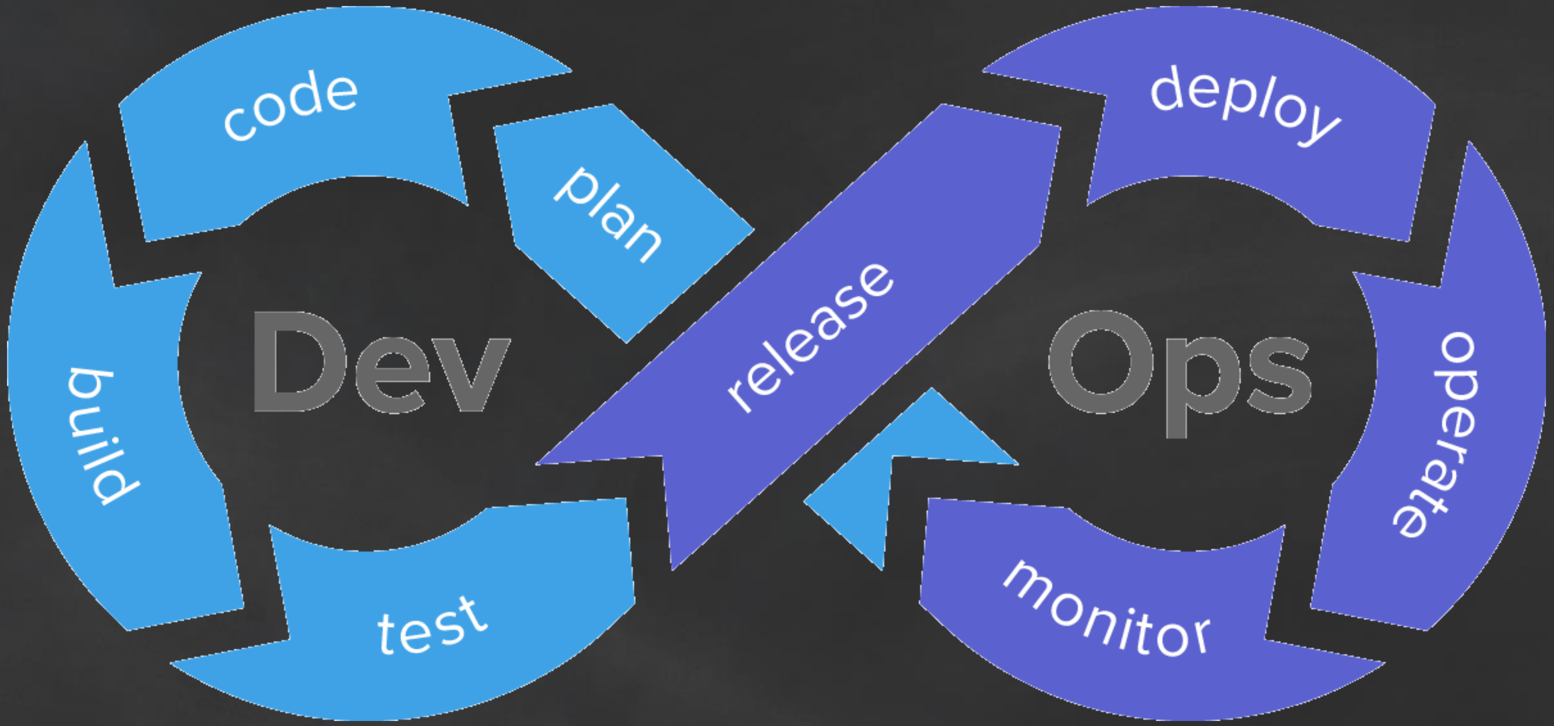


DevOps Project Workshop

CI/CD using Terraform, GitHub, Jenkins, Maven, SonarQube,
Jfrog Artifactory, Docker, Kubernetes, Helm Prometheus and Grafana

sonarqube

JFrog Artifactory

GitHub

Jenkins

HELM

kubernetes

HashiCorp Terraform

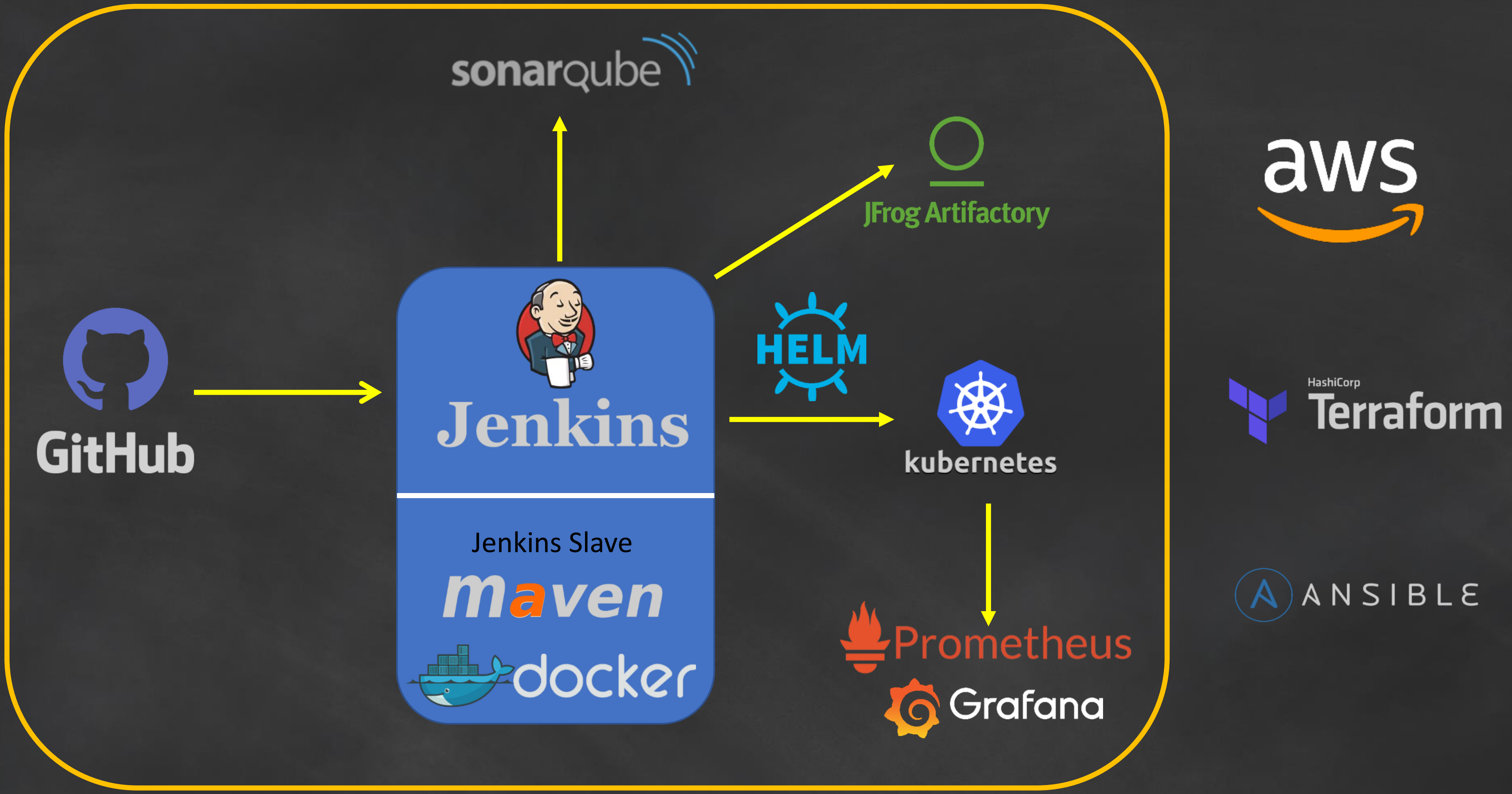
maven

docker

Prometheus

Grafana

ANSIBLE


sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp
Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

Topic to be covered

1. Setup Terraform
2. Provision Jenkins master, build node, and Ansible using Terraform
3. Setup Ansible server
4. Configure Jenkins master and build node using Ansible
5. Create Jenkins pipeline job
6. Create Jenkins file from scratch
7. Create multibranch pipeline
8. Enable webhook on GitHub
9. Configure SonarQube and Sonar scanner
10. Execute the Sonar analysis
11. Define rules and gates of SonarQube

Topic to be covered

13. Sonar callback rules
14. Jfrog Artifactory setup
15. Create Dockerfile
16. Store Docker images on Artifactory
17. Provision Kubernetes cluster using Terraform
18. Create Kubernetes objects
19. Deploy the Kubernetes objects using Helm
20. Setup Prometheus and Grafana using Helm charts
21. Monitor Kubernetes cluster using Prometheus.

