

AzureDevops: How to configure your laptop as self-hosted-agent to run Azure DevOps jobs

We are going to explain how to set up **Jobs** that **run on machines** that you manage (your **laptop** or a **Cloud Virtual Machine**), where you **install agent software supplied by Microsoft**

For this purpose **Azure DevOps** provides a feature called "**self-hosted agents**" for exactly this purpose

Self-hosted agents allow you to **run build and deployment jobs** directly on **machines you manage**, rather than on **Microsoft-hosted agents**

This approach gives you **more control over the environment** in which your jobs run, including the ability to **customize the operating system, installed software, and hardware specifications**

Here's a step-by-step guide to setting up **self-hosted agents** in **Azure DevOps**:

1. Prepare the Machine

Ensure your target machine meets the requirements:

Supported operating system (Windows, macOS, Linux)

Sufficient hardware resources (CPU, memory, disk space)

Network connectivity to Azure DevOps services

Required software installed (e.g., development tools, SDKs)

2. Create a Personal Access Token (PAT) in Azure DevOps

You'll need a **PAT** to authenticate the agent with Azure DevOps:

We **Sign in** to your Azure DevOps organization

We go to **User settings** -> **Personal access tokens**

dev.azure.com/luiscoenenriquez/mysecondproject

Azure DevOps luiscoenenriquez / mysecondproject / Overview / Summary

mysecondproject

Overview

Summary

Dashboards

Wiki

Pipelines

About this project

Help others to get on board!

Describe your project and make it easier for other people to understand it.

+ Add Project Description

Project stats

Pipelines

50% Builds succeeded

Members 1

LE

Personal access tokens

We click New Token

Azure DevOps

Search

User settings luis coco enriquez

Personal Access Tokens

These can be used instead of a password for applications like Git or can be passed in the authorization header to access REST APIs

+ New Token

Active

Token name	Status	Organization	Expires on ↓
Git: https://dev.azure.com/luiscoenenriquez on the website. Code (Read & write)	Active		21/2/2024

We give it a descriptive **name**, choose an **expiration**, and select the **Agent Pools (read, manage)** scope under Agent Pools

We press on **Show all scopes** link and we select **Agent Pools (read, manage)**

Create a new personal access token



Name

myPATforAgent

Organization

luiscocoenriquez

Expiration (UTC)

30 days



21/2/2024



Scopes

Authorize the scope of access associated with this token

Scopes ☐ Full access

☒ Custom defined

Test management

Read, create, and update test plans, cases, and results

☐ Read

☐ Read & write

Packaging

Create, read, update, and delete feeds and packages

☐ Read

☐ Read & write

☐ Read, write, & manage

[Show all scopes \(29 more\)](#)

Create

Cancel

Then we click **Create** and copy the token

Create a new personal access token



Name

myPATforAgent

Organization

luiscoenriquez

Expiration (UTC)

30 days

21/2/2024



Scopes

Authorize the scope of access associated with this token

Scopes ☐ Full access

☒ Custom defined

Detection and alerting on security vulnerabilities in code

☐ Read

☐ Read & write

☐ Read, write, & manage

Agent Pools

Manage agent pools and agents

☒ Read

☒ Read & manage

[Show less scopes](#)

Create

Cancel

Store it securely; you won't be able to see it again

Success!



You have successfully added a new personal access token. Copy the token now!
myPATforAgent token

qhb7gmmchp3nky2yu2i7l



Warning - Make sure you copy the above token now.
We don't store it and you will not be able to see it again.

3. Download, Configure and Run the Agent

3.1. We create a Default Agent pool

In Azure DevOps, we go to **Project settings > Agent pools**

We choose the **default pool** or we create a new one

The screenshot shows the Azure DevOps web interface. The browser address bar displays the URL: `dev.azure.com/luiscoenriquez/mysecondproject/_settings/agentqueues`. The breadcrumb trail indicates the path: `luiscoenriquez / mysecondproject / Settings / Agent pools`. The left sidebar shows the 'Project Settings' menu, with 'Agent pools' selected under the 'Pipelines' section. The main content area displays the 'Agent pools' page, which includes a table with the following data:

Name	Queued jobs	Running jobs
Azure Pipelines		
Default		

We click **New agent** and we select the operating system of your target machine

Project Settings
mysecondproject

General

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards
- Pipelines**
- Agent pools
- Parallel jobs
- Settings

Jobs Agents Details Security Approvals and checks Analytics

Name	Project	Agent	Queued	Wait time	Duration
Job 1 20240122.2 luiscoco.Books_API	mysecondproject	LUISCOCOENRIQUE	Today at 14:03	<1s	4m 12s

Update all agents **New agent**

3.2. We download the agent package

Get the agent



Windows

macOS

Linux

x64

x86

System prerequisites

Configure your account

Configure your account by following the steps outlined [here](#).

