

Simple DevOps Project

Simple DevOps Project



kubernetes



GitHub



docker



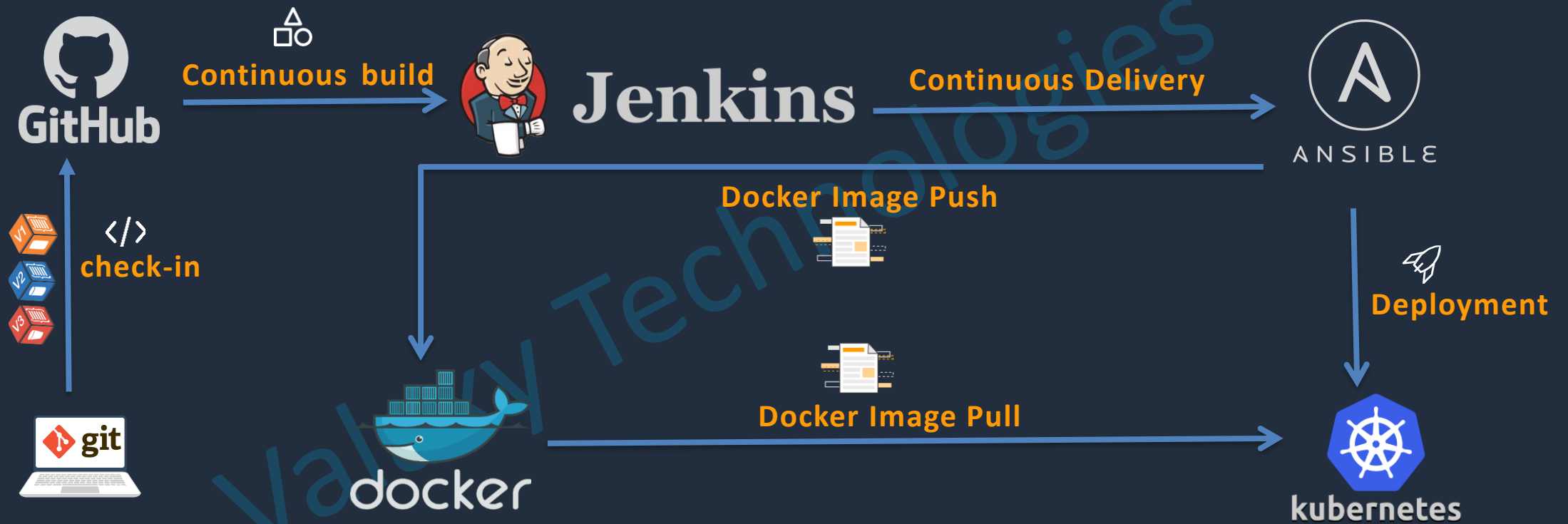
Jenkins

***m**aven*



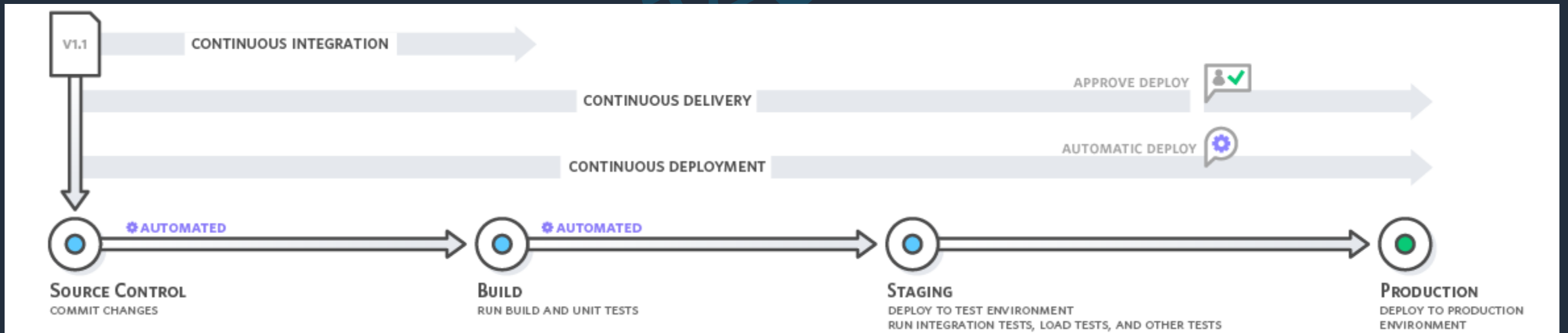
ANSIBLE

Simple DevOps Project

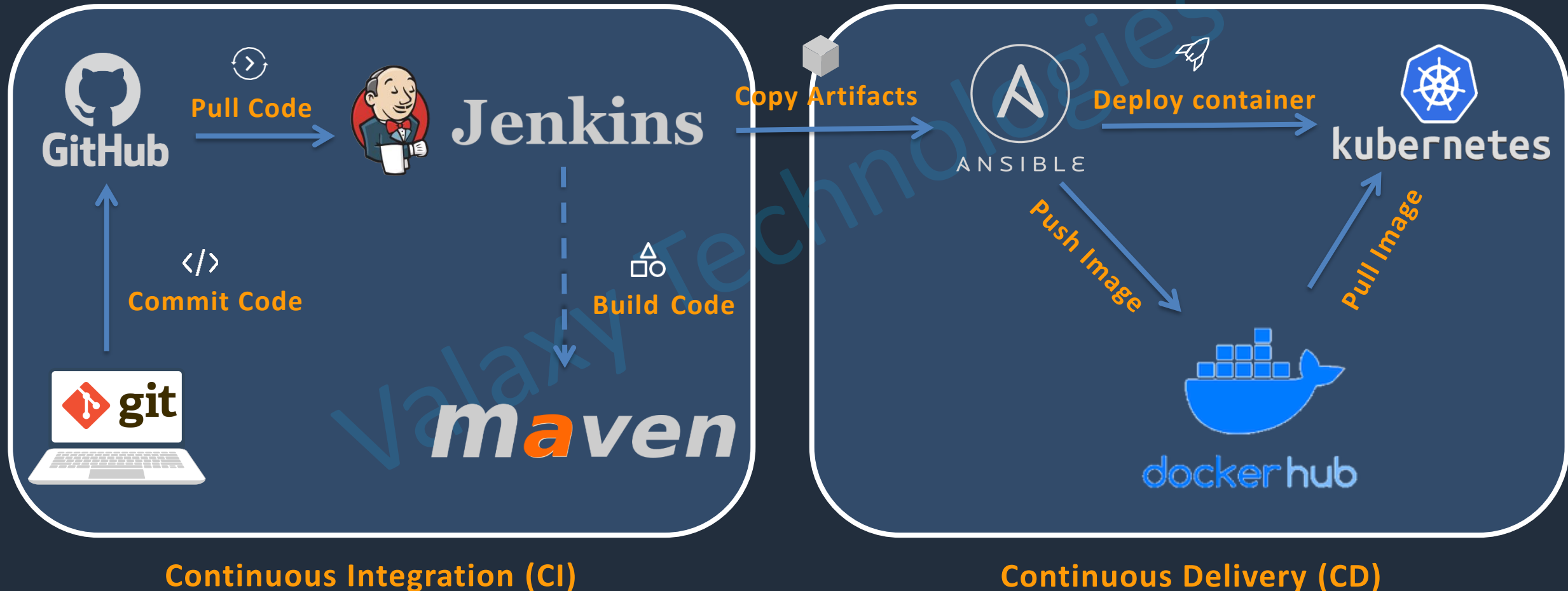


Simple DevOps Project

