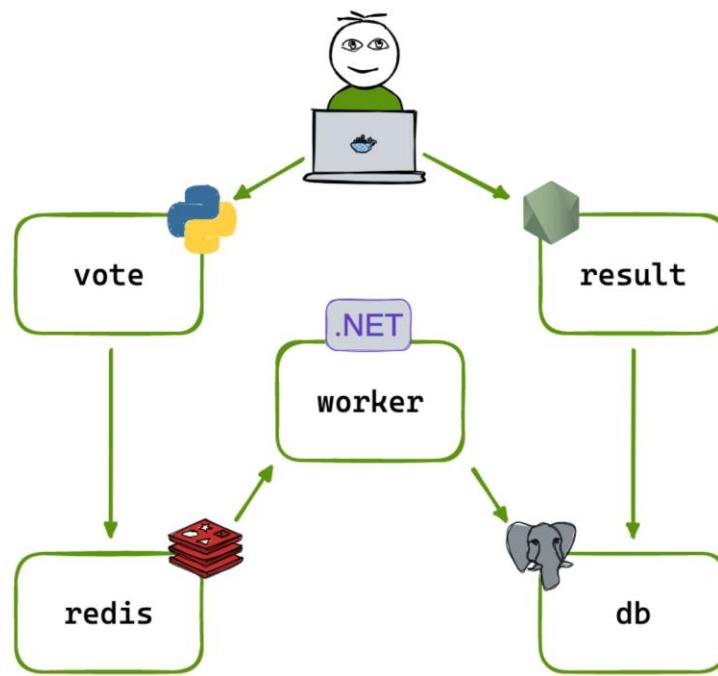


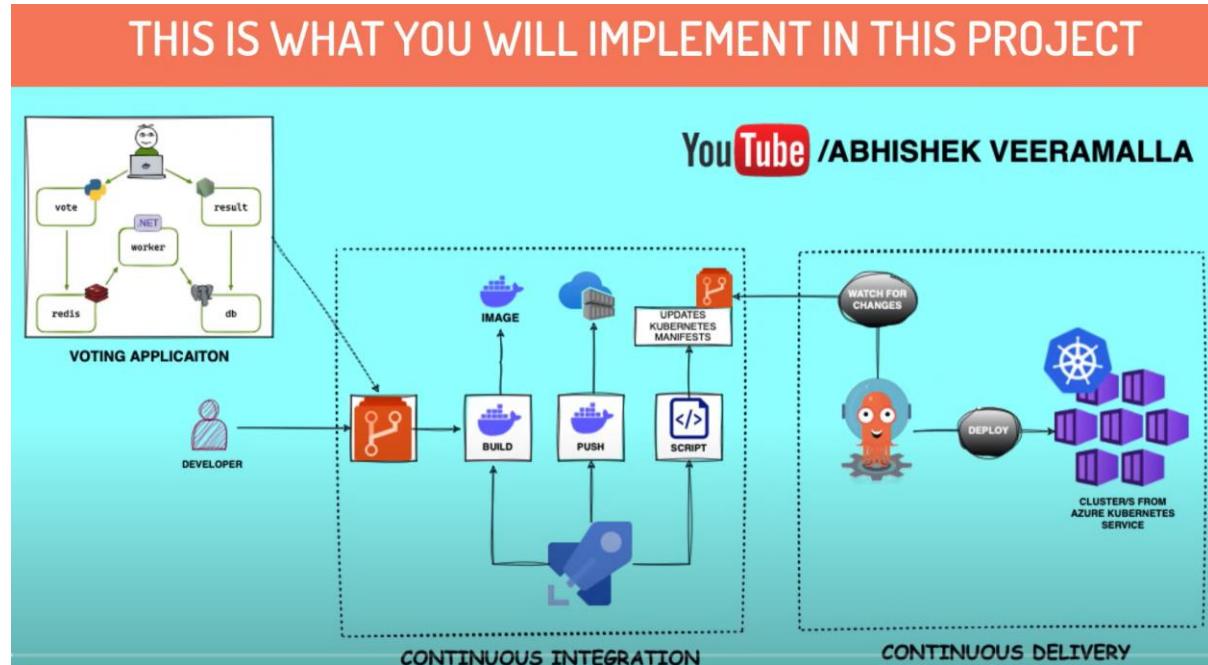
Day 14

App Repo: <https://github.com/dockersamples/example-voting-app>



Project Flow: Migrating the source code from GitHub repo to Azure DevOps Repo and the CICD using Azure DevOps Pipelines and Argo CD. During CI – Build using docker and push to Azure Container registry. And then updating the K8s manifest files in CI pipeline and deployment using Argo Cd into AKS Cluster.

Prefer to do entire project in Ubuntu machine:



Let's do this application CICD in Azure DevOps:

Let's first import the repo into **Azure DevOps Repos**:

Login to Azure DevOps and then create a new project with name Voting-App.

The screenshot shows the Azure DevOps interface for a project named 'Voting-App'. The left sidebar has 'Overview' selected. The main area displays 'Project stats' and a summary of the repository. At the top right, there are 'Private' and 'Invite' buttons.

Go to Repo and add then “Import a Repository”

Github Repo URL from Dockersamples: <https://github.com/dockersamples/example-voting-app>

The screenshot shows the GitHub repository 'example-voting-app'. It has 15 issues, 25 pull requests, and 122 watchers. The 'Code' tab is selected. A modal window titled 'Import a Git repository' is open, showing the 'Clone URL' field filled with 'https://github.com/dockersamples/example-voting-app.git'.

Copy the URL and give it in Azure DevOps.

The screenshot shows the 'Import a Git repository' dialog in Azure DevOps. The 'Repository type' is set to 'Git' and the 'Clone URL' field contains 'https://github.com/dockersamples/example-voting-app.git'. The 'Requires Authentication' checkbox is unchecked.

Entire Git repo is here now:

The screenshot shows the 'Files' view in Azure DevOps. The imported repository 'Voting-App' is visible under the 'Repos' section. The 'Files' tab is selected, showing the contents of the repository, including '.github', '.vscode', 'healthchecks', 'k8s-specifications', 'result', 'seed-data', and 'vote' directories.

The default branch is different than GitHub repo default branch:

The screenshot shows the 'Files' view in Azure DevOps. The imported repository 'Voting-App' is visible under the 'Repos' section. The 'Files' tab is selected, showing the contents of the repository. A modal window titled 'Switch branches/tags' is open, showing the 'Branches' tab with 'main' and 'default' branches listed. The 'default' branch is highlighted as the 'Default' branch.

We have to change the default branch in Repos:

Branches							New branch
Mine	All	Stale	Search branch name				
Branch	Com...	Author	Authored ...	Behind Ahead	Status	Pull ...	
> dependabot							
duplicate_key	67e2e08	sophia parafina	21 Jan 20...	154 0			
main	63e9158	Michael Irwin	10 Oct				
revert-54-connection_pooling	9352ed0	Mano Marks	27 Dec 2...	163 0			

Go to Repo to check the default branch:

The screenshot shows the Azure DevOps repository interface for the 'Voting-App' project. On the left, there's a sidebar with 'Overview', 'Boards', 'Repos' (selected), and 'Files'. The main area shows a tree view with 'github', 'vscode', 'healthchecks', 'k8s-specifications', and 'result' branches. A dropdown menu above the tree indicates 'main' is the current branch. Below the tree, a search bar says 'Filter branches' and a list shows 'main' (selected) and 'dependabot/npm_and_yarn/result/express-4.19.2'. At the bottom, a commit list includes one from 'dependabot/nugget/worker/Npgsql-8.0.3'.

Now we have to create pipelines. We have to create three pipelines for Voting app, Result app and worker app. And whatever the build artifact we will receive will be placed in azure container registry in azure portal.

Lets create container registry first, and also resource group:

Login to Azure Portal and create a resource group.

The screenshot shows the Azure Resource Groups page. It lists a single resource group named 'voting-app' under 'Subscription' 'Free Trial' and 'Location' 'East US'. There are filters at the top for 'Subscription equals all' and 'Location equals all'.

Go to Container Registry and create one registry:

The screenshot shows the 'Create container registry' page. Under 'Project details', it shows 'Subscription' set to 'Free Trial' and 'Resource group' set to 'voting-app'. Under 'Instance details', it shows 'Registry name' as 'votingappbuild' and 'Location' as 'East US'. There are sections for 'Use availability zones' and 'Pricing plan' (set to 'Standard').

Click on Review + Create.

The screenshot shows the 'Container registries' list page. It displays one record: 'votingappbuild' (Type: Container registry, Resource group: voting-app, Location: East US, Subscription: Free Trial). There are filters at the top for 'Subscription equals all', 'Resource group equals all', and 'Location equals all'.

Now go to Azure Devops and create pipelines for Result MS first: Click on create pipeline.

New pipeline
Where is your code?

- Azure Repos Git** **YAML**
Free private Git repositories, pull requests, and code search
- Bitbucket Cloud** **YAML**
Hosted by Atlassian

Select Azure Repos Git.

New pipeline
Select a repository

Filter by keywords **Voting-App**

Voting-App

Select the repo.

New pipeline
Configure your pipeline

- Docker** **docker** Build a Docker image
- Docker** **docker** Build and push an image to Azure Container Registry

Select the template for CI pipeline. Select the Second one “Buidl and push the image to Azure Conatiner Registry”. Then it will load the subscription. Select free trail.

Docker
Build and push an image to Azure Container Registry

Select an Azure subscription

Free Trial 00681e52-1746-42b2-b0a5-ad83a3b482af

Then login with the Azure portal account.

Docker
Build and push an image to Azure Container Registry

Container registry **votingappbuild**

Image Name **votingapp**

Dockerfile **\$(Build.SourcesDirectory)/result/Dockerfile**

Then it will show the Container registry we had in our account. It creates a connection to the Azure container registry with the CI pipeline. Click on Validate and configure.

Review your pipeline YAML

```

1  # Docker
2  # Build and push an image to Azure Container Registry
3  # https://docs.microsoft.com/azure/devops/pipelines/languages/docker
4
5  trigger:
6  - main
7
8  resources:
9  - repo: self
10
11 variables:
12  # Container registry service connection established during pipeline creation
13  dockerRegistryServiceConnection: '0dc91bee-d6cb-4755-9ff3-11df94af19af'
14  imageRepository: 'votingapp'
15  containerRegistry: 'votingappbuild.azurecr.io'
16  dockerfilePath: '$(Build.SourcesDirectory)/result/Dockerfile'
17  tag: '$(Build.BuildId)'
```

Pipeline template with some code snippets is already written. Lets modify this:

Change the pipeline name:

New pipeline

Review your pipeline YAML

Variables Save and run

Voting-App / azure-pipelines-result.yml * ⊕

Show assistant

```
trigger:
  paths:
    - include:
      - result/*
      # any change inside the result folder will trigger this pipeline.

resources:
- repo: self
variables:
  # Container registry service connection established during pipeline creation
  dockerRegistryServiceConnection: '0dc91bee-d6cb-4755-9ff3-11df94af19af'
  imageRepository: 'resultapp'
  containerRegistry: 'votingappbuild.azurecr.io'
  dockerfilePath: '${Build.SourcesDirectory}/result/Dockerfile'
  tag: '$(Build.BuildId)'
```

Update the pipeline:

```
trigger:
  paths:
    - include:
      - result/*
      # any change inside the result folder will trigger this pipeline.

resources:
- repo: self
variables:
  # Container registry service connection established during pipeline creation
  dockerRegistryServiceConnection: '0dc91bee-d6cb-4755-9ff3-11df94af19af'
  imageRepository: 'resultapp'
  containerRegistry: 'votingappbuild.azurecr.io'
  dockerfilePath: '${Build.SourcesDirectory}/result/Dockerfile'
  tag: '$(Build.BuildId)'
```

Azure doesn't provide an agent to run the pipeline in free tier account. Instead we can create our own agent:

```
pool:
  name: 'ciagent'
  # should have an agent VM with this name created in azure portal to use as agent to run this pipeline
```

Now we have to go back to azure and create a VM and this VM will act as a Agent for our CI pipeline.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ① Free Trial

Resource group * ① voting-app >Create new

Instance details

Virtual machine name * ① ciagent

Region * ① (US) East US

Security type ① Trusted launch virtual machines Configure security features

Image * ① Ubuntu Server 24.04 LTS - x64 Gen2 (free services eligible) See all images | Configure VM generation

VM architecture ① Arm64 x64

Administrator account

Authentication type ① SSH public key

Username * ① azureuser

SSH public key source Generate new key pair

SSH Key Type RSA SSH Format Ed25519 SSH Format

Key pair name * ciagent_key

VM is created.

