Projects

1. speaking about me ,I have 3 years of experience working in AWS,Jenkins,Linux,Chef,Ansible

2. I am currently working for techvision-us at there client location in HealthPartners for there cloud engineering team. which is a centralized team to support middleware technologies such as apache, tomcat, jboss, docker.

3. speaking about Healthpartners,

Healthpartners is an integrated, nonprofit health care provider and health insurance company located in Minneapolis.

It offers care, coverage, research and education to its members and patients in Minnesota, Western Wisconsin and Northern IOWA serving approximately 1.8 million medical and dental care to their plan members.

4. Healthpartners started adapting public/private cloud and devops culture and slowly moving away from traditional approach and investing heavily  to move to Agile, CI/CD and rapid provision of environments to developers in AWS and automating installations using ansible and chef.

HEALTHPARTERS

At HealthPartners, I’m part of cloud engineering team which is responsible for handling infrastructure and support for middleware services. Cloud engineering team handles all work related to AWS, **Enterprise Linux, Jenkins, GitLab, Artifactory, Chef, Ansible, Terraform, kubernetes, Docker, Apache, Tomcat, Jboss EAP, WebLogic Server and many more.**

Our team works on creating server infrastructure in AWS, build chef cookbooks and ansible playbooks for middleware components for apache,tomcat, jboss, kubernetes, docker. Install, configure, manage, upgrade gitlab, artifactory,chef server, ansible tower, dynatrace. Setup email alerts or SMS notifications for CPU utilization, memory usage, disk utilization, application health. Configure dashboards for monitoring server infrastructure using prometheus, grafana,dynatrace, and splunk.

Prometheus is our main monitoring tool for our on-premise/AWS server infrastructure, dynatrace for monitoring kubernetes/docker infrastructure and splunk is used for analyzing logs.

My role in this team as cloud engineer is to build AWS infrastructure for development, integration, staging and production environments using **VPC, EC2, S3, ELB, EBS, RDS, IAM, EKS CloudFormation, Route 53, CloudWatch, CloudFront, CloudTrail.** Created automation using cloudformation or terraform templates for rapid provisioning of environments integrating with jenkins work flows. Manage cookbooks or playbooks to install apache, Tomcat, jboss kubernetes/docker, continuous integration and delivery with jenkins and configure dashboards in **Dynatrace** to monitor server and application health in Kubernetes, configure dashboards in grafana, splunk, cloudwatch.

Create and maintain documentation for all activities on our cloud engineering internal wiki site in confluence. Participated in pager rotation for 24/7 support within our team.

OPTUM

At Optum I had worked on their A4ME team which is responsible for building services to provide different functionalities as widgets to host applications to enhance the features of old existing applications. These widgets are developed as independently testable applications.

Provider messaging widget service was developed for providing an effective communication channel for plan members and is used by ISET application.IBAAG widget service was developed for sending benefit information through secure messaging. BCM widget service was developed for sending benefit codes through secure messaging.

Services are developed with Java/J2EE platform using springboot framework and MySQL as database, packaged as docker containers and deployed to kubernetes/docker environment following agile development model in two weeks sprint iterations.

Server infrastructure was built on AWS cloud platform for both non-prod and prod environments. My role in this team as devops engineer is to support AWS infrastructure, develop cookbooks and recipes to install and manage kubernetes/docker, continuous integration and delivery with jenkins and configure dashboards in **Dynatrace** to monitor server and application health in Kubernetes.

BESTBUY

At bestbuy I had worked for their activated devices team which is responsible for mobile sales and activations with different carriers through bestbuy cart or in store. Mobile purchase involves many work flows for in store or cart as new line, existing line, choosing device, choosing plan, eligibility on device, protection plan, add on services like upgrade, reminder services and billing.

Services are developed with Java/J2EE platform using springboot framework and deployed on Tomcat server and Oracle as database in agile/scrum environment following two weeks sprint iterations.

Server infrastructure was built on AWS for both non-prod and prod environments. My role in this team as devops engineer is to support AWS infrastructure, continuous integration and delivery with jenkins, infrastructure monitoring through AWS cloudwatch and nagios, manage user administration in AWS.

1. Day to Day Activities.

We will have a daily standup @9AM where we discuss about the stories that are completed previous day or update progress of the current story that had been made .if we have a dependency and waiting for other team to complete then we will move the story to waiting queue and pickup a new story . Since we follow pair programming for our stories we partner with another team member in the team and continue working on it. Our on-call pager support for the team comes once in 4 weeks I will get my turn as primary or secondary. Most of the oncall issues that we receive are for out of memory for tomcat/jboss, disk space issues on filesystems or teams requesting help for assistance to diagnose issues.

2. My Role:

Since our team adapted devops culture , we follow agile,scrum and extreme programming methodologies , we are expected to work on all technologies used with in the team. My core competency areas are AWS,Jenkins,chef,ansible,terraform and prometheus I will pick work in those areas. our team also supports weblogic,activemq ,terracota but I have only minimal knowledge in those areas.

