JENKINS 2 PIPELINE AS A CODE

DENVER JENKINS AREA MEETUP

MAYANK PATEL

APPLICATION ARCHITECT @ OILDEX

Linkedin / @maxy_ermayank / Medium

OILDEX

SOFTWARE AS A SERVICE PROVIDER FOR OIL AND GAS COMPANIES

- 7.5 Years

FOCUSED ON

- Streaming, Reactive, Non-blocking Architecture
- API Design
- DevOps
- Cloud Native Architecture
- Empowering software development teams
- Digital Transformation and Digital Optimization

AGENDA

- Overview of Jenkins 2
- Jenkins Pipeline Features
- Jenkins Environment
- Ideal Pipeline Flow
- Important pipeline plugins
- Demo
  - Setup and configure a pipeline
  - Using Docker with pipelines

OVERVIEW OF JENKINS 2

- Built-in support for delivery pipelines.
- Improved usability.
- Fully backwards compatible.

JENKINS PIPELINE FEATURES

Can support complex, real-world, CD Pipeline requirements: pipelines can fork/join, loop, parallel, to name a few programmatically

Is Resilient: pipeline executions can survive master restarts

Is Pausable: pipelines can pause and wait for human input/approval

Is Efficient: pipelines can restart from saved checkpoints

Is Visualized: Pipeline StageView provides status at-a-glance dashboards including trending
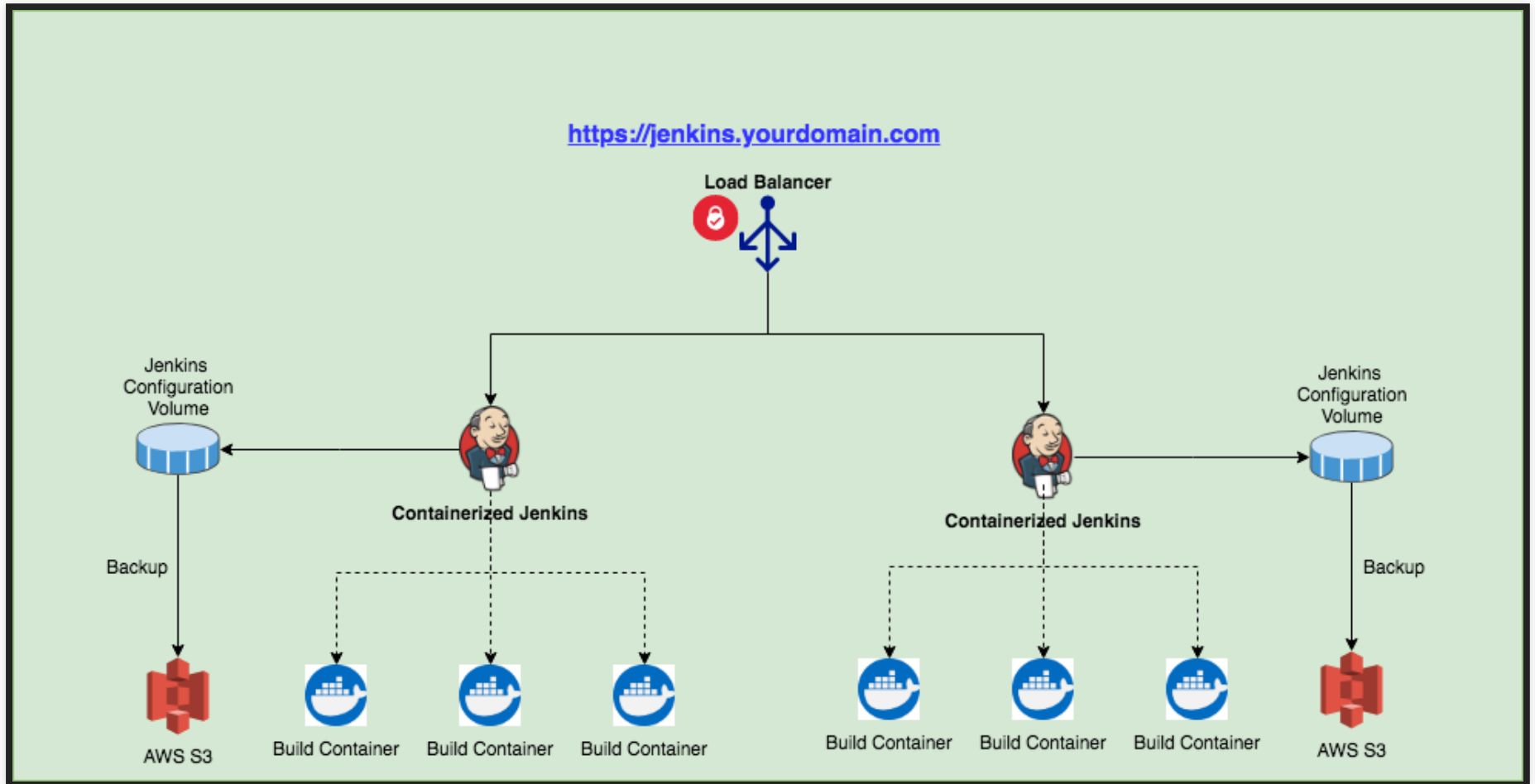
Pipeline configuration as a code in source control