Download the agent

Download



Create the agent

```
PS C:\> mkdir agent ; cd agent
PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ;
[System.IO.Compression.ZipFile]::ExtractToDirectory("$HOME\Downloads\vsts-agent-win-x64-3.232.3.zip", "$PWD")
```

Configure the agent [Detailed instructions](#)

```
PS C:\agent> .\config.cmd
```

Configure the agent [Detailed instructions](#)

```
PS C:\agent> .\config.cmd
```

Optionally run the agent interactively

If you didn't run as a service above:

```
PS C:\agent> .\run.cmd
```

That's it!

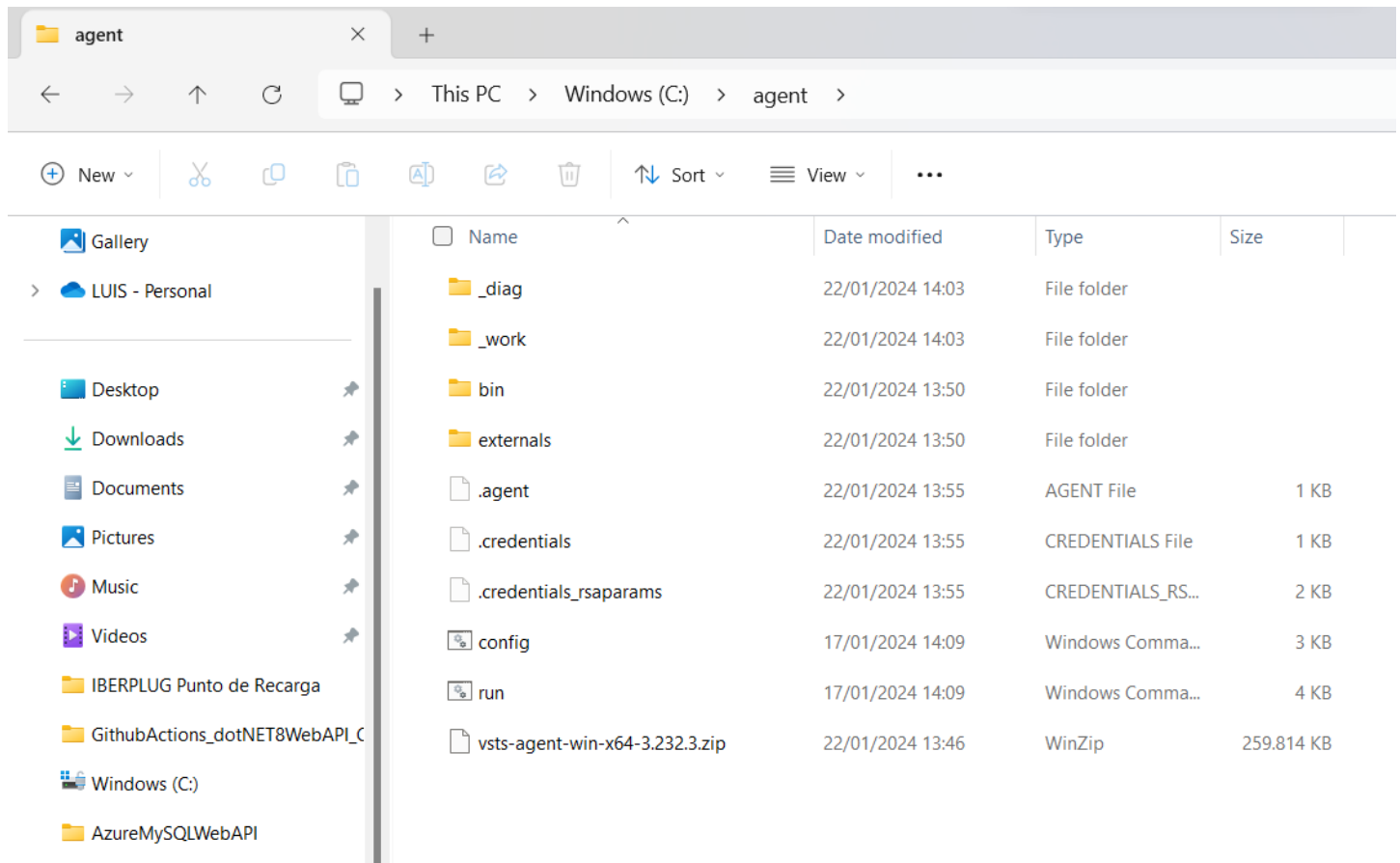
[More Information](#)

