Automation Of Security Groups Using Terraform

Grant Thornton

MORNINGSTAR®

AGENDA

I. Morningstar's Cloud Environment

II. Transit Gateway & Outposts

III. Why We Need To Automate Security Groups

IV. Technical Automation Solution

About Me

About Me

- Nick Bausch
- Sr Systems Engineer for Cloud Services Team
- Joined Morningstar in 2015

About Me - Career

I. Website Hosting
II. Financial Services
III. Algorithmic Trading Companies
IV. Department of Energy Lab
V. Large Online Retailer
VI. Morningstar Inc

About Me - Focus

I. Unix/Linux Admin

II. Configuration Manage With Chef & Puppet

III. Process Automation

Morningstar’s Cloud Environment

Morningstar's Cloud Environment

- In The Process From On Premises To AWS
- Started Journey in 2017
- 3 of 8 Datacenters Migrated

Morningstar's Cloud Environment

- Very Aggressive Migration Timeline
- We Are Using AWS Outposts
- We Will Be Implementing Transit Gateway (TGW)

Morningstar's Cloud Environment

1,600
Developers
&
Technologists

160
Development
Teams

300+
Unique
Applications

Morningstar's Cloud Environment

70 Teams In AWS	**140** AWS Accounts
2 Regions Per Account	**280** VPCs
3 AZs Per VPC	**840** Subnets

What Is Morningstar's Account Strategy?

- Team Accounts – Team Of Developers Working In AWS Account
  - Prod – Higher Level Environments (Prod/UAT)
  - Non-Prod – Lower Level Environments (DEV/QA/STG)
- Individual Accounts – Sandbox Accounts For Learning & Testing

Team Account Blast Radius

- Accounts Can't Talk To Other Accounts
- Prod Is Only Allowed To Talk To Prod Accounts
- Non-Prod Is Only Allowed To Talk To Non-Prod Accounts

Morningstar's Cloud Environment

- Standardized Security Groups In All Accounts
- Teams Can Create Security Groups Without Oversight
- Standardized Public/Private default NACLs In All Accounts

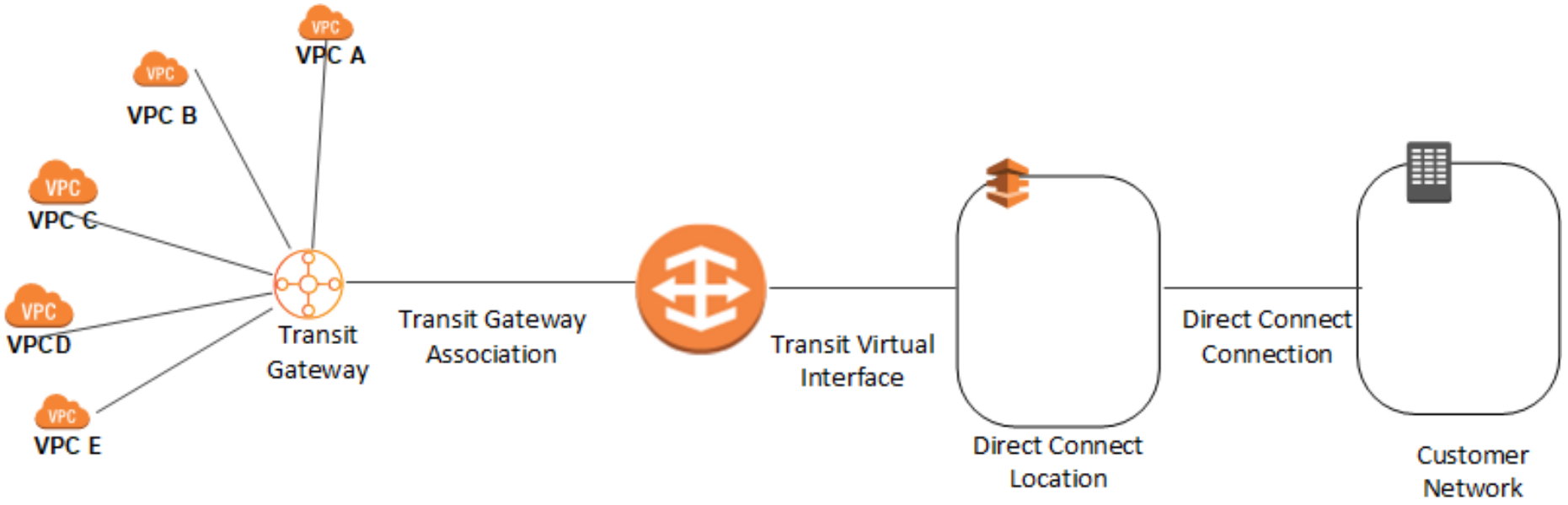
We Use Terraform To Manage Everything

- Stateful Configuration
- Scales Well To Manage Our Team Accounts
- Infrastructure As Code

What Is Transit Gateway?

What Is Transit Gateway?

- Hub To Connect Team Accounts In Same Region
- Acts As Router To Pass Traffic Between Accounts



Why Is Transit Gateway Great?

- 22 Direct Connect Lines
- From ~~1100~~ -> To 22 Virtual Interfaces (VIFs) To Manage

What Security Concerns Does TGW Present?

- Any VPC Connected To TGW Can Communicate
- No Longer Have Blast Radius At Account Level

What Security Concerns Does TGW Present?

- Security Groups Are Maintained By Development Teams
- TGW Acts As A Big Router With No Security Considerations
- Security Groups Are More Critical Now

What Security Concerns Does TGW Present?