Home > CreateVm-canonical.ubuntu-24_04-lts-server-20241028013144 | Overview >

ciagent Virtual machine

ciagent virtual machine agent status is not ready. Troubleshoot the issue →

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Connect Start Stop Hibernate Capture Delete Refresh Open in mobile Feedback CLI / PS

Essentials

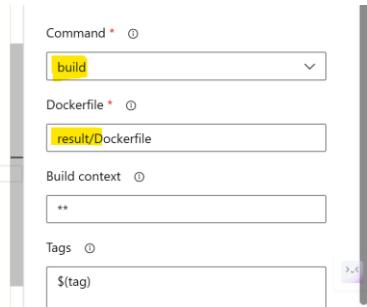
Resource group (move)	voting-app	Operating system	: Linux
Status	: Running	Size	: Standard B1s (1 vcpu, 1 GiB memory)
Location	: South India	Public IP address	: 52.172.99.184
Subscription (move)	: Free Trial	Virtual network/subnet	: ciagent-vnet/default
Subscription ID	: 00601e52-1746-42b2-b0a5-ad83a3b482af	DNS name	: Not configured
Health state	: -	Time created	: 27/10/2024, 20:07 UTC

JSON View

Now lets update the stages:

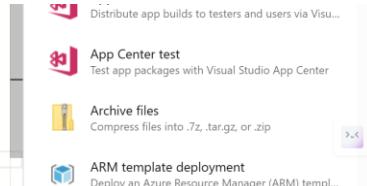
We will take only build in one stage. You can get help by clicking on settings and you can update in the right side popup menu and it will be added into the pipeline file.

```
26 stages:
27 - stage: Build
28   displayName: Build
29   jobs:
30     - job: Build
31       displayName: Build
32       steps:
33         - task: Docker@2
34           displayName: Build image
35           inputs:
36             command: buildAndPush
37             repository: $(imageRepository)
38             dockerfile: $(dockerfilePath)
39             containerRegistry: $(dockerRegistryServiceConnection)
40             tags: |
41               $(tag)
```



Then click on Add button. Task automatically updates.

```
33   - task: Docker@2
34     displayName: Build
35     inputs:
36       containerRegistry: '$(dockerRegistryServiceConnection)'
37       repository: '$(imageRepository)'
38       command: 'build'
39       Dockerfile: 'result/Dockerfile'
40       tags: '$(tag)'
```



Copy the whole stage and paste it to add another stage for push.

```
- stage: Push
  displayName: Push
  jobs:
    - job: Push
      displayName: Push the Image
      steps:
        Settings
        - task: Docker@2
          displayName: Build
          inputs:
            containerRegistry: '$(dockerRegistryServiceConnection)'
            repository: '$(imageRepository)'
            command: 'build'
            Dockerfile: 'result/Dockerfile'
            tags: '$(tag)'
```

Click on settings button and update the command from build to push for Build task.

```
41   - stage: Push
42     displayName: Push
43     jobs:
44       - job: Push
45         displayName: Push
46         steps:
47           Settings
48           - task: Docker@2
49             displayName: Push the Image
50             inputs:
51               containerRegistry: '$(dockerRegistryServiceConnection)'
52               repository: '$(imageRepository)'
53               command: 'push'
54               tags: '$(tag)'
```

Click on Save and Run:

The screenshot shows the Azure Pipelines pipeline run page. At the top, it displays the pipeline name '#20241027.1 • Update azure-pipelines-result.yml for Azure Pipelines' and the project 'Voting-App'. There are buttons for 'Rerun failed jobs', 'Run new', and more. A note says 'This run will be cleaned up after 1 month based on your project settings.' Below this, there are tabs for 'Summary' (which is selected) and 'Code Coverage'. The summary section shows that the pipeline was manually run by 'Varuntej Yadla'. It provides details about the repository ('Repository and version: Voting-App main 89780ae7'), the run time ('Time started and elapsed: Just now <1s'), related work items ('Related: 0 work items'), and artifacts ('Tests and coverage: Get started'). A message at the bottom states: 'There was a resource authorization issue: "The pipeline is not valid. Could not find a pool with name ciagent. The pool does not exist or has not been authorized for use. For authorization details, refer to https://aka.ms/yamlauthz. Could not find a pool with name ciagent. The pool does not exist or has not been authorized for use. For authorization details, refer to https://aka.ms/yamlauthz."'. There is also a link to 'Authorize resources'.

Now, we have to make a connection to the pipeline and VM we created.

Documentation: <https://learn.microsoft.com/en-us/azure/devops/pipelines/agents/linux-agent?view=azure-devops>

Follow this documentation to add the VM we created in the pool. For our Project we will go with Project settings in the bottom of the screen inside the project page (Voting-App) → Agent Pools → Click on add pool

The screenshot shows the 'Agent pools' section within the 'Project Settings' of the 'Voting-App' project. On the left, there is a sidebar with 'General' settings like Overview, Teams, Permissions, Notifications, Service hooks, Dashboards, Boards, Project configuration, Team configuration, GitHub connections, and Pipelines. The main area shows 'Agent pools' with three entries: 'Azure Pipelines' (selected), 'Default', and 'ciagent'. A modal window titled 'Add agent pool' is open, showing fields for 'Name' (set to 'ciagent'), 'Pool type' (set to 'Self-hosted'), 'Description (optional)' (set to 'pool for running CI pipelines'), and 'Pipeline permissions' (with 'Grant access permission to all pipelines' checked). There is also a note: 'Agent pools are shared across an organization.'

Once pool is added it will look like this:

The screenshot shows the 'Agent pools' list in the 'Project Settings' of the 'Voting-App' project. The table lists three agent pools: 'Azure Pipelines' (selected), 'ciagent', and 'Default'. Each row includes a 'Name' column, 'Queued jobs' column, 'Running jobs' column, and a delete icon. The 'ciagent' row shows 'Varuntej Yadla' under 'Name'.

Name	Queued jobs	Running jobs
Azure Pipelines		
ciagent	Varuntej Yadla	
Default		

Click on the Agent → Again click on Agent and then click on New Agent

We need to copy these commands and paste in our VM we created as a agent.

The screenshot shows the 'Get the agent' dialog in the Azure DevOps Portal. The 'Linux' tab is selected. Under 'System prerequisites', there's a link to 'Configure your account'. Below that is a 'Download' button. The 'Create the agent' section contains three code snippets:

- Snippet 1: `~\$ mkdir myagent && cd myagent
~/myagent\$ tar zxvf ~/Downloads/vsts-agent-linux-x64-3.246.0.tar.gz`
- Snippet 2: `~/myagent\$./config.sh`
- Snippet 3: `~/myagent\$./run.sh`

At the bottom, there's a 'That's it!' link.

Connect to the VM: Using Public IP when SSH.

```
yvaru@Tej MINGW64 ~  
$ pwd  
/c/Users/yvaru  
yvaru@Tej MINGW64 ~  
$ cd Downloads  
yvaru@Tej MINGW64 ~/Downloads  
$ ls -al | grep 'cia'  
-rw-r--r-- 1 yvaru 197121 2494 Oct 28 01:36 ciagent_key.pem  
yvaru@Tej MINGW64 ~/Downloads  
$ ssh -i ciagent_key.pem azureuser@52.172.99.184  
The authenticity of host '52.172.99.184 (52.172.99.184)' can't be established.  
ED25519 key fingerprint is SHA256:3g1eQhqcNhyK2TnSi64Q/2AX5tLmeD5NiT2wXKHbUUc.  
This key is not known by any other names.
```

give sudo apt update command first.

```
azureuser@ciagent: $ mkdir myagent && cd myagent  
azureuser@ciagent: ~/myagent$ wget https://vstsagentpackage.azureedge.net/agent/3.246.0/vsts-agent-linux-x64-3.246.0.tar.gz  
--2024-10-27 20:30:31-- https://vstsagentpackage.azureedge.net/agent/3.246.0/vsts-agent-linux-x64-3.246.0.tar.gz  
Resolving vstsagentpackage.azureedge.net (vstsagentpackage.azureedge.net)... 117.18.232.200, 2606:2800:147:120f:30c:1ba0:fc6:265a  
Connecting to vstsagentpackage.azureedge.net (vstsagentpackage.azureedge.net)|117.18.232.200|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 146311562 (140M) [application/octet-stream]  
Saving to: 'vsts-agent-linux-x64-3.246.0.tar.gz'  
  
vsts-agent-linux-x64-3.246.0.tar.gz 100%[=====] 139.53M 95.5MB/s in 1.5s  
2024-10-27 20:30:33 (95.5 MB/s) - 'vsts-agent-linux-x64-3.246.0.tar.gz' saved [146311562/146311562]  
azureuser@ciagent:~/myagent$
```

Before giving the second command in the create agent, give wget URL (Copy it from Download the agent from Azure DevOps Portal in above image)

Then give the second command: Remove the part before vsts-agent.....

```
azureuser@ciagent:~/myagent$ tar zxvf vsts-agent-linux-x64-3.246.0.tar.gz
```

Then give the next command:

```
azureuser@ciagent:~/myagent$ ./config.sh  
[██████████] ██████████ [██████████] ██████████ [██████████]  
agent v3.246.0 (commit 59c8fe5)  
  
>> 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 sou  
rees 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) > Y  
>> Connect:  
Enter server URL > |
```

Type Y and then add server URL: Modify the Organisation name in the URL.

URL we can see what to give in the documentation.

- Agents & pools
 - About agents & agent pools
 - Managed DevOps pools
 - Agent version 3.x
 - Create & manage agent pools

8. Unpack the agent into the directory of your choice. `cd` to that directory and run `./config.sh`.

Server URL

Azure Pipelines: `https://dev.azure.com/{your-organization}`

My organisation name:

The screenshot shows the Azure DevOps interface. In the top left, there's a profile icon and the text "varuntej06". Below it, a "New organization" button is visible. The main navigation bar has three items: "Projects" (which is underlined), "My work items", and "My pull requests". A large text input field at the bottom contains the placeholder "Enter server URL > https://dev.azure.com/varuntej06" and the instruction "Enter authentication type (press enter for PAT) >".

For PAT: Click on User settings → PAT

The screenshot shows the "User settings" page for the user "varuntej06". On the left, there's a sidebar with options like "Account", "Preferences", and "Notifications". The main content area is titled "Personal Access Tokens" and contains a sub-instruction "These can be used instead of a password for applications like Git or can be passed in the authorization header to access REST APIs". It shows a single token entry for "ciagent" with "Full access" rights, belonging to the organization "varuntej06". The status is "Active" and it expires on "27/11/2024". On the right, a vertical sidebar lists "User settings" options: Preview features, Profile, Time and Locale, Permissions, Notifications, Theme, Usage, Personal access tokens (which is currently selected), and SSH public keys.

Click on New Token and give full access to this token and make sure not to miss copying the code.

The screenshot shows the "Manage tokens" page. It lists a single token named "ciagent" with "Full access" rights, belonging to the organization "varuntej06". The status is "Active" and it expires on "27/11/2024". At the top, there are buttons for "Revoke", "Regenerate", and "Edit". Below the table, there's a text input field with placeholder text: "Enter server URL > https://dev.azure.com/varuntej06", "Enter authentication type (press enter for PAT) >", "Enter personal access token > ****", and "Connecting to server ...".

Hit Enter and then paste the PAT and then again hit enter key.

The screenshot shows a terminal window with the command "Register Agent:" followed by "Enter agent pool (press enter for default) > |".

It is asking for agent pool: give the agent pool you created before:

The screenshot shows the "Project Settings" page for the "Voting-App" project. On the left, there's a sidebar with "General" settings like "Overview", "Teams", "Permissions", "Notifications", and "Service hooks". The main content area is titled "Agent pools" and lists three pools: "Azure Pipelines" (under "Azure Pipelines"), "ciagent" (under "Varuntej Yadla"), and "Default" (under "Azure Pipelines"). At the top, there are buttons for "Search", "Security", and "Add pool". Below the table, there's a text input field with placeholder text: "Enter agent pool (press enter for default) > ciagent", "Enter agent name (press enter for ciagent) > |", and "Connecting to the server...".

It is asking for agent name. It is also same so press enter.

The screenshot shows a terminal window with the following output:
 Registering agent...
 Enter agent pool (press enter for default) > ciagent
 Enter agent name (press enter for ciagent) >
 Scanning for tool capabilities.
 Connecting to the server.
 Successfully added the agent
 Testing agent connection.
 Enter work folder (press enter for _work) > |

press enter.

The screenshot shows a terminal window with the message "2024-10-27 20:47:36Z: Settings Saved." and the prompt "azureuser@ciagent: /myagent\$".

Run the last command:

```
azureuser@ciagent:~/myagent$ ./run.sh
Scanning for tool capabilities.
Connecting to the server.
2024-10-27 20:49:39Z: Listening for Jobs
```

We are building our CI build from docker. So, we need to install docker in the VM agent. We missed it lets install it.

```
azureuser@ciagent:~/myagent$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
```

We are not using the root user for this VM. We configured the agent using this azureuser which is the default user for this machine. So, we have to run below commands.

```
azureuser@ciagent:~/myagent$ sudo usermod -aG docker azureuser
azureuser@ciagent:~/myagent$ sudo systemctl restart docker
```

Now, start the agent again:

```
azureuser@ciagent:~/myagent$ ./run.sh
Scanning for tool capabilities.
Connecting to the server.
2024-10-27 20:55:45Z: Listening for Jobs
```

Agent is added in the agent pool:

The screenshot shows the 'Agents' tab of the 'ciagent' agent pool in the Azure DevOps interface. The table lists one agent named 'ciagent' which is currently online and idle, running version 3.246.0. There are buttons for 'Update all agents' and 'New agent'.

