Scaling Jenkins on Azure

it’s basically clouds all the way down
hey
R Tyler Croy

- github.com/rtyler
- twitter.com/agentdero
- Jenkins board member, infra lead
- “Community Concierge”
- send gifs to tyler@cloudbees.com
Running Jenkins in the Cloud™
Jenkins ♥ Docker
Containerized master

docker pull jenkins
docker run \
  -p 8080:8080 \
  -v `pwd`/jenkins:/var/jenkins_home \
jenkins
Containerized master: Pros

➢ Requires Linux master node
➢ Easy to manage/update
  ○ LTS release updates by the Jenkins project
➢ Portability
  ○ Pack up your JENKINS_HOME and move to a new box
Containerized master: Cons

➢ Requires Linux master node
➢ I/O performance concerns
  ○ Mapping JENKINS_HOME through to container
➢ CPU scheduling
  ○ “Noisy neighbor” problem on Docker daemon
Containerized Build Nodes

➢ more to come later..
Running Jenkins in the Cloud™
Azure plugin
### General Configuration

**Name**: JenkinsUbuntuLTS

**Description**: A Trusty Tazi Ubuntu

**Labels**: ubuntu docker

**Region**: Central US

**Virtual Machine Size**: Standard_D1

**Storage Account Name**: Standard_DL

**Retention Time (in minutes)**: 60

**Shutdown Only (Do Not Delete) After Retention Time**

**Usage**: Utilize this node as much as possible

### Image Configuration

**Image Family or Id**: b39f27a84e6d92bb05ca5ca80ebadd85_Ubuntu-14.04-LTS-amd64-conor-20160217.1-en-us-30GB

**Launch Method**: SSH

**Init Script**:

```
sudo apt-get update -y && sudo apt-get install -y default-jdk docker.io && sudo usermod -append -g docker jenkins
```

**Username**: jenkins

**Password**: ********
This build is parameterized
Throttle builds
Disable Build (No new builds will be executed until the project is re-enabled.)
Execute concurrent builds if necessary
Restrict where this project can be run
Label Expression

Advanced Project Options

Source Code Management
Docker plugin
Containerized Build Nodes

➢ Neat!
➢ Point it at:
  ○ local Docker daemon on the Jenkins master
  ○ a remote Docker daemon
  ○ a Docker Swarm endpoint
➢ Docker! DOCKER! OMG DOCKER!
Containerized Build Nodes: Pros

➢ Jenkins administrator governs images used
➢ Easy creation/management/deployment of new build environments
➢ Portability across computing environments
  ○ Run it anywhere you want! As long as
    ■ it’s Linux
    ■ with a recent kernel
Containerized Build Nodes: Cons

➢ Jenkins administrator governs images used
➢ Docker-in-Docker is a failwhale
Running Jenkins in the Azure™
Azure plugin

➢ Dynamically provision Linux machines (or Windows)
➢ Minimum of 30 minutes “Retention Time”
➢ Use specific Labels
  ○ “Standard_D1”, “linux” : Bad
  ○ “ubuntu”, “docker”, “rhel”, “highram”, “highcpu” : Good
➢ Keep “Init Script” definitions small
Docker

- Define Dockerfiles for build/testing environments
- Enable different teams to use different images
- Define pipelines for those Docker images
Pipeline plugin

➢ Define your delivery pipeline in one place
➢ Check a Jenkinsfile directly into SCM
Pipeline plugin

```java
node('docker') {
    checkout scm

    /* Using this hack to grab the appropriate abbreviated SHA1 of *
     * our build's commit. Currently I cannot refer to `env.GIT_COMMIT` *
    */
    sh 'git rev-parse HEAD > GIT_COMMIT'
    def shortCommit = readFile('GIT_COMMIT').take(6)

    stage 'Build'
    def image = docker.build("jenkinsciinfra/bind:build-$shortCommit")

    stage 'Deploy'
    image.push()
}
```
Pipeline plugin

Pipeline Hello World

Stage View

<table>
<thead>
<tr>
<th></th>
<th>Build</th>
<th>Unit Test</th>
<th>Acceptance Test</th>
<th>Deploy to Staging</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>102ms</td>
<td>81ms</td>
<td>30s</td>
<td>40ms</td>
</tr>
<tr>
<td>Feb 24, 19:23</td>
<td>No Changes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb 24, 19:23</td>
<td>No Changes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average stage times:
(Average full run time: ~30s)
Scary Demo Time

can this better work
neat plugins shown

- Pipeline
- Azure
- CloudBees Folders
- GitHub
- Timestamper
- NodeJS
- Pipeline Stage View
Other Scaley Things

that aren’t lizards
Scaling Masters is Hard™

➢ Jenkins will be better at this in the future
➢ Partition masters along pipeline boundaries
  ○ “Dev Jenkins” “Ops Jenkins” : Bad
  ○ “Middleware Jenkins” “Mobile Apps Jenkins” : Good
➢ Buy the most memory and fastest disks possible
➢ Offload as much as possible to build nodes
Pay the bills
questions

jenkins-ci.org
@jenkinsci
github.com/jenkinsci