# Doozer Software ReactJS Developer

Job URL : <https://www.indeed.com/viewjob?jk=f86122fd7a2ed694&from=myjobs&tk=1efcs45ttqg1l800>

Technology Stack Being Used

**Application**

* Server - Redhat Enterprise Linux
* Front End – React JS (and/or) Xamarin for Mobile team.
* Back End - .NET CORE
* Architecture - Micro services, Cloud Native development skills
* Database – High Availability Microsoft SQL Server 2016 using EF Core
* Infrastructure/ Platform
* Containerized OpenShift, Kubernetes
* Container Platform - Docker
* Caching – Redis
* Secure coding practices
* Security – OAuth 2.0, data encryption and signing, (used OAuth 2.0, not sure about encryption & signing)
* Experience working on high availability, large concurrent user base applications

**CI/CD Pipeline Tools**

* **Automated Testing Frameworks: XUnit, NUnit, Selenium**
* **Source Code Management: Git, Bitbucket**
* **Build & CI - Bamboo Build**
* **Deployment - Bamboo Deploy**
* **Artifact Build – Docker**
* **Artifact Repository - Artifactory**
* **Build Scripting Language – bash scripting (maybe? I use Git Bash a lot), PowerShell**

**Other Tools**

* Development - Visual Studio Code, Visual Studio 2019
* API Tool – Postman
* API Build/Test – Mulesoft

I have Experience In

**Application**

* Frontend – ReactJS/Redux
* Architecture – Micro Services AWS
  + Code Pipeline
  + Cognito
  + Amplify
  + S3
  + Ec2
  + Lightsail
  + Codebuild
  + CodeDeploy
  + Code pipeline
* Security – Google OAuth2.0
* Backend – NodeJS
* Containerized AWS ECS (Elastic Container Service)
  + Same thing as Kubernetes, just AWS specific. It’s a container management service that will run, stop, and manage docker containers on a cluster.
  + Kubernetes is the same thing, just specific to AWS micro services.
  + Runs a cluster of ec2 instances pre-installed with docker. Handles installing containers, scaling, monitoring, and managing these instances both through an API and the AWS management console.
  + Task definition – the blueprint describing which docker containers to run and represents your app.
  + Task instance – an instance of a task definition, running the containers detailed within it. Multiple tasks can be created by one task definition, as demand requires.
  + Service – defines the minimum and maximum tasks from one task definition. Defines long running of the same task definition. This can be 1 running container or multiple running containers all using the same task definition.

**CI/CD Pipeline**

* Source Code Management – Git
* Build -CodeBuild
* Deployment – Code Deploy
* Artifact repository – s3
* Bash scripting – used in AWS shell scripts to send instructions to an instance at launch. used it before when setting up a lamp stack and needed php, apache, mysql on an instance, initiating httpd (apache) when instance starts, etc.

**Other Tools**

* **API Tool – Postman – Used for testing and modifying APIs. Used it to test/modify (API path & variables and logic to handle them) api built using AWS Lambda & NodeJS (Calculator).**