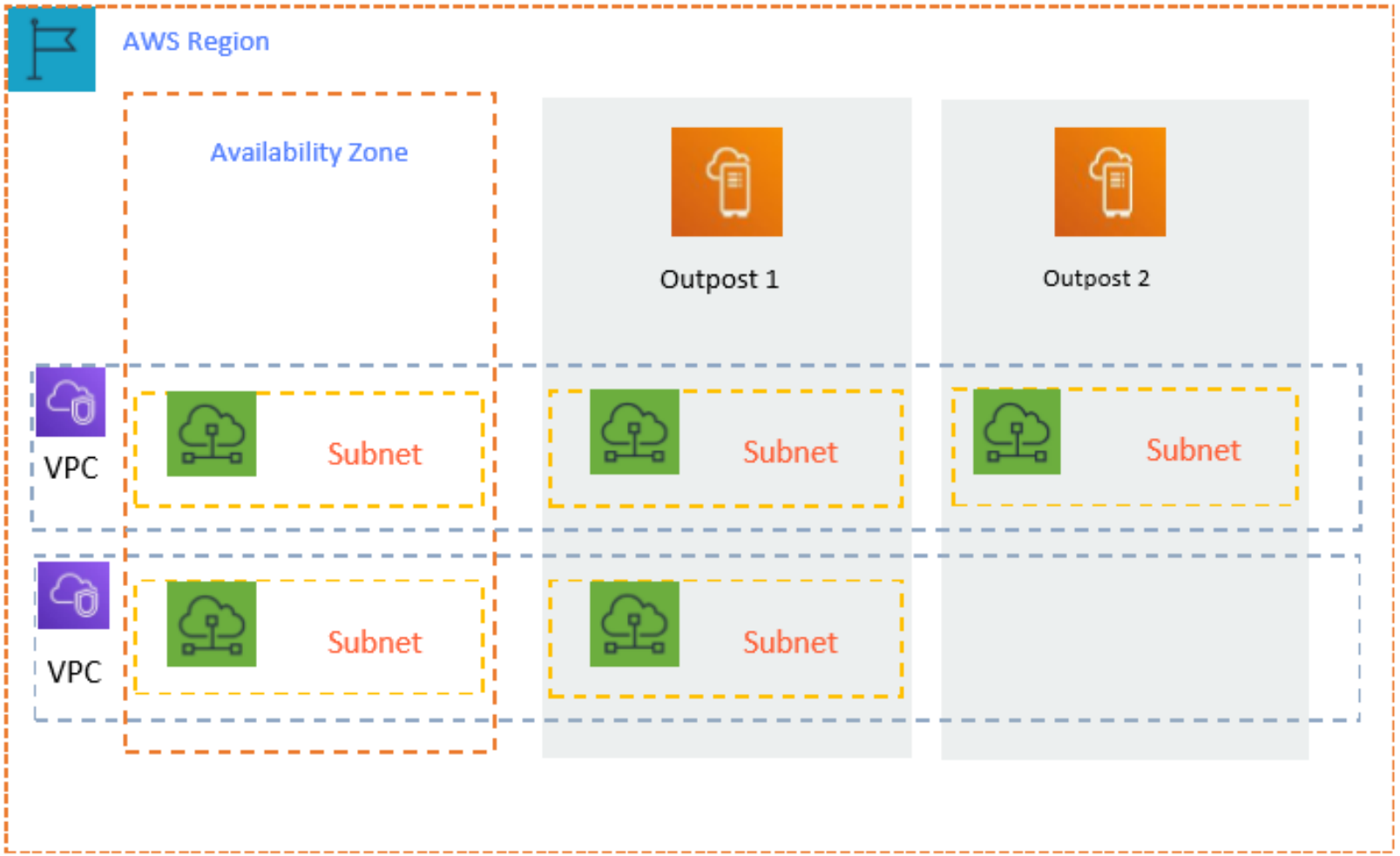
- Management Of Security Groups Becomes Very Important
- Security Groups Need Oversight

What Is Outposts?

What Is Outposts?

- Amazon AWS In A Rack
- Placed In Morningstar's Datacenter
- Presented To Team Account As Shared Subnets

What Is Outposts?



MORNINGSTAR®

Before Outposts On Premises

- Security Access Controls Using Physical Firewalls
- Networking Teams Needed To Support Firewall Requests
- Security Teams Needed To Review Firewall Requests

What Security Concerns Does Outposts Present?

- Teams Can Manage Access Using Security Groups
- No Oversight From Security Teams
- Removes Blast Radius For Teams Using Outposts

After Outposts On Premises

- Security Groups Become Very Important
- Security Groups Need Oversight

Why We Need To Automate Security Groups

What Is TGW & Outposts Security Solution?

- Remove Developer Ability to Create Security Groups
- Oversight When Developers Create Security Groups
- Automate Everything Possible

Technical Automation Solution

Goals Of Automation Solution

- Reduce Operational Burden For Security & Networking
- Static Code Analysis For Security Best Practices
- Document Service Dependencies For Team Applications

Software Used For Solution

- Bitbucket
- tfLint
- tfSec
- Terraform
- Jenkins

Software Used For Solution

- Bitbucket (GIT) – Source Code Repository

Software Used For Solution

- tfLint – Terraform Code Best Practices
- tfSec – Terraform & AWS Security Best Practices

