**Release 2.3.0:**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#release-230)

Release 2.3.0 is a (minor) feature release in the 2.x series.

**Upgrade notes:**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#upgrade-notes)

* The 2.3.0 release does not support rollback to the 2.1.x releases.
* When upgrading the control-plane from any version before 2.2.0, due to [FNDD-6200 Control-plane upgrade doesn't upgrade the hitachi-keycloak image](https://hv-eng.atlassian.net/browse/FNDD-6200) the user must modify the configuration to change the image tag for the keycloak image. The full value will include the registry name. The user must change the tag portion of the key keycloakoperator.keycloakImage from hitachi-keycloak:11.0.1-hv1 to hitachi-keycloak:12.0.3-hv2. If this is missed during upgrade, it can be changed afterwards.
* Upgrade of control-plane from older version to 2.3.x is only supported on kubernetes version 1.18. For more information on supported platforms for particular version please see [Supported Platforms](http://docs.foundry.wal.hds.com/docs/SupportedPlatforms)

See [Upgrading the Solution Control Plane](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/UpgradingTheControlPlane) for specific steps to upgrade the control plane. For more details on how to upgrade your solution, please see [Upgrading solutions](http://docs.foundry.wal.hds.com/docs/DevelopingSolutions/Upgrade)

**Kubernetes resources deprecation warnings**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#kubernetes-resources-deprecation-warnings)

Foundry 2.3.0 is now supported on kubernetes versions 1.18 up to 1.21. As a part of Foundry 2.3.0 installation on kubernetes version 1.19+, following set of kubernetes resources deprecation warnings can be seen and can be safely ignored.

W0625 07:25:33.723958 22893 warnings.go:70] apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition

W0625 07:25:35.180761 22893 warnings.go:70] rbac.authorization.k8s.io/v1beta1 ClusterRole is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 ClusterRole

W0625 07:25:35.240835 22893 warnings.go:70] rbac.authorization.k8s.io/v1beta1 ClusterRoleBinding is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 ClusterRoleBinding

W0625 07:25:35.286893 22893 warnings.go:70] rbac.authorization.k8s.io/v1beta1 Role is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 Role

W0625 07:25:35.304826 22893 warnings.go:70] rbac.authorization.k8s.io/v1beta1 RoleBinding is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 RoleBinding

W0625 07:25:35.345863 22893 warnings.go:70] admissionregistration.k8s.io/v1beta1 MutatingWebhookConfiguration is deprecated in v1.16+, unavailable in v1.22+; use admissionregistration.k8s.io/v1 MutatingWebhookConfiguration

W0625 07:25:35.350660 22893 warnings.go:70] admissionregistration.k8s.io/v1beta1 ValidatingWebhookConfiguration is deprecated in v1.16+, unavailable in v1.22+; use admissionregistration.k8s.io/v1 ValidatingWebhookConfiguration

W0625 06:23:21.123726 243 warnings.go:70] policy/v1beta1 PodDisruptionBudget is deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget

**New Features in release 2.3.0:**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#new-features-in-release-230)

* The Foundry install/uninstall scripts have been moved into container images that wrap each utility. This means that "helm" and "yq" are no longer required on the installation host. Wrapper scripts have been created that (mostly) keep the same API as the previous scripts. The only differences are that:

\* the "utility image" concept is gone, along with the '-X' and '-x' arguments to scripts and the 'utilityimage' key in config files.

\* DISABLE\_DOCKER support has been removed. If you need to run a docker command to execute our scripts, please use our images.

* Foundry install scripts will now install istio 1.10.4 instead of 1.6.8.
* All the solution helm charts should now be compatible with helm 3.6.3.
* Foundry install scripts will now install cert-manager 1.5.3 instead of 0.15.1
* Foundry deployment and functionality is supported on RHEL 8.4.
* Foundry now provides a mechanism to register a solution chart already exists in the registry that the solution control plane is configured to use.

See [Manual Installation of Solution Charts](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/ManualInstallation/ManualInstallationSolutionCharts) for more information.

* Foundry now runs a cluster url pre-check that will make sure that the cluster url provided is accessible. See the install-control-plane.sh help for more information and options.
* Foundry deployment is now supported on on-prem OpenShift platform. See [Foundry On OpenShift](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/FoundryOnOpenshift)

**Issues fixed in release 2.3.0:**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#issues-fixed-in-release-230)

* [FNDD-7583 Gatekeeper needs to account for the path attribute when splitting cookies](https://hv-eng.atlassian.net/browse/FNDD-7583) When a user has a large number of permissions the kc-access cookie set by Gatekeeper is split into multiple Cookies. Gatekeeper now accounts for the length of the path so that the cookie is split correctly and accepted by all browsers.
* [FNDD-7653 Forge modifies solution charts in a way that is incompatible with helm 3.x, where x < 4](https://hv-eng.atlassian.net/browse/FNDD-7653) When Forge is used to package charts, the order of files inside the tar archive can be ordered in a way that causes older Helm versions to fail during operations. Forge has been fixed to no longer order files in this way.
* [FNDD-7911](https://hv-eng.atlassian.net/browse/FNDD-7911) After control plane upgrade control plane link is giving error. Workaround is to restart all the pod in control-plane namespace.
* [FNDD-7923](https://hv-eng.atlassian.net/browse/FNDD-7923) Idempotency of upgrade-cluster-service.sh fails.
* [FNDD-7850](https://hv-eng.atlassian.net/browse/FNDD-7850) uninstall-control-plane.sh is not cleaning keycloakoperator-post-install from hitachi-solution namespace.
* [FNDD-8160](https://hv-eng.atlassian.net/browse/FNDD-8160) Getting 403 error while creating/editing/cloning network routes.
* [FNDD-8192](https://hv-eng.atlassian.net/browse/FNDD-8192) Istio envoy header check is failing when we configure Application Load Balancer with AWS EKS kubernetes cluster.
* [FNDD-8155](https://hv-eng.atlassian.net/browse/FNDD-8155) Uninstall-control-plane.sh script is not working on OpenShift on-prem setup.
* [FNDD-8156](https://hv-eng.atlassian.net/browse/FNDD-8156) On on-prem OpenShift setup installation scripts are working only with --verbose option.
* [FNDD-8157](https://hv-eng.atlassian.net/browse/FNDD-8157) Foundry installation on OpenShift on-prem setup is failing.
* [FNDD-8122](https://hv-eng.atlassian.net/browse/FNDD-8122) Secured registry secrets are not getting replaced for cert-manager and istio.
* [FNDD-8135](https://hv-eng.atlassian.net/browse/FNDD-8135) Kiali login should not happen with an anonymous user.
* [FNDD-8150](https://hv-eng.atlassian.net/browse/FNDD-8150) Install cluster services is failing for openshift cluster

**Metrics Addon releases:**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#metrics-addon-releases)

**Release 1.0.0:**[**Release Notes**](http://docs.foundry.wal.hds.com/addons/metricsaddon/docs/1.0.0/UserManuals/ReleaseNotes)[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#release-100-release-notes)

**Release 1.0.1:**[**Release Notes**](http://docs.foundry.wal.hds.com/addons/metricsaddon/docs/UserManuals/ReleaseNotes)[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#release-101-release-notes)

**Open issues:**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#open-issues)

**Newly identified Open Issues in release 2.3.0**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#newly-identified-open-issues-in-release-230)

* [FNDD-7929](https://hv-eng.atlassian.net/browse/FNDD-7929) Istio init container should run with non root user.
* [FNDD-8154](https://hv-eng.atlassian.net/browse/FNDD-8154) keycloakclient fails to get created/reconciled when installing a solution, need to restart keycloak-operator.
* [FNDD-8035](https://hv-eng.atlassian.net/browse/FNDD-8035) Navigation menu is showing on left side in Fedora.
* [FNDD-7980](https://hv-eng.atlassian.net/browse/FNDD-7980) Istio side cars are not starting up quickly enough.
* [FNDD-7862](https://hv-eng.atlassian.net/browse/FNDD-7862) Uninstalling Messaging Service solution shows as successful on UI/API while its resources are still in termination state.
* [FNDD-7845](https://hv-eng.atlassian.net/browse/FNDD-7845) Logs stop being forwarded from Fluentd to elasticsearch.
* [FNDD-7843](https://hv-eng.atlassian.net/browse/FNDD-7843) SSO implementation hard-codes upstream port to first service port.
* [FNDD-7441](https://hv-eng.atlassian.net/browse/FNDD-7441) Foundry has a NullPointerException when a chart is missing appVersion
* [FNDD-8129](https://hv-eng.atlassian.net/browse/FNDD-8129) [Qualys scan CS] EOL/Obsolete Operating System: CentOS 8.x detected by Qualys CS. Foundry admin-app and all utility images uses CentOs 8.4 as base image, which has been marked as EOL in December 2021. CentOS 8.x is included in this release as a base image for one of the containers. Foundry team intends to replace CentOS 8.x base image with CentOS Stream 8 in Foundry 2.4.0 release. The Foundry container using CentOS 8.x as base image is not exposing any port that can be directly accessed from public network hence reducing the probability of exploiting any vulnerability exposeded by CentOS 8.x post its EOL. At the time of this (i.e. 2.3.0) release there is no known critical vulnerability explosed by CentOS 8.x.

**Security issues in release 2.3.0**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#security-issues-in-release-230)

Following issues are already triaged and are planned for the next release.