- Continuous Integration (CI)
- Continuous Delivery (CD)
- Continuous Deployment (CD)

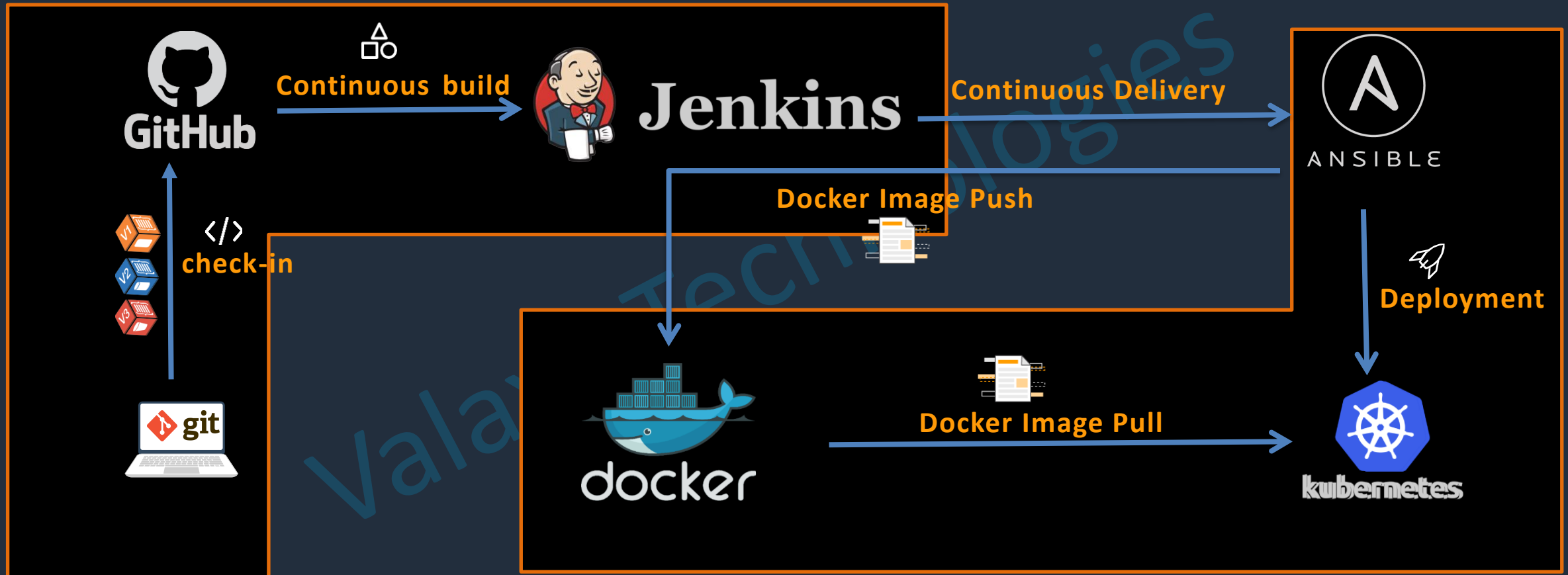


DevOps Project



Simple DevOps Project

Continuous Integration (CI)



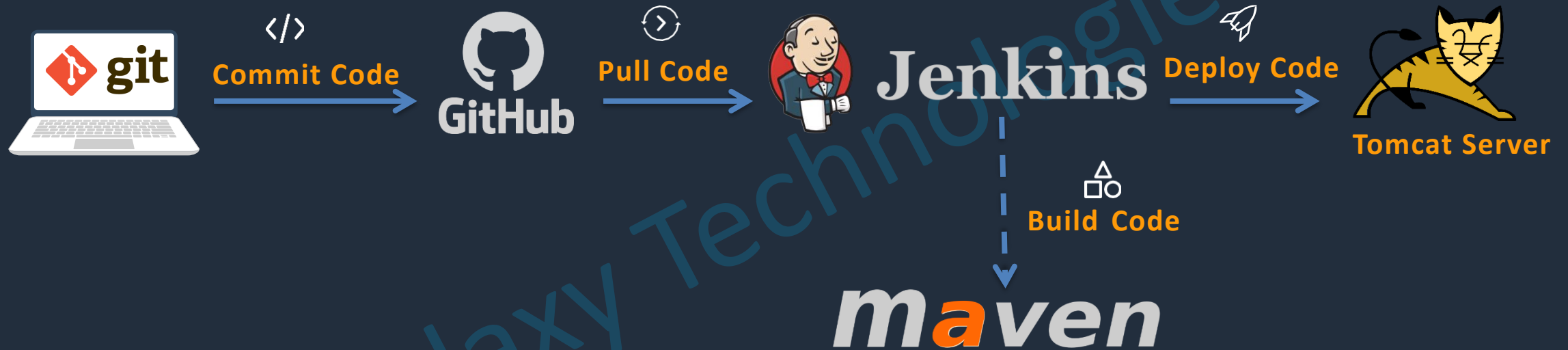
Continuous Delivery (CD)