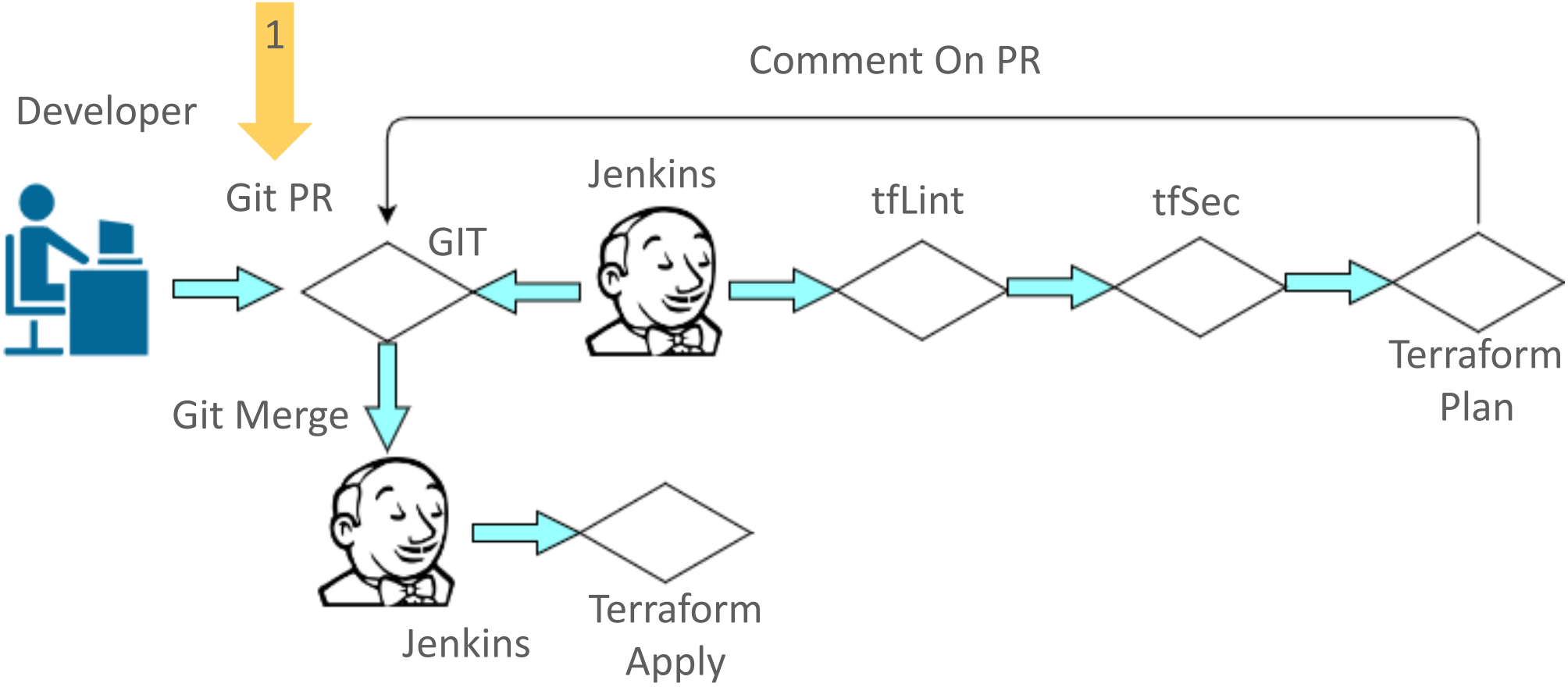
Software Used For Solution

- Terraform – Infrastructure As Code

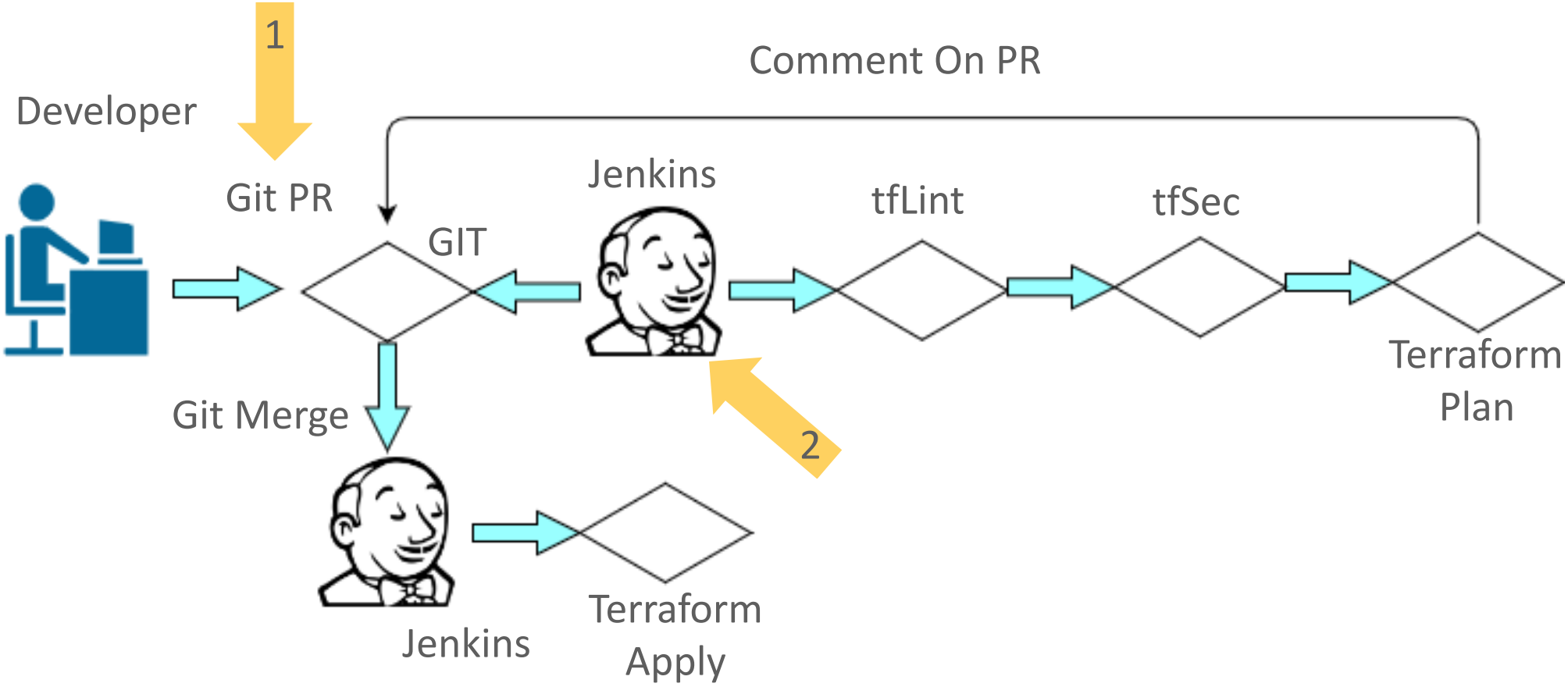
Software Used For Solution

- Jenkins – Tool To Automate All The Different Pieces

Solution Workflow

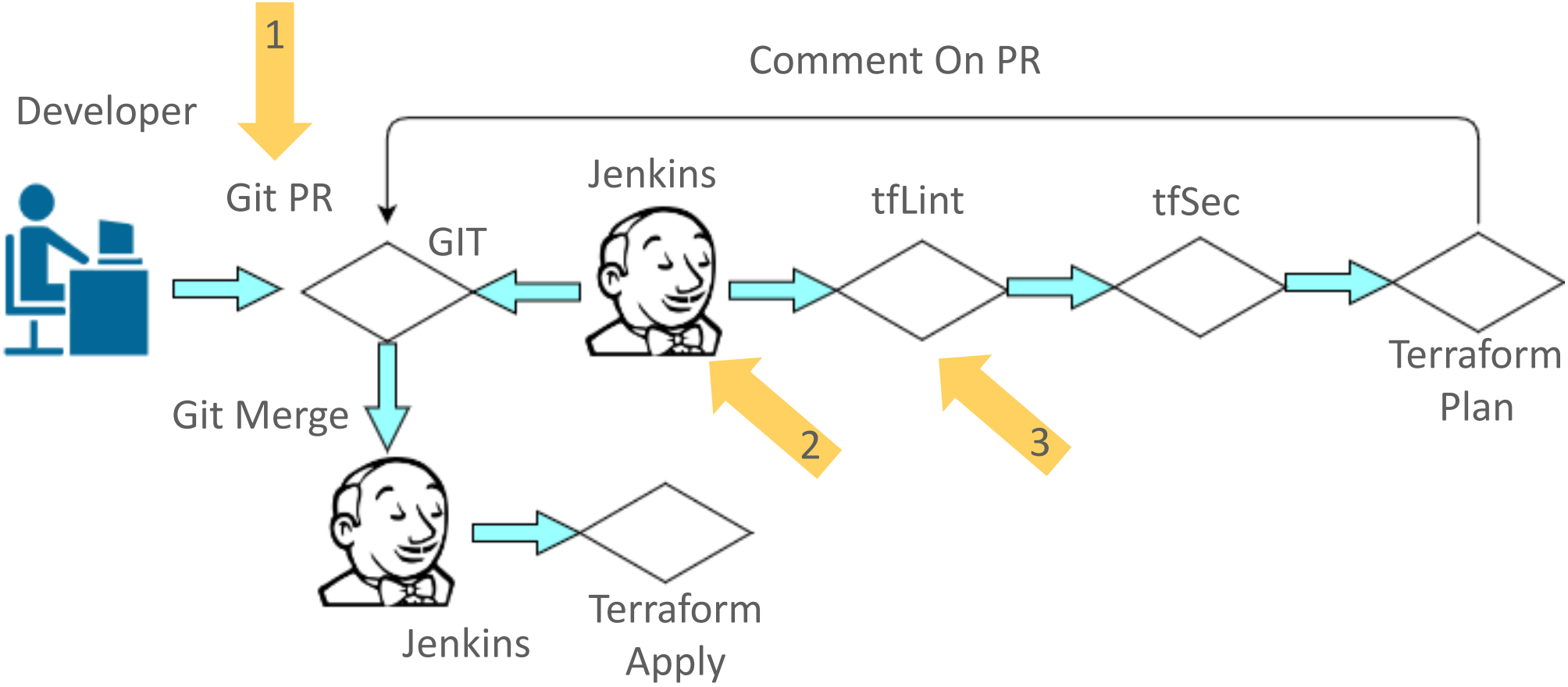


Solution Workflow



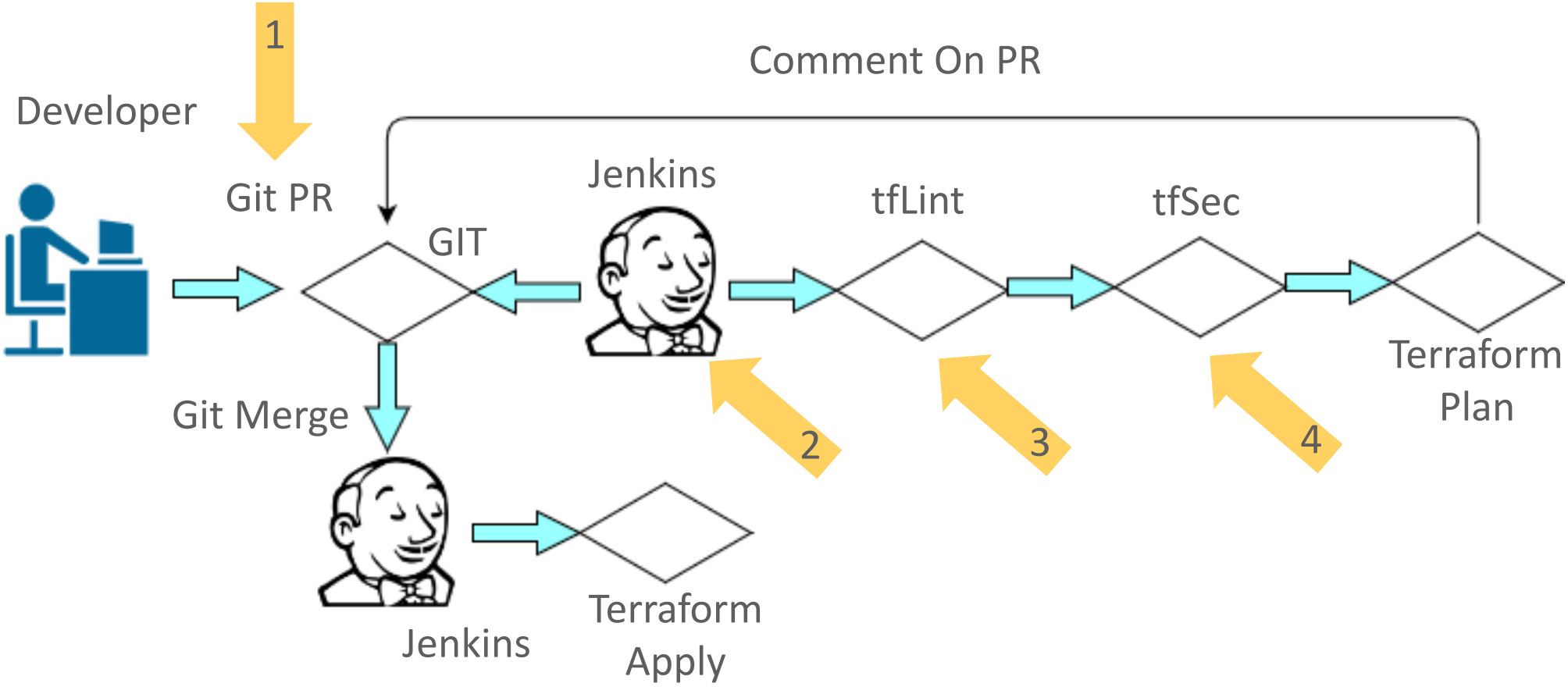
MORNINGSTAR®

Solution Workflow



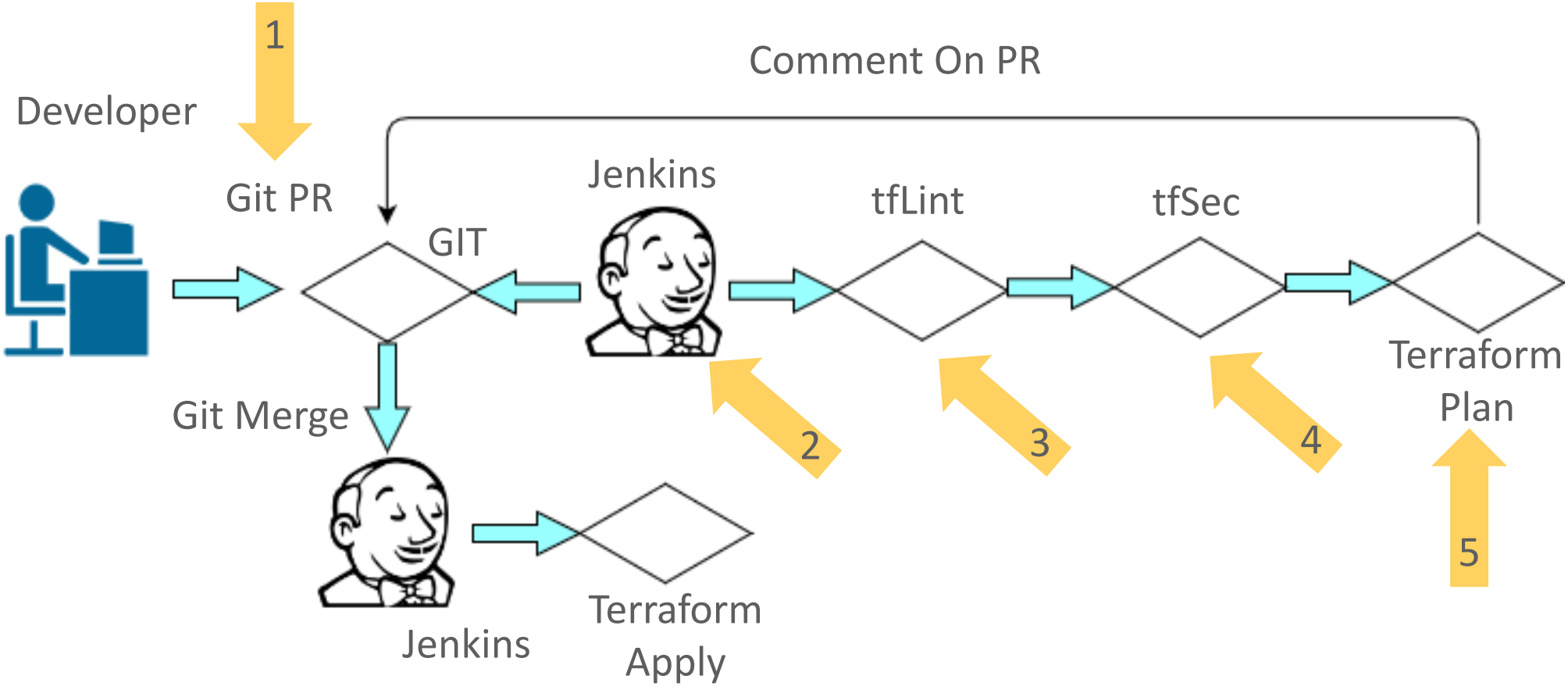