* [FNDD-7933](https://hv-eng.atlassian.net/browse/FNDD-7933) [Keycloak] (Qualys scan CS) Keycloak related issues
* [FNDD-7934](https://hv-eng.atlassian.net/browse/FNDD-7934) [Kibana] (Qualys scan CS) Kibana related issues.
* [FNDD-7935](https://hv-eng.atlassian.net/browse/FNDD-7935) [Logging-solution-operator] (Qualys scan CS) Logging-solution-operator related issues.
* [FNDD-8073](https://hv-eng.atlassian.net/browse/FNDD-8073) [Keycloak] (Owasp scan CS) Keycloak related critical issues
* [FNDD-8074](https://hv-eng.atlassian.net/browse/FNDD-8074) [Tomcat] (Owasp scan CS) Tomcat related Critical issues
* [FNDD-8075](https://hv-eng.atlassian.net/browse/FNDD-8075) [Elasticsearch] (Owasp scan CS) Elasticsearch related Critical issues
* [FNDD-8123](https://hv-eng.atlassian.net/browse/FNDD-8123) [Elasticsearch] (Qualys scan VM) elasticsearch container related issues
* [FNDD-8128](https://hv-eng.atlassian.net/browse/FNDD-8128) [Istio-Kiali] (Qualys scan CS) Istio-Kiali related issues.

**Installation**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#installation)

* Installing to a namespace composed of all digits will fail. As a workaround, use a namespace that contains alphabetic characters. ([FNDD-4586](https://hv-eng.atlassian.net/browse/FNDD-4586))
* When install-control-plane.sh finds that the control plane has already been installed in the target namespace, it says to use helm to delete the control plane solution. This is not recommended. If you wish to uninstall the existing control plane in this namespace, use the uninstall-control-plane.sh script as describe in [Uninstalling the Control Plane](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Uninstallation/UninstallControlPlane). ([FNDD-3427](https://hv-eng.atlassian.net/browse/FNDD-3427))
* When install-control-plane.sh finds that the control plane has already been installed in the target namespace, it says to use helm to delete the control plane solution. This is not recommended. If you wish to uninstall the existing control plane in this namespace, use the uninstall-control-plane.sh script as describe in [Uninstalling the Control Plane](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Uninstallation/UninstallControlPlane)). ([FNDD-3427](https://hv-eng.atlassian.net/browse/FNDD-3427))
* It is possible when running install-control-plane.sh for the checks that wait for Keycloak or the control plane to be available to time out. When this happens it is likely that an intermittent install failure occurred. Check if the pods in the installed namespace are healthy, if they are not healthy after 10 minutes the simplest resolution is to uninstall and reinstall the control plane. ([FNDD-4756](https://hv-eng.atlassian.net/browse/FNDD-4756) and [FNDD-4553](https://hv-eng.atlassian.net/browse/FNDD-4553))
* Installing cluster services may take 15 minutes to exit if it fails to validate resources. The output from the install-cluster-services script would look something like:

Istiod encountered an error: failed to wait for resource: resources not ready after 5m0s: timed out waiting for the condition Deployment/istio-system/istiod

Egress gateways encountered an error: failed to wait for resource: resources not ready after 5m0s: timed out waiting for the conditiont/istio-system/istio-ingressgateway

* If this occurs you may terminate the script if you do not wish to wait. ([FNDD-4830](https://hv-eng.atlassian.net/browse/FNDD-4830))
* When using install-cluster-services.sh, some installation errors may require manual cleanup before the script can be used again to successfully install cluster services. For example, if you attempt to install using an incorrect registry URL, you need to run uninstall-control-plane.sh -F before you can run the script again with the correct URL ([FNDD-7209](https://hv-eng.atlassian.net/browse/FNDD-7209))
* On a GCP VM, the install scripts will use the logged in Docker user from docker login, rather than the credentials passed in to the install scripts via -u and -p. ([FNDD-6625](https://hv-eng.atlassian.net/browse/FNDD-6625))

**Uninstall**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#uninstall)

* Uninstall can report success even if Kubernetes is not reachable, please check the output carefully to make sure it worked. The following output is an example of what it looks like when this scenario happens. ([FNDD-4608](https://hv-eng.atlassian.net/browse/FNDD-4608))

$ ./bin/uninstall-control-plane.sh

Mon Jul 13 12:08:18 UTC 2020 Start uninstalling the control plane

Error: Kubernetes cluster unreachable

Mon Jul 13 12:08:20 UTC 2020 Release, "hscp-hitachi-solutions" in namespace "hitachi-solutions" was not found

Mon Jul 13 12:08:29 UTC 2020 Deleted solution-control-plane-registry and solution-control-plane-regcred

Mon Jul 13 12:08:29 UTC 2020 Finished uninstalling the control plane

* When uninstalling the control plane and cluster services with uninstall-control-plane.sh -F, the script must be run from the same directory where the cluster services were originally installed from, or it will fail with the message "ERROR: Did not uninstall istio. Installer seems to be missing." ([FNDD-7135](https://hv-eng.atlassian.net/browse/FNDD-7135))

**Upgrade**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#upgrade)

* During the upgrade the Foundry keycloak user can sometimes lose its permission to access the control plane. If this happens simply patch the sso and keycloakuser objects with a harmless label as seen below.

sso\_output=$(kubectl get sso -o name -n [your control plane namespace])

for x in $sso\_output; do kubectl patch $x --type='merge' -p '{"metadata":{"labels":{"change-to-force-update":"ignored"}}}'; done;

user\_output=$(kubectl get keycloakuser -o name -n [your control plane namespace])

for x in $user\_output; do kubectl patch $x --type='merge' -p '{"metadata":{"labels":{"change-to-force-update":"ignored"}}}'; done;

* The control-plane upgrade may fail and leave the control-plane in a bad state, if an upgrade to 2.2.0 from 2.1.1 is attempted *without* upgrading cluster-services to 2.2.0 beforehand. This will result in a failed upgrade entry in the history, and a Warn state for the control-plane. It is possible to rollback from this state back to 2.1.1, then upgrade cluster-services to 2.2.0, before attempting to upgrade the control-plane to 2.2.0. ([FNDD-7108](https://hv-eng.atlassian.net/browse/FNDD-7108))

**Networking**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#networking)

* Creating multiple routes to SSO protected apps with rewrite path other than '/' does not work during authentication ([FNDD-4509](https://hv-eng.atlassian.net/browse/FNDD-4509))
* If an app has a fixed rewritePath other than '/', but does not support changing the path (rewritePathSupported=false) the UI shows an empty box instead of the fixed path for the Path of any Network Routes. ([FNDD-4450](https://hv-eng.atlassian.net/browse/FNDD-4450))
* Currently, only the default routes that come with the control plane (control plane, admin-ui, swagger-ui and kibana) have their permissions automatically added to the foundry user. This means that for new routes that are added or for the default routes if they are edited the foundry user will not automatically have permission to access those solutions. It is recommended to only use the foundry user to configure keycloak. A separate user should be created if a local user is needed to access solutions. ([FNDD-4515](https://hv-eng.atlassian.net/browse/FNDD-4515))
* It is possible to have to multiple App CRs with same name (spec:name) when deployed into same control plane. This impacts the Network Route UI by not allowing you to select an application ([FNDD-4559](https://hv-eng.atlassian.net/browse/FNDD-4559))
* Leaving out the trailing slash in a URL will result in a 404 page. For example, <https://example.com/hitachi-solutions/hscp-hitachi-solutions/solution-control-plane/> will resolve, but <https://example.com/hitachi-solutions/hscp-hitachi-solutions/solution-control-plane> will not resolve. ([FNDD-5731](https://hv-eng.atlassian.net/browse/FNDD-5731))
* Creating a network route that conflicts with an already existing one does not provide a detailed error message. ([FNDD-4815](https://hv-eng.atlassian.net/browse/FNDD-4815)) A network route defines a URL that can be used to access an application. This URL must be unique. Creation of new routes, or update of existing network routes cannot collide with an existing URL or the operation will fail with an error. The current error is generic and does not indicate which URL and network route resulted the conflict. (NOTE: conflict checking is done cluster-wide; a route may conflict with a route defined in another plane)
* It is possible to delete all of the Network Routes for an App by using the UI and / or APIs. If all routes have been deleted for your App you can restore the generated route by running:

kubectl edit my-app -n my-namespace

Set generateNetworkRoute: true and then save and Foundry will generate a new route. ([FNDD-4752](https://hv-eng.atlassian.net/browse/FNDD-4752))

* When creating a network route the public path must start and end with a slash(/). ([FNDD-4985](https://hv-eng.atlassian.net/browse/FNDD-4985))
* When your solution has a SSO enabled App CR, it is possible to create conflicting network routes having identical external host name and path. Following the conflicting route url may result in access to arbitarily selected internal application if different destination were specified. ([FNDD-4973](https://hv-eng.atlassian.net/browse/FNDD-4973))
* If an App CR is deleted and then quickly created Foundry may end up setting generateNetworkRoute: false on the CR. This will make it so the Network Route is not created for the solution. The work around is to either wait a minute between recreating the App CR or edit the App CR and set generateNetworkRoute: true. ([FNDD-6724](https://hv-eng.atlassian.net/browse/FNDD-6724))

**Logging**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#logging)

* Logs will not be collected by Foundry logging solution if you customized docker data-root. The default, supported location is /var/lib/docker. ([FNDD-3395](https://hv-eng.atlassian.net/browse/FNDD-3395))
* Intermittently, after the control plane is reconfigured with non-default logging outputs, logs messages may fail to reach those outputs ([FNDD-7078](https://hv-eng.atlassian.net/browse/FNDD-7078)). In this case, restart fluentd and fluentbit by running kubectl delete on all fluentd and fluentbit pods. For example:

kubectl get -A pods -o name | grep -E "fluent(d-0|bit)" | xargs kubectl delete -n hitachi-solutions

**Solution Management**[**#**](http://docs.foundry.wal.hds.com/docs/ReleaseNotes#solution-management)

* If an operation on a solution (install, upgrade, rollback, or delete) takes longer than 2 minutes, but less than the control plane setting for configuration.helm.defaultTimeoutSeconds, the request to the server will fail, but the operation will continue until the timeout. The message in the UI will be similar to "The server had a problem installing the solution. Please try again later." Consult the Installed Solutions page for the currently installed solutions, and the history tab for each specific installed solution for the ongoing status of operations. ([FNDD-7164](https://hv-eng.atlassian.net/browse/FNDD-7164))

**Prerequisites**

Before attempting control-plane deployment, administrators must be aware of the following prerequisites:

* A Kubernetes system (with a kubeconfig file for API server access)
  + The Kubernetes cluster must have a default storage class configured
* A secured, OCI-compatible container registry
* Proxy or load balancer (optional)

**Kubernetes Requirements**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites#kubernetes-requirements)

A secured Kubernetes system (v1.18+) is required.

