Adding Terraform to a CI/CD Pipeline

Ned Bellavance

FOUNDER, NED IN THE CLOUD LLC

@ned1313 www.nedinthecloud.com

Overview

Level setting with terminology

Automation with Globomantics

Considerations for automation and CI/CD

Source Control Management

Multiple formats

- Git, TFVC, Subversion

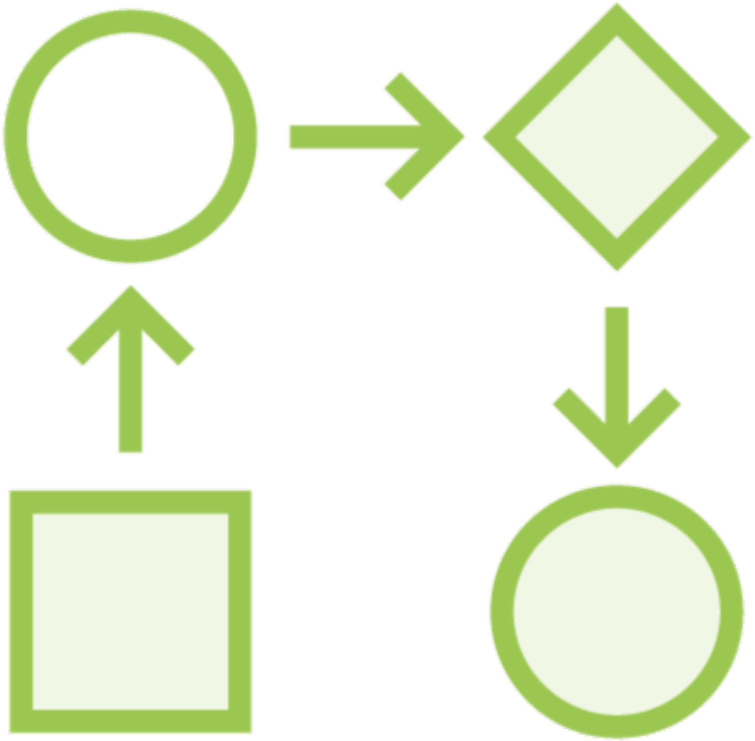
Multiple platforms

- GitHub, BitBucket, GitLab, CodeCommit

Enable collaboration

Version controlled

CI/CD Pipelines



Multiple platforms

- Jenkins, CodePipeline, Bamboo

Continuous Integration for code check-in

Continuous Delivery of builds

Automated testing and validation

Multiple environments

- Development, UAT, QA, Production

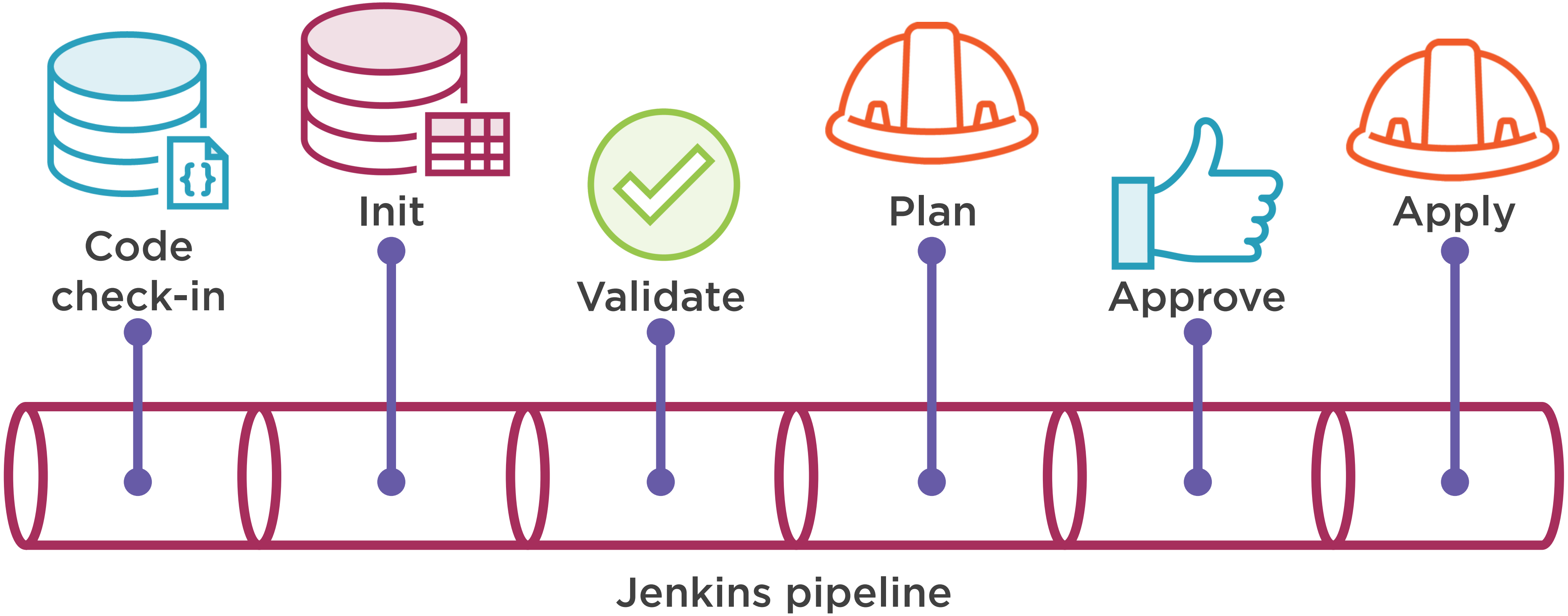
Globomantics Automation Toolset

GitHub for SCM

Jenkins for CI/CD

Consul for Config Data

Automation at Globomantics



Jenkins Setup

Install Jenkins as a traditional app

Deploy Jenkins in a container

Deploy Jenkins in the cloud

Running Automation with Terraform

- Plug-in source
- Workspace usage
- State control
- Output control
- Deploy pattern
- Error Handling

Automation Environment Variables

TF_IN_AUTOMATION = TRUE

TF_LOG = "INFO"

TF_LOG_PATH = "tf_log_MMDDYY_hhmmss"

TF_INPUT = FALSE

TF_VAR_name = "value"

TF_CLI_ARGS = "-input=false"

Summary

Automation is challenging!

Automation is rewarding!

Focus on concepts not tools

Coming up:

- Implementing configuration management
- Using Ansible with Terraform