KPLABS Course

Docker Certified Associate 2020

Installation and Configuration of Docker EE

ISSUED BY

Zeal Vora

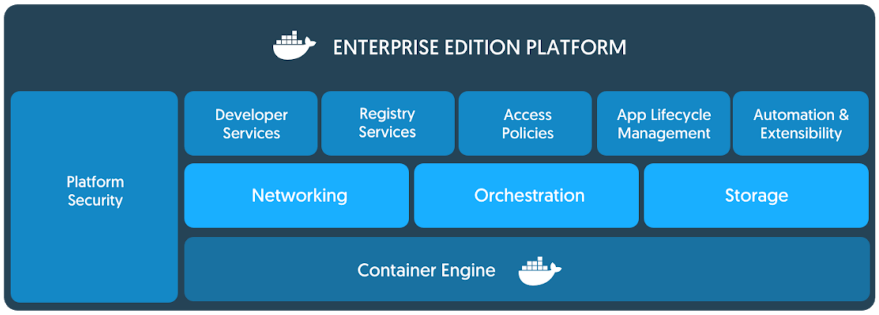
REPRESENTATIVE

instructors@kplabs.in

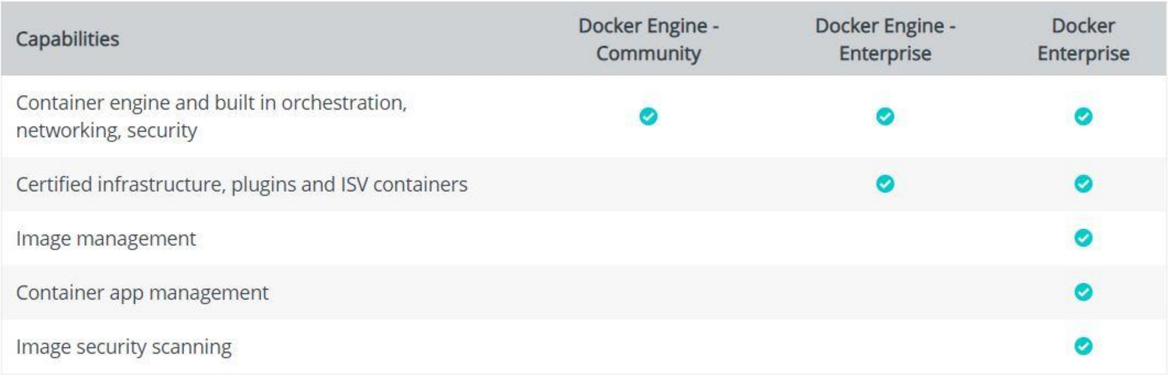
Module 1: Overview of Docker Enterprise Edition

Docker Enterprise Edition (also referred to as Docker EE) is designed for enterprise development as well as IT teams who build, ship, and run business-critical applications in production and at scale.

Docker EE is officially supported by the Docker Team.



There are multiple tiers available for the Docker Enterprise Edition. Referred to as Basic, Standard, and Advanced.

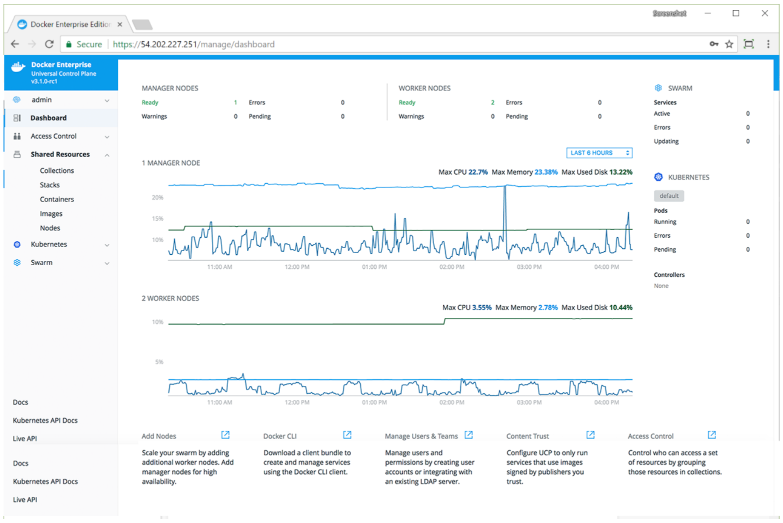


Capabilities	Docker Engine - Community	Docker Engine - Enterprise	Docker Enterprise
Container engine and built in orchestration, networking, security	✓	✓	✓
Certified infrastructure, plugins and ISV containers		✓	✓
Image management			✓
Container app management			✓
Image security scanning			✓

