

Sol4Cloud ALinkServer documentation

Version: 1.0.9

Date: October 28, 2026

Copyright © 2026 Sol4Cloud s.r.o. All rights reserved.

Trademark Disclaimer: SAP and S/4HANA are registered trademarks of SAP SE in Germany and other countries. Amazon Web Services, AWS, and Amazon S3 are trademarks of Amazon.com, Inc. or its affiliates. All other trademarks are the property of their respective owners.

Table of Contents

ALinkServer Introduction....4
Components of AWS ALinkServer deployment....9
ALinkServer AWS ECS deployment components....9
AMI EKS deployment components....13
AMI EC2 deployment components....14
Deploying ALinkServer....17
ECS „Quick create stack“ Deployment....17
EKS HELM Deployment....22
Amazon Machine Image EC2 deployment....24
First start of the ALinkServer deployment....27
ALinkServer Configuration....31
ALinkServer Administration....33
Administration of ECS deployment....33
Administration of EKS deployment....34
Administration of AMI EC2 deployments....34
Known product limitations....35
Q&A....36

ALinkServer Introduction

ALinkServer is a containerized implementation of the SAP® ArchiveLink 4.6 interface designed to archive SAP S/4HANA and SAP ECC data and documents directly to customer owned Amazon® S3 buckets. ALinkServer is available as a pay-per-use solution on the AWS Marketplace for deployment in Amazon Elastic Container Service (ECS) or Amazon Elastic Kubernetes Service (EKS) or as an Amazon Machine Image (AMI). AWS Marketplace listing of ECS/EKS is here, AMI product AWS Marketplace listing can be found here.

Architecture & AWS Footprint

Designed for effortless deployment and a minimal AWS infrastructure footprint, ALinkServer can be provisioned using AWS CloudFormation templates and runs natively on Docker containers via AWS Fargate within Amazon Elastic Container Service (ECS) or Amazon Elastic Kubernetes Service (EKS). AMI based version of the product provides CloudFormation deployment template and one AMI image that is deployed as part of the archiving infrastructure deployment

To minimize operational costs and complexity:

- Content Metadata: All content-related metadata is stored directly as S3 object metadata, eliminating the need for a relational database.
- Configuration: Server and SAP ArchiveLink repository configurations are stored in Amazon DynamoDB configuration table.
- Security: Administrator credentials and passwords are managed securely in AWS Secrets Manager.

Pricing & Billing

ALinkServer usage is paid per SAP ArchiveLink request processed by the ALinkServer. The application uses a two-tier billing model with an annual counter reset:

- Starting Tier: Covers up to 100,000 ArchiveLink requests per calendar year processed in AWS account in the respective AWS region.
- Volume Tier: Offers a reduced per-request rate for all processed requests exceeding 100,000 per AWS account per calendar year in the AWS region.

Chapter Components of AWS ALinkServer deployment below provides detailed information about supporting AWS objects that are necessary to run ALinkServer. You should expect additional cost in the following range:

- **VPC Endpoints:** 6 Interface Endpoints (Secrets Manager, STS, ECR, and AWS Marketplace Metering) incur a base hourly fee (typically $0.01 per endpoint per Availability Zone per hour) plus a nominal data processing fee per gigabyte. 3 Gateway Endpoints (S3, S3 in us-east-1 for Container access, DynamoDB) are provided free of charge.

- **Compute Cluster Control Plane:** If deployed on **Amazon ECS**, there is no additional control plane charge (using EC2 or Fargate launch types). If deployed on **Amazon EKS**, AWS charges a fixed control plane fee of $0.10 per hour (approximately $73/month) per cluster. AMI EC2 deployment uses two Auto Scaling Groups to manage ALinkServer instances. Auto Scaling Groups have no service fees.
- **Compute Resources (Containers or EC2 instances):** Billed based on allocated vCPU and RAM (e.g., via AWS Fargate rates or backing EC2 worker node instances):
    - **Admin Container/EC2 instance:** 1 vCPU and 2 GB RAM (runs as a single standalone instance).
    - **Worker Containers/EC2 instances:** 2 instances totaling 8 vCPUs and at least 16 GB RAM combined (4 vCPUs / 8 GB RAM per container).
- **Application Load Balancer (ALB):** Billed at a base rate of $0.0225 per hour (approximately $16.40/month) plus Load Balancer Capacity Units (LCU) based on active connections, new connections, and processed bandwidth.

ALinkServer ECS/EKS deployment relies on AWS backend services, EKS/ECS deployments are therefore at this moment only available in the following AWS regions:

us-east-1, us-east-2, us-west-1, us-west-2, eu-west-1, eu-central-1, eu-west-2, eu-west-3, eu-north-1, ap-east-1, ap-southeast-1, ap-northeast-1, ap-southeast-2, ap-northeast-2, ap-south-1, ca-central-1, sa-east-1

ALinkServer deployed using AMI EC2 deployment is available in the following AWS regions:

eu-north-1, me-south-1, ap-south-1, eu-west-3, ap-southeast-3, us-east-2, af-south-1, eu-west-1,me-central-1, eu-central-1, sa-east-1, ap-east-1, ap-south-2, us-east-1, ap-northeast-2, ap-northeast-3, eu-west-2, ap-southeast-4, eu-south-1, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-southeast-2, il-central-1, ca-central-1, eu-south-2, eu-central-2, us-gov-east-1, us-gov-west-1

Application Components

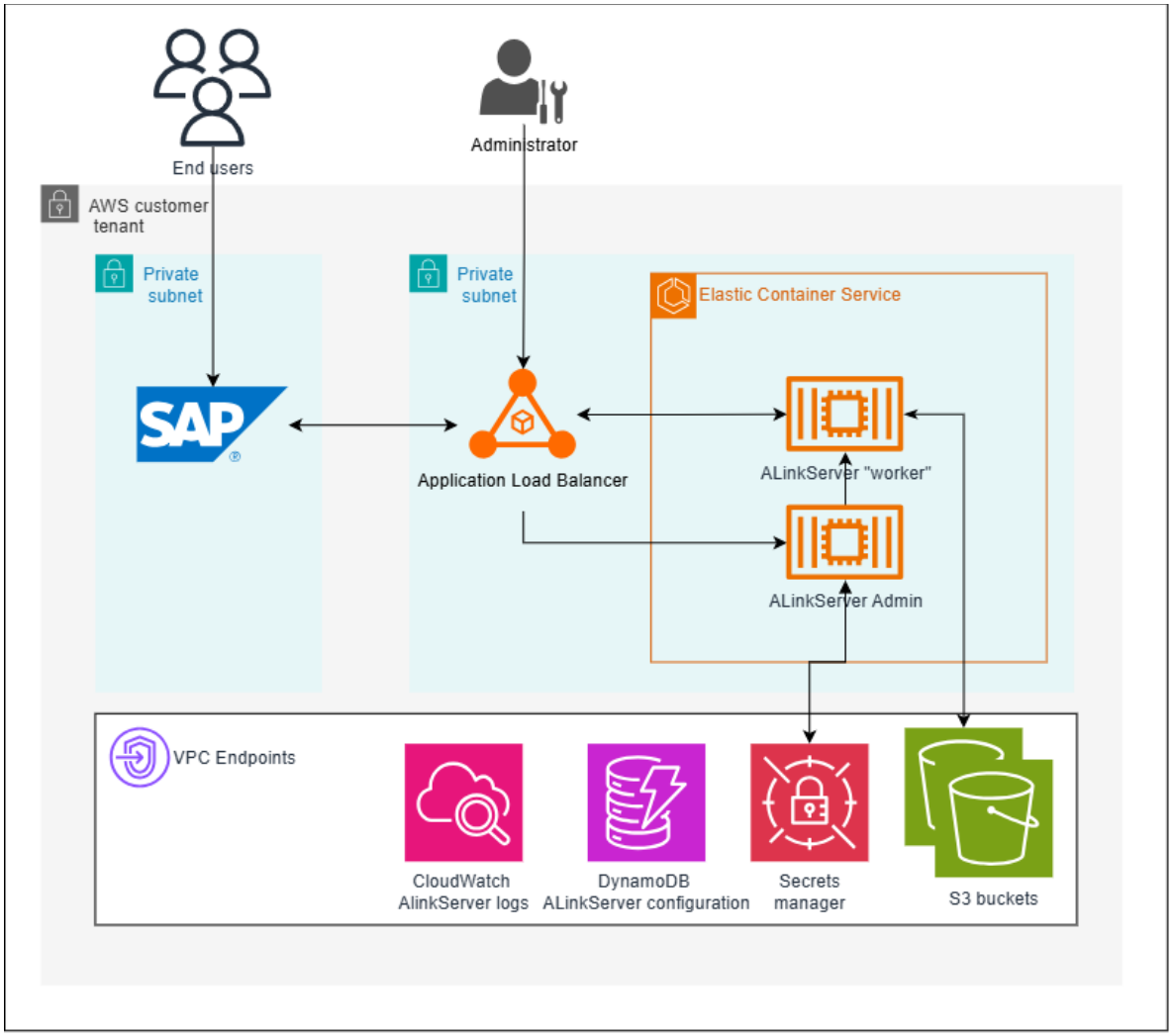
The ALinkServer solution consists of two main core containers/services:

1. ALinkServer (Worker Service): The main web application implementing the SAP ArchiveLink protocol. Its endpoint URLs are configured directly in the SAP transaction `OAC0`.
2. ALinkServer Admin: The management interface used to configure server settings, create content repositories, and manage administrator accounts. This process is also responsible for reporting solution usage metrics to AWS.

AWS CloudFormation stack is provided for ECS deployment of ALinkServer in customer's region of choice.