Who can enroll

- Who is trying to switch to DevOps
- Who are planning to setup an end-to-end DevOps pipeline
- Keen to learn DevOps workflow
- Candidates who are attending interviews on DevOps

Before starting

Visual Studio Code

HashiCorp Terraform

AWS CLI

git

MobaXterm

Install Visual Studio Code

1. Download Visual Studio Code
2. Double click to install

Visual Studio Code

Install Git

1. Download Gitbash
2. Double click to install
3. Open the gitbash terminal

Install Terraform

1. Download Terraform
2. Create a folder / directory and store downloaded terraform file
3. Set Environment variable
    a. Edit the system environment variables → Environment variables
    b. System variables → Path

HashiCorp Terraform

Install and Setup AWSCLI

1. Install AWSCLI
2. Create IAM User with administrator access.
3. Run “aws configure”

Terraform

1. Install terraform
2. Create IAM user on AWS
3. Configure AWS CLI to connect with AWS cloud
4. Provision VPC
5. Provision 3 instances
   a. 1 for Jenkins
   b. 1 for Jenkins slave
   c. 1 for Ansible

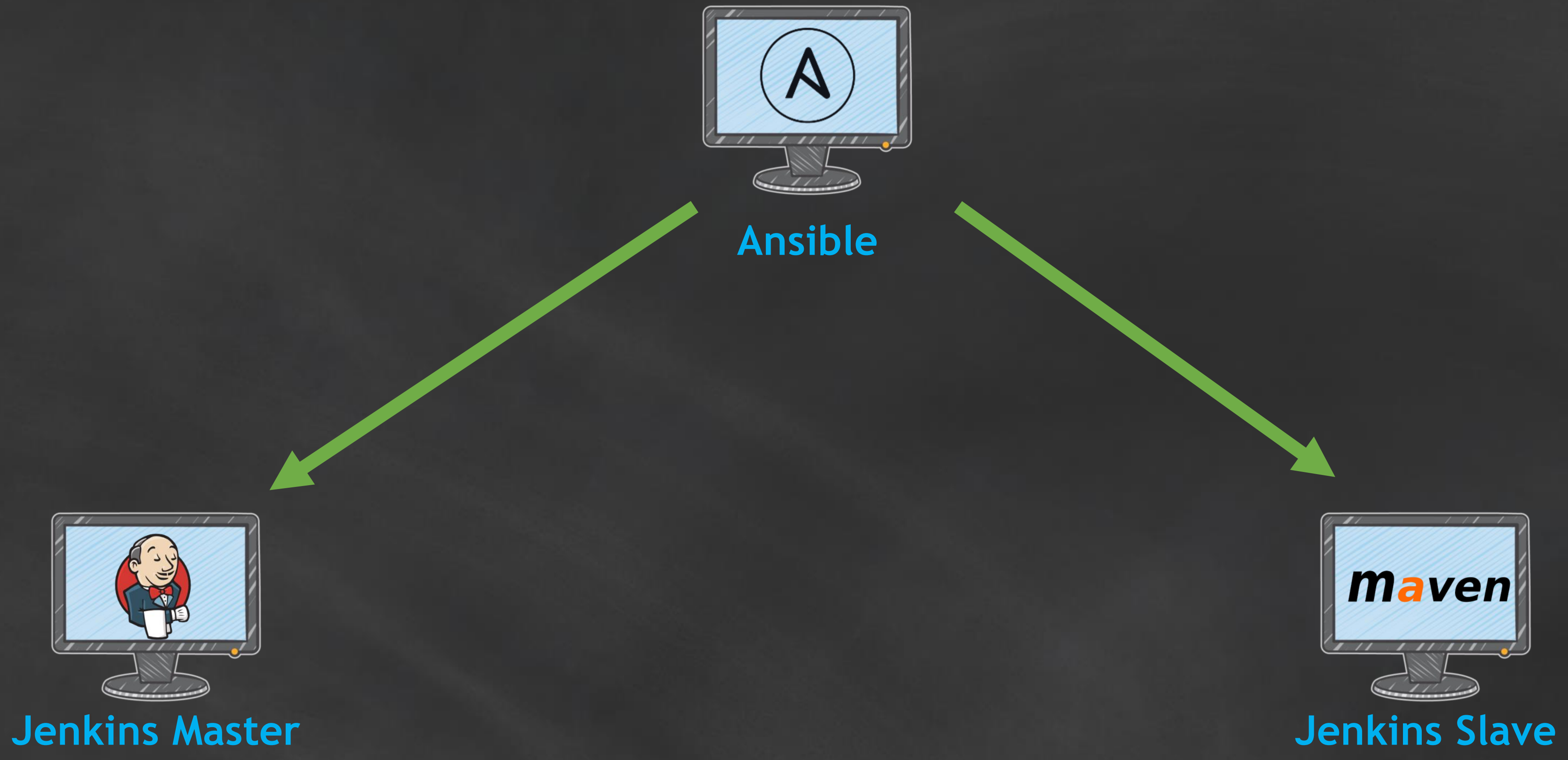
Terraform Commands

`terraform init`	:	Prepare your working directory for other commands
`terraform validate`	:	Check whether the configuration is valid
`terraform plan`	:	Show changes required by the current configuration
`terraform apply`	:	Create or update infrastructure
`terraform delete`	:	Destroy previously-created infrastructure

Topic to be covered

1. Setup Terraform
2. Provision Jenkins master, build node, and Ansible using Terraform
3. Setup Ansible server
4. Configure Jenkins master and build node using Ansible
5. Create Jenkins pipeline job
6. Create Jenkinsfile from scratch
7. Create multibranch pipeline
8. Enable webhook on GitHub
9. Configure SonarQube and Sonar scanner
10. Execute the Sonar analysis
11. Define rules and gates of SonarQube