Module 2: Universal Control Plane

Docker Universal Control Plane (UCP) is the enterprise-grade cluster management solution from Docker.

UCP helps you manage your Docker cluster and applications through a single interface.



https://54.202.227.251/manage/dashboard

MANAGER NODES

Ready 1 Errors 0
Warnings 0 Pending 0

WORKER NODES

Ready 2 Errors 0
Warnings 0 Pending 0

SWARM

Services
Active 0
Errors 0
Updating 0

1 MANAGER NODE

Max CPU 22.7% Max Memory 23.38% Max Used Disk 13.22%

KUBERNETES

Pods
Running 0
Errors 0
Pending 0

Controllers
None

2 WORKER NODES

Max CPU 3.55% Max Memory 2.78% Max Used Disk 10.44%

Add Nodes

Scale your swarm by adding additional worker nodes. Add manager nodes for high availability.

Docker CLI

Download a client bundle to create and manage services using the Docker CLI client.

Manage Users & Teams

Manage users and permissions by creating user accounts or integrating with an existing LDAP server.

Content Trust

Configure UCP to only run services that use images signed by publishers you trust.

Access Control

Control who can access a set of resources by grouping those resources in collections.

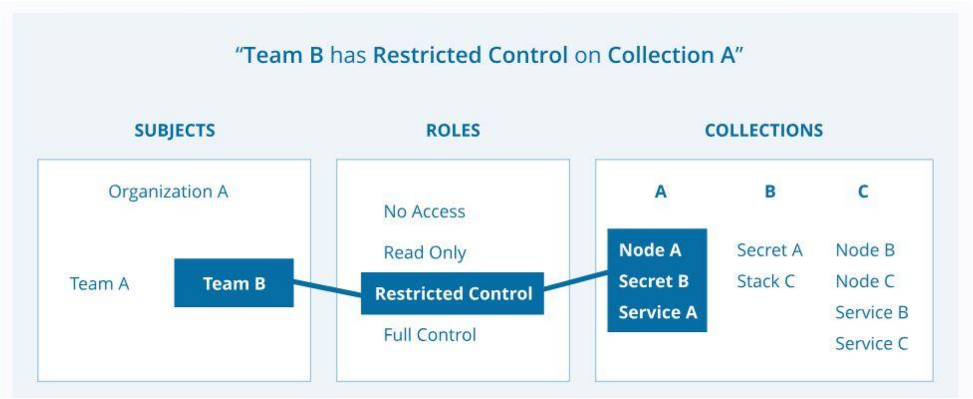
Following are the minimum system requirements for UCP:

- Docker EE Engine
- 8GB of RAM for Manager Nodes
- 4GB of RAM for Worker nodes
- 5GB of free disk space for the /var partition for manager nodes
- 500MB of free disk space for the /var partition for worker nodes

.

Module 3: UCP - Access Control

Docker UCP allows us provides granular access control features that allow us to define various aspects like who can create and edit container resources in your swarm and others.



3.1 Understanding Subject:

A subject represents a user, team, or organization.
A subject is in-turn granted an appropriate role.

User: Individual User.

Organization: A group of users that share a specific set of permissions, defined by the roles of the organization.