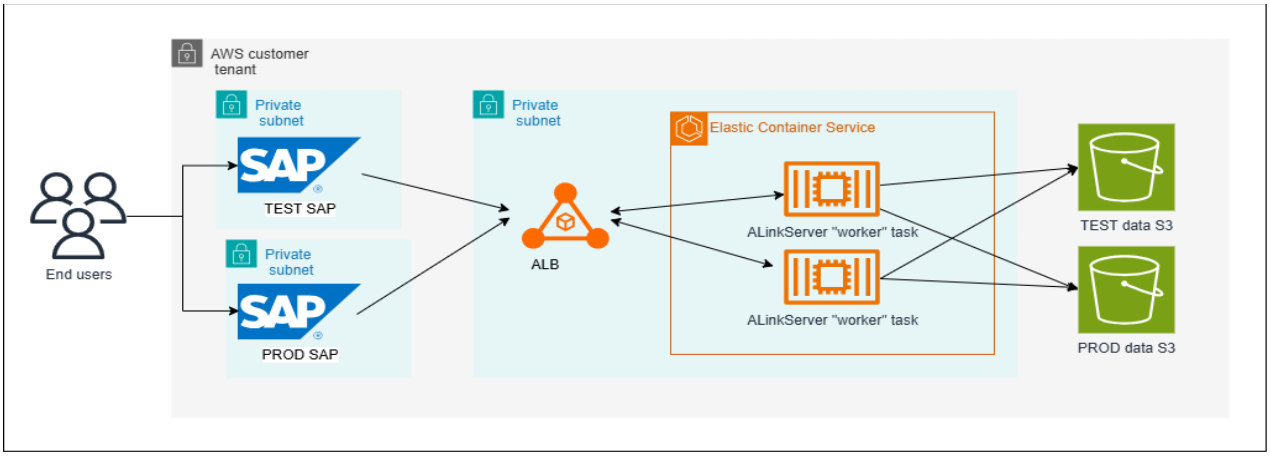
- ALinkServer containers are configured as Tasks in AWS Elastic Container Service (ECS).
- ECS is frontended by Application Load Balancer.
- CloudFormation template configures a number of VPC endpoints to ensure that all ALinkServer communication only runs over AWS internal network.

Following figure shows high level architecture of ALinkServer deployed on ECS in Amazon AWS (In EKS deployment the cluster is represented with Kubernetes cluster, in AMI EC2 deployment each service is scaled independently through own Auto Scaling Group):

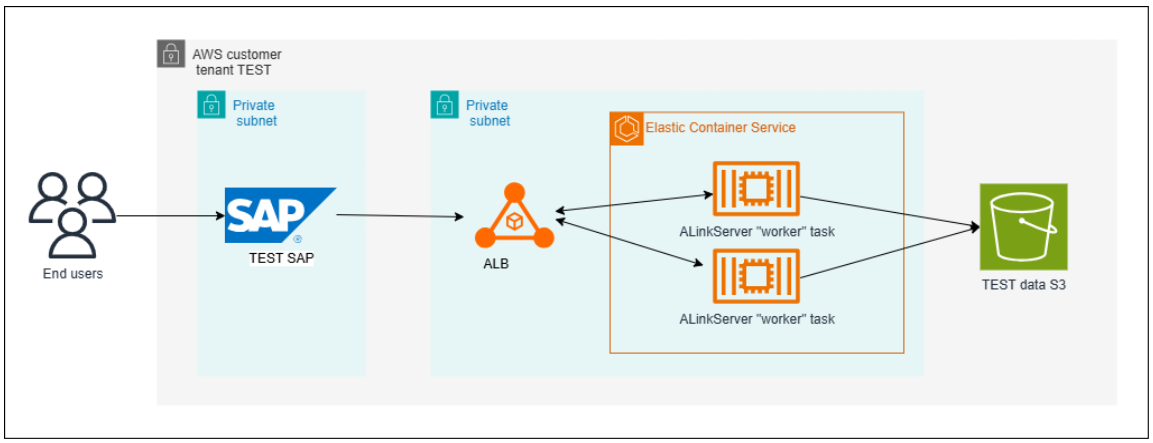


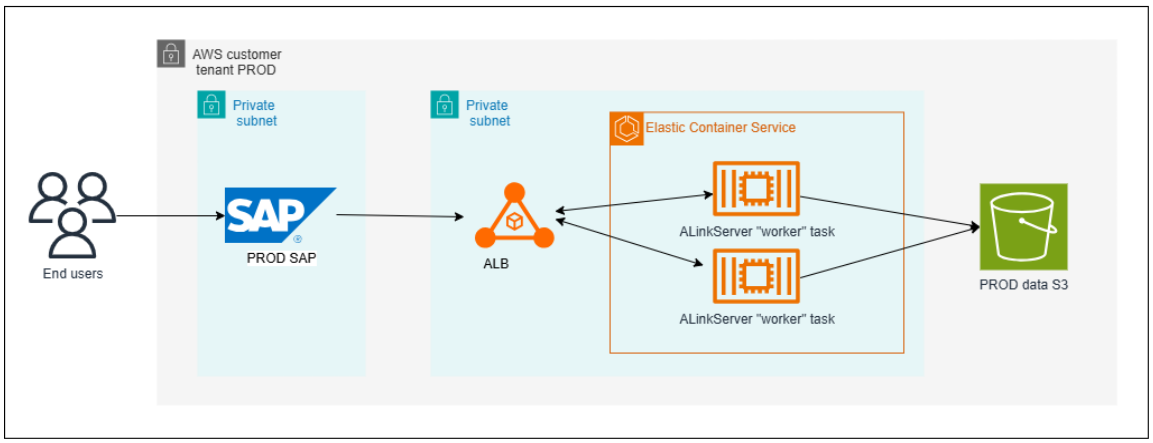
Minimum ALinkSever deployment would utilize the above architecture as one shared ALinkServer cluster for multiple SAP landscapes (DEV, TEST, PROD). In this configuration different target S3

buckets would be configured per SAP landscape/SAP repository. Picture below shows a conceptual architecture of this deployment:



In a full installation separate ALinkServer clusters would be deployed for each of the SAP landscapes (DEV, TEST, PROD), each in own AWS organizational account. Sizing and number of ECS tasks would differ by target SAP landscape – e.g. one 2 vCPU ECS ALinkServer “worker” task for DEV/TEST landscape, three 4 vCPU ECS tasks for PROD SAP landscape. Picture below shows a conceptual architecture of fully separated deployment in two AWS accounts:


AWS customer
tenant TEST
Private
subnet
Private
subnet
Elastic Container Service
End users
TEST SAP
ALB
ALinkServer "worker" task
ALinkServer "worker" task
TEST data S3

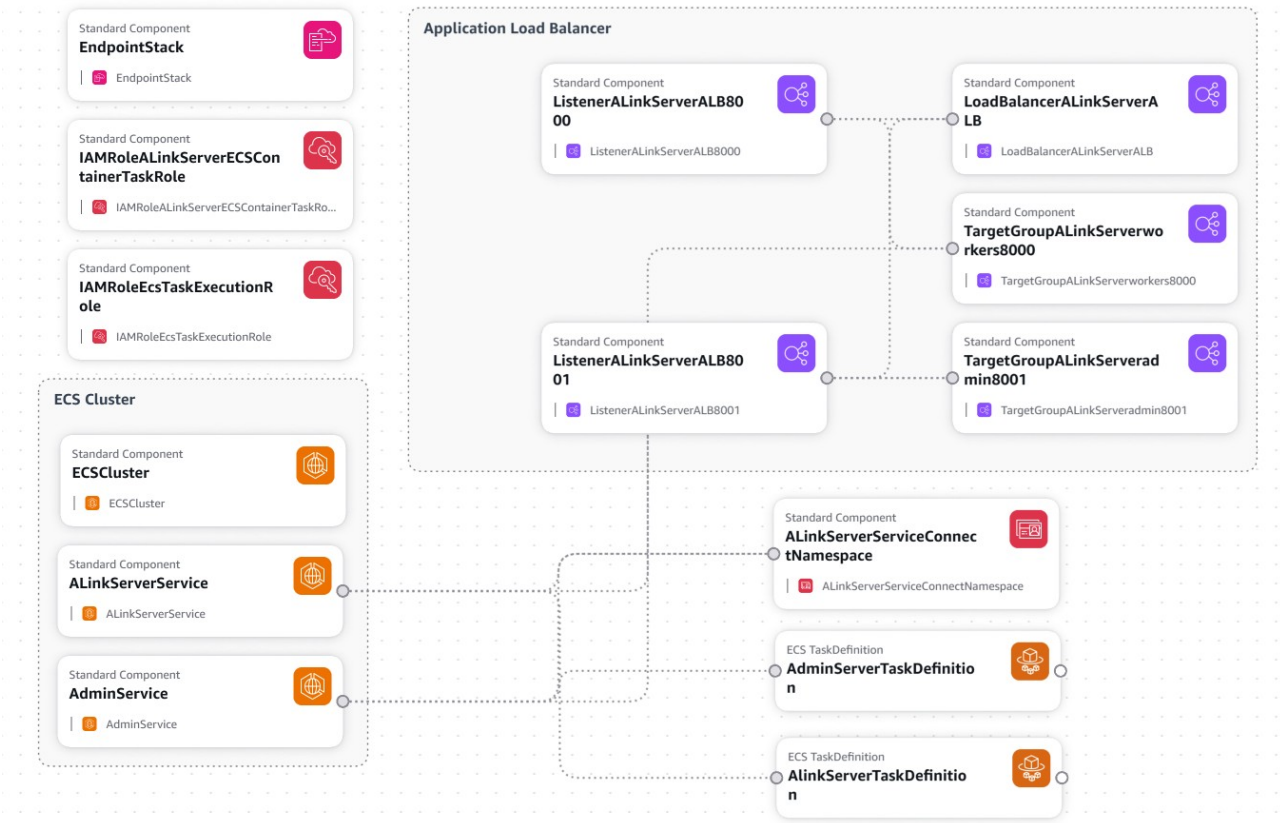

AWS customer
tenant PROD
Private
subnet
Private
subnet
Elastic Container Service
End users
PROD SAP
ALB
ALinkServer "worker" task
ALinkServer "worker" task
PROD data S3

Components of AWS ALinkServer deployment

This chapter provides details about AWS objects deployed for various AWS ALinkServer runtimes. Continue reading for details about ECS deployment, jump to ...

ALinkServer AWS ECS deployment components

ECS CloudFormation configuration file automatically provisions the following resources (see detailed explanation about all deployed objects on the following pages:



VPC endpoints

ALinkServer ECS/EKS CloudFormation deployment creates 9 VPC Endpoints for runtime operations. Us-east-1 Endpoints are not needed and therefore not created for AMI EC2 deployment.



