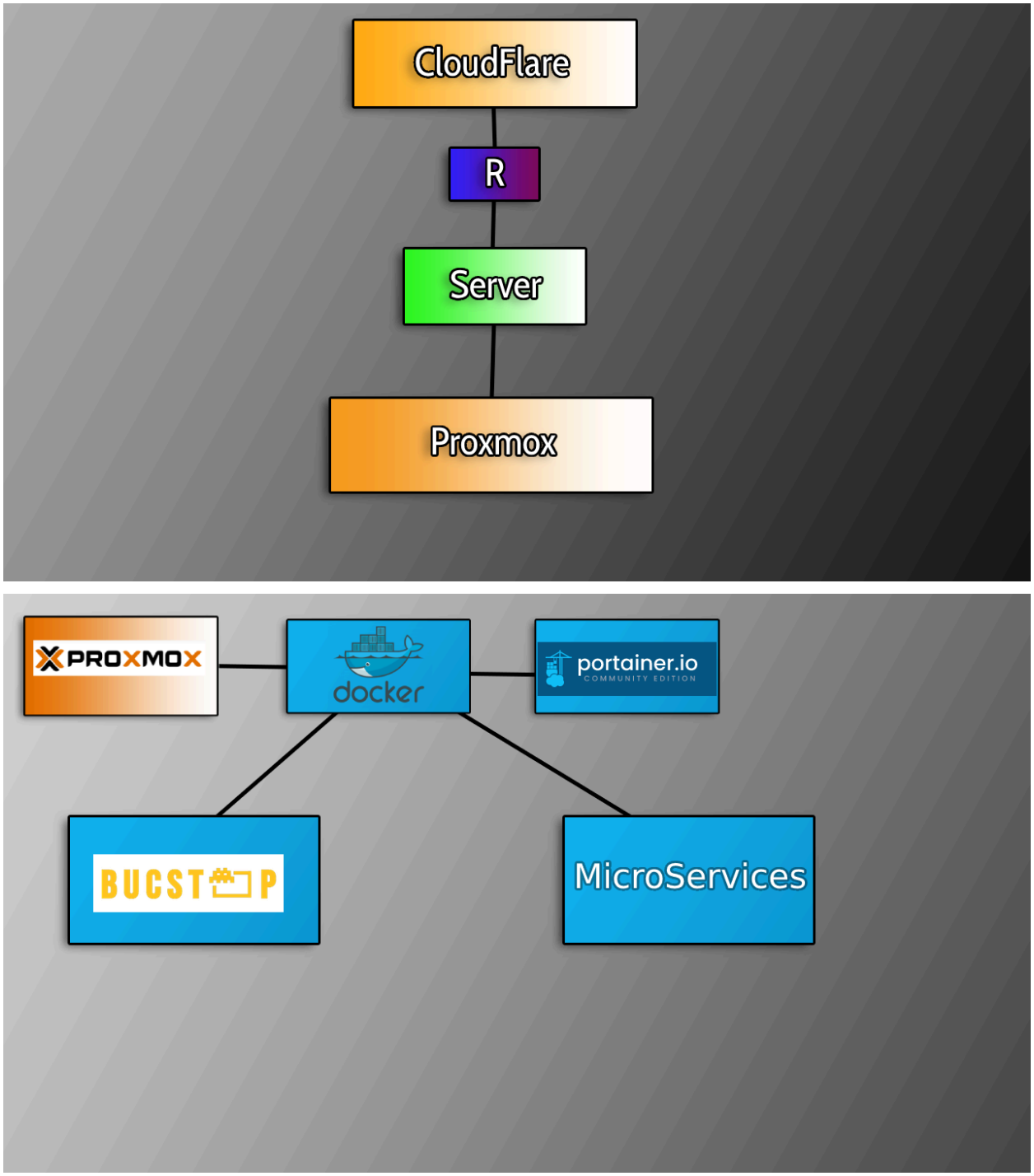
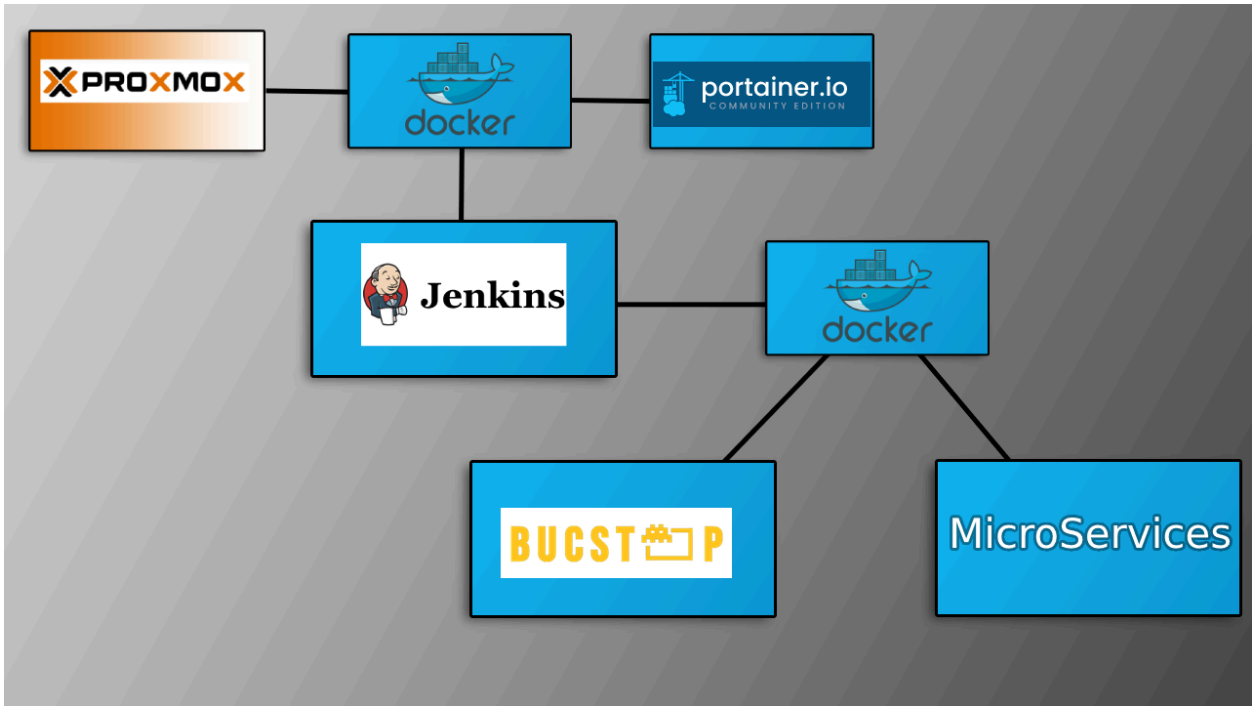


