* Hi, my name is Sai. I have around 5 years of experience in IT as a DevOps engineer/ Cloud Engineer/Build and release engineer and I am currently working as a DevOps Engineer in NTT data. ***As a DevOps Engineer/Cloud Automation Engineer, I majorly worked on Automating the services for our deployments* and, I help teams in designing Solutions on Cloud. Coming to Cloud providers I worked on both AWS & Azure, and I have a good understanding of GCP.**  I help teams to implement new applications on the cloud and work on migrations, Planning, and execution. I have hands-on experience in infrastructure as a code using **Terraform, CFT & Azure ARM templates**which automatically creates a stack with EC2, S3, Load Balancers, Autoscaling.
* I have created several docker images by developing docker files as per the need of the Dev, QA, and UAT environments and help orchestrate these using Kubernetes. I have exposure working with AKS & ACR in Azure, EKS & ECR in AWS.
* I work very closely with Development teams and involve and help in all phases of the SDLC Process, Starting from Design Phase to the Post Implementation, Automation & Support Phases.
* I have good experience in writing shell and python scripts for Unix platforms like ubuntu, fedora, Sun Solaris, etc. I have identified several manual processes and automated with my scripting skills. I configured Jenkins and deployed several build pipelines thereby reducing the effort of the manual tasks. I have configured several central libraries in groovy scripting to reuse the functionality for different configurations and platforms.
* Coming to Monitoring I have experience working with Nagios, Splunk, ELK (Elastic Search/Logstash/Kibana), Prometheus, Grafana, New Relic.
* Implemented Multiple deployments like Blue Green**Pattern**, Canary, Atomic, Active Passive Deployments, and rollbacks with zero downtime
* Overall, this is my experience working at a high level. I work with an enterprise application group and actively support Corporate DevOps.
* I can dive deep for more details on tools if required.

Kubernetes, at its core, Container orchestrator, Kubernetes job is to start/stop container-based applications based on requirements of system administrator or developer

**Benefits of Kubernetes:**

Speed of deployment

Ability to absorb change quickly

Ability to recover quickly

Hide infrastructure complexity in the cluster

**Kubernetes principles:**

Desired state/desired configuration

Controllers/control logs

One master/the API server

Pods is a construct that represents one or more containers inside a Kubernetes cluster

*Shell scripting cheat sheet:*

* $1, $2, $3, ... are the [positional parameters](https://www.gnu.org/software/bash/manual/html_node/Positional-Parameters.html).
* "$@" is an array-like construct of all positional parameters, {$1, $2, $3 ...}.
* "$\*" is the IFS expansion of all positional parameters, $1 $2 $3 ....
* $# is the number of positional parameters.
* $- current options set for the shell.
* $$ pid of the current shell (not subshell).
* $\_ most recent parameter (or the abs path of the command to start the current shell immediately after startup).
* $IFS is the (input) field separator.
* $? is the most recent foreground pipeline exit status.
* $! is the PID of the most recent background command.
* $0 is the name of the shell or shell script.

***NTT data role description***

As a DevOps/Cloud Engineer, we assisted in migrating our on-premises servers to the cloud, majorly focusing on design, implementation, management, and automation of the Azure/AWS public cloud offerings for our client and providing proof of concepts for Azure/AWS Cloud Service deployments. Focused on Configuration, SCM, Build/Release Management, Infrastructure as a code (IAC) and as Azure DevOps/Jenkins operations Production and cross-platform environments.

***Rane Pharmaceuticals Inc.,***

*As a DevOps Engineer/Cloud Automation Engineer, majorly worked on Automating the services for our deployments.*Primary tasks include **Build** and **Deployment** of the **Java applications** onto different environments like**Dev, QA, UAT, and Production in AWS and** focused on creating Cloud infrastructure and implementing CI/CD pipeline using **GitHub, Jenkins, Maven, Chef on AWS and handling the issues in AWS cloud environments**

**Salary expectations:**

* **While I am certainly flexible, I am looking to receive between $87,000 and $92,000 annually. Due to my skill set and experience level, I feel that this is a comfortable and appropriate range for my work.**
* **Before I answer, I would like to ask a few more questions to get a better idea of what the position entails. That way, I can provide a more realistic expectation**
* **I am seeking a position that pays between $87,000 and $92,000 annually, but I am open to negotiate salary depending on benefits, bonuses, equity, stock options and other opportunities.**

Docker:

The Docker image help to create Docker containers. You can create the Docker image with the build command. Due to this, it creates a container that starts when it begins to run. Every docker images are stored in the Docker registry.

