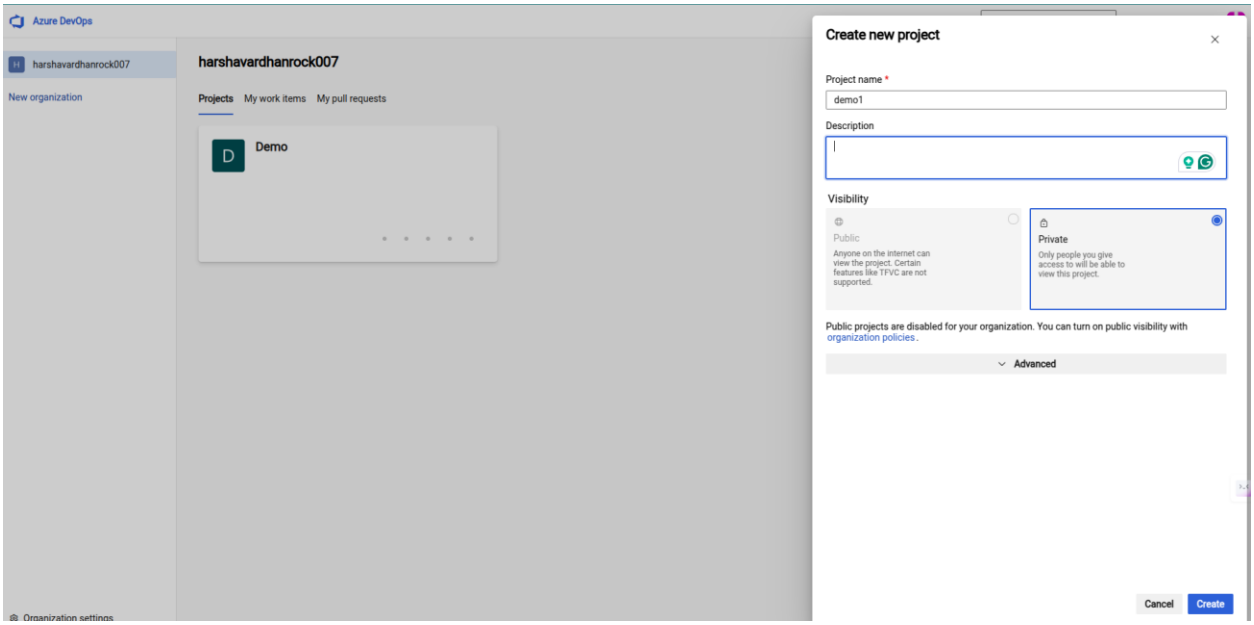
🚀 Azure DevOps Project with Kubernetes & ArgoCD Integration

1. Azure Account Initialization

- Set up a valid **Azure Account** with an active subscription.
- Log into Azure DevOps using the **same credentials** as your Azure portal.

2. Create a New Azure DevOps Project

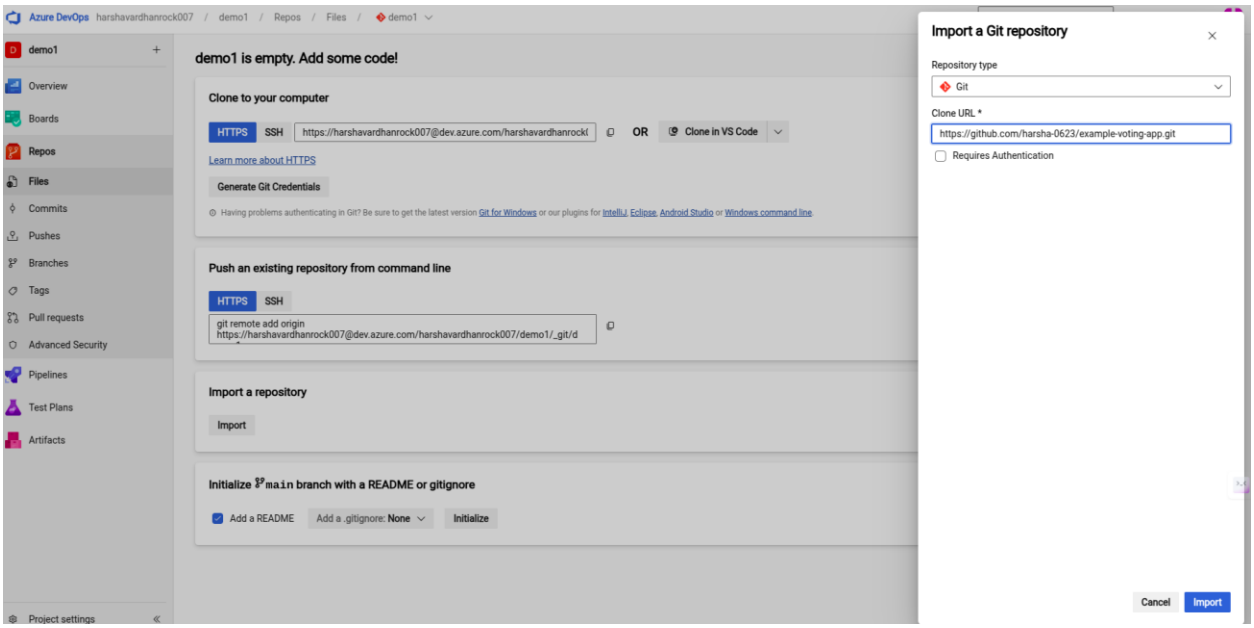
- Navigate to Azure DevOps Dashboard → Click on **New Project**.
- Name the project appropriately based on your objectives.



3. GitHub Repo Migration to Azure Repos

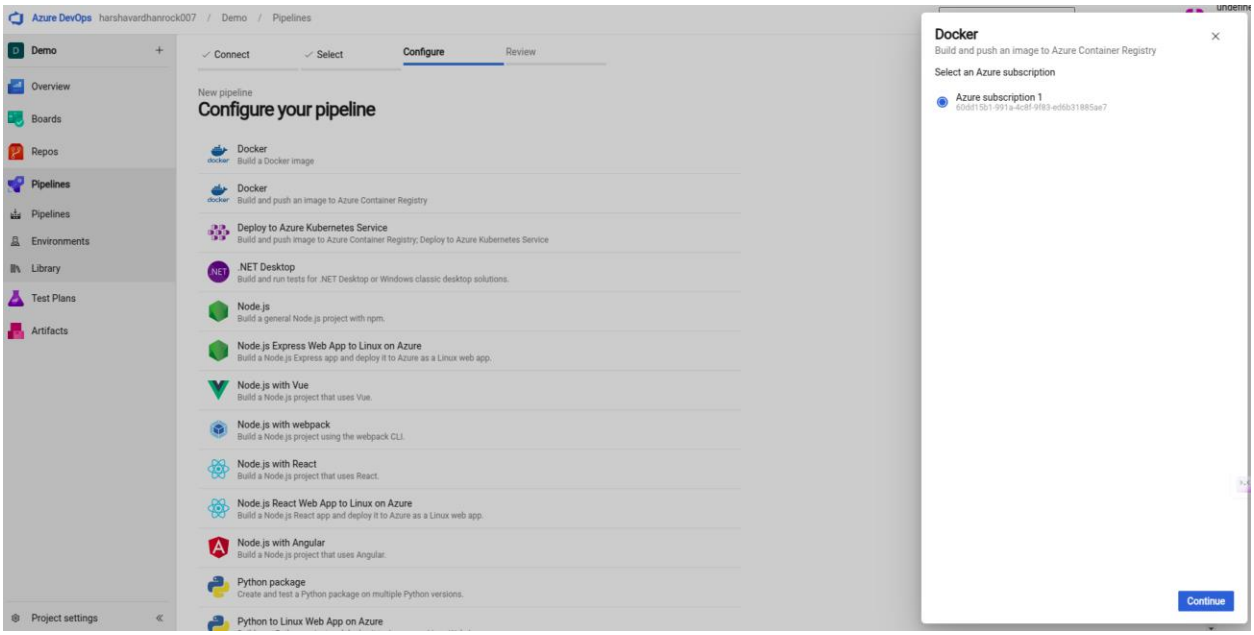
- Go to **Repos** → **Import Repository**.

