

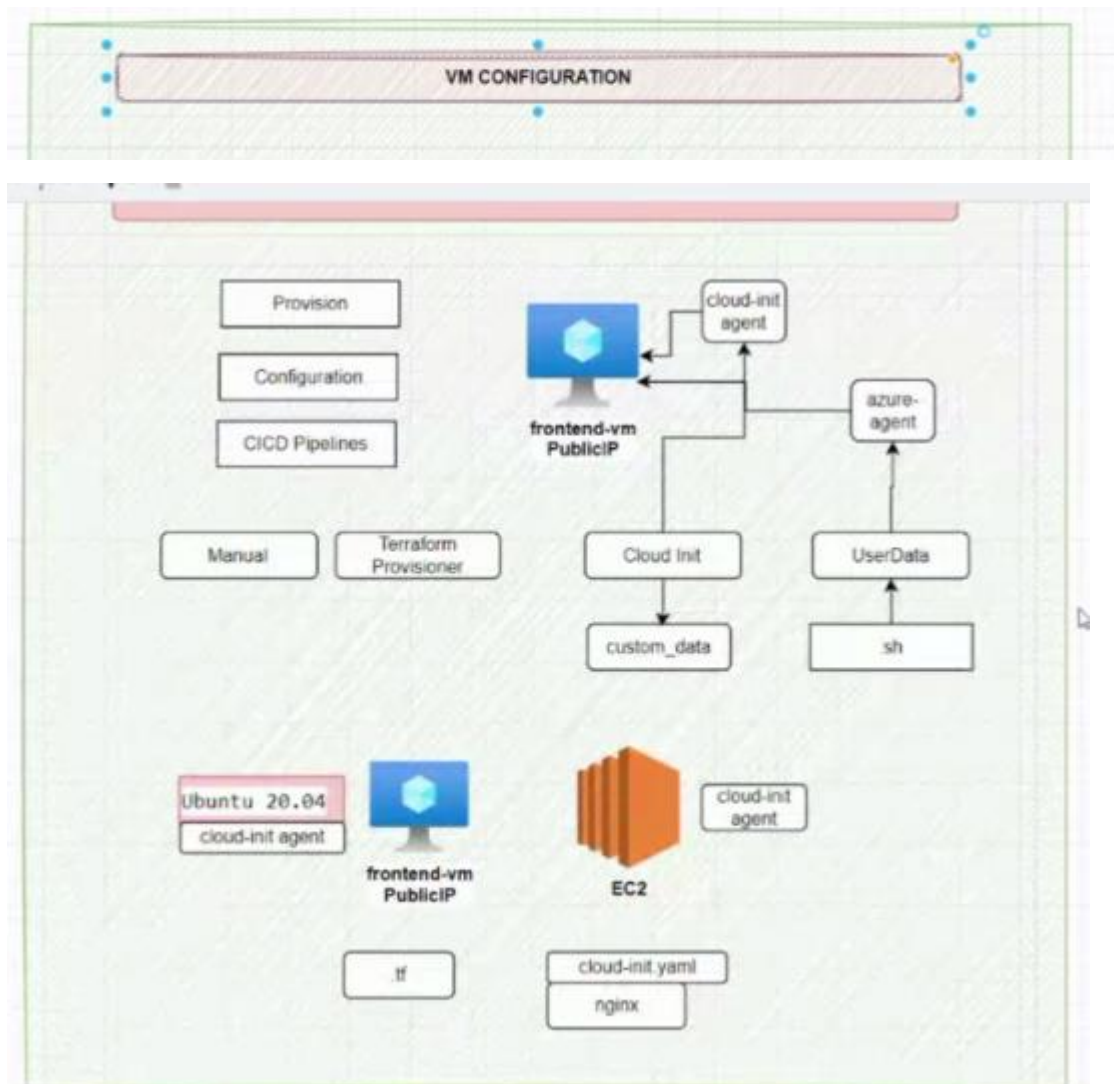
11 August 2024

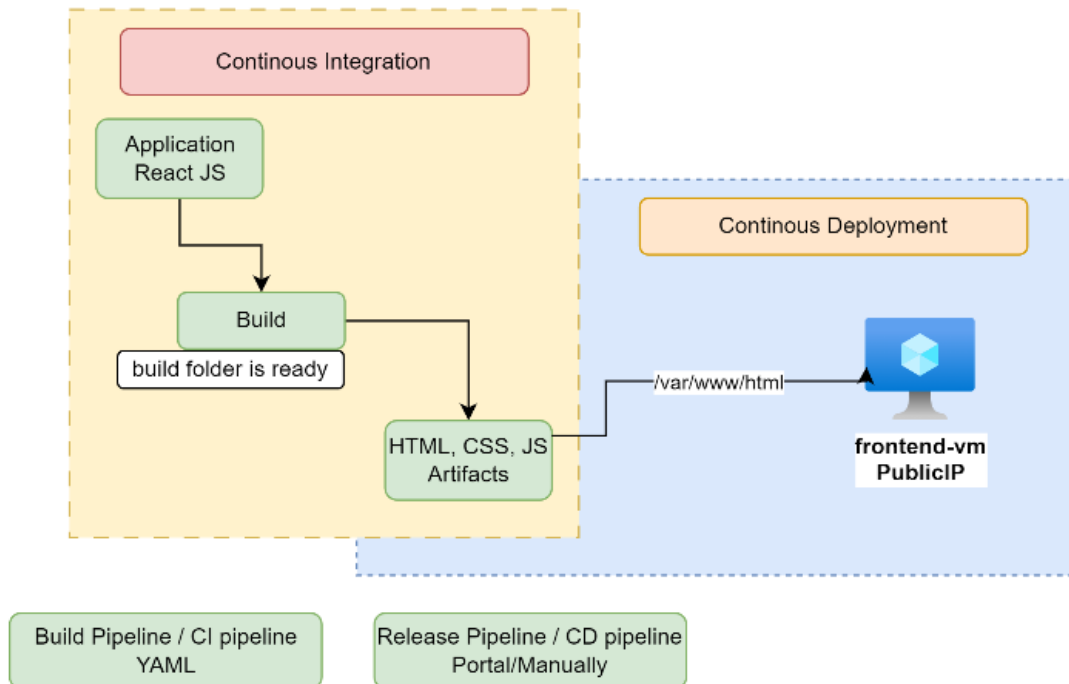
AGENDA – Writing in yaml

1) What is backlog, sprint?

2) 2 branching strategy

i) trunk based in terraform – in main branch we made 2 folders preprod and prod. We made 2 pipelines for them. Now using git clone we made feature branch for dev folder. Now pushed it and raised a PR then if plan