Build and Deploy on Tomcat Server

➤ Setup CI/CD with GitHub, Jenkins, Maven and Tomcat

- Setup Jenkins
- Setup & configure Maven and Git
- Setup Tomcat Server
- Integrating GitHub, Maven, Tomcat Server with Jenkins
- Create a CI and CD job
- Test the deployment

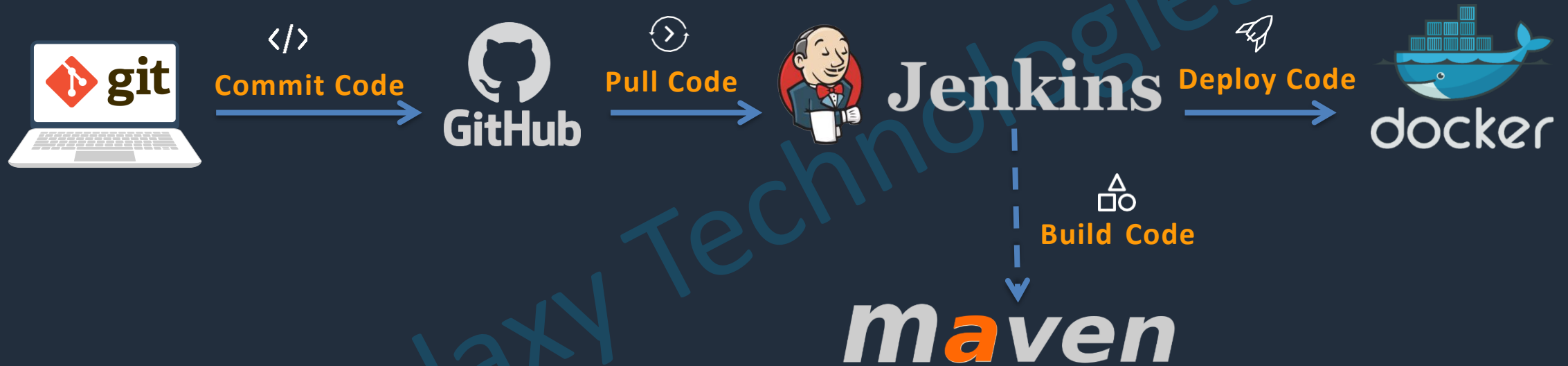
Deploy Artifacts on a Tomcat Server



Deploy Artifacts on a Container

- Setup CI/CD with GitHub, Jenkins, Maven and Docker
 - Setting up Docker environment
 - Write Dockerfile
 - Create an image and container on docker host
 - Integrate docker host with Jenkins
 - Create CI/CD job on Jenkins to build and deploy on a container

Deploy Artifacts on a Container



Deploy Artifacts on a Container

- CI/CD with GitHub, Jenkins, Maven, Ansible and Docker
 - Setup Ansible server
 - Integrate Docker host with Ansible
 - Ansible playbook to create image
 - Ansible playbook to create container
 - Integrate Ansible with Jenkins
 - CI/CD job to build code on ansible and deploy it on docker container

Deploy Artifacts on a Container



Deploy Artifacts on Kubernetes

- CI/CD with GitHub, Jenkins, Maven, Ansible and Kubernetes
 - Setup Kubernetes (EKS)
 - Write pod, service and deployment manifest files
 - Integrate Kubernetes with Ansible
 - Ansible playbooks to create deployment and service
 - CI/CD job to build code on ansible and deploy it on Kubernetes

Deploy Artifacts on Kubernetes



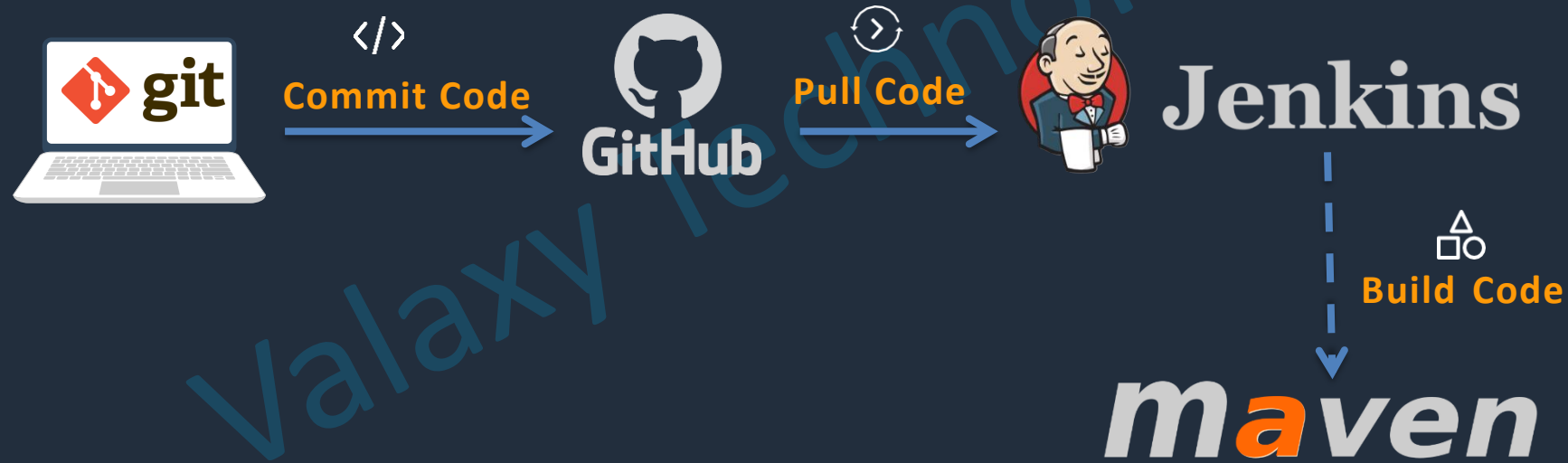
Resources Before Start

- An AWS account
- An GitHub account
- MobaXterm / Putty
- Git Bash setup
- Documentation: <https://github.com/yankils/Simple-DevOps-Project>
- Source Code: <https://github.com/yankils/hello-world>