- Paste your **GitHub HTTPS URL** to import all code into Azure Repos.



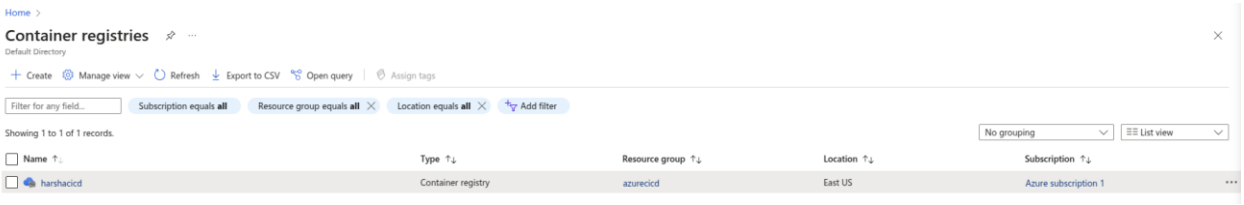
🔧 4. Build & Push Pipeline Configuration

1. Navigate to **Pipelines** → **New Pipeline**.
2. Select **Azure Repos Git** as the source.
3. Choose your required repository.
4. Under **Configure Pipeline**, select:
    a. **Docker - Build and push an image to Azure Container Registry**.
5. Choose the **correct Azure subscription**.



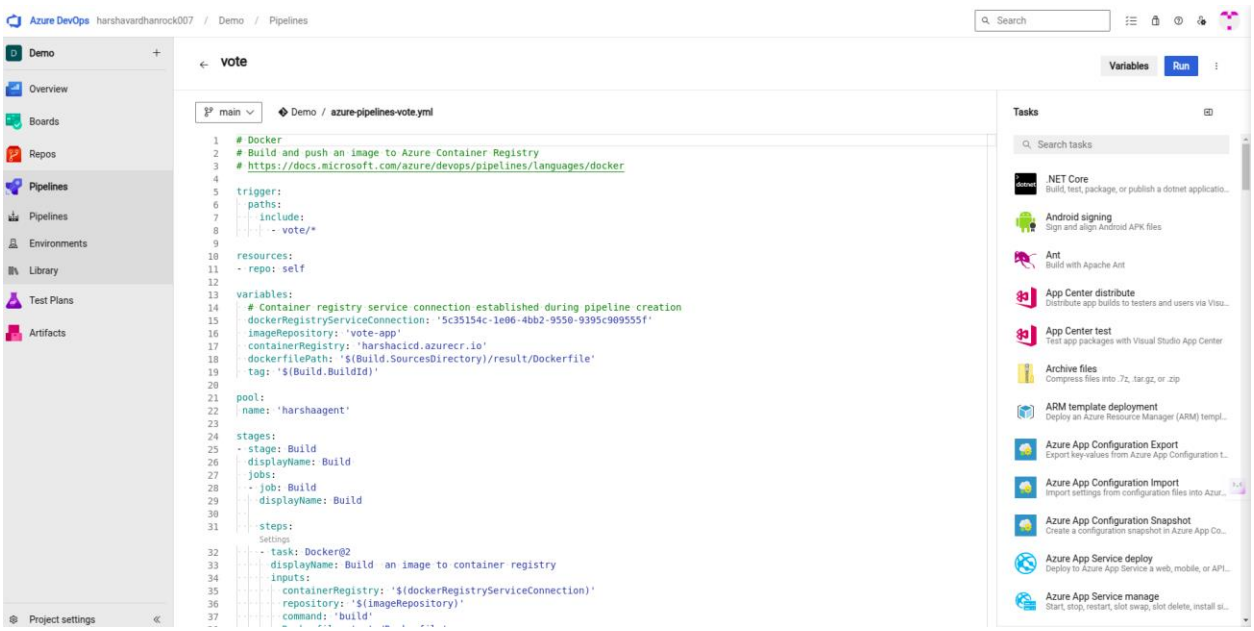
5. Create Azure Container Registry (ACR)

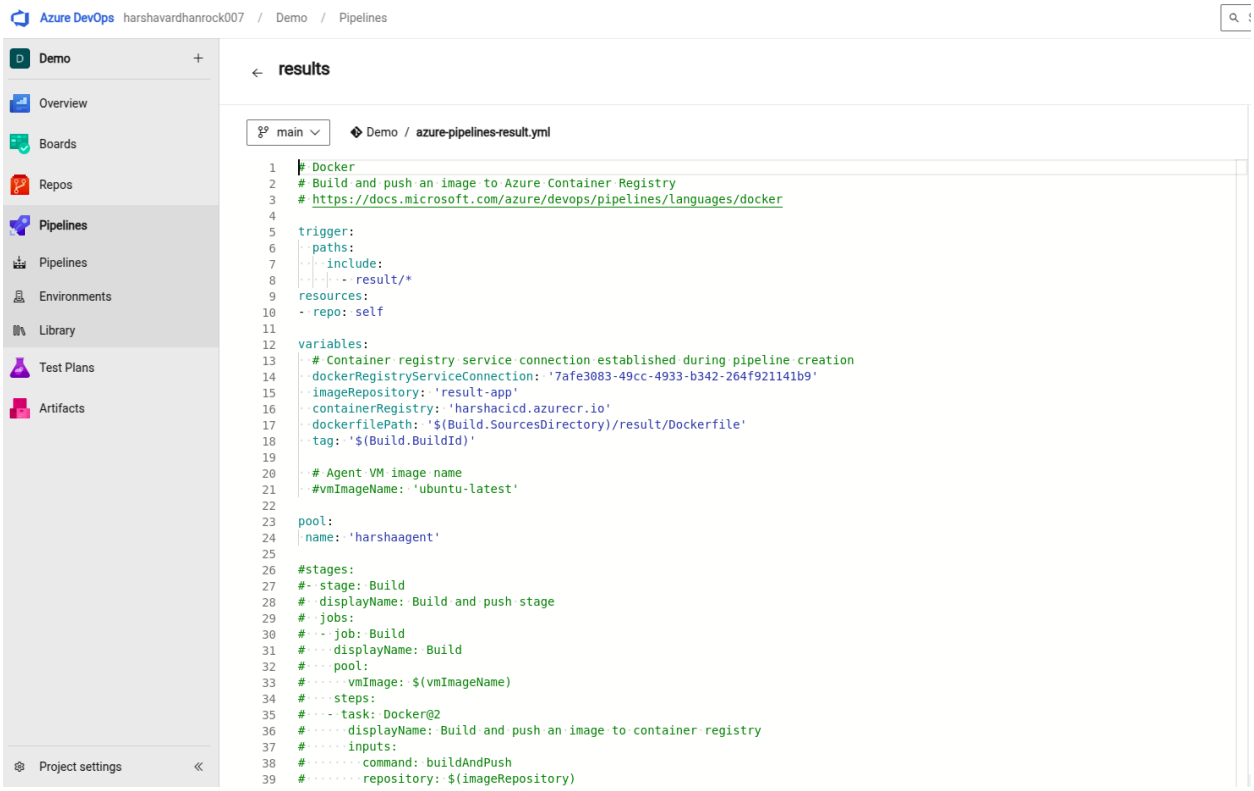
- Go to Azure Portal → Search **Container Registry** → Create a new instance.
- Group all project services in a **standard Resource Group** for manageability.



6. YAML Pipeline Customization

- Modify the generated YAML to:
  - Replace default VM image with a **custom agent pool**.
  - Split the **build** and **push** stages.
  - Create separate pipelines for **worker**, **results**, and **vote** services.