Go to the pipeline: Run the pipeline again.

The screenshot shows the details of a pipeline run titled '#20241027.2 • Update azure-pipelines-result.yml for Azure Pipelines'. It includes a summary of the run, repository information (Voting-App, main branch, commit 89780ae7), and a list of errors. One error is highlighted: 'DEPRECATED: The legacy builder is deprecated and will be removed in a future release.' Below the errors, there is a terminal session showing the command 'sudo systemctl restart docker' being run.

Build failed as it is not able to connect to the Docker daemon.

Lets try to disconnect the VM, connect again and restart the docker service and check.

```
azureuser@ciagent:~/myagent$ logout
Connection to 52.172.99.184 closed.

vvaru@Tej MINGW64 ~/Downloads
$ ssh -i ciagent_key.pem azureuser@52.172.99.184
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-azure x86_64)
```

It is able to connect with docker deamon now:

```
azureuser@ciagent:~$ ls
myagent
azureuser@ciagent:~$ cd myagent/
azureuser@ciagent:~/myagent$ ls
.dia_ .work bin config.sh env.sh externals license.html reauth.sh run-docker.sh run.sh svc.sh vsts-agent-linux-x64-3.246.0.tar.gz
azureuser@ciagent:~/myagent$ sudo systemctl restart docker
azureuser@ciagent:~/myagent$ sudo docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:d211f485f2dd1de407a80973c8f129f00d546042c90732e8c320e5038a0348
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
azureuser@ciagent:~/myagent$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
azureuser@ciagent:~/myagent$
```

```
azureuser@ciagent:~/myagent$ ./run.sh
Scanning for tool capabilities.
Connecting to the server.
2024-10-27 21:05:14Z: Listening for Jobs
```

Now run the pipeline again:

The screenshot shows the Azure DevOps Pipeline interface for a project named 'Voting-App'. A manual run of pipeline #20241027.3 is displayed. The pipeline consists of two stages: 'Build' and 'Push'. The 'Build' stage has one job completed (0/1) in 22s. The 'Push' stage is not started. The pipeline summary indicates it was just now started and took 30s. There are 0 work items and 0 artifacts related to the pipeline.

#20241027.3 • Update azure-pipelines-result.yml for Azure Pipelines

Manually run by Varuntej Yadla

View 237 changes

Repository and version

Just now

Related

Get started

Time started and elapsed

30s

Related

0 work items

0 artifacts

Stages Jobs

Build Push

Build

0/1 completed 22s

Push

Not started

Cancel

Azure DevOps varuntej06 / Voting-App / Pipelines / Voting-App / 20241027.3

Finalize Job

Starting: Finalize Job

Cleaning up task key

Start cleaning up orphan processes.

Finishing: Finalize Job

Jobs in run #20241027.3

Build

Initialize job

Checkout Voting-App@...

Build the Image

Post-job: Checkout Vo...

Finalize Job

Push

Initialize job

Checkout Voting-App@...

Push the Image

Post-job: Checkout Vo...

Finalize Job

View raw log

Connecting to the server.

2024-10-27 21:05:14Z: Listening for Jobs

2024-10-27 21:05:49Z: Running job: Build

2024-10-27 21:07:02Z: Job Build completed with result: Succeeded

2024-10-27 21:07:07Z: Running job: Push

2024-10-27 21:08:03Z: Job Push completed with result: Succeeded

Now, you can modify some file in worker folder and check pipeline is triggering or not. It should not trigger, but if we change something in result folder it should trigger.

I added an empty line in server.js file:

The screenshot shows the Azure DevOps interface. On the left, the navigation bar has 'Pipelines' selected. In the center, a 'Commit' dialog is open for the 'server.js' file in the 'main' branch. The commit message is 'Updated server.js'. Below the commit dialog, the 'Pipelines' section shows a recent run for the 'Voting-App' pipeline, which completed successfully just now.

Run started:

The screenshot shows the 'Runs' tab for the 'Voting-App' pipeline. It lists two recent runs: one for 'Updated server.js' (status: Succeeded) and another for 'Update azure-pipelines-result.yml for Azure Pipelines' (status: Succeeded). The log output at the bottom shows the job details.

```

2024-10-27 21:11:27Z: Running job: Build
2024-10-27 21:12:12Z: Job Build completed with result: Succeeded
2024-10-27 21:12:16Z: Running job: Push
  
```

Now add pipeline for vote and worker as well in the same process:

The screenshot shows the 'azure-pipelines-vote.yml' pipeline configuration. The YAML code defines a pipeline with a single stage named 'Build'. This stage contains a job named 'Build' with a Docker task. The Docker task uses a container registry service connection and pushes the image to 'votingapp-build.azurecr.io'. The pipeline also includes variables for the Docker registry and tag, and specifies a pool named 'ciagent'.

```

1 # Docker
2 # Build and push an image to Azure Container Registry
3 # https://docs.microsoft.com/azure/devops/pipelines/languages/docker
4
5 trigger:
6   paths:
7     - include:
8       - vote/*
9   resources:
10  - repo: self
11
12 variables:
13   # Container registry service connection established during pipeline creation
14   dockerRegistryServiceConnection: '2e475f1b-951c-4742-ba6c-c438bb383341'
15   imageRepository: 'votingapp'
16   containerRegistry: 'votingapp-build.azurecr.io'
17   dockerfilePath: '$(Build.SourcesDirectory)/vote/Dockerfile'
18   tag: '$(Build.BuildId)'
19
20 # Agent VM image name
21 pool:
22  name: 'ciagent'
23
24 stages:
25  - stage: Build
26    displayName: Build
27    jobs:
28      - job: Build
29        displayName: Build
30        steps:
31          - task: Docker@2
32            displayName: Build image
33            inputs:
34              containerRegistry: '$(dockerRegistryServiceConnection)'
35              repository: '$(imageRepository)'
36              command: 'build'
37              Dockerfile: 'vote/Dockerfile'
38              tags: '$(tag)'
39
40 - stage: Push
  
```

```

39   - stage: Push
40     displayName: Push
41   jobs:
42     - job: Push
43       displayName: Push
44       steps:
45         - task: Docker@2
46           displayName: Push image to Container Registry
47           inputs:
48             containerRegistry: '$(dockerRegistryServiceConnection)'
49             repository: '$(imageRepository)'
50             command: 'push'
51             tags: '$(tag)'
```

#20241027.1 • Set up CI with Azure Pipelines

This run is retained as one of 3 recent runs by main (Branch).

[View retention leases](#)

[Run new](#)

[Summary](#) [Code Coverage](#)

Triggered by Varuntej Yadla [View 239 changes](#)

Repository and version Voting-App main ↗ 5b1d56a2

Time started and elapsed Today at 2:53 am 1m 54s

Related 0 work items 0 artifacts

Tests and coverage Get started

[Stages](#) [Jobs](#)

Rename the pipelines: Click on three dots on the pipeline and select rename:

[Pipelines](#) [New pipeline](#)

[Recent](#) [All](#) [Runs](#) [Filter pipelines](#)

Recently run pipelines

Pipeline	Last run	
Vote-service	#20241027.1 • Set up CI with Azure Pipelines Individual CI for ↗ main	2m ago 1m 54s
Result-service	#20241027.4 • Updated server.js Individual CI for ↗ main	15m ago 1m 22s

Creating pipeline for worker:

Voting-App / [azure-pipelines-worker.yml](#) *

```

1  # Docker
2  # Build and push an image to Azure Container Registry
3  # https://docs.microsoft.com/azure/devops/pipelines/languages/docker
4
5  trigger:
6    paths:
7      include:
8        - worker/*
9  resources:
10   - repo: self
11
12  variables:
13    # Container registry service connection established during pipeline creation
14    dockerRegistryServiceConnection: 'ffaea5df-46d6-4d00-8c4b-c537f41f40db'
15    imageRepository: 'votingapp'
16    containerRegistry: 'votingappbuild.azurecr.io'
17    dockerfilePath: '$(Build.SourcesDirectory)/worker/Dockerfile'
18    tag: '$(Build.BuildId)'
```

```

20   . . # Agent VM image name
21   pool:
22     name: 'ciagent'
23   stages:
24     - stage: Build
25       displayName: Build
26       jobs:
27         - job: Build
28           displayName: Build
29           steps:
30             - task: Docker@2
31               displayName: Build an Image
32               inputs:
33                 containerRegistry: '$(dockerRegistryServiceConnection)'
34                 repository: '$(imageRepository)'
35                 command: 'build'
36                 Dockerfile: 'worker/Dockerfile'
37                 tags: '$(tag)'

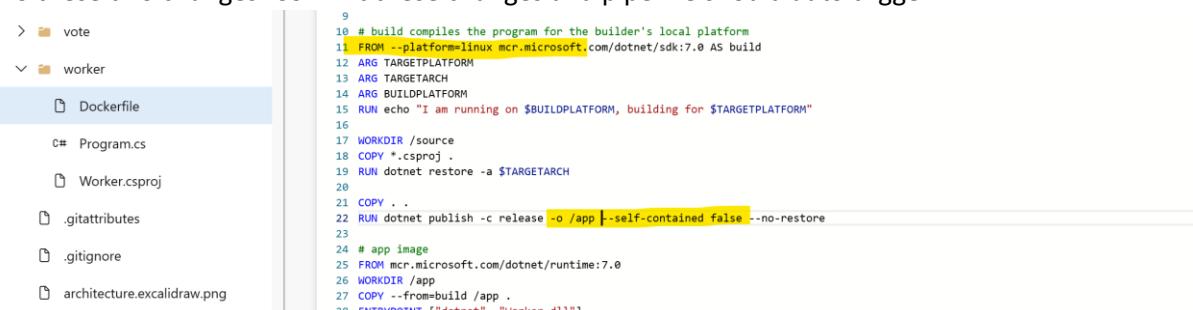
```

We added till build stage only. It will fail as there are some env variables defined in the docker file of this service.

Recently run pipelines

Pipeline	Last run	
Worker-service	#20241027.1 • Set up CI with Azure Pipelines ⌚ Individual CI for 🏙 main	⌚ Just now ⌚ 34s
Vote-service	#20241027.1 • Set up CI with Azure Pipelines ⌚ Individual CI for 🏙 main	⌚ 12m ago ⌚ 1m 54s
Result-service	#20241027.4 • Updated server.js ⌚ Individual CI for 🏙 main	⌚ 24m ago ⌚ 1m 22s
Errors 6		
✖ DEPRECATED: The legacy builder is deprecated and will be removed in a future release. Build an Image		
✖ Install the buildx component to build images with BuildKit: Build an Image		
✖ https://docs.docker.com/go/buildx/ Build an Image		
✖ No message found for this issue Build an Image		∅
✖ failed to parse platform : "" is an invalid component of "": platform specifier component must match "^[A-Za-z0-9_-]+\$": invalid argument Build an Image		
✖ The process '/usr/bin/docker' failed with exit code 1 Build an Image		

Do these two changes: Commit these changes and pipeline should auto trigger.



```

9  # build compiles the program for the builder's local platform
10 FROM --platform=linux mcr.microsoft.com/dotnet/sdk:7.0 AS build
11 ARG TARGETPLATFORM
12 ARG TARGETARCH
13 ARG BUILDPLATFORM
14 ARG echo "I am running on $BUILDPLATFORM, building for $TARGETPLATFORM"
15
16 WORKDIR /source
17 COPY *.csproj .
18 RUN dotnet restore -a $TARGETARCH
19
20 COPY . .
21 RUN dotnet publish -c release -o /app --self-contained false --no-restore
22
23
24 # app image
25 FROM mcr.microsoft.com/dotnet/runtime:7.0
26 WORKDIR /app
27 COPY --from=build /app .
28 ENTRYPOINT ["dotnet", "Worker.dll"]

```

Pipelines

Recent All Runs

Filter pipelines

Recently run pipelines

Pipeline	Last run
Worker-service	#20241027.2 • Updated Dockerfile Individual CI for main 17617f7b Just now 18s
Vote-service	#20241027.1 • Set up CI with Azure Pipelines Individual CI for main 15m ago 1m 54s
Result-service	#20241027.4 • Updated server.js Individual CI for main 28m ago 1m 22s
#20241027.2 • Updated Dockerfile	Individual CI for main 17617f7b 8m ago 1m 7s
#20241027.1 • Set up CI with Azure Pipelines	Individual CI for main 54467ed9 12m ago 38s

Failed: Remove everything after restore in line 19. Commit changes & pipeline triggers automatically.