Quick tips to learn fast

- Watch video with 1.5x speed
- Watch complete video or topic
- Understand logical flow
- Search in Q&A for your query
- Create a pull request if you find any improvements
- Connect with me over the LinkedIn or slack channel
- Request you to leave rating and review

Build Code



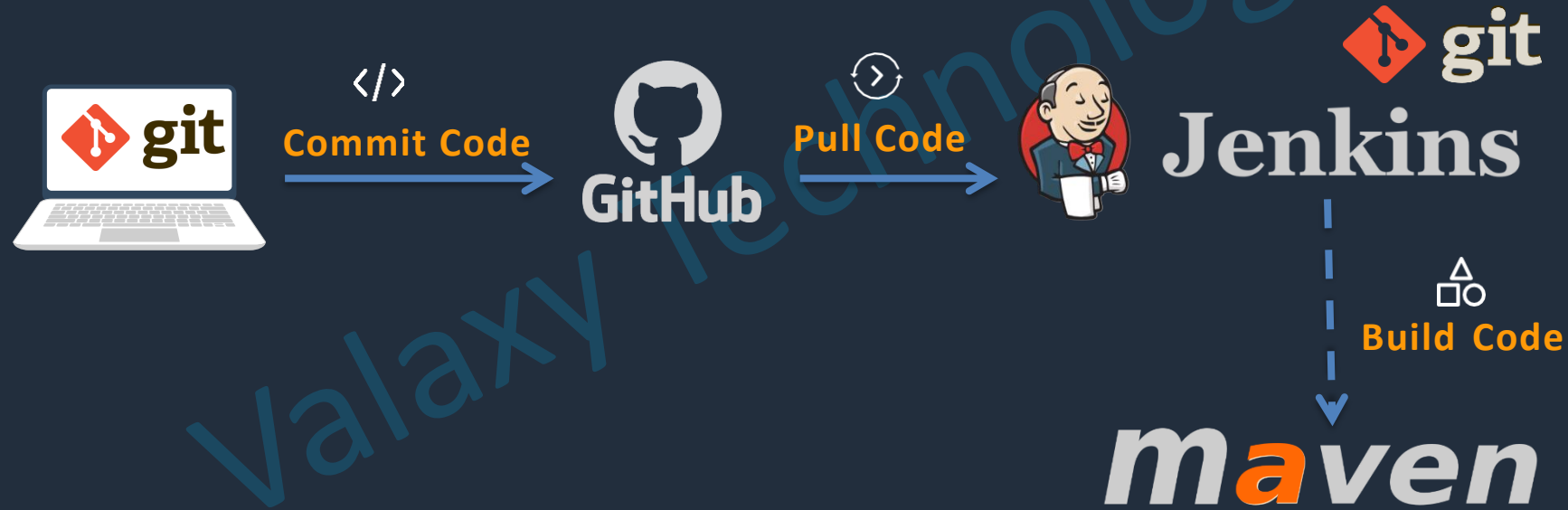
Setup Jenkins Server

- Setup a Linux EC2 Instance
- Install Java
- Install Jenkins
- Start Jenkins
- Access Web UI on port 8080



Jenkins

Build Code



Integrate Maven with Jenkins

- Setup Maven on Jenkins Server
- Setup Environment Variables
 - `JAVA_HOME`, `M2`, `M2_HOME`
- Install Maven Plugin
- Configure Maven and Java



Jenkins



Build Code



maven

Integrate GitHub with Jenkins

- Install Git on Jenkins Instance
- Install GitHub Plugin on Jenkins GUI
- Configure Git on Jenkins GUI



Deploy Artifacts on a Tomcat Server



Setup Tomcat Server

- Setup a Linux EC2 Instance
- Install Java
- Download and configure Tomcat
- Start Tomcat Server
- Access Web UI on port 8080



Tomcat Server

Integrate Tomcat with Jenkins

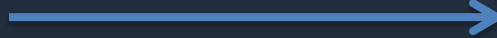
- Install “Deploy to container”
- Configure tomcat server with Credentials



Jenkins



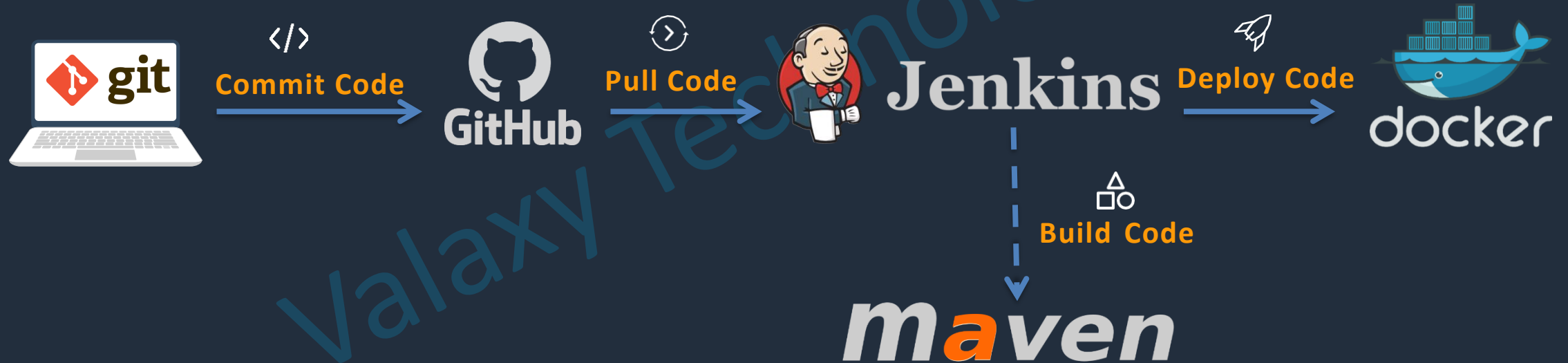
Deploy Code



Deploy Artifacts on a Tomcat Server



Deploy on a Container

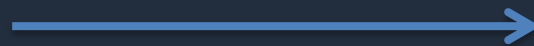


Integrate Docker with Jenkins

- Create a dockeradmin user
- Install “Publish Over SSH” plugin
- Add Dockerhost to Jenkins “configure systems”