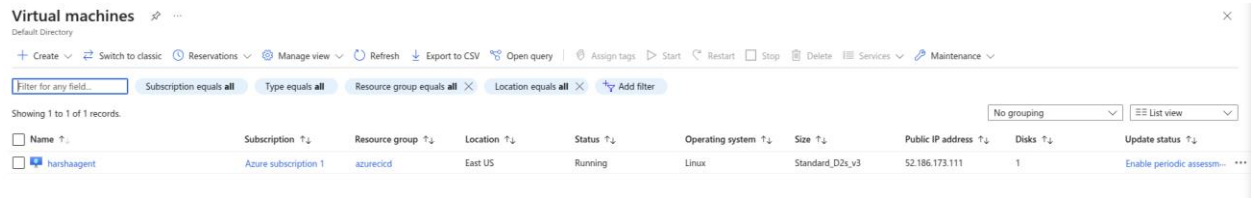




7. Setup Virtual Machine for Self-hosted Agent

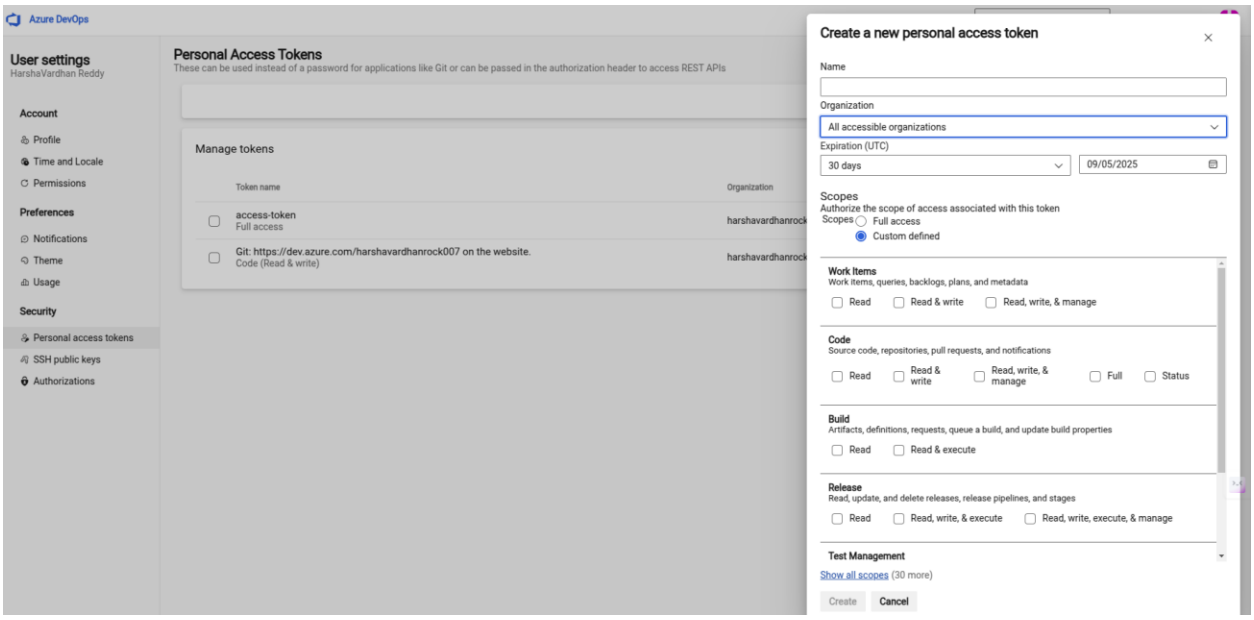
- Create a VM and install:
  - Docker
  - Kubernetes CLI (kubectl)

- Go to **Project Settings > Agent Pools** → Create a **New Pool** (mark as **Self-hosted**).



8. Generate Personal Access Token (PAT)

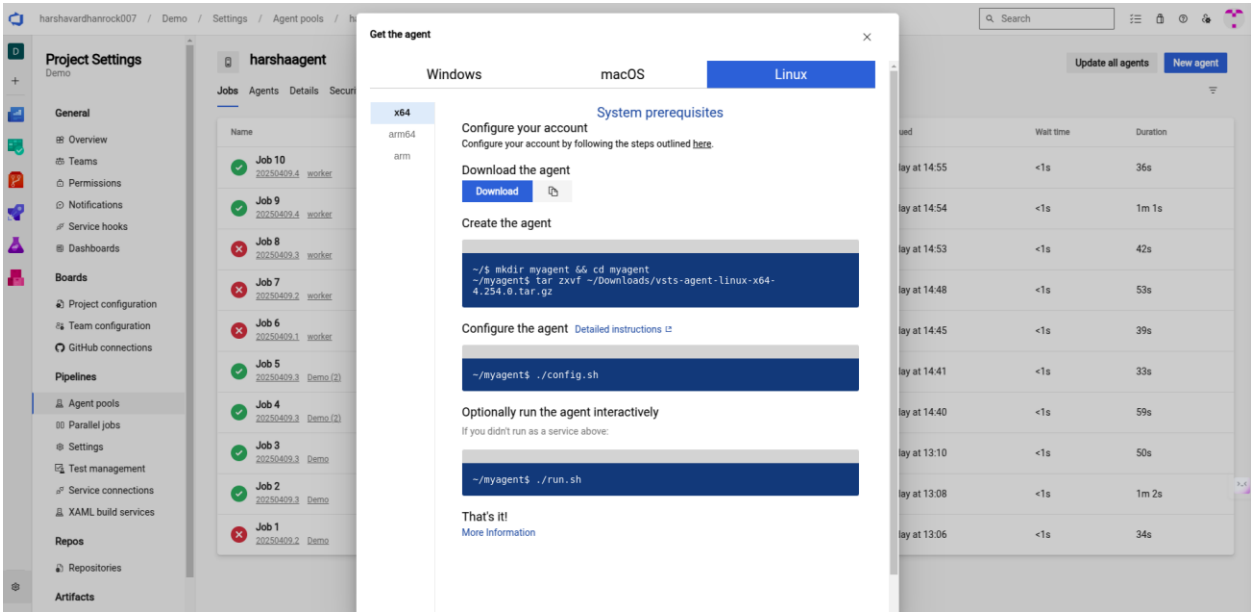
- Go to **User Settings > Personal Access Tokens** → Generate a new token with relevant scopes.



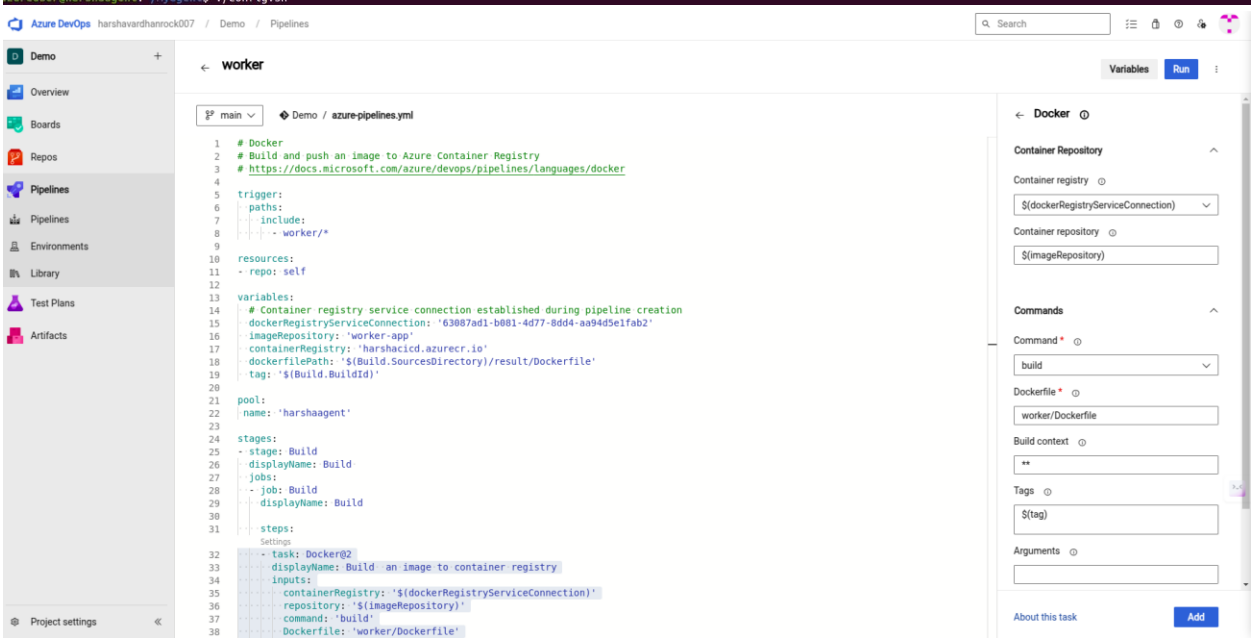
🔗 9. Connect VM to Azure DevOps as Agent

