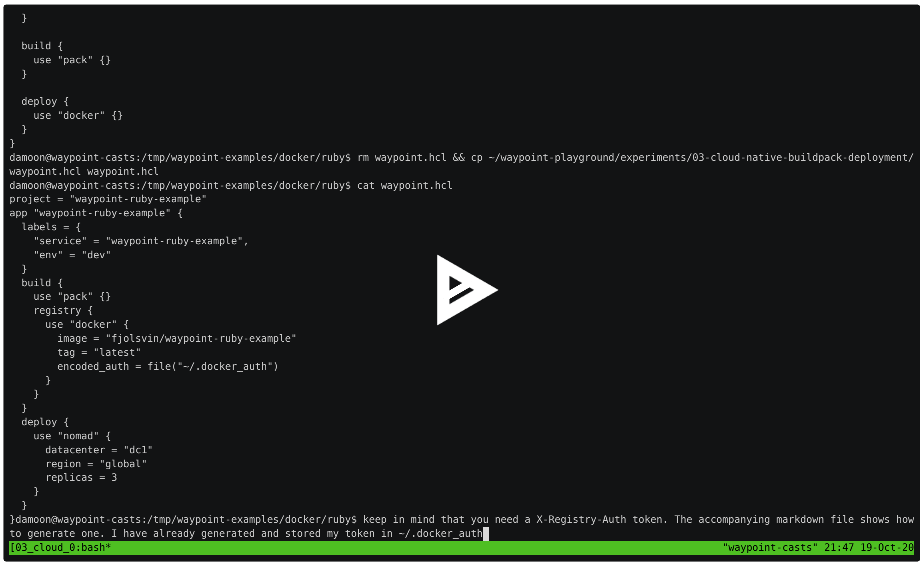
cloud native buildpack deployment



```
  }

  build {
    use "pack" {}
  }

  deploy {
    use "docker" {}
  }
}
damoon@waypoint-casts:/tmp/waypoint-examples/docker/ruby$ rm waypoint.hcl && cp ~/waypoint-playground/experiments/03-cloud-native-buildpack-deployment/
waypoint.hcl waypoint.hcl
damoon@waypoint-casts:/tmp/waypoint-examples/docker/ruby$ cat waypoint.hcl
project = "waypoint-ruby-example"
app "waypoint-ruby-example" {
  labels = {
    "service" = "waypoint-ruby-example",
    "env" = "dev"
  }
  build {
    use "pack" {}
    registry {
      use "docker" {
        image = "fjolsvin/waypoint-ruby-example"
        tag = "latest"
        encoded_auth = file("~/.docker_auth")
      }
    }
  }
  deploy {
    use "nomad" {
      datacenter = "dc1"
      region = "global"
      replicas = 3
    }
  }
}damoon@waypoint-casts:/tmp/waypoint-examples/docker/ruby$ keep in mind that you need a X-Registry-Auth token. The accompanying markdown file shows how
to generate one. I have already generated and stored my token in ~/.docker_auth
[03_cloud_0:bash*                                                                                              "waypoint-casts" 21:47 19-Oct-20
```

one way to build project for deploying to nomad with docker builder plugin is to use CloudNative Buildpacks through `pack` builder plugin . by default , nomad uses `heroku/buildpacks:18` image which is based on `ubuntu bionic`.

the main advantage of using cloudnative buildpacks is that these images have withstood the test of time and we know their build pipelines are stable. to customize builds, one must define a `procfile` and put it in repo's root.

for our example, we will use hashicorp's example ruby app for deployment which already has a procfile.

the only con that I can think of is that the created images are hardly minimal. as you would see in our example, the sample ruby app image is around `~850MB`.

to make sure the nomad client running the apllication container has access to image, you must use a remote docker image repository and push the created image in build step before deploy step. to have waypoint push the image to repository, you must use `encoded_auth` directive in `registry/docker` stanza which is the value stored in `X-Registry-Auth` header when pushing image to docker repostitory. for the sake of security, do not store raw value there and load it from a file. to generate the encoded auth token and store it in `~/.docker_auth` running the following snippet :

```
echo "{ \"username\": \"your-username\", \"password\": \"your-password\", \"email\": \"your-
```

you can now use the following snippet to clone the repo and build and deploy the image to your cluster

```
git clone https://github.com/hashicorp/waypoint-examples.git /tmp/waypoint-examples && \
pushd /tmp/waypoint-examples/docker/ruby && \
cat << EOF | sed -e '/^\s\s*$/d' -e "s/\(^.*$\)/echo '\1' | tee -a waypoint.hcl/g"
project = "waypoint-ruby-example"
app "waypoint-ruby-example" {
  labels = {
    "service" = "waypoint-ruby-example",
    "env" = "dev"
  }
  build {
    use "pack" {}
    registry {
      use "docker" {
        image = "fjolsvin/waypoint-ruby-example"
        tag = "latest"
        encoded_auth = file("~/.docker_auth")
      }
    }
  }
  deploy {
    use "nomad" {
      datacenter = "dc1"
      region = "global"
      replicas = 3
    }
  }
}
EOF
waypoint init && \
NOMAD_ADDR="http://10.33.235.43:4646" waypoint up && \
popd
```

reference

- docker plugin
- cloud native plugin
- nomad plugin
- exec plugin
- procfile format