Team: A group of users that share a set of permissions defined in the team itself.
A team exists only as part of an organization

3.2 Understanding Roles

A role is a set of permitted API operations that you can assign to a specific subject and collection

3.3 Understanding Collections

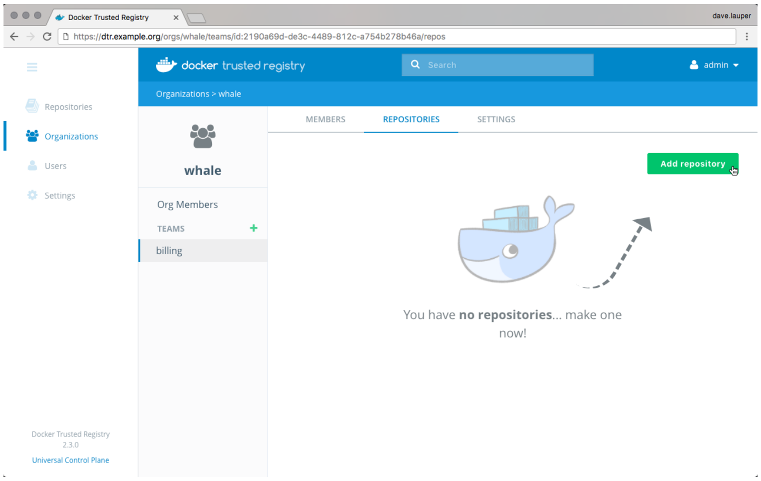
Docker EE enables controlling access to swarm resources by using collections.

Swarm resources that can be placed in a collection include:

- Physical or virtual nodes
- Containers
- Services
- Networks
- Volumes
- Secrets
- Application configs

Module 4: Overview of Docker Trusted Registry

Docker Trusted Registry (DTR) is the enterprise-grade image storage solution from Docker.



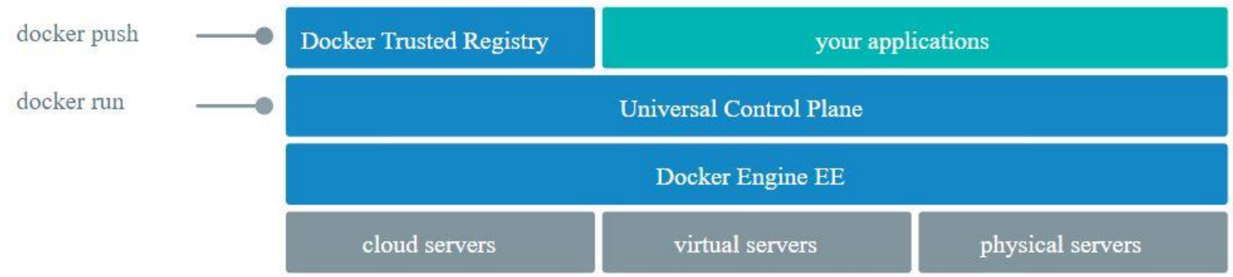
You have **no repositories**... make one now!

DTR provides many features that help in overall enterprise-grade container management.

Some of these features include:

- Security Scanning
- Image Signing
- Built-in access control
- Caching Images

Docker Trusted Registry (DTR) is a containerized application that runs on a Docker Universal Control Plane cluster.



The following describes the system requirement for DTR

Minimum Requirements for DTR:

- 16GB of RAM for nodes running DTR
- 2 vCPUs for nodes running DTR
- 10GB of free disk space

To install DTR, all nodes must:

- Be a worker node managed by UCP (Universal Control Plane)
- Have a fixed hostname.

Module 5: Overview of DTR Backup

Docker DTR has a backup command that allows us to take the backup.

The backup command doesn’t cause downtime for DTR, so you can take frequent backups without affecting your users.

```
 docker run --log-driver none -i --rm docker/dtr backup --ucp-url https://172.31.40.237
-ucp-insecure-tls --ucp-username admin --ucp-password YOUR-PASSWORD-HERE >
backup.tar
```

Important Pointers:

This command only creates backups of configurations and image metadata.

It does not back up users and organizations. Users and organizations can be backed up during a UCP backup.

It also does not back up Docker images stored in your registry.

Module 6: Backup DTR Images

DTR supports multiple backends to store the images.

Depending on the backend that is being used, the backup procedure for images differs.