was fine then PR was approved. Then it merged and automatically pipeline ran then all stages of pipeline ran along with tfsec, tuple of, linter, checkov all tools ran and then apply ran and got deployed.



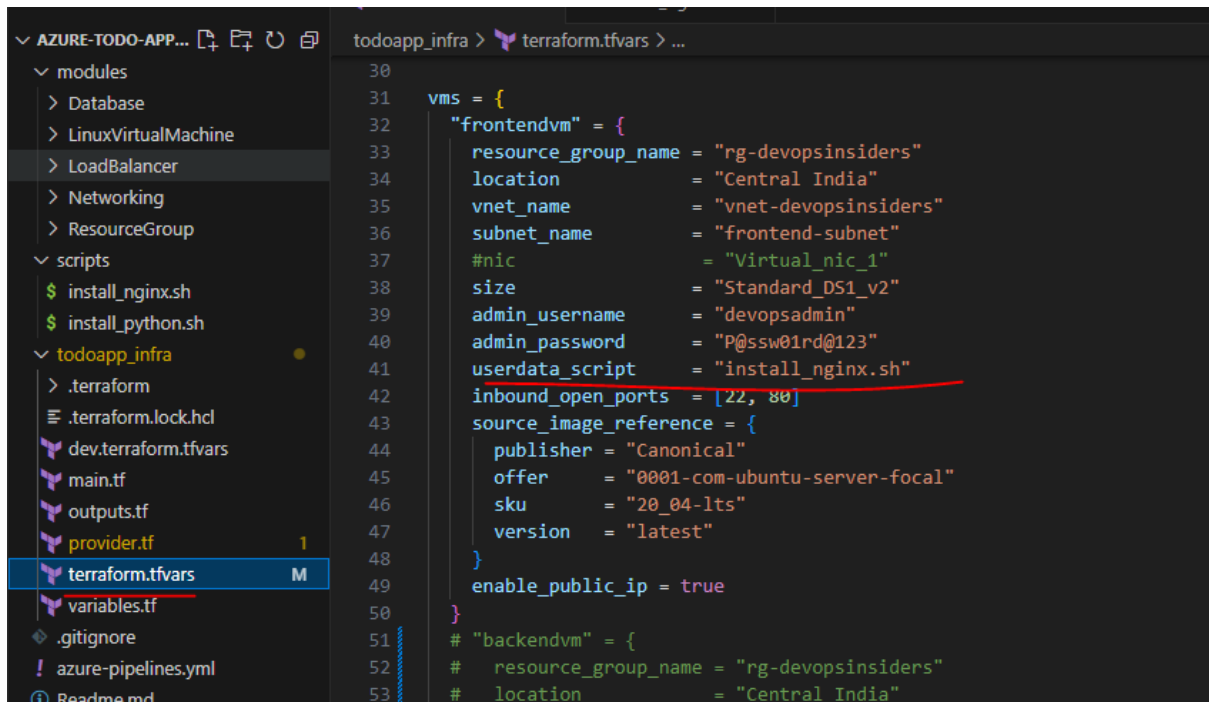


3) Now put nginx script in code

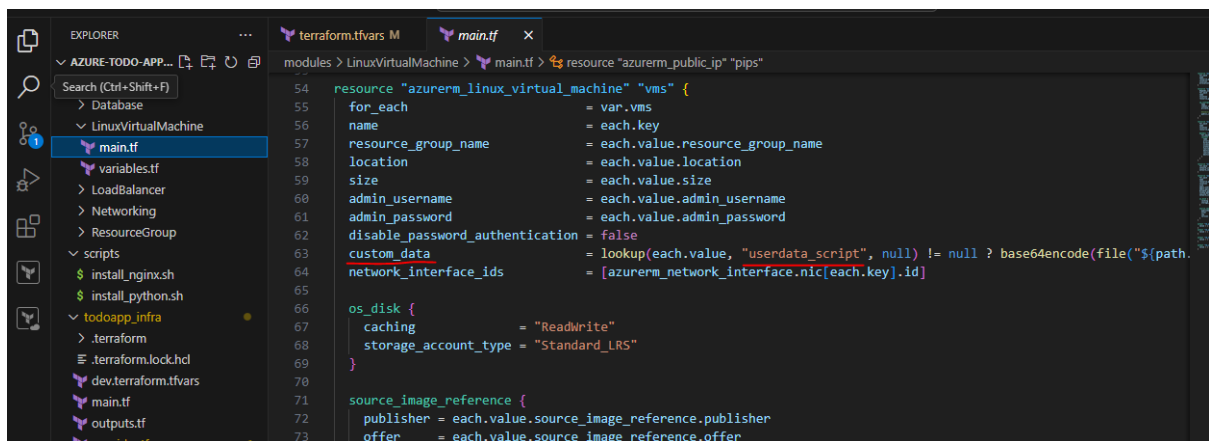
The screenshot shows a code editor with the following components:

- EXPLORER:** A sidebar on the left showing the file structure. The file `install_nginx.sh` is selected and highlighted with a red circle.
- Code Editor:** The main area showing the content of `install_nginx.sh`. The script is as follows:

```
1  #!/usr/bin/env bash
2
3  set -euo pipefail
4
5  echo "[INFO] Installing Nginx"
6
7  sudo apt update
8  sudo apt install nginx -y
```
- Terminal:** A terminal window at the bottom showing the command `scripts > $ install_nginx.sh` being executed.



```
30
31
32 vms = {
33   "frontendvm" = {
34     resource_group_name = "rg-devopsinsiders"
35     location             = "Central India"
36     vnet_name            = "vnet-devopsinsiders"
37     subnet_name          = "frontend-subnet"
38     #nic                  = "Virtual_nic_1"
39     size                  = "Standard_DS1_v2"
40     admin_username        = "devopsadmin"
41     admin_password        = "P@ssw01rd@123"
42     userdata_script       = "install_nginx.sh"
43     inbound_open_ports    = [22, 80]
44     source_image_reference = {
45       publisher = "Canonical"
46       offer     = "0001-com-ubuntu-server-focal"
47       sku       = "20_04-lts"
48       version   = "latest"
49     }
50     enable_public_ip = true
51   }
52   # "backendvm" = {
53     # resource_group_name = "rg-devopsinsiders"
54     # location             = "Central India"
```



```
54 resource "azurerm_linux_virtual_machine" "vms" {
55   for_each = var.vms
56   name     = each.key
57   resource_group_name = each.value.resource_group_name
58   location  = each.value.location
59   size      = each.value.size
60   admin_username = each.value.admin_username
61   admin_password = each.value.admin_password
62   disable_password_authentication = false
63   custom_data = lookup(each.value, "userdata_script", null) != null ? base64encode(file("${path.
64   network_interface_ids = [azurerm_network_interface.nic[each.key].id]
65
66   os_disk {
67     caching           = "ReadWrite"
68     storage_account_type = "Standard_LRS"
69   }
70
71   source_image_reference {
72     publisher = each.value.source_image_reference.publisher
73     offer     = each.value.source_image_reference.offer
```

4) git add .

git commit -m "updated"

git push

5) run pipeline

dev.azure.com/kuberdev66666/ToDoapp/_build/results?buildId=71&view=logs&j=e2822545-b2ff-5121-a946-f8d19f74d0b3&t=e28

Azure DevOps kuberdev66666 / ToDoapp / Pipelines / todoapp_infra_pipeline / 20241009.2

ToDoapp

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Releases
- Library
- Task groups

Jobs in run #20241009.2
todoapp_infra_pipeline

initandPlan		
> ✓ initandPlan	30s	
apply		
> ✓ manualvalidationjob	9s	
▼ ✓ apply	5m 5s	
✓ Initialize job	4s	
✓ Checkout azure-todo-a...	5s	
✓ TerraformInstaller	1s	

✓ apply

```

1 Pool: Agentpool21
2 Agent: Testagent21
3 Started: Today at 11:29 PM
4 Duration: 5m 5s
5
6 ▶ Job preparation parameters
41 Job live console data:
42 Starting: apply
43 Async Command Start: DetectDockerContainer
44 Async Command End: DetectDockerContainer
45 Async Command Start: DetectDockerContainer
46 Async Command End: DetectDockerContainer
47 Finishing: apply

