

This configuration creates a Kubernetes cluster for application deployment and management by utilizing Amazon Web Services (AWS) EC2 instances. In order to maintain security, the infrastructure is built inside a single Virtual Private Cloud (VPC) and makes use of Network Access Control Lists (NACLs). It has two subnets: one for production and the other for testing and development.

**Setting Up a VPC:**

The complete infrastructure is housed in a single VPC, providing the Kubernetes cluster with a logically isolated network environment.

**Subnet Configuration:**

**Development and Testing Subnet: Runs four instances of EC2**

The master node in charge of managing the Kubernetes cluster is one EC2 instance.

Three EC2 instances are used as worker nodes to carry out tasks and workloads related to applications.

**Production Subnet: Has four instances of EC2:**

The master node overseeing the Kubernetes cluster is one EC2 instance.

Three EC2 instances operating in the production environment as worker nodes to carry out application duties.

**Kubernetes Cluster Setup:**

* **Master and Worker Nodes:**
  + Each subnet contains a distinct master node responsible for overseeing the Kubernetes cluster's operations.
  + The worker nodes in both subnets handle the execution of application tasks as directed by the master nodes.
* **Networking and Security:**
  + Network Access Control Lists (NACLs) are utilized to regulate inbound and outbound traffic to the subnets, enhancing network security.
  + Security Groups associated with the EC2 instances are appropriately configured to allow necessary communication between nodes while maintaining a secure environment.

**Deployment Process:**

**Application Deployment on Kubernetes:**

Deploying applications onto the Kubernetes cluster via Kubernetes manifests (YAML files) that define the desired state of the application components (pods, services, deployments, etc.).

Leveraging Kubernetes' capabilities for orchestrating and managing containerized applications across the nodes in the cluster.

**Conclusion:**

This setup establishes a robust infrastructure on AWS utilizing EC2 instances to create a Kubernetes cluster. The segregated subnets cater to distinct phases, ensuring efficient testing and development separate from the production environment. Leveraging Kubernetes facilitates the seamless deployment and management of applications across the configured nodes.