Worker-service

```

11 FROM --platform=linux mcr.microsoft.com/vulnerabilities/b11d0444 AS BUILDDOCKER
12 ARG TARGETPLATFORM
13 ARG TARGETARCH
14 ARG BUILDPLATFORM
15 RUN echo "I am running on $BUILDPLATFORM, building for $TARGETPLATFORM"
16
17 WORKDIR /source
18 COPY *.csproj .
19 RUN dotnet restore
20

```

Runs **Branches** **Analytics**

Description	Stages
#20241027.3 • Updated Dockerfile Individual CI for main d02f5f15	Just now 1m 14s
#20241027.2 • Updated Dockerfile Individual CI for main 17617f7b	10m ago 1m 7s

Home > Container registries > votingappbuild

Container registries

votingappbuild | Repositories

Container registry

Search Refresh Manage Deleted Repositories

New to ACR. Artifact streaming helps pull images faster from AKS clusters. The 'Artifact streaming status' column shows which repositories are using this feature. Learn more

Repositories resultapp votingapp

votingappbuild | Repositories

Container registry

Search Refresh

New to ACR. Artifact streaming helps pull images faster from AKS clusters. The 'Artifact streaming status' column shows which repositories are using this feature. Learn more

Repositories resultapp votingapp

We had two repositories in the container registry for two frontend services. We can see the images of those services.

Home > Container registries > votingappbuild | Repositories

votingappbuild | Repositories

Container registry

Search Refresh

New to ACR. Artifact streaming helps pull images faster from AKS clusters. The 'Artifact streaming status' column shows which repositories are using this feature. Learn more

Repositories resultapp votingapp

resultapp

Repository resultapp Tag count 2 Manifest count 2

Last updated date 28/10/2024, 02:42 GMT+5:30

Tags Digest Last modified

5 sha256:85c5de898f79b109b05505e9b15db99587e... 28/10/2024, 02:42 GMT+5:30

4 sha256:b6c73631f465a18d6398bce3af178885094f8... 28/10/2024, 02:38 GMT+5:30

The screenshot shows the Azure Container Registry (ACR) interface. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Quick start, Events, Settings, and Services. The 'Repositories' option is selected. The main area shows a repository named 'votingapp' with a single tag 'votingapp'. A tooltip provides information about the 'Artifact streaming' feature, stating it helps put images faster from AKS clusters. The repository details show 'Last updated date' as 28/10/2024, 02:55 GMT+5:30, 'Tag count' as 1, and 'Manifest count' as 1.

Day 15:

Create a Azure Kubernetes service:

Go to Kubernetes service and click on create → Kubernetes cluster

This screenshot shows the 'Create Kubernetes cluster' wizard in the Azure portal. It's step 1: Cluster details. The 'Subscription' dropdown is set to 'Free Trial'. The 'Resource group' dropdown is set to 'voting-app'. Under 'Cluster details', the 'Cluster preset configuration' is set to 'Dev/Test'. The 'Kubernetes cluster name' is 'DevSit'. The 'Region' is '(US) East US'. 'Availability zones' is 'Zones 1'. 'AKS pricing tier' is 'Free'. 'Kubernetes version' is '1.29.9 (default)'. Under 'Automatic upgrade', it's set to 'Enabled with patch (recommended)' with a schedule of 'Every week on Sunday (recommended)'. The 'Start on' field shows 'Tue Oct 29 2024 00:00 +00:00 (Coordinated Universal Time)'. Under 'Node security channel type', it's 'Node Image' with a schedule of 'Every week on Sunday (recommended)'. The 'Start on' field shows 'Tue Oct 29 2024 00:00 +00:00 (Coordinated Universal Time)'.

We need agent pool to our AKS cluster. AKS is services managed by Azure but the pods that we have to run need nodes.

[Basics](#) [Node pools](#) [Networking](#) [Integrations](#) [Monitoring](#) [Advanced](#) [Tags](#) [Review + create](#)

Node pools

In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads [Learn more](#)

<input type="checkbox"/>	Name	Mode	Node size	OS SKU	Node count	Available
<input type="checkbox"/>	agentpool	System	Standard_DS2_v2 (...)	Ubuntu	2 - 5	None

Click on agentpool and modify configurations:

This screenshot shows the 'Edit node pool' configuration page for the 'agentpool' node pool. Under 'Scale method', the 'Autoscale - Recommended' option is selected. The 'Minimum node count' is set to 1. The 'Maximum node count' is set to 2, with a note stating 'The maximum node count allowed for an AKS cluster is 1000 per node pool and 5000 nodes across all node pools in this cluster.' Under 'Optional settings', the 'Max pods per node' is set to 30, with a range of 30 - 250. The 'Enable public IP per node' checkbox is checked.

We don't need much nodes for this project and 30 pods per node is also fine. Also enable the public IP per node. Then click on Update.

Select the agent pool:

Node pools

In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads [Learn more](#).

	Name	Mode	Node size	OS SKU	Node count	Availability
<input checked="" type="checkbox"/>	agentpool	System	Standard_DS2_v2 (Ubuntu)		1 - 2	None

We can also setup multiple agent pools. Then click on Review + create: Then click on create.

Faced VM size unavailability in us east region changed to us west region.

Home > microsoft.aks-1730092572047 | Overview >

DevSit Kubernetes service

Search < > + Create & Connect > Start & Stop & Delete & Refresh & Open in mobile & Give feedback

Overview

Essentials

Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Cost analysis, Kubernetes resources.

Resource group: voting-app, Power state: Running, Cluster operation status: Succeeded, Subscription: Free Trial, Location: West US 2, Subscription ID: 00681e52-1746-42b2-b0a5-ad83a3b482af.

Kubernetes version: 1.29.9, API server address: devsit-dns-j3nmxbeu.hcp.westus2.azmk8s.io, Network configuration: Azure CNI Overlay, Node pools: 1 node pool, Container registries: Attach a registry.

Tags (edit) : Add tags

JSON View

Connect to this cluster from terminal or Cloud shell:

Azure CLI and Kubectl to be installed in local PC if want to connect from local terminal.

We can connect through cloushell:

Home >

DevSit Kubernetes service

Search < > + Create & Connect > Start & Stop & Delete & Refresh & Open in mobile & Give feedback

Overview

Essentials

Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Cost analysis, Kubernetes resources.

Resource group: voting-app, Power state: Running, Cluster operation status: Succeeded, Subscription: Free Trial, Location: West US 2, Subscription ID: 00681e52-1746-42b2-b0a5-ad83a3b482af.

Kubernetes version: 1.29.9, API server address: devsit-dns-j3nmxbeu.hcp.westus2.azmk8s.io, Network configuration: Azure CNI Overlay, Node pools: 1 node pool, Container registries: Attach a registry.

Tags (edit) : Add tags

Get started Properties Monitoring Capabilities (5) Recommendations (0) Tutorials

Connect to DevSit

Cloud shell Azure CLI Run command

Connect to your cluster using command line tooling to interact directly with cluster using kubectl, the command line tool for Kubernetes. Kubectl is available within the Azure Cloud Shell by default and can also be installed locally.

Set cluster context

- 1 Open Cloud Shell
- 2 Run the following commands

Set the cluster subscription
az account set --subscription 00681e52-1746-42b2-b0a5-ad83a3b482af

Download cluster credentials
az aks get-credentials --resource-group voting-app --name DevSit --overwrite-existing

Connecting to AKS cluster from terminal:

```
yvaru@Tej MINGW64 ~
$ az aks get-credentials --resource-group voting-app --name DevSit --overwrite-existing
Merged "DevSit" as current context in C:\Users\yvaru\.kube\config

yvaru@Tej MINGW64 ~
$ kubectl version
Client Version: v1.31.2
Kustomize Version: v5.4.2
Server Version: v1.29.9
WARNING: version difference between client (1.31) and server (1.29) exceeds the supported minor version skew of +/-1

yvaru@Tej MINGW64 ~
$ kubectl get pods
No resources found in default namespace.

yvaru@Tej MINGW64 ~
$
```

Now, we need to install Argo CD in the cluster:

Documentation: https://argo-cd.readthedocs.io/en/stable/getting_started/

Installing Argo CD in argocd namespace where all Argo CD services and application resources will be live.

```
yvaru@Tej MINGW64 ~
$ kubectl create namespace argocd
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
namespace/argocd created
customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/applicationsets.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/approjects.argoproj.io created
serviceaccount/argocd-application-controller created
serviceaccount/argocd-applicationset-controller created
```

Lot of argocd resources are created in the namespace.

```
yvaru@Tej MINGW64 ~
$ kubectl get pods -n argoocd
NAME                               READY   STATUS    RESTARTS   AGE
argoocd-application-controller-0   1/1     Running   0          3m4s
argoocd-applicationset-controller-5b899f5459-h8mjf   1/1     Running   0          3m10s
argoocd-dex-server-7f4f5476db-vwknn   1/1     Running   0          3m9s
argoocd-notifications-controller-75d8587699-4qbks   1/1     Running   0          3m8s
argoocd-redis-5bc565bfc4-f2dkb      1/1     Running   0          3m8s
argoocd-repo-server-55c7764fd4-hj4vm   1/1     Running   0          3m7s
argoocd-server-5f6567647b-z2gsh     1/1     Running   0          3m6s
```

Argo Cd in the cluster need to check the changes in K8s manifets files in Git Repo. So, we need to configure that. But before that we need to expose argoocd server UI. And need to know the admin password of argoocd.

```
yvaru@Tej MINGW64 ~
$ kubectl get secrets -n argoocd
NAME              TYPE        DATA   AGE
argoocd-initial-admin-secret  Opaque     1      5m21s
argoocd-notifications-secret Opaque     0      5m49s
argoocd-redis      Opaque     1      5m23s
argoocd-secret     Opaque     5      5m48s

yvaru@Tej MINGW64 ~
$ kubectl edit secret argoocd-initial-admin-secret -n argoocd
Edit cancelled, no changes made.

# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  password: MG9SNUNsc3FBb1J4LUVNUw==
kind: Secret
metadata:
  creationTimestamp: "2024-10-28T08:35:40Z"
  name: argoocd-initial-admin-secret
  namespace: argoocd
  resourceVersion: "46725"
  uid: 29e9ea7e-b85d-4ac2-bdb9-b69b6e9f4673
type: Opaque
```

Copy it. And decode it.

```
yvaru@Tej MINGW64 ~
$ echo MG9SNUNsc3FBb1J4LUVNUw== | base64 --decode
OoR5ClsqAoRx-EMS
yvaru@Tej MINGW64 ~
$
```

Copy this password. Now lets expose the argoocd server to NodePort so that we can access in our browser. OoR5ClsqAoRx-EMS

```
yvaru@Tej MINGW64 ~
$ kubectl get svc -n argoocd
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
argoocd-applicationset-controller   ClusterIP  10.0.139.152 <none>        7000/TCP,8080/TCP   9m36s
argoocd-dex-server                 ClusterIP  10.0.147.22  <none>        5556/TCP,5557/TCP,5558/TCP   9m35s
argoocd-metrics                   ClusterIP  10.0.182.46  <none>        8082/TCP          9m34s
argoocd-notifications-controller-metrics   ClusterIP  10.0.220.223 <none>        9001/TCP          9m33s
argoocd-redis                      ClusterIP  10.0.128.111 <none>        6379/TCP          9m32s
argoocd-repo-server                ClusterIP  10.0.226.184 <none>        8081/TCP,8084/TCP   9m31s
argoocd-server                     ClusterIP  10.0.227.106 <none>        80/TCP,443/TCP    9m30s
argoocd-server-metrics             ClusterIP  10.0.162.0   <none>        8083/TCP          9m29s

yvaru@Tej MINGW64 ~
$ kubectl edit svc argoocd-server -n argoocd

protocol: TCP
targetPort: 8080
selector:
  app.kubernetes.io/name: argoocd-server
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}

yvaru@Tej MINGW64 ~
$ kubectl get svc -n argoocd
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
argoocd-applicationset-controller   ClusterIP  10.0.139.152 <none>        7000/TCP,8080/TCP   11m
argoocd-dex-server                 ClusterIP  10.0.147.22  <none>        5556/TCP,5557/TCP,5558/TCP   11m
argoocd-metrics                   ClusterIP  10.0.182.46  <none>        8082/TCP          11m
argoocd-notifications-controller-metrics   ClusterIP  10.0.220.223 <none>        9001/TCP          11m
argoocd-redis                      ClusterIP  10.0.128.111 <none>        6379/TCP          11m
argoocd-repo-server                ClusterIP  10.0.226.184 <none>        8081/TCP,8084/TCP   11m
argoocd-server                     NodePort   10.0.227.106 <none>        80:31336/TCP,443:32317/TCP   11m
argoocd-server-metrics             ClusterIP  10.0.162.0   <none>        8083/TCP          11m
```

So, now we need to know the node IP and port. Port we already knew 31336 from above pic.

```
yvaru@Tej MINGW64 ~
$ kubectl get nodes -o wide
NAME               STATUS   ROLES   AGE    VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE          KERNEL-VERSION   CONTAINER-RUNTIME
aks-agentpool-35213278-vmss000000   Ready    <none>   3h29m   v1.29.9   10.224.0.4   51.143.91.198   Ubuntu 22.04.5 LTS   5.15.0-1073-azure   containerd://1.7.22-1
```

Copy the External IP: 51.143.91.198: 31336 but we will not be able to access it. We need to open the ports for the AKS cluster node.