We run PowerShell and we execute the following commands:

3.3. We create the agent

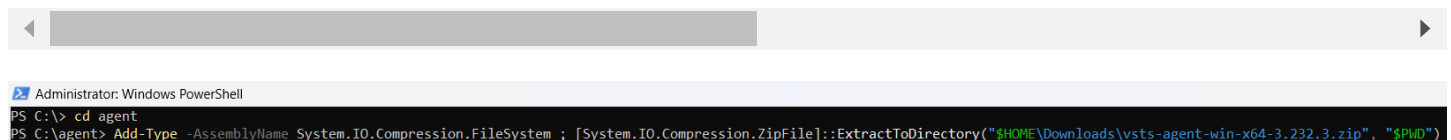
```
PS C:\> mkdir agent ; cd agent
```

We move the downloaded agent ZIP file from the **Downloads** folder to the in the **agent** directory



We create the agent from the ZIP file

```
PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ; [System.IO.Compression.
```

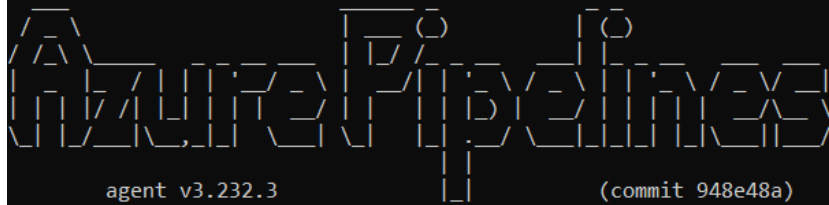


3.4. We configure the agent

```
PS C:\agent> .\config.cmd
```



```
PS C:\agent> .\config.cmd
```



```
>> Conectar:
```

```
Escribir dirección URL del servidor > https://dev.azure.com/luiscoenriquez/
```

```
Entrar tipo de autenticación (presione Entrar para PAT) > 6znuwv53z6bx6aelmaabi75utl4xg5qadcsrz56ndypjg33wpja
```

```
Escriba un valor válido para tipo de autenticación.
```

```
Entrar tipo de autenticación (presione Entrar para PAT) > 6znuwv53z6bx6aelmaabi75utl4xg5qadcsrz56ndypjg33wpja
```

```
Escriba un valor válido para tipo de autenticación.
```

```
Entrar tipo de autenticación (presione Entrar para PAT) >
```

```
Escribir token de acceso personal > *****
```

```
Conectando con el servidor...
```

```
>> Registrar agente:
```

```
Entrar grupo de agentes (presione Entrar para default) >
```

```
Entrar nombre del agente (presione Entrar para LUISCOENRIQUE) >
```

```
Examinando las capacidades de la herramienta.
```

```
Conectando al servidor.
```

```
El agente se agregó correctamente
```

```
Probando la conexión del agente.
```

```
Entrar carpeta de trabajo (presione Entrar para _work) >
```

```
2024-01-22 12:55:35Z: Configuración guardada.
```

```
Entrar ¿Ejecutar el agente como servicio? (S/N) (presione Entrar para N) >
```

```
Entrar ¿Configurar el inicio de sesión automático y ejecutar el agente al iniciar? (S/N) (presione Entrar para N) >
```