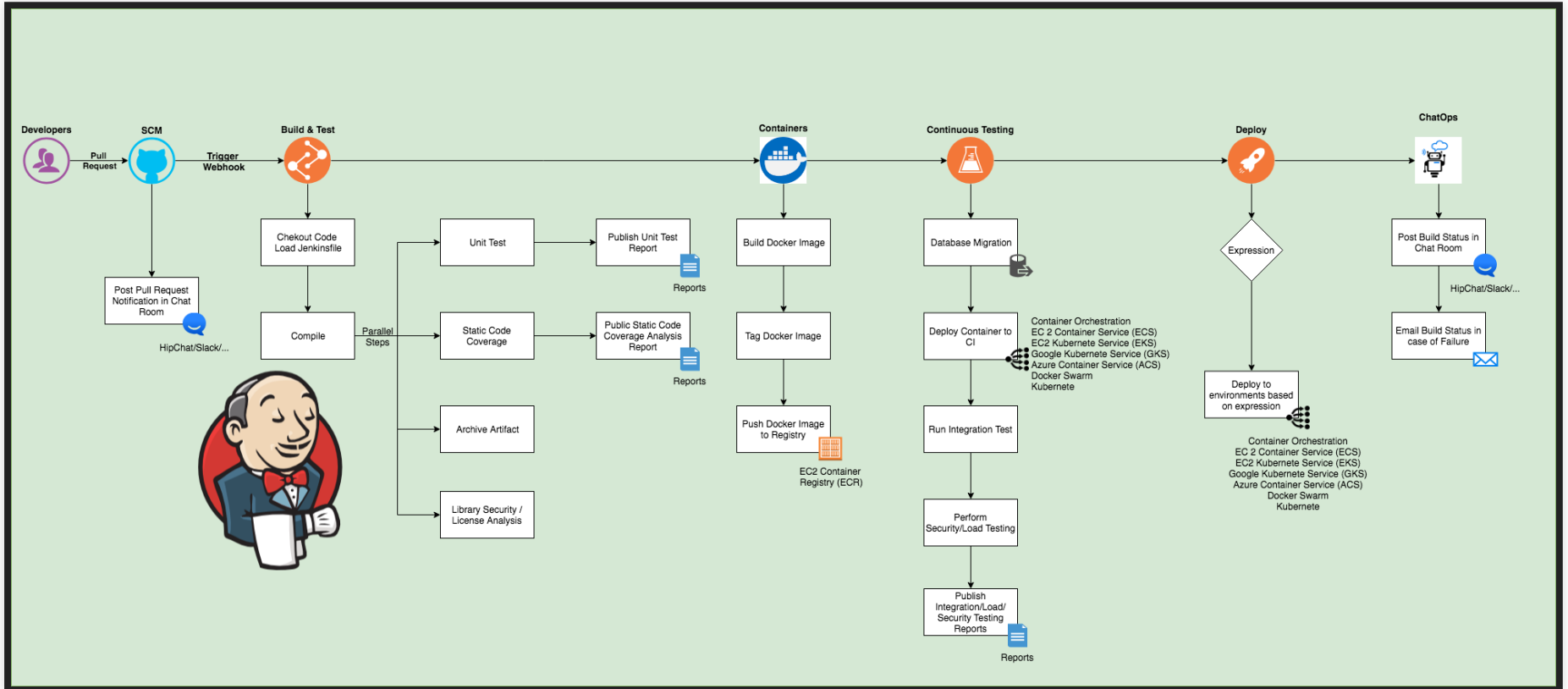Ability to embed Security part of DevOps pipeline