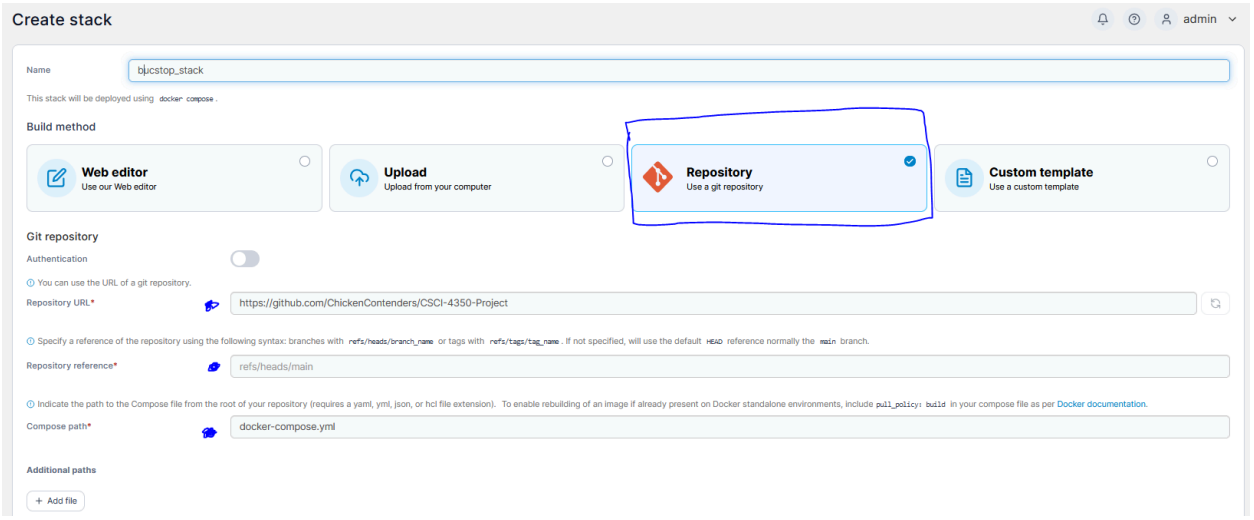
Current Architecture

We are currently using Proxmox as a Hypervisor to run Docker. Docker houses the Bucstop and microservice containers alongside the cloudflare containers to connect tunnels to the Domain name. Cloudflare handles the security to reach the server by using a private tunnel to connect to the private IP in my home network.