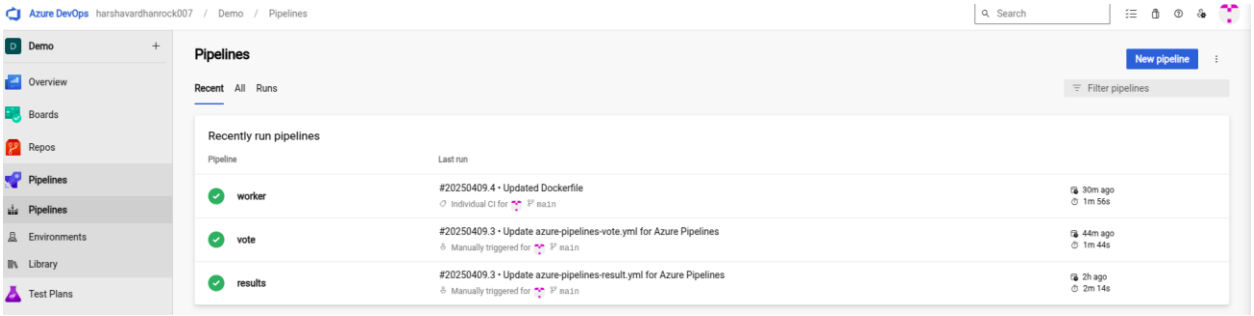
1. From **Agent Pools**, click on **New Agent**.
2. Follow the commands provided:

./config.sh # Enter server URL and PAT
./run.sh # Run agent in active state



```
Enter personal access token >
Enter personal access token > Exiting...
azureuser@harshaagent:~/myagent$ ./config.sh

  ___                      ______ _            _ _
 / _ \                     | ___ (_)          | (_)
/ /_\ \_____   _ _ __ ___  | |_/ /_ _ __   ___| |_ _ __   ___  ___
|  _  |_  / | | | '__/ _ \ |  __/| | '_ \ / _ \ | | '_ \ / _ \/ __|
| | | |/ /| |_| | | |  __/ | |   | | |_) |  __/ | | | | |  __/\__ \
\_| |_/___|\__,_|_|  \___| \_|   |_| .__/ \___|_|_|_| |_|\___||___/
                                    | |
       agent v4.253.0               |_|         (commit 5263f72)


>> End User License Agreements:

Building sources from a TFVC repository requires accepting the Team Explorer Everywhere End User License Agreement. This step is not required for building sources from Git repositories.

A copy of the Team Explorer Everywhere license agreement can be found at:
  /home/azureuser/myagent/license.html

Enter (Y/N) Accept the Team Explorer Everywhere license agreement now? (press enter for N) >

>> Connect:

Enter server URL > https://dev.azure.com/harshavardhanrock007
Enter authentication type (press enter for PAT) > 7IlBqZMe4gyq6DVF2SCVxqTTZyMdFPEOXv2IAgKDMmzUsyB7mjXNJQQJ99BDACAAAAAAAAAAAAASAZDO1u0V
Enter a valid value for authentication type.
Enter authentication type (press enter for PAT) > 7IlBqZMe4gyq6DVF2SCVxqTTZyMdFPEOXv2IAgKDMmzUsyB7mjXNJQQJ99BDACAAAAAAAAAAAAASAZDO1u0V
Enter a valid value for authentication type.
Enter authentication type (press enter for PAT) >
Enter personal access token > **********************************************************************************
Connecting to server ...

>> Register Agent:

Enter agent pool (press enter for default) > harshaagent
Enter agent name (press enter for harshaagent) >
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
Enter work folder (press enter for _work) >
2025-04-09 07:32:19Z: Settings Saved.
azureuser@harshaagent:~/myagent$ ./config.sh
```





10. Create Kubernetes Cluster (AKS)

- In Azure Portal → Search **Kubernetes services** → Create a new AKS cluster.
- On the VM:

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
az login
az aks get-credentials --name azuredevops --resource-group azurecicd --overwrite-existing
sudo snap install kubectl --classic
kubectl get pods

11. Install & Configure ArgoCD

1. Check ArgoCD pods:

kubectl get pods -n argocd

2. Retrieve & decode initial admin password:

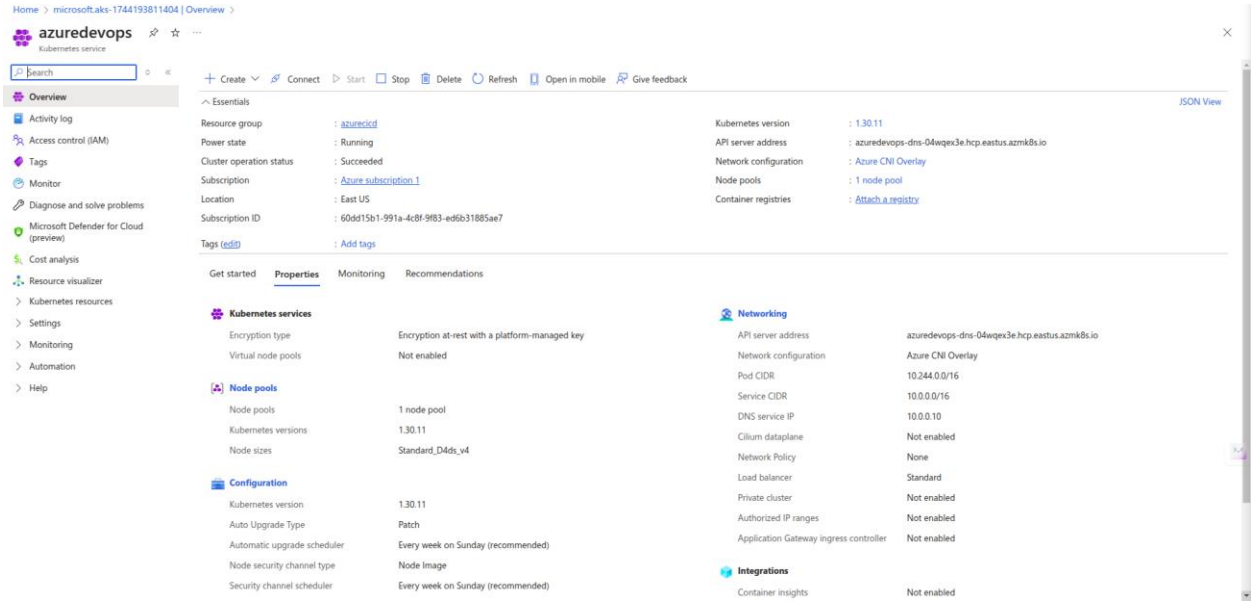
kubectl get secrets -n argocd
kubectl edit secret argocd-initial-admin-secret -n argocd
echo <base64-password> | base64 --decode