Go to azure portal and search for virtual machine scale sets:

Virtual machine scale sets																													
Default Directory																													
+ Create Manage view Refresh Export to CSV Open query Assign tags Start Restart Stop Delete Maintenance Leave preview																													
<input type="button" value="Filter for any field..."/> <input type="button" value="Subscription equals all"/> <input type="button" value="Resource group equals all"/> <input type="button" value="Location equals all"/> <input type="button" value="Add filter"/>																													
Showing 1 to 1 of 1 records.																													
<table border="1"> <thead> <tr> <th>Name</th><th>Subscription</th><th>Resource group</th><th>Location</th><th>Provisioning state</th><th>Status</th><th>Operating system</th><th>Size</th><th>Instances</th><th>Orchestration mode</th></tr> </thead> <tbody> <tr> <td><input type="checkbox"/> aks-agentpool-35213278-vmss</td><td>Free Trial</td><td>MC_voting-app_DevS1</td><td>West US 2</td><td>Succeeded</td><td>All succeeded</td><td>Linux</td><td>Standard_DS2_v2</td><td>1</td><td>Uniform</td></tr> </tbody> </table>										Name	Subscription	Resource group	Location	Provisioning state	Status	Operating system	Size	Instances	Orchestration mode	<input type="checkbox"/> aks-agentpool-35213278-vmss	Free Trial	MC_voting-app_DevS1	West US 2	Succeeded	All succeeded	Linux	Standard_DS2_v2	1	Uniform
Name	Subscription	Resource group	Location	Provisioning state	Status	Operating system	Size	Instances	Orchestration mode																				
<input type="checkbox"/> aks-agentpool-35213278-vmss	Free Trial	MC_voting-app_DevS1	West US 2	Succeeded	All succeeded	Linux	Standard_DS2_v2	1	Uniform																				

Click on Instances

Home > Virtual machine scale sets > aks-agentpool-35213278-vmss

Virtual machine scale sets

aks-agentpool-35213278-vmss | Instances

Virtual machine scale set

Search Start Restart Stop Hibernate Reimage Delete Upgrade Refresh Protection

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Instances

Search virtual machine instances

Instance	Computer name	Status	Protection policy	Provisioning state	Health state	Latest model
aks-agentpool-3521...	aks-agentpool-3521...	Running	Succeeded	Succeeded	Yes	

Now click on aks-agentpool: Click on network and open the port in inbound port rules:

Networking

Network settings Load balancing Application security groups Network manager

Private IP address: 10.224.0.4 Admin security rules: 0 (Configure)

Accelerated networking: Enabled Effective security rules: 0

Network security group aks-agentpool-12995614-nsg (attached to subnet: aks-subnet)

Impacts 1 subnets, 0 network interfaces

Inbound port rules (3)

Priority ↑	Name	Port	Protocol	Source	Destination	Action
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Outbound port rules (3)

Just update Port number and click on Add.

Networking

Network settings Load balancing Application security groups Network manager

Private IP address: 10.224.0.4 Admin security rules: 0 (Configure)

Accelerated networking: Enabled Effective security rules: 0

Network security group aks-agentpool-12995614-nsg (attached to subnet: aks-subnet)

Impacts 1 subnets, 0 network interfaces

Inbound port rules (3)

Priority ↑	Name	Port	Protocol	Source	Destination	Action
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Add port rule

Inbound port rule

Outbound port rule

Destination: Any

Service: Custom

Destination port ranges: 31336

Protocol: Any

Now give the Node IP and port number: Give admin as user name and password of it.

ArgoCD

Applications

+ NEW APP SYNC APPS REFRESH APPS Search applications...

APPLICATIONS TILES

Log out

Argo v2.12.6+4dab5bd

Applications

Settings User Info Documentation

This ArgoCD can only read the changes in repos. So, we can create PAT that had read only permission while connecting this to repos.

Now, create a PAT from azure DevOps and copy it.

Now in the ArgoCD click on Setting → Repositories → Connect Repo

Azure DevOps varuntej06 / Voting-App / Repos / Files / Voting-App

Voting-App Overview Boards

main Type to find a file or folder...

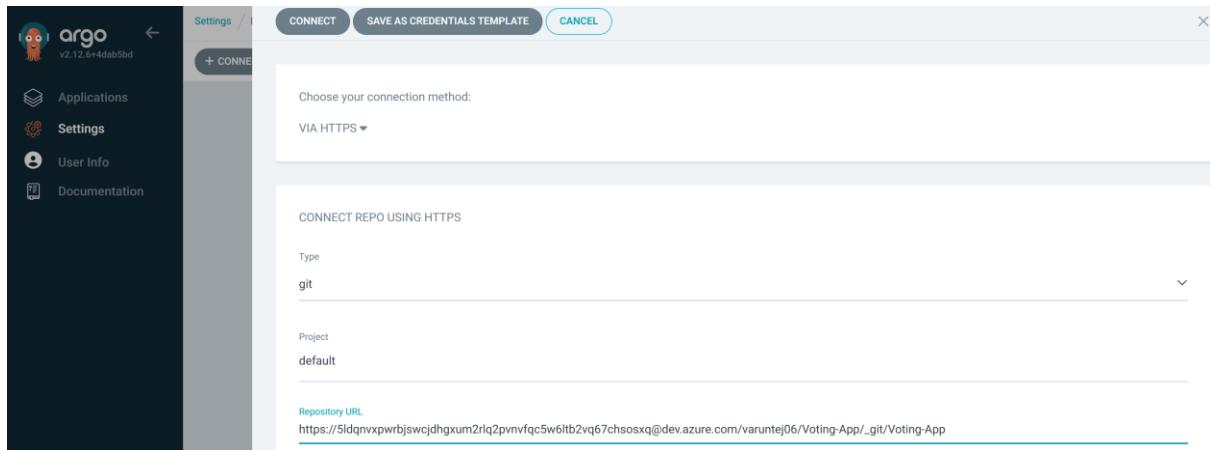
Files Contents History

Clone Repository

Command line

HTTPS SSH https://varuntej06@dev.azure.com/varun/ Copied to clipboard!

Generate Git Credentials



Add these details and in the repo URL you need to replace organisation name with access token.

Type	Name	Project	Repository	Connection Status
git	default		https://5ldqnvxpwrbswcjdhgxum2rlq2pvnvfqc5w6ltb2vq67chsosxq@dev.azure.com/varuntej06/Voting-App/_git/Voting-App	Successful

Now, we need to deploy the manifests to Cluster:

Go to applications → New Application → Application should be lowercase.

The screenshot shows the Argo UI interface. On the left, there's a sidebar with navigation links: Applications, Settings, User Info, Documentation, Favorites Only, SYNC STATUS, and HEALTH STATUS. The main area displays the 'votingapp' application details, including its project (default), status (Progressing, Synced), repository (https://5ldqnxpwrblsjwcdhgxum2rlq2pvn...), target ref (HEAD), path (k8s-specifications), destination (in-cluster), namespace (default), creation time (10/28/2024 14:42:28), and last sync time (10/28/2024 14:42:32). Below this are buttons for SYNC, REFRESH, and DELETE. To the right, the 'APPLICATION DETAILS TREE' section shows a hierarchical tree of deployed components, with a tooltip for 'Open Merlin'.

It deployed everything:

```
yvaru@Tej MINGW64 ~
$ kubectl get pods -n default
NAME          READY   STATUS    RESTARTS   AGE
db-6d9f87bb-8m2gv   1/1   Running   0          3m38s
redis-77fccb7f9-r25xt   1/1   Running   0          3m38s
result-54b5ccfc95-7xbwz   1/1   Running   0          3m38s
vote-5655bd759-vx6ps   1/1   Running   0          3m38s
worker-7dd74bcbbb-wfv8n   1/1   Running   0          3m38s
```

It deployed the application, but it should also trigger deployments on changing something in repo.
We need to implement CD now.

We will add a folder in the repo root level and add a script file which will update the manifest files on successful build.

Script: <https://github.com/VarunTej06/sampleScripts/blob/main/script.sh>

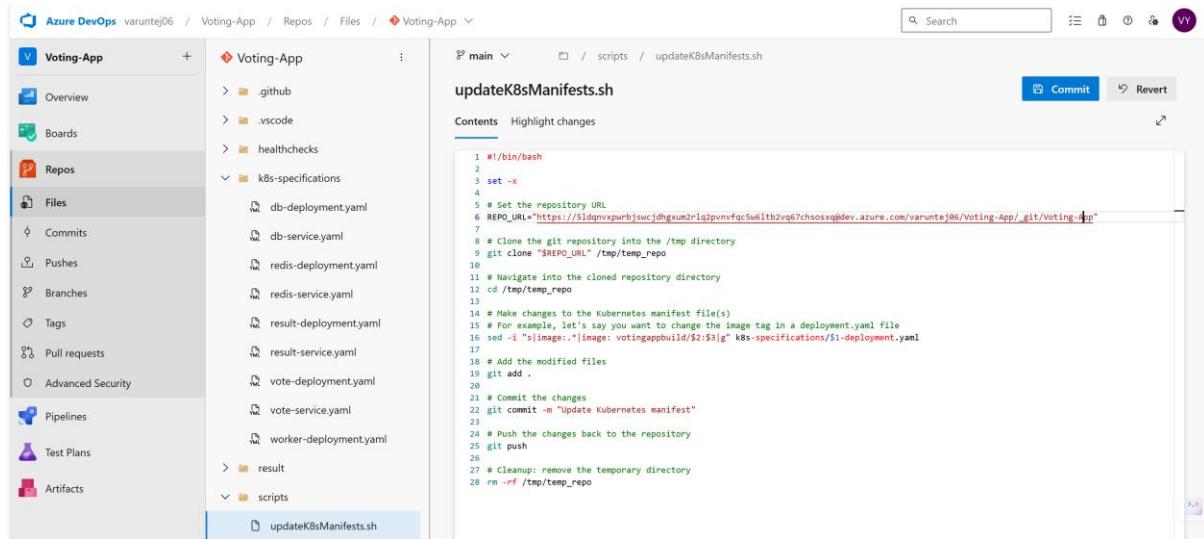
The screenshot shows the Azure DevOps repository interface. The left sidebar includes sections for Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, Pipelines, Test Plans, and Artifacts. The 'Repos' section is currently selected. In the center, the 'Voting-App' repository is shown, with the 'main' branch selected. A file named 'updateK8sManifests.sh' is open in the editor. The script content is as follows:

```
#!/bin/bash
set -x
# Set the repository URL
REPO_URL="https://@dev.azure.com//voting-app/_git/voting-app"
# Clone the git repository into the /tmp directory
git clone "$REPO_URL" /tmp/temp_repo
# Navigate into the cloned repository directory
cd /tmp/temp_repo
# Make changes to the Kubernetes manifest file(s)
# For example, let's say you want to change the image tag in a deployment.yaml file
sed -i '$image::$2:$3' 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
```

Replace the access token in line 6. Use the same one that we used for argocd. Since we granted all

permissions to it. You can copy it from repositories from argocd where we added. Also change the organisation name in the same line. Also update Project name properly at 2 places in the same line.

Now update the container registry name in line 16.



```

#!/bin/bash
set -x
# Set the repository URL
REPO_URL="https://$1dgnvxpurbjswcjhgxum2rlq2pvnfqc5w6lth2vg67chosoqgdev.azure.com/varunaje06/Voting-App/_git/Voting-App"
# Clone the git repository into the /tmp directory
git clone "$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: votingappbuild.azurecr.io/$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

```

\$1, \$2, \$3 are command line arguments passed to this script file from CI pipeline. Commit the file.

\$1 is deployment file name, \$2 is container registry repository name and \$3 is tag name.

Lets try to Edit pipelines and add stage to trigger this script file from a new stage called update.