3.5. We run the agent

```
PS C:\agent> .\run.cmd
```

```
PS C:\agent> .\run.cmd
Examinando las capacidades de la herramienta.
Conectando al servidor.
2024-01-22 12:59:44Z: Escuchando trabajos
2024-01-22 13:03:48Z: Ejecutando el trabajo: Job
2024-01-22 13:08:02Z: el trabajo Job se completó con el resultado: Succeeded
```

4. Use the Self-hosted Agent in Your Pipelines

When defining a pipeline (YAML or through the UI), specify the pool where your self-hosted agent resides:

```
pool:
  name: MyPool # Replace with your agent pool name
```

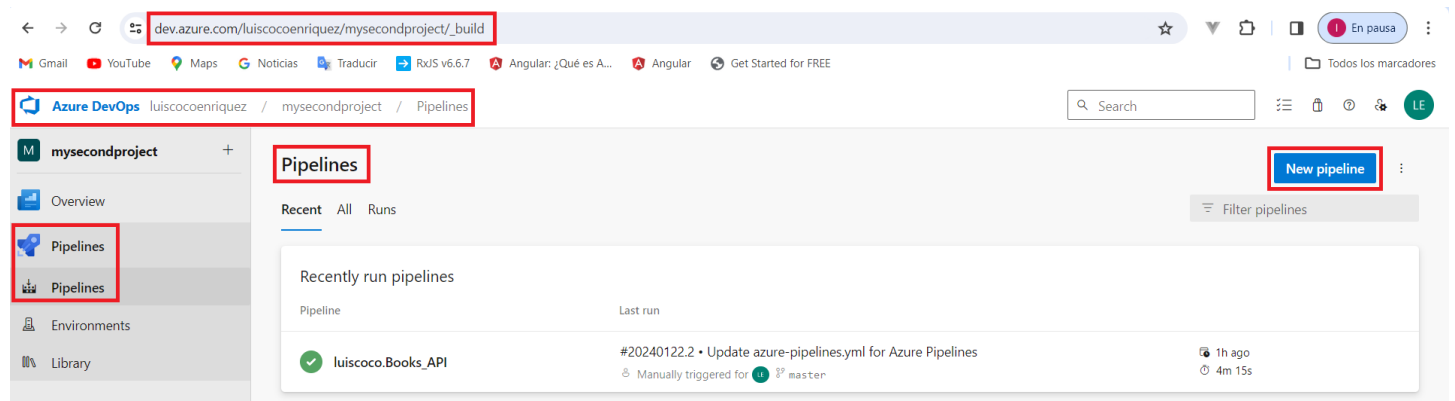
For example in our case we created a Default agent:

```
pool:  
  name: 'Default'
```

In our case see the Azure DevOps Pipeline configuration yaml file:

```
trigger:  
- main  
  
# Modificar aquí para usar un agente autohospedado  
pool:  
  name: 'Default' # Asegúrate de que este sea el nombre de tu grupo de agentes autohospedados  
  
variables:  
  solution: '**/*.sln'  
  buildPlatform: 'Any CPU'  
  buildConfiguration: 'Release'  
  
steps:  
- task: UseDotNet@2  
  inputs:  
    version: '8.x'  
    packageType: 'sdk'  
  
- task: DotNetCoreCLI@2  
  inputs:  
    command: 'restore'  
    projects: '**/*.csproj'  
    feedsToUse: 'select'  
  
- task: DotNetCoreCLI@2  
  inputs:  
    command: 'build'  
    projects: '**/*.csproj'  
    arguments: '--configuration $(buildConfiguration)'  
  
# Optional: Add steps for running tests here  
  
- task: DotNetCoreCLI@2  
  inputs:  
    command: 'publish'  
    publishWebProjects: true  
    arguments: '--configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)  
    zipAfterPublish: true  
  
- task: PublishBuildArtifacts@1  
  inputs:  
    PathToPublish: '$(Build.ArtifactStagingDirectory)'  
    ArtifactName: 'drop'  
    publishLocation: 'Container'
```