Jenkins



Setup Docker Host

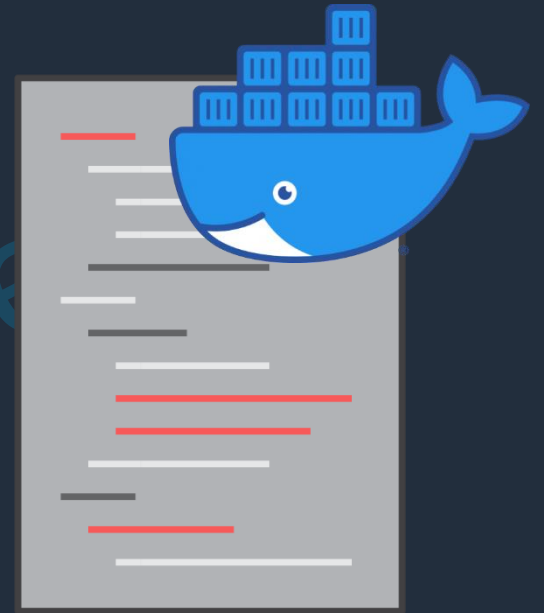
- Setup a Linux EC2 Instance
- Install docker
- Start docker services
- Basic docker commands



docker

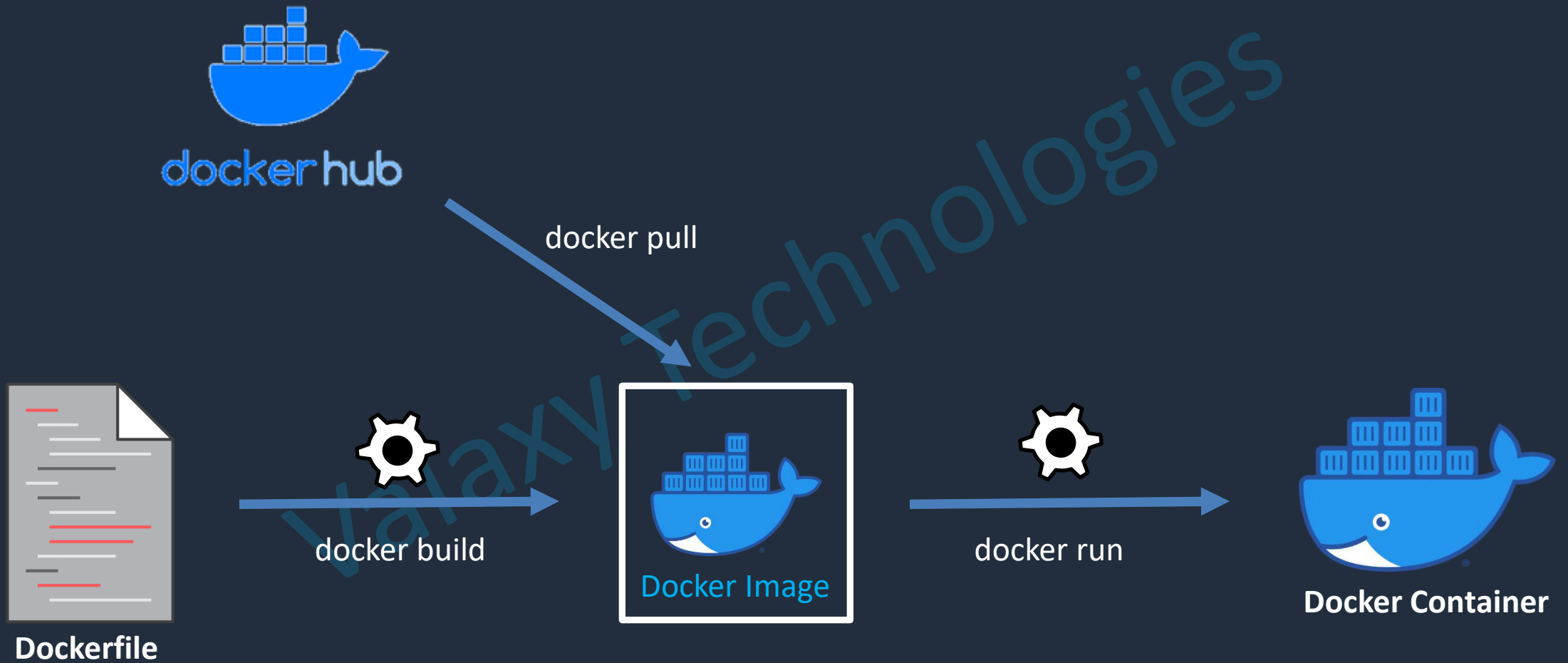
Valaxy Technology

Write
Your 1st



Docker File

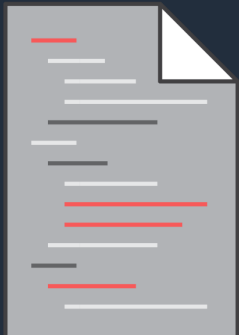
How to create Dockerfile



How to create docker container



docker pull



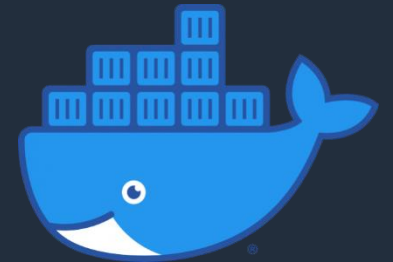
Dockerfile



docker build



docker run



Docker Container

Write Your 1st Docker File

- **FROM:** To pull the base image
- **RUN:** To execute commands
- **CMD:** To provide defaults for an executing container
- **ENTRYPOINT:** To configure a container that will run as an executable
- **WORKDIR:** To sets the working directory
- **COPY:** To copy a directory from your local machine to the docker container
- **ADD:** To copy files and folders from your local machine to docker containers
- **EXPOSE:** Informs Docker that the container listens on the specified network ports at runtime
- **ENV:** To set environment variables



Dockerfile

Install tomcat on Centos

- Pull centos from dockerhub
- Install java
- Create /opt/tomcat directory
- Change work directory to /opt/tomcat
- Download tomcat packages
- Extract tar.gz file
- Rename to tomcat directory
- Tell to docker that it runs on port 8080
- Start tomcat services

