Portainer POC Test Plan

Objective

The objective of this document is to provide an outline of the recommended steps and related resources for building a proof of concept (POC) for a Portainer installation.The POC is intended to target an enterprise-level deployment with best practices, and covers the most common scenarios that an enterprise is likely to require. Some tasks are required as part of any setup, and others may be optional depending on the features or functionality that are required in each individual solution.

Test Set	**Notes**	**Status**
Common Tasks		
Installing Portainer Server	Required	
On-Premise / Hybrid Cloud		
Adding a standard environment	Repeatable	
Deploy a stack (Docker)	Docker only	
Deploy an application (Kubernetes)	Kubernetes only	
Edge / IIoT		
Adding an Edge environment	Repeatable	
Configure Edge Groups		
Deploy an Edge Stack		

Common Tasks

Installing Portainer Server

Goal

Install the Portainer Server on a management environment, configure the Portainer Server instance according to security best practices, and add container registries to Portainer.

Concepts

- Installation and initial setup
- Custom SSL configuration
- Firewall configuration
- Portainer backups
- External authentication and groups
- Custom registries

Prerequisites

- □ A containerization environment upon which to install the Portainer Server.
- □ Access to the above environment (root / admin access preferred).
- □ A supported external authentication provider (MS AD / LDAP / OAuth).
- □ A trusted SSL certificate and key for use with the Portainer Server instance.
- □ An S3-compatible bucket (and the related credentials) for Portainer backups.
- □ A Portainer BE license key.

Installing Portainer Server - Tasks

Task	**Detail**	**Documentation**	**Completed?**
Install Portainer BE	Connect to your containerization environment and complete the initial installation steps for your particular environment type.	Choose **one**: - Docker Standalone: https://docs.portainer.io/start/install/server/docker - Docker Swarm: https://docs.portainer.io/start/install/server/swarm - Kubernetes: https://docs.portainer.io/start/install/server/kubernetes	
Initial setup	Complete the initial setup of the Portainer Server instance. - Create an admin user - Enable or disable stats - Add license key	https://academy.portainer.io/install/#/lessons/oIqyv45zpqmH2p3h9DT8ssTZv3Pui2ww	
Secure Portainer	Secure your new Portainer Server installation. - Add SSL certificate - Configure your firewall - Configure backups	https://academy.portainer.io/install/#/lessons/Za5gHy6QPI5IUy7PGqzq6Z8DMTNYowng	
Configure access	Set up access to Portainer for your users and groups. - Create teams - Add external auth provider	https://academy.portainer.io/install/#/lessons/O_jpOK1OjVo2_9305Wu1F6a26PpHugh3	
Add registries	Add custom container registries to Portainer. - Repeat for each registry to add.	https://academy.portainer.io/install/#/lessons/3gkxaLjJ6DSsVTLsoTHJGZK-CzD0dRka (points 1 and 2)	

On-Premise / Hybrid Cloud

Adding a standard environment

Goal

Add an environment you wish to manage to your Portainer Server instance, configure user and team access to the environment, configure security settings for the environment, and add registry access on the environment.

Note: This Test Set can be completed multiple times if you have multiple environments to add.

Concepts

- Standard environments
- Role-based Access Control (RBAC)
- Environment security
- Change windows
- Registry access

Prerequisites

- ☐ A working Portainer Server installation (see Installing Portainer Server).
- ☐ A containerization environment to add to Portainer.
- ☐ Access to the above containerization environment (root / admin access preferred).

Adding a standard environment - Tasks

Task	Detail	Documentation	Completed?
Add environment	Add an existing environment to Portainer. - Select **Environment-related** then **Add environment**. - Select the **environment type** and click **Start Wizard**. - Select the **Agent** and deploy on your environment using the provided script. - Complete the **Name** and **Environment address** fields. - Expand **More settings** and set **Tags** to help identify the environment (for example location, blue/green for deployments, etc). - Click **Connect** to complete the environment setup. - Click **Close** to exit the environment wizard.	Choose **one**: - Docker Standalone: https://docs.portainer.io/admin/environments/add/docker/agent - Docker Swarm: https://docs.portainer.io/admin/environments/add/swarm/agent - Kubernetes: https://docs.portainer.io/admin/environments/add/kubernetes/agent	
Configure environment access	Configure access to the new environment for your users and teams. - Select **Environment-related** then **Manage Access** for your new environment. - Select a **user** or **team** and a corresponding **role** to assign then click **Create Access**. - Repeat for each user / team that requires access.	https://academy.portainer.io/install/#/lessons/FRBtcFGedd2Vby2ntSBpBinU6OB9Lfn6	
Configure environment security	Set configuration options on the new environment to ensure security. - Select the environment from the **Dashboard** then	https://academy.portainer.io/install/#/lessons/k2qXexuRW6_AsgjZml-E_RKMYxmM_QZ_	