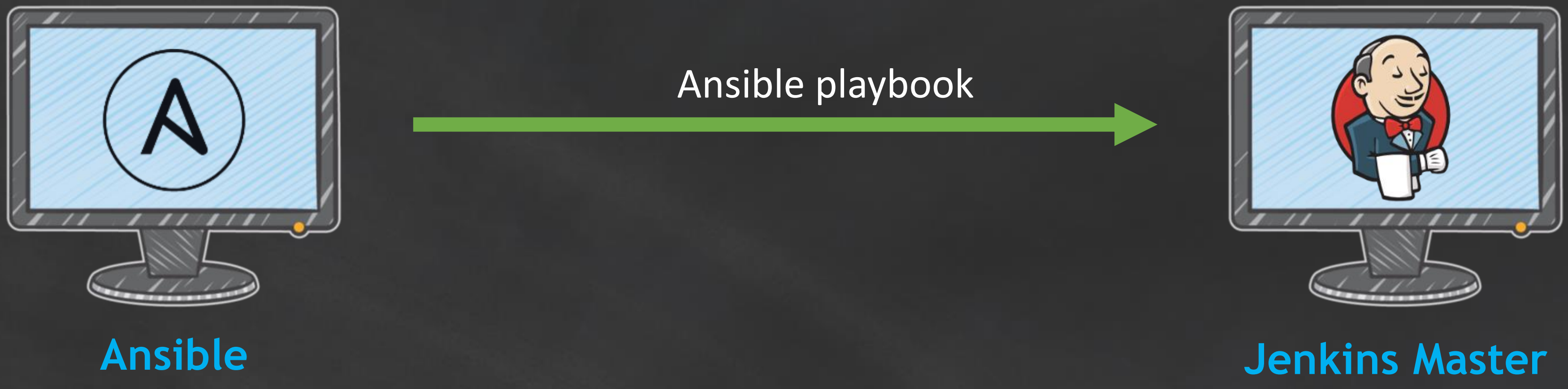
Ansible Setup



Ansible Setup

1. Install Ansible
2. Add inventory
3. Copy private key on to ansible
4. Test connection

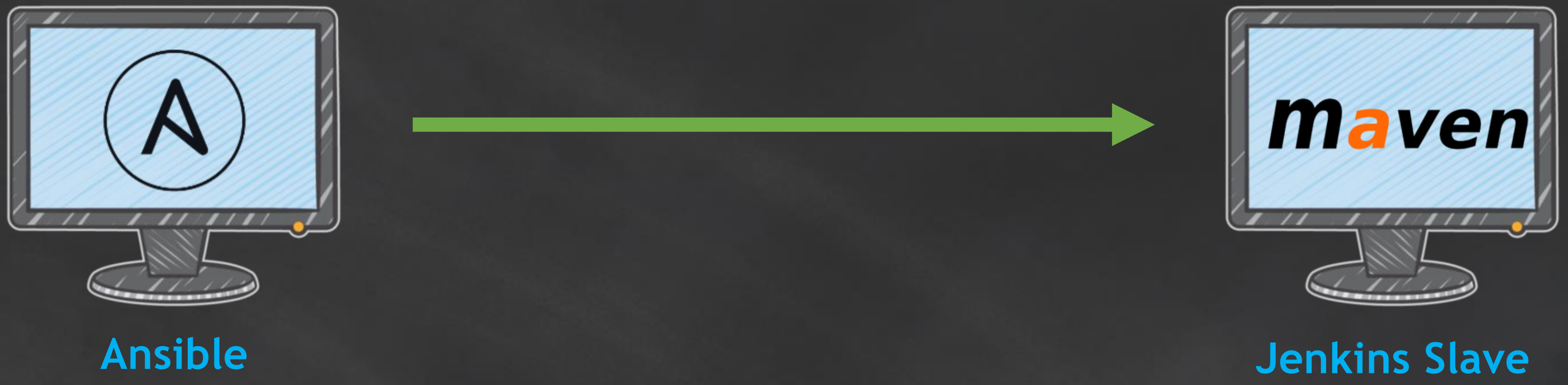
Setup Jenkins using Ansible



Jenkins Installation

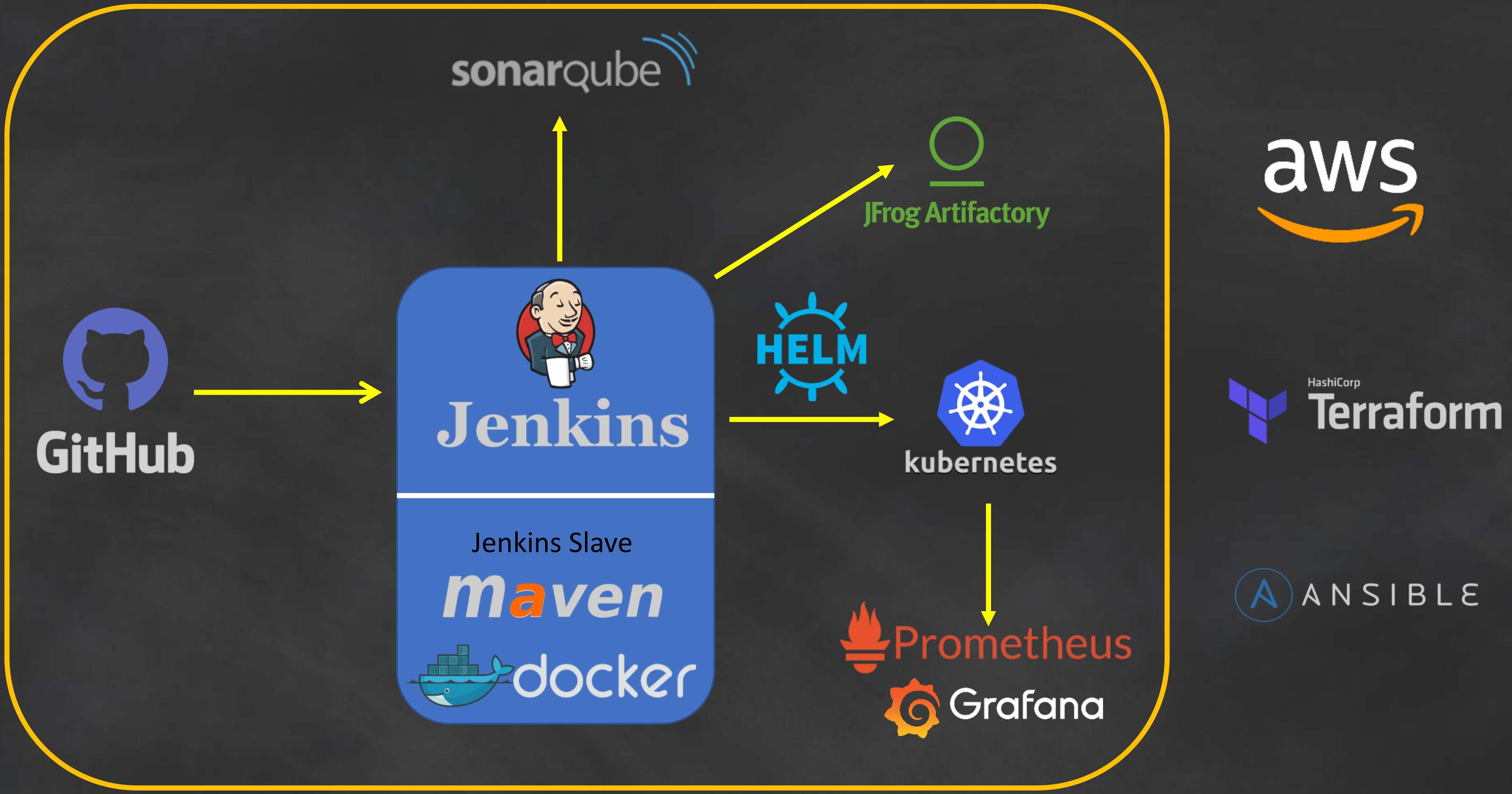
1. Add the Jenkins repo keys to system
2. Add repository to system
3. Install dependencies
4. Install Jenkins