3. My Environment

Our team contains 10 team members and is primarily responsible for designing, development,monitor infrastructure around apache,tomcat and jboss EAP on AWS.

We use jenkins as our CI/CD tool, chef and ansible for configuration management, chef server and ansible tower for delivering cookbooks and playbooks to our AWS infrastructure,

cloudformation or terraform templates are used to build AWS infrastructure.

Cloudwatch and prometheus are used for monitoring and sending alerts to group email ,hipchat for non-production and to group email, hip-chat & pager-duty for production incidents.

We have 2 teams using apache(web & research) and have 8 sites running. Web team uses apache for displaying static assets like images ,photos and sorryserver pages when healthpartners.com or other internet sites are down (journeywell.com,regionshospital.com). Research team has many drupal,php and wordpress sites running. Each site have sandbox,development and production instances.

We have 4 teams using tomcat(web,research,ivr,choreo) and have 6 sites running. Web team uses tomcat for keysurvey app, research & ivr for in-house java applications and choreo for contract management application. Each site have save sandbox,development,test and production instances.

We have one team using Jboss EAP bundled with BXMS(BPMS/BRMS) primary used as rules engine and have sandbox,development,testing,uat and production environments.

Vendor call

------------

Vendor: Hi, this is tim for avnet technologies.

You: Hello Tim, How are you?

Vendor: I am good, How are you?

You: I am fine.

Vendor: Is this a good time to talk with you?

You: Yes , Go ahead.

Vendor: We have a position open for devops/cloud engineer with our client at location xxxx . are you interested.

You: Yes, I am actively looking for a job and i would be interested.

You: How much is billing for this position.

Vendor: Its $55 corp-corp

You: I will be interested if its at-least $60. I am not looking for anything less than that.

Vendor: Have you looked at the requirement.

You: Give me a minute, i will go through the requirement.

You: looks like client is looking for an experienced resource in technologies (read out technologies tell him ). tell them on technologies that you have worked and later technologies you know but not worked. tell him the technologies that you have never worked.

Vendor: what is your visa status.

You: I have green card.

Vendor: How did you got your Greencard?

You : I got greencard through my husband.

Vendor: Are you willing to relocate?

You: Yes, I am open for relocation. My preference is Midwest or Big cities in central or East.

Vendor: Why are you changing?

You: For better role and challenging opportunity. Looking to work on some new technology.

Vendor : Who is your employer?

You: Techvision-us

Vendor: Can you come for face to face interview?

You: No problem, I can come for face to face after clearing telephonic or skype call for final interview.

Vendor: How soon you are available.

You: I am required to provide a 2 week notice to my current client. once position is confirmed i will need 2 weeks time.

Vendor: please send me an updated resume.

You: sure, i will send you updated resume in a hour.

Vendor: what is your experience in AWS?

You: I have around 3 years experience in AWS jenkins and linux.

Vendor: can you provide two references

You: No problem , I will send you the reference details along with updated resume. You can contact references once the position is confirmed not before that.

You: which client is this requirement for?

Is this requirement for your direct client?

You: Are there any layers in between ?

Can you send me confirmation email containing details of vendor or vendors.

You: What is the interview pattern?

Will there be coding test.

do you know which coding site they normally use?

Do you know How many interview rounds will be there?

You

You: Thanks for considering me for this position. it will be a pleasure working with you on this position. hope it will be successful

| PRODUCT | EXPERIENCE |
| --- | --- |
| AWS | 3 years |
| Jenkins | 3 years |
| Linux | 3 years |
| **Docker, kubernetes** | 2 years |
| Chef, Ansible | 2 years |
| Bash, Python | 2 years |
| Jira | 3 years |
| Artifactory | 1 year |
| Prometheus, Dynatrace | 1 year |
| Nexus | 2 years |
| Terraform | 2 years |
| Tomcat,Apache,Jboss | 2 years |

**Agile Development Methodologies**

Agile software development refers to a group of software development methodologies based on iterative development, in which requirements and solutions evolve through collaboration between self-organizing cross-functional teams. There are many methodologies under agile such as Extreme Programming (XP), Scrum, Lean Development etc. In the projects I worked we followed a combination of XP working in pairs /scrum for weekly time boxing / Kanban for restricting work in progress approach to organize development into weekly or bi-weekly sprints which contains a daily standup followed by a retrospective of previous sprint and planning for next sprint.

Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted.

Other elements of extreme programming include: programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat management structure, code simplicity and clarity, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels. As an example, code reviews are considered a beneficial practice; taken to the extreme, code can be reviewed continuously, i.e. the practice of pair programming.

Scrum is an agile framework for managing work with an emphasis on software development. It is designed for development teams from three to nine members who break their work into actions that can be completed within time boxed iterations, called sprints (30 days or less, most commonly two weeks) and track progress and re-plan in 15-minute stand-up meetings, called daily scrums. Approaches to coordinating the work of multiple scrum teams in larger organizations include Large-scale Scrum (LeSS), Scaled Agile Framework (SAFe), scrum of scrums, and Scrum@Scale, among others.

