***Argo CD Questionnaire***

**Overview**

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes that enables the automated deployment and management of applications. It synchronizes applications with their desired state stored in Git repositories, allowing developers to manage Kubernetes resources as code. Argo CD supports various deployment strategies, including Helm charts, Kustomize, and plain YAML manifests. It provides a user-friendly web interface, CLI, and API to monitor application health and synchronization status. To test your knowledge of Argo CD, you can take the Argo CD assessment, which covers key concepts like application management, sync policies, and multi-cluster setups.

Take this assessment to check where do you stand in the Argo CD with GotOps journey.

**30 Question/Answer set**

**What is Argo CD primarily used for?**

Continuous Integration (CI)  
Continuous Deployment (CD)  
Cloud service management  
Container orchestration

### In Argo CD, which component is responsible for monitoring changes in Git repositories?

Controller  
Application Controller  
Repository Server  
API Server

### In the context of Argo CD, what does the term “Application” refer to?

A deployment pipeline  
A configuration file for Kubernetes  
A group of Kubernetes resources as defined by a manifest.  
A Git repository that stores application code

### What is the ****Argo CD CLI**** command used to create a new project?

argocd app create <app-name> --project  
argocd project new <project-name>  
argocd create project <project-name>

argocd project create <project-name>

### What is the purpose of the Argo CD ****API server****?

To store Git repositories  
To serve the UI and expose APIs for interaction with Argo CD  
To deploy applications to Kubernetes clusters  
To manage application synchronization

### Which of the following is ****not**** a valid Argo CD Application Source Type option?

Helm Chart  
Kustomize  
GitOps  
Ansible Playbook

### What does the term ****GitOps**** refer to in the context of Argo CD?

Using Git for application lifecycle management  
A tool for managing Kubernetes nodes  
A CI/CD pipeline integration tool  
A service for managing Kubernetes namespaces

**Which of the following is required to deploy an application to multiple clusters in Argo CD?**

Different namespaces for each cluster  
Different Argo CD projects for each cluster  
Multiple application manifests for each cluster

A Kubernetes kubeconfig file for each cluster

### In Argo CD, which component is responsible for managing the application deployment process?

Controller  
Application Controller  
Repository Server  
API Server

### How does Argo CD achieve application auto synchronization with the Git repository?

By manually triggering sync from the CLI  
By continuously polling the Git repository  
By using a Kubernetes CronJob  
By running periodic Docker containers

### Which Argo CD feature allows automatic creation of Kubernetes namespaces when syncing an application?

Sync Hooks  
Resource creation policy

Auto-namespace creation  
Application manifests

**In Argo CD, the App of Apps pattern is used to?**

Automatically sync all applications in a project  
Rollback an application to a previous version  
Create applications based on a template

Deploy a parent application that manages multiple child applications

### What happens when Argo CD tries to sync an Application with ****auto namespace creation**** enabled, but the namespace already exists?

Argo CD will delete the existing namespace and recreate it  
The sync will fail due to an existing namespace conflict  
Argo CD will merge the namespaces together

Argo CD will skip the namespace creation step and continue syncing

### What is the purpose of ****Argo CD Projects****?

To define the settings for application sync  
To create backup copies of Kubernetes resources  
To automatically scale applications across clusters

To group and manage applications under a common set of policies and resources

### Which sync policy in Argo CD ensures that changes are automatically applied when the Git repository is updated?

Manual sync  
Periodic sync

Auto-sync  
Rollback sync

### How can you list all Argo CD applications within a specific project using the ****Argo CD CLI****?

argocd project list-applications <project-name>  
argocd app list --project <project-name>  
argocd project get <project-name> --apps  
kubectl get applications --project <project-name>

### Which of the following is true about ****Argo CD Projects****?

Projects are used to define user roles and permissions for specific applications  
Projects are tied to a single namespace and can only manage one application at a time  
Projects provide a mechanism to restrict which repositories, clusters, and namespaces can be used by the applications within that project

Projects are used exclusively for application rollback management

### What is a ****Sync Policy**** in Argo CD?

A set of rules for automatically backing up application data  
A configuration for how an application should sync with the Git repository (automatic or manual sync)  
A method for organizing applications into projects  
A policy for controlling access to Kubernetes clusters

### What does ****Argo CD Rollback**** do?

It reverts an application to its previous selected version   
It deletes the application from the Kubernetes cluster  
It pushes new changes from Git to the Kubernetes cluster  
It resets the entire project configuration

### Which of the following commands is used to display application details in Argo CD using the CLI?

argocd app info <app-name>  
argocd app get <app-name>  
argocd get status <app-name>  
argocd app sync-status <app-name>

### What is the default ****reconciliation time**** for auto-sync in Argo CD?

60 seconds  
  
300 seconds  
600 seconds

180 seconds

**What does the "prune" feature in ArgoCD do?**

It automatically removes unused Kubernetes resources from the cluster  
It updates Helm charts to their latest versions  
It syncs resources across multiple clusters  
It adds new resources to the cluster based on Git changes

**What is the primary goal of multi-cluster management in Argo CD?**

To deploy applications to multiple clusters using a single configuration  
To scale the number of applications in a single cluster  
To monitor application metrics across clusters  
To automate resource pruning across all clusters

**Which of the following monitoring systems is commonly used with Argo CD metrics?**

Prometheus  
Datadog  
Grafana  
All of the above

### How can ****Argo CD**** use ****Helm**** charts for application deployment?

By referencing Helm charts stored in Git repositories or external chart repositories  
By using Kubernetes' native kubectl command to deploy Helm charts  
By creating Kubernetes YAML manifests from Helm charts manually  
By using a separate Helm server outside of Argo CD

**What is a key advantage of using the App of Apps pattern in Argo CD?**

It allows for complex applications to be managed in a modular way  
It automatically scales applications based on load  
It synchronizes Git repositories faster  
It eliminates the need for Kubernetes namespaces

**How can you declaratively create applications in Argo CD?**

By using the Argo CD UI and manually inputting settings  
By writing YAML manifests for the application and storing them in a Git repository  
By configuring Helm charts for each application  
By using a Kubernetes kubectl command for application deployment

### Which of the following is a core feature of Argo CD?

Multi-cluster management  
Code linting  
Application build pipelines  
Container registry management

**Answer:** A) Multi-cluster management

**Which Argo CD API endpoint can you use to retrieve application health metrics?**

/api/v1/applications/<app-name>/metrics  
/metrics  
/api/v1/metrics/<namespace>  
/api/v1/apps

### When deploying applications from ****private Git repositories**** in Argo CD, what must be configured to enable access?

API keys in the Argo CD configuration  
SSH keys or personal access tokens for authentication  
ConfigMaps with repository credentials  
A Kubernetes service account with admin privileges