A kubeconfig file for accessing the kube-api-server should be supplied to the Foundry installation scripts.

**Default Storage Class**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites#default-storage-class)

To maximize solution portability, Foundry requires the Kubernetes system to declare a "default" storage class for dynamic provision of storage resources.

To verify that your Kubernetes cluster declares a "default" storage class (or to enable one), follow [these instructions](https://kubernetes.io/docs/tasks/administer-cluster/change-default-storage-class/#changing-the-default-storageclass).

**Registry Requirements**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites#registry-requirements)

An OCI-compatible container image registry with an SSL certificate is required. Authentication credentials are also recommended.

When deploying both cluster services and the control plane, the *fully qualified domain name* of this registry must be specified, either using the **-r** argument or the [installer config file](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#config-file). The value you specify needs to include both the host and port for your registry. For example:

-r myregistry.example.com:6000

If your registry is available at port 443, you don't need to specify the port number.

If you are using a registry that supports multitenancy, you also need to include the specific location within the registry that you want to use. For example, if you are using [Harbor](https://goharbor.io/" \t "_blank), you need to include the name of the Harbor project you want to use:

-r myharbor.example.com:6000/my\_project

If the registry you are using is insecure (i.e. has a self-signed or otherwise not trusted SSL certificate) you must do the following:

* Configure your docker daemon on the installation node to allow the insecure registry
  + This is often done by adding the registry to the "insecure-registries" section of /etc/docker/daemon.json, and restarting the docker service.
* Configure the container runtime on the cluster to allow the insecure registry
* Specify the -I flag for install-control-plane.sh and for install-cluster-services.sh

For production use, we recommend using a trusted, CA-signed certificate.

For information on setting up a non production development only registry that meets requirements, see [Installing a Registry](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/InstallRegistry)

**Cluster Prerequisites**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites#cluster-prerequisites)

Control-plane deployment involves first installing a set of *cluster level prerequisites*, which includes services which have only a single deployment per cluster. These services may be installed to the Kubernetes cluster using Foundry provided installers (recommended), or you may wish to choose to use existing or alternate deployments of these services:

* Istio Service Mesh (V1.10+)
* cert-manager

For details on installing cluster prerequisite services on Kubernetes, see [Install Cluster Services](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices).

**Pod Security Policy**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites#pod-security-policy)

A [Pod Security Policy](https://kubernetes.io/docs/concepts/policy/pod-security-policy/) limits what capabilities a Pod is allowed to use on the Kubernetes cluster. It also provides restrictions on what volumes, groups and users can be used. The provided restricted PSP is the current most secure PSP that also works with Foundry. Updates to the PSP will be made as Foundry implements improvements to the security posture of the product.

apiVersion: policy/v1beta1

kind: PodSecurityPolicy

metadata:

name: restricted

spec:

allowPrivilegeEscalation: false

allowedCapabilities:

- NET\_ADMIN

- NET\_RAW

fsGroup:

ranges:

- max: 65535

min: 1

rule: MustRunAs

privileged: false

runAsUser:

rule: RunAsAny

seLinux:

rule: RunAsAny

supplementalGroups:

ranges:

- max: 65535

min: 1

rule: MustRunAs

volumes:

- configMap

- emptyDir

- projected

- secret

- downwardAPI

- persistentVolumeClaim

- hostPath

**Deploying Solution Management Control Plane(s)**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites#deploying-solution-management-control-planes)

Once cluster prerequisites have been successfully installed, administrators may then deploy one or more control-plane instances into dedicated namespaces.

For details on installing control-plane instances, see [Install Control Plane](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane).

**Proxy or Load Balancer (optional)**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites#proxy-or-load-balancer-optional)

By default, the control-plane exposes solution endpoints via an Istio Ingress Gateway using the configured public port of the Istio Ingress Gateway Controller. URL paths are also are generated that will function properly when multiple controls planes are deployed onto a Kubernetes cluster (e.g. "https://clusterhostname:30443/namespace/solution/service"). This allows for leveraging DNS to directly access services via the gateway. With DNS and Foundry "Network Routes", you may also expose alternate DNS-based access (e.g. "[https://service.solution.namespace:30443"](https://service.solution.namespace:30443%22/)).

To expose public endpoints via alternate hostnames, paths, and ports (e.g. [https://admin.foo.com](https://admin.foo.com/)), the use of a load balancer or proxy is recommended for use with Foundry "Network Routes".

**Installing Cluster Services**

**Overview**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#overview)

Foundry's Control plane expects that the Kubernetes cluster is running with, [Istio](https://istio.io/), [cert-manager](https://cert-manager.io/) and has a default StorageClass defined (see [The StorageClass Resource](https://kubernetes.io/docs/concepts/storage/storage-classes/#the-storageclass-resource))

**Prerequisites**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#prerequisites)

* A Kubernetes cluster, version 1.18 or later.
* An OCI-compatible Registry, such as a Docker v2 registry. See the [Registry Prerequisites](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites) section for more details.
* The installation process must be run from a node with:
  + network access to the Kubernetes cluster
  + a kubeconfig file that provides authentication to that cluster
  + a local docker daemon running. The cluster itself does not need to use docker as a container runtime, but it is needed on the installation node.

**Installing Cluster Services**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#installing-cluster-services-1)

The cluster services are included in the Solution Package produced by Forge:

1. Un-tar the tar file to a directory[1] of your choosing.
2. Execute the install-cluster-services.sh script inside the control-plane/bin/ directory in the extracted directory. See below for required command-line options.

./control-plane/bin/install-cluster-services.sh -r <registry\_fqdn>:<registry\_port>[/optional/registry/sub/path]

Copy

The install-cluster-services.sh script will only install those services that are missing from cluster.

[1]Foundry install directory has to be at least two directories deep in file system, we suggest to start with user home. e.g.

$ cd $HOME

$ mkdir control-plane

$ tar -C control-plane -zxf &lt;path-to-foundry-tar-file&gt;

**Optional Configuration parameters**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#optional-configuration-parameters)

**Kubeconfig[#](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices" \l "kubeconfig" \o "Direct link to heading)**

The install-cluster-service.sh script requires a kubeconfig file for the Kubernetes cluster you're using. By default, the script looks for that file in $HOME/.kube/config. To use a different kubeconfig file, pass the path to that file with the **-k** argument, e.g. -k /path/to/kube/config or in the config file below.

**Skip the installation of a specified cluster service**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#skip-the-installation-of-a-specified-cluster-service)

The installation of a cluster service may be skipped by using the **-s** argument. The value of the argument may be one of the following: istio or cert-manager. If you would like to skip multiple services then use the **-s** again, for example: -s cert-manager

**Configure deployment of istio's ingress gateway**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#configure-deployment-of-istios-ingress-gateway)

When deploying istio as part of cluster services you have the option of configuring the ingress gateway. You do this by using the **-w** argument for each configuration option. Each configuration option must be provided by a **key=value** pair. For example: -w service-type=NodePort -w http-port=31080. The following configuration options are supported:

* name A unique (Kubernetes) name for the ingress controller. The actual name during deployment will always be prefixed with *istio-*. The default name, if not specified is *ingressgateway*.
* service-type The service type to use for the ingress controller, which must be either *LoadBalancer* or *NodePort*.
* http-nodeport A pre-fixed value for the node port to use for http traffic. Default value is: *30080*
* https-nodeport A pre-fixed value for the node port to use for https traffic. Default value is *30443*
* tcp-nodeport A pre-fixed value for the node port to use for tcp traffic. Default value is: *30500*
* status-nodeport A pre-fixed value fo the node port to use for health monitoring of the ingress controller. Default is: *30020*

**Configuring Registry Authentication**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#configuring-registry-authentication)

One of the command line options to the installer is -r which indicates the registry to use when pushing and pulling images and charts. There are a few additional options that can be passed to the installer when the registry used requires authentication. The options and rules are as follows:

* There are few types of credentials that can be specified. Credentials to be used by the installer to push images and charts and that will not persisted. Credentials that are to be used post installation to pull images and charts from the registry and as such, will be stored by the installer. Lastly, anonymous access may be used for either install time or runtime.
* To pass credentials for pushing images, use the -u and -p command line options for *username* and *password* accordingly.
* In addition, to pass credentials for pulling images, use the -R for *username* and -P for password. These credentials will default to the installer credentials if omitted.
* If the registry supports anonymous access for pulling images and charts, you may pass the -A command line option to the installer, instead of -R and -P.
* The installer will reject using combinations that are contradictory. (For instance, using -A with -R at the same time. Or passing -u without a -p)
* As an example, the following install command uses different set of credentials for pushing and pulling images and charts:

$ ./install-cluster-services.sh -r foo.registry.io -u superman -p shhh123 -R spiderman -P spiderpass...

**Insecure Registry**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#insecure-registry)

Use flag -I if the provided registry is insecure (does not have a valid SSL certificate).