Setup build node using Ansible

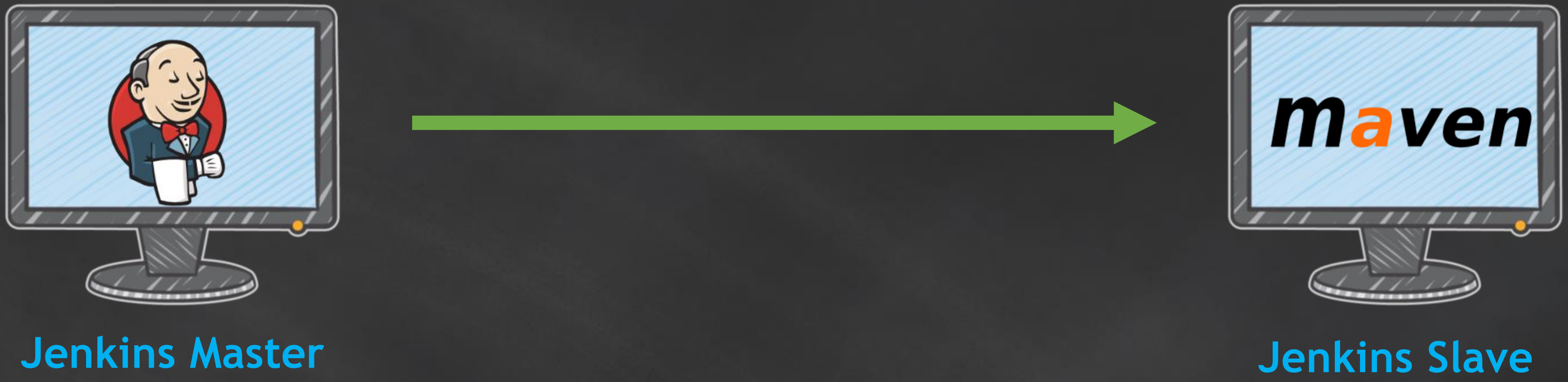


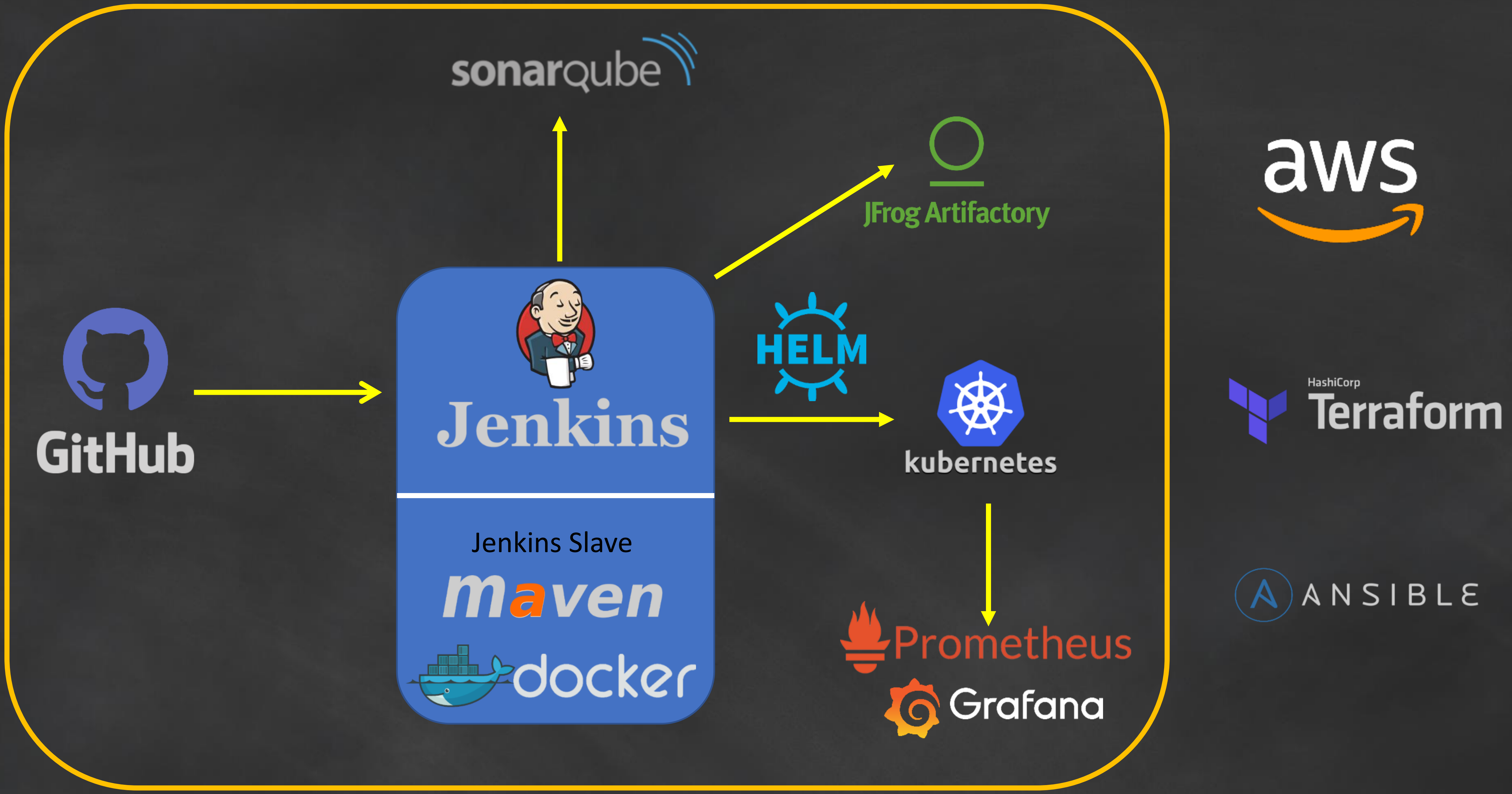
Maven Setup

1. Update the system
2. Install java
3. Download Maven packages
4. Extract it
5. Add path to bash_profile (Optional)


sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp
Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

Jenkins master and slave setup




sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

Topic to be covered

1. Setup Terraform
2. Provision Jenkins master, build node, and Ansible using Terraform
3. Setup Ansible server
4. Configure Jenkins master and build node using Ansible
5. Create Jenkins pipeline job
6. Create Jenkinsfile from scratch
7. Create multibranch pipeline
8. Enable webhook on GitHub
9. Configure SonarQube and Sonar scanner
10. Execute the Sonar analysis
11. Define rules and gates of SonarQube

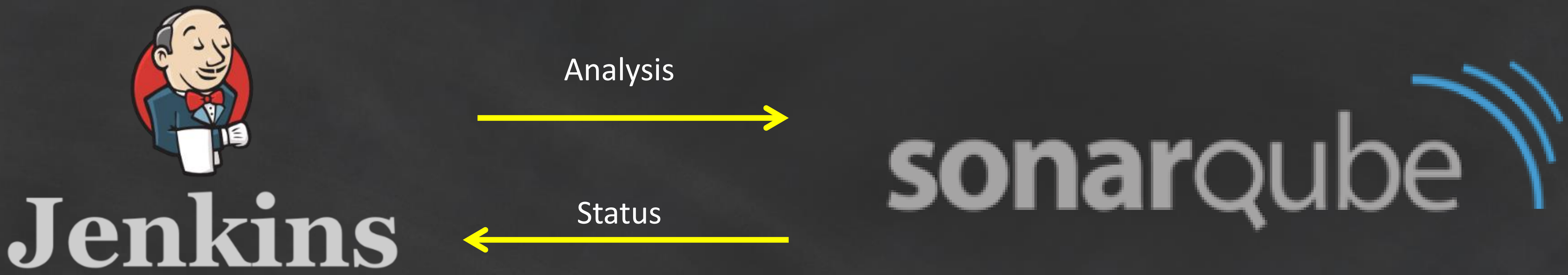
SoarQube Integration

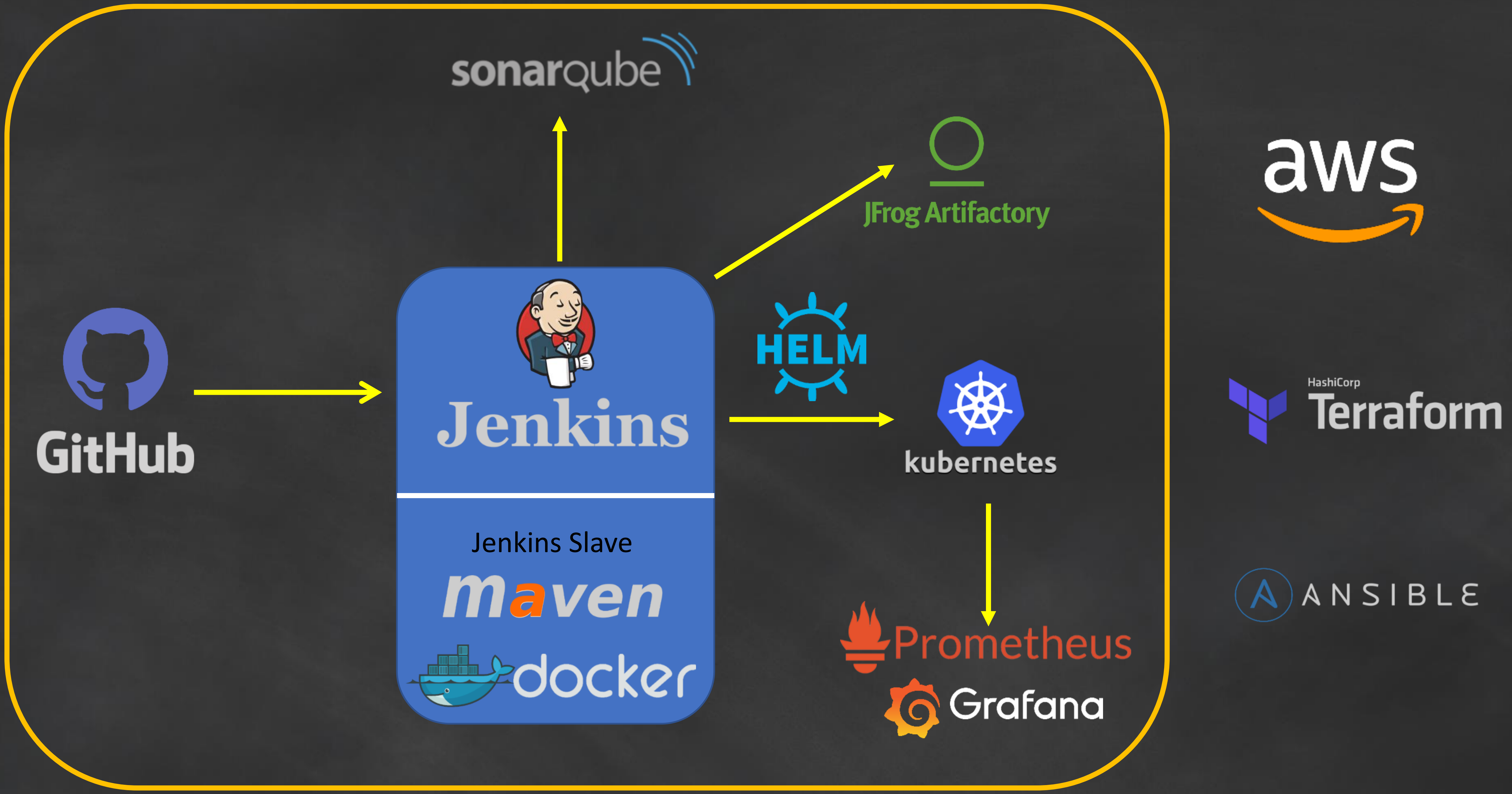
sonarqube
Jenkins

SoarQube Integration

- Create an account at https://sonarcloud.io
- Generate an authentication token on SonarQube
- Create credentials for token in the Jenkins
- Download “SonarQube scanner” plugin on Jenkins
- Configure SonarQube server
- Add SonarQube scanner to Jenkins
- Create SonarQube Properties file
- Add SonarQube stage in the Jenkinsfile