ALinkServer utilizes the following Amazon AWS services during runtime:

- Amazon S3: S3 bucket is configured per SAP ArchiveLink repository to store archived data and documents.
- Amazon DynamoDB: Stores server and SAP content repository configuration data.
- AWS Secrets Manager: Stores administrator credentials securely.
- Amazon CloudWatch: Stores ALinkServer logs
- AWS Security Token Service: Requests temporary, limited-privilege security credentials
- Amazon Marketplace billing service: Reports usage metrics

To ensure traffic to these services does not traverse the public internet, CloudFormation provisions VPC Endpoints via AWS PrivateLink. All requests travel exclusively over AWS's internal network without requiring a NAT Gateway or Internet Gateway

Additionaly, 3 VPC Endpoints are created in us-east-1 region for access to AWS Marketplace infrastructure (these VPC Endpoints are not deployed for AMI EC2 installations):

- Marketplace Elastic Container Registry (ECR)
- Marketplace product Docker containers
- S3 storage hosting container binaries (this Endpoint is not created if you’re deploying ALinkServer in us-east-1 as only one S3 endpoint instance can exist per region)

ECS Task definitions

CloudFormation Stack script creates two ECS Task definitions:

- **AdminServerTaskDefinition**: Minimum specs (1 vCPU and 2 GB RAM) FARGATE task for the administration service.
- **ALinkServerTaskDefinition**: Configures container parameters, port mappings, and logging for worker tasks. Default specs are 4 vCPUs with 8GB RAM. Each ALinkServer worker is configured to launch 8 workers for the ArchiveLink requests processing.

Service Connect Namespace

In ECS deployment Worker ALinkServer service communicates with the Admin Server via AWS Service Connect. Service Connect publishes an internal DNS host name that is utilized by the worker servers. CloudFormation Stack template creates the unified namespace shared by both ALinkServer service and Admin service configurations. EKS deployment uses preconfigured internal hostname that is resolved in Kubernetes cluster namespace instead of Service Connect Namespace.

IAM Roles

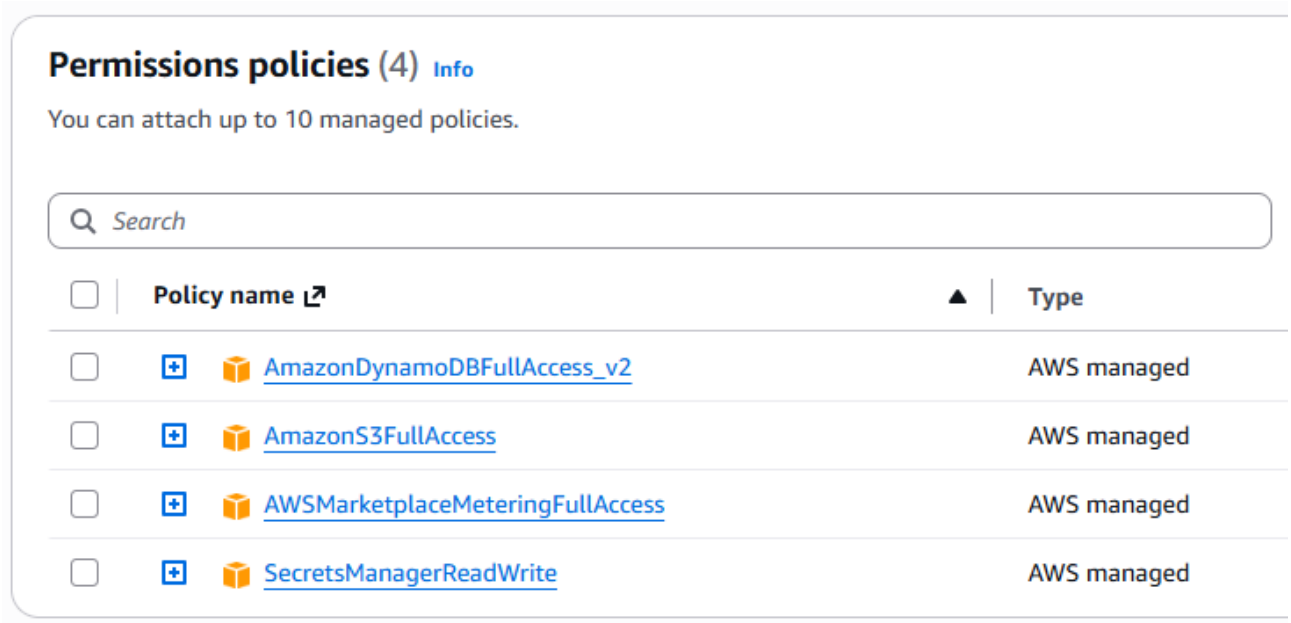
ECS CloudFormation stack creates two IAM roles:

- **Task execution role:** This role is used to launch ECS Service container and therefore needs access to get to Elastic Container Registry, Authenticate, downoad container and write logs. AWS managed policies are being used in the task execution role.
- **Task role:** ALinkServer worker and ALinkServer Admin services need authorizations to access several AWS resources – see paragraph “VPC Endpoints” above for more details. TaskRole needs the following authorizations:
    - **Amazon DynamoDB**: ALinkServer creates new DynamoDB table upon initial configuration and is reading, editing and deleting entries in this table when admin user changes the server configuration or when the Admin server reports the executed work to AWS Metering service. Default installation uses AWS Managed „Full Access“ policy – this can be replaced with a customer managed policy that will only provide access to ALinkServer configuration table – see below.
    - **S3 access:** ALinkServer creates repository folders in the target bucket and stores all ArchiveLink data and documents there. By design, neither worker nor Admin service create S3 buckets. Buckets need to be configured by AWS administrator, before

configuring ALinkServer repositories, in order to manage any optional Bucket configuration (e.g. encryption, bucket lifecycles etc.).

- **AWS Metering services**: Used for reporting the executed work to AWS
- **SecretsManager**: to safely manage admin user names and passwords. Default installation uses AWS managed SecertsManager policy that can be replaced with customer defined policy providing access only to ALinkServer secret – see below.

Default ECS CloudFormation Stack template uses the below AWS managed policies in the TaskRole:



ECS Cluster and services

CloudFormation Stack creates an ECS Cluster using Amazon FARGATE compute with two services:

- **AdminServer service (port 8001)**: Web administration UI for server settings, log levels, runtime environments, content repositories, and admin users. .
- **ALinkServer worker service (port 8000)**: Main worker application that implements SAP ArchiveLink interface.

Note: By default, public IP addresses are not assigned to running tasks. VPC internal Application Load Balancer will be created to route the traffic to available ALinkServer worker tasks.

Application Load Balancer

Application Load Balancer (ALB) is created to route the traffic to running ECS/EKS Cluster tasks or ASG EC2 instances. ALB default configuration is to use internal IP address. This configuration

will enable you to plug the ALB into your existing network architecture for SAP data and document archiving.

ALinkServer default configuration (as configured in the CloudFormation Stack) doesn’t terminate HTTPS/SSL connections. Your AWS administrator can change this configuration and let ALB terminate HTTPS traffic on ALB. EKS deployment configuration supports HTTPS/SSL configuration during installation.

AMI EKS deployment components

EKS deployment uses HELM charts with values.yaml file for simple deployment configuration. HELM chart deployment creates:

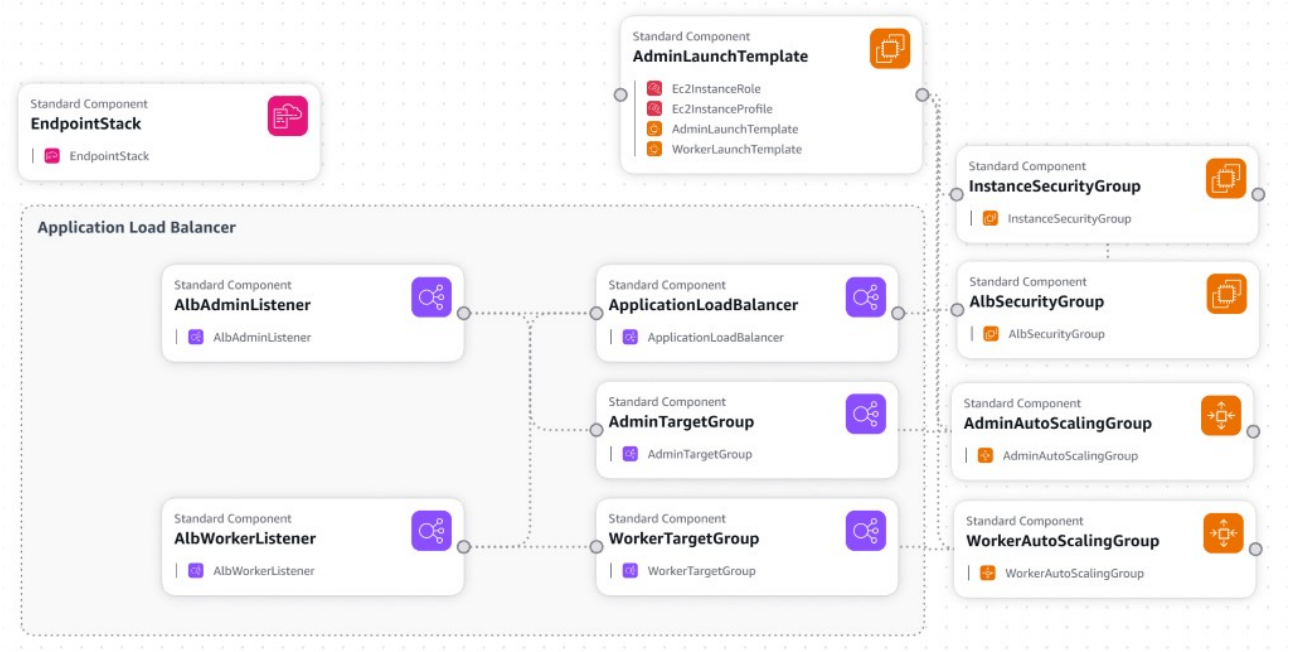
- Kubernetes pod definitions – one for admin pod and one for ALinkServer worker pods
- Service definitions for Admin service running on port 8001 and worker service running on port 8000
- Ingress configuration for creation of AWS Application Load Balancer

Additional CloudFormation template for VPC endpoints creation is provided as part of the installation to ensure that ALinkServer Kubernetes services communicate only over internal AWS network – for further details see VPC Endpoints description in the ECS chapter above.

IAM Role for execution of the ALinkServer pods is created during EKS installation – see IAM chapter in ECS section above for more details.

AMI EC2 deployment components

Picture below shows an overview of resources configured by AMI EC2 CloudFormation template. Paragraphs below provide details on objects that differ from the ECS CloudFormation template deployment.



AdminLaunchTemplate and WorkerLaunchTemplate: specify what types of EC2 instances should be started for each service and in the UserData section specify the services that should be started on the respective instance (Admin and Worker share the same AMI template and launch template defines the role that the instance will adopt at runtime)

Ec2InstanceRole and Ec2InstanceProfile: For a detailed overview of necessary AWS authorizations see chapter Components of AWS ALinkServer deployment). AMI EC2 CloudFormation template creates a role with all AWS Policies and assigns it to EC2 instances using EC2 instance profile.