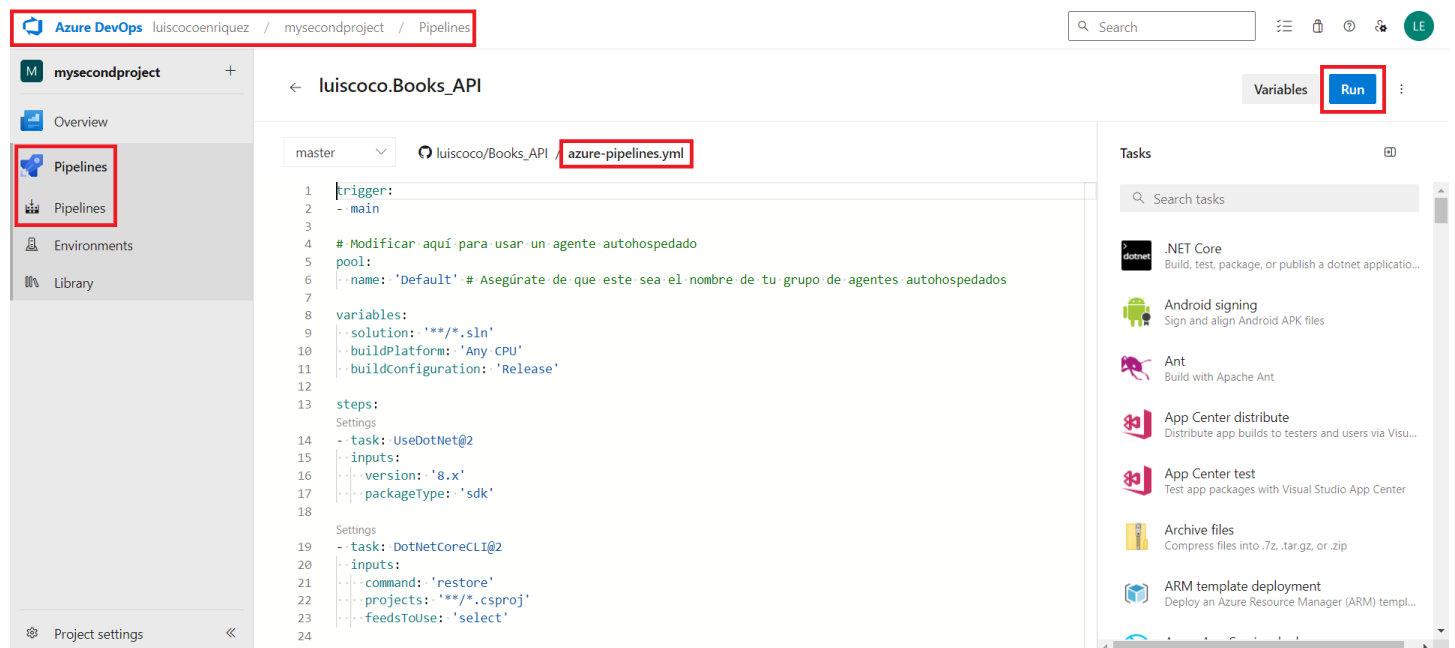
We create a New Pipeline



The screenshot shows the Azure DevOps web interface. The browser address bar displays `dev.azure.com/luiscocoenriquez/mysecondproject/_build`. The left sidebar shows the project 'mysecondproject' with the 'Pipelines' tab selected. The main content area shows the 'Pipelines' view for 'mysecondproject'. Under the 'Recent' tab, there is a table titled 'Recently run pipelines' with the following data:

Pipeline	Last run
luiscoco.Books_API	#20240122.2 • Update azure-pipelines.yml for Azure Pipelines Manually triggered for master 4m 15s

We input the above yml file source code to configure the pipeline and we Run the pipeline