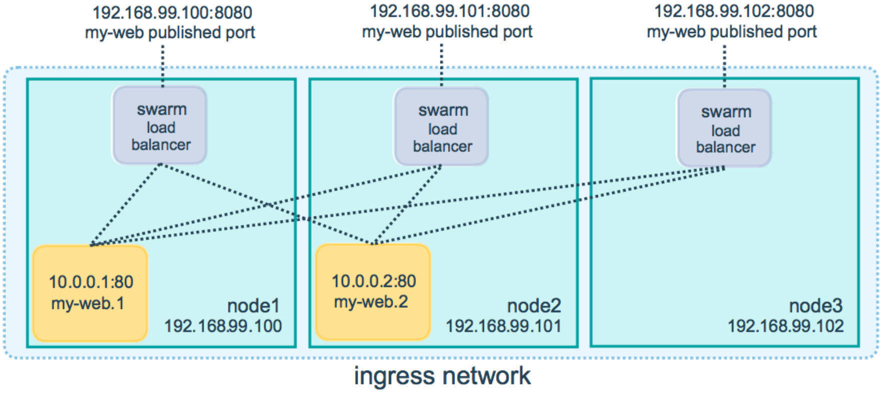
For local volumes based setup, you can backup the volume associated with DTR registry.

Module 7: Overview of Swarm Routing Mesh

There can be multiple nodes within a swarm cluster.

When creating a service, the task can be assigned to nodes depending on various factors.

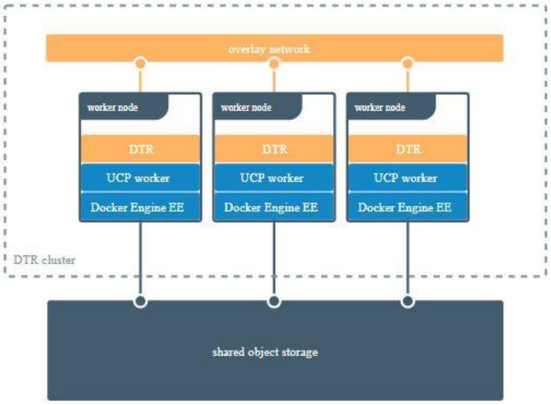
Routing mesh enables each node in the swarm to accept connections on published ports for any service running in the swarm, even if there’s no task running on the node.



Module 8: DTR - Storage BackEnds

By default, Docker Trusted Registry stores images on the filesystem of the node where it is running, but you should configure it to use a centralized storage backend.

You can configure DTR to use an external storage backend, for improved performance or high availability.



Some of the supported storage systems in DTR are:

Local:

- NFS
- Bind Mount
- Volume

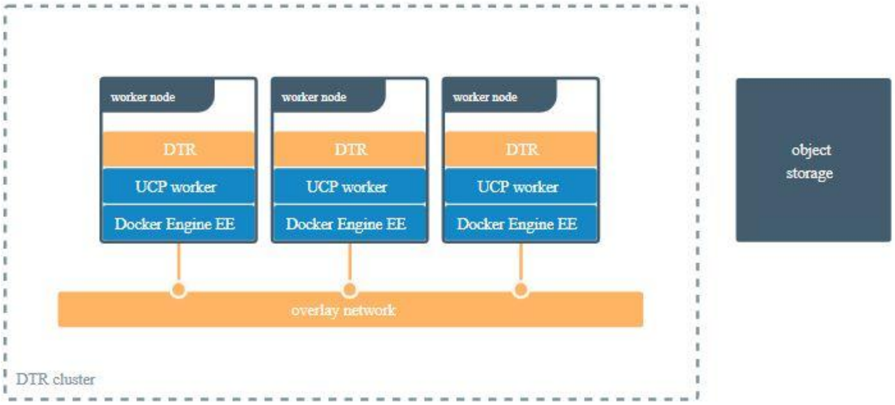
Cloud Storage Provider:

- AWS S3
- Azure
- Google Cloud

Module 9: DTR - High Availability

Docker Trusted Registry is designed to scale horizontally as your usage increases. You can add more replicas to make the DTR scale to your demand and for high availability.

All DTR replicas run the same set of services and changes to their configuration are automatically propagated to other replicas.