To comply with AWS least privilege requirement for EC2 Marketplace products a customer managed policy is being created during deployment in oder to limit ALinkServer access to customer resources. Policy limits the access to:

- DynamoDB table `<aws customer id>-alinkserver-config`
- SecretsManager `Sol4Cloud_alinkserver_users` secret
- Account ID’s own S3 buckets with an (optional) bucket name prefix. Bucket name prefix can be specified when configuring CloudFormation parameters or can be left empty for ALinkServer to be able to use any name of S3 bucket within the account and region context.

AMI EC2 deployment creates a role for the execution of the EC2 instances as well. EC2 deployment adds the following two AWS managed policies to the instance role during ALinkServer deployment:

- **CloudWatchAgentServerPolicy** – This policy is necessary to enable EC2 instances' Cloud Watch Agent to write ALinkServer log data from system journal to AWS CloudWatch.
- **AmazonSSMManagedInstanceCore** – Authorization necessary for the ALinkServer code to access information about running instance from AWS Systems Manager

InstanceSecurityGroup and AlbSecurityGroup: CF template creates two security groups that allow inbound traffic on ports 8000 and 8001

AdminAutoScalingGroup and WorkerAutoScalingGroup: two Auto Scaling Groups for respective ALinkServer services. Configure Worker ASG to scale as needed based on your ArchiveLink load. Leave Admin ASG to maximum 1 instance due to below described limitation of the single threaded admin server.

For more information on ALinkServer jump to the next chapters in this documentation:

Deploying ALinkServer: Provides a step-by-step guide on how to deploy ALinkServer on AWS using CloudFormation stack scripts, EKS HELM scripts or AMI EC2 CloudFormation template to automatically create required resources (IAM roles, VPC endpoints, ECS/EKS tasks/cluster, and ALB)

First start of the ALinkServer deployment: Describes the steps required immediately after CloudFormation stack/HELM chart deployment, including initial server setup.

ALinkServer Configuration: Details application settings (logging, admin users, and SAP Content Repositories) managed via the ALinkServer Admin UI.

ALinkServer Administration: Details monitoring and scaling operations for the deployed ECS cluster.

Deploying ALinkServer

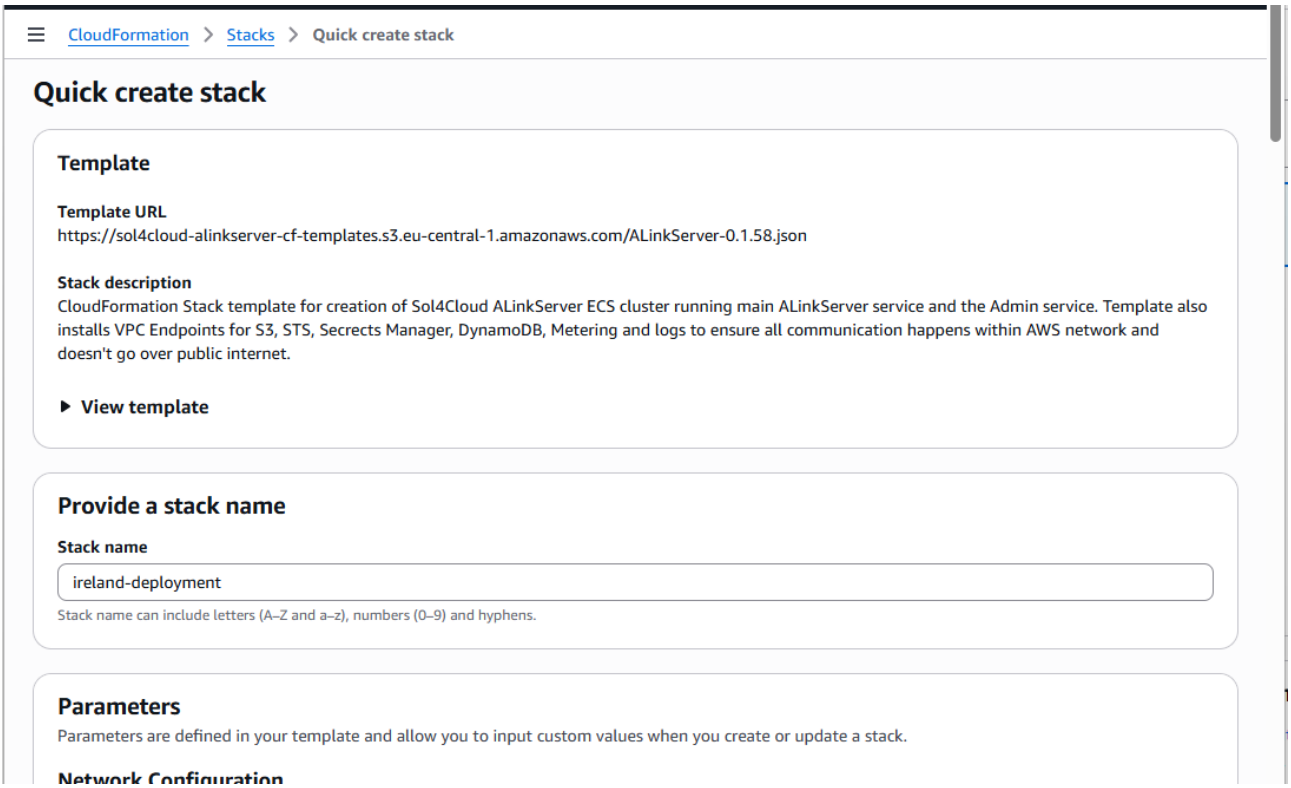
Once you've subscribed to ALinkServer (Containerized version or AMI EC2 version)¨on AWS Marketplace you will be able to access the **Launch your software** link on your subscription page.

Continue to read here for ECS deployment instructions or jump to EKS HELM Deployment chapter for instruction and details of the EKS deployment or jump to Amazon Machine Image EC2 deployment chapter below for details of the AMI EC2 deployment.

ECS „Quick create stack" Deployment

Following paragaphs provide detailed information about the configuration objects that are created using the CloudFormation template.

1. In the AWS Marketplace launch page, select **ECS** as the deployment target.
2. Click the **Quick create stack** link under **Deployment templates**.
3. AWS Console will open the CloudFormation Quick create stack dialog. Make sure that your deployment user is assigned AWS AdministratorAccess role and AWSCloudFormationFullAccess role. Select the desired target region in the top right corner of the AWS Console. Fill in the required parameters on the CloudFormation creation form:



Populate the following parameters for the ALinkServer Stack creation:

a) Network Configuration

- **Stack Name:** Enter a recognizable name for the deployment.
- **Target VPC:** Select the existing VPC where the application deployment will run.
- **Target Subnets:** Select at least two subnets within the chosen VPC. Make sure that the selected subnet allows installation of VPC Endpoints (some Availability zones don't support creation of VPC Endpoints). You can run the following AWS CLI command (replace <region> with your target region e.g. eu-west-2) to see list of supported availability zones:

```
aws ec2 describe-vpc-endpoint-services \
--service-names com.amazonaws.<region>.metering-marketplace \
--query "ServiceDetails[0].AvailabilityZones" \
--region <region>
```

- **Security Group ID:** Select a Security Group. Security group needs to be configured to allow inbound traffic on ports `8000` and `8001.` Security group also has to allow HTTPS/443 communication for VPC Endpoint connectivity.
- **Route Table ID:** Enter the ID of the VPC Route Table associated with your target subnets (required for VPC Endpoint routing). CloudFormation UI doesn't support listing route tables (as it does for other network configuration objects above) so an ID of route table needs to be inserted manually. Open list of Route tables in your VPC in a separate AWS Console browser window and copy-paste the ID of the route table in this parameter.
- **VPC Endpoints creation preference**: Select whether CloudFormation template should create the necessary VPC Endpoits. If you choose „No VPC Endpoints…" option than ALinkServer AWS calls will be routed over public internet unless you e.g. configure ALinkServer NAT networking separately.

b) ECS Cluster Settings

- ECS Deployment creates new ECS cluster. Default deployment utilizes FARGATE compute. Specify the below ECS cluster parameters:
- **ECS Cluster Name:** Keep default or enter a custom name.
- **ALinkServer Service Name:** Keep default or enter a custom worker service name.
- **Admin Service Name:** Keep default or enter a custom admin service name.
- **Service Connect Namespace Name:** Specify the AWS Service Connect namespace name or keep the default value.

c) ECS Cluster IAM Roles Settings

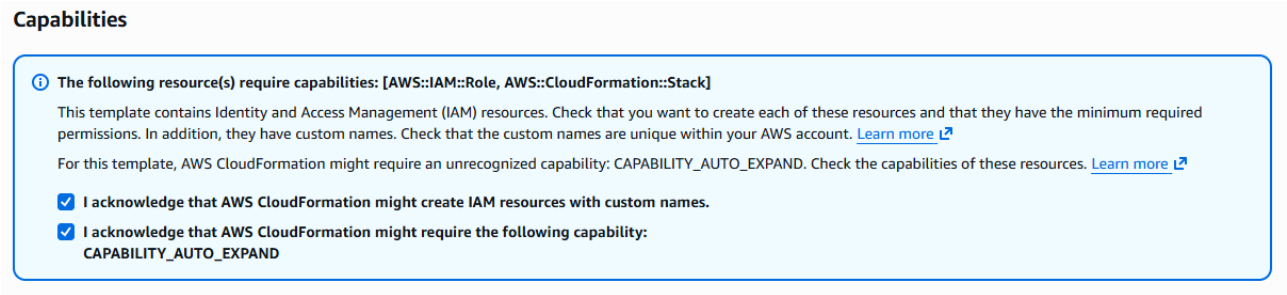
- **Role creation preference:** Select **Create New Roles** if installer should create IAM Roles for ALinkSever execution or **Use Existing Roles** if the roles with the below names have already been created before.

- **Task Role Name:** Name of the container task role.
- **Task Execution Role Name:** Name for the container execution role.

d) Application Load Balancer Settings

- CloudFormation template creates new Application Load Balancer (ALB) for routing ALinkServer traffic. HTTPS/SSL termination can be configured in the EC2 ALB configuration after creation. Specify the following parameters:
- **ALB Name:** Specify the Application Load Balancer name
- **ALB Target Group Names:**
  - Specify target group name for worker tasks (port 8000)
  - And target group name for admin tasks (port 8001).
- ALB settings is the last section of the ECS deployment configuration – all further configurations are standard AWS CloudFormation stack deployment parameters.