**Docker daemon or Docker engine represents the server.** The **docker daemon** and the **clients** should be run on the same or remote host, which can communicate through command-line client binary and full RESTful API.

You can communicate between **Docker client** and **Docker Daemon** with the combination of **Rest API, socket.IO, and TCP**

|  |  |
| --- | --- |
| What command should you run to see all running container in Docker?  Write the command to stop the docker container  **What is the command to run the image as a container?**  **Where the docker volumes are stored?** | $ docker ps  $ sudo docker stop container name  $ sudo docker run -i -t alpine /bin/bash  You need to navigate: /var/lib/docker/volumes |

**Docker states and Docker Events** are used to **monitoring** docker in the production environment.

|  |  |
| --- | --- |
| Docker life cycle:   * Build * Pull * Run | **Three components of Docker Architecture**   * Client * Docker-Host * Registry |

These are major advantage of using Docker.

* Offers an efficient and easy initial set up
* Allows you to describe your application lifecycle in detail
* Simple configuration and interacts with Docker Compose.
* Documentation provides every bit of information.

**Day to day operations**

  - I am a Developer Administrator, 50% of doing Developments using DevOps Technologies & 50% doing the Production Support.

  - In DevOps Work, I receive Several tickets on New Automations, Changes to Pipelines, Executing & Monitoring Monthly/Weekly/Daily Builds.

  - I do the Release Management work using Jenkins.

  - I am responsible for Ensuring that Monitoring is Done on Each Asset and Ensure we are getting the logs.

  - 25% of my work is on exploring new Technologies and doing POC on new tickets and Provide Feedback and Make Production-Ready Applications.

  - Day starts with Checking the tickets, Responding to them, and Gatekeeping Production Environment.

Jenkins is used to continuously monitor the large code base in real-time. It enables developers to find bugs in their code and fix them. Email notifications are made to the developers regarding their check-ins as a post-build action.

**Advantages of Jenkins are as follows:**

* Build failures are cached during the integration stage.
* Notifies the developers about build report status using LDAP (Lightweight Directory Access Protocol) mail server.
* Maven release project is automated with simple steps.
* Easy bug tracking.
* Automatic changes get updated in the build report with notification.
* Supports Continuous Integration in agile development and test-driven development.

**Plugins in Jenkins includes:**

* Gits
* Maven 2 Project
* HTML Publisher
* Copy Artcraft
* Join
* Green Balls
* Amazon EC2

**you can follow the below-mentioned steps to start Jenkins:**

1. Open Command Prompt
2. From the Command Prompt browse the directory where Jenkins. war resides
3. Run the command given below:

D:\>Java –jar Jenkins.war

\*\***CI/CD**

1. One of my key responsibilities when it’s comes to CI/CD is to improve the process using Automation.

2. I help the teams in Creating Automated Parameterized Pipelines

3. I have experience working with Groovy pipeline syntax, and also created scripted/Declarative Pipelines.

4. I have experience making the CI/CD automated by creating the webhooks for seamless integration on version control and CI/CD.

5. I have strong experience in Jenkins in configuring several jobs, freestyle projects, conditional pipelines, building with multiple branches, and so on.

6. Also have strong experience in Azure DevOps using Azure Repos (Git)/ Azure Build and Release pipelines / Azure Test Plans for automated testing/ Azure Artifacts.

7. In the Build process I used various technologies

     e.g.: - .net based applications we used MSBuild and Nuget based .net artifacts.