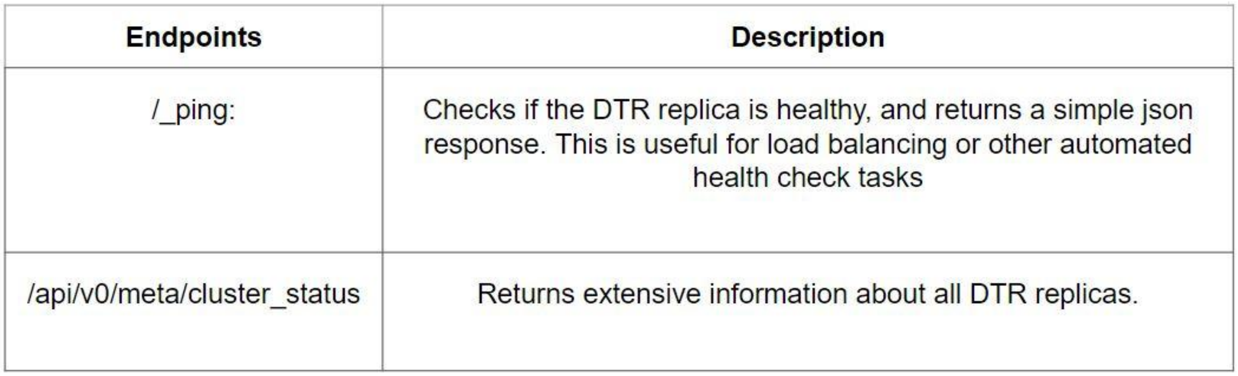
If your DTR deployment only has a single replica, you can continue using the local filesystem to store your Docker images.

If your DTR deployment has multiple replicas, for high availability, you need to ensure all replicas are using the same storage backend.

Ensure a sufficient amount of replicas are added to provide fault tolerance.

DTR Replicas	Fault Tolerated
1	0
3	1
5	2
7	3

DTR also exposes several endpoints you can use to assess if a DTR replica is healthy or not:



Endpoints	Description
/_ping:	Checks if the DTR replica is healthy, and returns a simple json response. This is useful for load balancing or other automated health check tasks
/api/v0/meta/cluster_status	Returns extensive information about all DTR replicas.

Important Pointers

To have high-availability on UCP and DTR, you need a minimum of:

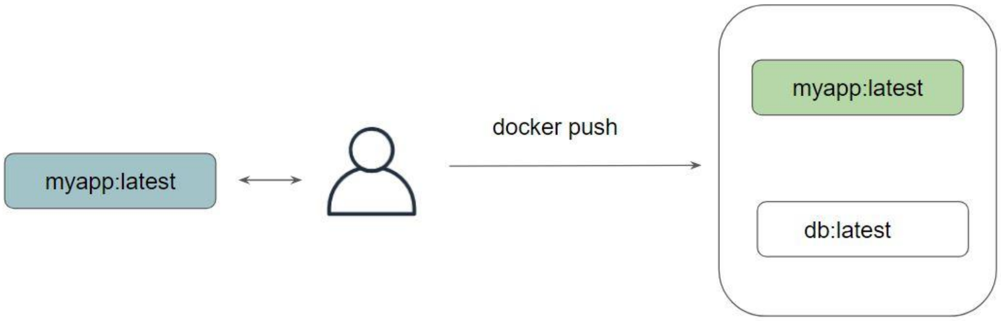
- 3 dedicated nodes to install UCP with high availability,
- 3 dedicated nodes to install DTR with high availability,

When a replica fails, the number of failures tolerated by your cluster decreases. Don’t leave that replica offline for long.

Adding too many replicas to the cluster might also lead to performance degradation, as data needs to be replicated across all replicas.