MORNINGSTAR®

Solution Workflow



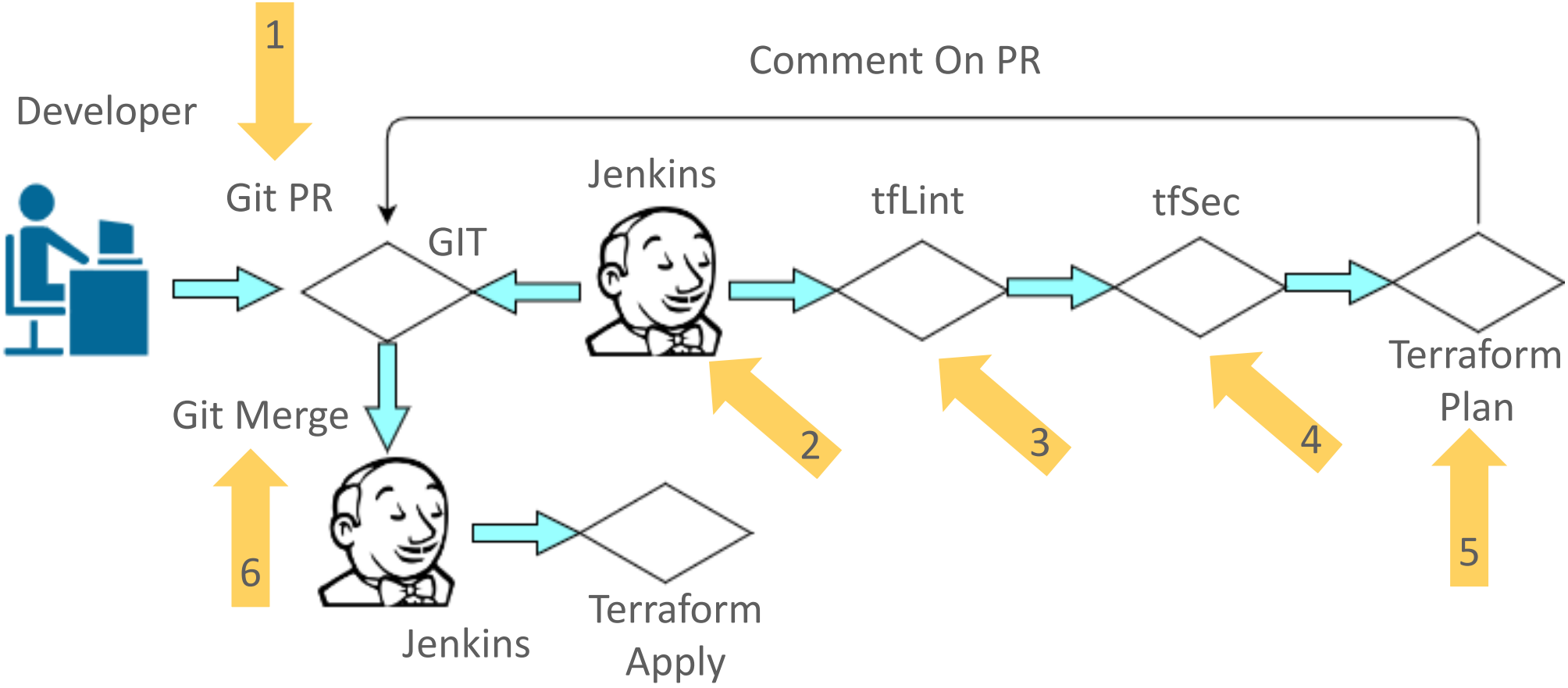
MORNINGSTAR®

Solution Workflow



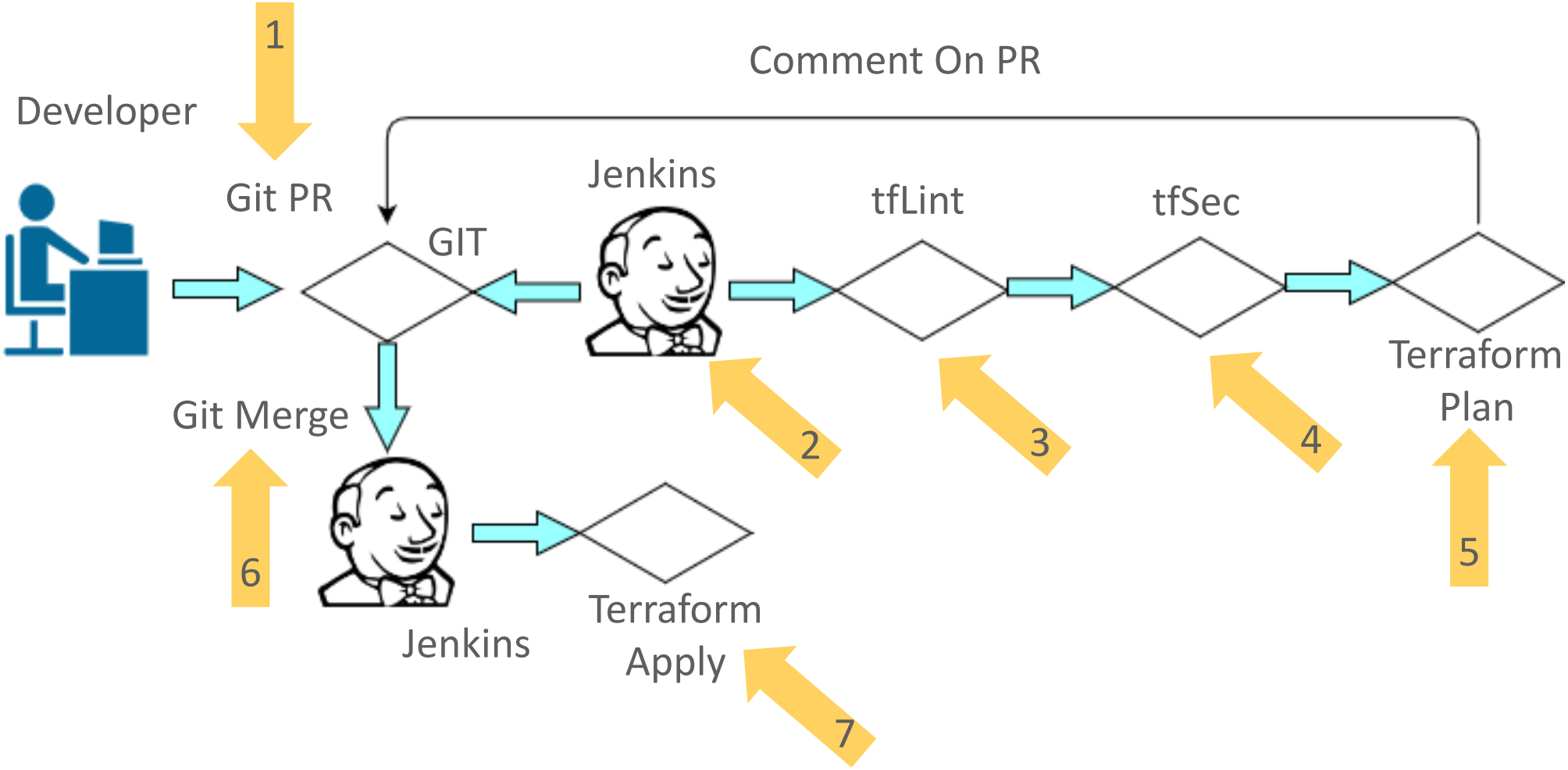
MORNINGSTAR®

Solution Workflow



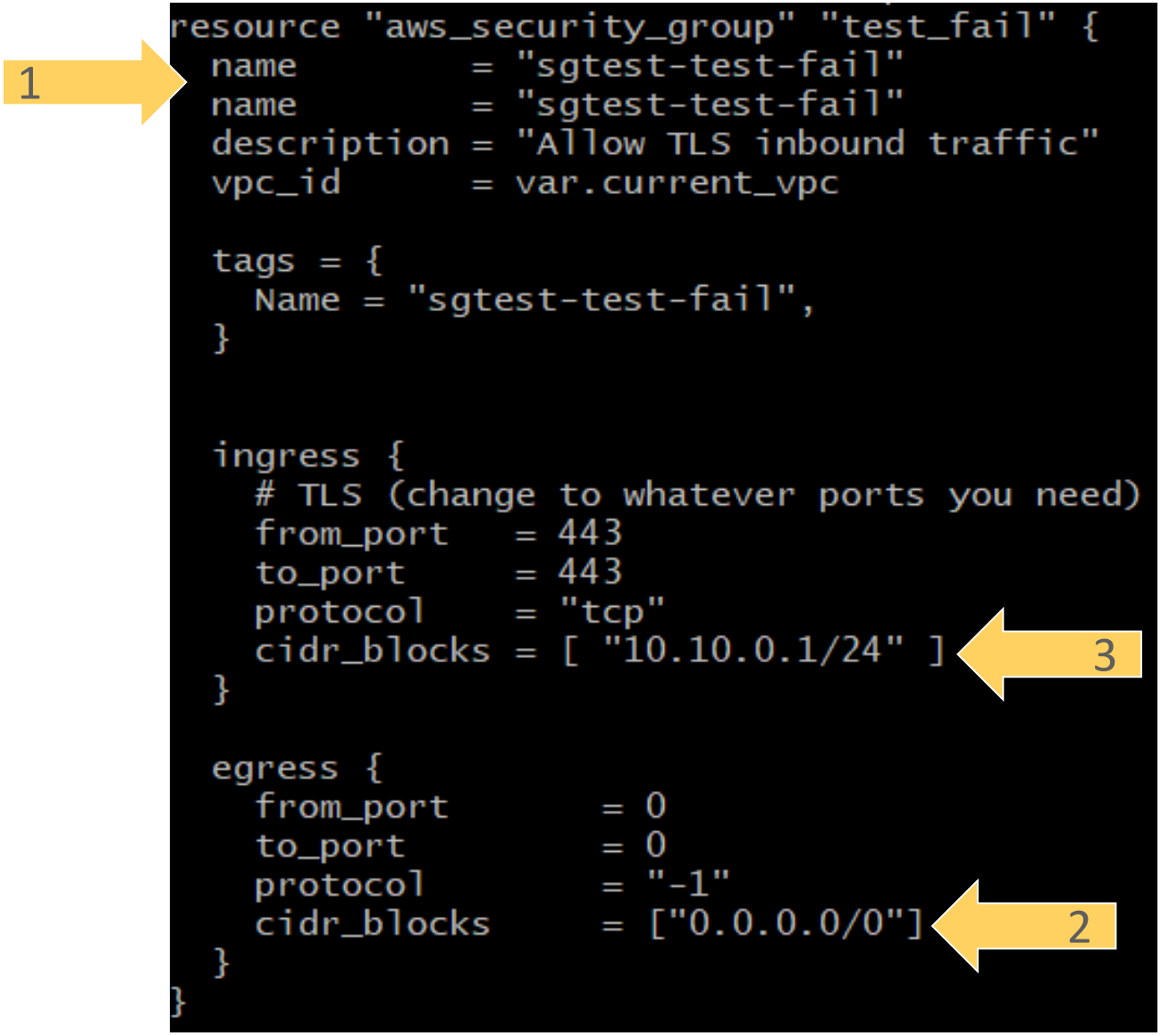
MORNINGSTAR®

Solution Workflow



MORNINGSTAR®

Developer Writes Code



MORNINGSTAR®

Developer Submits Pull Request

Diff | Commits

Find text in diff and context lines