          Angular Based Apps we used npm, here we do npm install, npm build-prod

          Java-based we used remote maven artifact feeds., here we do mvn validate, mvn compile, mvn test, mvn install.

           we can configure a local or remote maven depending upon the requirement.

          Python-based using Pip.

\*\***AWS**

1. AWS - In AWS I worked mainly on services like EC2, S3, VPC, Route 53, ELB, Classic Load Balancer, ECR (Elastic Container Registry), ECS (Elastic Container Service), AMI (Amazon Machine Images), cloud watch, cloud trail, configuring the Lambda functions to make them event-driven, or schedule-driven.

2. We support the teams who request a cloud infrastructure in our company standard.

Our Company standards include hosting services only in specific regions, having a Jump Server to Access any Private instances, etc.

3. I worked on Monitoring AWS using the Cloud watch, Cloud trail, VPC Flow Logs, and identifying the Misconfigurations on Environments using a tool called Prisma Cloud.

4. I also do the Infrastructure as a Code using the Cloud Formation as well as Terraform. Mostly Terraform is what I work and have expertise on.

5. In Addition to this, we support some services like API Gateway, AWS Lambda, AWS Quick Sight, Sage Maker, etc.

6. I have been involved in the Complete Implementation of AWS EKS/ECR etc. to Manage our Kubernetes Environments.

**AWS Lambda Functions:**

I wrote a Python script for AWS Firewall enforcer. It is a Lambda function which runs once in every 24 hours. Security team and information security team will update the queue rates, which set of firewalls should be applicable in AWS VPC's which are added to GIT repository. My Lambda script is python based and it clones the GIT repository and parses those JSON files and uses the BOTO to make API calls to VPC and enforce the firewall rules. If any of the persons in the team opens a port to the public, my enforcer will apply the secured rules and makes the port private according to the pre-defined rules.

Another LAMBDA function which i wrote is connect to an RDS instance and to run a known long running query and if the query response time is more than the threshold it will send the CloudWatch metric that would trigger an SNS topic which will send an Email Notification that our database is getting overloaded. It will add a read replica for the RDS instance.

**\*\*Ansible**

1. I have experience working with Ansible, Ansible is used in our case for both configuration Management and Application Automation.

Configuration Management - if we want to change one simple file like server.xml in tomcat for 100 servers, you need a powerful config mgmt. a tool to help you and we use Ansible in this case.

Application Automation - we use this to deploy using Jenkins, we use this for creating and running Playbooks to install and configure the application and do the builds.

basically, we create an inventory file and group the hosts and trigger based on the inventory.

**\*\*Docker & Kubernetes: -**

1. My Responsibility in Docker is to work with Various teams and understand the Application and convert the VM based application to a docker image and scale them and deploy them on the Kubernetes cluster like EKS(Elastic Kubernetes Service)/AKS(Azure Kubernetes cluster)

2. I know about Creating Pods, Deployments, Replica sets, Pod Autoscaling, Creating Services (NodePort/CLusterIP/Load Balancer), Kubernetes cronjob, job, etc. we can do.

Apart from this, I have good knowledge of working with YAML as well.

**\*\*TERRAFORM: -**

Kubernetes cluster was in a unhealthy state and I got the notification from the monitoring tool and from there I investigated the issue where I figured out

Our application got deployed successfully and it was our EFT(early field trail) but we could not be able to access one of the API, so I have investigated the issue and figured out as proxy network issue, it was unable to reach the external API. I have added Kubernetes network by using CIDR notation to no proxy which solved the issue. This was my recent issue that I faced and where I got good appreciation from the client for handling this well at a right time

What are your strengths and weakness?

I am person who can do multi tasking and handle the situations well and deliver the tasks at right time with no procrastination.\

I believe more in learning; I am fascinated to learn new technologies and

I can’t say NO to other

**Interview questions:**

AWS RDS events, how to handle events in RDS

AWS Lambda functions and the events that triggered

Cloud Formation Templates Anatomy

AWS Lambda

What type of projects you used in JENKINS?

JAVA Spring boot

AWS BOTO3

CI/CD pipeline

Python modules,