

lib.mkIf

????????????, ?????????? ? ?????????????????

`lib.mkIf` — это функция, используемая для **условного определения** конфигурационных опций в модулях.

`lib.mapAttrsToList condition definition`

Описание:

- `condition` — логическое выражение (true/false);
- `definition` — значение или набор атрибутов, которые будут применены, если условие истинно.

Возвращает `definition`, если `condition = true`, иначе возвращает особое значение "пустоты".

Использование:

- Позволяет включать и выключать части конфигурации на основе условий.

?????????

????????? ??????????

Исходный код:

```
{ config, lib, ... }:  
{  
  config = lib.mkIf (config.networking.hostName == "webserver") {  
    services.nginx.enable = true;  
    services.nginx.virtualHosts."example.com".root = "/var/www";  
  };  
}
```

?????????? ??????????

```

{ config, lib, ... }:
{
  config = lib.mkIf config.services.xserver.enable {
    sound.enable = true;

    hardware.pulseaudio = lib.mkIf config.services.pipewire.enable {
      enable = false;
    };
  };
};
}

```

???????????????? ? `mkMerge`

```

{ config, lib, ... }:
{
  config = lib.mkMerge [
    # Общие настройки
    {
      environment.systemPackages = with pkgs; [ vim wget ];
    }

    # Условные настройки для сервера
    (lib.mkIf (config.networking.hostName == "server") {
      services.openssh.enable = true;
      services.nginx.enable = true;
    })

    # Условные настройки для рабочей станции
    (lib.mkIf config.services.xserver.enable {
      services.xserver.desktopManager.gnome.enable = true;
      hardware.bluetooth.enable = true;
    })
  ];
};
}

```

????????? ?????????? ???????

```

{ config, lib, ... }:
{
  imports = [

```

```
(lib.mkIf config.virtualisation.docker.enable ./docker-extra.nix)
(lib.mkIf config.services.mysql.enable ./mysql-backup.nix)
];
}
```

???????? ?

```
{ config, lib, ... }:
let
  isProduction = config.networking.hostName == "prod-server";
  hasGPU = config.hardware.opengl.enable;
in
{
  config = lib.mkIf (isProduction && hasGPU) {
    services.tensorflow-serving.enable = true;
    services.cuda.enable = true;
  };
}
```

???????? ?

```
{ config, lib, ... }:
{
  options.myService = {
    enable = lib.mkEnableOption "My custom service";
    extraConfig = lib.mkOption {
      type = lib.types.str;
      default = "";
    };
  };
};

config = lib.mkIf config.myService.enable {
  systemd.services.my-service = {
    description = "My Service";
    wantedBy = [ "multi-user.target" ];
    script = ''
      echo "Starting my service"
      ${lib.optionalString (config.myService.extraConfig != "") ''
        echo "Extra config: ${config.myService.extraConfig}"
      ''}
  }
}
```

```
    '';  
};  
};  
}
```

????????????????????????????????

????????????????????????????????

```
# Правильно:  
config = lib.mkIf condition1 {  
    services.xyz = lib.mkIf condition2 {  
        enable = true;  
    };  
};  
  
# Неправильно (может вызвать ошибки):  
config = {  
    services.xyz = lib.mkIf condition1 {  
        enable = lib.mkIf condition2 true;  
    };  
};
```

??????? ? **mkOverride**

```
{ config, lib, ... }:  
{  
    config = lib.mkIf config.services.foo.enable {  
        services.foo.configFile = lib.mkOverride 90 "/etc/foo/custom.conf";  
        # Приоритет 90 (меньше = выше приоритет)  
    };  
}
```

???????????????????? ? **mkDefault** ? **mkForce**

```
{ config, lib, ... }:  
{  
    config = lib.mkIf config.networking.wireless.enable {  
        networking.networkmanager.wifi.backend = lib.mkDefault "iwd";  
        # Установит значение только если оно не было задано явно
```

```
};  
}
```

????????? ?????????????? ??????????

```
{ config, lib, pkgs, ... }:  
let  
  roles = {  
    web = config.node.role.web or false;  
    db = config.node.role.db or false;  
    cache = config.node.role.cache or false;  
  };  
in  
{  
  options.node.role = {  
    web = lib.mkEnableOption "Web server role";  
    db = lib.mkEnableOption "Database server role";  
    cache = lib.mkEnableOption "Cache server role";  
  };  
  
  config = lib.mkMerge [  
    # Базовая конфигурация для всех узлов  
    {  
      environment.systemPackages = with pkgs; [ htop tmux ];  
      services.openssh.enable = true;  
    }  
  
    # Конфигурация для веб-сервера  
    (lib.mkIf roles.web {  
      services.nginx.enable = true;  
      services.phpfpm.enable = true;  
      networking.firewall.allowedTCPPorts = [ 80 443 ];  
    })  
  
    # Конфигурация для БД сервера  
    (lib.mkIf roles.db {  
      services.postgresql.enable = true;  
      services.postgresql.ensureDatabases = [ "appdb" ];  
      networking.firewall.allowedTCPPorts = [ 5432 ];  
    })  
  ]
```

```
# Конфигурация для кэш-сервера
(lib.mkIf roles.cache {
  services.redis.enable = true;
  networking.firewall.allowedTCPPorts = [ 6379 ];
})
];
}
```

?????? ??????????

1. **Ленивые вычисления:** `lib.mkIf` использует ленивые вычисления, поэтому определение вычисляется только если условие истинно.
2. **Обработка ошибок:** Условие должно быть полностью определено. Нельзя использовать значения, которые могут быть `undefined`.
3. **Композиция:** `mkIf` хорошо сочетается с другими функциями библиотеки: `mkMerge`, `mkOption`, `mkDefault`.
4. **Читаемость:** Для сложных условий рекомендуется использовать `let`-блоки для присвоения имен условиям.
5. **Отладка:** Если нужно увидеть, какие условия срабатывают, можно использовать `lib.traceIf` для отладки.

???????? ??????????

```
{ config, lib, ... }:  
{  
  config = lib.mkIf (lib.traceValFn (v: "Condition value: ${toString v}")  
    (config.services.nginx.enable)) {  
    # конфигурация  
  };  
}
```

Revision #1

Created 2025-12-22 08:29:33 UTC by Антон Сергеевич Абраменко

Updated 2025-12-22 08:46:02 UTC by Антон Сергеевич Абраменко