acct-cloudsvc-non-prod/us-east-1_vpc-0df9e969cf3bc9

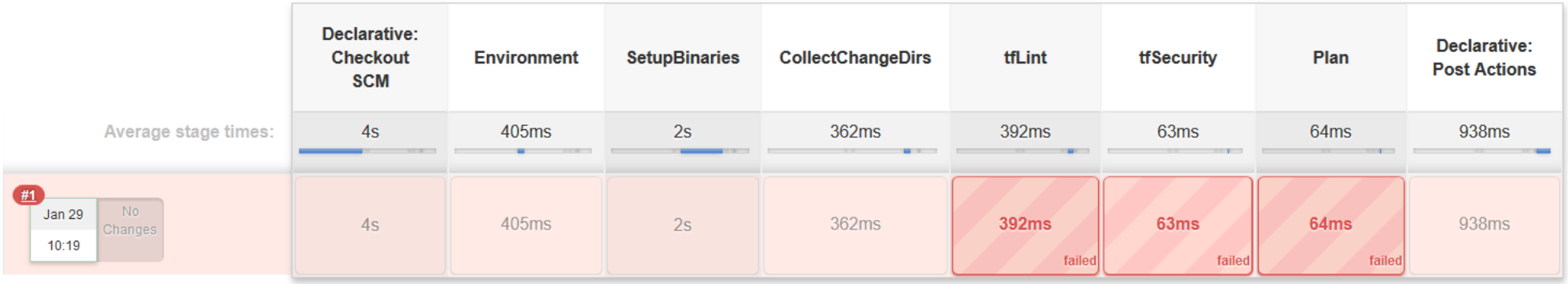
sg_test_fail.tf

acct-cloudsvc-non-prod / us-east-1_vpc-0df9e969cf3bc991d / **sg_test_fail.tf** ADDED

```
1 + resource "aws_security_group" "test_fail" {
2 +   name        = "sgtest-test-fail"
3 +   name        = "sgtest-test-fail"
4 +   description = "Allow TLS inbound traffic"
5 +   vpc_id      = var.current_vpc
6 +
7 +   tags = {
8 +     Name = "sgtest-test-fail",
9 +   }
10 +
11 +
12 +   ingress {
13 +     # TLS (change to whatever ports you need)
14 +     from_port   = 443
15 +     to_port     = 443
16 +     protocol    = "tcp"
17 +     cidr_blocks = [ "10.10.0.1/24" ]
18 +   }
19 +
20 +   egress {
21 +     from_port       = 0
22 +     to_port         = 0
23 +     protocol        = "-1"
24 +     cidr_blocks     = ["0.0.0.0/0"]
25 +   }
26 + }
```