Lean software development (LSD) is a translation of lean manufacturing principles and practices to the software development domain. Adapted from the Toyota Production System,it is emerging with the support of a pro-lean subculture within the Agile community. Lean offers a solid conceptual framework, values and principles, as well as good practices, derived from experience, that support agile organizations.

**Software Configuration Management (SCM) processes**

Software configuration management (SCM) is a set of processes, policies, and tools that organizes the development process to maintain the current state of the software, while enabling developers to work on new versions for features or fixes.

Following SCM practices will help to track and control changes to the code and if something goes wrong will help to determine what was changed and who changed it.

I have good experience using GIT in many projects and have good working knowledge of creating repositories, managing branches, creating tags and integrating with Jenkins to build continuous integration and delivery process performing integrated and unit tests.

**Chaos engineering**

Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system’s capability to withstand turbulent conditions in production.

As part of the DevOps movement, special attention is paid to the safe operation of computer systems, thus providing a sufficient level of confidence despite frequent releases. By contributing to the Devops Tool Chain, Chaos Monkey meets the need for continuous testing.

They are part of the pattern Design for failure, designed to support failure a computer application must be able to support the failure of any underlying software or hardware component.

The Simian Army is a suite of tools developed by Netflix to test the reliability, security, or resiliency of its Amazon Web Services infrastructure and includes the following tools

Chaos Monkey

The first tool developed by Netflix, it allows random selection of instances in the production environment and deliberately put them out of service.

Chaos Gorilla

At the very top of the Simian Army hierarchy, Chaos Gorilla, drops a full Amazon Availability Zone.

Latency Monkey

Introduces communication delays to simulate degradation or outages in a network.

Doctor Monkey

Performs health checks, monitoring performance metrics like CPU load to detect unhealthy instances for root-cause analysis and eventual fixing or retirement.

Janitor Monkey

Identifies and disposes unused resources to avoid waste and clutter.

Conformity Monkey

Tool that determines whether an instance is nonconforming by applying a set of rules on it. If any of the rules determines that the instance is not conforming, the monkey sends an email notification to the owner of the instance.

Security Monkey

Derived from Conformity Monkey, a tool that searches for and disables instances that have known vulnerabilities or improper configurations.

10-18 Monkey

Tool that detects problems with localization (l10n-i18n) for software serving customers across different geographic regions.

Chaos Toolkit

The Chaos Toolkit was born from the desire to simplify access to the discipline of Chaos Engineering and demonstrate that the experimentation approach can be done at different levels: infrastructure, platform but also application. The Chaos Toolkit is an open-source tool

**Information Technology Infrastructure Library (ITIL) framework**

ITIL is a set of detailed practices for IT service management (ITSM) that focuses on aligning IT services with the needs of business.

ITIL processes are grouped into stages: **Service Strategy, Service Design, Service Transition, Service Operation, and Continual Service Improvement**

1. ITIL Service Strategy: understands organizational objectives and customer needs.
2. ITIL Service Design: turns the service strategy into a plan for delivering the business objectives.
3. ITIL Service Transition: develops and improves capabilities for introducing new services into supported environments.
4. ITIL Service Operation: manages services in supported environments.
5. ITIL Continual Service Improvement: achieves services incremental and large-scale improvements.

**SRE practices**

Site Reliability Engineering (SRE) is a discipline that incorporates aspects of software engineering and applies that to IT operations problems. The main goals are to create ultra-scalable and highly reliable software systems. A site reliability engineer (SRE) will spend up to 50% of their time doing "ops" related work such as issues, on-call, and manual intervention. Since the software system that an SRE oversees is expected to be highly automatic and self-healing, the SRE should spend the other 50% of their time on development tasks such as new features, scaling or automation. The ideal SRE candidate is a highly skilled system administrator with knowledge of code and automation.

SRE is more focused on the system engineer role of core infrastructure and it is generally more applicable to a production environment. DevOps on the other hand is a practice used to automate and simplify the development teams and their non-production computing environments.

**Devops Culture**

DevOps (a clipped compound of "development" and "operations") is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.

The goals of DevOps span the entire delivery pipeline. They include:

1. Improved deployment frequency;
2. Faster time to market;
3. Lower failure rate of new releases;
4. Shortened lead time between fixes;
5. Faster mean time to recovery (in the event of a new release crashing or otherwise disabling the current system).

Simple processes become increasingly programmable and dynamic, using a DevOps approach. DevOps aims to maximize the predictability, efficiency, security, and maintainability of operational processes. Very often, automation supports this objective.

DevOps integration targets product delivery, continuous testing, quality testing, feature development, and maintenance releases in order to improve reliability and security and provide faster development and deployment cycles. Many of the ideas (and people) involved in DevOps came from the enterprise systems management and agile software development movements.