- **FROM**
- **RUN**
- **RUN**
- **WORKDIR**
- **ADD /RUN**
- **RUN**
- **RUN**
- **EXPOSE**
- **CMD**



Dockerfile

DockerFile

- FROM centos
- RUN yum -y install java
- RUN mkdir /opt/tomcat/
- WORKDIR /opt/tomcat
- ADD <https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.54/bin/apache-tomcat-9.0.54.tar.gz> /opt/tomcat
- RUN tar xvfz apache*.tar.gz
- RUN mv apache-tomcat-9.0.54/* /opt/tomcat
- EXPOSE 8080
- CMD ["/opt/tomcat/bin/catalina.sh", "run"]



Dockerfile

Deploying as a Container



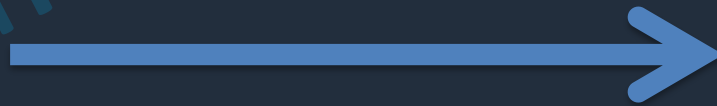
Manage DockerHost with Ansible

- **On Docker Host**

- Create ansadmin
- Add ansadmin to sudoers files
- Enable password based login

- **On Ansible Node**

- Add to hosts file
- Copy ssh keys
- Test the connection



Integrate Ansible with Jenkins



Jenkins



ANSIBLE

Valaxy technologies

Deploy Ansible playbook

- Remove existing container
- Remove existing image
- Create new container



Prepare Ansible Server

- Setup **EC2** instance
- Setup **hostname**
- Create **ansadmin** user
- Add user to **sudoers** file
- Generate **ssh keys**
- Enable **password based** login
- Install **ansible**



AN S I B L E

Deploying on Kubernetes



Setup Kubernetes

Deployment Tools:

- Bootstrapping clusters with **kubeadm**
- Installing Kubernetes with **kops**
- Installing Kubernetes with **Kubespray**

Managed Services:

Amazon **EKS**

Microsoft **AKS**



kubernetes

Kubernetes Setup Methods

- kubeadm
 - kops
 - Kubespray
- } Deployment Tools
- Amazon EKS
 - Microsoft AKS
- } Managed Services



kubernetes

EKS Setup

- Launch EC2 instance - **Bootstrap**
- Latest version of **AWSCLI**
- Setup **kubect**l
- Setup **eksct**l
- Create **IAM** role
- **Create** a cluster
- **Validate** cluster
- **Delete** cluster



kubernetes

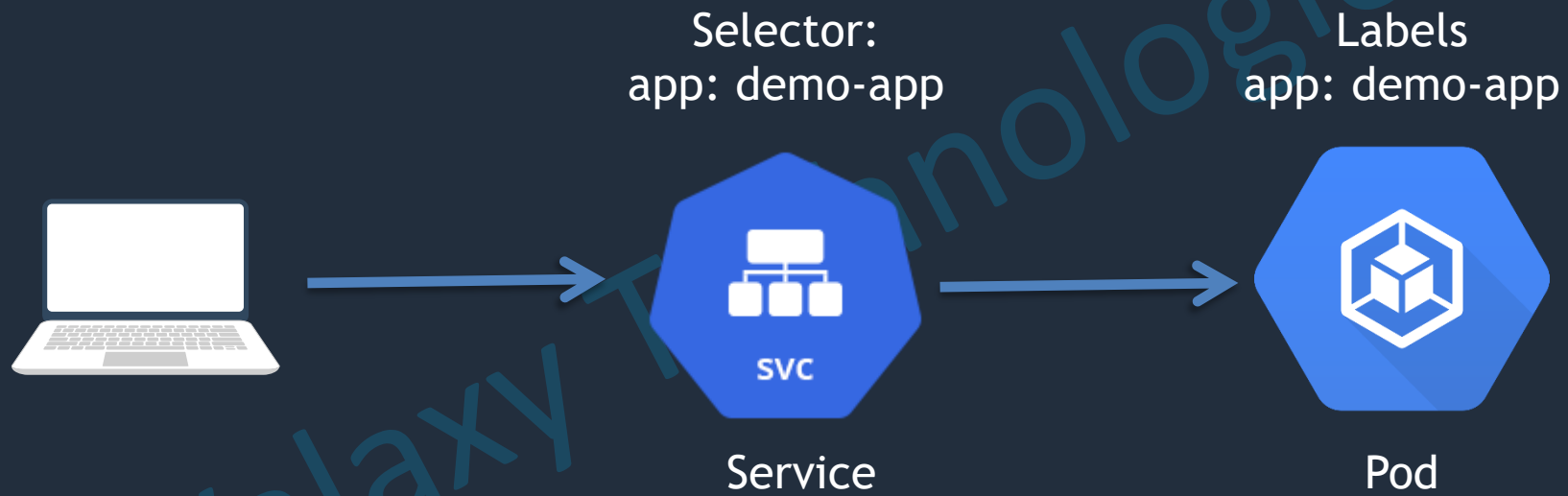
What do we cover in section

- Setup Kubernetes in EKS (using eksctl)
- Deploy demo app through kubectl command
- Who to write your 1st manifest file
- Manifest files for our register app
- Integrate Kubernetes with Ansible
- Create ansible playbook for deployments
- Create Jenkins