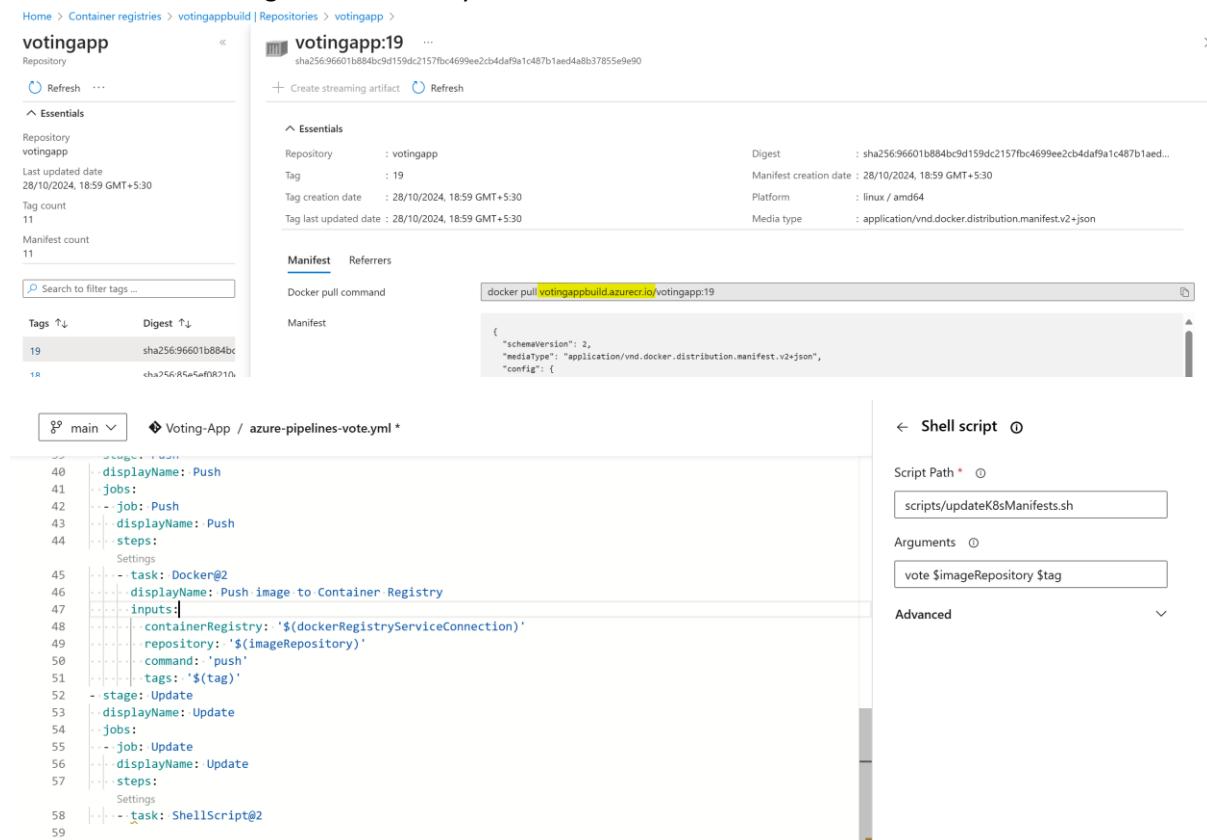
Note: This point added at the end of the project when faced an issue. Here we mentioned image name wrongly that needs to be updated.

```

13 # Make changes to the Kubernetes manifest file(s)
14 # For example, let's say you want to change the image tag in a deployment.yaml file
15 sed -i "s|image:.*|image: votingappbuild.azurecr.io/$2:$3|g" k8s-specifications/$1-deployment.yaml
16
17 # Add the modified files

```

To know how we can give this correctly:



Home > Container registries > votingappbuild | Repositories > votingapp >

votingapp

Repository

Refresh ...

Essentials

Repository: votingapp

Last updated date: 28/10/2024, 18:59 GMT+5:30

Tag count: 11

Manifest count: 11

Search to filter tags ...

Tags ↑↓	Digest ↑↓
19	sha256:96601b884bc9d159dc2157fbca699ee2cb4da9a1c487b1aed4a8b37855e9e0
18	sha256:85a5ef0f821fb

votingapp:19

sha256:96601b884bc9d159dc2157fbca699ee2cb4da9a1c487b1aed4a8b37855e9e0

+ Create streaming artifact Refresh

Essentials

Repository: votingapp

Tag: 19

Tag creation date: 28/10/2024, 18:59 GMT+5:30

Tag last updated date: 28/10/2024, 18:59 GMT+5:30

Docker pull command: docker pull votingappbuild.azurecr.io/votingapp:19

Manifest

```
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
  "config": {
    ...
  }
}
```

azure-pipelines-vote.yml

```

40   - displayName: Push
41     jobs:
42       - job: Push
43         displayName: Push
44         steps:
45           - task: Docker@2
46             displayName: Push image to Container Registry
47             inputs:
48               containerRegistry: '$(dockerRegistryServiceConnection)'
49               repository: '$(imageRepository)'
50               command: 'push'
51               tags: '$(tag)'
52       - stage: Update
53         displayName: Update
54         jobs:
55           - job: Update
56             displayName: Update
57             steps:
58               - task: ShellScript@2

```

← Shell script

Script Path *: scripts/updateK8sManifests.sh

Arguments: vote \$imageRepository \$tag

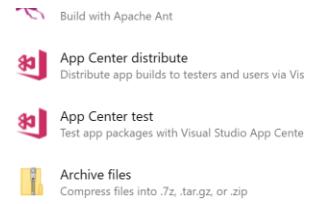
Advanced

Passing the arguments here.

```

52   - stage: Update
53     displayName: Update
54   - jobs:
55     - job: Update
56       displayName: Update
57     - steps:
58       - task: ShellScript@2
59       - inputs:
60         - scriptPath: 'scripts/updateK8sManifests.sh'
61         - args: 'vote ${imageRepository} ${tag}'

```



Click on validate and save. When you run the pipeline it will update the line:

The screenshot shows the Azure Pipelines interface for the 'Update azure-pipelines-vote.yml' pipeline. At the top, there's a code editor window with the pipeline YAML file. Below it is a summary card for the latest run (#20241028.1). The summary includes details like repository version (main), start time (Just now), duration (10s), and artifact count (0). Below the summary is a 'Stages' section showing three stages: 'Build' (Not started), 'Push' (Not started), and 'Update' (Not started).

New stage is added and it should update the image name properly after the build push and update is success in the pipeline.

The screenshot shows the 'Vote-service' pipeline details. It lists a recent run (#20241028.2) which was manually triggered and completed successfully ('Just now'). Below the pipeline details is a 'Container registries' section for 'votingappbuild'. It shows a list of repositories: 'resultapp', 'votingapp', and 'votingapp'. The 'votingapp' repository is selected, showing its details: repository name 'votingapp', last updated date '28/10/2024, 17:29 GMT+5:30', tag count '9', manifest count '9', and a list of tags with their digests and last modified times.

The latest image is 17. It should be updated in the vote-deployment.yaml

The screenshot shows the 'vote-deployment.yaml' file content. The file defines a deployment with a single container named 'vote' using image 'votingappbuild/votingapp:17'. The file also includes 'result-deployment.yaml', 'result-service.yaml', and 'vote-service.yaml'.

Updated. So, everytime this vote pipeline is triggered, it will automatically update the image version.

Usually argocd takes 3 mins time to sync. To get the changes done in manifest file. We can reduce it as well to 10 seconds.

```

yvaru@Tej MINGW64 ~
$ kubectl edit cm argocd-cm -n argocd
configmap/argocd-cm edited

```

Add last two lines (data:.....) at the end of the file:

```

resourceVersion: "46410"
uid: c4a54732-dbc7-458e-8253-f78a03df8f38
data:
  timeout.reconciliation: 10s

```

Vote Pod went into ImagePullBackOff issue:

```
yvaru@Tej MINGW64 ~
$ kubectl get pods -n default
NAME           READY   STATUS    RESTARTS   AGE
db-6d9f87bb0b-8m2gv   1/1    Running   0          3h13m
redis-77fc7bf9-r25xt  1/1    Running   0          3h13m
result-54b5ccfc95-7xbwz  1/1    Running   0          3h13m
vote-5655bd759-vx6ps  1/1    Running   0          3h13m
vote-74bf8459c-bgvjc  0/1    ImagePullBackOff 0          20m
worker-7dd74bcbb-wfv8n 1/1    Running   0          3h13m

Events:
  Type     Reason     Age                   From               Message
  Normal   Scheduled  21m      default-scheduler  Successfully assigned default/vote-74bf8459c-bgvjc to aks-agentpool-35213278-vmss000000
  Normal   Pulling    19m (x4 over 21m)  kubelet            Pulling image "votingappbuild/votingapp:17"
  Warning  Failed    19m (x4 over 21m)  kubelet            Failed to pull image "votingappbuild/votingapp:17": failed to pull and unpack image "docker.io/votingappbuild/votingapp:17": failed to resolve reference "docker.io/votingappbuild/votingapp:17": failed to pull and unpack image "docker.io/votingappbuild/votingapp:17": pull access denied, repository does not exist or may require authorization; server message: insufficient_scope: authorization failed
  Warning  Failed    19m (x4 over 21m)  kubelet            Error: ErrImagePull
  Warning  Failed    19m (x6 over 21m)  kubelet            Error: ImagePullBackOff
  Normal   BackOff   77s (x87 over 21m)  kubelet            Back-off pulling image "votingappbuild/votingapp:17"
```

```
yvaru@Tej MINGW64 ~
$ kubectl edit pod vote-74bf8459c-bgvjc
Edit cancelled, no changes made.
```

Image updated in Cluster level also:

```
resourceVersion: "101748"
uid: a2663413-bed5-48ba-a01-1d8bfb703f20
spec:
  containers:
    - image: votingappbuild/votingapp:17
      imagePullPolicy: Always
      name: vote
```

Reason	Message	Count	First Occurred	Last Occurred
BackOff	Back-off pulling image 'votingappbuild/votingapp:17'	132	33m ago Today at 5:35 PM	3m ago Today at 6:05 PM
Failed	Error: ImagePullBackOff	6	33m ago Today at 5:35 PM	31m ago Today at 5:37 PM
Pulling	Pulling image 'votingappbuild/votingapp:17'	4	33m ago Today at 5:35 PM	31m ago Today at 5:37 PM

We are trying to pull the image from a private repository from azure container registry: We need to use a concept called ImagePullSecrets inorder to pull image from private repositories or private registries. Go to Azure portal and container registries:

Click on Access keys:

Enable Admin User:

Then give this command in the cluster:

Command to create ACR ImagePullSecret

```
kubectl create secret docker-registry <secret-name> \
--namespace <namespace> \
--docker-server=<container-registry-name>.azurecr.io \
--docker-username=<service-principal-ID> \
--docker-password=<service-principal-password>
```

secret name: any; namespace: default; principal-ID: Username from access keys; principal password: password from access keys.

```
yvaru@Tej MINGW64 ~
$ kubectl create secret docker-registry acr-secret --namespace default --docker-server=votingappbuild.azurecr.io --docker-username=votingappbuild --docker-password=RhxhiCyy+SLPzhXrbIM8j43FkK4ai2guUTXxF50HM+ACRcfj0tK
secret/acr-secret created
yvaru@Tej MINGW64 ~
```

Now, we will go to deployment.yaml and add these lines:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: vote
    name: vote
spec:
  replicas: 1
  selector:
    matchLabels:
      app: vote
  template:
    metadata:
      labels:
        app: vote
    spec:
      containers:
        - image: votingappbuild/votingapp:17
          name: vote
          ports:
            - containerPort: 80
              name: vote
      imagePullSecrets:
        - name: acr-secret
```

Now, we will change the some lines of code in Vote directory so that vote pipeline triggers automatically. I updated dogs and cats to Docker and Kubernetes in app.py in Vote directory.

```
import os
import socket
import random
import json
import logging
option_a = os.getenv('OPTION_A', "Docker")
option_b = os.getenv('OPTION_B', "Kubernetes")
hostname = socket.gethostname()
app = Flask(__name__)
```

Once committed, pipeline should auto trigger.

Description	Stages	Created	Last Run
#20241028.9 • Updated app.py Individual CI for main	✓ - ✓ - ✓	3m ago	Just now 3s

Image updated:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: vote
spec:
  containers:
    - image: votingappbuild/votingapp:18
      name: vote
```

Bt still the pod is failing:

```
yvaru@Tej MINGW64 ~
$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
db-6d9f87bb9b-8m2gy   1/1    Running   0          3h52m
redis-77fcbb7f9-r25xt 1/1    Running   0          3h52m
result-54b5ccfc95-7xbwz 1/1    Running   0          3h52m
vote-5655bd759-vx6ps  1/1    Running   0          3h52m
vote-6f4675d9f7-vnm2l  0/1    ErrImagePull 0          38s
worker-7dd74bcbbb-wfv8n 1/1    Running   0          3h52m
Events:          node.kubernetes.io/unreachable:Not execute opExists for 300s
Events:          Type Reason     Age From             Message
Normal  Scheduled  55s  default-scheduler  Successfully assigned default/vote-6f4675d9f7-vnm2l to aks-agentpool-35213278-vmss000000
Normal  Pulling    13s (x3 over 55s) kubelet  Pulling image "votingappbuild/votingapp:18"
Warning Failed    13s (x3 over 54s) kubelet  Failed to pull image "votingappbuild/votingapp:18"; failed to pull and unpack image "docker.io/votingappbuild/votingapp:18": failed to resolve reference "docker.io/votingappbuild/votingapp:18": pull access denied, repository does not exist or may require authentication; server message: insufficient_scope: authorization failed
Warning Failed    13s (x3 over 54s) kubelet  Error: ErrImagePull
Normal  BackOff    1s (x3 over 53s)  kubelet  BackOff pulling image "votingappbuild/votingapp:18"
Warning Failed    1s (x3 over 53s)  kubelet  Error: ImagePullBackOff
```

REASON	MESSAGE	COUNT	FIRST OCCURRED	LAST OCCURRED
BackOff	Back-off pulling image "votingappbuild/votingapp:18"	7	5m ago Today at 6:33 PM	3m ago Today at 6:35 PM
Failed	Error: ImagePullBackOff	6	5m ago Today at 6:33 PM	4m ago Today at 6:35 PM
Pulling	Pulling image 'votingappbuild/votingapp:18'	4	5m ago Today at 6:33 PM	4m ago Today at 6:35 PM
Failed	Failed to pull image "votingappbuild/votingapp:18": failed to pull and unpack image "docker.io/votingappbuild/votingapp:18": failed to resolve reference "docker.io/votingappbuild/votingapp:18": pull access denied, repository does not exist or may require authorization: server message: insufficient_scope: authorization failed	4	5m ago Today at 6:33 PM	4m ago Today at 6:35 PM

