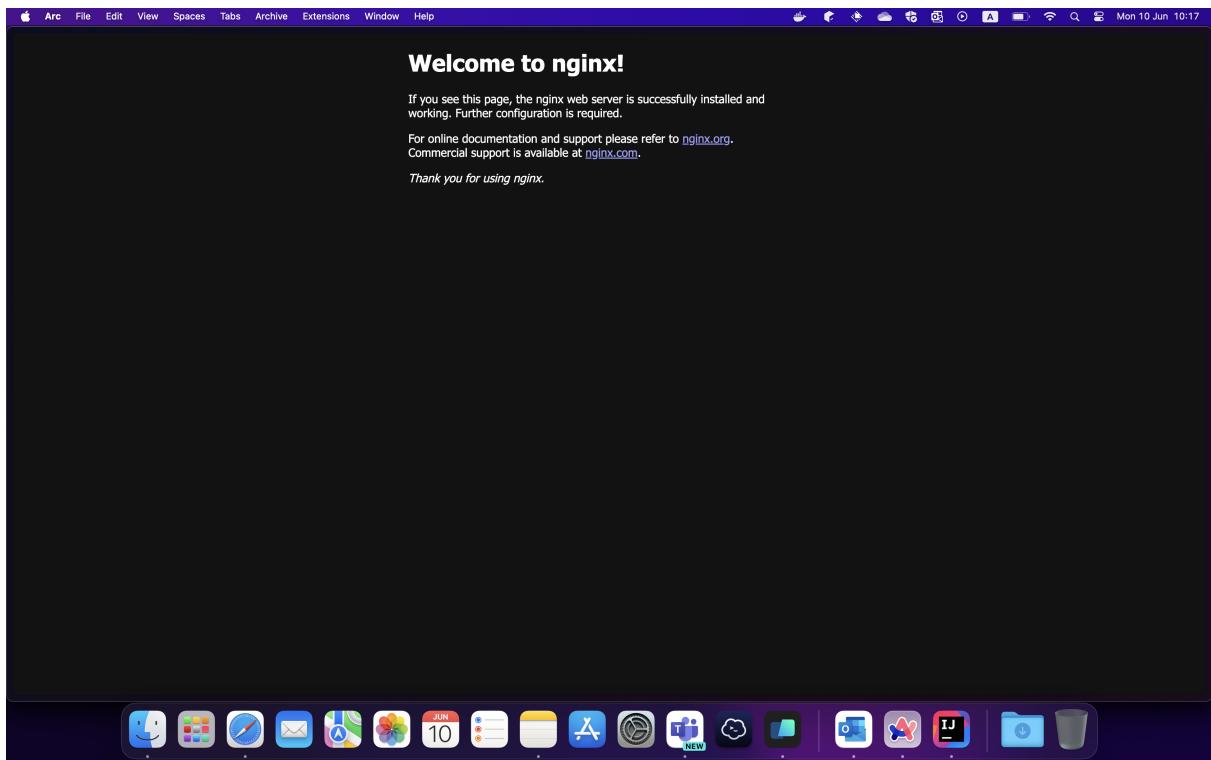
Create first simple groovy script for run Nginx on docker.

- Firstly, let's test more simple script to check if jenkins is working.



```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
    }
}
```



- now create let pull Nginx docker image and run it with Jenkins.

```
pipeline {  
    agent any  
  
    stages {  
        stage('Pull Docker Image') {  
            steps {  
                script {  
                    sh 'docker pull nginx:alpine'  
                }  
            }  
        }  
  
        stage('Remove Docker Container') {  
            steps {  
                script {  
                    sh 'docker rm -f mynginx'  
                }  
            }  
        }  
    }  
}
```

```

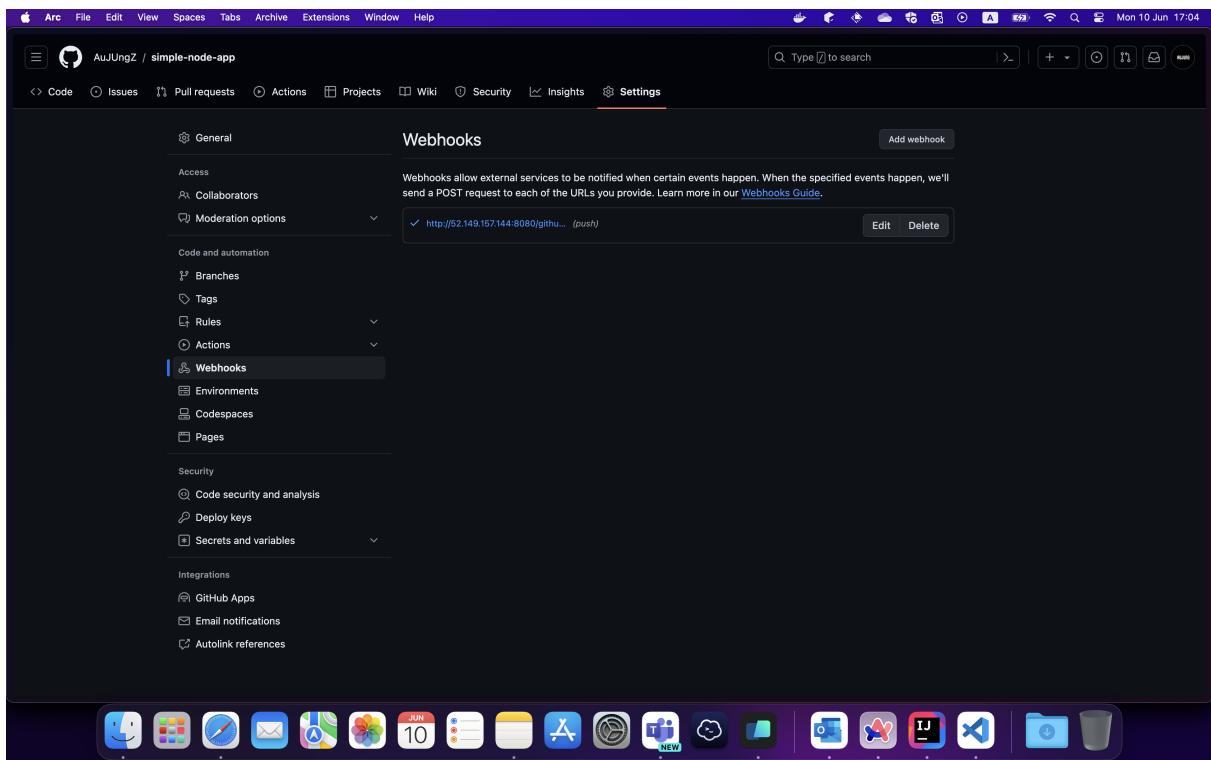
}

stage('Run Docker Container') {
    steps {
        script {
            sh 'docker run -d -p 80:80 --name mynginx'
        }
    }
}

```

Github web hook with Jenkins

- add Jenkins web hook for GitHub web hook add `/github-webhook` at the end of Jenkins url.



- Now when on git repository has new commit Jenkins pipeline will automatically redo again.

Set Slave agent

- master Node
 - install java jdk and install jenkins sercie.
 - generate ssh key for that master node via [ssh-keygen](#)
- Slave Node
 - install only **java jdk**.
 - place public ssh key of master agent on working user on slave node.

Master Node UI settings.

- Create global credential that store ssh private key of Master Node.
- Connect to slave node
 - add new node with ssh method with created credential