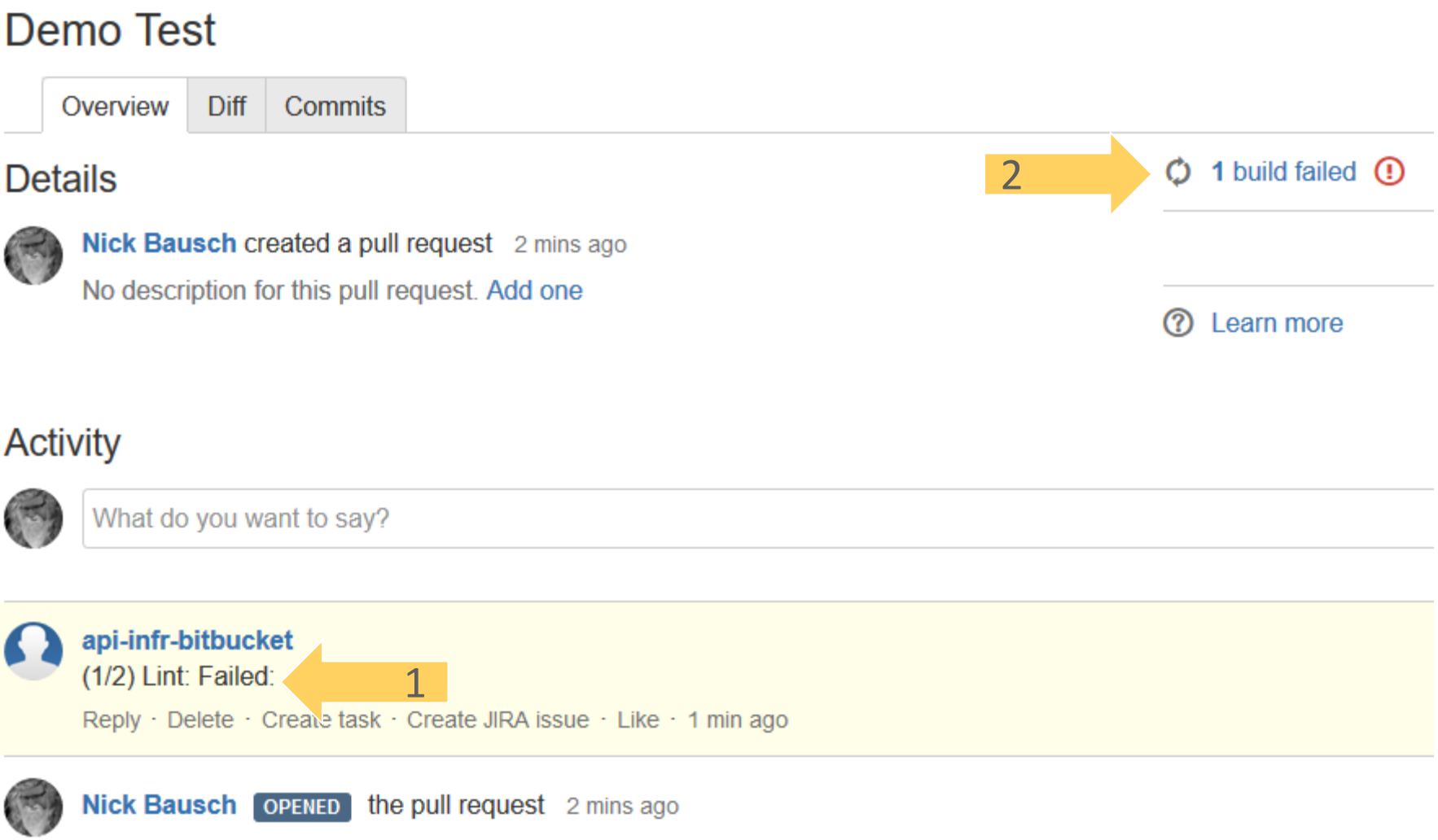
Pull Request Kicks Off Jenkins Job

Stage View



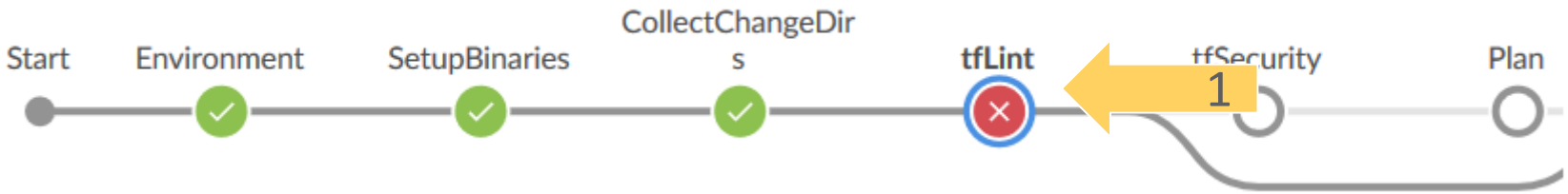
	Declarative: Checkout SCM	Environment	SetupBinaries	CollectChangeDirs	tfLint	tfSecurity	Plan	Declarative: Post Actions
Average stage times:	4s	405ms	2s	362ms	392ms	63ms	64ms	938ms
#1 Jan 29 10:19 No Changes	4s	405ms	2s	362ms	392ms failed	63ms failed	64ms failed	938ms

tfLint Failed!



MORNINGSTAR®

Why Did tfLint Fail?




```
 9
10    Error: Attribute redefined
11
12      on sg_test_fail.tf line 3, in resource "aws_security_group" "test_fail":
13       3:   name        = "sgtest-test-fail"
14
15    The argument "name" was already set at sg_test_f          . Each argument may be set
      only once.
16
17    Error: Duplicate resource "aws_security_group" configuration
18
19      on sg_test_fail.tf line 1, in resource "aws_security_group" "test_fail":
20       1: resource "aws_security_group" "test_fail" {
21
```


MORNINGSTAR®

tfSec Failed!

Activity

What do you want to say?

api-infr-bitbucket

(1/2) tfLint: Ok - (2/2) tfSecurity: Failed:

Reply · Delete · Create task · Create JIRA issue · Like · A moment ago

Why Did tfSec Fail?

```
Problem 1

  [AWS009] Resource 'aws_security_group.test_fail2' defines a fully open egress security
group.
  /data/jenkins_slave/epjenkins81/workspace/terraform-sg-automate-lint_PR-13/acct-
cloudsvc-non-prod/us-east-1_vpc-0df9e969cf3bc991d/sg_test_fail.tf:23

      20 |     from_port       = 0
      21 |     to_port         = 0
      22 |     protocol        = "-1"
      23 |     cidr_blocks     = ["0.0.0.0/0"]
      24 |   }
      25 | }
      26 |
```

Terraform Plan Failed!

Activity

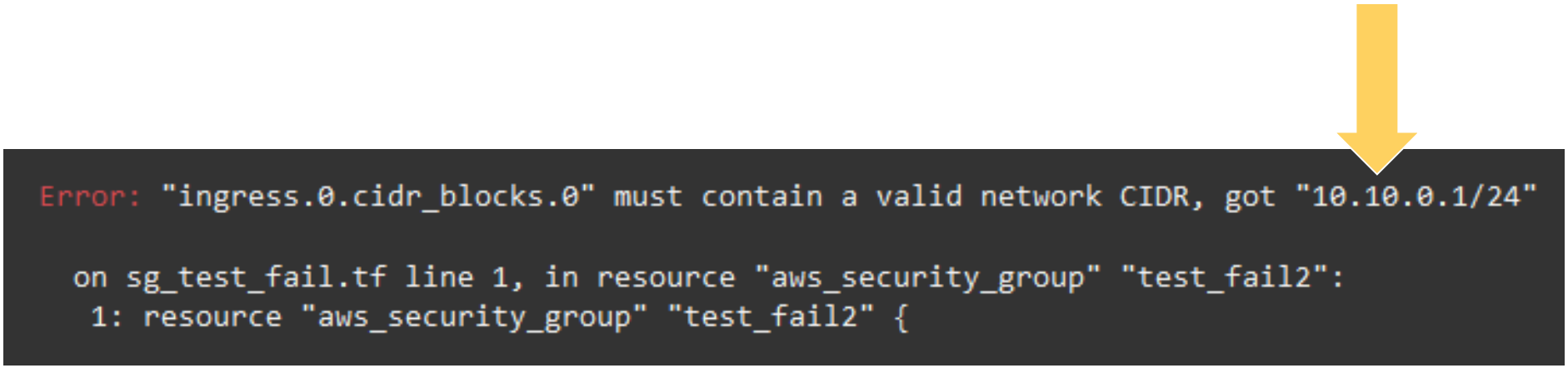
What do you want to say?