SoarQube Integration

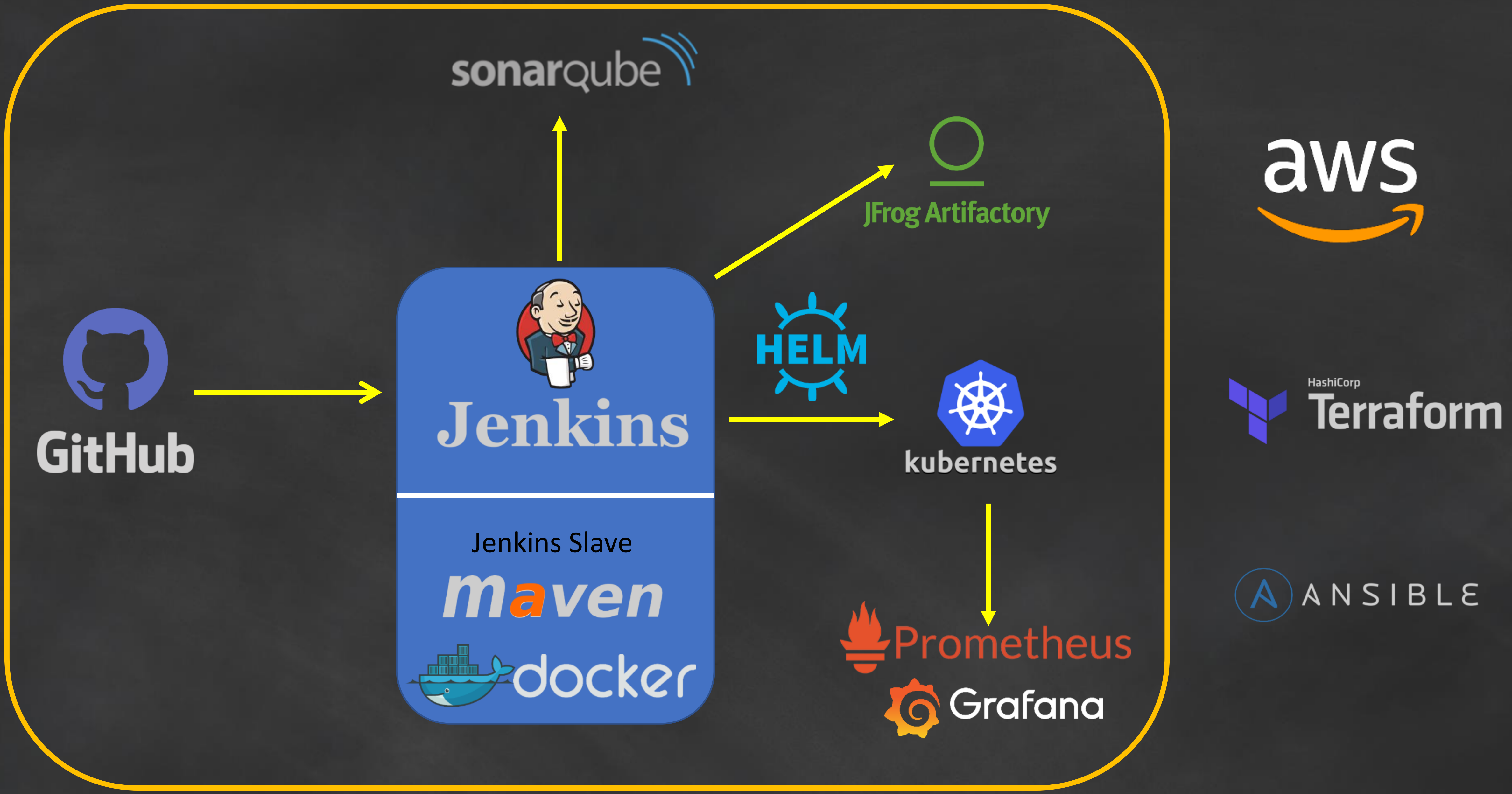



sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

Artifactory Integration

- Create an Artifactory Account
- Generate access token with username
- Add Username and Password under Jenkins Credentials
- Install Artifactory plugin
- Update Jenkinsfile with jar publish stage
- Create a Dockerfile
- Create and publish docker image on Artifactory

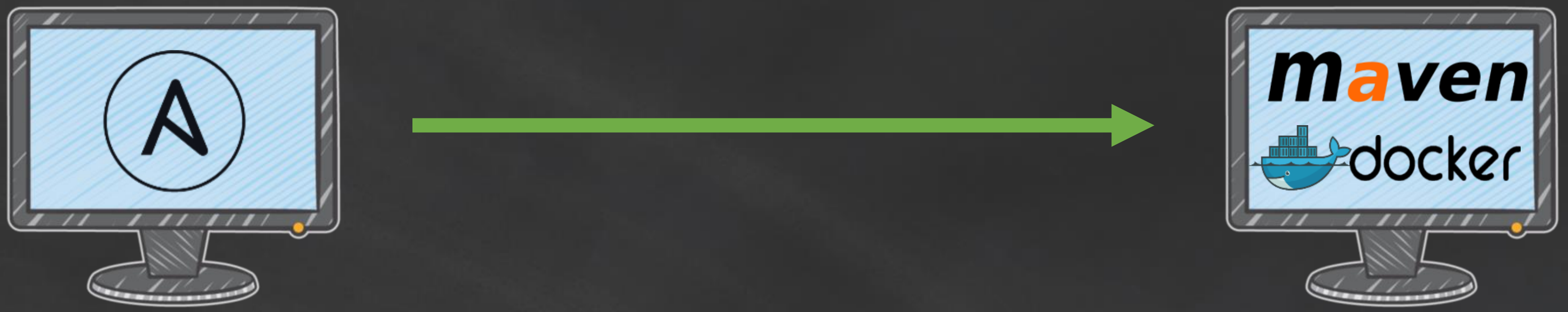
JFrog Artifactory


sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

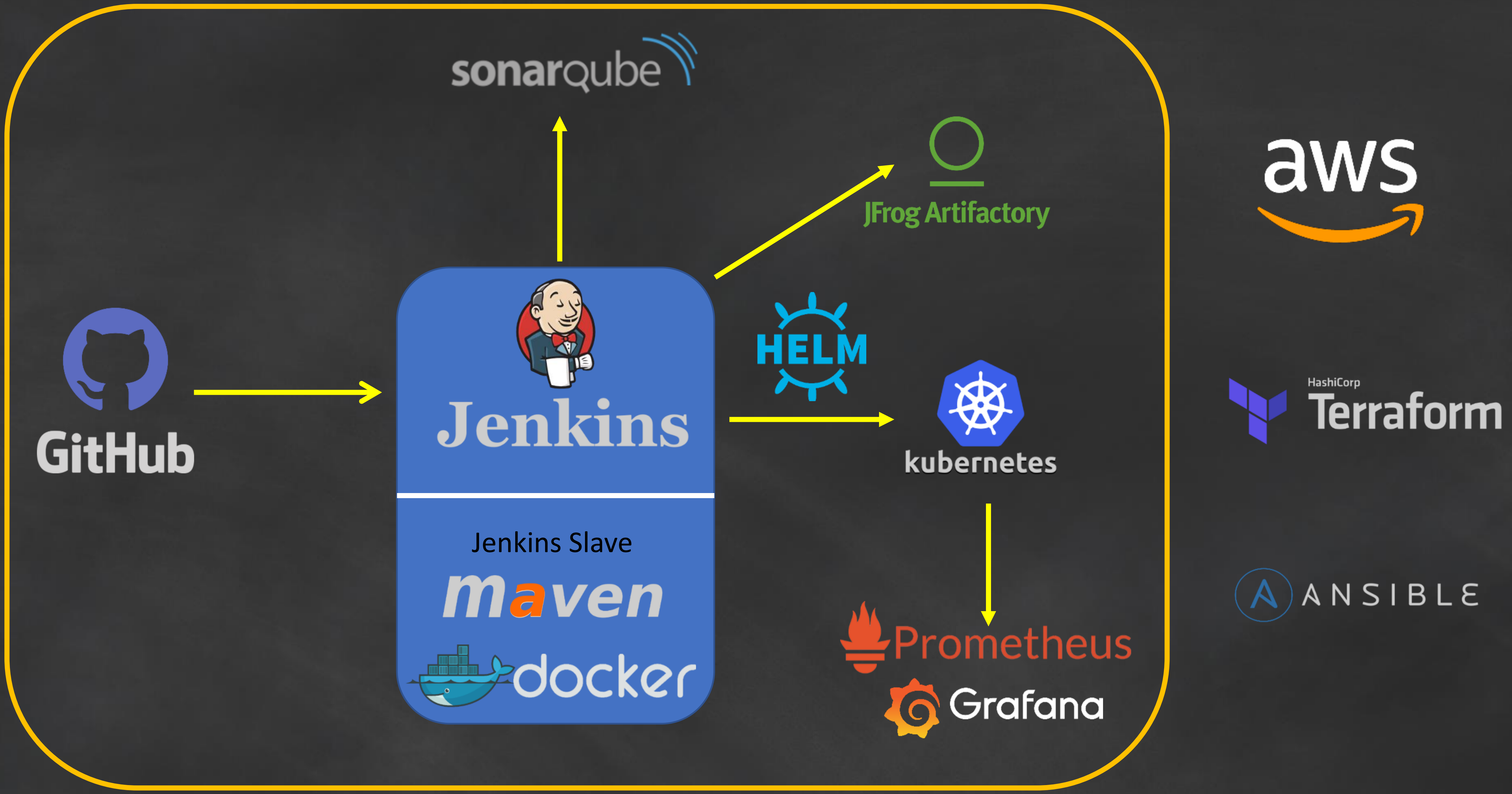
Integrate docker with Jenkins

- Install docker on Jenkins slave system
- Create a Dockerfile
- Create a docker repository in jfrog
- Install “docker pipeline” plugin
- Update Jenkins file with docker build and publish stage