```

6) Now ssh into frontend vm and check whether nginx is there or not as we had already used nginx script in our code. So nginx is there.

```

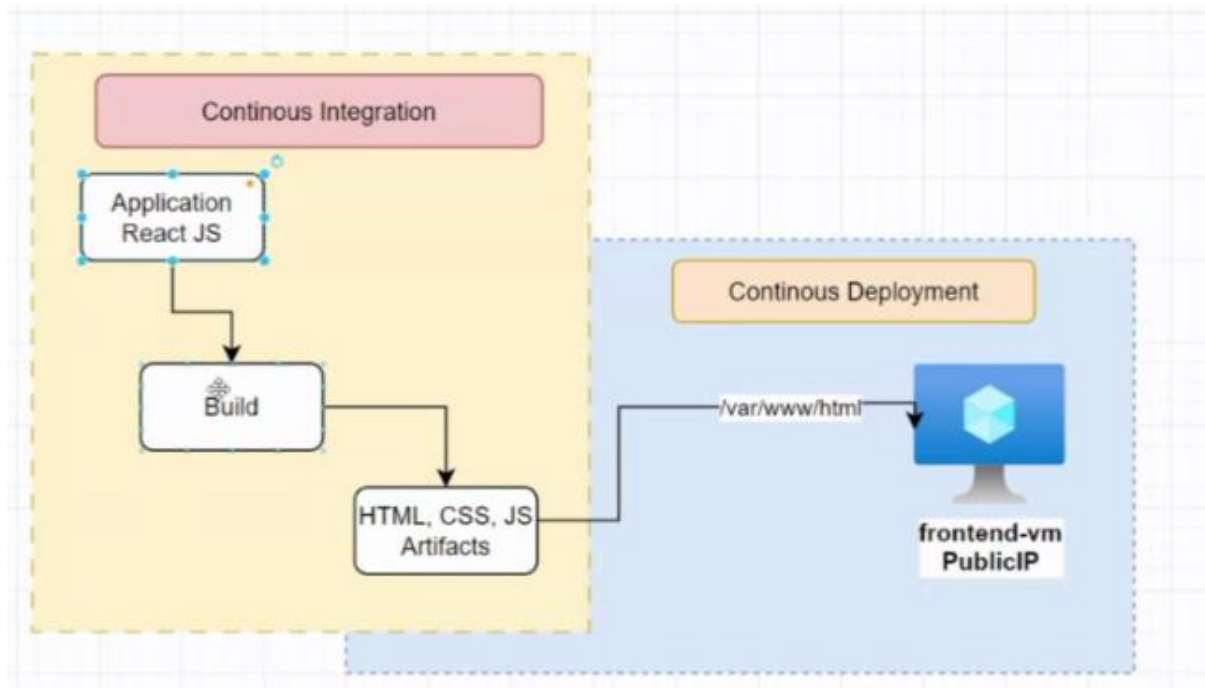
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

devopsadmin@frontendvm:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-10-09 18:02:18 UTC; 4min 5s ago
     Docs: man:nginx(8)
  Main PID: 2337 (nginx)
    Tasks: 2 (limit: 4081)
   Memory: 4.9M
    CGroup: /system.slice/nginx.service
            └─2337 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              └─2338 nginx: worker process

Oct 09 18:02:17 frontendvm systemd[1]: Starting A high performance web server and a reverse proxy server...
Oct 09 18:02:18 frontendvm systemd[1]: Started A high performance web server and a reverse proxy server.
devopsadmin@frontendvm:~$

```

7) Now we want that in single click our application can be made, then gets build, artefacts can be made (converted into html, css, javascript) and then it gets deployed on our vm as shown in below diagram.



8) Continuous Integration – The process of making artefacts by building the code of application automatically

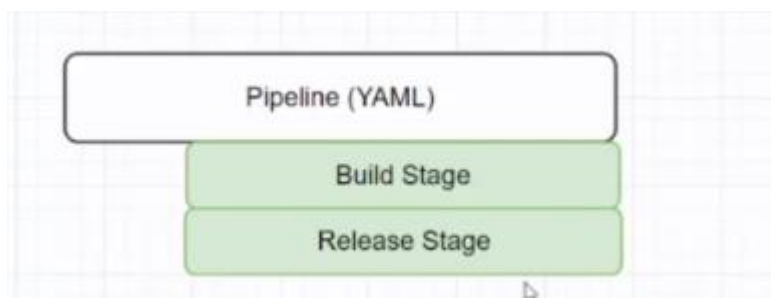
9) Continuous Deployment – The process of transferring artefacts on to vm automatically.

10) Build pipeline and release pipeline is shown below



11) Release or CD pipeline is not available in yaml

12) So in single pipeline we will create 2 stages



13) Now go to dev.azure.com and import below code in repo from github

ReactTodoUIMonolith Public

ReactTodoUIMonolith Repository is housing the user interface code developed in React for the Todo application's monolithic architecture. This repository is responsible for managing the frontend...

reactjs monolithic todoapp

JavaScript 6 26 Updated on Jan 4

/ ReactTodoUIMonolith Public

Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags

Go to file Code

Anshuman2121 Change name to DevOpsInsiders (#1)

- build Change name to DevOpsInsiders (#1)
- public Change name to DevOpsInsiders (#1)
- src updated code
- .gitignore added build repo
- README.md Update README.md
- package-lock.json initial commit

Clone

HTTPS GitHub CLI

https://github.com/devopsinsiders/ReactTodoUIMonolith

Clone using the web URL

Open with GitHub Desktop

Download ZIP

ChatGPT Settings - Agent pool azure-todo-app-terra Pipelines - Run 2024 azure-devsecops-bst GitHub - devopsinsiders

dev.azure.com/kuberdev66666/ToDoapp/_git/azure-todo-app-terraform.git

Azure DevOps kuberdev66666 / ToDoapp / Repos / Files / azure-todo-app-terraform.git

Filter repositories

- azure-todo-app-terraform.git
- ToDoapp
- New repository
- Import repository
- Manage repositories

