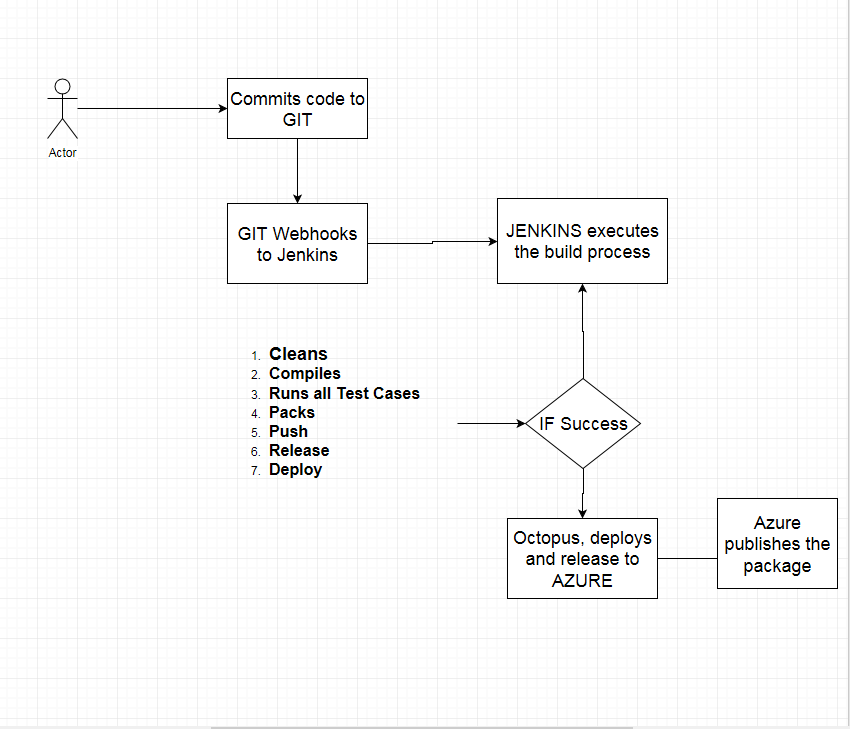
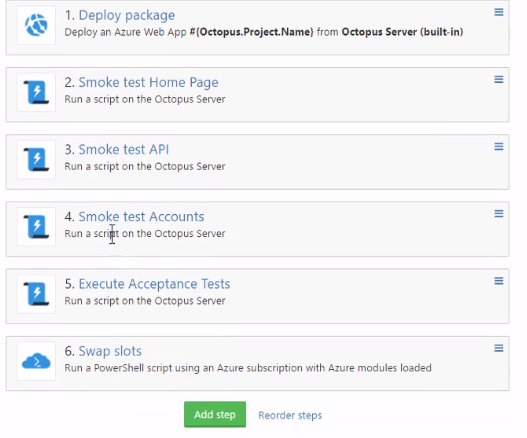
Flow Chart:





Continuous Delivery: Every change is proven to be deployable at any time

Continuous Deployment: Every change that passes the automated tests is deployed to production automatically.

## Octopus Deploy Lifecycles:

* Order or promotion
* Automate Deployments to Environments
* Retention Policies

## Octopus Deploy Advantages:

* for creating repeatable deployment process
* version a package management and reporting
* write quick scripts

Disadvantages:

* Reporting test results
* Source control scripts
* Not good for complex acceptance tests.

API Key

Scripts can access Octopus deploy and deploy services

Azure Service Principal

Octopus Deploy can deploy applications to Azure

QA Environment

Build User as a service account, so that never logon using API Keys

**API-JTBAGTU6EYLKLHA57BBPVMXFYQ**

Octopus Deploy and Azure

Azure

Application url <http://octopusdeploy-csitaram.net>

Deployment Steps

1. Create a Project
2. Add Step, Deploy An Azue Web App

Link up

1. Push the package to Octopus Deploy
2. Create Release
3. Deploy of the release.

Azure:

Blue Green Deployment

Two slots, and swap

1. Create a staging slot in Azure
2. Deploy to staging slot
3. Swap slots after deploy
4. Smoke Tests
5. Reusable build script
6. Acceptance Test

Jenkins:

Point to git server, and runs the build

Smoke test script

$environment = $OctopusParameters["Octopus.Environment.Name"]

$project = $OctopusParameters["Octopus.Project.Name"]

$uri = "http://$environment-$project-staging-azurewebsites.net"

$expectedStatusCode = 200

Write-Host "Making request for $url"

$response = Invoke-WebRequest -UseBasicParsing $url -MaximumRedirection 1

$statusCode = [int]$response.StatusCode

if($statusCode -ne $expectedStatusCode) {

throw "Smoke test failed for " + $url

}

Deploy staging -> Smoke Stage ->Swap SLots