3. Edit NodePort:

kubectl edit svc argocd-server -n argocd
kubectl get nodes -o wide

4. Adjust config (like session timeout):

kubectl edit cm argocd-cm -n argocd




```
azureuser@harshaagent:~$ az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code EH84HRQSM to authenticate.

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No     Subscription name     Subscription ID                       Tenant
-----  --------------------  ------------------------------------  -----------------
[1] *  Azure subscription 1  60dd15b1-991a-4c8f-9f83-ed6b31885ae7  Default Directory

The default is marked with an *; the default tenant is 'Default Directory' and subscription is 'Azure subscription 1' (60dd15b1-991a-4c8f-9f83-ed6b31885ae7).

Select a subscription and tenant (Type a number or Enter for no changes):

Tenant: Default Directory
Subscription: Azure subscription 1 (60dd15b1-991a-4c8f-9f83-ed6b31885ae7)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.

azureuser@harshaagent:~$ az aks get-credentials --name azuredevops --resource-group azurecicd --overwrite-existing
Merged "azuredevops" as current context in /home/azureuser/.kube/config
azureuser@harshaagent:~$ kubectl get pods
Command 'kubectl' not found, but can be installed with:
sudo snap install kubectl
azureuser@harshaagent:~$ sudo snap install kubectl --classic
2025-04-09T10:39:42Z INFO Waiting for automatic snapd restart...
kubectl 1.32.3 from Canonical✓ installed
azureuser@harshaagent:~$ kubectl get pods
No resources found in default namespace.
```

12. Create Docker Secret for K8s

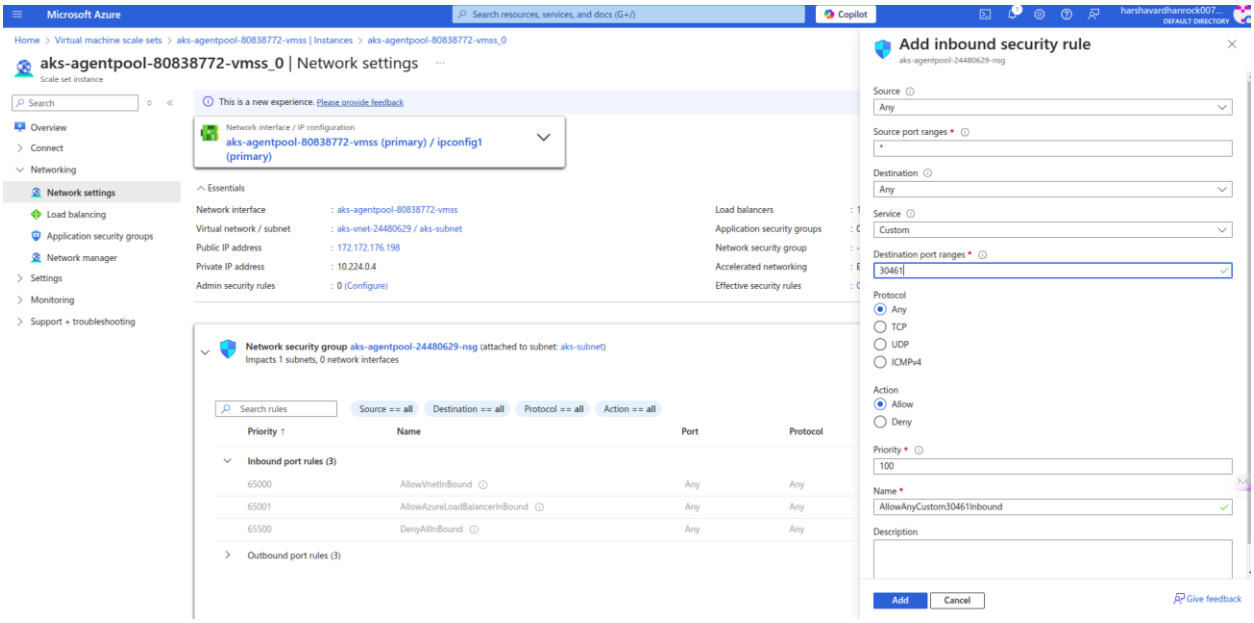
kubectl create secret docker-registry <secret-name> \
  --namespace <namespace> \
  --docker-server=<registry>.azurecr.io \
  --docker-username=<SP-ID> \
  --docker-password=<SP-Password>

- Verify using:

kubectl get pods –w

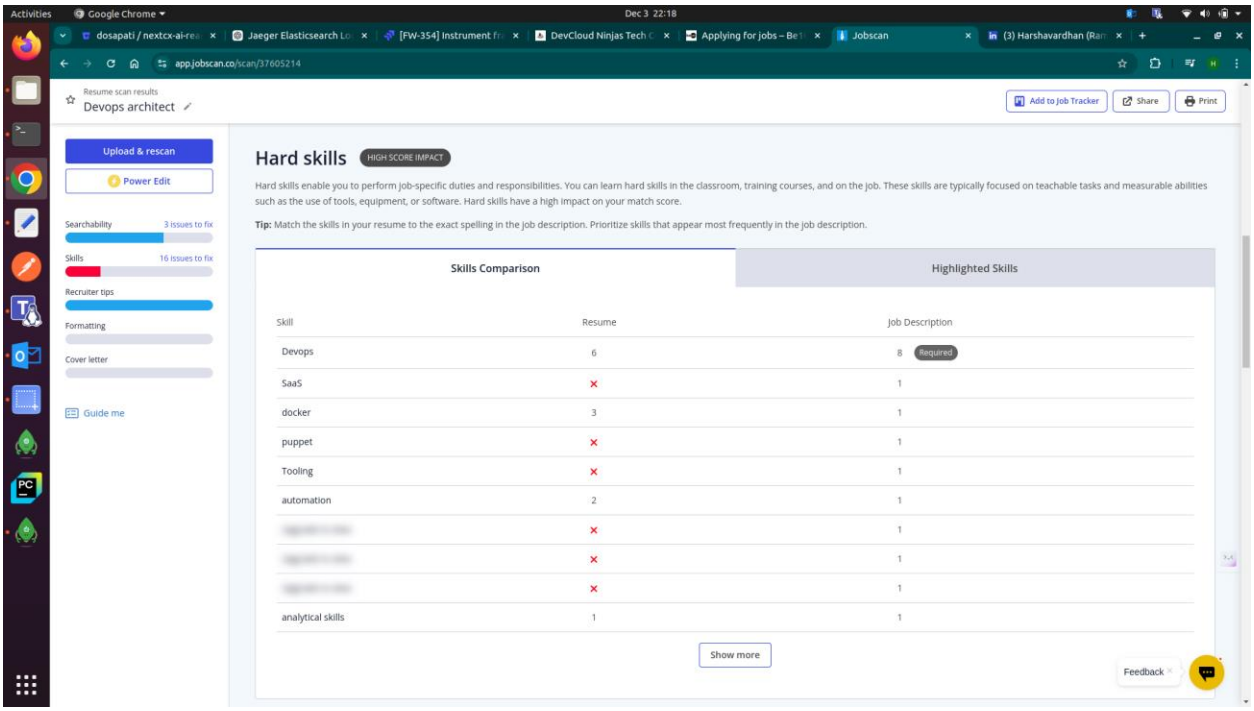
13. Access ArgoCD UI

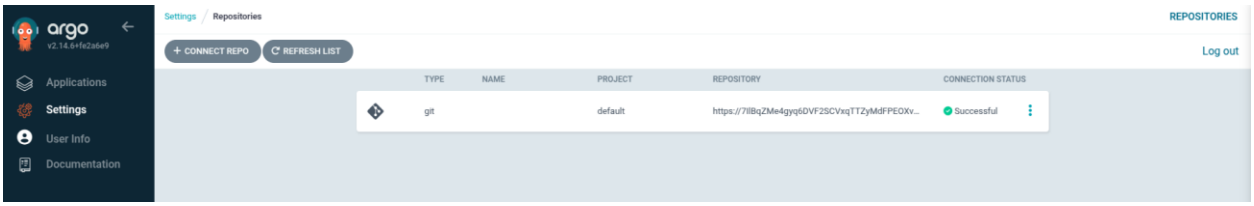
- Allow the ArgoCD NodePort in **Inbound NSG Rules**.
- Access the ArgoCD UI at: http://<external-ip>:<node-port>