Scripts

Feb 11

f1657bc3 Updated Code DevOps Insiders

Azure DevOps kuberdev66666 / ToDoapp / Repos / Files / ReactTodoUIMonolith.git

main

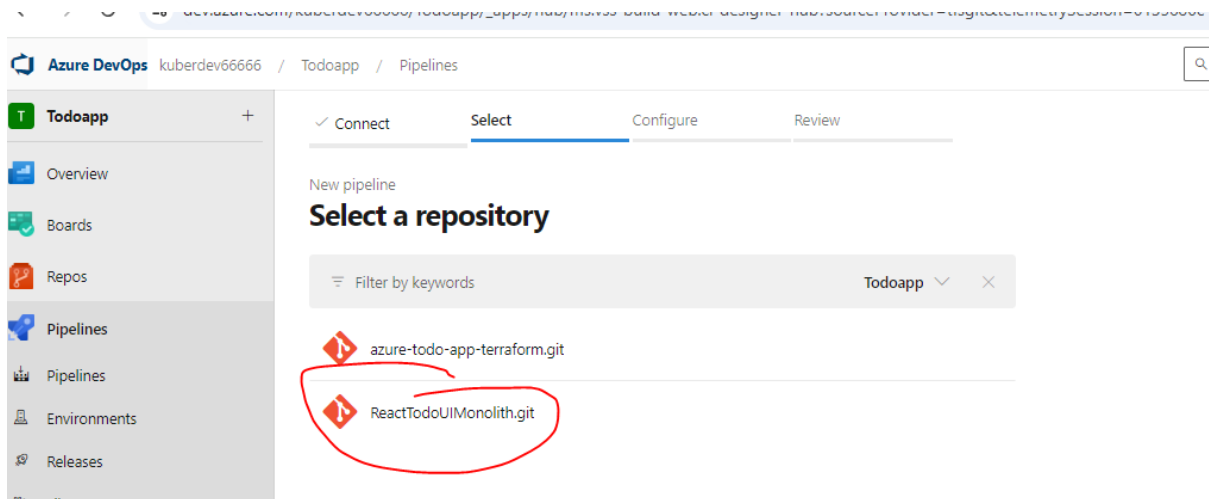
Files

Contents History

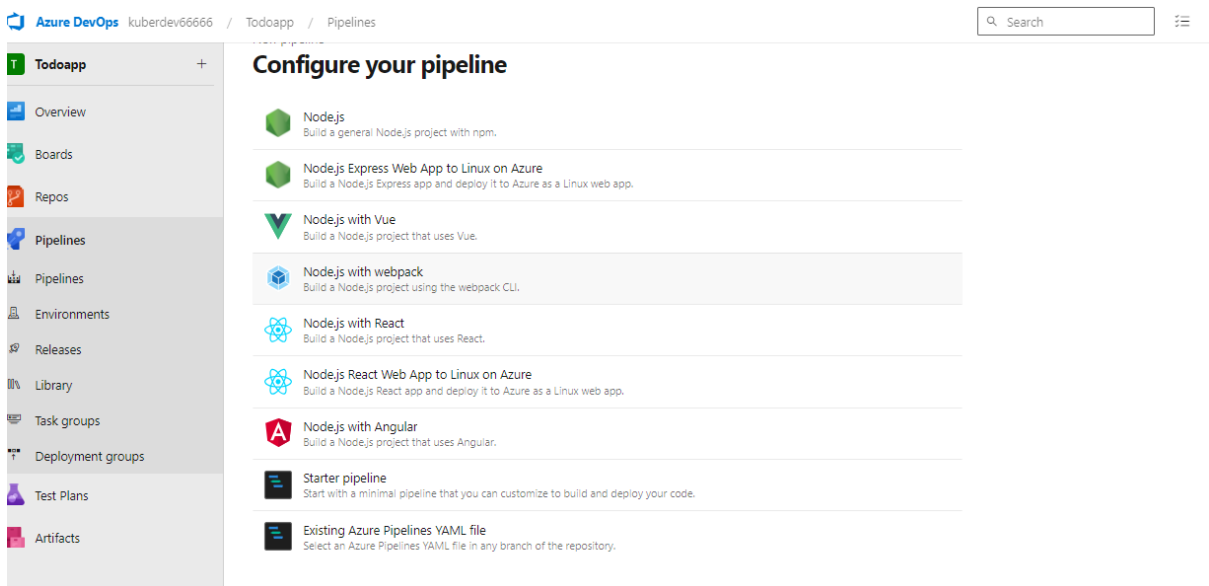
Name	Last change	Commits
build	Jan 4	22f48ea1 Change name to DevOpsInsiders (#1) Anshuman Ab...
public	Jan 4	22f48ea1 Change name to DevOpsInsiders (#1) Anshuman Ab...
src	Nov 7, 2023	73b91d58 updated code Aman Gupta
.gitignore	Nov 5, 2023	7280e194 added build repo Aman Gupta
package-lock.json	Nov 5, 2023	be413cc4 initial commit Aman Gupta
package.json	Nov 5, 2023	be413cc4 initial commit Aman Gupta
README.md	Dec 24, 2023	7b693633 Update README.md DevOps Insiders

Readme for Todo App

14) Now we will make our pipeline – select new pipeline – select repo below



15) Now azure devops is so smart that it identified that it's a node.js app

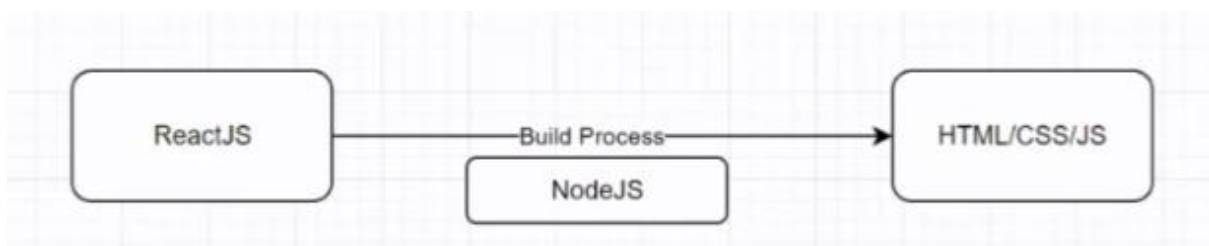


16) Select starter pipeline

17) SEARCH – azure YAML schema reference

<https://learn.microsoft.com/en-us/azure/devops/pipelines/yaml-schema/?view=azure-pipelines>