api-infr-bitbucket

(1/2) tfLint: Ok - (2/2) tfSecurity: Ok - (3/3) Plan: Failed:

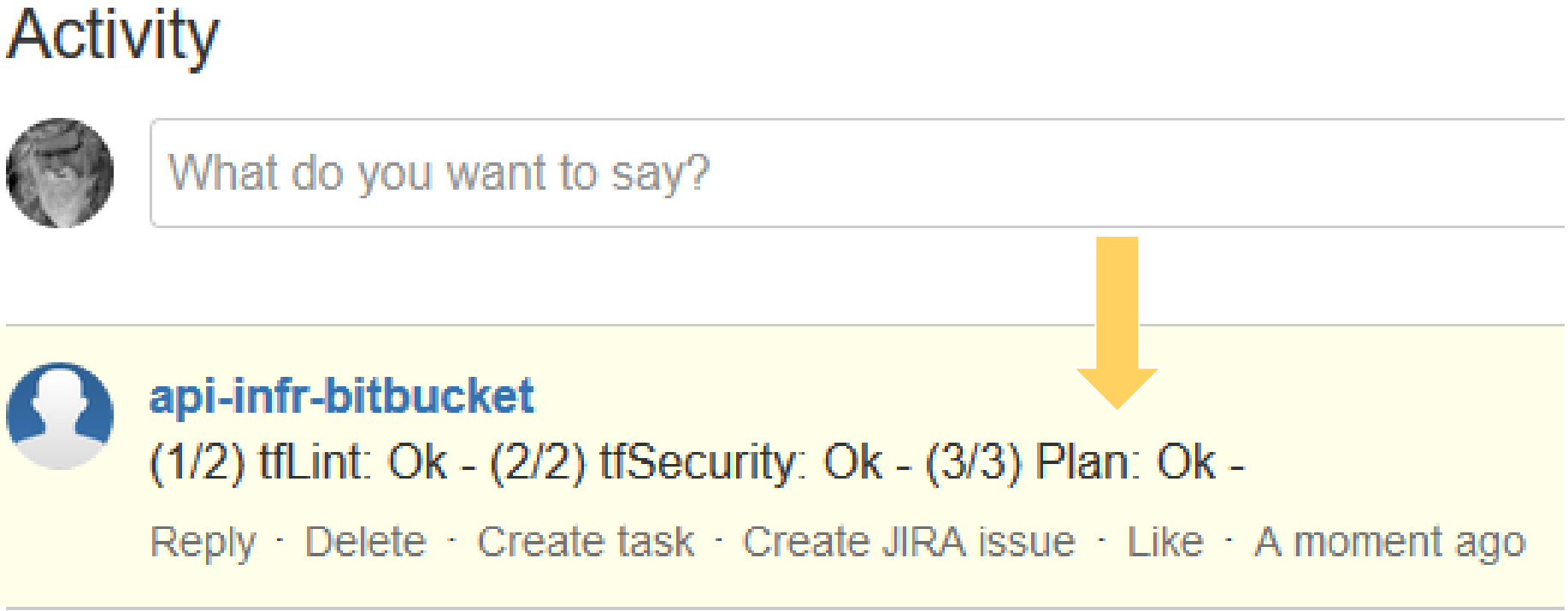
Reply · Delete · Create task · Create JIRA issue · Like · A moment ago

Why Did Terraform Plan Fail?



MORNINGSTAR®

Everything Passed

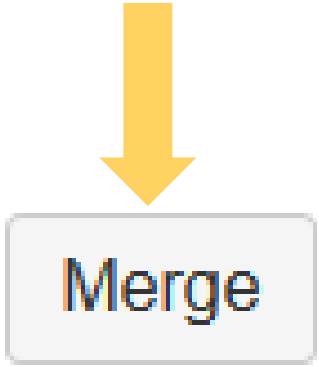


MORNINGSTAR®

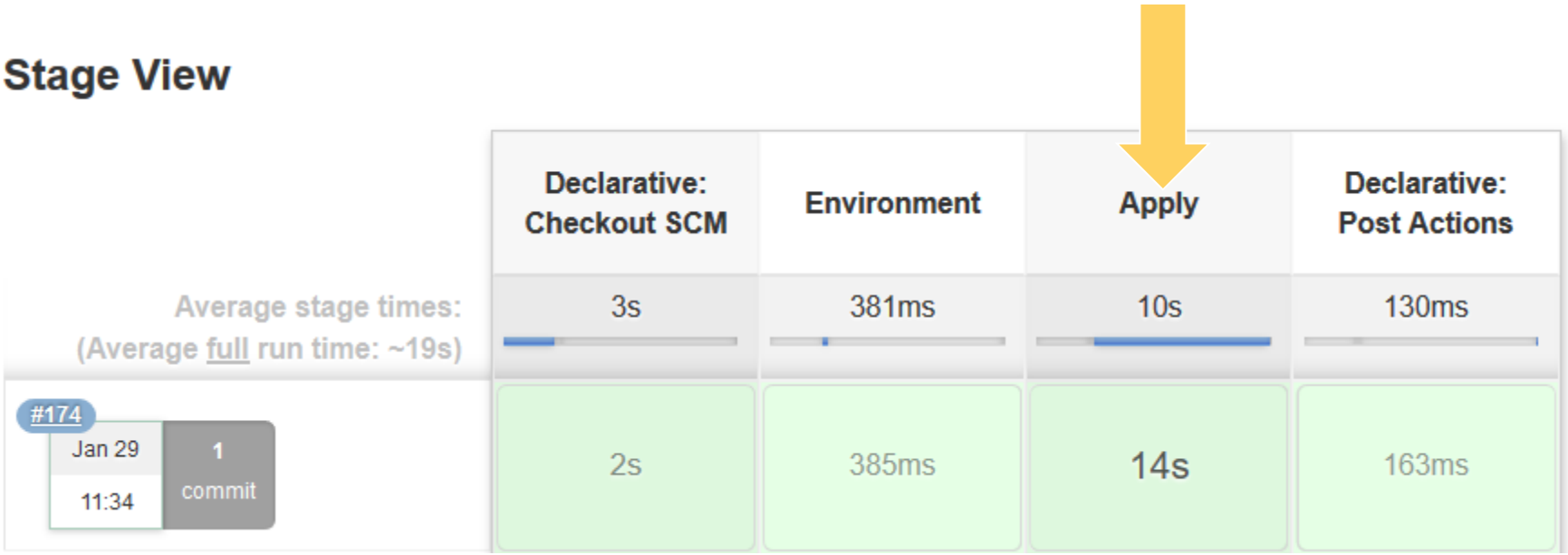
Now What?

- Networking Teams Can Review & Comment
- Security Groups Can Review & Comment
- New Custom Automations & Tests For Source Code

What Is The Final Step?



Jenkins Runs Terraform Apply



MORNINGSTAR®

Solution Workflow

- Pull Request In Bitbucket
- tfLint & tfSec & Terraform Plan
- Code Review
- Commit
- Terraform Apply

My Contributions

- Foster Everything As Code Mindset
- Multi-Account Strategy
- Provide Security Conscious AWS Environments
- Helping Build A Culture Of Continuous Learning