Docker setup using Ansible



Jenkins Slave


sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

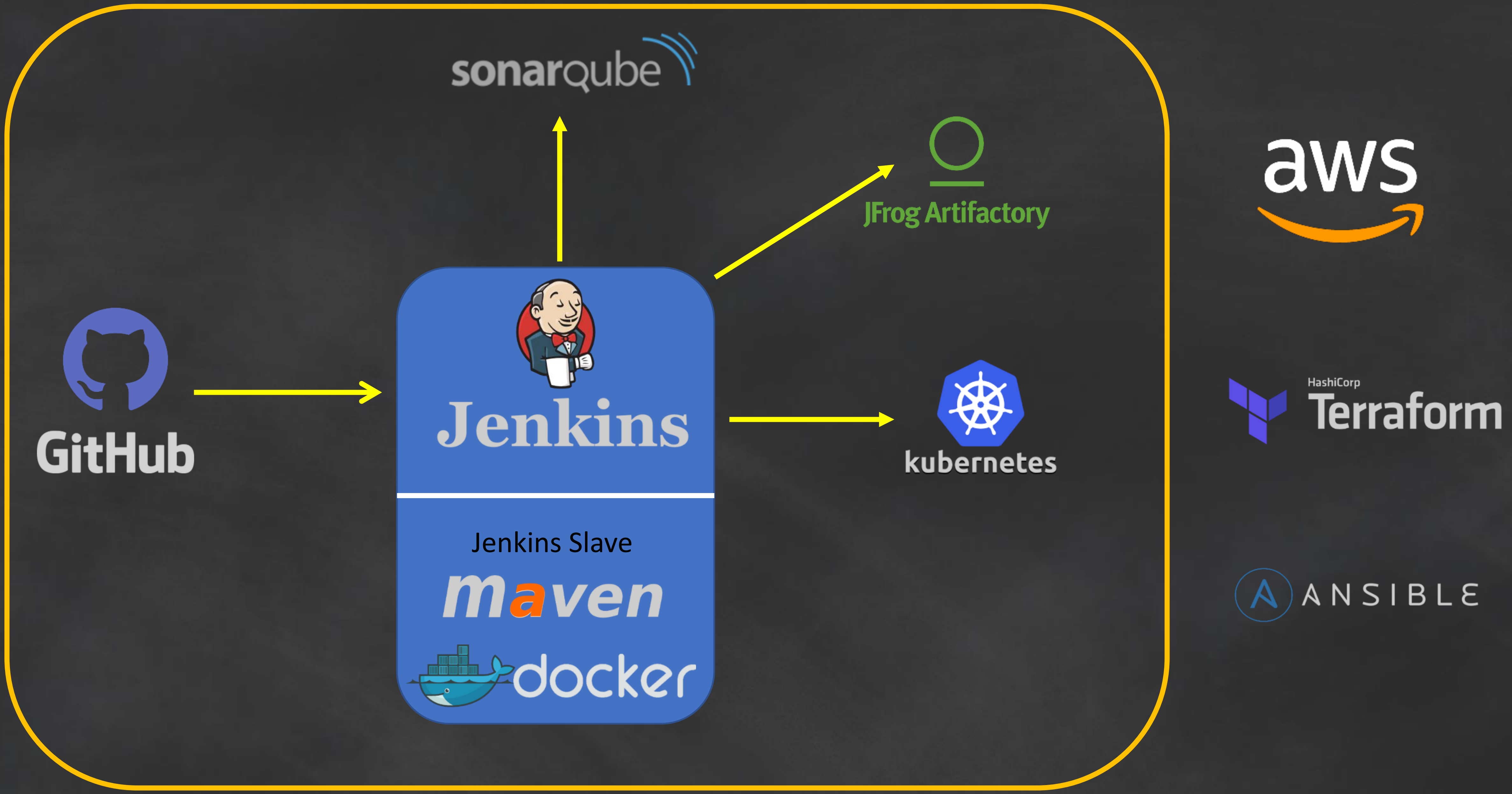
Kubernetes Setup

- Setup Kubernetes cluster (EKS) using terraform
- Create deployment and service files
- Create secrets
- Using secrets in the manifest files

Setup EKS

- Write Terraform manifest file to create EKS
- Write Terraform manifest file to create security group for EKS

Amazon EKS

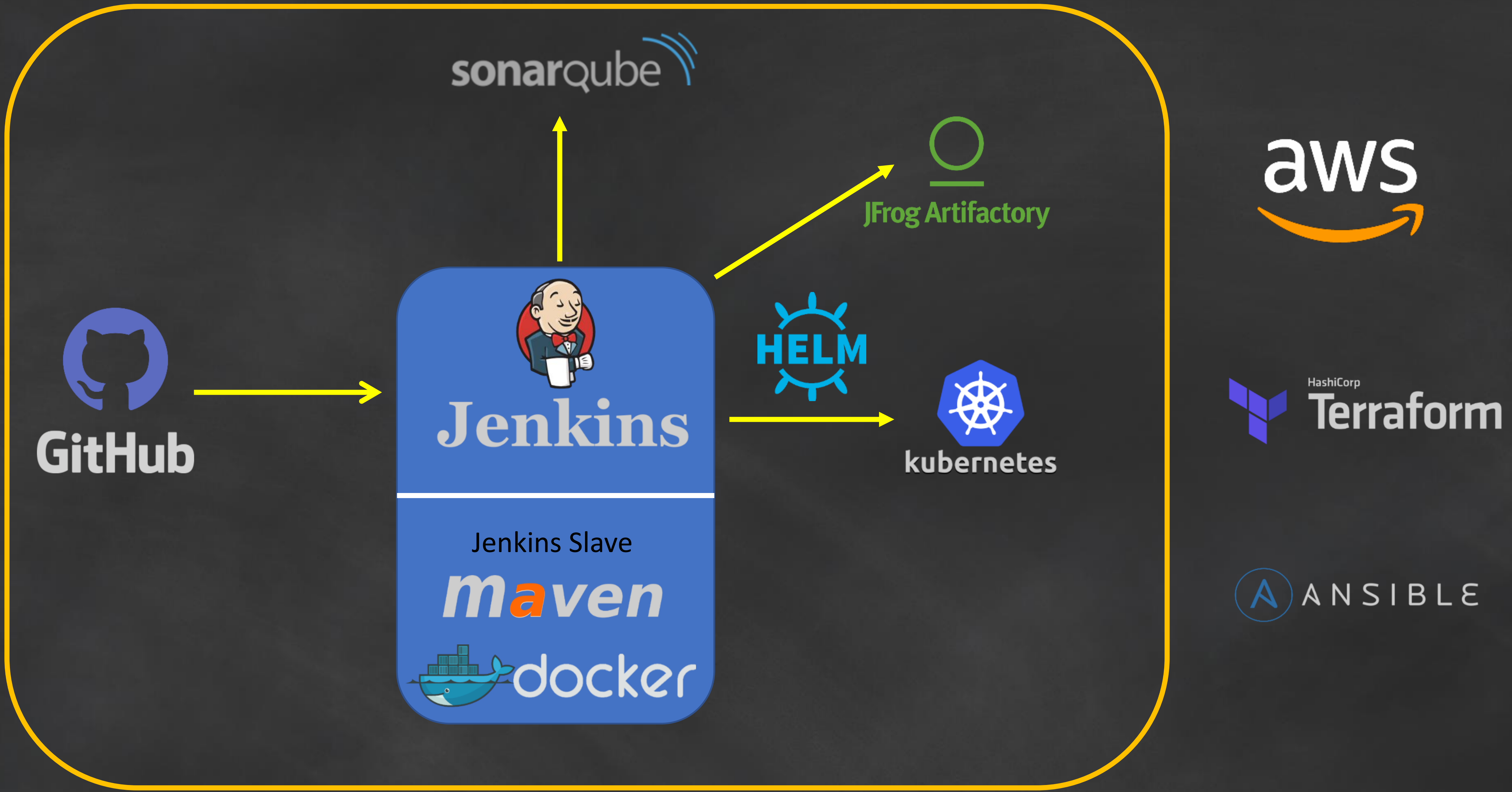

sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
kubernetes
HashiCorp Terraform
Jenkins Slave
maven
docker
ANSIBLE

Topic to be covered

1. Setup Terraform
2. Provision Jenkins master, build node, and Ansible using Terraform
3. Setup Ansible server
4. Configure Jenkins master and build node using Ansible
5. Create Jenkins pipeline job
6. Create Jenkinsfile from scratch
7. Create multibranch pipeline
8. Enable webhook on GitHub
9. Configure SonarQube and Sonar scanner
10. Execute the Sonar analysis
11. Define rules and gates of SonarQube