Reusability

Easy Recoverability

JENKINS ENVIRONMENT

PIPELINE FLOW

IMPORTANT PIPELINE PLUGINS

- Pipeline Plugin
- Docker Pipeline
- Docker
- Pipeline Stage View
- GitHub
- Blue Ocean
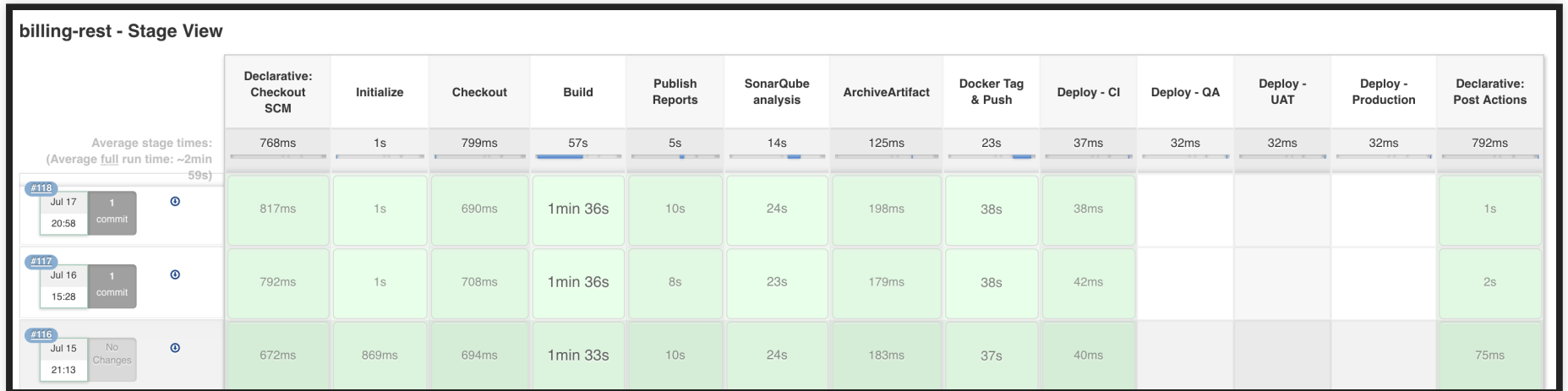- Many More...