	select **Host** / **Swarm** / **Cluster** > **Setup**. - Enable and configure a **Change window** for the environment. For Docker: - Configure Docker security settings. For Kubernetes: - Configure Kubernetes settings. - Select **Cluster** > **Security Constraints**, enable **pod security constraints**, and configure settings.		
Add registries to environment	Configure access to registries added in Task Set 1 on the environment. - Select the environment from the **Dashboard** then: - Select **Host** > **Registries** (Docker Standalone). - Select **Swarm** > **Registries** (Docker Swarm). - Select **Cluster** > **Registries** (Kubernetes). - Repeat for each registry you wish to add.	https://academy.portainer.io/install/#/lessons/3gkxaLjJ6DSsVTLsoTHJGZK-CzD0dRka (point 3)	

Deploy a stack (Docker)

Goal

Deploy a stack on a Docker environment using both a manually provided stack file and a Git-sourced stack file, ensuring deployment completes successfully. Make a change to the stack and deploy the change.

Concepts

- Stack deployment
- Web editor vs GitOps
- Automatic vs manual updates

Prerequisites

- □ A working Portainer Server installation (see Installing Portainer Server).
- □ A Docker Standalone or Docker Swarm environment configured (see Adding a standard environment).
- □ A basic stack file to deploy.
- □ A Git repository that you are able to commit to that contains a stack file to deploy.

Deploy a stack (Docker) - Tasks

Task	Detail	Documentation	Completed?
Deploy a stack manually	Create a stack in Portainer using a YAML file. - Select **Stacks** and click **Add stack**. - **Name** the stack and paste the YAML file in the **Web editor**. - Click **Deploy the stack**. - Once deployed, check the status of the stack by clicking the stack name and checking the **Services** section.	https://academy.portainer.io/deploy/docker/#/lessons/xT-nglvregGf_ASo7qPxRB_RJkMMbcv4	
Update a manually deployed stack	Make a change to a stack in Portainer and deploy the change. - Select **Stacks** and click the name of the stack to edit. - Select the **Editor** tab. - Make a change to the YAML in the **Web editor**. - Click **Update the stack**. - Once updated, check the status of the stack by clicking the stack name and checking the **Services** section.	https://docs.portainer.io/user/docker/stacks/edit#editing-a-stack	
Deploy a stack from Git	Create a stack in Portainer from a Git repository containing a stack file. - Select **Stacks** and click **Add stack**. - **Name** the stack and select **Repository**. - Provide the **Repository URL**, **reference**, and **compose path**. - If required, enable **Authentication** and provide auth details. - Enable **GitOps updates**, select **Polling**, and define	https://docs.portainer.io/user/docker/stacks/add#option-3-git-repository	

	a **Fetch interval**. - Click **Deploy** the stack. - Once deployed, check the status of the stack by clicking the stack name and checking the **Services** section.		
Update a Git deployed stack	Make a change to a stack in Git and deploy the change. - In the Git repository for your stack, make a change to the stack YAML file. - Commit the change. EITHER: Automatically update: - Wait for the stack's **fetch interval** to pass and the deployed stack to automatically update. OR: Manually trigger an update: - Select **Stacks** and click the name of the stack. - Click **Pull and redeploy**. Once updated: - Check the status of the stack by clicking the stack name and checking the **Services** section.	https://docs.portainer.io/user/docker/stacks/add#gitops-updates	

Deploy an application (Kubernetes)

Goal

Deploy an application on a Kubernetes environment using both a manually provided manifest file and a Git-sourced manifest file, ensuring deployment completes successfully. Make a change to the application and deploy the change.

Concepts

- Application deployment
- Web editor vs GitOps
- Automatic vs manual updates

Prerequisites

- ☐ A working Portainer Server installation (see Installing Portainer Server).
- ☐ A Kubernetes environment configured (see Adding a standard environment).
- ☐ A basic manifest file to deploy.
- ☐ A Git repository that you are able to commit to that contains a manifest file to deploy.