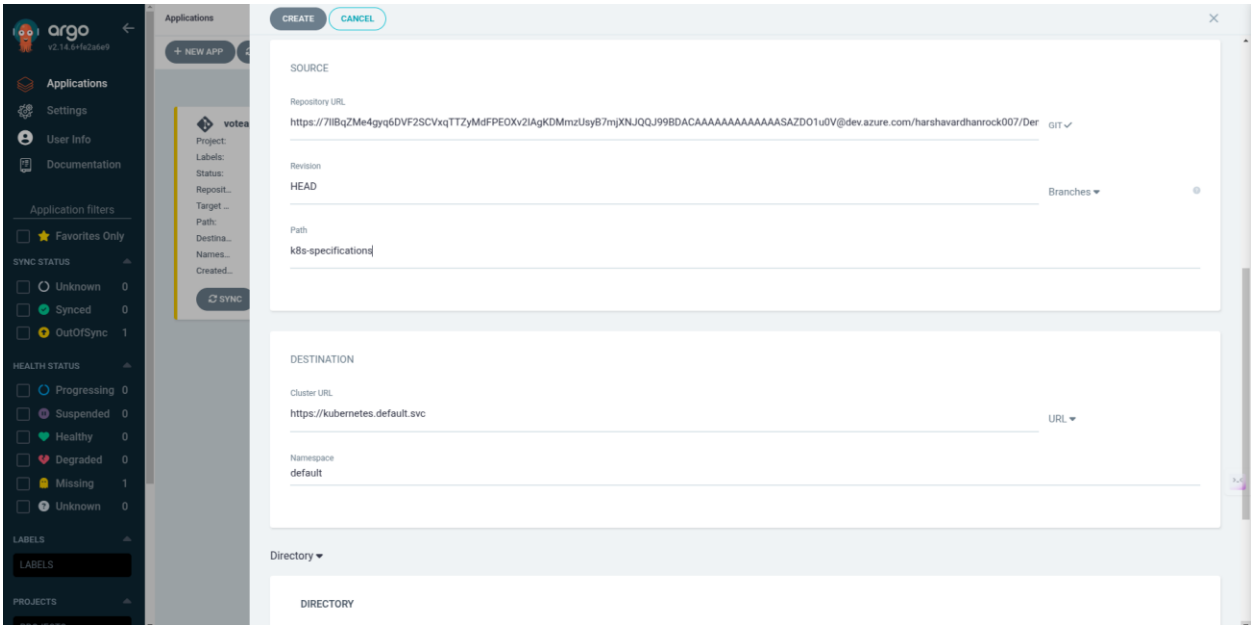


14. Configure ArgoCD Application

- In ArgoCD UI → Go to **Settings > Repositories** → Add Git Repo.
- Create **Application**:
  - Git Repo URL
  - Revision/Path
  - Cluster + Namespace


Activities
Google Chrome
Dec 3 22:18
Jobscan
app.jobscan.co/scan/37605214
Resume scan results
Devops architect
Add to Job Tracker
Share
Print
Upload & rescan
Power Edit
Searchability
3 issues to fix
Skills
16 issues to fix
Recruiter tips
Formatting
Cover letter
Guide me
Hard skills
HIGH SCORE IMPACT
Hard skills enable you to perform job-specific duties and responsibilities. You can learn hard skills in the classroom, training courses, and on the job. These skills are typically focused on teachable tasks and measurable abilities such as the use of tools, equipment, or software. Hard skills have a high impact on your match score.
Tip: Match the skills in your resume to the exact spelling in the job description. Prioritize skills that appear most frequently in the job description.
Skills Comparison
Highlighted Skills
Skill
Resume
Job Description
Devops
6
8
Required
SaaS
1
docker
3
1
puppet
1
Tooling
1
automation
2
1
analytical skills
1
1
Show more
Feedback


argo
v2.14.6+fe2a6e9
Settings
Repositories
REPOSITORIES
Log out
Applications
Settings
User Info
Documentation
+ CONNECT REPO
REFRESH LIST
TYPE
NAME
PROJECT
REPOSITORY
CONNECTION STATUS
git
default
https://7lIBqZMe4gyq6DVF2SCVxqTTZyMdFPE0Xv...
Successful

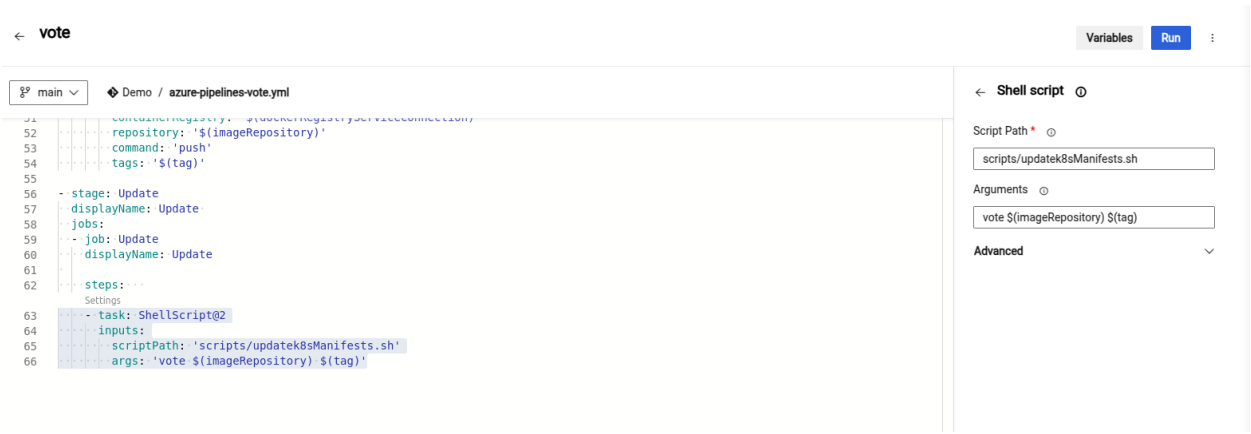

argo
v2.14.6+fe2a6e9
Applications
Settings
User Info
Documentation
Application filters
Favorites Only
SYNC STATUS
Unknown
Synced
OutOfSync
HEALTH STATUS
Progressing
Suspended
Healthy
Degraded
Missing
Unknown
LABELS
PROJECTS
Applications
+ NEW APP
SYNC
CREATE
CANCEL
SOURCE
Repository URL
https://7lIBqZMe4gyq6DVF2SCVxqTTZyMdFPE0Xv2lAgKDMmzUsyB7mjXNJQQJ99BDACAAAAAAAAAAAAASAZDO1u0V@dev.azure.com/harshavardhanrock007/Der
GIT
Revision
HEAD
Branches
Path
k8s-specifications
DESTINATION
Cluster URL
https://kubernetes.default.svc
URL
Namespace
default
Directory
DIRECTORY