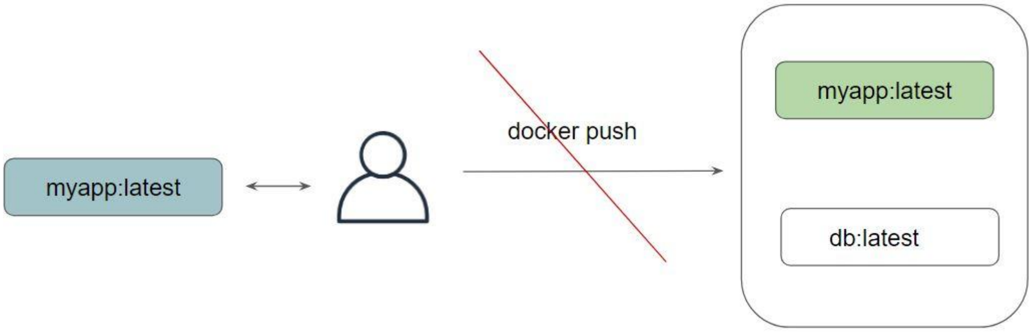
Module 10: Immutable Tags in DTR

By default, users with read and write access to a repository can push the same tag multiple times to that repository.



To prevent tags from being overwritten, you can configure a repository to be immutable.

Once configured, DTR will not allow anyone else to push another image tag with the same name.

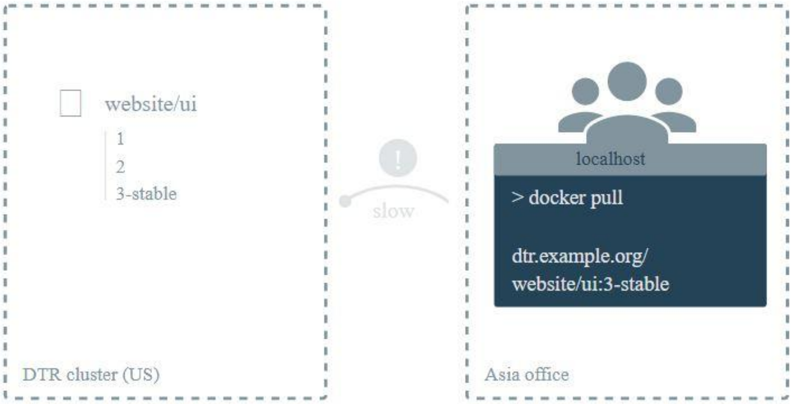


Module 11: DTR - Caching

11.1 Understanding the Challenge

The further away you are from the geographical location where DTR is deployed, the longer it will take to pull and push images.

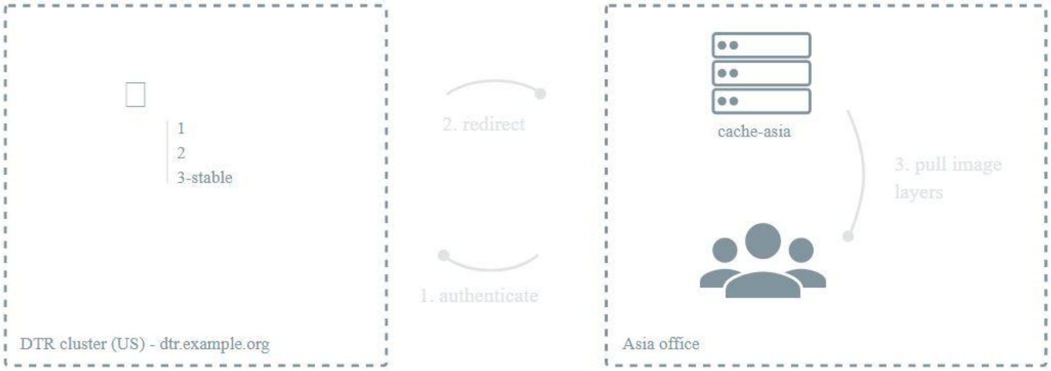
This happens because the files being transferred from DTR to your machine need to travel a longer distance, across multiple networks.