18) node.js acts as medium to convert react.js into html, css, javascript so that it can run on browser



19) SEARCH - predefined variables azure devops

<https://learn.microsoft.com/en-us/azure/devops/pipelines/build/variables?view=azure-devops&tabs=yaml>

System. Default WorkingDirectory	The local path on the agent where your source code files are downloaded. For example: <code>c:\agent_work\1\s</code> By default , new build pipelines update only the changed files. You can modify how files are downloaded on the Repository tab . This variable is agent-scoped. It can be used as an environment variable in a script and as a parameter in a build task, but not as part of the build number or as a version control tag.
---	--

← npm ⓘ

Command * ⓘ

install ▼

Working folder that contains package.json ⓘ

`$(System.DefaultWorkingDirectory)`

Custom registries and authentication ▼

Advanced ▼

20 Jis jagah agent ke andar code clone hoga, us s wali directory ka path yaha aajayega aur usi directory ya folder ke andar package.json file hogi

System.DefaultWorkingDirectory	The local path on the agent where your source code files are downloaded. For example: <code>c:\agent_work\1\s</code>
---------------------------------------	---

← npm ⓘ

Command * ⓘ

custom ▼

Working folder that contains package.json ⓘ

\$(System.DefaultWorkingDirectory)

Command and arguments * ⓘ

run build

Custom registries and authentication ▼

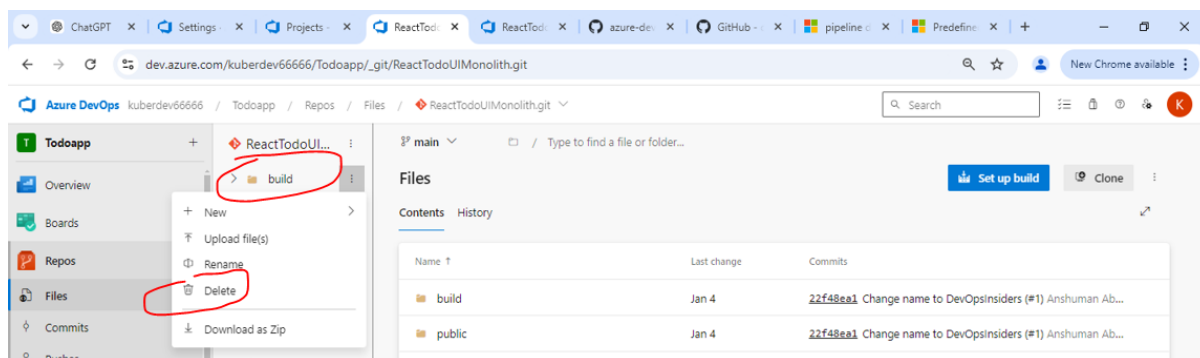
21) Now checking whether build folder is creating or not so we can use command line or power shell

```

- customCommand: 'run build'
- task: CmdLine@2
  inputs:
    script: 'ls'

```

22) Now delete build folder because it has to be generated or created



23) Now running pipeline

dev.azure.com/kuberdev66666/ToDoapp/_build/results?buildId=75&view=logs&g=f419e671-ec88-59c0-3f95-a934b22cee26&t=78db2542-c627-41...

Azure DevOps kuberdev66666 / ToDoapp / Pipelines / ReactToDoUIMonolith.git / 20241010.2

Jobs in run #20241010.2
ReactToDoUIMonolith.git

Buildstage	Task	Duration
BuildJob	Initialize job	4s
	Checkout ReactToDoUI...	7s
	Node Install	2s
	npm Install	30s
	npm run build	1m 19s
	CmdLine	2s
	Post-job: Checkout Re...	<1s
	Finalize Job	<1s
	Report build status	<1s

BuildJob

```

1 Pool: AgentPool21
2 Queued: Today at 12:42 PM [manage_parallel_jobs]
3 Agent: Testagent21
4 Started: Today at 12:42 PM
5 Duration: 2m 7s
6
7 The agent request is already running or has already completed.
8 Job preparation parameters
9 Job live console data:
10 Starting: BuildJob
11 Async Command Start: DetectDockerContainer
12 Async Command End: DetectDockerContainer
13 Async Command Start: DetectDockerContainer
14 Async Command End: DetectDockerContainer
15 Finishing: BuildJob
  
```

24) Now we can see in pipeline that our build folder is present, after running ls

Buildstage	Task	Duration
BuildJob	Initialize job	3s
	Checkout ReactToDoUI...	6s
	Node Install	3s
	npm Install	30s
	npm run build	1m 21s
	CmdLine	1s
	Post-job: Checkout Re...	<1s
	Finalize Job	<1s
	Report build status	<1s

```

1 Starting: CmdLine
2 =====
3 Task      : Command line
4 Description: Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
5 Version   : 2.244.3
6 Author    : Microsoft Corporation
7 Help      : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
8 =====
9 Generating script.
10 Script contents:
11 ls
12 ===== Starting Command Output =====
13 /usr/bin/bash --noprofile --norc /home/azureuser/agents/_work/_temp/1bf9f248-1952-486a-ae4f-81b76e3f6713.sh
14 README.md
15 azure-pipelines.yml
16 build
17 node_modules
18 package-lock.json
19 package.json
20 public
21 src
22
23 Finishing: CmdLine
  