main / scripts / updatek8sManifests.sh

updatek8sManifests.sh Edit

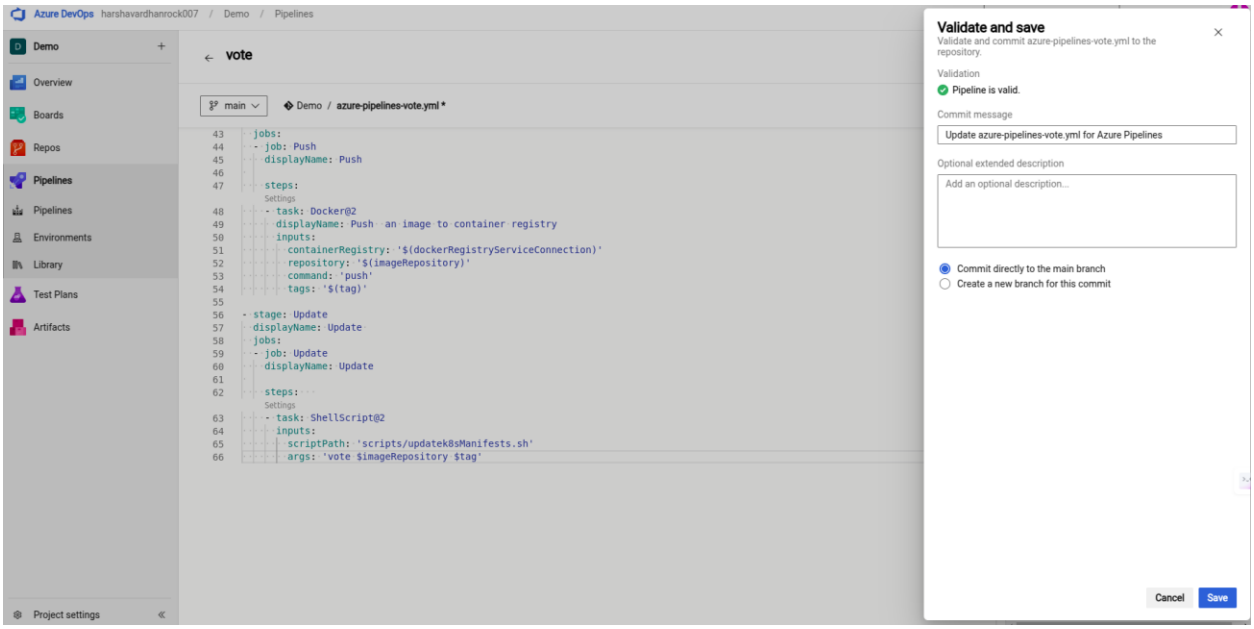
Contents History Compare Blame

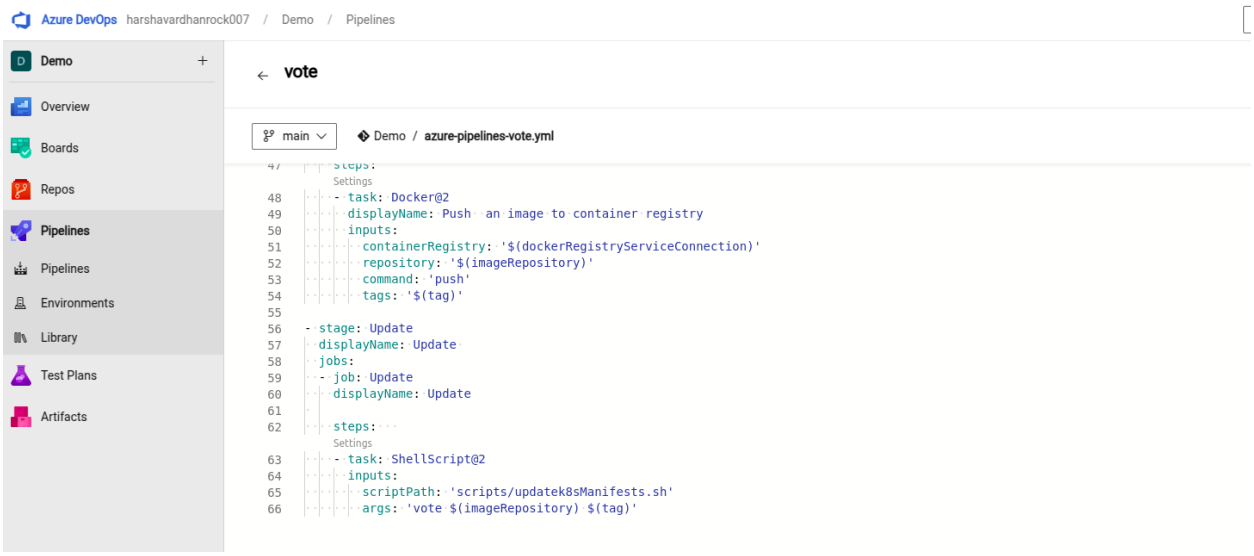
```
#!/bin/bash

set -x

# Set the repository URL
REPO_URL="https://7IlBqZMe4gyq6DVF2SCVxqTTZyMdFPEOXv2IAgKDMmzUsyB7mjXNJQQJ99BDACAAAAAAAAAAAAASAZDO1u0V@dev.azure.com/harshavardhanrock007/Demo/_git/Demo"

# Clone the git repository into the /tmp directory
git clone "$REPO_URL" /tmp/temp_repo

# Navigate into the cloned repository directory
cd /tmp/temp_repo

# Make changes to the Kubernetes manifest file(s)
# For example, let's say you want to change the image tag in a deployment.yaml file
sed -i "s|image:.*|image: harshacicd/$2:$3|g" k8s-specifications/$1-deployment.yaml

# Add the modified files
git add .

# Commit the changes
git commit -m "Update Kubernetes manifest"

# Push the changes back to the repository
git push

# Cleanup: remove the temporary directory
rm -rf /tmp/temp_repo
```

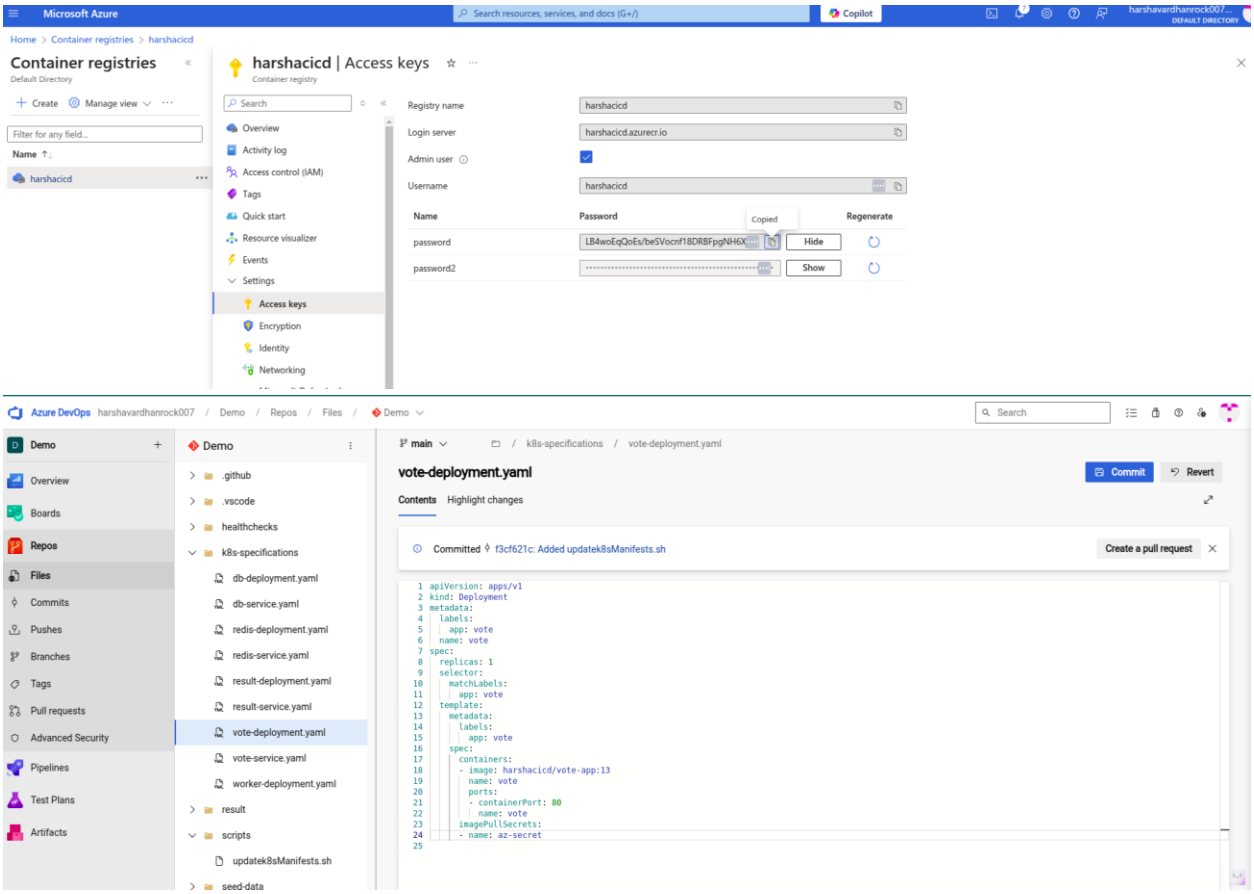


15. Add Automation Scripts

- In your repo → Create folder scripts/
- Add necessary .sh files
- In YAML, add a new stage update to run scripts.