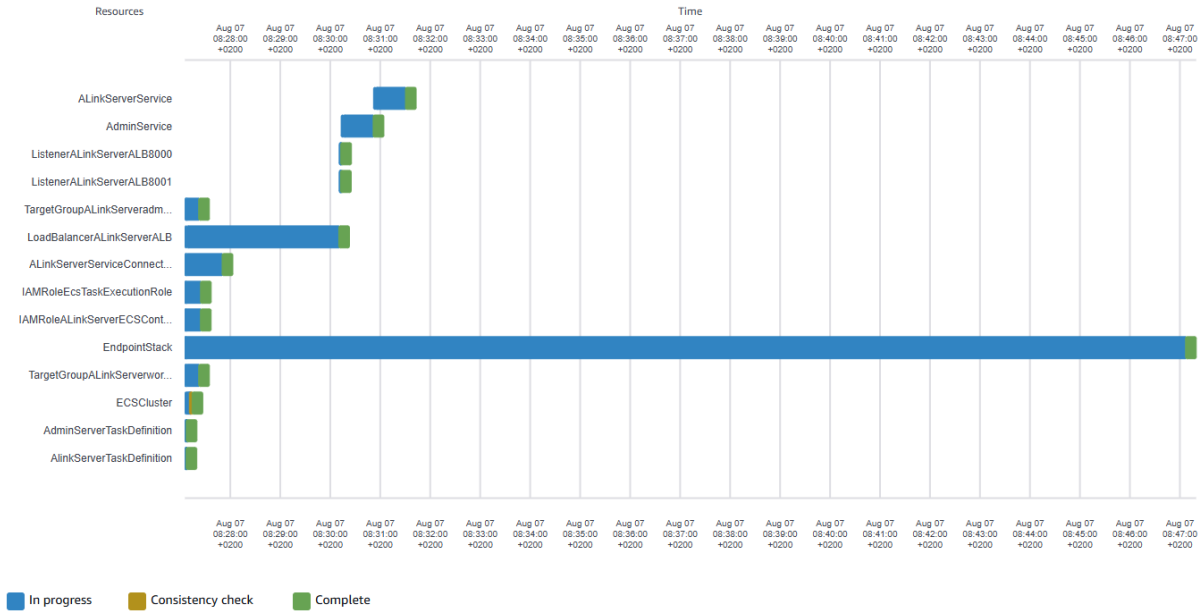
4. At the bottom of the page acknowledge that the CloudFormation script will create IAM roles as specified above. Endpoints creation is provided in a separate CloudFormation substack. Because this substack is referenced in the main stack you'll need to ackowledge the „CAPABILITY_AUTO_EXPAND“ usage.



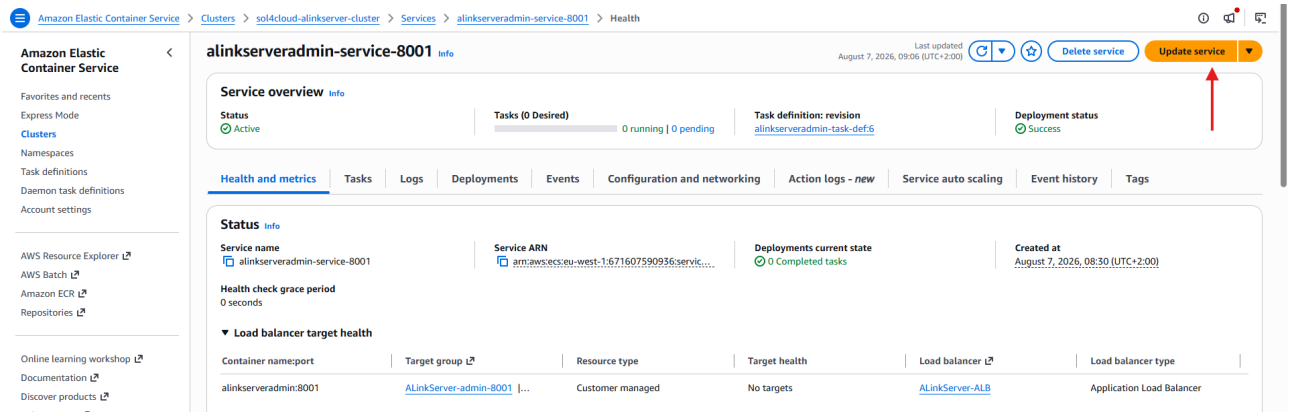
5. Click „Create stack“ button and wait for the Stack to be created.

   Note: Due to the long duration of creation of us-east-1 „Marketplace“ endpoints the Stack creation can take up to 20 minutes – see screenshot example below.

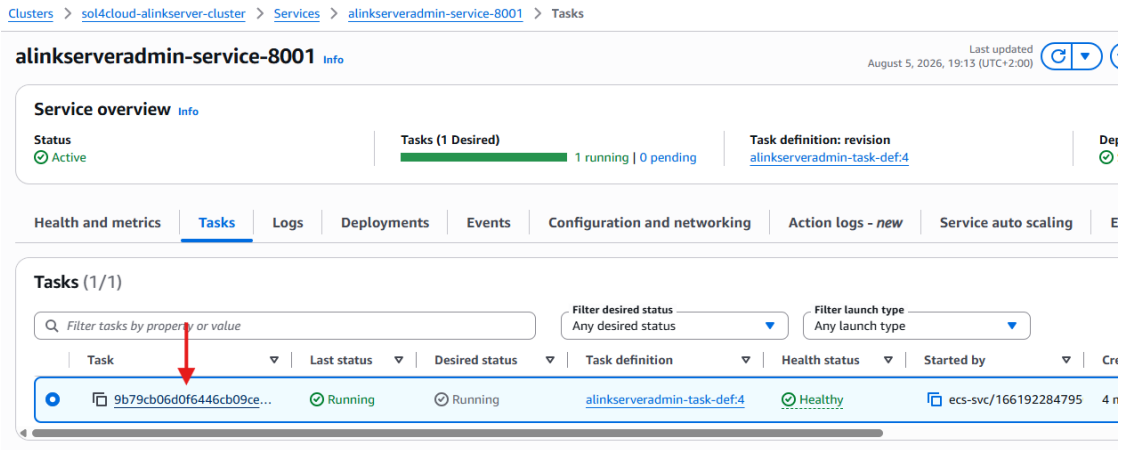
6. Upon completion switch to the Resources tab to see a list of all resources created using the CloudFormation Stack script.



7. Upon CloudFormation completion the Endpoints, ALinkServer ECS Cluster and Application Load Balancer have been succesfully deployed.
8. Navigate to the newly created ECS Cluster (via the **Resources** tab in the CloudFormation console).
9. Select the **`alinkserveradmin-service-8001`** service and click **Update service**



10. In the **Update alinkserveradmin-service-8001** page specify 1 in the **Desired tasks** field and click **Update** at the bottom of the page. ECS Server will start one instance of the admin task.
11. Wait for the task to start (it takes approximately 2 minutes). Verify that the task status displays **Last status: Running** and **Health status: Healthy** as shown below:



12. Open the configuration page of the newly created Application Load Balancer (e.g. by going to the CloudFormation **Resources** tab in the newly created CloudFormation Stack and clicking the ALB link next to „LoadBalancerALinkServerALB").

13. Open a web browser and navigate to the ALB DNS name on port `8001` (e.g., `http://<ALB-DNS-Name>:8001`). (In this step we assume that your local machine is connected to AWS network via AWS Client VPN, AWS Direct Connect, or VPC Peering to access ALB with internal-only AWS hostname).

14. Now that your ALinkServer Admin instance is running continue to chapter First start of the ALinkServer deployment for a quick guide on first server start and initial ALinkServer configuration.

    Note: ECS CloudFormation template utilizes FARGATE compute for ALinkServer ECS cluster. If you prefer to use EC2 instances managed by Auto Scaling Group than you'll need to change the following:

    1. *Create new ECS cluster with Infrastructure configuration option set to **Fargate and Managed Instances** or to **Fargate and Self Managed Instances***
    2. *Set EC2 Launch Template **Token hop limit** parameter to 2 to ensure that AlinkServices are able to discover AWS services*
    3. *Switch **Network mode** from default `awsvpc` to `bridge` mode in the ECS Task Definitions*

    Warning: Deleting the Stack in CloudFormation deletes all AlinkServer configuration objects created above – DeletionPolicy of all infrastructure resources in the CloudFormation template is set to „Delete"

EKS HELM Deployment

Use this option if you don’t want to run ALinkServer in ECS and prefer to run ALinkServer in your own EKS cluster instead. AWS Marketplace EKS Deployment offers two **Delivery Methods**:

- **Container image:** This is AWS „expert option“. If you want to create completely custom EKS deployment, with your own HELM script than choose this option. URL/ARN of the ALinkServer images can be found in the **Launch** section of the page.
- **HELM chart:** Helm chart deployment simplifies the first start of the ALinkServer by providing Kubernetes, EKS and HELM scripts. This chapter provides the details about each deployment step

Prerequisites

HELM chart deployment assumes that you have the following prerequisites in place:

- `kubectl, aws cli, helm` installed to run the deployment commands
- Running instance of EKS cluster
- EKS cluster has a configured aws-load-balancer-controller (see AWS documentation on how to install AWS load balancer controller to EKS: https://docs.aws.amazon.com/eks/latest/userguide/lbc-helm.html)

Environment setup

1. Create namespace for Sol4Cloud AlinkServer installation

   ```
   kubectl create namespace sol4cloud
   ```

2. Pods running in the EKS cluster need authorizations to access AWS resources (see chapter Components of AWS ALinkServer deployment for more details) therefore custom IAM service account needs to be created before installation. Copy the eksctl command from the deployment page. Replace cluster name placehoder with the name of your EKS cluster.

   ```
   eksctl create iamserviceaccount \
       --name alinkserver-sa \
       --namespace sol4cloud \
       --cluster <REPLACE_WITH_YOUR_EKS_CLUSTER_NAME> \
       --attach-policy-arn arn:aws:iam::aws:policy/AWSMarketplaceMeteringFullAccess \
       --attach-policy-arn arn:aws:iam::aws:policy/AWSMarketplaceMeteringRegisterUsage \
       --attach-policy-arn
   arn:aws:iam::aws:policy/service-role/AWSLicenseManagerConsumptionPolicy \
       --attach-policy-arn arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess_v2 \
       --attach-policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess \
       --attach-policy-arn arn:aws:iam::aws:policy/SecretsManagerReadWrite \
       --approve \
       --override-existing-serviceaccounts
   ```

3. Same as in the case of ECS deployment above it is recommended to deploy VPC Endpoints for all network traffic of the ALinkServer to remain AWS internal instead of traversing

public internet. Switch to the **Delivery Method** -> **Container image** and click the **"Quick create" VPC Endpoints** link to jump to CloudFormation page and install 9 required VPC Endpoints. Return to the HELM chart page to continue installation.