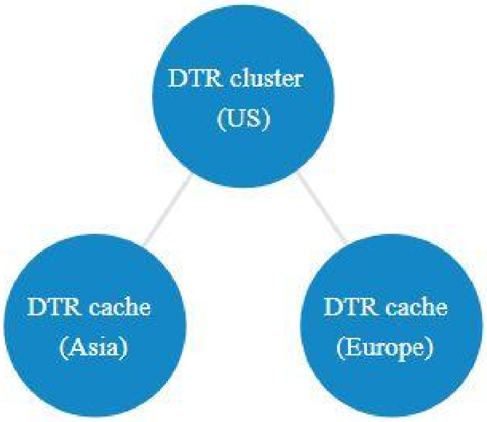


11.2 Implementing Cache at Location

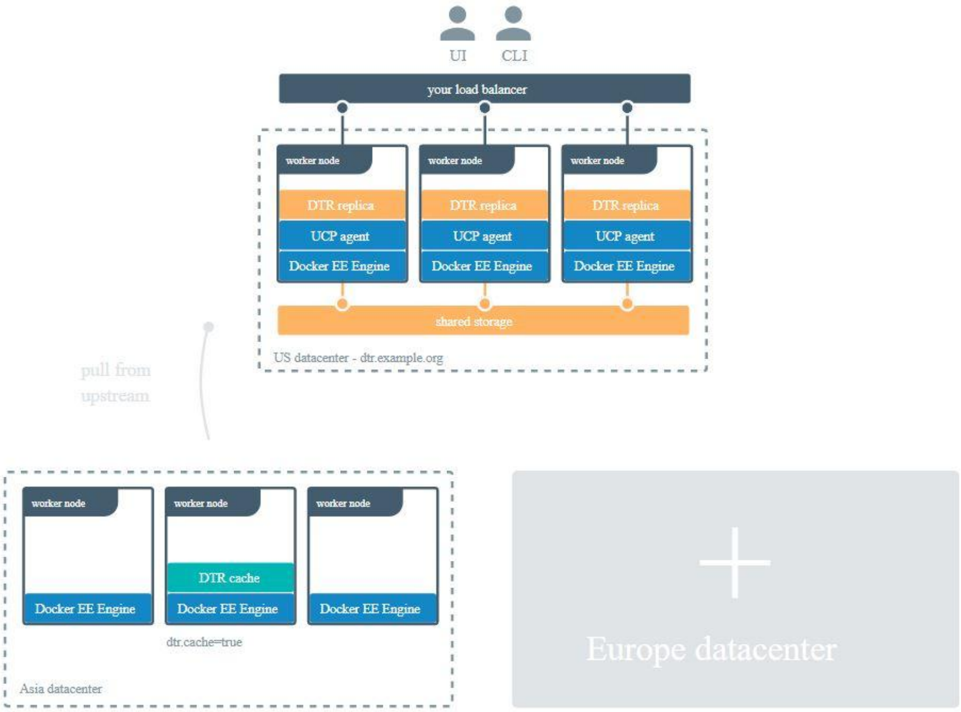
To decrease the time to pull an image, you can deploy DTR caches geographically closer to users.

Caches are transparent to users since users still log in and pull images using the DTR URL address. DTR checks if users are authorized to pull the image, and redirects the request to the cache.





The following diagram illustrates the architecture:



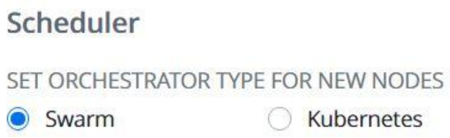
Module 12: Setting Orchestrator in UCP

12.1 Orchestrator Types in UCP

Docker UCP supports both Swarm and Kubernetes.

When you add a node to the cluster, the node’s workloads are managed by a default orchestrator, either Docker Swarm or Kubernetes.

When you install Docker Enterprise, new nodes are managed by Docker Swarm, but you can change the default orchestrator to Kubernetes in the administrator settings.



Scheduler

SET ORCHESTRATOR TYPE FOR NEW NODES

● Swarm ○ Kubernetes

12.2 Orchestrator Types For Nodes

You can change the current orchestrator for any node that’s joined to a Docker Enterprise cluster. The available orchestrator types are Kubernetes, Swarm, and Mixed.

The Mixed type enables workloads to be scheduled by Kubernetes and Swarm both on the same node.

When you change the orchestrator type for a node, existing workloads are evicted, and they’re not migrated to the new orchestrator automatically

12.3 Change Orchestrator Type via CLI

To change the orchestrator type for a node from Swarm to Kubernetes:

```
docker node update --label-add com.docker.ucp.orchestrator.kubernetes=true <node-id>
```