Topic to be covered

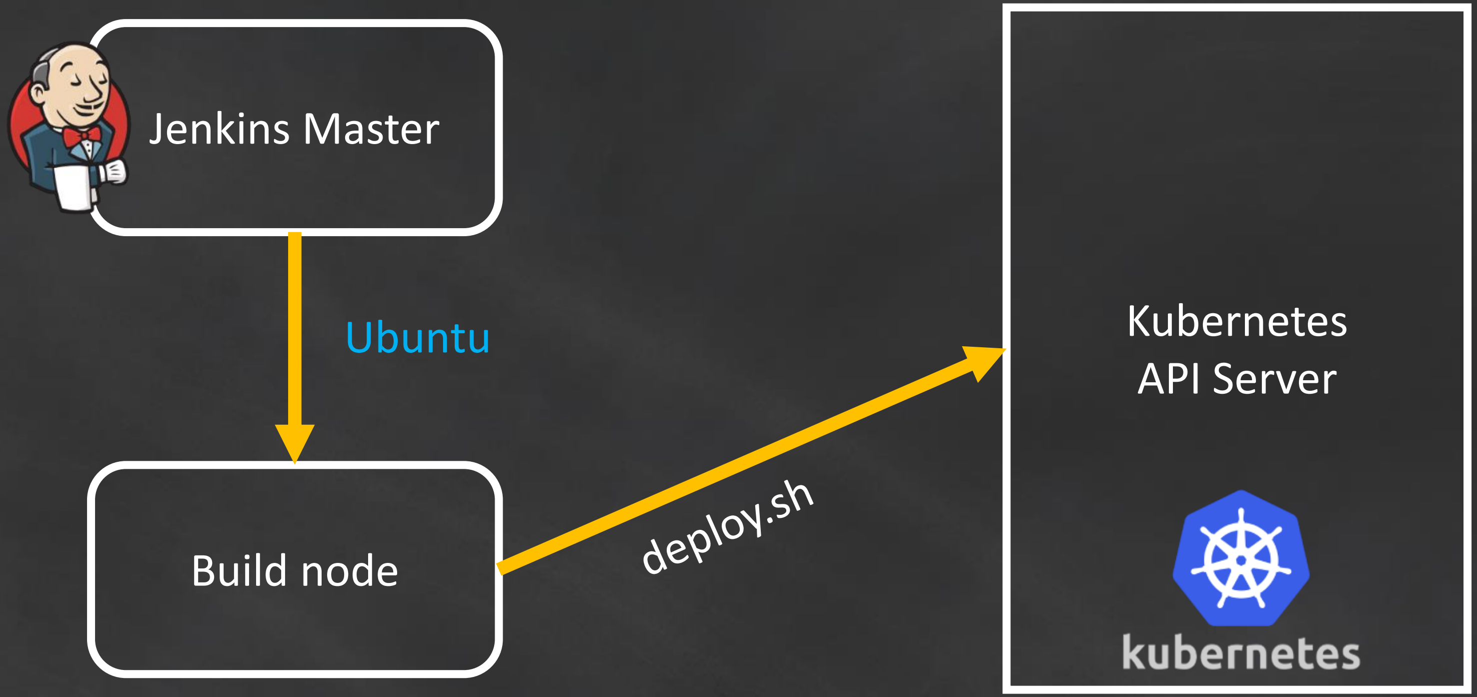
13. Sonar callback rules
14. Jfrog Artifactory setup
15. Create Dockerfile
16. Store Docker images on Artifactory
17. Provision Kubernetes cluster using Terraform
18. Create Kubernetes objects
19. Deploy the Kubernetes objects using Helm
20. Setup Prometheus and Grafana using Helm charts
21. Monitor Kubernetes cluster using Prometheus.


sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp Terraform
Jenkins Slave
maven
docker
ANSIBLE