```

25) Now what is made inside build folder can see as below

PowerShell

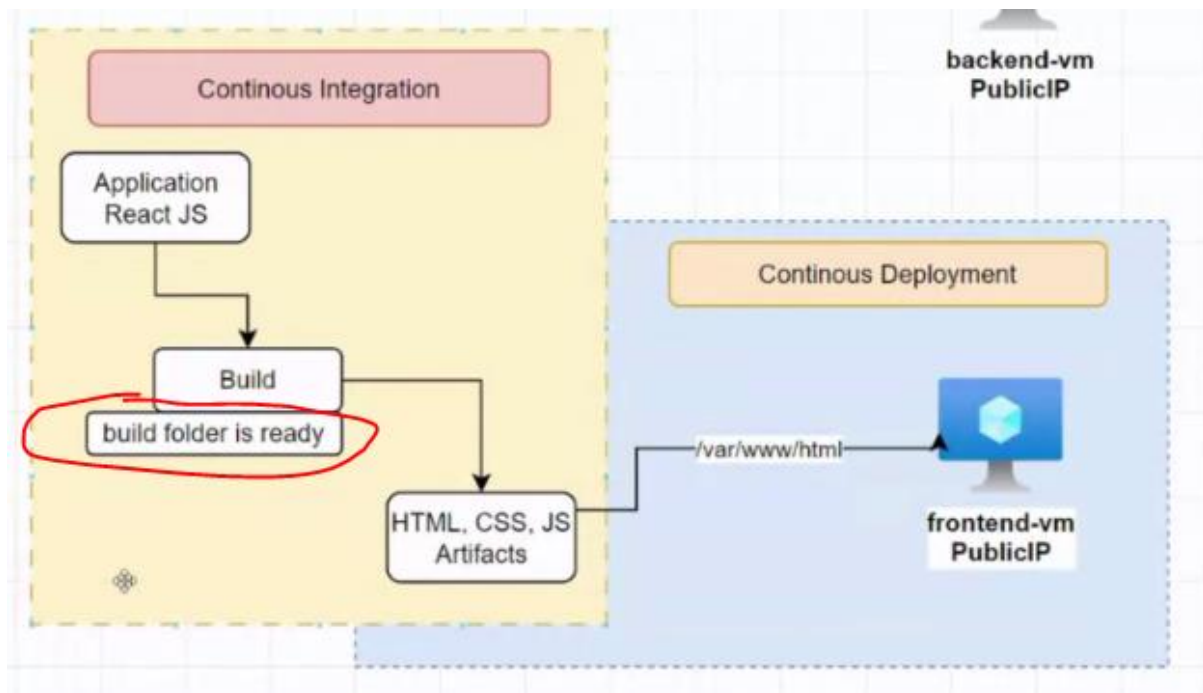
Type: ☐ File Path ☒ Inline

Script *

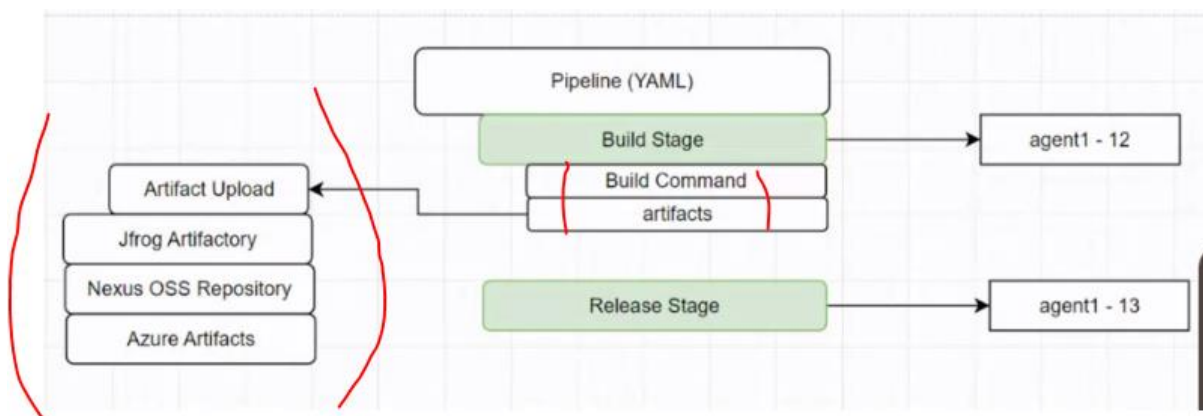
```
cd build/
ls
```

Preference Variables

Advanced



26)



27) Use “publish pipeline artifacts” or storage account to upload and publish artefacts i.e. build folder. So basically transferring content of build folder into storage account.

28) SEARCH – azure storage blob directory

<https://learn.microsoft.com/en-us/cli/azure/storage/blob/directory?view=azure-cli-latest#az-storage-blob-directory-upload>

Upload a local directory to a storage blob directory.

```
Azure CLI Copy Open Cloud Shell
az storage blob directory upload -c MyContainer --account-name MyStorageAccount
```

az storage blob directory upload -c container12 --account-name stbackend122 -s "\$(System.DefaultWorkingDirectory)/build" -d directory --recursive

← Azure CLI ⓘ

Azure Resource Manager connection * ⓘ

TESTSC21 ▼

Script Type * ⓘ

Shell ▼

Script Location * ⓘ

Inline script ▼

Inline Script * ⓘ

```
az storage blob directory upload -c
container12 --account-name stbackend122 -
s " $(System.DefaultWorkingDirectory)/build"
-d directory --recursive
```

Script Arguments ⓘ

[About this task](#) [Add](#)

29) Install azure cli on vm

30) Now write release stage and use below one

Download the entire directory in a storage container.

