

CONTRIBUTIONS:

This is the current contributing page for Agorakube:

How to contribute

<Beginning of file>...

Testing

We use Ansible-lint with GithubActions to test Ansible code quality.

Sample needs

All contributions are welcome :)

Here is a non-exhaustive list of contributions needed for this project:

- Documentation: Add features lagorakube new runtime support, new Ingress Controller installation, new CNI plugin, etc...
- Code Quality: Improve the code quality

Beginner-Friendly Tasks (Added)

New to contributing? Here are some beginner-friendly tasks to get started:

- Update existing documentation with clearer instructions or examples.
- Fix simple bugs or typos in the codebase.
- Add comments to clarify the purpose of functions or sections of code.

Code Contribution Workflow (Added)

Contributing code to Agorakube follows a standard workflow:

1. Fork the AGORAKUBE-CORE repository.
2. Create a new branch for your feature or bug fix.
3. Make changes, commit them with clear and concise messages following our guidelines.
4. Push your changes to your fork.
5. Submit a pull request to AGORAKUBE-CORE, detailing what changes you've made.

Code Review Process (Added)

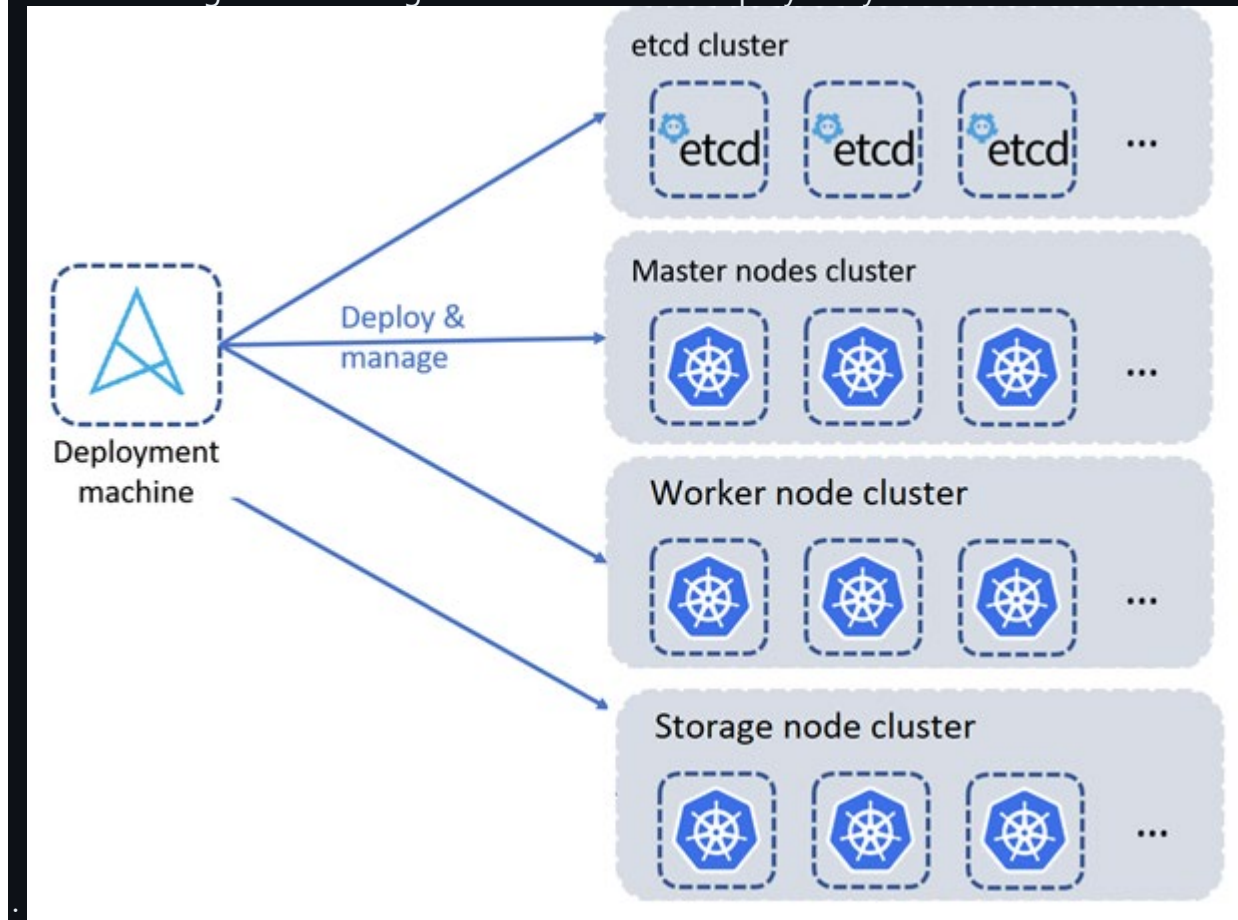
Pull requests are reviewed by maintainers to ensure quality and adherence to project standards. Please be patient during the review process and be responsive to any feedback or questions raised.

Submitting changes

INSTRUCTIONS:

High-level Architecture

Below is a diagram of the high-level architecture deployed by AGORAKUBE



Notes : This distribution is aimed to be customizable so you can choose : Agorakube's architecture is designed to be highly customizable, allowing users to configure various aspects according to their specific needs. This includes the deployment location of etcd, the number of master nodes, etcd nodes, worker nodes, and storage nodes.

- Where the **etcd** will be deployed (with the master or not)

- The number of **master** nodes to deploy (from 1 to many - 5 nodes for production)
- The number of **etcd** nodes to deploy (from 1 to many - 5 nodes for production)
- The number of **worker** nodes to deploy (from 1 to many)
- The number of **storage** nodes to deploy (from 0 to many - 3 nodes for production needs)

Components: The architecture likely consists of several key components, including:

- Master Nodes: Responsible for managing the cluster and scheduling tasks.
- Etcd Nodes: Used for distributed key-value store, often critical for cluster coordination and configuration management.
- Worker Nodes: Responsible for running application workloads.
- Storage Nodes: Optionally deployed for persistent storage needs.

Scalability: The architecture supports scaling both horizontally and vertically, allowing users to add more nodes as needed to accommodate increased workload or redundancy requirements.

Production Considerations: Production deployments typically involve multiple master nodes for high availability and fault tolerance, along with a recommended number of etcd nodes and storage nodes to meet performance and reliability needs.