ALinkServer EKS deployment

1. Login to AWS Marketplace registry

```
aws ecr get-login-password --region us-east-1 | helm registry login \
    --username AWS \
    --password-stdin 709825985650.dkr.ecr.us-east-1.amazonaws.com
```

2. Create temporary directories for unpacking the HELM chart, pull the HELM chart from the AWS Marketplace registry and unpack it with the tar command

```
mkdir awsmp-chart && cd awsmp-chart

helm pull oci://709825985650.dkr.ecr.us-east-1.amazonaws.com/sol4cloud/sol4cloud-alinkserver-chart --version 0.1.58

tar -xf ./*.tgz && rm -f ./*.tgz
```

3. Run the below `helm install` command to install ALinkServer to your EKS cluster.
    - Replace the aws.region placeholder with the target region for your installation
    - Specify HTTPS certificaate ARN for the EKS Load Balancer to respond on secure HTTPS only or leave the value blank for HTTP/port 80 traffic.

```
helm install alinkserver ./* \
    --namespace sol4cloud \
    --set aws.region=<SPECIFY_TARGET_AWS_REGION> \
    --set ingress.scheme=internal \
    --set ingress.certificateArn=
```

4. Check the server startup progress in your EKS cluster. Approximately 3 minutes after installation the Application Load Balancer will be up and running. Find the ingress URL of your EKS cluster and open the ALinkServer Admin URL by going to:

    `http://<YOUR_ALB_DNS_NAME>/admin`

5. Jump to chatper First start of the ALinkServer deployment to complete the initial server configuration and wait for the server to create DynamoDB configuration table and SecrectsManager configuration. Upon completion login to Admin UI and go to **Repositories** menu and create at least one SAP Content Repository

6. Return to your EKS cluster management shell and restart the ALinkServer Worker pods using `kubectl rollout restart deployment -n sol4cloud` to load the newly created server and repository configuration from ALinkServer configuration.

    Worker Pods do not have to be restarted if administrators have only created new SAP content repositories. Worker Pod will try to fetch new repository configuration from ALinkServer configuration object if the repository ID didn't exist at ALinkServer worker startup.

7. Open the URL of the worker server on your load balancer DNS address:

   ```
   http://<YOUR_ALB_DNS_NAME>
   ```

8. You will see the list of repositories configured in this instance of ALinkServer. Click the serverInfo page link on the page to check whether ALinkServer worker installation is working as expected – you should see a simple HTML page with server config details and repository details.

9. Now you're ready to configure ArchiveLink content repository in SAP. Jump to page 29 and complete the SAP configuration in transaction `OAC0` to start using the installation for SAP data and document archiving.

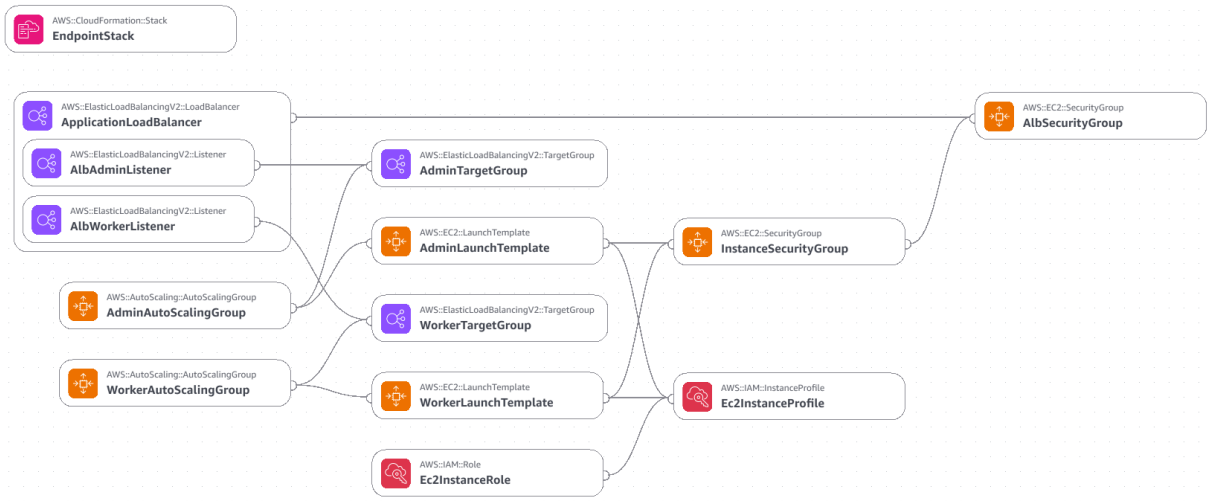
Amazon Machine Image EC2 deployment

Use this option if signed up for the AMI EC2 product option and want to deploy the product. Open the AWS Marketplace **Manage subsciptions** page. At the top of the page you can review high level Usage instructions, Write review and Launch CloudFormation stack. Click the **Launch CloudFormation stack** to start the deployment.

1. On the Launch page review the Setup details – Version of ALinkServer, Release notes and launch and connection instructions at the bottom of the page. You can also view the CloudFormation template in CloudFormation Designer (or in the next step in the CloudFormation template studio).

2. When you're ready to deploy ALinkServer then select target Region and click the **Launch with CloudFormation** button to start the standard 4 step CloudFormation Create stack wizard.

3. On the Create stack page you can click the Open in template studio button to view the CloudFormation template objects and/or modify some of the default settings. AMI EC2 deployment creates AWS objects as shown on the resource diagram below (see chapter Components of AWS ALinkServer deployment above for details about ALinkServer AWS configuration objects deployed during installation):



4. When you finished reviewing the CF template click **Next** to jump to **Specify stack details** step.
5. Populate the details on the **Specify stack details** page
   a) Provide Stack name
   b) In the Parameters section select **Target VPC** from the list of available VPCs
   c) Specify at least two **Target Subnets** across different availability zones within selected VPC. Some Target Subnets don't allow installation of VPC Endpoints – see Target Subnets section in the ECS deployment guide above for instructions how to identify subnets where VPC Endpoint installation is allowed
   d) Do not change the Marketplace specified **Amazon Machine Image ID**
   e) S3 and DynamoDB Endpoints need VPC Route table for its configuration. Open a second browser window with AWS Console and copy ID of your VPC Route Table to the **VPC Route table for use with S3 and DynamoDB Endpoints** parameter.
   f) Select whether VPC Endpoints should be created as part of the installation. Default option is Create VPC Endpoints. VPC Endpoints configuration file is nested in the main CloudFormation script. Alternatively choose the option **No VPC Endpoints, custom network configuration** if you intend to configure NAT networking or prefer to manually configure VPC Endpoints
   g) Select instance type for the ALinkServer Admin instance.
   h) Select instance type for the AlinkServer „Worker" instances (4 vCPU instances with 8GB RAM).
   i) Specify **Min, Max** and **Desired** Auto Scaling Group (ASG) capacity. For initial deployment leave Min and Desired set to 0 (zero) since ALinkServer Admin first needs to create initial configuration objects before first „Worker" server start.

j) In the **Role Creation Preference** list box select whether the CF script should create the IAM Role that is necessary for ALinkServer functionality (role that provides ALinkServer processes access to AWS S3, STS, DynamoDB, Logging and Secrets Manager). If the role has been created before select **Use Existing Role** otherwise keep **Create New Role** and specify the IAM Role name in the next box.

k) If needed specify **Optional bucket names prefix**. If left empty ALinkServer will be able to use any bucket name within the account and region context as target for SAP repository configuration otherwise AWS IAM Role will enforce that only buckets with the prefix name can be used for repository configuration.

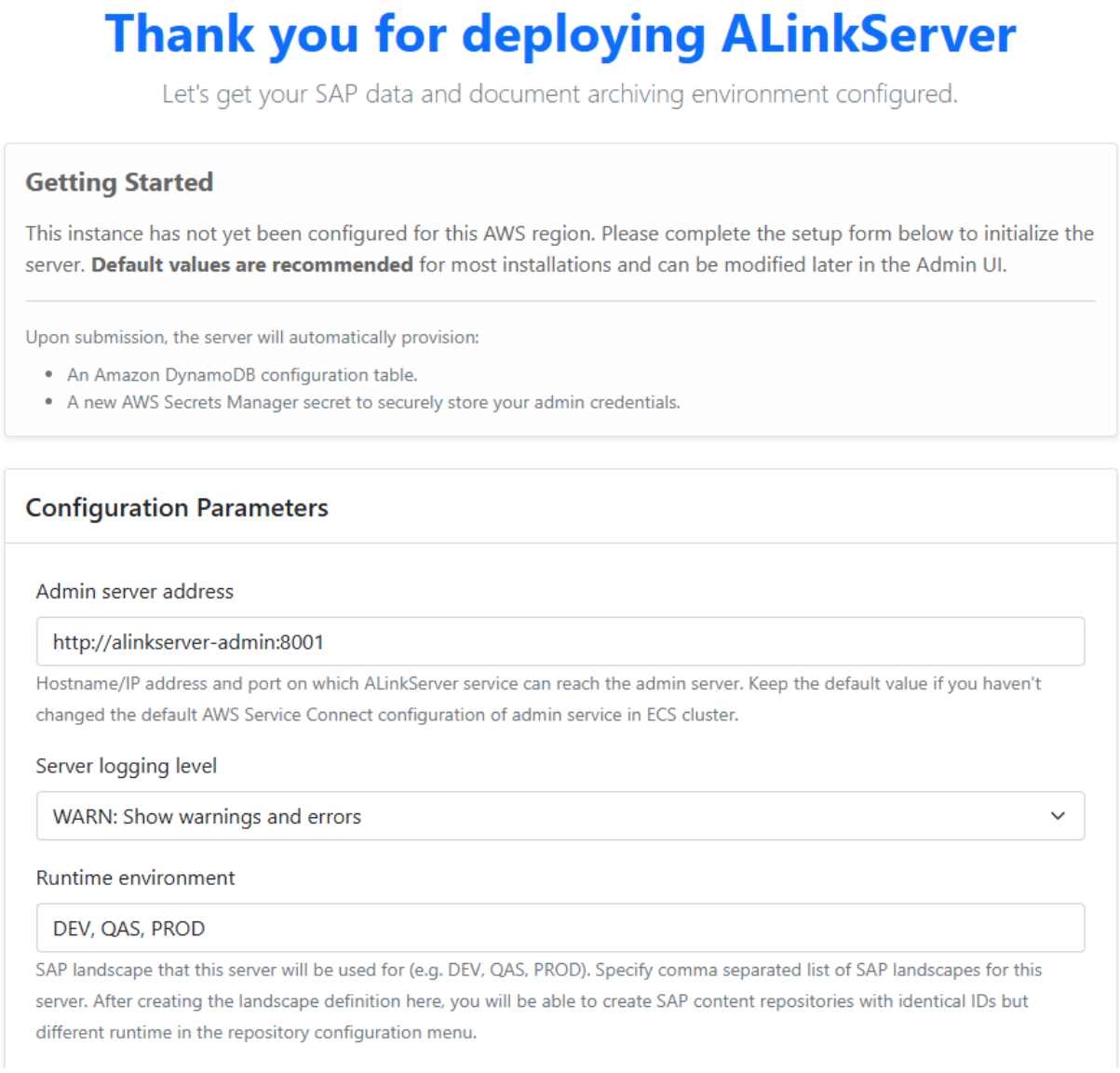
l) End of the Specify Stack Details page contains mandatory AWS Marketplace parameters – AMI ID, Bucket name, Bucket region and Key Prefix. These parameters are used to launch the nested VPC Endpoints Stack and shouldn't be changed.