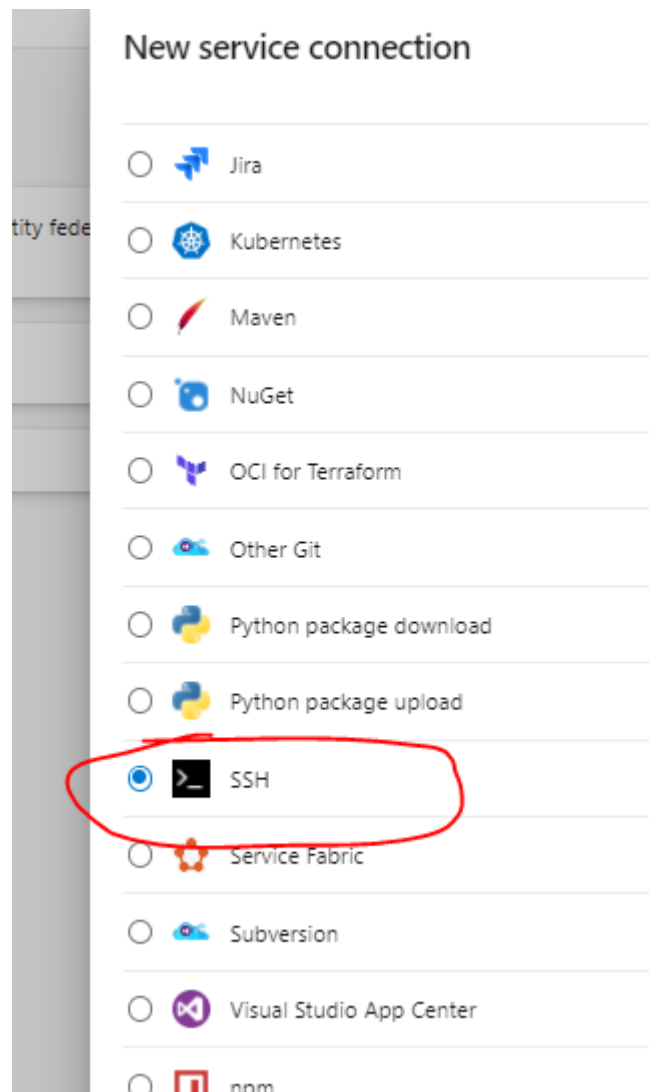
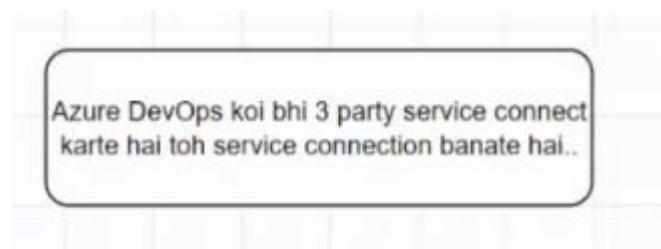
Azure CLI [Copy](#) [Open Cloud Shell](#)

```
az storage blob directory download -c MyContainer --account-name MyStorageAccou
```

az storage blob directory download -c **container12** --account-name **stbackend122** -s
SourceDirectoryPath -d "<local-path>" --recursive

```
>Settings
--task: AzureCLI@2
--inputs:
....azureSubscription: 'TESTSC21'
....scriptType: 'bash'
....scriptLocation: 'inlineScript'
....inlineScript: 'az storage blob directory download -c container12 --account-name stbackend122 -s "directory/build" -d "$(Sys
```

31) In ADO, for making 3rd party service to connect, we use service connection



31a) Give details of frontend vm

New SSH service connection

Host name

4.247.157.137

Host name or IP address of the remote machine.

Port number (optional)

22

Port number on the remote machine to use for connecting.

Private Key (optional)

Private Key for connecting to the endpoint

[Upload SSH private key file.](#)

Authentication

Username

devopsadmin

Username for connecting to the endpoint

Password/Passphrase (optional)

Password for connecting to the endpoint

Details

Service connection name

Frontend vm service connection

Service Management Reference (optional)

Description (optional)

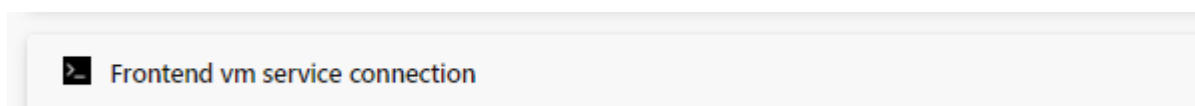
Security

☐ Grant access permission to all pipelines

[Learn more](#)
[Troubleshoot](#)

Back

Save

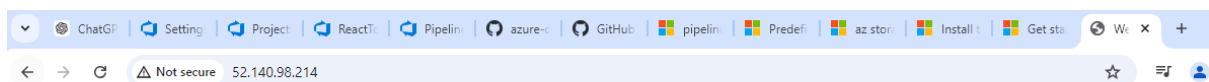


32) Lets transfer build artifacts folder into release folder

'az storage blob directory download -c container12 --account-name stbackend122 -s "directory/build" -d "\$(System.DefaultWorkingDirectory)/release" --recursive'

```
'az storage blob directory download -c container12 --account-name stbackend122 -s "directory/build" -d "$(System.DefaultWorkingDirectory)/release" --recursive'
```

33) After this run public ip of frontend vm in browser



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