Let's check both pod and deployment got secret added or not.

```
yvaru@Tej MINGW64 ~
$ kubectl edit pod vote-6f4675d9f7-vnm2l

spec:
  containers:
    - image: votingappbuild/votingapp:18
      imagePullPolicy: Always
      name: vote
      ports:
        - containerPort: 80
          name: vote
          protocol: TCP
      resources: {}
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
    volumeMounts:
      - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
        name: kube-api-access-6rqhm
        readOnly: true
  dnsPolicy: ClusterFirst
  enableServiceLinks: true
  imagePullSecrets:
    - name: acr-secret
  nodeName: aks-agentpool-35213278-vmss000000

yvaru@Tej MINGW64 ~
$ kubectl get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
db         1/1     1           1           4h1m
redis     1/1     1           1           4h1m
result    1/1     1           1           4h1m
vote       1/1     1           1           4h1m
worker    1/1     1           1           4h1m

yvaru@Tej MINGW64 ~
$ kubectl edit deploy vote

# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "4"
    kubectl.kubernetes.io/last-applied-configuration: |
{"apiVersion":"apps/v1","kind":"Deployment","metadata":{"annotations":{},"labels":{"app":"vote","app.kubernetes.io/instance":"votingapp"},"name":"vote","namespace":"default"},"spec":{"replicas":1,"selector":{"matchLabels":{"app":"vote","template":{"metadata":{"labels":{"app":"vote}}}},"spec":{"containers":[{"image":"votingappbuild/votingapp:18","name":"vote","ports":[{"containerPort:80}]}]},"strategy":{"type":"Recreate"}, "terminationGracePeriodSeconds": 30}, "status": {"availableReplicas": 1, "conditions": [{"lastTransitionTime": "2024-10-28T09:12:22Z", "status": "True", "type": "Available"}, {"lastTransitionTime": "2024-10-28T09:12:22Z", "status": "True", "type": "Progressing"}]}, "version": 4}, "labels": {"app": "vote", "app.kubernetes.io/instance": "votingapp"}
```

Secret is also created:

```
yvaru@Tej MINGW64 ~
$ kubectl get secret
NAME        TYPE           DATA   AGE
acr-secret  kubernetes.io/dockerconfigjson  1      24m
```

Still in pod is having issues to pull image:

```
yvaru@Tej MINGW64 ~
$ kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
db-6d9f87bb9-8m2gv   1/1     Running   0          4h7m
redis-77fccb7f9-r25xt 1/1     Running   0          4h7m
result-54b5cef95-7xbwz 1/1     Running   0          4h7m
vote-5655bd759-vx6ps  1/1     Running   0          4h7m
vote-6f4675d9f7-jkqgz 0/1     ErrImagePull 0          12s
worker-7dd74beb8b-wfv8n 1/1     Running   0          4h7m
vote-6f4675d9f7-jkqgz  0/1    ImagePullBackOff 0          15s
vote-6f4675d9f7-jkqgz  0/1    ErrImagePull 0          26s
vote-6f4675d9f7-jkqgz  0/1    ImagePullBackOff 0          40s
vote-6f4675d9f7-jkqgz  0/1    ErrImagePull 0          56s
vote-6f4675d9f7-jkqgz  0/1    ImagePullBackOff 0          67s
```

This is because: We mentioned the image pull command wrongly in the scripting file:

votingapp

votingapp:18

Docker pull command: docker pull votingappbuild.azurecr.io/votingapp:18

Manifest

```
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.dockerdistribution.manifest.v2+json",
  "config": {
    "MediaType": "application/vnd.docker.container.image.v1+json",
    "size": 1586,
    "digest": "sha256:47212938e36d4f57414fc6431ad735f3923f60148a62b73aa462d9a36389317f5"
}
```

When we write script to update the Image name in the deployment.yaml file we haven't mentioned correctly:

```

13 # Make changes to the Kubernetes manifest file(s)
14 # For example, let's say you want to change the image tag in a deployment.yaml file
15 sed -i "s|image:.*|image: votingappbuild.azurecr.io/votingapp:18|g" k8s-specifications/$1-deployment.yaml
16 |
17 # Add the modified files
18 git add .

```

This is wrong. Update it correctly:

```

13 # Make changes to the Kubernetes manifest file(s)
14 # For example, let's say you want to change the image tag in a deployment.yaml file
15 sed -i "s|image:.*|image: votingappbuild.azurecr.io/v2:$3|g" k8s-specifications/$1-deployment.yaml
16
17 # Add the modified files

```

Now do a change in app.py again and check the pipeline run and deployment and pod.

#20241028.10 • Updated app.py

Triggered by Varuntej Yadla

Repository and version: Voting-App main 5879dee0

Time started and elapsed: Just now 8s

Related: 0 work items 0 artifacts

Tests and coverage: Get started

Vote-service

Runs Branches Analytics

Description Stages

#20241028.10 • Updated app.py

Individual CI for main 5879dee0 3m ago 1m 38s

Container registry: votingappbuild | Repositories

Overview Activity log Access control (IAM) Tags Quick start Events Settings Access keys Encryption Identity

Repositories resultapp votingapp

```

11 app: vote
12 template:
13 metadata:
14   labels:
15     app: vote
16 spec:
17   containers:
18     - image: votingappbuild.azurecr.io/votingapp:18
19       name: vote
20     ports:

```

Allow Argo CD to pull the change in manifest file. If still failed delete the pod and wait for sometime it will be up and running.

Argo v2.12.6+4dab5bd

APPLICATION DETAILS TREE

Sync Status: Synced to HEAD (3a11b49)

Last Sync: Sync OK to 3a11b49

Sync OK to 3a11b49

Pods (votingapp):

- vote-6d9ff87bb9b-8m2gv (Ready, 1/1, Running, 0 restarts, 4h22m)
- redis-77fcbb7f9-r25xt (Ready, 1/1, Running, 0 restarts, 4h22m)
- result-54b5ccfc95-7xbwz (Ready, 1/1, Running, 0 restarts, 4h22m)
- vote-7fbfb4cd6-zw6ks (Ready, 1/1, Running, 0 restarts, 78s)
- worker-7dd74bcbb-wfv8n (Ready, 1/1, Running, 0 restarts, 4h22m)

```
vvaru@Tej MINGW64 ~
$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
db-6d9ff87bb9b-8m2gv   1/1     Running   0          4h22m
redis-77fcbb7f9-r25xt  1/1     Running   0          4h22m
result-54b5ccfc95-7xbwz 1/1     Running   0          4h22m
vote-7fbfb4cd6-zw6ks   1/1     Running   0          78s
worker-7dd74bcbb-wfv8n  1/1     Running   0          4h22m
```

Let's try to access Vote app:

```
vvaru@Tej MINGW64 ~
$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
db        ClusterIP  10.0.209.116  <none>        5432/TCP    4h31m
kubernetes  ClusterIP  10.0.0.1    <none>        443/TCP     8h
redis     ClusterIP  10.0.104.117  <none>        6379/TCP    4h31m
result    NodePort   10.0.71.247   <none>        8081:31001/TCP 4h31m
vote      NodePort   10.0.88.30    <none>        8080:31000/TCP 4h31m

vvaru@Tej MINGW64 ~
$ kubectl get node -o wide
NAME            STATUS   ROLES   AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE   KERNEL-VERSION   CONTAINER-RUNTIME
aks-agentpool-35213278-vms000000  Ready    <none>   8h    v1.29.9   10.224.0.4   51.143.91.198   Ubuntu    5.15.0-1073-azure   containerd/1.7.22-1

51.143.91.198:31000
```

Microsoft Start

Vishakhapatnam 28°C

Search the web

Not opening because we need to allow the port number for this node pool in inbound rules:

Go to Azure portal → Virtual machine scale sets → Network Settings:

Networking

Network settings

- Load balancing
- Application security groups
- Network manager

Settings

Availability + scale

Security

Operations

Monitoring

Automation

Network security group aks-agentpool-12995614-nsg (attached to subnet: aks-subnet)

Impacts 1 subnets, 0 network interfaces

+ Create port rule

Priority ↑	Name	Port	Protocol	Source	Destination
100	AllowAnyCustom31336Inbound	31336	Any	Any	Any
110	AllowAnyCustom31000Inbound	31000	Any	Any	Any
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any
65500	DenyAllInbound	Any	Any	Any	Any

Helm vs No-helm!

HELM

NO-HELM

(Tip: you can change your vote)

Now we have to follow the process from Day 15 for result and worker (No Nodeport – Build satge not written – take care of this).

Result service:

Added new stage to the pipeline:

← Result-service

```
main ✘ Voting-App / azure-pipelines-result.yml *  
55 - stage: Update  
56   displayName: Update  
57   jobs:  
58     - job: Update  
59       displayName: Update  
60       steps:  
61         - task: ShellScript@2  
62           inputs:  
63             scriptPath: 'scripts/updateK8sManifests.sh'  
64             args: 'result $(imageRepository) $(tag)'
```

result-deployment.yaml file: adding imagePullSecrets



Result Pod should be NodePort:

```
yvaru@Tej MINGW64 ~  
$ kubectl get svc  
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE  
db        ClusterIP  10.0.209.116 <none>        5432/TCP    8h  
kubernetes  ClusterIP  10.0.0.1    <none>        443/TCP     12h  
redis     ClusterIP  10.0.104.117 <none>        6379/TCP    8h  
result     NodePort   10.0.71.247  <none>        8081:31001/TCP 8h  
vote      NodePort   10.0.88.30   <none>        8080:31000/TCP 8h
```

Already NodePort:

```
Committed 388e7e6: Updated result-deployment.yaml  
Create a pull request X  
k8s-specifications  
db-deployment.yaml  
db-service.yaml  
redis-deployment.yaml  
redis-service.yaml  
result-deployment.yaml  
result-service.yaml  
vote-deployment.yaml  
result-service.yaml  
spec:  
  type: NodePort  
  ports:  
    - name: "result-service"  
      port: 8081  
      targetPort: 80  
      nodePort: 31001  
      selector:  
        app: result
```

Need to allow port 31001 in Inbound rules: Update from Virtual machine scale sets.

Diagnose and solve problems

Instances

Networking

Network settings

- Load balancing
- Application security groups
- Network manager

Settings

Availability + scale

Security

Network security group aks-agentpool-12995614-nsg (attached to subnet: aks-subnet)
Impacts 1 subnets, 0 network interfaces

+ Create port rule

Priority ↑	Name	Port	Protocol	Source	Destination
100	AllowAnyCustom31336Inbound	31336	Any	Any	Any
110	AllowAnyCustom31000Inbound	31000	Any	Any	Any
120	AllowAnyCustom31001Inbound	31001	Any	Any	Any

Do a small change in result directory (like adding empty line or removing empty line) and it should trigger CI pipeline automatically:

Pipelines

Recent All Runs

New pipeline ⋮

Filter pipelines

Recently run pipelines

Pipeline	Last run
Result-service	#20241028.1 • Updated server.js ⌚ Individual CI for 🏃 main

It is in queue for very long check the agent is running or not. If not then restart the agent.

```
yvaru@Tej MINGW64 ~/Downloads
$ ssh -i ciagent_key.pem azureuser@52.172.99.184
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-azure x86_64)

azureuser@ciagent:~$ cd .cache/.ssh/myagent/
azureuser@ciagent:~$ cd myagent/
azureuser@ciagent:~/myagent$ ls
_diag bin env.sh license.html run-docker.sh svc.sh
_work config.sh externals reauth.sh run.sh vsts-agent-linux-x64-3.246.0.tar.gz
azureuser@ciagent:~/myagent$ ./run.sh
Scanning for tool capabilities.
Connecting to the server.
2024-10-28 18:09:41Z: Listening for Jobs
2024-10-28 18:09:44Z: Running job: Build
```

After starting the agent, CI pipeline started first Job.

Result-service

Runs Branches Analytics

Description Stages

#20241028.1 • Updated server.js ⌚ Individual CI for 🏃 main	🕒 - ✅ - ✅	6m ago 1m 35s
---	-----------	------------------

Home > Container registries > votingappbuild | Repositories >

votingappbuild | Repositories Container registry

Search Refresh ...

Overview Activity log Access control (IAM) Tags Quick start Events Settings

Access keys Encryption Identity

New to ACR, Artifact streaming helps pull images faster from AKS clusters. The 'Artifact streaming status' column shows which repositories are using this feature. [Learn more](#)

Repositories ↑↓

- resultapp
- votingapp

resultapp ... Repository

Refresh Start artifact streaming Manage deleted artifacts Delete repository

Essentials

Repository	resultapp	Tag count	3
Last updated date	28/10/2024, 23:40 GMT+5:30	Manifest count	3

Search to filter tags ...

