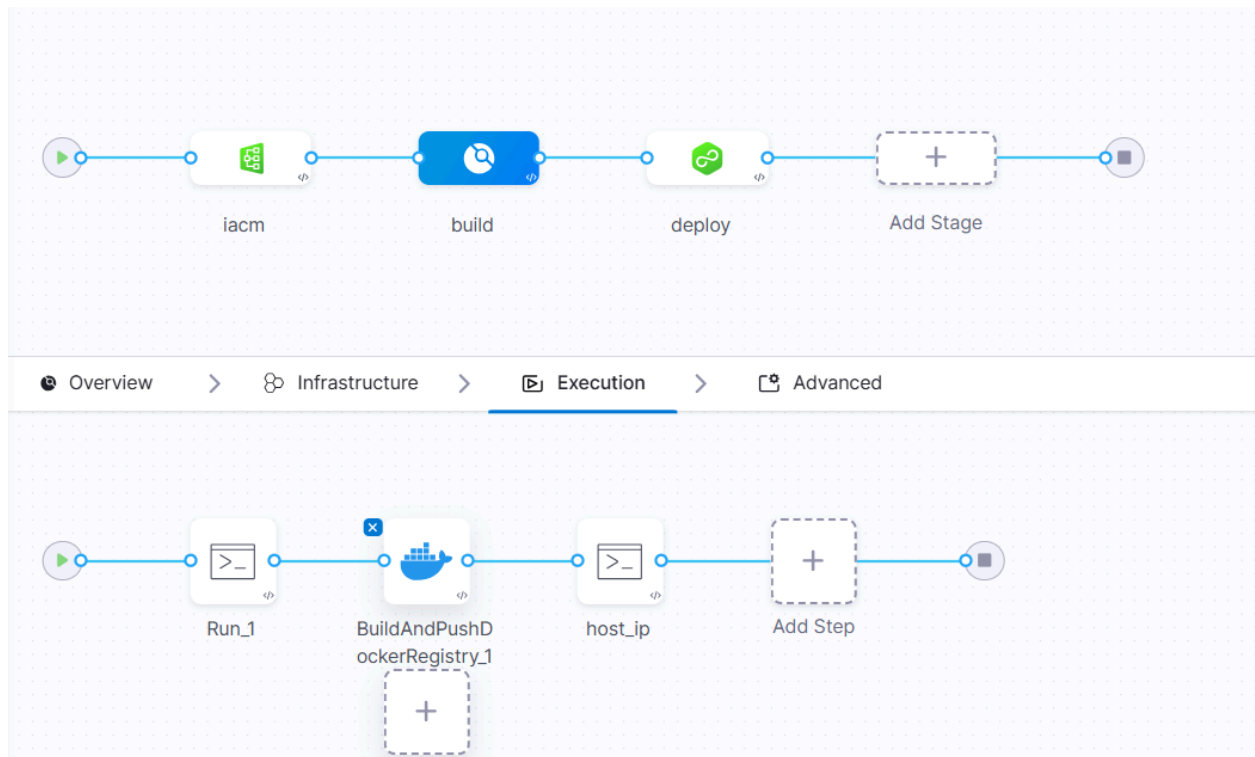


Step 1: iacm



Stage: build

Step-1:

Run ⓘ

Apply Changes Discard

Step Parameters Advanced

Save as Template

Name ⓘ Id ⓘ : Run_1 Enhance [AI]

Run_1

Description

Shell Sh

Command ⓘ

```
1 ls -la
2
3 mvn clean package
```

Optional Configuration ▾

Step 2:

Build and Push an image to D... ⓘ

Apply Changes Discard

Step Parameters Advanced

Save as Template

Name ⓘ Id ⓘ : BuildAndPus... Enhance [AI]

BuildAndPushDockerRegistry_1

Registry Type

Artifact Registry

Third-Party Artifact Registry

Docker Connector ⓘ

dockerhub-ganesh PROJECT

Docker Repository ⓘ

ganesh6498/disk-partition

Tags ⓘ

latest

+ Add

☐ Enable Docker Layer caching ⚡ ⓘ

Step 3:

Command

```
1 ip="_ansible"
2 echo ${ip}
3 ls -l /root
4 # Use with Ansible
5 #ansible-playbook -e image_name="${DOCKER_IMAGE}:${TAG}" ansible.yaml
6
7 ansible-inventory --list > hosts
8
9
10 jq -r ".[\"${ip}\"] .hosts[]" hosts > vault_ips.txt
11
12 cat vault_ips.txt
13
14 host=$(paste -sd' ' vault_ips.txt)
15 echo $host
```

```
ip="_ansible"
```

```
echo ${ip}
```

```
ls -l /root
```

```
# Use with Ansible
```

```
#ansible-playbook -e image_name="${DOCKER_IMAGE}:${TAG}" ansible.yaml
```