GSoC Project Intro: Pipeline as YAML

JENKINS PIPELINE SAMPLE

```groovy
#!groovy

pipeline {

    agent {
            docker {
                image 'jenkinsslave:latest'
                registryUrl 'http://8598567586.dkr.ecr.us-west
                registryCredentialsId 'ecr:us-east-1:534875345
                args '-v /home/centos/.ivy2:/home/jenkins/.ivy
                }
    }
    environment {
            APP_NAME = 'billing-rest'
            BUILD_NUMBER = "${env.BUILD_NUMBER}"
```
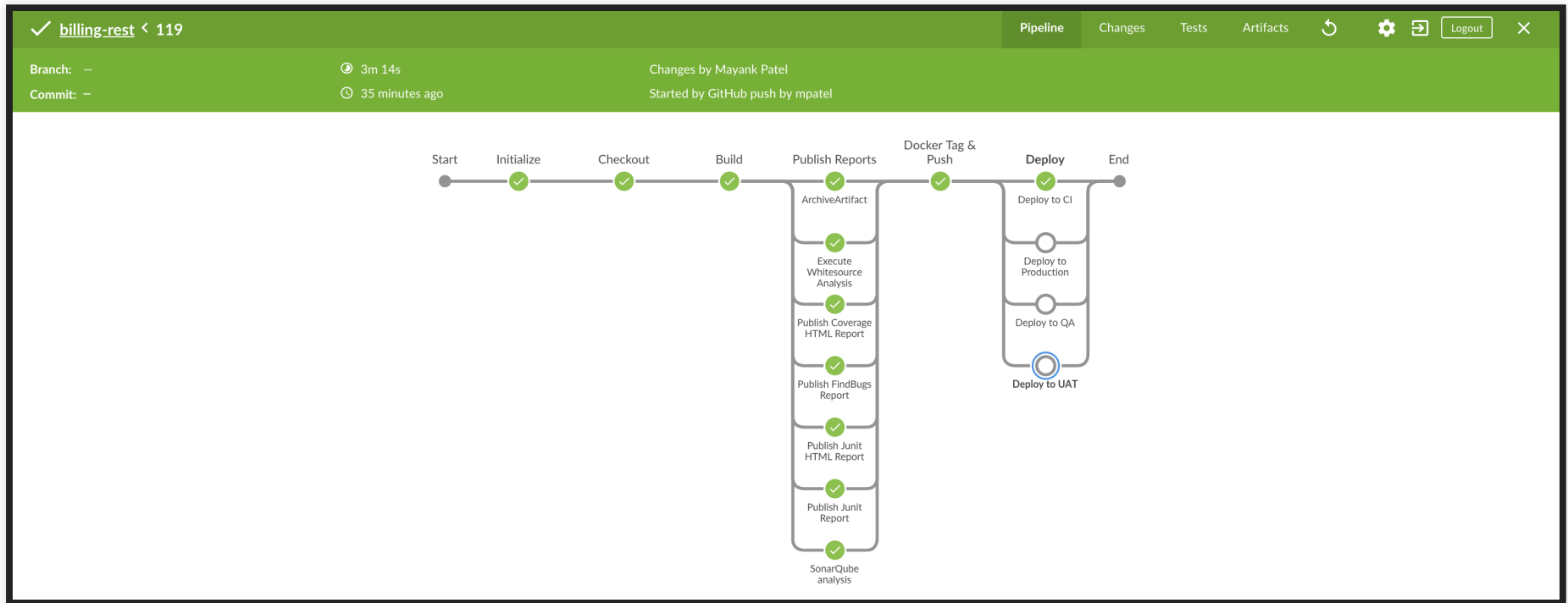
CLASSIC PIPELINE STAGE VIEW

billing-rest - Stage View

	Declarative: Checkout SCM	Initialize	Checkout	Build	Publish Reports	SonarQube analysis	ArchiveArtifact	Docker Tag & Push	Deploy - CI	Deploy - QA	Deploy - UAT	Deploy - Production	Declarative: Post Actions
Average stage times: (Average full run time: ~2min 59s)	768ms	1s	799ms	57s	5s	14s	125ms	23s	37ms	32ms	32ms	32ms	792ms
#118 Jul 17 20:58 — 1 commit	817ms	1s	690ms	1min 36s	10s	24s	198ms	38s	38ms				1s
#117 Jul 16 15:28 — 1 commit	792ms	1s	708ms	1min 36s	8s	23s	179ms	38s	42ms				2s
#116 Jul 15 21:13 — No Changes	672ms	869ms	694ms	1min 33s	10s	24s	183ms	37s	40ms				75ms

BLUE OCEAN PIPELINE STAGE VIEW

RESOURCES

- Best Practices for Scalable Pipeline Code
- Google Trends
- Plugin Compatibility with Pipeline
- Pipeline Example Repository
- jenkins-pipeline-talk
- jenkins-pipeline-demo
- Jenkins Plugins
- Top 10 Best Practices for Jenkins Pipeline Plugin

THANK YOU!

QUESTIONS?