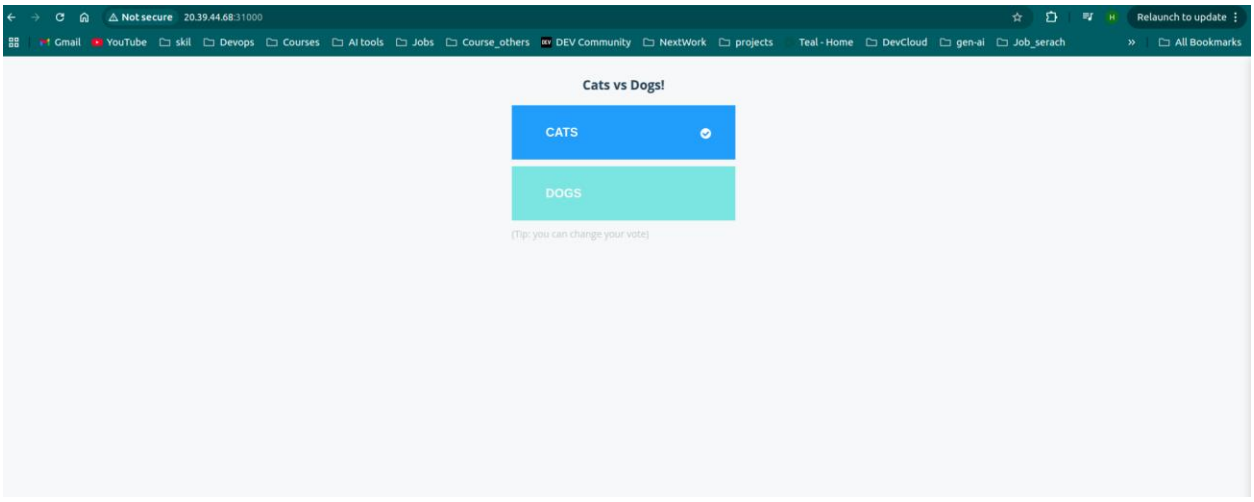

Azure DevOps harshavardhanrock007 / Demo / Pipelines
Demo
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
Project settings
vote
main
Demo / azure-pipelines-vote.yml *
jobs:
- job: Push
displayName: Push
steps:
Settings
- task: Docker@2
displayName: Push an image to container registry
inputs:
containerRegistry: '$(dockerRegistryServiceConnection)'
repository: '$(imageRepository)'
command: 'push'
tags: '$(tag)'
- stage: Update
displayName: Update
jobs:
- job: Update
displayName: Update
steps:
Settings
- task: ShellScript@2
inputs:
scriptPath: 'scripts/updatek8sManifests.sh'
args: 'vote $imageRepository $tag'
Validate and save
Validate and commit azure-pipelines-vote.yml to the repository.
Validation
Pipeline is valid.
Commit message
Update azure-pipelines-vote.yml for Azure Pipelines
Optional extended description
Add an optional description...
Commit directly to the main branch
Create a new branch for this commit
Cancel
Save

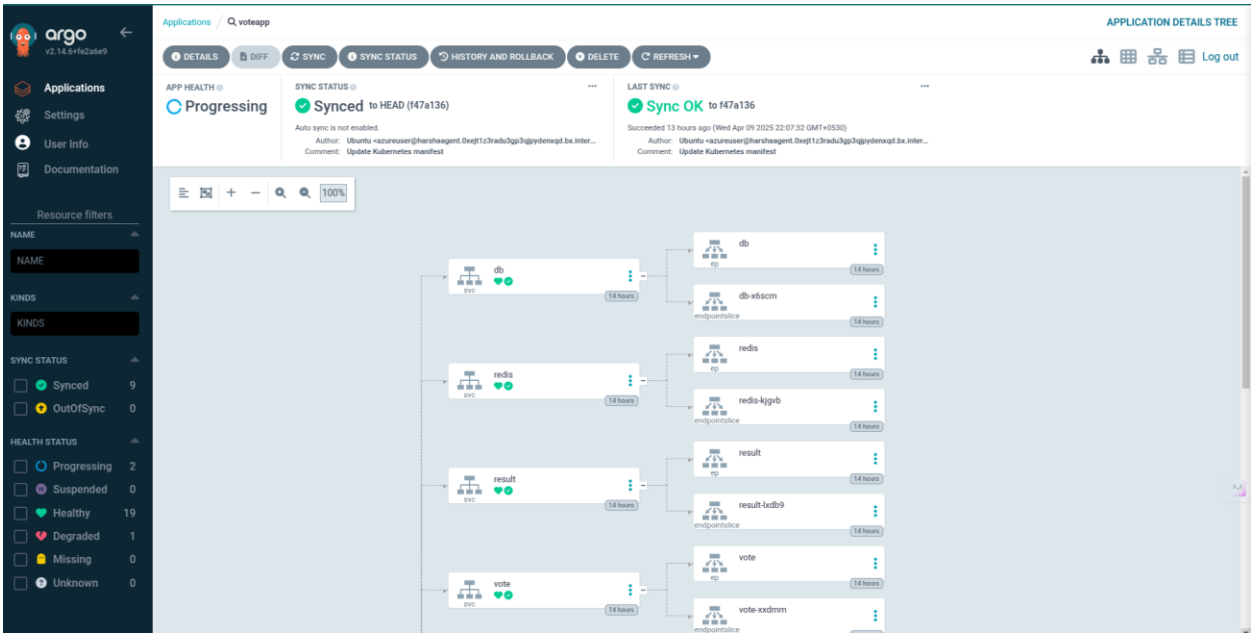

Azure DevOps harshavardhanrock007 / Demo / Pipelines
Demo
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
vote
main
Demo / azure-pipelines-vote.yml
steps:
Settings
- task: Docker@2
displayName: Push an image to container registry
inputs:
containerRegistry: '$(dockerRegistryServiceConnection)'
repository: '$(imageRepository)'
command: 'push'
tags: '$(tag)'
- stage: Update
displayName: Update
jobs:
- job: Update
displayName: Update
steps:
Settings
- task: ShellScript@2
inputs:
scriptPath: 'scripts/updatek8sManifests.sh'
args: 'vote $(imageRepository) $(tag)'



16. Redeploy and Validate

- Redeploy the pods
- Access the ArgoCD UI and validate your deployment visually





🖼️ Visual Breakdown

- 💡 Include diagrams showing worker, vote, and results jobs.
- 🖥️ Screenshots of ArgoCD UI, pipeline execution, and VM agent.

☑ Final Thoughts

This setup enables a complete **CI/CD workflow** using **Azure DevOps**, **Kubernetes**, and **ArgoCD**. You now have a scalable, secure, and automated deployment pipeline with modern DevOps best practices.