$ ./install-cluster-services.sh -r foo.registry.io -I -k kubeconfig

Copy

**Note:** In case of insecure registry with Auth is using *Self-Signed Certificate*/does not have a valid SSL certificate, use flag -I

$ ./install-cluster-services.sh -r foo.registry.io -u superman -p shhh123 -I -k kubeconfig

Copy

**Skip Loading Images**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#skip-loading-images)

If all the images are already pushed into the registry and user wants to skip pushing the images to registry again then script should be executed with -L command line option

**Platform**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#platform)

The script accepts **--platform** argument and sets the platform for which the installation is happening. Acceptable values for this argument are kubernetes and openshift. Default values is set as kubernetes.

**Additional Istio configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#additional-istio-configuration)

Additional parameters can be passed to istioctl using the **--iset** argument. All the parameters accepted by istioctl can be passed here

* As an example, the following install command uses will pass parameters to istioctl

$ ./install-cluster-services.sh --iset foo=bar --iset --foo --iset --foo-bar......

Copy

For the list of supported parameters by istioctl refer [link](https://istio.io/v1.10/docs/reference/config/istio.operator.v1alpha1/#IstioOperatorSpec%20(default%20%60%5B%5D%60))

**Config file**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#config-file)

Some configuration options may be specified in a configuration file rather than as command line arguments. The config file is a key-value file separated by "=" characters.

You can specify the path to a config file using the **-f** argument.

**Option precedence**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#option-precedence)

If an option is specified both in the config file and as a flag, the last argument passed will be used.

For example, in this case:

install-cluster-services.sh -f config-file -k kubeconfig

Copy

The kubeconfig from the command line flag will be used, whereas in this case:

install-cluster-services.sh -k kubeconfig -f config-file

Copy

The kubeconfig from the config file will be used.

**Supported options**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#supported-options)

The supported options are:

* ingressgatewayconfig ingress gateway configuration. Each configuration option is specified as a key=value pair. Use space to delimit options. For the list of options see the *-w* switch description.
* insecureregistry Whether the provided registry is insecure (does not have a valid SSL certificate)
* istioconfig Additional istio parameters. Use space to delimit options.
* kubeconfig The path to a Kubeconfig file. By default uses $HOME/.kube/config.
* registry The fully qualified domain name and port of the registry.
* skipimages Whether to skip loading images into the registry. Use with caution.
* skipservice Skips the installation of one or more cluster services. Each service to skip is separated by a space.
* username Username to use to authenticate to the registry.
* password Password to use to authenticate to the registry.
* runtimeusername An optional registry username to use at runtime. If neither this or *runtimeanonymous* is specified, the *username* value will be used at runtime
* runtimepassword An optional registry password to use at runtime.
* runtimeanonymous An optional flag to indicate that runtime registry access should be anonymous, I.E foundry should not use any username or password to authenticate with the registry for cluster services.

An example of a configuration file:

kubeconfig=/home/username/.kube/config registry=registry.host.name:5000 insecureregistry=true

Copy

**External Certificates**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#external-certificates)

The certificates used by Cert-Manager can be replaced with certificates from third party certificate authorities.

There are two possible ways to replace the certificates:

1. Replace the Foundry CA certificate with an external intermediate CA certificate.
2. Manually replace the certificate in use by the Istio gateway for each hostname being used.

**Replace Cert-Manager CA certificate**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#replace-cert-manager-ca-certificate)

After installing cluster services remove the CA certificate created for Cert-Manager.

kubectl delete secret ca-key-pair -n istio-system

Copy

Create a new ca-key-pair secret

kubectl create secret generic -n istio-system ca-key-pair --from-file=./tls.crt --from-file=./tls.key

Copy

* tls.crt is a PEM encoded X509 CA certificate
* tls.key is a PEM encoded RSA private key without a password

The control plane can be installed as described in [InstallControlPlane](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane) and Cert-Manager will use the provided CA certificate to create certificates that are needed.

**Manually replace certificate in Istio Gateway**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices#manually-replace-certificate-in-istio-gateway)

The certificates created by Cert-Manager are used by the Istio Gateway to secure connections created via the Istio Ingress Gateway. This gateway can be manually modified to use a certificate provided by a third party authority.

If you are using a CSR (Certificate Signing Request) to generate the certificate you would use the private key created as part of the CSR process as the tls.crt. The certificate created by the CA would be used as the tls.key and the provided Certificate Authority chain would be used as the ca.cert.

Create a secret with the CA certificate chain, the certificate, and the certificate private key.

kubectl create secret generic -n istio-system certificate-manual-<domain-name> --from-file=./ca.crt --from-file=./tls.crt --from-file=./tls.key

Copy

* ca.crt is a PEM encoded certificate chain of the CA certificates
* tls.crt is a PEM encoded X509 certificate
* tls.key is a PEM encoded RSA private key without a password

Edit the https-gateway-<domain-name> Istio gateway and replace the certificate with the manual one that was just created.

kubectl edit gw -n istio-system https-gateway-<domain-name>

Copy

The Istio Ingress Gateway will immediately start using the new certificate.

**Installing Custom Resource Definitions**

**Overview**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#overview)

From version 2.2.0 onwards, Foundry manages Custom Resource Definitions (CRDs) for the Solution Control Plane, Addons and Solutions in Helm charts, to facilitate re-use between multiple control planes.

Foundry requires that Custom Resource Definitions for the control-plane or solutions be installed before the control-plane or the solutions, respectively. To do so, use the apply-crds.sh script in the bin/ directory of the control-plane tar, to install the CRD charts located in the crd-charts/ directory and the images in the images/ directory.

apply-crds.sh can also be used to install and manage Solution CRDs that have opted into being managed by Foundry. See the [development documentation on CRDs](http://docs.foundry.wal.hds.com/docs/DevelopingSolutions/DevelopingCRDs) for more information.

apply-crds.sh will attempt to upload the CRD charts to the registry, and install them on to the cluster if the CRD charts do not exist, or upgrade them if the CRD charts are newer than the ones installed on the cluster.

If the CRD charts are already installed and the same version or newer, apply-crds.sh will NOT install the CRD chart.

**Pre-requisites**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#pre-requisites)

* A Kubernetes cluster, version 1.18 or later.
* An OCI-compatible Registry, such as a Docker v2 registry. See the [Registry Prerequisites](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites) section for more details.
* Installation of control-plane cluster prerequisites. See [Install Cluster Services](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices) for instructions
* The installation process must be run from a node with:
  + Network access to the Kubernetes cluster
  + A kubeconfig file that provides authentication to that cluster
  + A local docker daemon running. The cluster itself does not need to use docker as a container runtime, but it is needed on the installation node.
  + No outgoing proxy in use. The shell environment should not have a proxy configured.

**Options**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#options)

**Expected Folders**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#expected-folders)

When the apply-crds.sh script is located in a directory that matches the expected directory, the -e parameter will fill in the arguments for Charts and Images using the expected layout.

The expected layout is:

solution-package├── bin| └── apply-crds.sh├── crd-charts| ├── crdpackages.hitachivantara.com-2.2.0.tgz| └── ...└── images ├── foundry-admin-app-2.2.0.tar └── ...

Copy

**CRD Charts**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#crd-charts)

To specify charts for apply-crds.sh there are two parameters that can be used:

* -C to specify a directory of CRD charts to upload into the registry, and install onto the cluster
* -c to specify a single CRD chart in a tar archive to upload into the registry, and install onto the cluster

Each parameter is allowed to be repeated multiple times, to pass multiple charts or directories in.

**Example**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#example)

To install CRD charts and images in specific directories, or specific charts, use flags similar to upload-solutions.sh:

bin/apply-crds.sh -r <registry> -I <directory of images> -C ./charts -c ~/another-chart.tgz -C ./some-other-charts

Copy

**Images**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#images)

To specify charts for apply-crds.sh there are two parameters that can be used:

* -I to specify a directory of images to upload into the registry
* -i to specify a single CRD image in a tar archive to upload onto the registry

Each parameter is allowed to be repeated multiple times, to pass multiple images or directories in.

**Example**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#example-1)

To install CRD charts and images in specific directories, or specific charts, use flags similar to upload-solutions.sh:

bin/apply-crds.sh -r <registry> -I <directory of images> -C ./charts -c ~/another-chart.tgz -C ./some-other-charts

Copy

**Kubeconfig[#](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs" \l "kubeconfig" \o "Direct link to heading)**

The apply-crds.sh script requires a kubeconfig file for the Kubernetes cluster you're using. By default, the script looks for that file in $HOME/.kube/config. To use a different kubeconfig file, pass the path to that file with the **-k** argument, e.g. -k /path/to/kube/config or as part of the config file argument. [See below](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#config-file) for more information about the config file argument.

For more information and to get a complete reference to all options type ./apply-crds.sh -h.

**Examples**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#examples)

The following are some examples of common use cases of apply-crds.sh:

**Install all CRDs from an expected package layout**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#install-all-crds-from-an-expected-package-layout)

bin/apply-crds.sh -r <registry> -e

Copy

**Install to an insecure registry (for development)**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#install-to-an-insecure-registry-for-development)

bin/apply-crds.sh -r <registry> -e --insecure

Copy

**Note:** In case of insecure registry with Auth is using *Self-Signed Certificate*/does not have a valid SSL certificate, use flag --insecure

bin/apply-crds.sh -r <registry> -e -u <username> -p <password> --insecure

Copy

**Install all CRDs with a registry using Auth:**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#install-all-crds-with-a-registry-using-auth)

bin/apply-crds.sh -r <registry> -e -u <username> -p <password>

Copy

**Install CRD charts and images in specific directories, or specific charts**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#install-crd-charts-and-images-in-specific-directories-or-specific-charts)

Use flags are similar to use flags similar to upload-solutions.sh

bin/apply-crds.sh -r <registry> -I <directory of images> -i <specific image tar> -C <directory of charts> -c <specific chart tgz>

Copy

**Optional Configuration parameters**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#optional-configuration-parameters)

**Config file**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#config-file)

Some configuration options may be specified in a configuration file rather than as command line arguments. The config file is a key-value file separated by "=" characters.