Deploy an application (Kubernetes) - Tasks

Task	Detail	Documentation	Completed?
Deploy an application manually	Create an application in Portainer using a YAML file. - Select **Applications** and click **Create from manifest**. - Select **Web editor** and a **namespace**. - Paste the YAML file in the **Web editor**. - Click **Deploy**. - Once deployed, check the status of the application by clicking the name and checking the **Application containers** section.	https://academy.portainer.io/deploy/kubernetes/#/lessons/V8P48DFV8S4CSQL7aiWLcFvD0pHLf4pk	
Update a manually deployed application	Make a change to an application in Portainer and deploy the change. - Select **Applications** and click the name of the application to edit. - Click **Edit this application**. - Make a change to the YAML in the **Web editor**. - Click **Update application**. - Once updated, check the status of the application by clicking the name and checking the **Application containers** section.	https://docs.portainer.io/user/kubernetes/applications/edit	
Deploy an application from Git	Create an application in Portainer from a Git repository containing a manifest file. - Select **Applications** and click **Create from manifest**. - Select **Repository** and a **namespace**. - Provide the **Repository URL**, **reference**, and **manifest path**.	https://docs.portainer.io/user/kubernetes/applications/manifest#option-1-git-repository	

	- If required, enable **Authentication** and provide auth details. - Click **Deploy**. - Once deployed, check the status of the application by clicking the name and checking the **Application containers** section.		
Update a Git deployed application	Make a change to an application in Git and deploy the change. - In the Git repository for your application, make a change to the YAML manifest. - Commit the change. EITHER: Automatically update: - Wait for the application's **fetch interval** to pass and the deployed application to automatically update. OR: Manually trigger an update: - Select **Applications** and click the name of the application. - Click **Edit this application**. - Click **Pull and update application**. Once updated: - Check the status of the application by clicking the name and checking the **Application containers** section.	https://docs.portainer.io/user/kubernetes/applications/manifest#gitops-updates	

Edge / IIoT

Adding an Edge environment

Goal

Add an Edge environment or device you wish to manage to your Portainer Server instance, configure user and team access to the Edge environment, configure security settings for the Edge environment, and add registry access on the environment.

Note: This Test Set can be completed multiple times if you have multiple Edge environments to add.

Concepts

- Edge environments
- Role-based Access Control (RBAC)
- Environment security
- Change windows
- Registry access

Prerequisites

- ☐ A working Portainer Server installation (see Installing Portainer Server).
- ☐ An Edge environment or device running a containerization environment to add to Portainer.
- ☐ Access to the above Edge environment or device (root / admin access preferred).

Adding an Edge environment - Tasks

Task	Detail	Documentation	Completed?
Add environment	Add an Edge environment to Portainer. - Select **Environment-related** then **Add environment**. - Select the **environment type** and click **Start Wizard**. - Select **Edge Agent Standard**. - Complete the **Name**, **Portainer API server URL** and **Portainer tunnel server address** fields. - Expand **More settings** and set **Tags** to help identify the environment (for example location, blue/green for deployments, etc). - Click **Connect**. - Deploy the Edge Agent on your environment using the provided **script**. - Click **Close** to exit the environment wizard.	Choose **one**: - Docker Standalone: https://docs.portainer.io/admin/environments/add/docker/edge - Docker Swarm: https://docs.portainer.io/admin/environments/add/swarm/edge - Kubernetes: https://docs.portainer.io/admin/environments/add/kubernetes/edge	
Configure environment access	Configure access to the Edge environment for your users and teams. - Select **Environment-related** then **Manage Access** for your Edge environment. - Select a **user** or **team** and a corresponding **role** to assign then click **Create Access**. - Repeat for each user / team that requires access.	https://academy.portainer.io/install/#/lessons/FRBtcFGedd2Vby2ntSBpBinU6OB9Lfn6	
Configure environment security	Set configuration options on the Edge environment to ensure security. - Select the environment	https://academy.portainer.io/install/#/lessons/k2qXexuRW6_AsgjZml-E_RKMYxmM_QZ_	

	from the **Dashboard** then select **Host** / **Swarm** / **Cluster** > **Setup**. - Enable and configure a **Change window** for the environment. For Docker: - Configure **Docker security** settings. For Kubernetes: - Configure Kubernetes settings. - Select **Cluster** > **Security Constraints**, enable **pod security constraints**, and configure settings.		
Add registries to environment	Configure access to registries added in Task Set 1 on the environment. - Select the environment from the **Dashboard** then: - Select **Host** > **Registries** (Docker Standalone). - Select **Swarm** > **Registries** (Docker Swarm). - Select **Cluster** > **Registries** (Kubernetes). - Repeat for each registry you wish to add.	https://academy.portainer.io/install/#/lessons/3gkxaLjJ6DSsVTLsoTHJGZK-CzD0dRka (point 3)	

Configure Edge Groups

Goal

Configure Edge Groups containing logical separations of your Edge environments for your setup, both statically and dynamically.

