   Min 2 + yrs, Experience working on Azure cloud

•             3+ yrs, Experience in working with DevOps practices such as CI/CD, automation, real time monitoring

•             3+ yrs, Experience supporting and / or developing Java / C# Applications

•             Solid command over scripting languages such as Python, Bash or PowerShell

•             Experience Provisioning infrastructure on Azure using infrastructure as code

•             Experience in latest DevOps technologies such as Jenkins, APM, Terraform

•             Provision Azure resources using powershell and ARM templates.

•             Experience with Azure resources like vnet, sql, function app

Continuous Integration and Continuous Deployment: CI/CD

CI/CD pipeline: Executing the tasks one after other in automated manner with continuous integration and continuous deployment.

Stages of CI/CD pipeline:

Code Repository: GIT/GITHUB

Build : Mevan, but it is developers responsibility to configure pom.xml

Deployment : through powershell, ARM templates and Terraform.

Test

Monitoring

I majorly involved in Deployment stage.

We have separate devops Team, I am more into the Azure infrastructure administration

TERRAFORM

Terraform is an infrastructure as code (IaC) tool that lets you build, edit, and version infrastructure in a secure and efficient manner.

Teraform: It is an infrastructure as code deployment tool.

.tf is the extension

TFstate file : It holds the current state.

Teraform is IDEMPOTENT, because it maintains the state.

Resource information we can get from Terraform documentation

Tfvars

Vars.tf

We need to change the variable

Module: repeated script block using in multiple terraform files in same folder

Variable: repeated value

Terraform state file save in the central secure location. Make sure it should not modify.

Teraform version 12

Create Main.tf

To execute Main.tf login to Azure using “Az login”

Terraform init : It initialize and it will download required providers to execute this script.

Terraform plan: It will show you what resources it is going to deploy

Terraform apply: It will apply and deploy the resources what ever mentioned in main.tf

Terraform destroy: what ever we mentioned in main.tf it will destroy

GIT

It is a centralized source code repository.

It manintains versions of the code

1. Git clone <repo name> It will copy remote Git repository to local computer.
2. Git checkout -b <branch name>
3. Add/modify/update/delete the code/scripts
4. Git add .(modified files add to staging area)
5. Git commit – m “Reason for the modification”
6. Git push origin master : it pushes changes to the centralized repository.

Git status

Git diff

1. Git clone: delete and copy from the origin
2. Git pull : only copies updated files in the local

GIT branching strategy:

Master

development

feature

hotfix/bug

API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software.