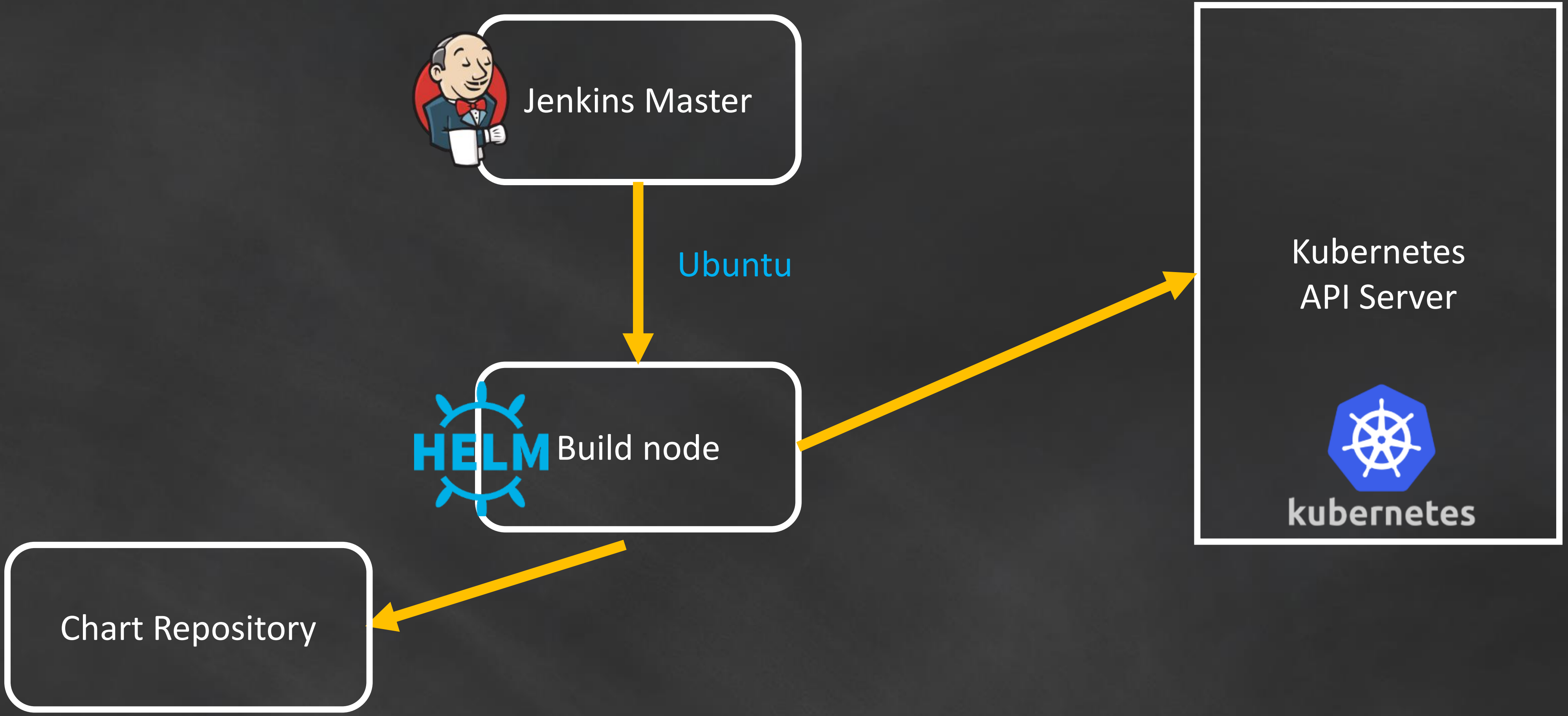
Helm Charts

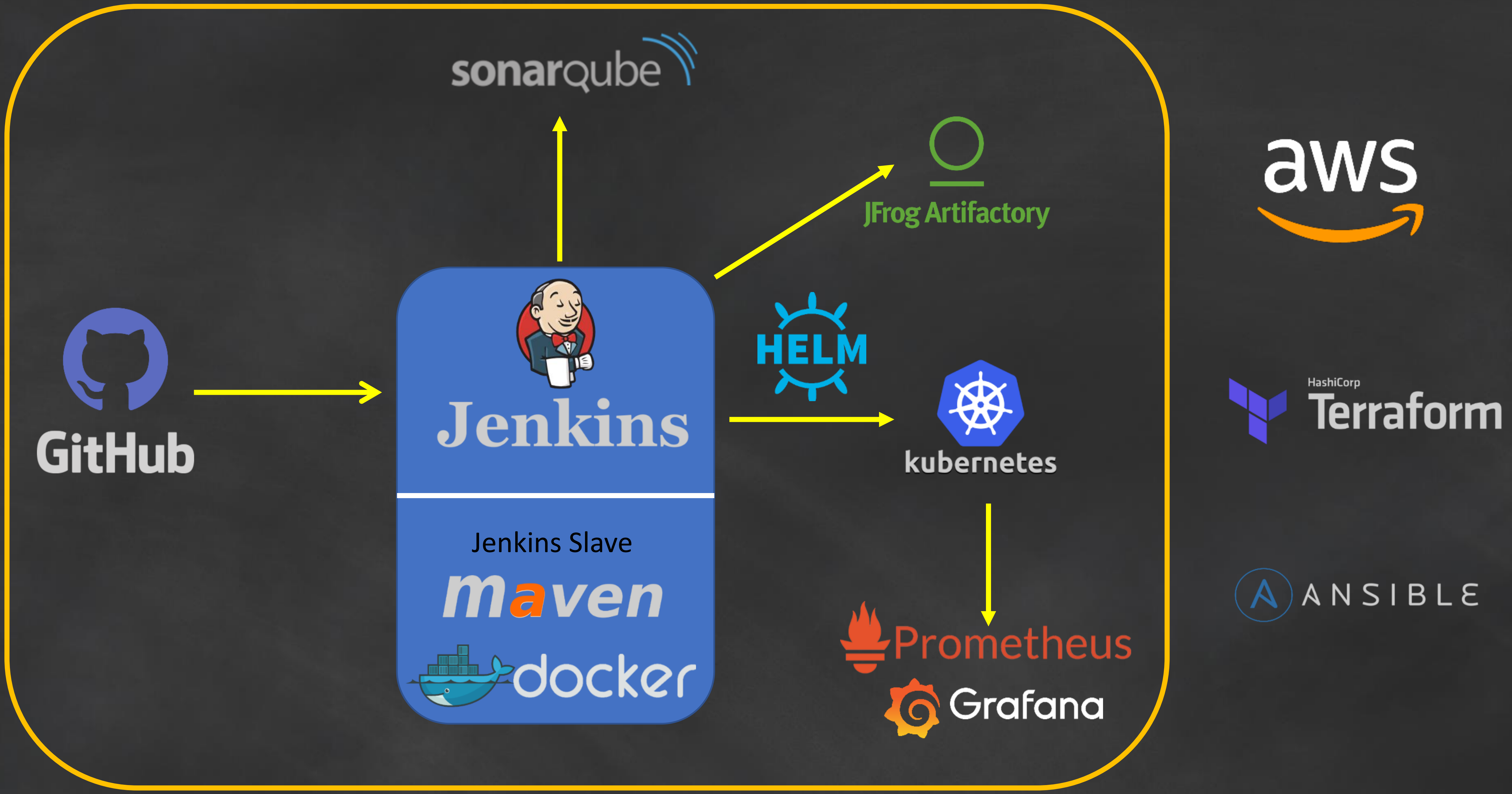
- ✓ **Helm** is a package manager for Kubernetes
- ✓ A **chart** is a package of pre-configured Kubernetes resources
- ✓ A **repository** is a group of published charts which can be made available to others
- ✓ Helm used for the repetitive tasks and application
- ✓ Helm should be installed on Jenkins salve (build server)

Deployment


Jenkins Master
Ubuntu
Build node
deploy.sh
Kubernetes
API Server
kubernetes

Helm deployment




sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

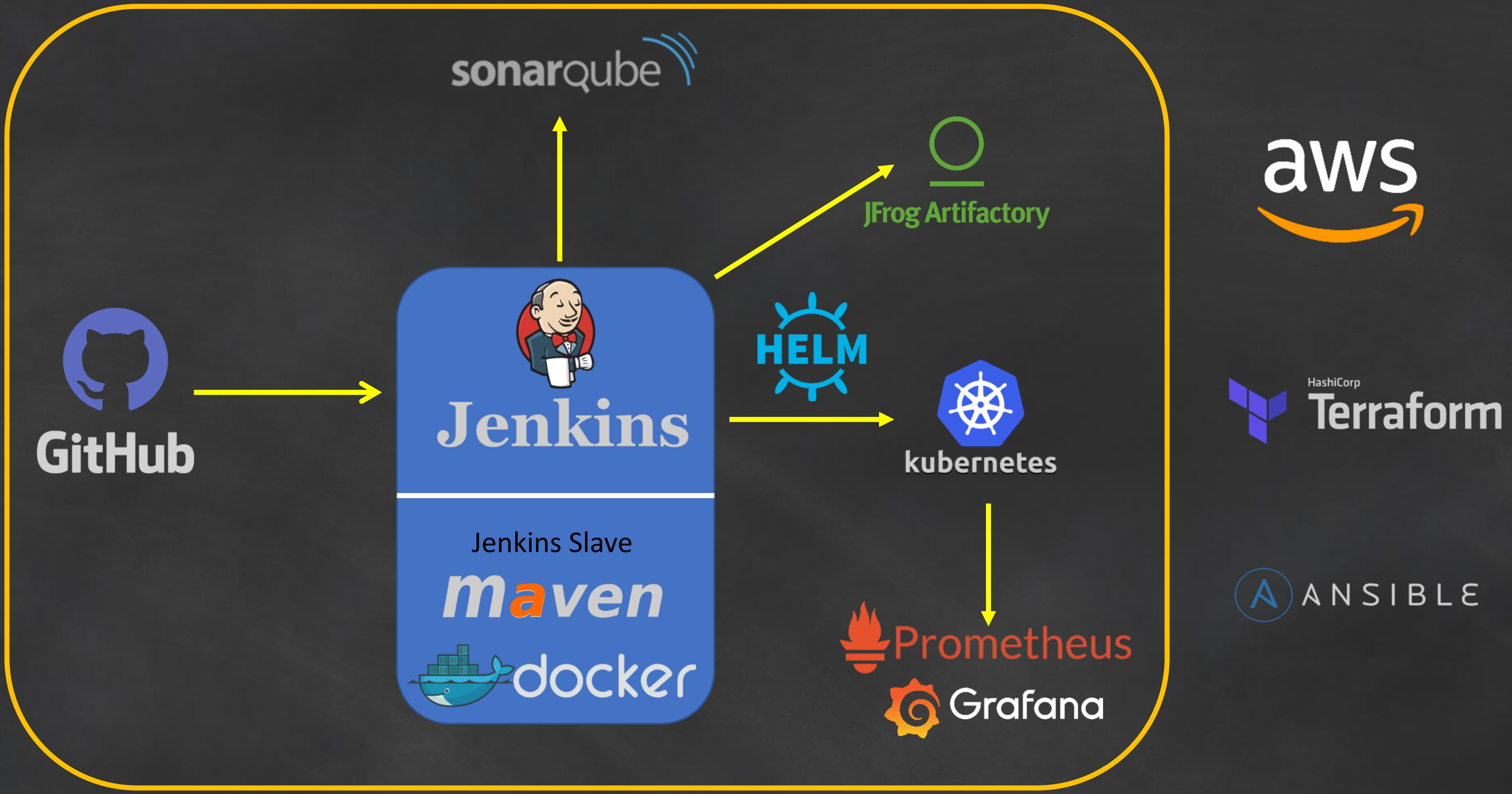
Prometheus

- ✓ Prometheus is an Open-source systems monitoring and alerting toolkit.
- ✓ Prometheus collects and stores the metrics as time series data.
- ✓ Prometheus Scrapes targets
- ✓ PromQL is language to query time series in Prometheus
- ✓ Service Discovery helps find our services and monitor them
- ✓ Exporters helps to monitor 3rd party components
- ✓ Prometheus can send alerts to the Alert manager
- ✓ Prometheus runs on port 9090 and Alert Manager runs on 9093

Grafana

- ✓ Grafana is a multi-platform open-source analytics and interactive visualization web application.
- ✓ It provides charts, graphs and alerts for the web when connected to supported data services.
- ✓ Grafana allows us to query, visualize, alert and understand our metrics, no matter where they are stored. Some supported data sources in addition to Prometheus are AWS CloudWatch, AzureMonitor, PostgreSQL, Elasticsearch and many more.
- ✓ We can create our own dashboards or use the existing ones provided by Grafana. We can personalize the dashboards as per our requirements.

Congratulations!


sonarqube
JFrog Artifactory
aws
GitHub
Jenkins
HELM
kubernetes
HashiCorp
Terraform
Jenkins Slave
maven
docker
ANSIBLE
Prometheus
Grafana

Congratulation

I also want to express my gratitude for giving me the opportunity to teach throughout this course. I have put in my best efforts to deliver high-quality content that is both informative and engaging. I hope you have thoroughly enjoyed this learning journey.

If you found the course valuable and enjoyable, I kindly request you to share your review and give it a 5-star rating. Your feedback will not only motivate me but also help others who are considering taking this course.

However, if you feel that certain aspects of the course could be improved, I encourage you to provide your feedback through direct messaging. Your input will be immensely valuable in shaping future iterations of this course and making it even better.

Once again, I want to express my heartfelt gratitude for giving me the opportunity to teach you. I wish you all the best in your future endeavors, and may your learning continue to bring you joy and success.

Happy learning!

AR Shankar