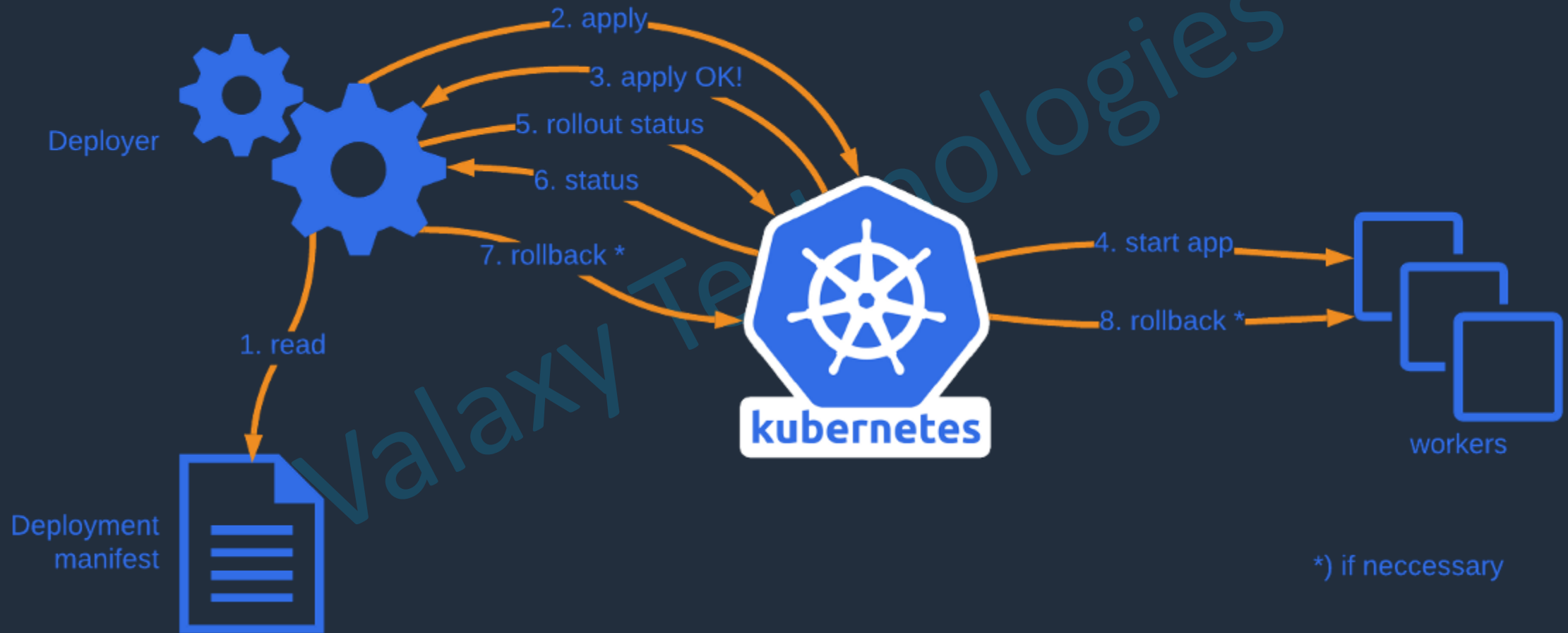


kubernetes

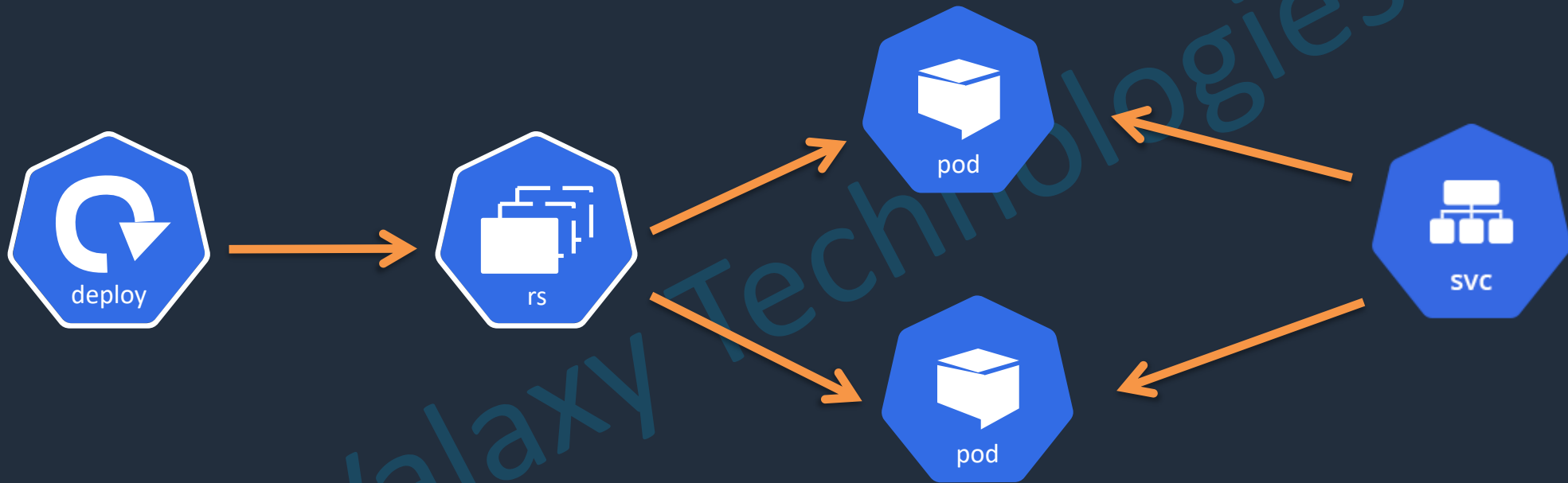
Setup Pod and Service



Setup Pod and Service



Create a Pod



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: valaxy-regapp
  labels:
    app: regapp
```

```
spec:
  replicas: 2
  selector:
    matchLabels:
      app: regapp
```

```
template:
  metadata:
    labels:
      app: regapp
  spec:
    containers:
      - name: regapp
        image: valaxy/regapp
        imagePullPolicy: Always
        ports:
          - containerPort: 8080
```

```
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 1
```

Deployment name and Deployment label

Create 2 pods from the pod template

Pod definition

template to create a pod
image name

Service file

```
apiVersion: v1
kind: Service
```

Resource Type

```
metadata:
  name: valaxy-service
  labels:
    app: regapp
```

Service name and label