You can specify the path to a config file using the **-f** argument.

**Option precedence**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#option-precedence)

If an option is specified both in the config file and as a flag, the last argument passed will be used.

For example, in this case:

apply-crds.sh -f config-file -k kubeconfig

Copy

The kubeconfig from the command line flag will be used, whereas in this case:

apply-crds.sh -k kubeconfig -f config-file

Copy

The kubeconfig from the config file will be used.

**Supported options**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs#supported-options)

The supported options are:

* debug Whether to include additional debug output for this script.
* chartdir Include a directory with chart files for upload. Can be used multiple times.
* chartfile Include a chart file for upload. Can be used multiple times.
* imagedir Include a directory with image files for upload. Can be used multiple times.
* imagefile Include an image file for upload. Can be used multiple times.
* insecureregistry Whether the provided registry is insecure (does not have a valid SSL certificate)
* kubeconfig The path to a Kubeconfig file. By default uses /home/rsmieja/.kube/config.
* useexpectedresources An optional flag to indicate if expected charts and images should be used.
* password Password to use to authenticate to the registry.
* registry The URL to an external CRI-compatible Registry. Required.
* runtimeanonymous An optional flag to indicate that runtime registry access should be anonymous, I.E. the control plane should not use any username or password to authenticate with the registry.
* runtimepassword An optional registry password to use at runtime.
* runtimeusername An optional registry username to use at runtime. If neither this or runtimeanonymous is specified, the username value will be used at runtime.
* username Username to use to authenticate to the registry.

An example of a configuration file:

kubeconfig=/home/username/.kube/configimagedir=/home/username/build/images/chartdir=/home/username/build/charts/insecureregistry=true

**Installing Control Plane**

**Prerequisites**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#prerequisites)

* A Kubernetes cluster, version 1.18 or later.
* An OCI-compatible Registry, such as a Docker v2 registry. See the [Registry Prerequisites](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Prerequisites) section for more details.
* Installation of control-plane cluster prerequisites. See [Install Cluster Services](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices) for instructions
* Installation of required Custom Resource Definitions. See [Install Custom Resource Definitions](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallCRDs) for instructions
* The installation process must be run from a node with:
  + network access to the Kubernetes cluster
  + a kubeconfig file that provides authentication to that cluster
  + a local docker daemon running. The cluster itself does not need to use docker as a container runtime, but it is needed on the installation node.
  + No outgoing proxy in use. The shell environment should not have a proxy configured.
* A Solution Package produced by the Forge tool containing the Admin Control Plane and any number of solutions.
  + If you would like to install directly from the control plane artifacts, see [Manual Installation](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/ManualInstallation/ManualInstallation) for more information.
* As part of the logging solution, an elasticsearch container is deployed. Elasticsearch requires vm.max\_map\_count kernel setting to be set to at least 262144 on the node where it is scheduled. Before installing Control Plane make sure this kernel setting is set as required by following any one of the following ways:
  + Set the vm.max\_map\_count kernel setting on all the worker nodes by executing the command listed in the [elastic document](https://www.elastic.co/guide/en/elasticsearch/reference/current/vm-max-map-count.html). This can be followed if you have less number of worker nodes in the kubernetes cluster, or there could be other reasons.
  + Set the vm.max\_map\_count kernel setting on one or multiple worker node(s) by executing the command listed in the [elastic document](https://www.elastic.co/guide/en/elasticsearch/reference/current/vm-max-map-count.html) and add labels to that nodes, Pass these labels as the NodeSelector under logging.elasticsearch.nodeSelector as described in [Control Plane Configuration Values](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-nodeselector). This can be followed if you have more number of worker nodes in the kubernetes cluster and thus selecting few of them, or there could be other reasons.
  + An init container of the elasticsearch pod can do the required kernel setting on the worker node where it is scheduled. But it is less secure as it will be running as root and uses privileged. To use this, enable the logging.elasticsearch.sysctlInitContainer.enabled setting as described in [Control Plane Configuration Values](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#enable-elasticsearch-sysctl-init-container).

**Basic installation**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#basic-installation)

To use the Solution Package produced by Forge:

1. Un-tar the tar file to a directory[1] of your choosing.
2. Execute the install-control-plane.sh script inside the control-plane/bin/ directory in the extracted directory. This will run the admin-app on the Kubernetes cluster in the background. See below for required command-line options.

./control-plane/bin/install-control-plane.sh -r <registry\_fqdn>:<registry\_port>[/optional/registry/sub/path]

1. Go to the admin UI at https://$HOSTNAME:30443/hitachi-solutions/hscp-hitachi-solutions/solution-control-plane/ where $HOSTNAME is the full hostname of one of the nodes in the cluster. This name must match a name produced by kubectl get nodes.

See [Authentication](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Authentication) for information about how to authenticate to the Foundry Control Plane.

[1]Foundry install directory has to be at least two directories deep in file system, we suggest to start with user home. e.g.

$ cd $HOME

$ mkdir control-plane

$ tar -C control-plane -zxf &lt;path-to-foundry-tar-file&gt;

**Optional Configuration parameters**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#optional-configuration-parameters)

**Cluster URL**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#cluster-url)

The **-c** argument allows the user to specify the cluster URL(e.g. https://cluster.example.com:30443) that can be used to address the nodes in the cluster. The cluster URL is used to generate redirect URLs for solutions in the cluster. For example the SSO feature uses this URL to redirect the user to the Keycloak login page. If a cluster URL isn't provided, one of the Kubernetes nodes will be chosen for the cluster URL.

**NOTE:** The host name in the cluster URL must be a fully qualified domain name.

-c cluster.url Specify a URL that will be used to address the entire cluster. This URL must route to one or more cluster nodes on the Istio https ingress port (typically 30443), from both outside the cluster and from pods within the cluster. The URL should NOT contain a path, the URL protocol can be omitted as https is the default. If the port is not specified *then* the default https port (443) will be used. If this option is omitted, the first node found *in* the cluster will be used. FOR CLOUD USAGE: a cluster URL must be provided as the nodes may not be FQDNs or otherwise directly accessible.

Copy

**ALSO NOTE:** If the cluster URL is too long this can cause problems using Keycloak/SSO see [Keycloak Limitations](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Authentication" \l "keycloak-limitations) for information on the Keycloak limitation.

**Kubeconfig[#](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane" \l "kubeconfig" \o "Direct link to heading)**

The install-control-plane.sh script requires a kubeconfig file for the Kubernetes cluster you're using. By default, the script looks for that file in $HOME/.kube/config. To use a different kubeconfig file, pass the path to that file with the **-k** argument, e.g. -k /path/to/kube/config or as part of the config file argument. [See below](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#config-file) for more information about the config file argument.

**Namespace**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#namespace)

The control plane deploys itself and all solutions to the same namespace. By default the namespace used is "hitachi-solutions". The **-n** argument can be used to specify a different namespace.

**Multiple Control Planes on the same cluster**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#multiple-control-planes-on-the-same-cluster)

Multiple deployments of the control plane are possible on the same kubernetes cluster. Each control plane must be deployed to a different namespace using the **-n** argument. Control planes installed this way will only have access to solutions that they have deployed. It is not required that multiple control planes share any other configuration such as a registry.

**Ingress Gateway**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#ingress-gateway)

The **-w** argument allows the user to specify the *name* of an *istio* ingress gateway controller. The name here identifies a controller that is deployed in the Kubernetes cluster. This controller is in the namespace that istio has been deployed to and has *deployment* and *service* resources with that name. In addition the *service*, *deployment* and *pod* of the ingress controller must be labeled using this name, through the label *app*. Like so:

apiVersion: v1kind: Servicemetadata: labels: app: <name> ... name: <name> namespace: <istio-namespace>

Copy

Using the *app* label is a requirement on the istio deployment when *name* is passed in as the value to *-w*.

By default the solution control plane uses the name *istio-ingresscontroller*, which is *istio*'s default name. In addition, please note the following section from the [values.yaml](https://helm.sh/docs/chart_template_guide/values_files/" \t "_blank) file of the solution control plane:

istio: enabled: true ingress: gateway: istio-ingressgateway tls: [] port: 30080 securePort: 30443 tcpPort: 30500 ns: istio-system

Copy

The *gateway* property should be treated as read-only and is not-customizable. The install script of the control plane (install-control-plane.sh) will pass prechecks for *istio* based on the -w command line option only and does not honor this *values.yaml* customization.

Note: if you want to change the ingress gateway for ingress network routes post deployment of the solution control plane you can edit the control plane's config map, like so, and then restart the admin application (pod) to pickup the changes:

$ kubectl edit cm hscp-hitachi-solutions-config -oyaml

apiVersion: v1data: ... ingress.gateway: ingressgateway ...kind: ConfigMapmetadata: ... name: hscp-hitachi-solutions-config namespace: hitachi-solutions ...

Copy

modify the *ingress.gateway* property to point the desired ingress controller. Any network routes already created within the admin application need to be deleted and recreated.

**Configuring Registry Authentication**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#configuring-registry-authentication)

One of the command line options to the installer is -r which indicates the registry to use when pushing and pulling images and charts. There are a few additional options that can be passed to the installer when the registry used requires authentication. The options and rules are as follows:

* There are few types of credentials that can be specified. Credentials to be used by the installer to push images and charts and that will not persisted. Credentials that are to be used post installation to pull images and charts from the registry and as such, will be stored by the installer. Lastly, anonymous access may be used for either install time or runtime.
* To pass credentials for pushing images, use the -u and -p command line options for *username* and *password* accordingly.
* In addition, to pass credentials for pulling images, use the -R for *username* and -P for password. These credentials will default to the installer credentials if omitted.
* If the registry supports anonymous access for pulling images and charts, you may pass the -A command line option to the installer, instead of -R and -P.
* The installer will reject using combinations that are contradictory. (For instance, using -A with -R at the same time. Or passing -u without a -p)
* As an example, the following install command uses different set of credentials for pushing and pulling images and charts:

$ ./install-control-plane.sh -r foo.registry.io -u superman -p shhh123 -R spiderman -P spiderpass...