Concepts

- Edge Groups
- Static vs Dynamic groups

Prerequisites

- □ A working Portainer Server installation (see Installing Portainer Server).
- □ At least one Edge environment configured in Portainer (see Adding an Edge environment).

Configure Edge Groups - Tasks

Task	Detail	Documentation	Completed?
Enable Edge Compute	Enable the Edge Compute functionality in order to use Edge Groups. - Select **Settings** > **Edge Compute**. - Toggle **Enable Edge Compute features** to on.	https://docs.portainer.io/admin/settings/edge#edge-compute-settings	
Create a static Edge Group	Create a statically defined Edge Group containing specific environments. - From Edge compute select **Edge Groups**. - Click **Add Edge group** and enter a **Name** for the group. - With **Static** selected, manually add environments from the **Available environments** list to the **Associated environments** list by clicking on each environment. - Click **Add edge group**.	https://docs.portainer.io/user/edge/groups	
Create a dynamic Edge Group	Create a dynamically defined Edge Group containing environments with specific tags. - From Edge compute select **Edge Groups**. - Click **Add Edge group** and enter a **Name** for the group. - Select **Dynamic** and **Partial Match**. - Select a **tag** from the dropdown that will be used to determine which environments to add to the Edge Group. - Click **Add edge group**.	https://docs.portainer.io/user/edge/groups#option-2-dynamic	

Deploy an Edge Stack

Goal

Deploy a stack on a Docker environment using both a manually provided stack file and a Git-sourced stack file, ensuring deployment completes successfully.

Concepts

- Edge Stacks
- Web editor vs GitOps
- Automatic vs manual updates
- Edge Stack update methods (all at once vs parallel)

Prerequisites

- ☐ A working Portainer Server installation (see Installing Portainer Server).
- ☐ An Edge environment configured in Portainer (see Adding an Edge environment).
- ☐ An Edge Group configured containing your Edge environment(s) (see Configure Edge Groups).
- ☐ A basic stack file to deploy that matches your Edge environment type(s).
- ☐ A Git repository that you are able to commit to that contains a stack file to deploy that matches your Edge environment type(s).

Deploy an Edge Stack - Tasks

Task	Detail	Documentation	Completed?
Deploy an Edge Stack manually	Create an Edge Stack from a YAML file and deploy it to your Edge environment(s) via Edge Groups. - From Edge compute select **Edge Stacks**. - Click **Add stack**. - **Name** your stack then select the **Edge Group(s)** to use. - Paste your stack YAML file into the **Web editor**. - Click **Deploy the stack**. - Once deployed, check the status of the stack from the **Edge Stacks** page.	https://docs.portainer.io/user/edge/stacks/add	
Update a manually created Edge Stack	Make a change to a manually deployed Edge Stack and deploy the change. - Select **Edge Stacks** and click the name of the stack to edit. - Make a change to the YAML in the **Web editor**. - In the **Update configurations** section, choose **Parallel edge device(s)**. - Select from the dropdown whether to specify the **number of device(s)** or use an **exponential rollout**, and complete the related fields. - Optionally set a **Timeout** and **Update delay**, and choose the **Update failure** action. - Click **Update the stack**. - Once updated, check the status of the stack from the **Edge Stacks** page.	https://docs.portainer.io/user/edge/stacks/add#web-editor https://docs.portainer.io/user/edge/stacks/add#update-configurations	

Deploy an Edge Stack from Git	Create an Edge Stack from a Git repository and deploy it to your Edge environment(s) via Edge Groups. - From Edge compute select **Edge Stacks**. - Click **Add stack**. - **Name** your stack then select the **Edge Group(s)** to use. - Select **Repository**. - Provide the **Repository URL**, **reference**, and **compose** or **manifest path**. - If required, enable **Authentication** and provide auth details. - Enable **GitOps updates**, select **Polling**, and define a **Fetch interval**. - Click **Deploy the stack**. - Once deployed, check the status of the stack from the **Edge Stacks** page.	https://docs.portainer.io/user/edge/stacks/add#repository	
Update a Git deployed Edge Stack	Make a change to an Edge Stack deployed from Git and deploy the change. - In the Git repository for your application, make a change to the YAML. - Commit the change. EITHER: Automatically update: - Wait for the Edge Stack's **fetch interval** to pass and the deployed stack to automatically update. OR: Manually trigger an update: - Select **Edge Stacks** and click the name of the stack. - Click **Pull and update stack**. Once updated:	https://docs.portainer.io/user/edge/stacks/add#gitops-updates	

	- Check the status of the stack from the **Edge Stacks** page.		