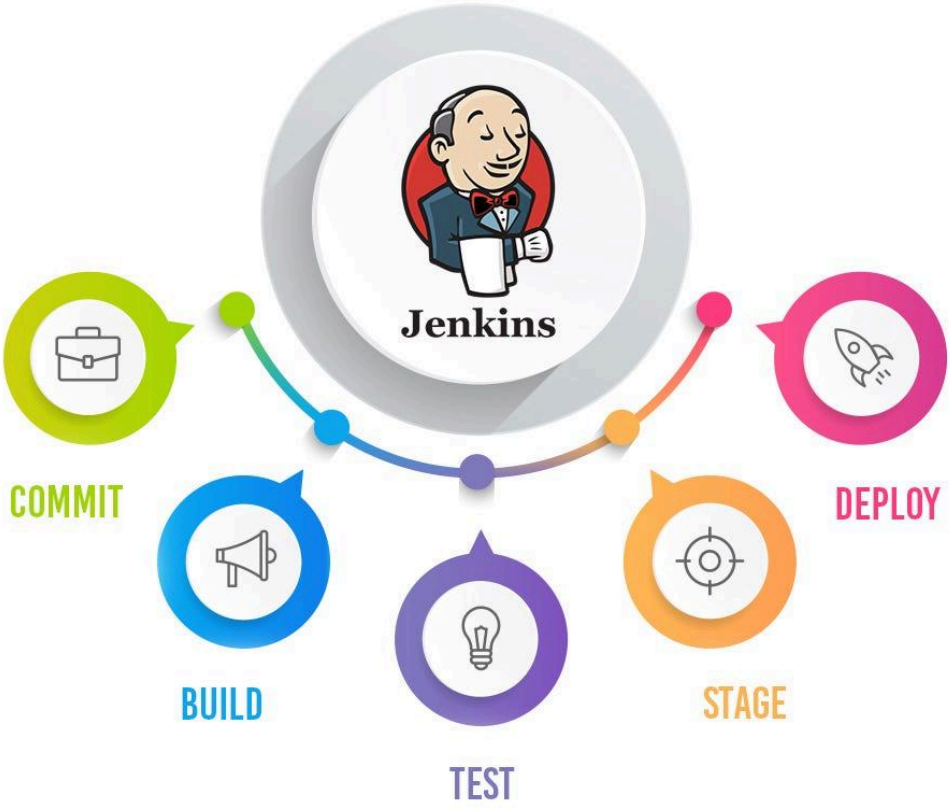


Potential Projected Architecture

Our current architecture runs on a Proxmox hypervisor and deploys containers directly with Docker. In the current architecture the only way we can automate deployment (server side) is by using portainer.



In portainer we can create a stack which will create multiple containers together using a docker compose file, and configuring webhooks. This solves the deployment issue when a new version of the github has been updated, but it creates ambiguous image names. This will make automation difficult and cause issues if we need to track bottlenecks in the pipeline. Additionally, this solution would contain several varieties of scripts to deploy correctly (example: the scripts folder in GitHub.) A solution to this is to swap the architecture to jenkins.



Scripted Pipeline Example

```
node {
    stage("Foo") {
        def data = new groovy.json.JsonSlurper().parseText(readFile('somefile.txt'))
        sh "make ${data.options}"
    }
    stage("Bar") {
        try {
            sh "make"
        } catch (err) {
            slackSend message: "Oh dude, didn't workout. ${err}"
            error "Things went wrong"
        }
    }
    if (env.BRANCH_NAME == 'master') {
        stage("Bar") {
            echo "Deploy!!"
        }
    }
}
```

By using Jenkins we can implement a true DevOps architecture. This will allow us to automate the process of deploying the application when we receive a GitHub push, add a variety of tests

(unit, regression, etc.), and building / updating images all into one file. This style would require a restructure and use jenkins as an overhead tool to then use a Docker Daemon inside to deploy.
TLDR- We would have to nest docker containers.