```
spec:
  selector:
    app: regapp
```

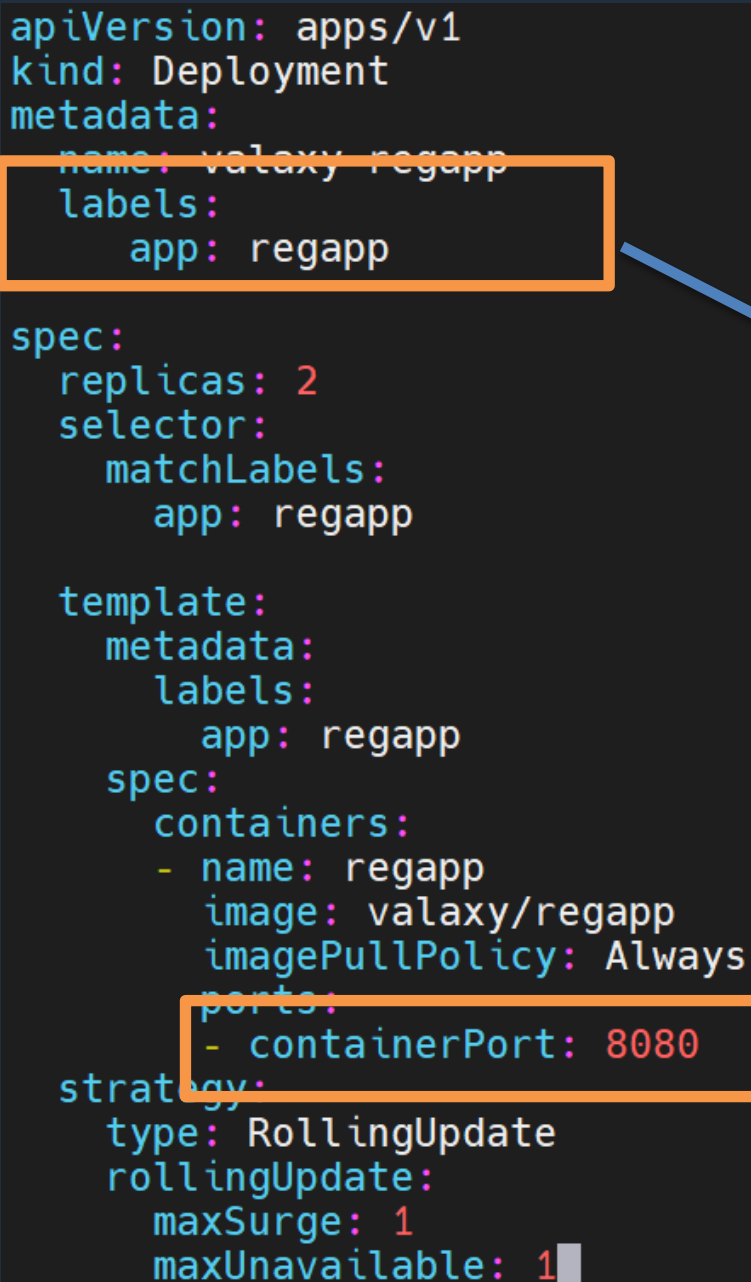
To which deployment it can send traffic

```
ports:
  - port: 8080
    targetPort: 8080
```

```
type: LoadBalancer
```

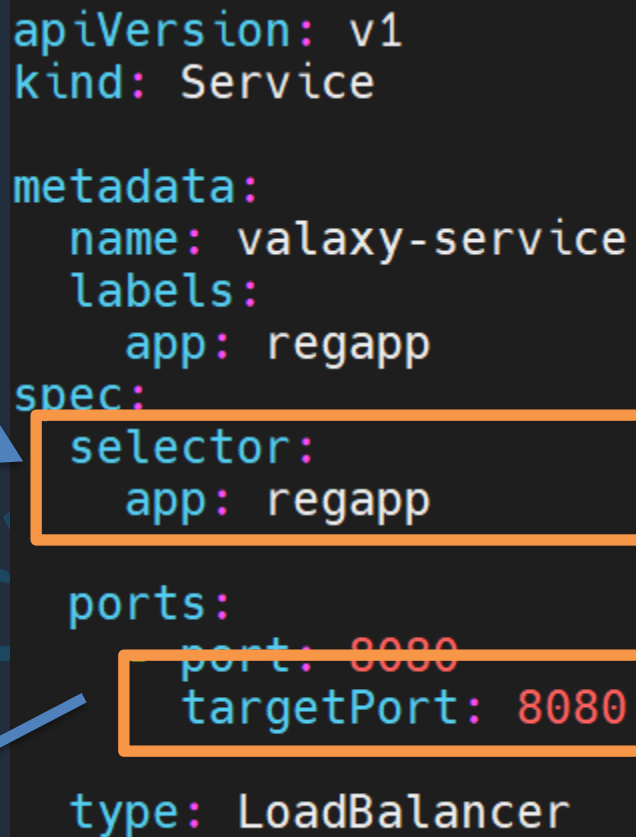
What is the service type

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: valaxy-regapp
  labels:
    app: regapp
spec:
  replicas: 2
  selector:
    matchLabels:
      app: regapp
  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
        - name: regapp
          image: valaxy/regapp
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
      strategy:
        type: RollingUpdate
        rollingUpdate:
          maxSurge: 1
          maxUnavailable: 1
```



The diagram illustrates the relationship between a Deployment and a Service. A Deployment resource is shown on the left, with its 'labels' field (name: valaxy-regapp, app: regapp) highlighted in an orange box. An arrow points from this box to the 'selector' field of a Service resource on the right, which is also highlighted in an orange box. Another arrow points from the 'ports' field of the Deployment (specifically the containerPort: 8080) to the 'targetPort' field of the Service, which is also highlighted in an orange box. The Service resource is configured with 'type: LoadBalancer'.

```
apiVersion: v1
kind: Service
metadata:
  name: valaxy-service
  labels:
    app: regapp
spec:
  selector:
    app: regapp
  ports:
    port: 8080
    targetPort: 8080
  type: LoadBalancer
```



The diagram illustrates the relationship between a Service and a Deployment. A Service resource is shown on the right, with its 'selector' field (app: regapp) highlighted in an orange box. An arrow points from this box to the 'labels' field of a Deployment resource on the left, which is also highlighted in an orange box. Another arrow points from the 'ports' field of the Service (specifically the targetPort: 8080) to the 'ports' field of the Deployment, which is also highlighted in an orange box. The Service resource is configured with 'type: LoadBalancer'.

Integrate Kubernetes with Ansible

- **On Bootstrap server**

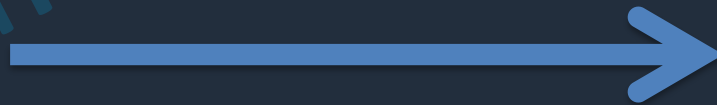
- Create ansadmin
- Add ansadmin to sudoers files
- Enable password based login

- **On Ansible Node**

- Add to hosts file
- Copy ssh keys
- Test the connection



ANSIBLE



kubernetes

Deploying as a Pod



Thank you

Valaxy Technologies