```
├──> Phase 5: DevSecOps and Monitoring
│      │
│      ├── Automate Infrastructure Security with Ansible/Terraform
│      │
│      └── Configure Logging and Monitoring with ELK Stack/Prometheus
│
```

Phase 5: DevSecOps and Monitoring

In this phase, you will automate infrastructure security using **Ansible** or **Terraform** and set up logging and monitoring using tools like **ELK Stack** and **Prometheus**.

Step 1: Automate Infrastructure Security with Ansible/Terraform

1. Terraform for Infrastructure Provisioning

Terraform allows you to manage your infrastructure as code (IaC), automating the creation of cloud resources such as EC2, RDS, or VPCs.

Example Terraform Configuration for AWS
File: `main.tf`

hcl
Copy code

```hcl
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "web" {
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t2.micro"
  tags = {
    Name = "Terraform-Web-Instance"
  }
}
```

Commands to Execute Terraform:

bash
Copy code

```
terraform init
terraform apply
```

---

2. Ansible for Security Configuration

Ansible can be used to enforce security policies, deploy patches, and harden infrastructure.

Example Ansible Playbook for Security
File: `security.yml`

yaml
Copy code
```
- name: Apply Security Configurations
  hosts: all
  become: true
  tasks:
    - name: Ensure UFW is installed
      apt:
        name: ufw
        state: present

    - name: Allow SSH
      ufw:
        rule: allow
        port: 22
        proto: tcp

    - name: Enable UFW
      ufw:
        state: enabled
```

Commands to Execute Ansible Playbook:

bash
Copy code
```
ansible-playbook -i inventory security.yml
```

Step 2: Configure Logging and Monitoring with ELK Stack/Prometheus

1. ELK Stack for Centralized Logging

The **ELK Stack** (Elasticsearch, Logstash, Kibana) helps centralize logs from all infrastructure resources for real-time monitoring and analysis.

Install ELK Stack (on Ubuntu):

bash
Copy code
```
# Install Elasticsearch
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo
apt-key add -
sudo apt-get install elasticsearch

# Install Logstash
sudo apt-get install logstash

# Install Kibana
sudo apt-get install kibana
```

Configure Logstash:
File: `/etc/logstash/conf.d/logstash.conf`

conf
Copy code
```
input {
  file {
    path => "/var/log/syslog"
    start_position => "beginning"
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
}
```

Start Services:

bash
Copy code
```
sudo systemctl start elasticsearch logstash kibana
```

Access **Kibana** dashboard at `http://localhost:5601`.

---

2. Prometheus for Metrics Monitoring

Prometheus collects and visualizes metrics, providing real-time insights into the health and performance of your systems.

Install Prometheus:

bash
Copy code
```
wget https://github.com/prometheus/prometheus/releases/download/v2.46.0/prometheus-2.46.0.linux-amd64.tar.gz
tar -xvzf prometheus-2.46.0.linux-amd64.tar.gz
cd prometheus-2.46.0.linux-amd64
./prometheus --config.file=prometheus.yml
```

Prometheus Configuration (*prometheus.yml*):

yaml
Copy code
```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "node_exporter"
    static_configs:
      - targets: ["localhost:9100"]
```

Install Node Exporter (for system metrics):

bash
Copy code
```
wget https://github.com/prometheus/node_exporter/releases/download/v1.6.0/node_exporter-1.6.0.linux-amd64.tar.gz
tar -xvzf node_exporter-1.6.0.linux-amd64.tar.gz
./node_exporter
```

Access **Prometheus Dashboard** at `http://localhost:9090`.

Step 3: Integrating ELK and Prometheus in CI/CD

You can enhance this setup by adding logging and monitoring capabilities to your CI/CD pipeline:

- Use **Logstash** to collect pipeline logs and send them to Elasticsearch.
- Monitor deployed applications' performance using Prometheus and create alerts for critical thresholds.

Directory Structure for Phase 5

bash
Copy code

```
/devsecops
    ├── terraform/
    │       └── main.tf           # Terraform configuration file
    ├── ansible/
    │       └── security.yml      # Ansible playbook for security
    ├── elk-stack/
    │       ├── logstash.conf     # Logstash configuration
    │       ├── elasticsearch.yml # Elasticsearch configuration
    │       └── kibana.yml        # Kibana configuration
    ├── prometheus/
    │       ├── prometheus.yml    # Prometheus configuration
    │       └── node_exporter/    # Node exporter for metrics
```

Final Workflow

1. **Infrastructure Security**:
    - Use **Terraform** for resource provisioning.
    - Use **Ansible** for enforcing security configurations.
2. **Centralized Logging**:
    - Use the **ELK Stack** to monitor application logs.
3. **Metrics Monitoring**:
    - Use **Prometheus** to monitor system and application performance.

By completing this phase, you'll have a robust DevSecOps setup with proactive logging, monitoring, and automated security enforcement.