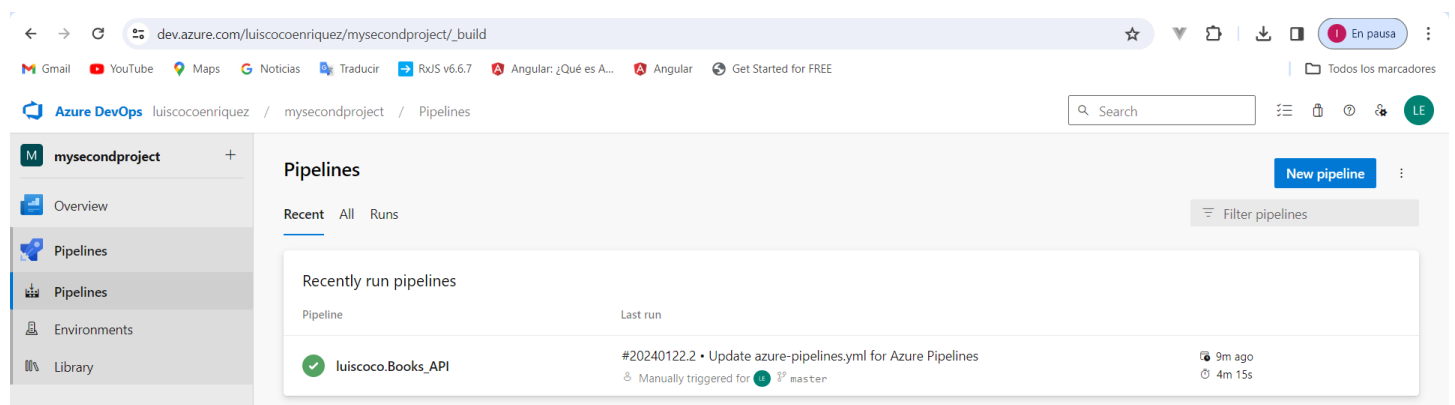


The screenshot shows the Azure DevOps web interface for the 'luiscoco.Books_API' pipeline. The 'Run' button is highlighted in the top right corner. The main content area shows the 'azure-pipelines.yml' file content, which defines a pipeline with a trigger on 'main', a pool named 'Default', and two steps: 'UseDotNet@2' and 'DotNetCoreCLI@2'.

```
1 trigger:
2   - main
3
4   # Modificar aquí para usar un agente autohospedado
5 pool:
6   - name: 'Default' # Asegúrate de que este sea el nombre de tu grupo de agentes autohospedados
7
8 variables:
9   - solution: '**/*.sln'
10  - buildPlatform: 'Any CPU'
11  - buildConfiguration: 'Release'
12
13 steps:
14   - task: UseDotNet@2
15     inputs:
16       - version: '8.x'
17       - packageType: 'sdk'
18
19   - task: DotNetCoreCLI@2
20     inputs:
21       - command: 'restore'
22       - projects: '**/*.csproj'
23       - feedsToUse: 'select'
```

Our jobs will now run on my laptop, where we intalled and setup the self-hosted agent

5. Verify the Job successfully run



The screenshot shows the Azure DevOps web interface. The browser address bar displays `dev.azure.com/luiscocoenriquez/mysecondproject/_build`. The left sidebar shows the project 'mysecondproject' with the 'Pipelines' tab selected. The main content area shows the 'Pipelines' view for 'mysecondproject'. Under the 'Recent' tab, there is a table titled 'Recently run pipelines' with the following data:

Pipeline	Last run
luiscoco.Books_API	#20240122.2 • Update azure-pipelines.yml for Azure Pipelines Manually triggered for master 4m 15s

Azure DevOps

luiscoenriquez / mysecondproject / Pipelines / luisco.Books_API

Search

LE

mysecondproject

Overview

Pipelines

Pipelines

Environments

Library

← luisco.Books_API

EditRun pipeline

Runs

Branches

Analytics

Description

Stages

✓ #20240122.2 • Update azure-pipelines.yml for Azure Pipelines

Manually triggered for master c0cd19a3

8m ago4m 15s

Azure DevOps

luiscoenriquez / mysecondproject / Pipelines / luisco.Books_API / 20240122.2

Search

LE

mysecondproject

Overview

Pipelines

Pipelines

Environments

Library

← Jobs in run #20240122.2

luisco.Books_API

Jobs

Job

Initialize job

Checkout luisco/Boo...

UseDotNet

DotNetCoreCLI

DotNetCoreCLI

DotNetCoreCLI

DotNetCoreCLI

PublishBuildArtifacts

Post-job: Checkout lui...

Finalize Job

✓ Job

View raw log

1 Pool: Default

2 Agent: LUISCOENRIQUE

3 Started: Today at 14:03

4 Duration: 4m 2s

5

6 Job preparation parameters

7 artifact produced

8 Job live console data:

9 Starting: Job

10 Async Command Start: DetectDockerContainer

11 Async Command End: DetectDockerContainer

12 Async Command Start: DetectDockerContainer

13 Async Command End: DetectDockerContainer

14 Finishing: Job