Copy

**Insecure Registry**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#insecure-registry)

Use flag -I if the provided registry is insecure (does not have a valid SSL certificate).

$ ./install-control-plane.sh -r foo.registry.io -I -k kubeconfig

Copy

**Note:** In case of insecure registry with Auth is using *Self-Signed Certificate*/does not have a valid SSL certificate, use flag -I

$ ./install-control-plane.sh -r foo.registry.io -u superman -p shhh123 -I -k kubeconfig

Copy

**Skip Loading Images**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#skip-loading-images)

If all the images are already pushed into the registry and user wants to skip pushing the images to registry again then script should be executed with -L command line option

**Skip Cluster URL Check**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#skip-cluster-url-check)

If the script cannot detect the cluster url connection correctly, because the LoadBalancer does not pass the expected headers, and the user has validated that the url is correct, then validation for cluster url check can be skipped using --skip\_cluster\_url\_check command line option.

**Config file**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#config-file)

Some configuration options may be specified in a configuration file rather than as command line arguments. The config file is a key-value file separated by "=" characters.

You can specify the path to a config file using the **-f** argument.

**Option precedence**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#option-precedence)

If an option is specified both in the config file and as a flag, the last argument passed will be used.

For example, in this case:

install-control-plane.sh -f config-file -k kubeconfig

Copy

The kubeconfig from the command line flag will be used, whereas in this case:

install-control-plane.sh -k kubeconfig -f config-file

Copy

The kubeconfig from the config file will be used.

**Supported options**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#supported-options)

The supported options are:

* clusterurl Specify a URL(without a path) that will be used to address the entire cluster. By default uses https://<node hostname>:<istio.ingress.securePort>.
* debug Whether to include additional debug output for this script.
* insecureregistry Whether the provided registry is insecure (does not have a valid SSL certificate).
* kubeconfig The path to a Kubeconfig file. By default uses $HOME/.kube/config.
* namespace The namespace to deploy to. Default is hitachi-solutions.
* registry The fully qualified domain name and port of the registry.
* runtimeanonymous An optional flag to indicate that runtime registry access should be anonymous, I.E. the control plane should not use any username or password to authenticate with the registry.
* runtimepassword An optional registry password to use at runtime. Required if runtimeusername has been supplied.
* runtimeusername An optional registry username to use at runtime. If neither this nor runtimeanonymous is specified, the username value will be used at runtime. Required if runtimepassword has been supplied.
* username The username to authenticate to the registry.
* password The password to authenticate to the registry.
* skipimages Whether to skip loading images into the registry. Use with caution.
* ingressgateway The (istio's) ingress gateway name to use. By default uses istio-ingressgateway.
* tlsmode The default TLS mode for generated network routes.
* valuesfile Specify customized values file for solution-control-plane chart configuration.

An example of a configuration file:

kubeconfig=/home/username/.kube/configregistry=registry.host.name:5000insecureregistry=true

Copy

**Additional values**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#additional-values)

The solution control plane is deployed with a default configuration which can be modified by providing a set of overriding values in a configuration file. The file to use, also known as a *values* file (or [values.yaml](https://helm.sh/docs/chart_template_guide/values_files/" \t "_blank)) is passed in via the **-v** command line option, like so:

./install-control-plane.sh -v <path/to/values.yaml>

Copy

When **-v** is used a path to the values file is required. **Note:** not all configuration values can be overridden; such values, if specified in the values file, will be ignored.

The values that can be customized are specified in [Control Plane Configuration Values](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig).

**Debugging**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#debugging)

Debugging can be enabled by passing -D true to install-control-plane.sh. This enables additional output from the install-control-plane.sh script that can be useful for troubleshooting, and also enables DEBUG-level logging in the admin-app containers in the deployed solution-control-plane.

**Platform**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#platform)

The script accepts **--platform** argument and sets the platform for which the installation is happening. Acceptable values for this argument are kubernetes and openshift. Default values is set as kubernetes.

**Verify only flag**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#verify-only-flag)

If you would like to run the pre-install verification steps without actually installing the control plane you may pass the --verify\_cluster option.

**Disabling the Logging Solution**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#disabling-the-logging-solution)

By default the [Logging](http://docs.foundry.wal.hds.com/docs/DevelopingSolutions/Logging) framework is deployed as part of the control plane, enabling solutions to opt-in to log collection and other features around logging.

If you want to exclude the *Logging Solution* from being deployed as a sub component of the control plane, override logging.enabled to *false* as described in [Control Plane Configuration Values](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#logging-enabled).

**Highly available control plane**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#highly-available-control-plane)

The control plane can be deployed in a manner that supports high availability (HA). The following section describes various aspects and considerations of such a deployment.

**Pod replicas**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#pod-replicas)

A foundation concept in Kubernetes to ensure applications are serviceable and provide uninterrupted operation is having more than one instance of a pod running at any time. In addition to having the load balanced between the pod replicas, should any of these pod replica fail, the rest of the replicas running will ensure the application is still available.

The following components of the control plane support configuration of the pod replica count and must be considered when having HA in mind.

**Admin App**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#admin-app)

To support continuous administration of the control plane the administration app supports running multiple replicas simultaneously.

In addition to providing a highly available UI and REST API to clients, the admin app also monitors and reacts to changes in the various resources of the Kubernetes API. When multiple instances of the admin app are running, some functions of the app are shared among replicas. If one instance goes down, its duties will be reassigned to the remaining running replicas.

A replica count of 3 is recommended for the admin app. To change the replica count set the desired value for the replicaCount property in [Control Plane Configuration Values](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#admin-app-replica-count).

**Logging Solution**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#logging-solution)

If you opted-in to deploy (default) the logging solution provided with the Solution Control Plane, you may choose to configure it fo HA. The logging solution uses *fluentd* and *elasticsearch* to collect, filter and store logs. These components can be configured to run more than one replica for HA (3 replicas are recommended). Set the logging.fluentd.scaling.replicas and logging.elasticsearch.replicas accordingly. Having multiple replica's of *fluentd* will ensure streamlined log processing and filtering on a node/pod replica failure (otherwise a backlog will form in *fluentbit*, the component collecting logs on every cluster node). Multiple instances of *elasticsearch* ensure HA and better performance.

If you intend to scale down Elasticsearch after install then special consideration needs to be given. Please consult the following Elasticsearch documentation on:

* [Add and Removing Elastic Nodes](https://www.elastic.co/guide/en/elasticsearch/reference/current/add-elasticsearch-nodes.html)
* [Quorum Based Decision Making](https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery-quorums.html#modules-discovery-quorums)

**Keycloak[#](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane" \l "keycloak" \o "Direct link to heading)**

An instance count of 3 is recommended for keycloak in a HA configuration; to change the desired number of instances set the value for configuration.keycloak.instances property in [Control Plane Configuration Values](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#keycloak-instances).

**Gatekeeper**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#gatekeeper)

A replica count of 3 is recommended for your solution's gatekeeper HA configuration. To set the replica count for gatekeeper in your solution refer to the AuthSSO documentation here: [Adding authentication/authorization/single sign on to your solutions's helm chart](http://docs.foundry.wal.hds.com/docs/DevelopingSolutions/AuthSSO#adding-authenticationauthorizationsingle-sign-on-to-your-solutionss-helm-chart) Specifically, right after the example where we detail all of the fields we expect. Control plane should also be configured with this replica count either by using the configuration.controlPlane.gatekeeperReplicas property or the configuration.gatekeeper.defaultGatekeeperReplicas property.

**Using Persistent Volumes**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#using-persistent-volumes)

Persistent volumes protect against loss of data, and support fault tolerance and HA. The control plane supports configuring persistent volume claims for its relevant components. In general, the configuration options are specified by [Kubernetes](https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistentvolumeclaims) in it's [API Reference](https://v1-16.docs.kubernetes.io/docs/reference/generated/kubernetes-api/v1.16/#persistentvolumeclaimspec-v1-core).

**Logging Solution**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#logging-solution-1)

*elasticsearch* used by the logging solution allows to configure a Persistent Volume Claim (PVC) for each of its replicas via the logging.elasticsearch.volumeClaimTemplate property as specified in [Control Plane Configuration Values](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-storageclass). Note: this configuration must be applied during the control plane installation and is not available post install.

The following is recommended for PVC configuration:

* accessModes Use the value of ReadWriteOnce (default) unless the backing storage supports and/or requires ReadWriteMany. The former is expected to work fine for elasticsearch on node failure, as each replica uses its own volume (mounted only once for read/write).
* storageClassName The system wide default storage class will be used and should work out of the box. This field should be configured if a customization is required and known during the control plane installation. A storage class backed by network/clustered storage is required for elasticsearch HA to work.

**Known HA Related Limitations**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#known-ha-related-limitations)

* If *istio* is configured with a single instance of *istiod, istio-ingressgateway, istio-egressgateway*, which is the default for *istio* deployed by Foundry as part of its [Cluster Services](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallClusterServices), the pod disruption budget will prevent draining the node that these pods are running on. These will fail over if the node is killed in some other way.
* If *istio* is configured with a single instance of *istio-ingressgateway*, and the node on which this instance is running fails, inbound traffic to the cluster is blocked
* If *Kubernetes* was deployed with a single master node, and that node fails, or otherwise if *etcd or the api-server* fail, all pods that communicate to the API server fail (e.g. operators etc). Effectively Kubernetes is broken.

**Control plane runtime environment**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#control-plane-runtime-environment)

**Users**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/InstallControlPlane#users)

Most containers in Foundry currently run as the "root" user. Other containers use different userids that are defined in their images. These processes will often appear with a raw number for the userid in the process table on the host.