Tags ↑↓	Digest ↑↓	Last modified
20	sha256:4804314922e02c02258d3865f1e20b85ad6...	28/10/2024, 23:40 GMT+5:30
5	sha256:85cde898f9b109b05505e9b15db99587e8...	28/10/2024, 02:42 GMT+5:30
4	sha256:b6c73631f465a18d6398bc3af178885094f8...	28/10/2024, 02:38 GMT+5:30

Image also updated in manifest file:

k8s-specifications

- db-deployment.yaml
- db-service.yaml
- redis-deployment.yaml
- redis-service.yaml
- result-deployment.yaml
- result-service.yaml
- vote-deployment.yaml
- vote-service.yaml
- worker-deployment.yaml

Committed 6ccb34f7: Updated server.js

Create a pull request ⋮

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   labels:
5     app: result
6     name: result
7 spec:
8   replicas: 1
9   selector:
10    matchLabels:
11      app: result
12   template:
13     metadata:
14       labels:
15         app: result
16     spec:
17       containers:
18         - image: votingappbuild.azurecr.io/resultapp:20
19         name: result
20       ports:
```

Wait for ArgoCd to pick up the changes in manifest file. Or refresh the voting app application in argocd.

APP HEALTH ⓘ Healthy

SYNC STATUS ⓘ Synced to HEAD (d2931e5)

Auto sync is enabled.

Author: Ubuntu <azureuser@ciagent.rhisesobamvedioifro23...
Comment: Update Kubernetes manifest

LAST SYNC ⓘ Sync OK to d2931e5

Succeeded a few seconds ago (Mon Oct 28 2024 23:43:38 GMT+0530)

Author: Ubuntu <azureuser@ciagent.rhisesobamvedioifro23...
Comment: Update Kubernetes manifest

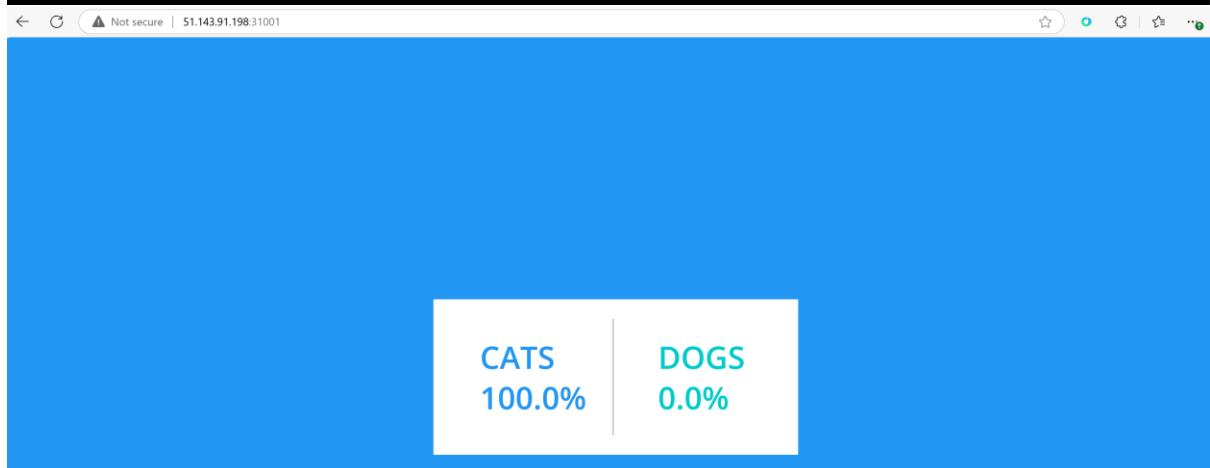
```
vvaru@Tej MINGW64 ~
$ kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
db-6d9f87bb9b-8m2gv   1/1     Running   0          9h
redis-77fcfb7f9-r25xt 1/1     Running   0          9h
result-66b587fc84-9cftx 1/1     Running   0          10m
vote-7fbff4cd6-zw6ks   1/1     Running   0          4h39m
worker-7dd74bcbbb-wfv8n 1/1     Running   0          9h
result-7bc5494bd7-wt52m 0/1     Pending   0          0s
result-7bc5494bd7-wt52m 0/1     Pending   0          0s
result-7bc5494bd7-wt52m 0/1     ContainerCreating 0          0s
result-7bc5494bd7-wt52m 0/1     Running   0          8s
result-66b587fc84-9cftx 1/1     Terminating 0          11m
result-66b587fc84-9cftx 0/1     Terminating 0          11m
result-66b587fc84-9cftx 0/1     Terminating 0          11m
result-66b587fc84-9cftx 0/1     Terminating 0          11m
```

The new pod created first then old one got terminated. Same can be seen from argo CD UI as well.

Try to access Result service also:

```
vvaru@Tej MINGW64 ~
$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
db        ClusterIP  10.0.209.116 <none>        5432/TCP    9h
kubernetes  ClusterIP  10.0.0.1    <none>        443/TCP     12h
redis     ClusterIP  10.0.104.117 <none>        6379/TCP    9h
result    NodePort   10.0.71.247  <none>        8081:31001/TCP 9h
vote      NodePort   10.0.88.30   <none>        8080:31000/TCP 9h

vvaru@Tej MINGW64 ~
$ kubectl get node -o wide
NAME           STATUS   ROLES   AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE   KERNEL-VERSION   CONTAINER-RUNTIME
aks-agentpool-35213278-vmss000000 Ready   <none>   12h   v1.29.9   10.224.0.4   51.143.91.198   Ubuntu 22.04.5 LTS   5.15.0-1073-azure   containerd://1.7.22-1
```



It is working eventough we haven't completed the process for worker service. Because worker image was taken from dockerhub. So it is working. We can create it and check it should be working.

Worker Service:

Added new stage for Push and Update:

← Worker-service

main ↴

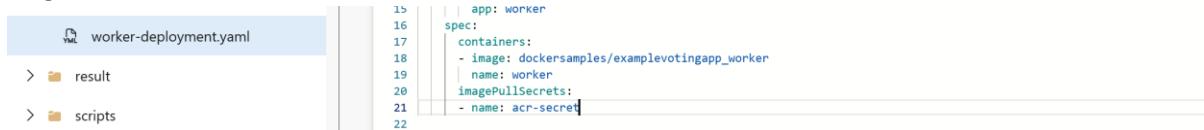
Voting-App / azure-pipelines-worker.yml *

```

12 variables:
13   - # Container registry service connection established during pipeline creation
14   - dockerRegistryServiceConnection: 'ffaea5df-46d6-4d00-8c4b-c537f41f40db'
15   - imageRepository: 'workerapp'
16   - containerRegistry: 'votingappbuild.azurecr.io'
17   - dockerfilePath: '$(Build.SourcesDirectory)/worker/Dockerfile'
18   - tag: '$(Build.BuildId)'
19
20 - stage: Push
21   - displayName: Push
22   - jobs:
23     - job: Push
24       - displayName: Push
25       - steps:
26         - Settings
27           - task: Docker@2
28             - displayName: Push the Image
29             - inputs:
30               - containerRegistry: '$(dockerRegistryServiceConnection)'
31               - repository: '$(imageRepository)'
32               - command: 'push'
33               - tags: '$(tag)'
34
35 - stage: Update
36   - displayName: Update
37   - jobs:
38     - job: Update
39       - displayName: Update
40       - steps:
41         - Settings
42           - task: ShellScript@2
43             - inputs:
44               - scriptPath: 'scripts/updateK8sManifests.sh'
45             - args: 'worker $(imageRepository) $(tag)'

```

We haven't created the repository for the worker image that will be build during CI. New repo in Container registry with exact same name will be created when Pipeline starts and build and push stage is successful.

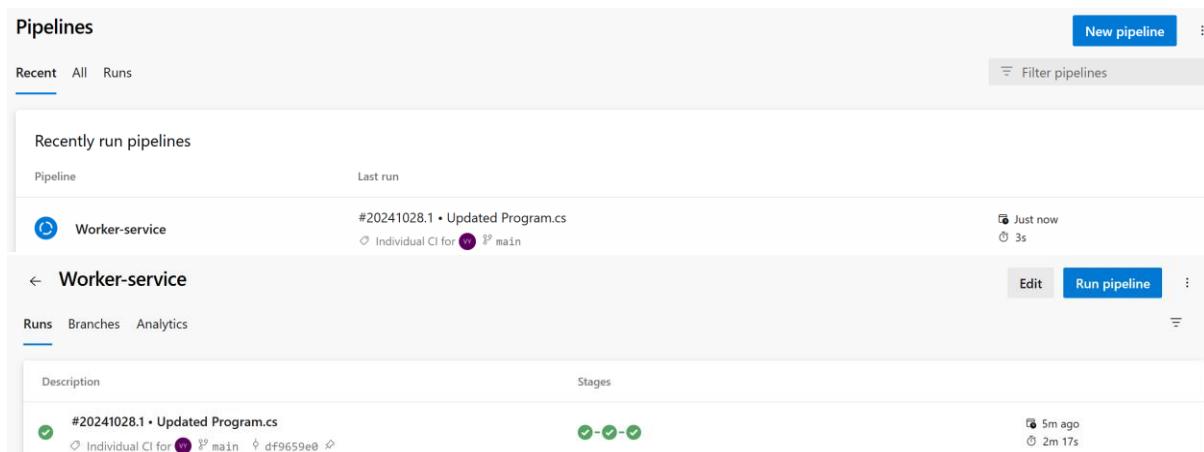


```

15 app: worker
16 spec:
17   containers:
18     - image: dockersamples/exemplevotingapp_worker
19       name: worker
20       imagePullSecrets:
21         - name: acr-secret

```

Update something in worker directory:



Pipelines

Recent All Runs

Recently run pipelines

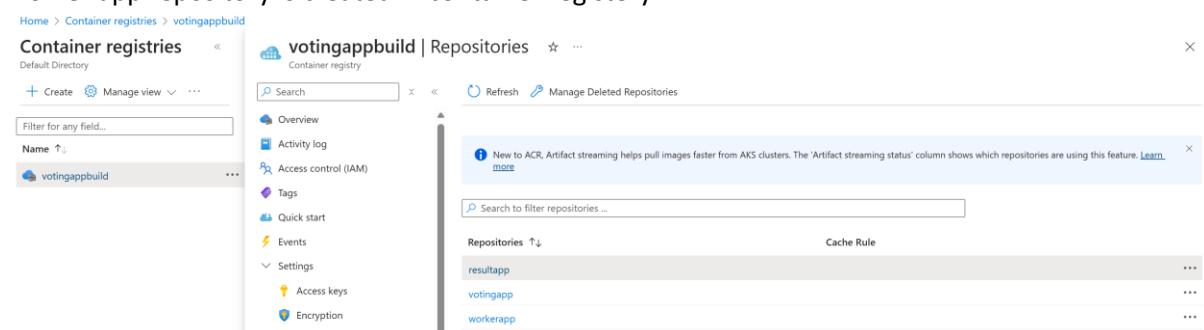
Pipeline	Last run	
Worker-service	#20241028.1 • Updated Program.cs Individual CI for main	Just now 3s

Worker-service

Runs Branches Analytics

Description	Stages	
#20241028.1 • Updated Program.cs Individual CI for main	✓ - ✓ - ✓	5m ago 2m 17s

worker app repository is created in container registry:



Container registries

votingappbuild | Repositories

Repositories ↑

- resultapp
- votingapp
- workerapp

Home > Container registries > votingappbuild | Repositories >

votingappbuild | Repositories

Container registry

Search ... Refresh ...

Overview Activity log Access control (IAM) Tags Quick start Events Settings Access keys

New to ACR? Artifact streaming helps pull images faster from AKS clusters. The Artifact streaming status column shows which repositories are using this feature. [Learn more](#)

Repositories ↑↓

vote-deployment.yaml vote-service.yaml worker-deployment.yaml

```

14
15
16
17
18
19
20
21
labels:
  app: worker
spec:
  containers:
    - image: votingappbuild.azurecr.io/workerapp:21
      name: worker
      imagePullSecrets:
        - name: acr-secret

```

workerapp Repository

Repository workerapp Last updated date 29/10/2024, 00:08 GMT+5:30

Tag count 1 Manifest count 1

Search to filter tags ...

Tags ↑↓ Digest ↑↓ Last modified

sha256:89162aa46acb87aff9d75e9aa0718d1fbe2b... 29/10/2024, 00:08 GMT+5:30

```

yvare@Tej MINGW64 ~
$ kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
db-6d9f87bb-8m2gv   1/1     Running   0          9h
redis-77fc7f9-r25xt 1/1     Running   0          9h
result-7bc5494bd7-wt52m 1/1     Running   0          27m
vote-7fbfb4cd6-zw6ks 1/1     Running   0          5h6m
worker-8667f6b499-qn54z 1/1     Running   0          21s

```

Try to update the vote in vote app and check in result app. It should work fine.

Project is done. Will delete resources created from azure portal:

It will delete AKS, Container registries, Agent pool, ArgoCd inside AKS.

Need to do all this process if you want to run this application again. Rest of the application source code, manifest files and scripts will be in azure DevOps.

Files changed in source code are: Manifest files for Vote, result and worker (Deployment.yaml)

Added Scriptting file for updating Image name in deployment manifest files.

Create CI pipelines for Vote, Result and Worker.