6. After populating all parameters click **Next** to jump to **Configure stack options** page
7. At the bottom of the **Configure stack options** page check both check boxes to acknowledge that the CF template will create the necessary IAM Role and needs CAPABILITY_AUTO_EXPAND capability to install the linked substack that installs VPC Endpoints.
8. Click **Next** to jump to the final **Review and create** page
9. After you've reviewed the summary of the deployment details on the page jump to the bottom of the page and click **Submit** button to start the deployment.
10. Depending on your target region and whether you've chosen to install VPC Endpoints the CF deployment finishes within 4 minutes or up to 20 minutes (non us-east-1 deployment with VPC Endpoints installation). It may need to be necessary to wait a few minutes after CF deployment for the AMI image to fully start before you continue with ALinkServer configuration. When Admin Server fully started switch to **Outputs** tab of the executed CloudFormation Stack and click the **AdminURL** to open the ALinkSever Admin UI page (this step assumes that you're connected to AWS network via AWS Client VPN, AWS Direct Connect, or VPC Peering to access ALB with internal-only AWS hostname). Jump to chapter First start of the ALinkServer deployment to complete initial deployment of ALinkServer.

First start of the ALinkServer deployment

This chapter assumes that you've deployed your ALinkServer infrastructure (ECS, EKS or AMI deployment) as described in the chapter Deploying ALinkServer and your ALinkServer Admin server is up and running

1. Upon first initial start ALinkServer Admin will display a welcome page for initial ALinkServer configuration. Initial ALinkServer Admin server configuration screen looks as shown below:



2. Populate the initial configuration parameters:

a) Change the **Admin server address** if you've changed the Service Connect Namespace configuration during ECS CloudFormation deployment or manually during CloudFormation installation, otherwise keep the default value. For AMI EC2 deployment replace the admin server name with the DNS name of the internal Application Load Balancer. Use server DNS name and port only with no trailing slash.

b) Select **Server logging level**. The same log level applies to Admin service and main ALinkServer worker service.

c) If needed update the **Runtime environment** parameter with comma separated list of SAP Runtime environments. Runtime parameter is used as part of the ALinkServer URL and it will enable SAP administrator to use the same SAP Content Repository ID across multiple SAP landscapes (DEV, QA, PROD, …) with just one ALinkServer deployment.

d) If needed specify the **Default document protection flag**. Combination of c, r, u, d (create, read, update, delete) specifying whether the server should enforce signature checking in case SAP didn't pass the docProt parameter during create request (e.g. optionally set the value to "ud" to enforce signatures for update and delete operations). See SAP ArchiveLink specification for details.

e) Leave the **Optional server parameters** box empty unless instructed otherwise.

f) Specify admin user name and password.

3. Click the **Create initial server configuration** button. This initial configuration takes approximately 1 minute - wait for the process to complete. Once complete, ALinkServer Admin UI will show a login screen. During initial configuration ALinkServer Admin process will:

   a) create DynamoDB configuration table – configuration table is used to store sever configuration data (server log level, http address of the admin server, definition of SAP runtimes etc.) and SAP content repositories configuration data. Each SAP content server repository configuration specifies the repository runtime, repository ID, repository description and most importantly the name of the S3 bucket where this SAP repository's data will be archived. Do not manually change the configuration table items – use the ALinkServer Admin UI to do that.

   b) store the admin user details in AWS Secrets Manager. Secret holds the user names and encrypted passwords of administrator users for ALinkServer Admin UI. Do not change or delete the content of the Secret.

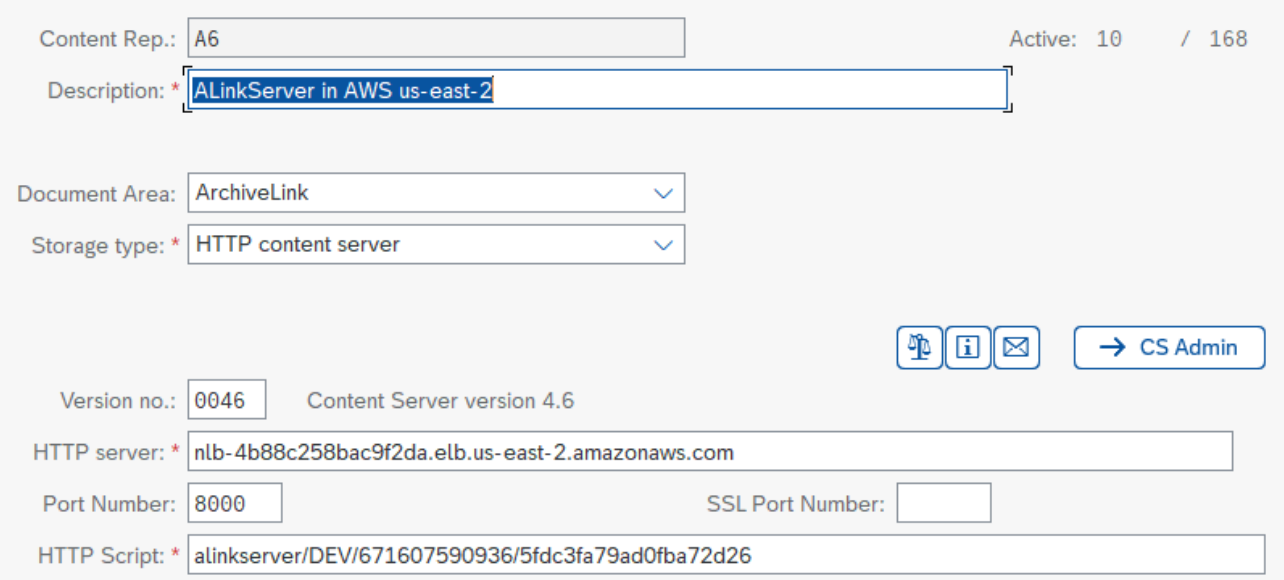
      If you accidentally deleted it you'll not be able to login to ALinkServer. In such case use API `aws secretsmanager restore-secret --secret-id Sol4Cloud_alinkserver_users` to restore accidentally deleted secret.

4. Login to ALinkServer Admin UI and go to **Repositories** menu at the top of the page. Click **Create new repository** and configure at least one SAP content repository so that you can test SAP data or document archiving in the next steps (see Menu „Repositories" paragraph in the next chapter for description of the parameters).

5. Return to your deployment tool and locate the ALinkServer „worker“. Start the worker process:
   - In ECS Cluster console update **`alinkserver-service-8000` Desired tasks** to `1` (or more). Wait for the service to start and report a **Healthy** status.
   - When using HELM deployment restart the Worker pods by running the command `kubectl rollout restart deployment -n sol4cloud`
   - For AMI EC2 deployment go to Worker Auto Scaling Group and start at least one worker instance
6. Once running, open `http://<ALB-DNS-Name>:8000` to view the ALinkServer worker main page with the list of configured repositories.

   Congratulations, now the ALinkServer deployment is complete. Align with your SAP administrator and continue with the next step to configure the SAP Content Repository.
7. Login to the target SAP system and create new SAP content repository in transaction `OAC0`.
   a) Set **HTTP Server** to the ALB IP/DNS name (do not include „http://“, port or last slash in the http server parameter).
   b) Set **Port Number** to `8000` (or use the port configured during ALinkServer deployment)
   c) Copy-paste the exact **HTTP Script** path as shown in the ALinkServer Admin UI or in the root page of the ALinkServer worker server (running on port 8000).

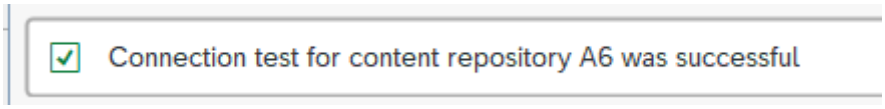
8. Resulting OAC0 configuration looks like the example below:



9. Save SAP content repository and test connection to ALinkServer repository from SAP by clicking the **Test connection** button



If ALinkServer has been configured correctly and the network route is open for the traffic you should see a status message at the bottom of the `OAC0` screen as shown below:



10. Send SAP certificate from the repository configuration screen by clicking the **Send certificate** envelope button. You should see a message „Certificate was sent“ at the bottom of the screen.



11. Congratulations! Your SAP content repository is ready for use. Use the newly configured SAP content repository in SAP data archiving or SAP document archiving scenarios. Reach out to your SAP team to configure the archiving scenarios. List below is just a high level example:

    For SAP document archiving:

    a) Create document type in SAP transaction OAC2

    b) Create link between SAP business object, document type and the above created content repository in SAP transaction OAC3

    c) Open the SAP transaction for your business object (e.g. FB03 for viewing invoices) and archive a test document from „Save business document“ menu in „Services for Object“

ALinkServer Configuration

All ALinkServer configuration is being done using ALinkServer Admin service (running by default on port 8001 in ECS/EKS Cluster and EC2 ALB). Main page of the ALinkServer Admin server provides an overview of currently running instance (AWS account id, region, server version etc.). Table on the main page shows information collected from all running ALinkServer worker services, namely the number and performance of requests processed by the ALinkServer service instances.

These are the main configuration menus available in the Admin UI:

Menu „Repositories“

Click the „Repositories“ menu for the list of currently configured SAP content repositories. You can see repository details by clicking on the repository entry in the repository list. From the details page you can edit repository details or delete the repository (deletion is only possible for empty repositories).