Containers that run as non-root users:

* The admin-app container uses userid 10001, which is defined as the "fnd" user within the container
* Some istio components use userid 1337, others run as root
* fluentd uses userid 100
* keycloak uses userid 1001

**Control Plane Configuration Values**

**Availability**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#availability)

The following configuration affects how many replicas of each piece of software are run. With only one replica, if that software fails for whatever reason, the software will be completely offline until a new instance is started. With multiple replicas, if one instance fails, failover to another instance will be quick, generally within seconds.

**Admin App Replica Count**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#admin-app-replica-count)

***replicaCount***

The number of replicas to use for the Foundry administration application. If deploying on a multi-node cluster then a count of 3 is recommended in order to achieve greater resiliency if a failure should occur.

**Default**: 1

Example:

replicaCount: 1

Copy

**Gatekeeper Replicas**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#gatekeeper-replicas)

***configuration.controlPlane.gatekeeperReplicas***

An integer value that will configure the number of gatekeeper pods the control plane will have for Control Plane, Swagger UI, Kibana. For HA configurations the recommended value is 3.

**Default**: Not set, will default to configuration.gatekeeper.defaultGatekeeperReplicas

Example:

configuration: controlPlane: gatekeeperReplicas: 1

Copy

**Default Gatekeeper Replicas**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#default-gatekeeper-replicas)

***configuration.gatekeeper.defaultGatekeeperReplicas***

An integer value that will configure the number of gatekeeper pods all SSOs in the cluster will have (Can be overridden by setting the SSO replicas field). For HA configurations, setting this to 3 will be easier than managing each solution independently.

**Default**: 1

Example:

configuration: gakeeeper: defaultGatekeeperReplicas: 1

Copy

**Keycloak instances**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#keycloak-instances)

***configuration.keycloak.instances***

The number of keycloak instances you want to have running. The recommended number of instances for an HA deployment is 3.

**Default**: 1

Example:

configuration: keycloak: instances: 1

Copy

**Elasticsearch Replicas**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-replicas)

***logging.elasticsearch.replicas***

The number of replicas to use for Elasticsearch. If deploying on a multi-node cluster then a count of 3 is recommended in order to achieve greater resiliency if a failure should occur. Please note that If you intend to scale down Elasticsearch after install then special consideration needs to be given. Please consult Elasticsearch's documentation. See: [Add and Removing Elastic Nodes](https://www.elastic.co/guide/en/elasticsearch/reference/current/add-elasticsearch-nodes.html) and [Quorum Based Decision Making](https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery-quorums.html#modules-discovery-quorums).

**Default**: 1

Example:

logging: elasticsearch: replicas: 1

Copy

**Fluentd replicas**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#fluentd-replicas)

***logging.fluentd.scaling.replicas***

The number of replicas to use for Fluentd. If deploying on a multi-node cluster then a count of 3 is recommended in order to achieve greater resiliency if a failure should occur.

**Default**: 1

Example:

logging: fluentd: scaling: replicas: 1

Copy

**Kibana Replicas**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#kibana-replicas)

***logging.kibana.replicas***

The number of replicas to use for Kibana. If deploying on a multi-node cluster then a count of 3 is recommended in order to achieve greater resiliency if a failure should occur.

**Default**: 1

Example:

logging: kibana: replicas: 1

Copy

**Storage configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#storage-configuration)

These settings affect the storage used by various components of the Solution Control Plane that have on-disk state.

**Elasticsearch storageClass[#](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig" \l "elasticsearch-storageclass" \o "Direct link to heading)**

***logging.elasticsearch.volumeClaimTemplate.storageClassName***

Name of the storage class to use for each Elasticsearch pod. Should generally be set at installation time. Not defined by default, which means it will use the cluster's default storage class. Note that changing this value only changes the defaults for new pods, it will not affect existing pods.

**Default**: N/A

Example:

logging: elasticsearch: volumeClaimTemplate: storageClassName: myCustomStorageClass

Copy

**Elasticsearch volume accessModes[#](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig" \l "elasticsearch-volume-accessmodes" \o "Direct link to heading)**

***logging.elasticsearch.volumeClaimTemplate.accessModes***

AccessModes for the PersistentVolumeClaim for each Elasticsearch pod. Should generally be set only at deployment time. Note that changing this value only changes the defaults for new pods, it will not affect existing pods.

See [Kubernetes Persistent Volume documentation](https://kubernetes.io/docs/concepts/storage/persistent-volumes/) for more information about accessModes.

**Default**: - ReadWriteOnce

Example:

logging: elasticsearch: volumeClaimTemplate: accessModes: - ReadWriteOnce

Copy

**ElasticSearch storage size**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-storage-size)

***logging.elasticsearch.volumeClaimTemplate.resources.requests.storage***

Initial size requested for log storage for each pod. Should generally be set at installation time. Note that changing this value only changes the defaults for new pods, it will not affect existing pods.

**Default**: 30Gi

Example:

logging: elasticsearch: volumeClaimTemplate: resources: requests: storage: 30Gi

Copy

**Keycloak storage class name**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#keycloak-storage-class-name)

***keycloakoperator.storageClassName***

The name of the Storage Class that you want the Keycloak Postgresql instance to use.

**Default**: Not set, uses cluster default Storage Class

Example:

keycloakoperator: storageClassName: myCustomStorageClass

Copy

**Keycloak volume access modes**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#keycloak-volume-access-modes)

***configuration.keycloak.accessModes***

AccessModes for the PersistentVolumeClaim for the Keycloak Postgresql pod. Should generally be set only at deployment time. Note that changing this value only changes the defaults for new pods, it will not affect existing pods.

See [Kubernetes Persistent Volume documentation](https://kubernetes.io/docs/concepts/storage/persistent-volumes/) for more information about accessModes.

**Default**: - ReadWriteOnce

Example:

configuration: keycloak: accessModes: - ReadWriteOnce

Copy

**Logging configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#logging-configuration)

These configuration settings affect the behavior of the logging solution, including Elasticsearch, kibana, fluentbit and fluentd.

**Logging Enabled**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#logging-enabled)

***logging.enabled***

Enables the logging solution. When disabled logging solution will be excluded and will not be deployed with the control plane.  
Any solutions that include the LoggingSolution CR will work but its logs will not be aggregated by Foundry. The *logging.enabled* property can only be set during install and cannot be changed at runtime.

**Default**: true

Example:

logging: enabled: true

Copy

**Enable Elasticsearch Sysctl Init container**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#enable-elasticsearch-sysctl-init-container)

***logging.elasticsearch.sysctlInitContainer.enabled***

This setting is responsible for launching the initContainer that runs as root inside the elasticsearch pod to set the vm.max\_map\_count kernal setting on the worker node where it is scheduled. This is the less secure way as the container runs as root and uses privileged.

**Default**: false

Example:

logging: elasticsearch: sysctlInitContainer: enabled: false

Copy

**Elasticsearch nodeSelector[#](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig" \l "elasticsearch-nodeselector" \o "Direct link to heading)**

***logging.elasticsearch.nodeSelector***

Add a set of labels of the worker node(s) onto which vm.max\_map\_count kernal setting is set to atleast 262144. So that elasticsearch pod will be scheduled on that node.

This is optional configuration, This is not required if some other mechanisms listed in the prerequisites section is used to set the vm.max\_map\_count kernel setting to atleast 262144.

**Default**: N/A

Example: The node(s) onto which the vm.max\_map\_count kernel setting is set to atleast 262144 has the labels deploy-es: yes and vm: 262144 applied to it. Then, following section needs to be added in values.yaml file to enable the elasticsearch nodeSelector.

logging: elasticsearch: nodeSelector: deploy-es: yes vm: 262144

Copy

**Elasticsearch java options**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-java-options)

***logging.elasticsearch.esJavaOpts***

These options are passed to the Elasticsearch jvm, typically to customize the java heap footprint.

Ensure that the memory limit below is sufficient to handle the requested heap size.

**Default**: -Xmx2g -Xms2g

Example:

logging: elasticsearch: esJavaOpts: -Xmx2g -Xms2g

Copy

**Elasticsearch cpu request**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-cpu-request)

***logging.elasticsearch.resources.requests.cpu***

The requested CPU resources for the Elasticsearch pods.

See [Managing Resources in Containers](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/) for more information.

**Default**: "100m"

Example:

logging: elasticsearch: resources: requests: cpu: 100m

Copy

**Elasticsearch memory request**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-memory-request)

***logging.elasticsearch.resources.requests.memory***

The requested memory resources for the Elasticsearch pods.

See [Managing Resources in Containers](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/) for more information.

**Default**: "4Gi"

Example:

logging: elasticsearch: resources: requests: memory: 4Gi

Copy

**Elasticsearch cpu limit**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-cpu-limit)

***logging.elasticsearch.resources.limits.cpu***

The CPU resource limit for the Elasticsearch pods.

See [Managing Resources in Containers](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/) for more information.

**Default**: "1000m"

Example:

logging: elasticsearch: resources: limits: cpu: 1000m

Copy

**Elasticsearch memory limit**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elasticsearch-memory-limit)

***logging.elasticsearch.resources.limits.memory***

The memory resource limit for the Elasticsearch pods.

See [Managing Resources in Containers](https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/) for more information.

**Default**: "100m"

Example:

logging: elasticsearch: resources: limits: memory: 4Gi

Copy

**Logging outputs**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#logging-outputs)

***global.loggingOutputs***

Customize the logging outputs.

See [Customize Log Outputs](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Logging#configuring-logging-outputs) for more information.