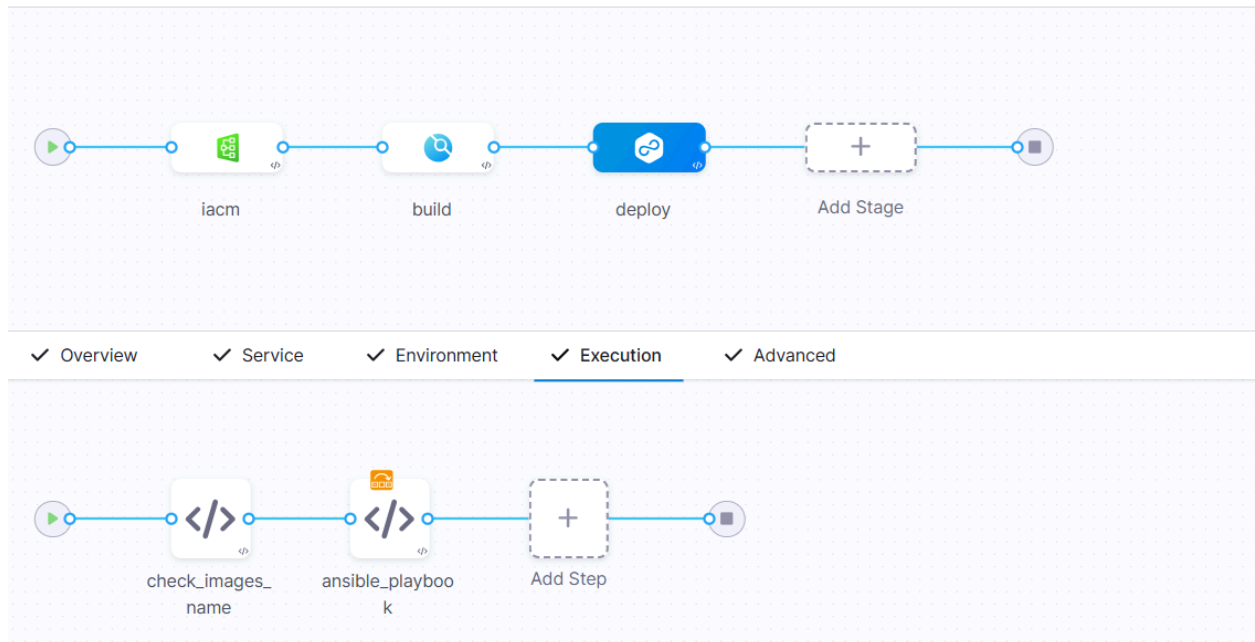
```
ansible-inventory --list > hosts
```

```
jq -r ".[\"${ip}\"] .hosts[]" hosts > vault_ips.txt
```

```
cat vault_ips.txt
```

```
host=$(paste -sd' ' vault_ips.txt)
```

```
echo $host
```



Stage 3

Step 2:

```
# -----
# Step 1: Variables from Harness
# -----
DOCKER_IMAGE="<+pipeline.stages.build.spec.execution.steps.BuildAndPushDockerRegistry_1.artifact_BuildAndPushDockerRegistry_1.stepArtifacts.publishedImageArtifacts[0].imageName>"
TAG="<+pipeline.stages.build.spec.execution.steps.BuildAndPushDockerRegistry_1.artifact_BuildAndPushDockerRegistry_1.stepArtifacts.publishedImageArtifacts[0].tag>"
HOSTS="<+pipeline.stages.build.spec.execution.steps.host_ip.output.outputVariables.host>"

echo "Image = $DOCKER_IMAGE"
echo "Tag    = $TAG"

# -----
# Step 2: Write private key
```

```
# -----  
echo '<+secrets.getValue("private-key-vault-ganesh")>' > /tmp/private_key  
chmod 600 /tmp/private_key  
username='<+secrets.getValue("oslogin-username-ganesh")>'  
echo "Using SSH username: ${username}"
```

```
# -----  
# Step 3: Setup Ansible inventory  
# -----  
mkdir -p /tmp/ansible  
echo "[vault]" > /tmp/ansible/hosts  
for ip in $HOSTS; do  
    echo "$ip" >> /tmp/ansible/hosts  
done
```

```
cat /tmp/ansible/hosts
```

```
# -----  
# Step 4: Setup Ansible config  
# -----  
  
cat <<EOF > /tmp/ansible/ansible.cfg  
[defaults]  
inventory = /tmp/ansible/hosts  
host_key_checking = False  
retry_files_enabled = False  
remote_user = ${username}  
private_key_file = /tmp/private_key  
EOF
```

```
export ANSIBLE_CONFIG=/tmp/ansible/ansible.cfg
```

```
cat /tmp/ansible/ansible.cfg
```

```
# -----
```

```
# Step 5: Test connection
```

```
# -----
```

```
ansible-inventory --graph
```

```
ansible all -m ping
```

```
# -----
```

```
# Step 6: Clone repo & run playbook
```

```
# -----
```

```
cd /tmp
```

```
rm -rf disk-partition-vms-ansible-harness
```

```
git clone
```

```
https://github.com/ganesh-redy/disk-partition-vms-ansible-harness.git
```

```
cd disk-partition-vms-ansible-harness
```

```
ls -la
```

```
ansible --version
```

```
# -----
```

```
# Step 7: Run playbook
```

```
# -----
```

```
ansible-playbook -e image_name="${DOCKER_IMAGE}:${TAG}" playbook.yaml
```