Click the „Create new repository“ button to create new content repository configuration in ALinkServer. For repository creation specify:

- SAP Repository ID, same ID needs to be used in SAP system in transaction OAC0
- Repository description
- Select one of predefined SAP runtime environment placeholders (e.g. DEV, TEST, PROD). New SAP runtime placeholders can be configured in the ALinkServer Server Configuration dialog.
- S3 Bucket Name: Name of the pre-existing target S3 bucket.

  Note: Buckets must be created prior to repository configuration – your AWS admin can activate selected enterprise policies (versioning, Object Lock/WORM, encryption).

  Important: Do NOT enable S3 Intelligent-Tiering Archive or Glacier auto-archiving options, as SAP ArchiveLink requires synchronous data access.
- If for any reason ArchiveLink signature verification in ALinkServer would fail (e.g. very old SAP instance with signature algorithm that is not supported by the ALinkServer implementation) than you can switch off signature verification routines for this content repository. Restart the ALinkServer worker service upon change of the parameter.

Upon repository configuration logon to SAP system and create the SAP repository configuration in transaction `OAC0`. Use ALinkServer Application Load Balancer hostname, port number based on your installation (ECS default is 8000, EKS default is 80/443) and SAP configuration HTTP Script value from the Repository detail page. Test connection and exchange SAP certificate as described in the previous chapter.

Menu „User administration“

Only admin users for ALinkServer Admin UI are maintained here – ArchiveLink protocol doesn’t authenticate users but instead relies on signing content repository requests for added security. In this menu you can create and delete administrator users or change existing user’s passwords.

If you’ve lost or forgotten all of the user accounts and passwords created before, you’ll need to reset all admin users:

1. Go to AWS Secrets Manager and display **Sol4Cloud_alinkserver_users** secret
2. Retrieve the Secret value and remove all admin user entries from the secrets json.
3. After that go to ALinkServer Admin UI and manually load `/admin/resetadmin` URL to create first admin user – this URL is only available if the server SecretsManager object exists but is empty. Use this user to login to ALinkServer Admin and create any additional users using the **User administration** menu

Menu „Server Configuration“

You can change the **Admin server address** in the Server Configuration – this is only to be used if you’ve changed the Service Connect Namespace configuration in ECS Server Services or changed the host name in EKS configuration, otherwise keep the default value.

ALinkServer Admin changes the used **Server logging level** after saving the server configuration. ALinkServer worker processes need to be restarted if you changed the log level in the Server Configuration screen – worker servers do not reload the server logging configuration after initial startup.

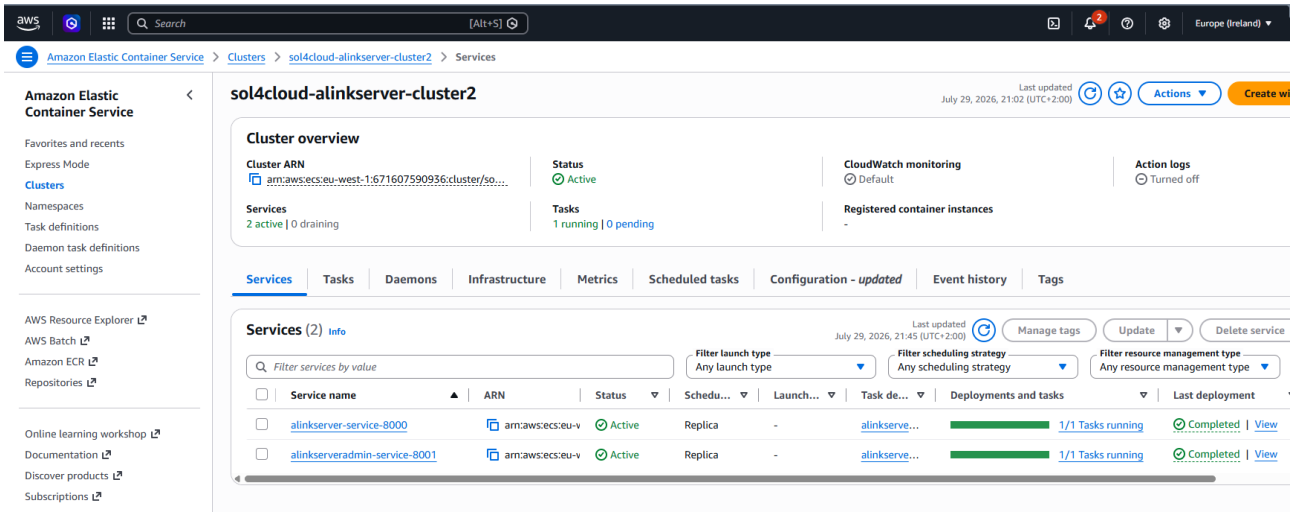
You can add additional SAP landscape/runtime shortcuts in the **Runtime environment** box – simply add additional value to the comma separated list. The runtime environment parameter is part of the `OAC0` **HTTP Script** URL in order to enable SAP administrators to keep the same SAP Content Repository IDs (and thus SAP customizing in `OAC3` and SARA transactions) across DEV, TEST and PROD landscapes while only changing HTTP Script value in each SAP environment.

Default document protection flag can be set in the ALinkServer for the case when SAP ArchiveLink create or mCreate request doesn’t pass „docProt“ URL parameter defining the document protection. Possible values are any combination of c, r, u, d (create, read, update, delete). If e.g. „ud“ is set in the parameter then ALinkServer would enforce signature validation for update and delete ArchiveLink operations in case if SAP ommited to specify docProt parameter in the document create request.

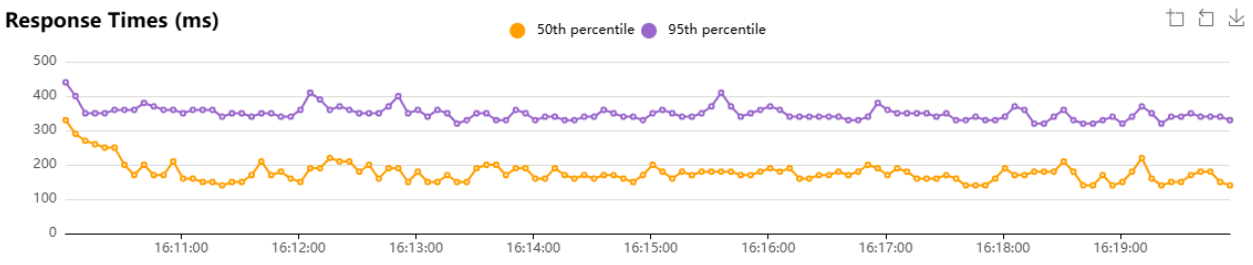
ALinkServer Administration

Administration of ECS deployment

ALinkServer services monitoring and administration happens through monitoring in Amazon Application Load Balancer and Amazon Elastic Container Services (ECS). ECS provides for each service in the cluster „Health and metrics“ overview, access to logs etc. Application logs are also accessible in AWS CloudWatch.



Number of started ALinkServer „worker“ services alinkserver-service-8000 depends on expected system load e.g. whether the system is to be used for backend data archiving only or whether it will be for frontend document archiving and display by end users as well. In our performance tests (100 concurrent users archiving documents, reading document information, reading content and deleting documents) we’ve seen average response time of 182ms (95th percentile response time was 350ms) with 45 requests per second processed when running 3 worker tasks (@4 vCPUs).



Important: Due to current implementation limitation, please, ensure that only one task of alinkserveradmin-service-8001 is started in the ECS cluster. Admin service is only used for ALinkServer configuration, ALinkServer statistics collection and AWS metering – there is no direct user performance impact on the worker tasks.

Administration of EKS deployment

Default HELM chart provided with the installer installs three main EKS objects used by ALinkServer application:

- ALinkServer Admin container – one 1 vCPU instance
- ALinkserver Worker containers – two 4 vCPU instances are started by default
- Application Load Balancer

Use standard Kubernetes/EKS methods to monitor and scale the deployment. For backend SAP data archiving requirements the number of Worker instances can be reduced to one as each worker starts 8 threads to respond to incoming SAP ArchiveLink requests. For frontend document archiving requirements auto-scaling can be configured to automatically respond increased load at peak times.

Administration of AMI EC2 deployments

Provided CloudFormation template deploys ALinkServer „Worker“ and Admin in minimum two EC2 instances, each managed by own Auto Scaling Group. Worker ASG can be configured as required by the expected ArchiveLink traffic. Both instances are served via Application Load Balancer. ALB configuration can be adjusted as required e.g. by customizing HTTPS/443 termination.

Known product limitations

Single threaded admin server

The Admin UI uses Flask-Login session management, which stores session state within the active worker process memory. Consequently, the Admin container is configured to run as a single-threaded process.

Signature verification API

ALinkServer successfully unpacks, decrypts, and verifies the incoming SAP signature payload against requested URL parameters. However, the low-level crypto verification API call may register a warning in the logs. Because the payload validation checks out, ALinkServer safely continues request processing

Q&A

Q: Are there any document size limits?

A: To maximize throughput and avoid running container storage overhead, ALinkServer only streams data in-memory without staging files to disk. Therefore it is recommended to limit document/data sizes to approximately **100 MB per file** (also a common threashold in SAP data archiving).

Q: Does ALinkServer support HTTPS/terminate HTTPS connections? Can it communicate with SAP over HTTPS?

A: Use AWS technologies to configure HTTPS communication between the SAP instance and ALinkServer. An example configuration could be:

1. Request SSL certificate in AWS Certificate Manager or import your own certificate
2. Create an HTTPS Listener in ALinkServer Application Load Balancer and attach to it the above certificate
3. Adjust Security Group to allow HTTPS inbound traffic

Q: Is ALinkServer SAP certified?

A: Official SAP certification is planned for a future release. The application has been rigorously validated using standard SAP ArchiveLink test suites (such as report `RSCMST`) for protocol compliance (see a screenshot from the test execution below) and tested using load testing tools for operational stability.



Q: Can we trust that Sol4Cloud will guarantee long term availability of the ALinkServer product and thus access to our important SAP archived data and documents?

A: We understand that entrusting the ArchiveLink routing of your important corporate data to “unknown startup” can be considered high risk. We are committed to continuing the support and development of the solution (in fact next versions of ALinkServer for Microsoft Azure and Google GCP are currently in development). If the solution wouldn’t prove economically viable for Sol4Cloud in long term we’d publish our private development GitHub repositories to ensure that our customers will have access to the ALinkServer code and can run it without limitations.