**Default**: See example below.

Example:

global: loggingOutputs: - name: default-output spec: |- elasticsearch: host: elasticsearch-master.{{ .Release.Namespace }}.svc.cluster.local port: 9200 scheme: http ssl\_verify: false ssl\_version: TLSv1\_2 buffer: timekey: 1m timekey\_wait: 30s timekey\_use\_utc: true logstash\_format: true include\_tag\_key: true logstash\_prefix: ${tag}

Copy

**Override kibana rewrite path Enabled**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#override-kibana-rewrite-path-enabled)

***logging.kibana.rewrite.overrideEnabled***

When true the default network path for Kibana will not be created, instead a network path can be created using the control plane and providing the logging.kibana.rewrite.overridePath value.

**Default**: false

Example:

logging: kibana: rewrite: overrideEnabled: false

Copy

**Override kibana rewrite path**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#override-kibana-rewrite-path)

***logging.kibana.rewrite.overridePath***

The network path for Kibana must be unique in the cluster. Once set this path can then be used to create a network path using the control plane and providing as the path the value set here.

Note that this path must start with a '/' character, but must not end with one.

**Default**: /kibana

Example:

logging: kibana: rewrite: overridePath: /kibana

Copy

**Keycloak configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#keycloak-configuration)

These configuration settings affect functionality of Keycloak.

**Enable Docker Authentication**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#enable-docker-authentication)

***configuration.keycloak.enableDockerAuthentication***

A boolean value that allows you to enable a feature described here [Docker Authentication with Keycloak](https://developers.redhat.com/blog/2017/10/31/docker-authentication-keycloak/).

**Default**: false

Example:

configuration: keycloak: enableDockerAuthentication: false

Copy

**Keycloak public path**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#keycloak-public-path)

***keycloakoperator.publicPath***

The public path used to create the Network Route that allows access to the Keycloak instance installed with the control plane. This path must be unique in the cluster.

**Default**: Not set, uses default auto generated network route

Example:

keycloakoperator: publicPath: /custom/public/path/

Copy

**Reject Request URI**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#reject-request-uri)

***configuration.keycloak.rejectRequestUri***

This setting controls a security feature that prevents CVE-2020-10770 by blocking the "request\_uri" parameter in OpenID Connect Authentication Requests. If you need to use request\_uri, and are aware of the security risks, this mechanism can be disabled by setting this field to false.

**Default**: true

Example:

configuration: keycloak: rejectRequestUri: true

Copy

**Other values**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#other-values)

Other values should not be changed, unless directed to by support. These properties generally exist to configure parts of the system to the supported state, to pass through install-time configuration to the running software, or to adjust values for debugging.

Most of these values have never been tested in anything except their default state. Other parts of the system have hard-coded dependencies on most of these values being in their default state.

**Helm operation timeout**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#helm-operation-timeout)

***configuration.helm***

This passes through certain configuration to control the timeouts for helm operations. defaultTimeoutSeconds applies to both "read" and "write" operations. readTimeoutSeconds and writeTimeoutSeconds override defaultTimeoutSeconds for read and write operations, respectively.

Due to an open bug, only one key=value pair may be specified at a time.

**Default**: defaultTimeoutSeconds=600

Example:

configuration: helm: defaultTimeoutSeconds=600

Copy

**Gatekeeper timeout**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#gatekeeper-timeout)

***configuration.gatekeeper.defaultTimeoutSeconds***

Example:

configuration: gatekeeper: defaultTimeoutSeconds: 120

Copy

**Registry configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#registry-configuration)

The values in the ***registry*** block pass command-line configuration along to the control plane runtime configuration. This is not the only place these registry values are stored, and changing them is not supported.

Example:

registry: name: registry.example.com insecure: false certificate: ""

Copy

**Image related settings**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#image-related-settings)

These settings control the image "repository" and "tag" for the images used by pods in the Solution Control Plane. The registry name will be included in these settings automatically.

These settings should not generally be changed by users. If they are changed, they will not be updated automatically when upgrading to new versions of the Solution Control Plane.

The registry is replaced with "registry.example.com" as an example.

image: repository: registry.example.com/foundry-admin-app pullPolicy: IfNotPresent

keycloakoperator: image: repository: registry.example.com/keycloak/keycloak-operator postgresImage: registry.example.com/rhscl/postgresql-10-rhel7:1 initImage: registry.example.com/keycloak/keycloak-init-container:master keycloakImage: registry.example.com/hitachi-keycloak:12.0.3-hv2 hookImage: registry.example.com/foundry-admin-app:2.3.0

ssoOperator: image: registry.example.com/sso-operator tag: 2.2.0.31 gatekeeperRepository: registry.example.com/keycloak/keycloak-gatekeeper gatekeeperTag: 9.0.2-hv1

logging: fluentbit: image: repository: registry.example.com/fluent/fluent-bit fluentd: image: repository: registry.example.com/banzaicloud/fluentd configReloaderImage: repository: registry.example.com/jimmidyson/configmap-reload volumeModImage: repository: registry.example.com/busybox logging-operator: image: repository: registry.example.com/banzaicloud/logging-operator elasticsearch: image: registry.example.com/elasticsearch/elasticsearch-oss imageTag: 7.8.0 kibana: image: registry.example.com/kibana/kibana-oss keycloak: gatekeeperRepository: registry.example.com/keycloak/keycloak-gatekeeper loggingSolutionOperator: image: repository: registry.example.com/logging-solution-operator operatorRepository: registry.example.com/untergeek/curator

Copy

**Runtime environment**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#runtime-environment)

The values in the ***env*** block pass command-line configuration along to the control plane runtime configuration. These are captured from installation options, and should not be modified.

Example:

env: hostname: cluster.example.com clusterurl: https://cluster.example.com debug: "true" tlsMode: simple

Copy

**Istio configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#istio-configuration)

The values in the ***istio*** block pass command-line configuration along to the control plane runtime configuration. These are captured from installation options, and should not be modified.

Example:

istio: enabled: true ingress: gateway: istio-ingressgateway tls: [] port: 30080 securePort: 30443 tcpPort: 30500 ns: istio-system

Copy

**Keycloak enabled**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#keycloak-enabled)

This setting enables the Keycloak operator, which is used to deploy keycloak. Modifying this value is not supported.

Example:

keycloakoperator: enabled: true

Copy

**elastic internal configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#elastic-internal-configuration)

The following values in ***logging.elasticsearch*** are internal state of the elasticsearch service and should not be modified.

Example:

logging: elasticsearch: persistence: enabled: true service: transportPortName: tcp-hscp-hitachi-solutions-transport

Copy

**Example deployed configuration**[**#**](http://docs.foundry.wal.hds.com/docs/AdministeringSolutions/Installation/ControlPlaneConfig#example-deployed-configuration)

Here is an example of the configuration of a deployed cluster with defaults. Note that not all values documented on this page are present, because by default they do not exist.

The registry is replaced with "registry.example.com" and the cluster hostname is replaced with "cluster.example.com". These values will be different in each deployment.

replicaCount: 1

image:

repository: registry.example.com/foundry-admin-app

pullPolicy: IfNotPresent

registry:

name: registry.example.com

insecure: false

certificate: ""

env:

hostname: cluster.example.com

clusterurl: https://cluster.example.com

debug: "true"

tlsMode: simple

configuration:

helm: defaultTimeoutSeconds=600

controlPlane:

gatekeeperReplicas: ""

gatekeeper:

defaultTimeoutSeconds: 120

defaultGatekeeperReplicas: 1

keycloak:

accessModes:

- ReadWriteOnce

instances: 1

enableDockerAuthentication: false

rejectRequestUri: false

istio:

enabled: true

ingress:

gateway: istio-ingressgateway

tls: []

port: 30080

securePort: 30443

tcpPort: 30500

ns: istio-system

keycloakoperator:

image:

repository: registry.example.com/keycloak/keycloak-operator

enabled: true

postgresImage: registry.example.com/rhscl/postgresql-10-rhel7:1

initImage: registry.example.com/keycloak/keycloak-init-container:master

keycloakImage: registry.example.com/hitachi-keycloak:12.0.3-hv2

hookImage: registry.example.com/foundry-admin-app:2.3.0

ssoOperator:

image: registry.example.com/sso-operator

tag: 2.2.0.31

gatekeeperRepository: registry.example.com/keycloak/keycloak-gatekeeper

gatekeeperTag: 9.0.2-hv1

logging:

enabled: true

fluentbit:

image:

repository: registry.example.com/fluent/fluent-bit

fluentd:

image:

repository: registry.example.com/banzaicloud/fluentd

configReloaderImage:

repository: registry.example.com/jimmidyson/configmap-reload

scaling:

replicas: 1

volumeModImage:

repository: registry.example.com/busybox

logging-operator:

image:

repository: registry.example.com/banzaicloud/logging-operator

elasticsearch:

esJavaOpts: -Xmx2g -Xms2g

resources:

requests:

cpu: 100m

memory: 4Gi

limits:

cpu: 1000m

memory: 4Gi

image: registry.example.com/elasticsearch/elasticsearch-oss

imageTag: 7.8.0

persistence:

enabled: true

replicas: 1

volumeClaimTemplate:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 30Gi

service:

transportPortName: tcp-hscp-hitachi-solutions-transport

kibana:

image: registry.example.com/kibana/kibana-oss

keycloak:

gatekeeperEnabled: true

gatekeeperRepository: registry.example.com/keycloak/keycloak-gatekeeper

replicas: 1

rewrite:

overrideEnabled: false

overridePath: /kibana

loggingSolutionOperator:

image:

repository: registry.example.com/logging-solution-operator

operatorRepository: registry.example.com/untergeek/curator

global:

loggingOutputs:

- name: default-output

spec: |-

elasticsearch:

host: elasticsearch-master.{{ .Release.Namespace }}.svc.cluster.local

port: 9200

scheme: http

ssl\_verify: false

ssl\_version: TLSv1\_2

buffer:

timekey: 1m

timekey\_wait: 30s

timekey\_use\_utc: true

logstash\_format: true

include\_tag\_key: true

logstash\_prefix: ${tag}